Statistics for Social and Behavioral Sciences

Russell G. Almond
Robert J. Mislevy
Linda S. Steinberg
Duanli Yan
David M. Williamson

Bayesian Networks in Educational Assessment

Springer

Statistics for Social and Behavioral Sciences

Series Editor
Stephen E. Fienberg
Carnegie Mellon University Dept. Statistics
Pittsburgh
Pennsylvania
USA

Statistics for Social and Behavioral Sciences (SSBS) includes monographs and advanced textbooks relating to education, psychology, sociology, political science, public policy, and law.

More information about this series at http://www.springer.com/series/3463

Russell G. Almond • Robert J. Mislevy
Linda S. Steinberg • Duanli Yan
David M. Williamson

Bayesian Networks in Educational Assessment

Springer

Russell G. Almond
Florida State University
Tallahassee
Florida
USA

Robert J. Mislevy
Educational Testing Service
Princeton
New Jersey
USA

Linda S. Steinberg
Pennington
New Jersey
USA

Duanli Yan
Educational Testing Service
Princeton
New Jersey
USA

David M. Williamson
Educational Testing Service
Princeton
New Jersey
USA

Statistics for Social and Behavioral Sciences
ISBN 978-1-4939-2124-9 ISBN 978-1-4939-2125-6 (eBook)
DOI 10.1007/978-1-4939-2125-6

Library of Congress Control Number: 2014958291

Springer New York Heidelberg Dordrecht London

The publisher, the authors and the editors are safe to assume that the advice and information in this book are believed to be true and accurate at the date of publication. Neither the publisher nor the authors or the editors give a warranty, express or implied, with respect to the material contained herein or for any errors or omissions that may have been made.

Printed on acid-free paper

Springer Science+Business Media LLC New York is part of Springer Science+Business Media (www.springer.com)

Dedication

Forward into future times we go
Over boulders standing in out way
Rolling them aside because we know
Others follow in our steps one day

Under deepest earth the gems are found
Reaching skyward ’till we grasp the heights
Climbing up to where the view surrounds
Hidden valleys offer new delights

Inch by inch and yard by yard until
Luck brings us to the hidden vale
Desiring a place to rest yet still
Returning home now to tell the tale

Ever knowing when that day does come
New hands will take up work left undone

Acknowledgements

Bayesian Inference in Educational Assessments (BNinEA) is the direct issue of two projects, and the descendant, cousin, or sibling of many more. We are grateful for all we have learned in these experiences from many collaborators and supporters over the years.

The first direct ancestor is the series of workshops we have presented at the annual meeting of the National Council on Measurement in Education (NCME) almost every year since 2001. We are grateful to NCME for this opportunity and to ETS for support in developing the materials they have granted us permission to use. Workshop participants will recognize many concepts, algorithms, figures, and hands-on exercises from these sessions.

Its second direct ancestor is the Portal project at Educational Testing Service. It was here that Linda, Russell, and Bob fleshed out the evidence-centered design (ECD) assessment framework, implemented it in an object model and design system, and carried out the applications using Bayes nets. We are grateful to Henry Braun and Drew Gitomer, successive vice-presidents of Research, and Len Swanson, head of New Product Development, for supporting Portal. Our collaborators included Brian Berenbach, Marjorie Biddle, Lou DiBello, Howie Chernik, Eddie Herskovits, Cara Cahallan Laitusis, Jan Lukas, Alexander Matukhin, and Peggy Redman.

Biomass (Chaps. 14 and 15) was a ground-up demonstration of Portal, ECD design, standards-based science assessment, web-delivered interactive testing, with automated scoring of inquiry investigations and Bayes net measurement models. The Biomass team included Andy Baird, Frank Jenkins, our subject matter lead Ann Kindfield, and Deniz Senturk. Our subject matter consultants were Scott Kight, Sue Johnson, Gordon Mendenhall, Cathryn Rubin, and Dirk Vanderklein.

The Networking Performance Skill System (NetPASS) project was an online performance-based assessment activity for designing and troubleshooting computer networks. It was developed in collaboration with the Cisco Networking Academy Program, and led by John Behrens of Cisco. NetPASS featured principled design of proficiency, task, and evidence models and a Bayes net psychometric model using the methods described in BNinEA. Team members included Malcolm Bauer, Sarah DeMark, Michael Faron, Bill Frey, Dennis Frezzo, Tara Jennings, Peggy Redman, Perry Reinert, and Ken Stanley. The NetPASS prototype was foundational for the Packet Tracer simulation system and Aspire game environment that Cisco subsequently developed, and millions of Cisco Network Academy (CNA) students around the world have used operationally to learn beginning network engineering skills.

The DISC scoring engine was a modular Bayes-net-based evidence accumulation package developed for the Dental Interactive Simulations Corporation (DISC), by the Chauncey Group International, ETS, and the DISC Scoring Team: Barry Wohlgemuth, DISC President and Project Director; Lynn Johnson, Project Manager; Gene Kramer; and five core dental hygienist members, Phyllis Beemsterboer, RDH, Cheryl Cameron, RDH, JD, Ann Eshenaur, RDH, Karen Fulton, RDH, and Lynn Ray, RDH. Jay Breyer was the Chauncey Group lead, and was instrumental in conducting the expert–novice studies and constructing proficiency, task, and evidence models.

Adaptive Content with Evidence-based Diagnosis (ACED) was the brainchild of Valerie J. Shute. It had a large number of contributors including Larry Casey, Edith Aurora Graf, Eric Hansen, Waverly Hester, Steve Landau, Peggy Redman, Jody Underwood, and Diego Zapata-Rivera. ACED development and data collection were sponsored by National Science Foundation Grant No. 3013202. The complete ACED models and data are available online; see the Appendix for details.

Bob's initial forays into applying Bayesian networks in educational assessment were supported in part by grants from the Office of Naval Research (ONR) and from the National Center for Research on Evaluation, Standards, and Student Testing (CRESST) at the University of California at Los Angeles. We are grateful to Charles Davis, Project Officer of ONR's Model-Based Measurement program and Eva Baker, Director of CRESST, for their support. Much of the work we draw on here appears in ONR and CRESST research reports. The findings and opinions expressed in BNinEA, however, do not reflect the positions or policies of ONR, the National Institute on Student Achievement, Curriculum, and Assessment, the Office of Educational Research and Improvement, or the U.S. Department of Education.

Some of the ideas in this book are based on Russell's previous work on the Graphical-Belief project. Thanks to Doug Martin at StatSci for sponsoring that project as well as the NASA Small Business Innovation Research (SBIR) program for supporting initial development. David Madigan made a number of contributions to that work, particularly pointing out the importance of the weight of evidence. Graphical-Belief is based on the earlier work on Belief while Russell was still a student at Harvard. Art Dempster and Augustine Kong both provided valuable advise for that work. The work of Glenn Shafer and David Schum in thinking about the representation of evidence has been very useful as well. Those contributions are documented in Russell's earlier book.

Along with the NCME training sessions and working on NetPASS and DISC, David completed his doctoral dissertation on model criticism in Bayes nets in assessment at Fordham University under John Walsh, with Russell and Bob as advisors.

Hydrive was an intelligent tutoring system for helping trainees learn to troubleshoot the hydraulics subsystems of the F-15 aircraft. Drew Gitomer was the Principal Investigator and Linda was the Project Manager. The project was supported by Armstrong Laboratories of the US Air Force, under the Project Officer Sherrie Gott. Design approaches developed in Hydrive were extended and formalized in ECD. Bob and Duanli worked with Drew and Linda to create and test an offline Bayes net scoring model for Hydrive.

Russell and Bob used drafts of BNinEA in classes at Florida State University (FSU) and the University of Maryland, respectively. We received much helpful feedback from students to clarify our ideas and sharpen our presentations. Students at Maryland providing editorial and substantive contributions included Younyoung Choi, Roy Levy, Junhui Liu, Michelle Riconscente, and Daisy Wise Rutstein. Students at FSU, providing feedback and advice, included Mengyao Cui, Yuhua Guo, Yoon Jeon Kim, Xinya Liang, Zhongtian Lin, Sicong Liu, Umit Tokac, Gertrudes Velasquez, Haiyan Wu, and Yan Xia.

Kikumi Tatsuoka has been a visionary pioneer in the field of cognitive assessment, whose research is a foundation upon which our work and that many others in the assessment and psychometric communities builds. We are grateful for her permission to use her mixed-number subtraction data in Chaps. 6 and 11.

Brent Boerlage, of Norsys Software Corp., has supported the book in a number of ways. First and foremost, he has made the student version of Netica available for free, which has been exceedingly useful in our classes and online training. Second, he has offered general encouragement for the project and offered to add some of our networks to his growing Bayes net library.

Many improvements to a draft of the book resulted from rigorous attention from the ETS review process. We thank Kim Fryer, the manager of editing services in the Research and Development division at ETS, Associate Editors Dan Eignor and Shelby Haberman, and the reviewers of individual chapters: Malcolm Bauer, Jianbin Fu, Aurora Graf, Shelby Haberman, Yue Jia, Feifei Li, Johnny Lin, Ru Lu, Frank Rijmen, Zhan Shu, Sandip Sinharay, Lawrence Smith, Matthias von Davier, and Diego Zapata-Rivera.

We thank ETS for their continuing support for BNinEA and the various projects noted above as well as support through Bob's position as Frederic M. Lord Chair in Measurement and Statistics under Senior Vice-President for Research, Ida Lawrence. We thank ETS for permission to use the figures and tables they own and their assistance in securing permission for the rest, through Juana Betancourt, Stella Devries, and Katie Faherty.

We are grateful also to colleagues who have provided support in more general and pervasive ways over the years, including John Mark Agosta, Malcolm Bauer, Betsy Becker, John Behrens, Judy Goldsmith, Geneva Haertel, Sidney

Irvine, Kathryn Laskey, Roy Levy, Bob Lissitz, John Mazzeo, Ann Nicholson, Val Shute, and Howard Wainer.

It has taken longer than it probably should have to complete *Bayesian Networks in Educational Assessment*. For their continuing encouragement and support, we are indebted to our editors at Springer: John Kimmel, who brought us in, and Jon Gurstelle and Hannah Bracken, who led us out.

Using This Book

An early reviewer urged us to think of this book not as a primer in Bayesian networks (there are already several good titles available, referenced in this volume), but to focus instead on the application: the process of building the model. Our early reviewers also thought that a textbook would be more useful than a monograph, so we have steered this volume in that particular way. In particular, we have tried to make the book understandable to any reasonably intelligent graduate students (and several of our quite intelligent graduate students have let us know when we got too obscure), as this should provide the broadest possible audience.

In particular, most chapters include exercises at the end. We have found through both our classes and the NCME training sessions, that students do not learn from our lectures or writing (no matter how brilliant) but from trying to apply what they heard and read to new problems. We would urge all readers, even just the ones skimming to try the exercises. Solutions are available from Springer or from the authors.

Another thing we have found very valuable in using the volume educationally is starting the students early with a Bayesian network tool. Appendix A lists several tools, and gives pointers to more. Even in the early chapters, merely using the software as a drawing tool helps get students thinking about the ideas. Of course, student projects are an important part of any course like this. Many of the Bayes net collections used in the example are available online; Appendix A provides the details.

We have divided the book into three parts, which reflect different levels of complexity. Part I is concerned with the basics of Bayesian networks, particularly developing the background necessary to understand how to use a Bayesian network to score a single student. It begins with a brief overview of the ECD. The approach is key to understanding how to use Bayesian networks as measurement models as an integral component of assessment design and use from the beginning, rather than simply as a way to analyze data once it is in hand. (To do the latter is to be disappointed—and it is not the fault of Bayes nets!) It ends with Chap. 7, which goes beyond the basics to start to describe how the Bayesian model supports inference more generally. Part II takes up the issue of the calibrating the networks using data from students. This is too complex a topic to cover in great depth, but this section explores parameterizations for Bayesian networks, looks at updating models from data and model criticism, and ends with a complete example. Part III expands from the focus on mechanics to embedding the Bayesian network in an assessment system. Two chapters describe the conceptual assessment framework and the four-process delivery architecture of ECD in greater depth, showing the intimate connections among assessment arguments, design structures, and the function of Bayesian networks in inference. Two more chapters are then devoted to the implementation of Biomass, one of the first assessments to be designed from the ground up using ECD.

When we started this project, it was our intention to write a companion volume about evidence-centered assessment design. Given how long this project has taken, that second volume will not appear soon. Chapters 2, 12, and 13 are probably the best we have to offer at the moment. Russell has used them with some success as standalone readings in his assessment design class. Although ECD does not require Bayesian networks, it does involve a lot of Bayesian thinking about evidence. Readers who are primarily interested in ECD may find that reading all of Part I and exploring simple Bayes net examples helps deepen their understanding of ECD, then moving to Chaps. 12 and 13 if they want additional depth, and the Biomass chapters to see the ideas in practice.

Several of our colleagues in the Uncertainty in Artificial Intelligence community (the home of much of the early work on Bayesian Networks) have bemoaned the fact that most of the introductory treatises on Bayesian networks fall short in the area of helping the reader translate between a specific application and the language of Bayesian networks. Part of the challenge here is that it is difficult to do this in the absence of a specific application. This book starts to fill that gap. One advantage of the educational application is that it is fairly easy to understand (most people having been subjected to educational assessment at least once in their lives). Although some of the language in the book is specific to the field of education, much of the development in the book comes from the authors' attempt to translate the language of evidence from law and engineering to educational assessment. We hope that readers from other fields will find ways to translate it to their own work as well.

In an attempt to create a community around this book, we have created a Wiki for evidence-centered assessment design (`http://ecd.ralmond.net/ecdwiki/ECD/ECD/`). Specific material to support the book, including example networks and data, are available at the same site (`http://ecd.ralmond.net/BN/BN`). We would like to invite our readers to browse the material there and to contribute (passwords can be obtained from the authors).

Notation

Random Variables

Random variables in formulae are often indicated by capital letters set in italic type, e.g., X, while a value of the corresponding random variable is indicated as a lowercase letter, e.g., x.

Vector-valued random variables and constants are set in boldface. For example, $\mathbf{X}$ is a vector valued random variable and $\mathbf{x}$ is a potential value for $\mathbf{X}$.

Random variables in Bayesian networks with long descriptive names are usually set in italic type when referenced in the text, e.g., *RandomVariable*. If the long name consists of more than one word, capitalization is often used to indicate word boundaries (so-called CamelCase).

When random variables appear in graphs they are often preceded by an icon indicating whether they are defined in the proficiency model or the evidence model. Variables preceded by a circle, ○, are proficiency variables, while variables preceded by a triangle, ▽, are defined locally to an evidence model. They are often but not always observable variables.

The states of such random variables are given in typewriter font, e.g., `High` and `Low`.

Note that Bayesian statistics does not allow fixed but unknown quantities. For this reason the distinction between variable and parameter in classical statistics is not meaningful. In this book, the term "variable" is used to refer to a quantity specific to a particular individual taking the assessment and the term "parameter" is used to indicate quantities that are constant across all individuals.

Sets

Sets of states and variables are indicated with curly braces, e.g., {`High`, `Medium`, `Low`}. The symbol $x \in A$ is used to indicate that x is an element of A. The elements inside the curly braces are unordered, so $\{A1, A2\} = \{A2, A1\}$. The use of parenthesis indicates that the elements are ordered, so that $(A1, A2) \neq (A2, A1)$.

The symbols $\cup$ and $\cap$ are used for the union and intersection of two sets. If A and B are sets, then $A \subset B$ is used to indicate that A is a proper subset of B, while $A \subseteq B$ also allows the possibility that $A = B$.

If A refers to an event, then $\overline{A}$ refers to the complement of the event; that is, the event that A does not occur.

Ordered tuples indicating vector valued quantities are indicated with parenthesis, e.g., $(x_1, \ldots, x_k)$.

Occasionally, the states of a variable have a meaningful order. The symbol $\succ$ is used to state that one symbol is lower than the other. Thus `High` $\succ$ `Low`.

The quantifier $\forall x$ is used to indicate "for all possible values of x." The quantifier $\exists x$ is used to indicate that an element x exists that satisfies the condition.

For intervals of real numbers a square bracket, '[' (']'), is used to indicate that the lower (upper) bound is included in the interval. Thus:

$$\begin{aligned}
[0,1] &\text{ is equivalent to } \{x : 0 \leq x \leq 1\} \\
[0,1) &\text{ is equivalent to } \{x : 0 \leq x < 1\} \\
(0,1] &\text{ is equivalent to } \{x : 0 < x \leq 1\} \\
(0,1) &\text{ is equivalent to } \{x : 0 < x < 1\}
\end{aligned}$$

Probability Distributions and Related Functions

The notation $\mathrm{P}(X)$ is used to refer to the probability of an event X. It is also used to refer to the probability distribution of a random variable X with the hope that the distinction will be obvious from context.

To try to avoid confusions with the distributions of the parameters of distributions, the term *law* is used for a probability distribution over a parameter and the term *distribution* is used for the distribution over a random variable, although the term *distribution* is also used generically.

The notation $\mathrm{P}(X|Y)$ is used to refer to the probability of an event X given that another event Y has occurred. It is also used for the collection of probability distributions for a random variable X given the possible instantiations of a random variable Y. Again we hope that this loose use of notation will be clear from context.

If the domain of the random variable is discrete, then the notation $p(X)$ is used for the probability mass function. If the domain of the random variable is continuous, then the notation $f(X)$ is used to refer to the probability density.

The notation $\mathrm{E}[g(X)]$ is used for the expectation of the function $g(X)$ with respect to the distribution $\mathrm{P}(X)$. When it is necessary to emphasize the distribution, then the random variables are placed as a subscript. Thus, $E_X[g(X)]$ is the expectation of $g(X)$ with respect to the distribution $\mathrm{P}(X)$ and $E_{X|Y}[g(X)]$ is the expectation with respect to the distribution $\mathrm{P}(X|Y)$.

The notation $\mathrm{Var}(X)$ is used to refer to the variance of the random variable X. The $\mathrm{Var}(\mathbf{X})$ is a matrix giving the $\mathrm{Var}(X_k)$ on the diagonal and the covariance of X_i and X_j in the off-diagonal elements.

If A and B are two events or two random variables, then the notation $A \perp\!\!\!\perp B$ and $I(A|\emptyset|B)$ is used to indicate that A is independent of B. The notations $A \perp\!\!\!\perp B \mid C$ and $I(A|C|B)$ indicate that A is independent of B when conditioned on the value of C (or the event C).

The notation $\mathrm{N}(\mu, \sigma^2)$ is used to refer to a normal distribution with mean μ and variance σ^2; $\mathrm{N}^+(\mu, \sigma^2)$ refers to the same distribution truncated at zero (so the random variable is strictly positive). The notation $\mathrm{Beta}(a, b)$ is used to refer to a beta distribution with parameters a and b. The notation $\mathrm{Dirichlet}(a_1, \ldots, a_K)$ is used to refer to K-dimensional Dirichlet distribution with parameters $a_1, \ldots, a_K$. The notation $\mathrm{Gamma}(a, b)$ is used for a gamma distribution with shape parameter a and scale parameter b.

The symbol $\sim$ is used to indicate that a random variable follows a particular distribution. Thus $X \sim N(0, 1)$ would indicate that X is a random variable following a normal distribution with mean 0 and variance 1.

Transcendental Functions

Unless specifically stated otherwise, the expression $\log X$ refers to the natural logarithm of X.

The notation $\exp X$ is used for the expression e^X, the inverse of the log function.

The notation $\operatorname{logit} x$, also $\Psi(x)$, is used for the cumulative logistic function:

$$\operatorname{logit} x = \Psi(x) = \frac{e^x}{1 + e^x} \ .$$

The notation $y!$, y factorial, is used for $y! = \prod_{k=1}^{y} k$, where y is a positive integer.

The notation $\Gamma(x)$ is used for the gamma function:

$$\Gamma(z) = \int_0^\infty t^{x-1} e^{-t} dt \ .$$

Note that $\Gamma(n) = (n-1)!$ when n is a positive integer.

The notation $B(a, b)$ is used for the beta function:

$$B(a, b) = \int_0^1 t^{a-1}(1-t)^{b-1} \, dt = \frac{\Gamma(a)\Gamma(b)}{\Gamma(a+b)}$$

The notation $\binom{n}{y}$ is used to indicate the combinatorial function $\frac{n!}{(n-y)!y!}$. The extended combinatorial function $\binom{n}{y_1 \cdots y_K}$ is used to indicate $\frac{y!}{y_1! \cdots \cdot y_K!}$, where $\sum y_k = n$.

The notation $\Phi(x)$ is used for the cumulative unit normal distribution function.

Usual Use of Letters for Indices

The letter i is usually used to index individuals.

The letter j is usually used to index tasks, with J being the total number of tasks.

The letter k is usually used to index states of a variable, with K being the total number of states. The notation $k[X]$ is an indicator which is 1 when the random variable X takes on the kth possible value, and zero otherwise.

If $\mathbf{x} = (x_1, \ldots, x_K)$ is a vector, then $\mathbf{x}< k$ refers to the first $k-1$ elements of $\mathbf{x}$, $(x_1, \ldots, x_{k-1})$, and $\mathbf{x} > k$ refers to the last $K - k$ elements $(x_{k+1}, \ldots, x_K)$. They refer to the empty set when $k = 1$ or $k = K$. The notation $\mathbf{x}_{-k}$ refers to all elements except the jth; that is, $(x_1, \ldots, x_{k-1}, x_{k+1}, \ldots, x_K)$.

Contents

Part I Building Blocks for Bayesian Networks

Part II Learning and Revising Models from Data

Part III Evidence-Centered Assessment Design

List of Figures

List of Tables

Part I

Building Blocks for Bayesian Networks

1

Introduction

David Schum's 1994 book, *The Evidential Foundations of Probabilistic Reasoning,* changed the way we thought about assessment. Schum, a psychologist cum legal scholar, was writing about evidence in the most familiar meaning of the word, looking at a lawyer's use of evidence to prove or disprove a proposition to a jury. However, Schum placed that legal definition in the context of the many broader uses of the term "evidence" in other disciplines. Schum notes that scientists and historians, doctors and engineers, auto mechanics, and intelligence analysts all use evidence in their particular fields. From their cross-disciplinary perspectives, philosophers, statisticians, and psychologists have come to recognize basic principles of reasoning from imperfect evidence that cut across these fields.

Mislevy (1994) shows how to apply the idea of evidence to assessment in education. Say, for example, we wish to show that a student having completed a reading course, is capable of reading, with comprehension, an article from *The New York Times*. We cannot open the student's brain and observe directly the level of comprehension, but we can ask the student questions about various aspects of articles she reads. The answers provide evidence of whether or not the student comprehended what she read, and therefore has the claimed skill.

Schum (1994) faced two problems when developing evidential arguments in practical settings: uncertainty and complexity. We face those same problems in educational assessment and have come to adopt the same solutions: probability theory and Bayesian networks.

Schum (1994) surveys a number of techniques for representing imprecise and uncertain states of knowledge. While approaches such as fuzzy sets, belief functions, and inductive probability all offer virtues and insights, Schum gravitates to probability theory as a best answer. Certainly, probability has had the longest history of practical application and hence is the best understood. Although other systems for representing uncertain states of knowledge, such as Dempster–Shafer models (Shafer 1976; Almond 1995), may provide a broader

R. G. Almond et al., *Bayesian Networks in Educational Assessment*,
Statistics for Social and Behavioral Sciences, DOI 10.1007/978-1-4939-2125-6_1

palate of states of knowledge that can be modeled, the greater flexibility incurs greater computational cost.

The idea that probability can be used to model an uncertain state of knowledge is often omitted from elementary statistics courses. However, the idea dates back to Bernoulli and other early developers of probability. This use of probability to represent states of knowledge is most often found in the Bayesian approaches to statistics (Chap. 3). It is a powerful concept which allows us to use probability theory to make complex inferences from uncertain and incomplete evidence.

In complex situations, it can be difficult to calculate the probability of an event, especially if there are many dependencies. The solution is to draw a picture. A graphical model, a graph whose nodes represent the variables and whose edges represent dependencies between them, provides a guide for both constructing and computing with the statistical models. Graphical models in which all the variables are discrete, have some particular computational advantages. These are also known as Bayesian networks because of their capacity to represent complex and changing states of information in a Bayesian fashion. Pearl (1988) popularized this approach to represent uncertainty, especially in the artificial intelligence community. Since then, it has seen an explosive growth.

This book explores the implications of applying graphical models to educational assessment. This is a powerful technique that supports the use of more complex models in testing, but is also compatible with the models and techniques that have been developing in psychometrics over the last century. There is an immediate benefit of enabling us to build models which are closer to the cognitive theory of the domain we are testing (Pelligrino et al. 2001). Furthermore, this approach can support the kind of complexity necessary for diagnostic testing and complex constructed response or interactive tasks.

This book divides the story of Bayesian networks in educational assessment into three parts. Part I describes the basics of properties of a Bayesian network and how they could be used to accumulate evidence about the state of proficiency of a student. Part II describes how Bayesian networks can be constructed, and in particular, how both the parameters and structure can be refined with data. Part III ties the mathematics of the network to the evidence-centered assessment design (ECD) framework for developing assessments and contains an extensive and detailed example. The present chapter briefly explores the question of why Bayesian networks provide an interesting choice of measurement model for educational assessments.

1.1 An Example Bayes Network

Bayesian networks are formally defined in Chap. 4, but a simple example will help illustrate the basic concepts.

Example 1.1 (Language Testing Example). *(Mislevy 1995c). Imagine a language assessment which is designed to report on four proficiency variables: Reading, Writing, Speaking and Listening. This assessment has four types of task: (1) a reading task, (2) a task which requires both writing and reading, (3) a task which requires speaking and either reading or listening, and (4) a listening task. Evaluating the work product (selection, essay or speech) produces a single observable outcome variable for each task. These are named* `Outcome R`, `Outcome RW`, `Outcome RLS`, *and* `Outcome L` *respectively.*

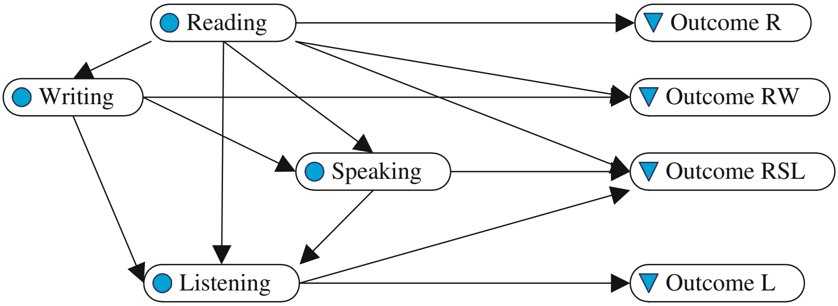


Fig. 1.1 A graph for the Language Testing Example

A Bayesian network for the language test example of Mislevy (1995c). *Rounded rectangles* in the picture represent variables in the model. *Arrows* ("edges") represent patterns of dependence and independence among the variables. This graph provides a visual representation of the joint probability distribution over the variables in the picture. Reprinted with permission from Sage Publications.

Figure 1.1 shows the graph associated with this example. Following conventions from ECD (cf. Chaps. 2 and 12), the nodes (rounded rectangles in the graph) for the proficiency variables are ornamented with a circle, and the nodes for the evidence variables are ornamented with a triangle. The edges in the graph flow from the proficiency variables to the observable variables for tasks which require those proficiencies. Thus the graph gives us the information about which skills are relevant for which tasks, providing roughly the same information that a Q-Matrix does in many cognitively diagnostic assessment models (Tatsuoka 1983).

The graphs used to visualize Bayesian networks, such as Fig. 1.1, act as a mechanism for visualizing the joint probability distribution over all of the variables in a complex model, in terms of theoretical and empirical relationships among variables. This graphical representation provides a shared working space between subject matter experts who provide insight into the cognitive processes underlying the assessment, and psychometricians (measurement experts) who are building the mathematical model. In that sense, Bayesian networks have a lot in common with path analysis and structural equation models.

However, an important difference between Bayesian networks and these other models using similar graphical structures is the way that the Bayes nets encode conditional independence conditions in the graph. Essentially, variables that are separated in the graph are independent given the values of the variables which separate them. (The exact rules depend on what evidence has and has not been observed and are given in Chap. 4.) These conditional independence constraints lead to efficient algorithms for updating probability distributions (cf, Chap. 5; Appendix A lists readily available software packages which implement those algorithms).

As the name implies, Bayesian networks are based on Bayesian views of statistics (see Chap. 3 for a review). The key idea is that a probability distribution holds a state of knowledge about an unknown event. As Bayesian networks represent a probability distribution over multiple variables, they represent a state of knowledge about those variables.

Usually, the initial state of a Bayesian network in educational assessment is based on the distribution of proficiency in the target population for an assessment and the relationship between those proficiencies and task outcomes in the population. The probability values could have come from theory, expert opinion, experiential data, or any mixture of the three. Thus, the initial state of the Bayes net in Fig. 1.1 represents what we know about a student who enters the testing center and sits down at a testing station to take the hypothetical language test: the distribution of proficiencies in the students we typically see, and the range of performance we typically see from these students.

As the student performs the assessment tasks, evaluating the work products using the appropriate evidence rules yields values for the observable outcome variables. The values of the appropriate variables in the network are then instantiated or set to these values, and the probability distributions in the network are updated (by recursive applications of Bayes rule). The updated network now represents our state of knowledge about this student given the evidence we have observed so far. This is a powerful paradigm for the process of assessment, and leads directly to mechanisms for explaining complex assessments and adaptively selecting future observations (Chap. 7). Chapter 13 describes how this can form the basis for an embedded scoring engine in an intelligent tutoring or assessment system.

The fact that the Bayes net represents a complete Bayesian probability model has another important consequence: such models can be critiqued and refined from data. Complete Bayesian models provide a predictive probability for any observable pattern of data. Given the data pattern, the parameters of the model can be adjusted to improve the fit of the model. Similarly, alternative model structures can be proposed and explored to see if they do a better job of predicting the observed data. Chapters 9–11 explore the problems of calibrating a model to data and learning model structure from data.

Bayesian network models for assessments are especially powerful when used in the context of ECD (Mislevy et al. 2003b) Chap. 2 gives a brief introduction to some of the language used with ECD, while Chap. 12 explains it in more detail. The authors have used ECD to design a number of assessment, and our experience has caused us to come to value Bayesian networks for two reasons. First, they are multivariate models appropriate for cognitively diagnostic assessment (Sect. 1.2). Second, they help assessment designers to explicitly draw the connection between measurement models and cognitive models that underlie them (Sect. 1.3).

1.2 Cognitively Diagnostic Assessment

Most psychometricians practicing today work with high-stakes tests designed for selection, placement, or licensing decisions. This is no accident. Errors and inefficiencies in such tests can have high costs, both social and monetary, so it is worthwhile to employ someone to ensure that the reliability and validity of the resulting scores are high. However, because of the prominence of selection/placement tests, assumptions based on the selection/placement purpose and the high stakes are often embedded in the justification for particular psychometric models. It is worth examining closely the assumptions which come from this purpose, to tease apart the purposes, the statistics, and the psychology that are commingled in familiar testing practices.

First, it is almost always good for a selection/placement assessment to be unidimensional. The purpose of a college admission officer looking at an assessment is to rank order the candidates so as to be better able to make decisions about who to admit. This rank ordering implies that the admissions officer wants the candidates in a single line. The situation with licensure and certification testing is similar; the concern is whether or not the candidate makes the cut, and little else.

Because of the high stakes, we are concerned with maximizing the *validity* of the assessment—the degree to which it provides evidence for the claims we would like to make about the candidate. For selection and placement situations, a practically important indicator of validity is the degree to which the test correlates with a measure of the success after the selection or placement. Test constructors can increase this correlation by increasing the *reliability* of the assessment—the precision of the measurement, or, roughly, the degree to which the test is correlated with itself. This can lead them to discard items which are not highly correlated with the main dimension of the test, even if they are of interest for some other reason.

Although high-stakes tests are not necessarily multiple choice, multiple choice items often play a large role in them. This is because multiple choice is particularly cost effective. The rules of evidence—procedures for determining the observable outcome variables—for multiple choice items are particularly easy to describe and efficient to implement. With thoughtful item writing, multiple choice items can test quite advanced skills. Most importantly, they take little of the student's time to answer. A student can solve 20–30 multiple choice items in the time it would take to answer a complex constructed response task like an essay, thus increasing reliability. While a complex constructed response task may have a lower reliability than 20–30 multiple choice items, it may tap skills (e.g., generative use of language) that are difficult to measure in any other way. Hence the complex constructed response item can increase validity even though it decreases reliability.

However, the biggest constraints on high stakes testing come from security concerns. With high stakes comes incentive to cheat, and the measures to circumvent cheating are costly. These range from proctoring and verifying the identity of all candidates, to creating alternative forms of the test. The last of these produces a voracious appetite for new items as old ones are retired. It also necessitates the process of equating between scores on alternative forms of the test.

Increasingly, the end users of tests want more than just a single score to use for selection or placement. They are looking for a set of scores to help diagnose problems the examinee might be facing. This is an emerging field called *cognitively diagnostic assessment* (Leighton and Gierl 2007; Rupp et al. 2010). The "cognitive" part of this name indicates that scores are chosen to reflect a cognitive model of how students acquire skills (see Sect. 1.3). The "diagnostic" part of the name reflects a phenomenon that seeks to identify and provide remedy for some problem in a students' state of proficiency. Such diagnostic scores can be used for a variety of purposes: as an adjunct to a high stakes test to help a candidate prepare, as a guidance tool to help a learner choose an appropriate instructional strategy, or even shaping instructions on the fly in an intelligent tutoring system. Often these purposes carry much lower stakes, and hence less stringent requirements for security.

Nowhere is the interplay between high stakes and diagnostic assessment more apparent than in the *No Child Left Behind (NCLB) Act* passed by the U.S. Congress in 2002 and the Race To the Top program passed as part of the American Reinvestment and Recovery Act of 2009. The dual purpose of assessments—accountability and diagnosis at some level—remains a part of the U.S. educational landscape. Under these programs, all children are tested to ensure that they are meeting the state standards. Schools must be making adequate progress toward bringing all students up to the standards. This, in turn, means that educators are very interested in *why* students are not yet meeting the standards and what they can do to close the gap. They need diagnostic assessment to supplement the required accountability tests to help them identify problems and choose remedies.

When we switch the purpose from selection to diagnosis, everything changes. First and foremost, a multidimensional concept of proficiency usually underlies cognitively diagnostic scoring. (A metaphor: Whereas for a selection exam we might have been content with knowing the volume of the examinee, in a diagnostic assessment we want to distinguish examinees who are tall but narrow and shallow from those who are short, wide and shallow and those who are short, narrow and deep.) As a consequence, the single score becomes a multidimensional profile of student proficiency. Lou DiBello (personal communication) referred to such tests as *profile score* assessments.

The most important and difficult part of building a multidimensional model of proficiency is identifying the right variables. The variables (or suitable summaries) must be able to produce scores that the end users of the assessment care about: scores which relate to *claims* we wish to make about the student and educational decisions that must be made. That is, it is not enough that the claims concern what students know and can do; they must be organized in ways that help teachers improve what they know and can do. A highly reliable and robust test built around the wrong variables will not be useful to end users and consequently will fall out of use.

Another key difference between a single score test and a profile score test is that we must specify how each task outcome depends on the proficiency variables. In a profile score assessment, for each task outcome, we must answer the questions “What proficiencies are required?”; “How are they related in these requirements?”; and “To what degree are they involved?” This is the key to making the various proficiency variables identifiable. In a single score assessment, each item outcome loads only onto the main variable; the only question is with what strength. Consequently, assessment procedures that are tuned to work with single score assessments will not provide all of the information necessary to build a profile score test.

Suppes (1969) introduced a compact representation of the relationship between proficiency and outcome variables for a diagnostic test called the Q-Matrix. In the Q-Matrix, columns represent proficiency variables and rows represent items (observable outcome variables). A one is placed in the cells where the proficiency is required for the item, and a zero is placed in the other cells. Note that an alternative way to represent graphs is through a matrix with ones where an edge is present and zero where there is no edge. Thus, there is a close connection between Bayesian network models and other diagnostic models which use the Q-Matrix (Tatsuoka 1983; Junker and Sijtsma 2001; Roussos et al. 2007b).

The situation can become even more complicated if the assessment includes complex constructed response tasks. In this case, several aspects of a student’s work can provide evidence of different proficiencies. Consequently, a task may have multiple observable outcomes. For example, a rater could score an essay on how well the candidate observed the rules of grammar and usage, how well the candidate addressed the topic, and how well the candidate structured the argument. These three outcomes would each draw upon different subsets of the collection of proficiencies measured by the assessment.

Some of the hardest work in assessment with complex constructed response tasks goes into defining the scored outcome variables. Don Melnick, who for several years led the National Board of Medical Examiners (NBME) project on computer-based case management problems, observed "The NBME has consistently found the challenges in the development of innovative testing methods to lie primarily in the scoring arena. Complex test stimuli result in complex responses which require complex models to capture and appropriately combine information from the test to create a valid score" (Melnick 1996, p. 117).

The best way to do this is to design forward. We do not want to wait for a designer to create marvelous tasks, collect whatever data result, and throw it over the wall for the psychometrician to figure out "how to score it." The most robust conclusion from the cognitive diagnosis literature is this: Diagnostic statistical modeling is far more effective when applied in conjunction with task design from a cognitive framework that motivates both task construction and model structure, than when applied retrospectively to existing assessments (Leighton and Gierl 2007).

Rather, we start by asking what we can observe that will provide *evidence* that the examinee has the skill we are looking for. We build situations with features that draw on those skills, and call for the examinee to say, do, or make something that provides evidence about them—*work products*. We call the key features of this work *observable outcome variables*, and the rules for computing them, *rules of evidence*. For example, in a familiar essay test the observable outcomes are the one or more scores assigned by a rater, and the rules of evidence are the rubrics the rater uses to evaluate the essay as to its qualities.

A richer example is HYDRIVE (Gitomer et al. 1995), an intelligent tutoring system built for the US Air Force and designed to teach troubleshooting for the hydraulics systems of the F-15 aircraft. An expert/novice study of hydraulics mechanics revealed that experts drew on a number of troubleshooting strategies that they could bring to bear on problems (Steinberg and Gitomer 1996). For example, they might employ a test to determine whether the problem was in the beginning or end of a series of components that all had to work for a flap to move when a lever was pulled. This strategy is called "space splitting" because it splits the problem space into two parts (Newell and Simon 1972). HYDRIVE was designed to capture information not only about whether or not the mechanic correctly identified and repaired the problem, but also about the degree to which the mechanic employed efficient strategies to solve the problem. Both of these were important observable outcomes.

However, when there are multiple aspects of proficiency and tasks can have multiple outcomes, the problem of determining the relationships between proficiencies and observable variables becomes even harder. In HYDRIVE, both knowledge of general troubleshooting strategies and the specific system being repaired were necessary to solve most problems. Thus each task entailed a many-to-many mapping between observable outcomes and proficiency variables.

The solution for HYDRIVE was to draw a graph (Mislevy and Gitomer 1996). By drawing arrows from the skills required to the observable outcomes, we could untangle the complex relationships. Furthermore, the joint probability distribution over the variables in the model could be represented with a Bayesian network. This network is a graphical model whose graph represents the relationships between the skills and observations we just described. Expressing our understanding of the problem as a Bayesian network brings with it a number of computational advantages which are described in this book.

ECD grew out of a desire to generalize what worked well about HYDRIVE to other assessments. Since that time, the authors have participated in many design projects using ECD and Bayesian networks, including DISC (Mislevy, Steinberg, et al. 1999b; Mislevy, Steinberg, Breyer, et al. 2002d), Biomass (Steinberg et al. 2003, Chaps. 14 and 15), NetPASS (Behrens et al. 2004), ACED (this book, Chaps. 7 and 13; Shute 2004; Shute et al. 2005; Shute et al. 2008), an alternative scoring method for ETS's ICT Literacy assessment (Katz et al. 2004), and a game-based assessment called SimCityEDU (Mislevy et al. 2014).

1.3 Cognitive and Psychometric Science

The HYDRIVE experience taught us many lessons. Among them was the amount of work required to build a diagnostic assessment that truly relates to variables learners and educators care about. Building such assessments consistently and in a cost effective way demands an approach to assessment design that supports many kinds of assessments, both familiar selection assessments and new kinds of diagnostic assessments. Furthermore, it requires a philosophy of assessment design that would provide a framework for answering questions when new problems inevitably arise.

As we based this new approach on the principle of finding evidence for the knowledges, skills, and abilities we were testing, we called this approach *ECD*. The basic approach can be laid out in four steps:

1. Gather together the *claims* we wish to make about an examinee, for example, "An examinee who scores highly can pick up and read with comprehension a journal article written in English in their field of expertise."
2. Organize these claims into a *proficiency model*, constructing variables representing the knowledges, skills, and abilities required to meet the claims.
3. Determine what we could *observe* in a candidate's work which would provide *evidence* that the candidate did (or did not, or did to what extent or in what way) have a particular complex of proficiencies.
4. Structure *tasks* which will provide us with the opportunity to make those kinds of observations.

Note that this design philosophy has the validity argument built right in. Ultimately, validity is how well the scores from an assessment support the claims the assessment makes. In an ECD assessment, this argument is the central core. Task results provide observable outcomes that provide evidence about proficiency variables which support the claims of the assessment. Any valid assessment obeys this principle, but ECD forces us to think about it from the very start of the assessment design process. Other authors who have thought about assessment design in similar ways include Susan Embretson (Embretson 1998), who focuses on psychological measurement, Grant Wiggins (Wiggins 1998), who focuses on instructional assessment, and Ric Luecht (Luecht 2012), who focuses on the re-usability of task design, delivery, and scoring components.

The proficiency model for HYDRIVE was based on the cognitive theory of the domain. This cognitive basis simplified the process of designing instructions to help learners acquire the knowledge, skills, and abilities underlying the proficiency variables. It further ensured that reporting would be in terms of concepts that were useful for the intended audience.

Contrast this to trait theories of psychology in which the latent trait being measured is effectively defined in terms of how it is measured. Take for example IQ tests, in which intelligence has been defined as being what the IQ test measures. This is unsatisfactory in that it is difficult to see how to provide training to increase one's intelligence.

Furthermore, the trait theory breaks down as we introduce multiple traits. When the traits are defined after the fact, there is a question of identifiability (rotational indeterminacy is the bane of factor analysis!). We can always relabel the traits to create a new "theory." Obviously, one needs a better way to identify the variables in the proficiency model.

Bayesian networks and evidence-centered design support multidimensional models in terms of cognitive theory—specifically, with models motivated by an information-processing perspective, but posited as a working approximation of proficiency in some domain rather than an authentic representation of the details of cognition. Variables are introduced to stand for aspects of an examinee's proficiency—elements of knowledge, strategies and procedures, tendencies to solve problems that have certain properties, and so on. By using probabilities to represent the examiner's uncertain state of knowledge about an examinee's proficiency variables, Bayesian networks can represent the theory of the domain, modeling complex relationships among proficiency variables. Bayesian networks can model quite complex relationships between observable and proficiency variables. These evidence models allow us to update our knowledge about a student's proficiency as more evidences (in the form of observations from tasks) arrive. The Bayesian network tracks our state of knowledge about a particular student. It starts with general knowledge based on the population of students that this individual is drawn from. Part I of this book describes how Bayesian networks can be used as a mechanism to update our knowledge about that student as more and more evidence arrives.

Part II discusses learning and revising models from data. When the model is based on a cognitive theory, these activities take on additional importance. Data which fit the model fairly well provide support for the underlying cognitive theory. Data which do not fit well provide information about potential gaps or weaknesses in the theory. The assessment designer is prompted to consider revising the theory, the statistical model, or the way data are being collected and interpreted. When the statistical model and cognitive theory mirror one another, developments in one can be reflected in the other.

Often, the problem with cognitive theories is that it is difficult or expensive to measure the knowledge structures they posit. In our experience there are two types of expertise that test designers bring to bear on the assessment design process. One is the knowledge of the cognitive processes of the domain, that is, of the proficiency model. The second is the knowledge about how to structure tasks, including what makes tasks easier and harder, and which aspects of knowledge or skills they tap. Often it is difficult to make the connection between these two types of expertise.

Evidence is the bridge between theories about tasks and theories about proficiency. Asking "How can I get evidence that this subject has this proficiency?" leads to designs for tasks to assess that proficiency. Similarly, by asking "What knowledge, skills, and abilities are required to perform this task?" we can understand what it is that the task provides evidence for. By driving forward and backward over this bridge we can iteratively build assessments that reflect our cognitive theory.

Although we are excited about the potential of graphical models to model a broad range of cognitive theories, we cannot get around some of the fundamental laws of psychometrics. First, no matter what we claim that the assessment is measuring, it effectively measures the ability to perform tasks like the ones in the assessment. Only if we have built the tasks with fidelity to the construct(s) we are trying to measure will the assessment provide evidence to support its claims. Furthermore, a certain amount of evidence is required to support each claim we are trying to make. For example, it would be very difficult to provide enough evidence to support a proficiency model with 30 variables on a test that contains only 10 items (unless those items represented large, complex tasks with many parts, and complex rules of evidence were used to pull out many partially-dependent bits of evidence). Thus, a fairly sophisticated knowledge of the strengths and limitations of the models we are proposing is required to construct graphical models for use in educational assessment.

1.4 Ten Reasons for Considering Bayesian Networks

Evidence-centered design, as set out in Mislevy et al. (2003b) and elsewhere, is neutral to the measurement model. The principles apply whether Bayesian networks, latent class analysis, classical test theory, factor analysis or item response theory and its many extensions are used to score the assessment. However, in applications, Bayes nets have been our first choice for the measurement model. Bayes nets enjoy a number of advantages, some immediately visible to test users, some under the hood. Other methods may share some of the properties, but the combination offered by Bayes nets is unique.

1. *Bayes nets report scores in terms of "Probability of Claim."* When built using evidence-centered design, each level of each proficiency variable in a Bayes net is associated with one or more ECD claims. Thus, the natural score report provides the probability that the claim holds. This is exactly the kind of information that test users need to make instructional planning decisions. It suggests that using the Bayes nets as a part of an artificial intelligence planning system would make a powerful engine for an intelligent tutoring system. However, even in the simpler world of human instructional planning system, the kind of score reports described here were thought to be useful by a focus group of score users (Jody Underwood, unpublished focus group results).
2. *Bayes nets use a graphical representation for the proficiency model.* Bayes nets take their name from the network diagram or graph they use to describe the relationship among the variables. This graph provides both a rigorous mathematical description of the model and an informal schematic description. This representation helps facilitate conversations between cognitive and measurement experts. However, it goes deeper than that. Daniel et al. (2003) suggest that even secondary school students find this representation valuable, and it can be used to facilitate a dialogue between the instructor and the learner.
3. *Bayes nets can incorporate expert knowledge about the cognitive domain.* Expert input is needed in any model building exercise, particularly in the difficult steps of defining the variables and the relationships among them—the "graphical structure" of the Bayesian network. Bayes net modeling encourages cognitive experts to get involved in the process. This means the structure of the model can be well suited to a particular purpose; for example, a proficiency model can be built to reflect a particular instructional theory about a domain. It also means that Bayes nets can take full advantage of other information gathered during an ECD design process, such as prior opinions about the difficulty of a task.
4. *Bayes nets can "learn" from data.* Bayes nets are probability models. This indicates that they make probabilistic predictions about what will happen. It also means that there is a measure for how well observed data meet the expectations of the model. This property can be used to improve the

original model as more and more data become available. This learning can be used to both adjust the parameters of the model and suggest changes to the structure. The latter is instructive as it gives us feedback into the cognitive models, which form the basis of the Bayes net. These last two points taken together give many possible strategies for constructing Bayes nets: from building networks entirely from expert opinion with no pretesting, to building networks entirely from pretest data, and any number of combinations of the two.

5. *Bayes nets handle complex models and tasks.* Complex tasks (e.g., simulations, multistep problems, and complex constructed response) are in high demand for assessments because they both feel more authentic and they can tap higher order, constructive, and interactive skills that are difficult to capture with simpler tasks. Bayes nets tackle large problems by parsing our reasoning about them as combinations of smaller more manageable chunks. Specifically, Bayes nets can model multiple dependent observables coming from complex tasks, each of which can provide evidence about different aspects of proficiency. All that is necessary to score such a task using Bayes nets is to specify the model. Fitting data with such complex dependencies is challenging for all measurement models, but Bayes nets offer an approach which reflects the cognitive model based on what happens during the task.
6. *Bayes nets are fast.* By using only discrete variables, Bayes nets can obtain exact, closed-form solutions to problems which would require numeric approximations to difficult integrals using other methods. This means that Bayes nets models can be updated very quickly, making them suited for embedded scoring engines (Biomass, NetPASS and ACED are all prototype systems using Bayes net scoring; ACED is even adaptive, and SimCityEDU is fully interactive). Paradoxically, Bayes net models have a reputation for being slow because their speed tempts designers to try larger models than they would using other methods.
7. *Bayes nets provide profile scores.* Bayes nets will provide scores on as many variables as are available in the proficiency model. This means that Bayes nets can provide subscores on dimensions that are meaningful according to the underlying cognitive model. It also indicates that Bayes nets can handle integrated tasks which address more than one proficiency. Furthermore, Bayes net can assess higher-level skills (such as science inquiry skills in Biomass) in ways that obtain evidence about lower-level skills, and partialling it out to understand what can be learned about the higher-level skills.
8. *Bayes nets provide real-time diagnosis.* Because Bayes nets provide profile information quickly, they can be queried at any time in an assessment situation. In particular, an intelligent tutoring system can use Bayes nets to make decisions about when to continue assessment, when to switch to instruction and what instruction would be expected to provide the most value.

9. *Building Bayes nets is natural in the context of evidence-centered design.* It seems like there is a lot of information that needs to go into the construction of a Bayesian network. However, much of this information must be generated in the context of the assessment design no matter which measurement model is eventually used to score the assessment. This is especially true in the context of complex tasks, where often ad hoc scoring software must be built to either score the whole assessment or simplify the observed outcomes so that the outcomes can be analyzed with an off-the-shelf measurement model. Essential questions such as which proficiencies are relevant for which tasks, and what scores will be reported must be answered no matter what the measurement model. Building Bayes nets requires only that the questioning goes slightly deeper, asking the experts questions about the strength of the relationships. With the ECD design perspective and ECD design tools, much of the work of building the Bayesian network flows from the process of designing the assessment.
10. *Bayes net models are "useful."* The statistician George Box (1976) stated "All models are false, but some models are useful." Bayes nets built using ECD fall into what Berliner (2005) called "Bayesian hierarchical modeling." In particular, they incorporate a probabilistic data model and a process model built around our cognitive understanding of the domain to be assessed. Berliner claims that "Simple models in conjunction with Bayesian hierarchical modeling may be better than either 'more faithful models' or 'statistical models.' " The most useful model for every purpose may not be a Bayes net, but Bayes nets will often be a worthwhile place to look for useful models.

1.5 What Is in This Book

This book gathers together in one place many of the ideas and structures that form the psychometric underpinnings of evidence-centered design. It concentrates on the mathematical models underlying evidence-centered design, in particular, the use of graphical models to represent cognitive models of a domain. It talks about how to build the models, score assessments with them, and refine them from data.

This book is not a complete description of ECD. In particular, it does not deal with many of the aspects of how one designs a cognitive model of a domain and then refines it for assessment. That part of the story will be left for a future book, although much of it has been laid out in a series of papers (Behrens et al. 2012, Mislevy et al., 1999b; Mislevy et al. 2003c; Mislevy et al. 2002c, 2003b; Mislevy et al. 2006). Chapters 2, 12, and 14 touch upon some of these broader issues, but mostly in service of setting the context for the more mathematical work.

The goal of this book is to give the reader enough familiarity with Bayesian networks and other graphical models to be able to build models to mirror a particular cognitive theory and support it with an assessment. It deliberately takes a model construction rather than computational emphasis. Many other texts (e.g., Pearl 1988; Jensen 1996; Almond 1995; Cowell et al. 1999; Neapolitan 1990; Neapolitan 2004) cover the computational aspects. Our goal is to give enough background to discuss the implications for model construction and understand the connections with other literature working with graphical models.

The book is designed to be used as a textbook for a graduate student in education or psychology. We have assumed that the student has taken at least one course in probability and statistics, or a discipline-based course with a strong statistical component. We also assume some familiarity with testing, in particular, item response theory. Although this is not a prerequisite per se, many of the examples and explanations draw on ideas from item response theory. We have added exercises at the end of the chapters to facilitate its use as a textbook. Chapters 6 and 11 concentrate on extensive data sets which can be used for larger projects. Appendix A contains links to software tools for working with Bayes nets and links to data sets which are available for classroom projects, including an online glossary to provide quick reference to definitions for terms relating to ECD and Bayesian networks.

The chapters are organized into three parts. Part I lays out the basic definitions and key ideas from the theory of graphical models that will be used throughout the rest of the book. Part II moves to the challenges of building, estimating, and revising models from data. Part III brings these tools fully to bear on problems of educational assessment.

Part I is not meant to be a complete course in Bayesian networks, but we have included enough of the essential definitions and algorithms to enable readers to follow the applications in educational testing. Readers looking for a more complete introduction should see Pearl (1988) or Jensen (1996). Readers looking for a more mathematical treatment should see Whittaker (1990) or Lauritzen (1996).

Chapter 2 provides an overview of evidence-centered design both to motivate the subsequent mathematics and to introduce some terms that will not be formally defined until Chap. 12. Chapter 3 provides a review of probability and Bayesian statistics, with careful attention to representing states of knowledge with probability distributions. Chapter 4 provides some of the basic definitions of graph theory and Bayesian networks, paying particular attention to the representation of conditional independence with graphs. Chapter 5 describes the basic algorithms for moving probability around graphs, and how we can use them to draw inferences about students' proficiencies based on the outcomes of assessment tasks. Chapter 6 looks at some examples of the application of Bayesian networks to educational testing. Chapter 7 defines the concept of *weight of evidence* and how it can be used to both explain scores and select items.

While Part I concentrates on models for a single learner, Part II discusses what can be done with data from many learners. Chapter 8 describes the parameters used in these models, and introduces some models with reduced parameters. Chapter 9 describes the EM algorithm and Markov Chain Monte Carlo (MCMC) estimation, the techniques we have been using to fit models to data. This is by no means a complete treatment for either of the two (e.g., Gilks et al. (1996) and Lynch (2007) provide good starting points for MCMC). Chapter 10 looks at the problem of diagnosing problems with the model from patterns in the data. Learning models from data is a natural extension of model criticism, and the chapter includes a brief survey of the literature on learning models. Chapter 11 looks at our experiences in applying these ideas to one particular example.

While the first two parts primarily address the mathematical aspects of using Bayesian networks as a measurement model for an assessment, Part III ties the mathematics back to psychology and the assessment design. Chapter 12 defines the basic design elements of evidence-centered design and describes the construction of a model. Chapter 13 shows how to use the mathematics to build a scoring engine for an online assessment or intelligent tutoring system.

Chapters 14 and 15 explore a prototype assessment for high school transmission genetics, named Biomass. Chapter 14 provides a sketch of how the system was designed; in particular, how the proficiency model was constructed from national science standards, how tasks were developed, and observable variables were defined for assessing higher order skills that involve applying the scientific method in the context of biology. Chapter 15 explores the construction of the Bayesian network scoring engine for Biomass, both how it was constructed from expert opinion and how pilot data could be used to update the model parameters.

The last chapter reviews some of what we have learned in the course of applying graphical models to educational assessment. This field is very new, and there is a lot more to learn. Many parts of psychometrics that have been well explored in the context of item response theory (IRT) and classical test theory remain to be developed for graphical models. The final chapter surveys some of these research frontiers.

2

An Introduction to Evidence-Centered Design

Although assessment design is an important part of this book, we do not tackle it in a formal way until Part III. Part I builds up a class of mathematical models for scoring an assessment, and Part II discusses how the mathematical models can be refined with data. Although throughout the book there are references to cognitive processes that the probability distributions model, the full discussion of assessment design follows the discussion of the more mathematical issues.

This presents two problems. First, a meaningful discussion of the statistical modeling of the assessment requires a basic understanding of the constraints and affordances of the assessment design process. The second is that the discussion of the statistical models and processes requires certain technical terms, in particular, *proficiency model, evidence model, task model,* and *assembly model,* that are not formally defined until Chap. 12. This chapter provides brief working definitions which will be sufficient to describe the mathematical models, leaving the more nuanced discussion of assessment design until after the mathematical tools have been defined.

Evidence-centered design (*ECD*) is an approach to constructing educational assessments in terms of evidentiary arguments. This chapter introduces the basic ideas of ECD, including some of the terminology and models that have been developed to implement the approach. In particular, it presents the high-level models of the Conceptual Assessment Framework (see also Chap. 12) and the four-process architecture for assessment delivery systems (see also Chap. 13). Special attention is given to the roles of probability-based reasoning in accumulating evidence across task performances, in terms of belief about unobservable variables that characterize the knowledge, skills, and/or abilities of students. This is the role traditionally associated with psychometric models, such as item response theory and latent class models. Later chapters will develop Bayesian network models which unify the ideas and provide a foundation for extending probability-based reasoning in assessment applications more broadly. This brief overview of evidence-centered design,

R. G. Almond et al., *Bayesian Networks in Educational Assessment*,
Statistics for Social and Behavioral Sciences, DOI 10.1007/978-1-4939-2125-6_2

then, provides context for where and how graphical models fit into the larger enterprise of educational and psychological assessment.

2.1 Overview

All educational assessments have in common the desire to reason from particular things students say, do, or make, to inferences about what they know or can do more broadly. Over the past century a number of assessment methods have evolved for addressing this problem in a principled and systematic manner. The measurement models of classical test theory and, more recently, item response theory (IRT), latent class analysis, and cognitive diagnosis modeling, have proved quite satisfactory for the large scale tests and classroom quizzes with which every reader is by now quite familiar.

But off-the-shelf assessments and standardized tests are increasingly unsatisfactory for guiding learning and evaluating students' progress. Advances in cognitive and instructional sciences stretch our expectations about the kinds of knowledge and skills we want to develop in students, and the kinds of observations we need to evidence them (Pelligrino et al. 2001; Moss et al. 2008). Advances in technology make it possible to evoke evidence of knowledge more broadly conceived, and to capture more complex performances. One of the most serious bottlenecks we face, however, is making sense of complex data that result.

Fortunately, advances in evidentiary reasoning (Schum 1994) and in statistical modeling (Gelman et al. 2013a) allow us to bring probability-based reasoning to bear on the problems of modeling and uncertainty that arise naturally in all assessments. These advances extend the principles upon which familiar test theory is grounded to more varied and complex inferences from more complex data (Mislevy 1994).

We cannot simply construct "good tasks" in isolation, however, and hope that someone else down the line will figure out "how to score them." We must design a complex assessment from the very start around the inferences we want to make, the observations we need to ground them, the situations that will evoke those observations, and the chain of reasoning that connects them (Messick 1994). We can expect iteration and refinement as we learn, from data, whether the patterns we observe accord with our theories and our expectations; we may circle back to improve our theories, our tasks, or our analytic models (Mislevy et al. 2012). But the point is that while more complex statistical models may indeed be required, they should evolve from the substance of the assessment problem, jointly with the purposes of the assessment and the design of tasks to provide observable evidence.

ECD lays out a conceptual design framework for the elements of a coherent assessment, at a level of generality that supports a broad range of assessment types, from familiar standardized tests and classroom quizzes, to coached practice systems and simulation-based assessments, to portfolios and student–tutor interaction. The design framework is based on the principles of evidentiary reasoning and the exigencies of assessment production and delivery. Designing assessment products in such a framework ensures that the way in which evidence is gathered and interpreted bears on the underlying knowledge and the purposes the assessment is intended to address. The common design architecture further aids coordination among the work of different specialists, such as subject matter experts, statisticians, instructors, task authors, delivery-process developers, and interface designers. While the primary focus of the current volume is building, fitting, testing, and reasoning with statistical models, this short chapter places such models into the context of the assessment enterprise. It will serve to motivate, we hope, the following chapters on technical issues of this sort. After that machinery has been developed, Chap. 12 returns to ECD, to examine it more closely and work through some examples.

Section 2.4 describes a set of models called the *Conceptual Assessment Framework*, or *CAF*, and the *four-process architecture* for assessment delivery systems. The CAF is not itself the assessment design process, but rather the end product of the assessment design process. Although this book does not cover the earlier stages of the design process, Sect. 2.3 touches on them briefly. Mislevy, Steinberg, and Almond (2003b) present a fuller treatment of ECD including connections to the philosophy of argument and discussions of the earlier stages of design. Almond et al. (2002a) and Almond et al. (2002b) amplify the delivery system architecture and its connection to the design.

One of the great strengths of evidence-centered design is that it provides a set of first principles, based on evidentiary reasoning, for answering questions about assessment design. Section 2.2 provides a rationale for assessment as a special case of evidentiary reasoning, with validity as the grounds for the inferences drawn from assessment data (Cronbach 1989; Embretson 1983; Kane 1992; Kane 2006; Messick 1989; Messick 1994; Mislevy 2009). ECD provides a structural framework for parsing and developing assessments from this perspective.

2.2 Assessment as Evidentiary Argument

Advances in cognitive psychology deepen our understanding of how students gain and use knowledge. Advances in technology make it possible to capture more complex performances in assessment settings, by including, for example, simulation, interactivity, collaboration, and constructed responses in digital form. Automated methods have become available for parsing complex work products and identifying educationally meaningful features of them Williamson et al. (2006b).

The challenge is in knowing just how to put all this new knowledge to work to best serve the purposes of an assessment. Familiar practices for designing and analyzing single-score tests composed of familiar items are useful because they are coherent, but the schemas are limited to the constraints under which they evolved—the kinds of tasks, purposes, psychological assumptions, cost expectations, and so on that define the space of tests they produce. Breaking beyond the constraints requires not only the means for doing so (through advances such as those mentioned above) but schemas for producing assessments that are again coherent, but in a larger design space; that is, assessments that may indeed gather complex data to ground inferences about complex proficiency models, to gauge multidimensional learning or to evaluate multifaceted programs—but which are built on a sound chain of reasoning from what we propose to observe to what we want to infer. We want to design in reverse direction: What do we want to infer? What then must we observe in what kinds of situations, and how are the observations interpreted as evidence?

Recent work on validity in assessment lays the conceptual groundwork for such an approach. The contemporary view focuses on the support—conceptual, substantive, and statistical—that assessment data provide for inferences or actions (Messick 1989). From this view, an assessment is a special case of evidentiary reasoning. Messick (1994) lays out the general form of an assessment design argument in the quotation below. (We will look more closely at assessment arguments in Sect. 12.1.2.)

> A construct-centered approach [to assessment design] would begin by asking what complex of knowledge, skills, or other attribute should be assessed, presumably because they are tied to explicit or implicit objectives of instruction or are otherwise valued by society. Next, what behaviors or performances should reveal those constructs, and what tasks or situations should elicit those behaviors? Thus, the nature of the construct guides the selection or construction of relevant tasks as well as the rational development of construct-based scoring criteria and rubrics (p. 17).

This perspective organizes thinking for designing assessments for all kinds of purposes, using all kinds of data, task types, scoring methods, and statistical models. An assessment interpretation reasons from what we observe to what we then believe about students' proficiencies. Assessment design reasons in the reverse direction, laying out the elements of an assessment in a way that will support the needed interpretations.

For the purpose of the assessment, what are the proficiencies we are interested in? In what situations do people draw on them, to accomplish what ends, using what tools and representations, and producing what kinds of outcomes? Taking context and resources into account, we consider task situations we can devise and observations we can make to best ground our inferences. If interactions are key to getting evidence about some proficiency, for example, we can delve into what features a simulation must contain, and what the student must be be able to do, in order to exhibit the knowledge and skills we care about. We craft scoring methods to pick up the clues that will then be present in performances. We construct statistical models that will synthesize evidence across multiple aspects of a given task performance, and across multiple task performances. These decisions in the assessment design process build the inferential pathway we then follow back from examinees' behaviors in the task setting to inferences about what they know or can do. From an evidentiary reasoning perspective, we can examine the impact of these design decisions on the inferences we ultimately want to make.

As powerful as it is in organizing thinking, simply having this conceptual point of view is not as helpful as it could be in carrying out the actual work of designing and implementing assessments. A more structured framework is needed to provide common terminology and design objects that make the design of an assessment explicit and link the elements of the design to the processes that must be carried out in an operational assessment. Such a framework not only makes the underlying evidentiary structure of an assessment more explicit, but it makes it easier to reuse and to share the operational elements of an assessment. The evidence-centered design models address this need.

2.3 The Process of Design

The first step in an assessment design is to establish the *purpose* of the assessment. Many fundamental design trade-offs, e.g., assessment length versus reliability, breadth across multiple aspects of proficiency versus depth in a single proficiency, are ultimately resolved by deciding how to best meet the purpose of the assessment. Fixing the purpose of the assessment early in the process has a marvelous focusing effect on the design and development processes.

Fixing the purpose, however, is easier said than done. Different test users may have different and competing purposes in mind for a proposed assessment. Expectations can be unrealistic, and can change over time. The purpose of an assessment often starts as somewhat vague in the beginning of the design process and becomes further refined as time goes on.

The ECD framework describes the assessment design process in three stages: *domain analysis*—gathering and organizing information related to the cognitive background of the assessment as well as the purposes and constraints of the design process; *domain modeling*—building a preliminary sketch of the assessment argument as a general, reusable framework for a family of possible assessments; and the *conceptual assessment framework*—filling in the details of the initial sketch, particularly resolving design decisions to focus the product on a particular purpose.

The lines between requirements-gathering, analysis, design, and implementation are difficult to draw (indeed, the authors have argued among themselves about which of the steps of the ECD process correspond to which steps of the general engineering workflow). Describing the ECD process in phases might seem to suggest a waterfall development process, where each stage flows into the next and the flow is just one way. Real-world assessment design processes are usually iterative, with prototypes and cycles; things learned at later stages of design often prompt the designer to revisit, rethink, and revise work done at the earlier stages. Mislevy, Steinberg, and Almond (2003b) discussed the ECD design process in more detail.

For the most part, this book does not delve deeply into these design issues so that it can focus on the theory, the roles, and the mechanics of Bayesian networks in the assessment argument. Most of the examples assume that the conceptual assessment framework has already been specified. Only with the Biomass example of Chaps. 14 and 15 do we work through the design process from the very beginning: from targeted educational standards, through the CAF, to the innovative, interactive tasks, and a Bayes nets scoring model that result from the unified design process. It does not hurt to say again, though, that complex measurement models such as Bayesian networks will provide the greatest value when they arise from a principled design process to serve an evidentiary argument, rather than applied retrospectively to data that are collected without clear hypotheses connecting proficiencies and the situations and performances that reveal them (iteration and refinement notwithstanding).

Section 2.4, then, describes the basic design objects of the CAF. The domain-model design objects are basically lighter weight versions of their CAF counterparts; detailed enough to support the assessment argument, but not yet detailed enough to support implementation. In the domain modeling phase, the design team are encouraged to think about how the assessment argument would play out for multiple purposes and in multiple settings. It helps to identify opportunities in which argument structures from one assessment can be reused in another.

One kind of design object, developed in the early stages of the design process but used extensively in the CAF, is the *claim*. A claim is a statement about a participant that the assessment will provide evidence for (or against). Claims are important because they give clarity to the purpose of an assessment. One of the most important design decisions is deciding which claims will be the primary focus of an assessment. Indeed, the whole question of validity could be framed as determining to what extent an assessment really supports its claims.

A simple example, used through the rest of the chapter, illustrates these ideas.

Example 2.1 (Calculus Placement Exam). *University C requires all students to take 2 years of calculus, in the form of a two-semester freshman sequence followed by a two-semester sophomore sequence. Typically a student starts with the first semester in the freshman year, but some students (particularly those who took an advanced calculus class in high school) start with the second semester, or with the third semester with the sophomore cal-*

culus class. Some students do not have the necessary background to begin the sequence, and should take a precalculus remedial course first. University C administers a placement exam to all incoming freshmen to determine how to best place them into the calculus sequence.

Claims in this assessment are based on the student having proficiencies that are addressed in each of the courses in the calculus series. Examples include, "Student can integrate functions of one variable," and "Student can find partial derivatives of multivariate functions." Note that there may be competing interest in the claims. For example, the Physics department may have more interest in the claim "Student can solve integrals in two and three dimensions" while the Math department is more interested in the claim "Student can construct a valid mathematical proof."

Often claims are arranged hierarchically. For example, the claim "Student can integrate functions of one variable" involves the subclaims, "Student can integrate polynomial functions" and "Students can integrate trigonometric functions" as well as the subclaims "Student can use transformation of variables to solve integrals" and "Student can use partial fractions to solve integrals." "Student can construct a valid mathematical proof" will need further specification with respect to the particular models and the kind of the proof at issue (e.g., existence proof, induction, construction, proof by contradiction). It will be seen that a set of claims is not sufficient to determine the proficiency model for a given purpose. Composite claims that bundle finer-grained claims dealing with skills in the same semester are good enough for course placement, but the finer-grained claims would be distinguished for quizzes and diagnostic tests during a semester.

In this particular case, the claims are relatively easy to establish. They will fall naturally out of the syllabus for the calculus series and the calculus text books. They are not simply a list of topics, but rather the kinds of problems, proofs, and applications a student is expected to be able to carry out.

Another frequent source of claims is the educational standards published by states and content area associations, such as the *Next Generation Science Standards* (NGSS Lead States 2013). Grain size and specificity vary from one set of standards to another, and often they need to be refined or clarified to take the form of claims. They may not be phrased in terms of targeted capabilities of students, or indicate what kinds of evidence is needed. It is not enough say, for example, that "Student understands what constitutes a valid mathematical proof." Chapter 14 provides an example of moving from standards to a framework of claims to ground an assessment.

Claims play two key roles in domain modeling: (1) including and excluding specific claims clarifies the purpose of the assessment, and (2) laying them out starts the process of developing an assessment argument. These roles are so important that while most domain modeling design objects are refined and expanded in the CAF, claims remain largely in their initial form.

2.4 Basic ECD Structures

In an ECD, an assessment design is expressed through a collection of objects called the CAF. In any particular assessment, the objects in the CAF models described in general terms in Sect. 2.4.1 will need to have been designed to address the purposes of that particular assessment. In line with the Messick quotation cited above, the characteristics of tasks have been selected to provide the opportunity to get evidence about the targeted knowledge and skills (i.e., the claims); the scoring procedures are designed to capture, in terms of observable variables, the features of student work that are relevant as evidence to that end; and the characteristics of students reflected as proficiency variables summarize evidence about the relevant knowledge and skills from a perspective and at a grain size that suit the purpose of the assessment. The CAF models provide the technical detail required for implementation: specifications, operational requirements, statistical models, details of rubrics, and so on.

CAF models provide specifications, but specifications are not an assessment. As examinees and users of assessment ourselves, we see activities: Tasks being administered, for example, and students interacting with task contexts to produce essays or solve problems, raters evaluating performances or automated algorithms evaluating work, score reports being generated, and feedback being given to students in practice tests. We will organize all of this activity in terms of *processes*, as described below. It is the CAF that specifies the *structure* and the *relationships* of the all content, messages, and products involved in the processes. In other words, the CAF lays out the structural elements of an assessment that embody an assessment argument. The delivery processes described below bring the assessment to life. They are real-world activities that interact with students, gather evidence, and support inference using those structures.

In describing both the design and implementation of scoring models and algorithms, it is useful to have a generic model of the assessment delivery process. Section 2.4.2 describes the four-process architecture that forms a reference model for the delivery of an assessment. The four processes of the delivery system carry out, examinee by examinee, the functions of selecting and administering tasks, interacting as required with the examinee to present materials and capture work products, then evaluating responses from each task and accumulating evidence across them. The information in the CAF models specs out details of the objects, the processes, and the messages that are all interacting when an assessment is actually in play. Any real assessment must have elements that correspond to the four processes in some way. Thus, exploring how assessment ideas play out in the four process framework provides an understanding about how they will play out in specific assessment implementations.

2.4.1 The Conceptual Assessment Framework

The blueprint for an assessment is called the CAF. To make it easier to rearrange the pieces of the framework (and deal with them one at a time when appropriate), the framework is broken up into pieces called *models*. Each model provides specifications that answer such critical questions as "What are we measuring?" or "How do we measure it?"

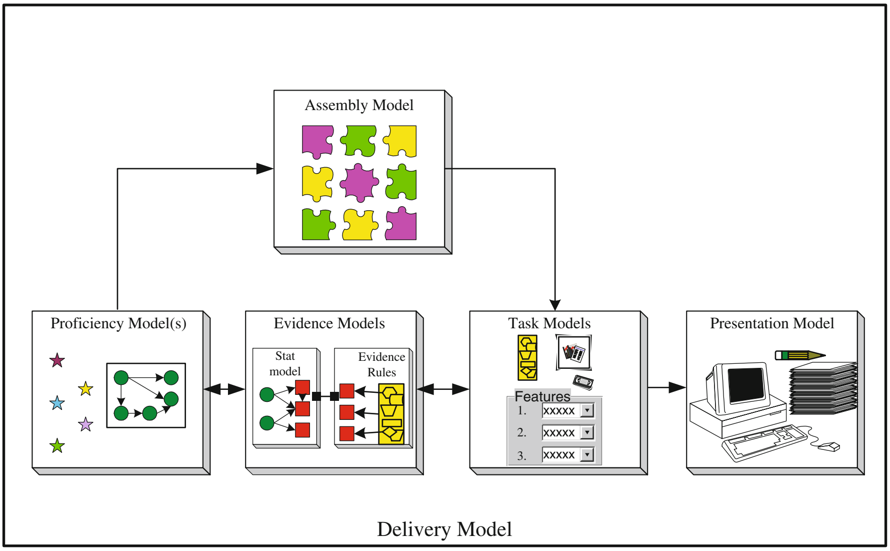


Fig. 2.1 The principle design objects of the conceptual assessment framework (CAF). These models are a bridge between the assessment argument and the operational activities of an assessment system. Looking at the assessment argument, they provide a formal framework for specifying the knowledge and skills to be measured, the conditions under which observations will be made, and the nature of the evidence that will be gathered to support the intended inference. Looking at the operational assessment, they describe the requirements for the processes in the assessment delivery system.

Reprinted from Mislevy et al. (2004) with permission from the Taylor & Francis Group.

What Are We Measuring? *The Proficiency Model*

A *proficiency model* defines one or more variables related to the knowledge, skills, and abilities we wish to measure. A simple proficiency model characterizes a student in terms of the proportion of a domain of tasks the student is likely to answer correctly. A more complicated model might characterize a student in terms of degree or nature of knowledge of several kinds, each of which may be required in different combinations in different tasks. It may address aspects of knowledge such as strategy use or propensity to solve problems with certain characteristics in certain situations. Looking ahead, the proficiency model variables will be the subset of the variables in a Bayesian net that accumulate evidence across tasks.

A closer look at the proficiency model in Fig. 2.1 reveals two kinds of elements. On the right is a graphical structure, a representation of the kinds of statistical models that are the focus of this book. On the left are a number of stars that represent claims. Claims are what users of assessments want to be able to say about examinees, and are the basis of score reports. A reporting rule maps information from probability distributions for proficiency model variables to summary statements about the evidence a student's performance provides it to support a claim.

Example 2.2 (Calculus Proficiency Model; Example 2.1 Continued). *Given that the primary purpose of the assessment is placement, only one variable is necessary in the proficiency model. This is a discrete variable whose levels correspond to the various placement options:* `Remedial Class`*,* `1st Semester Freshman`*,* `2nd Semester Freshman`*,* `1st Semester Sophomore`*,* `2nd Semester Sophomore`*,* `Junior Math Classes`*. Fig. 2.2 shows the graphical representation of this model. If there were a secondary purpose of trying to diagnose problems in low performing students, there might be a need for additional proficiency variables that would accumulate evidence about more specific skills. However, in a short test, the designers typically need to choose between good reliability for the main variables and good differential diagnosis for problems in the assessment. University C could use two tests: This placement test first, followed by a diagnostic test just for students placed into the remedial class, addressing only claims concerning precalculus skills and accumulating evidence at a grainsize that matches the instructional modules.*

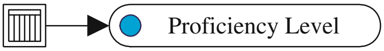


Fig. 2.2 The proficiency model for a single variable, *Proficiency Level*. The *rounded rectangle* with the *circle symbol* represents the proficiency variable, and the *square box* with the table represents its probability distribution

Reprinted with permission from ETS.

Associated with each level of the proficiency variable are one or more claims. Which claim is associated with which level depends on how the various skills are taught in the calculus series. For example, the level `2nd Semester Freshman` *would be associated with all of the claims that constitute the kinds of performances in the kinds of tasks we would want a student successfully completing that course to be able to do. If multivariate calculus is not taught* *until the sophomore year, then all of the claims associated with multivariate calculus would be associated with the levels corresponding to the sophomore year.*

Completing the proficiency model requires the specification of a prior distribution for the proficiency variable. This distribution represents our state of knowledge about a student who sits down to take the assessment before we learn their responses to any of the problems they were given. In this case, a uniform distribution does not seem appropriate as only a very few incoming freshmen will be ready for junior level coursework. However, university records for the past 5 years might provide a reasonable prior distribution. This distribution is represented by the square box in Fig. 2.2.

The key idea of the proficiency model is that it represents our state of knowledge about an examinee's state of knowledge about calculus. Chapter 13, which talks about constructing a scoring engine based on Bayesian networks, talks about making a copy of the proficiency model that tracks our state of knowledge as we gather more evidence about the examinee's proficiency. This copy is called the *scoring model*. When the examinee first sits down for the assessment, it is identical to the proficiency model. However, as we see the answer from each task the examinee attempts, the scoring model will be updated to reflect our state of knowledge about this particular examinee. The evidence models determine how that updating is done.

In succeeding chapters, we will look at proficiency models with several variables, each representing some aspect of knowledge, skill, or ability posited to influence students' performance. In each case, the idea is the same as in the simple placement test case: These variables are how we characterize students' knowledge; we do not get to observe their values directly; we express what we do know about them in terms of a probability distribution; and evidence in the form of behavior in assessment situations allow us to update our knowledge, by updating the probability distributions accordingly.

How Do We Measure it? *The Evidence Model*

Evidence models provide detailed instructions on how we should update our information about the proficiency model variables given a performance in the form of examinees' *work products* from tasks. An evidence model contains two parts, which play distinct roles in the assessment argument. They are the evidence rules for identifying information in students' performances and statistical machinery for accumulating information across tasks.

- *Evidence rules* describe how *observable variables* characterize an examinee's performance in a particular task, from the *work product*(s) that the examinee produced for that task. A work product is the capture of some aspect(s) of a student's performance. It could be as simple as a binary digit conveying a response to a true–false item, or as complex as a sequence of diagnostic test orders and medical treatments in an extended patient-management problem in a medical simulation. The observables are the primary outcomes from task performances, and they provide both information that will be used to update our beliefs about proficiency model variables and information that will be used for task-level feedback. In an operational assessment, evidence rules guide the evidence identification process. Evidence rules concern the identification and summary of evidence *within tasks*, in terms of observable variables.[1] Summary of evidence *across tasks* will be the role of the statistical part of the evidence model.

Example 2.3 (Calculus Evidence Rules; Example 2.1 Continued). *A prerequisite to specifying the evidence rules for the calculus placement test, is specifying the form of the work product. If the test is presented as multiple choice, then the work product would be the selection made by the examinee. The evidence rule would match the selection against the key to determine whether the outcome was correct or not. The observable variable would be a binary variable indicating whether the answer was correct or not. If the test is presented as a free response but the observable outcome variable was correct or incorrect, then the evidence rule would be to compare the student answer to the correct answer and code it correct if they are mathematically equivalent. If the observable outcome variable has more than two values to allow for partial credit, then the evidence rules would be the scorer's rubric used to determine partial credit. As typically there is an evidence model for each task model in an assessment, an assessment could have a mixture of different types of tasks with different evidence rules.*

Evidence rules are indifferent as to whether the scoring is done by computers or humans. What is necessary is that they provide a clear set of instructions on how to determine the value of the observables. The key-matching rule of a multiple-choice test can be done by hand but lends itself readily to computerization. A value of 0 or 1 for an observable variable based on a multiple-choice item no longer depends on how that value was calculated. Short answer questions are more difficult for computers, as they need to be able to parse and recognize equivalent mathematical expressions. Partial credit scoring can be quite difficult even for human raters. The problem of achieving agreement among multiple human raters is well studied, and we know that clearly written evidence rules and worked-through examples are a big help. Sophisticated automatic methods such as neural networks and multistage rules can be used to evaluate observable variables from rich performances in, for example, problem solving in computer simulations Williamson et al. (2006b)

[1] Note the distinction between the conceptual notion of evidence about proficiency and the "stuff" one sees in an operating assessment: The work product as a capture of something the student has done in the task situation and the observable variables as evaluated features of it. They do not constitute "evidence" absent the assessment argument, and their embodiment of elements of it.

- The *statistical part* of the evidence model provides information about the connection between *proficiency model variables* and *observable variables.* Psychometric models are often used for this purpose, including the familiar classical test theory and item response theory, and the less familiar latent trait models, cognitive diagnosis models, and Bayes nets. In an operational assessment, the statistical part of the evidence model, together with the proficiency model, structure the evidence accumulation process. The statistical part of the evidence model concerns the accumulation and synthesis of evidence *across tasks*, in terms of proficiency variables.

The proficiency model together with the statistical part of the evidence model constitute the *measurement model* of the assessment. The theory of ECD is broad enough that the measurement model does not need to be a probability model. For example, if the measurement model was a sum of scores or "number right" model, then the statistical part of the evidence model would simply state how many points to give for each answer. However, if both the proficiency model and evidence models are expressed in terms of probability distributions, then we can use Bayes theorem as the update mechanism. Chapter 3 (and most of the rest of the book) explains this in detail. In this case, the statistical part of the evidence model is specified as a conditional probability distribution providing the probability of the observable outcome variables given the examinee's state of proficiency. The familiar measurement models from IRT, the logistic function and the normal ogive function, take this form; that is, conditional probability distributions for a correct item response given the proficiency variable θ.

Example 2.4 (Calculus Evidence models; Example 2.1 Continued). *Assume that according to the evidence rules, the observable outcome for a task was a single variable isCorrect taking on values* `true` *and* `false`*. It is necessary to specify for each of the possible levels of proficiency the probability that an examinee at that level will get the item correct. This is shown as the square box in Fig. 2.3. Note that this figure shows two kinds of variables: evidence model variables (observables) labeled with a triangle, and proficiency variables (borrowed from the proficiency model) labeled with a circle.*

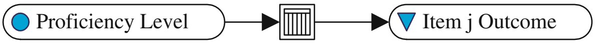


Fig. 2.3 The measurement model for a dichotomously-scored item. Variables labeled with a *triangle* are local to the evidence model, while variables labeled with a *circle* are borrowed from the proficiency model (and hence shared across evidence models for different tasks that provide evidence about them). The *square box* represents the probability distribution, which must be specified to make the model complete
Reprinted with permission from ETS.

As there are six possible levels for the proficiency variable, six different probability values must be specified for each evidence model. This is a lot. Chapter 8 discusses some ways of reducing this work. Another possibility is to learn the probabilities from data. This is called calibration; Part II discusses this in detail.

Where Do We Measure it? *The Task Model*

Task models describe how to structure the kinds of situations we need to evoke the evidence we need for the evidence models. They describe the *presentation material* that is presented to the examinee and the *work products*, which the examinee generates in response. They also contain *task model variables* that describe features of tasks as well as how those features are related to the presentation material and work products. Those features can be used by task authors to help structure their work, by psychometricians to help reduce the number of pretest subjects needed, and by test assemblers to help ensure that a particular form of the assessment is balanced across particular kinds of tasks. Mislevy, Steinberg, and Almond (2002c) explore the myriad uses of task model variables.

A task model does not represent a single task, but rather a family of potential tasks waiting to be written. Tasks are made from task models by filling in the specification made by the task model, i.e., finding or authoring presentation material and setting the values of the task model variables to the corresponding values. A typical assessment may have several task models representing different families of tasks.

Example 2.5 (Calculus Task Model; Example 2.1 Continued). *Consider the task model for a unidimensional integration task. The presentation material for this type of task consists of the integrand, the limits of the integral, the instructions given to the examinee (could be shared by several tasks) and, if the format is multiple-choice, the values of the options. Task model variables are related to this choice of material. For example, task model variables might indicate the number of factors, whether or not trigonometric functions, logarithms or exponential expressions are used, and what integration techniques must be applied to solve the integral. Note that task model variables could be set before or after the presentation material is authored: e.g., the author could note that a particular task involves two factors with trigonometric functions, or be requested to write a task that requires using integration by parts.*

The task model also must contain the expected form of the work product. If the format is multiple choice, the work product will be some form of capture of the selection that was made, such as a 0–9 digit in a data file. If the response is open-ended, the work product from a paper-and-pencil test might be the student's written production on the physical answer sheet (to be scored by a human); from a computer-based test it might be the text in a rich text file (.rtf) produced by the student's interaction with the task.

How Much Do We Need to Measure? *The Assembly Model*

Assembly models describe how the proficiency models, evidence models, and task models must work together to form the psychometric backbone of the assessment. The assembly model specifies what constitutes a valid form of the assessment. This is especially important if not all examinees get the same form (e.g., there are multiple forms for security reasons, or the test is adaptive). The rules for constructing a form are specified through *targets* and *constraints*. Targets describe how accurately each proficiency model variable must be measured (see Chap. 7), and constraints describe how tasks must be balanced to properly reflect the breadth and diversity of the domain being assessed.

Example 2.6 (Calculus Assembly Model; Example 2.1 Continued). *In constructing the assembly model for the calculus placement test, it is important that the range of tasks reflect the syllabus for the calculus sequence. This is generally achieved through constraints. It is important that the test give good information about whether or not the student is in the lower placement categories for all students. Only for a few students will we be interested in the sophomore and junior levels. This is easier to handle in an adaptive test. If the delivery mode is computer-based, then we could use the techniques described in Chap. 7 to make an adaptive test. If the delivery mode is paper-and-pencil, we could use a brief self-scored routing test or put the advanced items into a separate section at the end and instruct the students to work on this part only if they feel confident of their performance in the earlier sections.*

How Does It Look? *The Presentation Model*

Assessments today can be delivered through many different means; for example, paper and pencil, standalone computer or through the web, on a handheld device, read aloud over the phone, or as portfolios assembled by the students. A *presentation model* describes how the tasks appear in various settings, providing a style sheet for organizing the material to be presented and captured.

A common use of this idea is to support presentation of the same assessment in both paper and pencil and computer format. A more recent but increasingly important use of the presentation model is alternative presentation modes to accommodate examinees with disabilities (Shaftel et al. 2005; Russell 2011). This latter usage requires a careful examination of exactly what is being claim and what constitutes evidence, as different students may be able provide to evidence about the same capabilities despite different ways of accessing information, interacting with tasks, or producing performances—in measurement terms, exactly what constitutes construct-relevant and construct-irrelevant sources of variance (Hansen et al. 2003; Mislevy et al. 2013).

Putting It All Together: *The Delivery System Model*

The *delivery system model* describes the collection of proficiency, evidence, task, assembly, and presentation models necessary for the assessment and how they will work together. It also describes issues that cut across all of the other models, such as platform, security, and timing.

Breaking the assessment specification up into many smaller pieces enables us to reassemble pieces in different configurations for different purposes. For example, a diagnostic assessment requires a finer grain size proficiency model than a selection/placement assessment. If we want to use the same tasks in both the diagnostic and selection assessment, we can use the same task models (written generally enough to address both purposes). However, we will want different evidence models, each one appropriate to the level of detail consistent with the purpose of the assessment.

2.4.2 Four-Process Architecture for Assessment Delivery

As we have noted, assessments are delivered in a variety of platforms, from the more familiar paper-and-pencil tests, oral exams, and more recent computer-based tests, to the newer ways of delivering tests through the Web, over the phone, and with handheld devices such as minitablet computers and smartphones.

To assist in planning for all these diverse ways of delivering a test, ECD provides a generic framework for test delivery: the four -process delivery architecture (Almond et al. 2002a; Almond et al. 2002b). The four-process delivery architecture shown in Fig. 2.4 is an ideal system; any realized assessment system must contain these four processes in some form or other. They are essential to making the observations and drawing the inferences that comprise an assessment argument. This is true whether some of the processes are collapsed or degenerate in a given system, and regardless of whether they are carried out by humans, computers, or human–computer interactions. The IMS Consortium adopted this idealization as a reference model for use with their standards on question and test interoperability (IMS 2000), although they used different names for different pieces.

How Is the Interaction with the Examinee Handled? *The Presentation Process*

The *presentation process* is responsible for presenting the task and all supporting presentation material, managing interaction with the student, and gathering the work products. Examples include a display engine for computer-based testing, a simulator which can capture an activity trace, a protocol for a structured interview and the human administering it, and a system for distributing test booklets and capturing and scanning the answer sheets. In a paper-and-pencil assessment, the presentation process concerns administering preassembled test booklets to examinees and collecting and possibly scanning

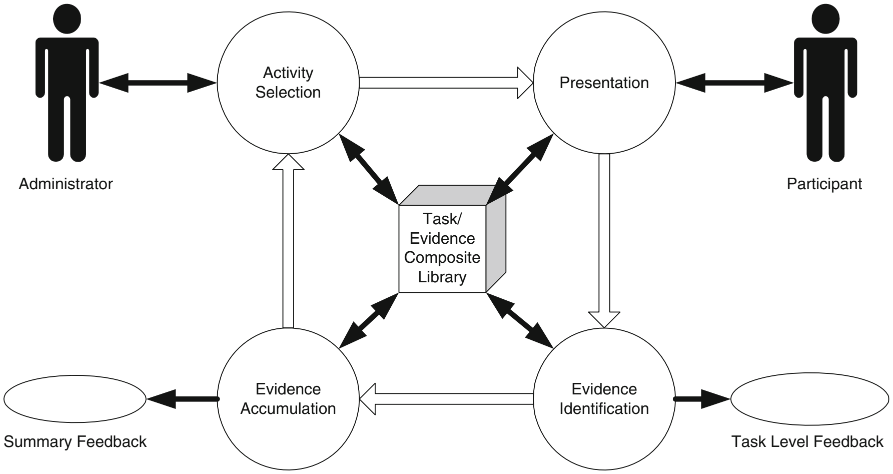


Fig. 2.4 The four principle processes in the assessment cycle. The activity selection process selects a task ("tasks" could include items, sets of items, simulation problems, or learning activities, as examples) and directs the presentation process to display it. When the participant has finished interacting with the task, the presentation process sends the results (one or more work products) to the evidence identification process. This process identifies essential observations about the results and passes them to the evidence accumulation process, which updates the scoring record, tracking our beliefs about the participant's knowledge. All four processes add information to the Results Database. The activity selection process then makes a decision about what to do next, based on the current beliefs about the participant or other criteria

Reprinted from Mislevy et al. (2004) with permission from the Taylor & Francis Group.

the answer sheets. In a computerized adaptive assessment, presentation concerns presenting a sequence of tasks to an examinee one at a time (as directed by the activity selection process), in each instance capturing a response. The next processes will evaluate it on the spot and use the information to guide the selection of the next task.

How Is Evidence Extracted from a Task Performance? ***The Evidence Identification Process***

The *evidence identification process* (called *response processing* in the IMS specification) is responsible for identifying the key features of the work product that are the observable outcomes for one particular task. The observable outcomes can go back to the participant for task-level feedback, be passed on to the evidence accumulation process, or both. Examples include matching a selected response to an answer key, running an essay through an engine, and having a human rater score a student portfolio according to a rubric. The evidence rules from the CAF specify how this is to be accomplished. Evidence identification can consist of multiple stages, as when lexical and syntactic features are identified in an essay and a regression model is used to summarize them into a single score for a response to this task.

A question sometimes arises as to whether a particular operation is part of the presentation process or the evidence identification process. Often the answer lies with how the system could possibly be reused or reconfigured, and steps in a continuous process are parsed out in CAF models accordingly.

Consider a multiple-choice item presented on a computer for which the evidence model calls for a binary observable *isCorrect*. The presentation process must present the stem, key, and distractors to the examinee, and provide some mechanism for making a selection. The examinees make some kind of gesture (clicking the mouse, pressing a key) to indicate their selections. The assessment delivery system must translate that gesture first into a code indicating which option was selected, then match that against the key.

In this setup, the ideal division of labor is achieved when the work product consists of a specified representation of the selection made by the examinee. Using the raw mouse click as the work product is too detailed. It requires the evidence identification process to know details of the formatting of the item on the screen in order to interpret the raw data. We would like the freedom to use an alternative presentation process that uses key presses to indicate the selections without having to also change the rules of evidence. On the other hand, having the presentation process match the selection to the key goes too far in interpreting the raw response. We want the freedom to substitute an alternative evidence identification process that uses which distractor was selected to help diagnose misconceptions without needing to also change the presentation process.

How Is Evidence Accumulated Across Tasks? *The Evidence Accumulation Process*

The *evidence accumulation process* (called *summary scoring* in the IMS specification) process is responsible for synthesizing the information from observable outcomes across multiple tasks to produce section and assessment level scores. Examples include the IRT engine used in GRE computerized adaptive testing (CAT) testing, the Bayesian network evidence accumulation process at the heart of this book, and simply counting up the number of right answers. The measurement model in the CAF associated with a particular task specifies how this is to be accomplished.

What Happens Next? *The Activity Selection Process*

The *activity selection process* is responsible for deciding what the next task should be and when to stop the assessment. When making these decisions, adaptive assessments consult the current state of what is known about a student, in terms of the values of the proficiency-model variables as they have been updated thus far by the evidence accumulation process (Chap. 7 talks about some possible measures). An instructional system will also make decisions about switching between assessment and instruction (Shute 2003April). Examples of activity selection processes include simple linear sequencing (for example, paper-and-pencil tests, although the student may chose the order in which to answer items within each section as it is administered), and computerized adaptive item selection (e.g., the GRE CAT), and student choice as to when to move on in a self-paced practice system.

Where Do Processes Get the Information They Need? *The Task/Evidence Composite Library*

All four processes require certain kinds of data in order to do their jobs: The presentation process requires the text, pictures, and other material to be displayed. The evidence identification process requires the "key," the parameters for algorithms, or other evidence rule data with which to evaluate the work products. The evidence accumulation process requires the parameters that provide the "weights of evidence" for each task, such as scoring weights, item response theory parameters, or the conditional probabilities in Bayes nets discussed in this book. The activity selection process requires classification and information codes to balance the assessment form. The Task/Evidence Composite Library is a unified database that stores this information.

We have suggested, without detailing, the mapping between the Design models in the conceptual assessment framework and the four processes. All of the design decisions made in the blueprint are reflected either directly in the implementation or in one of the processes leading up to the implementation. Again, further discussion and examples are available in Almond et al. (2002a); Almond et al. (2002b).

Example 2.7 (Calculus Test Delivery System; Example 2.1 Continued). *Consider first a paper-and-pencil version of the calculus placement test. The activity selection process consists of rules for assembling and distributing the paper-and-pencil forms. One possible mechanism is to group the tasks into sections of increasing difficulty and instruct the examinee to not attempt the next section unless they are fairly confident of their answers to the previous section. The presentation process consists of the mechanism for distributing the forms and collecting and scanning the answer sheets. In this case, the work product for each task is a free response that has been scanned and stored as bitmap image. The evidence identification process consists of scoring system in which the work products are distributed to raters who mark them as correct or incorrect, and record the scored outcomes. The evidence accumulation process is a Bayesian network that incorporates the information from the observable variables (i.e., "absorbs the evidence") and calculates the probability that the examinee is in each of the six possible groups.*

Now, consider an alternative computer delivered version of the assessment using multiple-choice versions of the tasks. In this case, the activity selection *process is a version of the critiquing algorithm of Chap. 7 that tries to establish whether the examinee is at or above each of the possible levels in sequence (ready for* `1st semester`*,* `2nd semester`*, and so on). The presentation process displays the task and records the selection made by the examinee. The evidence identification process matches the selection to the key and sets the value of the observable outcome variable to correct or incorrect as appropriate. The evidence accumulation process is again Bayesian network engine which absorbs the evidence from the evidence identification process and calculates the probability that the examinee is in each group based on the evidence so far. Note that the activity selection process can query the Bayes net when making the decision about which task to select or whether to stop or move on to target the next level of proficiency.*

2.4.3 Pretesting and Calibration

In order to score an assessment, the evidence identification process or the evidence accumulation process (or both) may need to build in empirical information from previous administrations of the tasks. In the case of evidence identification, this information is incorporated into evidence rules. For example, an automated essay-scoring system can be "trained" to match human ratings given values of lower-level lexical and syntactic features of a particular essay (Deane 2006). (Calibration of evidence-identification processes is not discussed in this book, but see Williamson et al. (2006b).) In the case of evidence accumulation, it appears in scoring weights or task-specific evidence model parameters. We refer to a start-up set of data from which to estimate these values as *pretest data*, and the operation of determining the values as *calibration*. An evidence model tuned to work with a specific task is called a *link model* (Chap. 13).

If the measurement model for the assessment is a probability model, we can use Bayes theorem to "learn" the task-specific parameters in the link model. The original evidence model provides a prior distribution for the parameters, based on the experts' understanding of the domain and our previous experience with similar task models. We can then use Bayes theorem to refine those priors with pretest data. Part II describes two methods for doing this with Bayesian network models.

Using probability-based models has another advantage as well. The model makes a prediction for what the pretest data will look like before they are gathered. If the pretest data look "surprising" compared to the model, then this suggests we might want to refine our model. This principle extends to ways to compare models and to search for a best model. Chapter 10 looks at some techniques for model criticism and model search. This is particularly important when the measurement model has been built to reflect our understanding of the underlying cognitive processes. In this case, critiquing the model will help us refine our knowledge of the cognitive processes we are trying to measure.

2.5 Conclusion

Developments in statistical methodologies and new kinds of psychometric measurement models hold the promise of supporting a wider variety of educational assessments than have been traditionally used. To capitalize on their potential, however, one cannot think of using them in isolation from the other components of assessment design. All must work in concert to create an assessment that is at once coherent and practicable.

Toward this end, it will be of significant benefit to have a shared framework for talking about the roles that each facet of the design elements and delivery processes play in the support of a coherent assessment argument. Evidence-centered design provides such a framework, and can thus prove useful for understanding how graphical modeling techniques fit into assessment systems.

Exercises

2.1. The basic models of the ECD "conceptual assessment framework" are the proficiency model, the evidence model, and the task model. In which of these models are variables that concern *characteristics of the situations* in which students say, do, or make things? In which are variables that concern *characteristics of the students*? In which are variables that concern *characteristics of the particular things students say, do, or make*?

2.2. What are the two submodels of the evidence model? How do their roles differ from one another?

2.3. How are the proficiency model and the evidence model related to each other, in terms of both shared or overlapping information and connections in the assessment argument?

2.4. How are the evidence model and the task model related to each other, in terms of both shared or overlapping information and connections in the assessment argument?

2.5. How are the proficiency model and the task model related to each other, in terms of both shared or overlapping information and connections in the assessment argument?

2.6. How are the assembly model and the task model related to each other, in terms of both shared or overlapping information and connections in the assessment argument?

2.7. How are the assembly model and the proficiency model related to each other, in terms of both shared or overlapping information and connections in the assessment argument?

2.8. An important part of the process of designing any assessment is selecting items (or tasks) for a form. Which ECD model specifies what constitutes a valid form?

2.9. A common task teachers use to assess a student is the *book report*, where the student reads a book and then writes (or presents) a report based on the contents. Consider the Book Report as a task model. What are the presentation material and work products? List some possible task model variables.

2.10. The Book Report is used across a large number of grades. How do the values of the task model variables change when the Book Report task is used in a 6th grade classroom as opposed to a 4th grade classroom? Which ECD model(s) must be changed to ensure that a Book Report task is grade appropriate?

2.11. If the form of the work product in the Book Report task model is changed from a written report to an oral presentation, how much do the other models change? (Changing the expected work product form variable changes the *evidentiary focus* of the task. Often when changing a task model variable changes the evidentiary focus of a task, it is helpful to split the task model into two task models, in this case Oral Book Report and Written Book Report task models.)

2.12. Decision analysts often use what is called the *clarity test* when defining variables. A variable passes the clarity test if a person who had access to all of the available data (referred to as a “clairvoyant” in this literature) could unambiguously assign a value to the variable. A variable that does not meet the clarity test must be refined in order to begin statistical modeling. For example, “the SAT score of a candidate” does not meet the clarity test, but “the most recent SAT score of a candidate” and “the highest SAT score of a candidate” are both refinements that do meet the clarity test.

For each of the following variables, state whether or not they meet the clarity test. If the do not, suggest how the definition might be revised to meet the clarity test.

a. The Gender of a participant.
b. The Race of a participant.
c. The Socioeconomic Status of a participant.
d. Whether or not a participant receives a free or reduced price lunch.
e. Whether a participant lives in a high crime area.
f. Whether a participant lives in an urban, suburban or rural location.

2.13. What elements of ECD are used to ensure that proficiency variables pass the clarity test? For observable variables?

3

Bayesian Probability and Statistics: a Review

Our ultimate goal is to draw inferences about the state of a student's proficiency, which is unknown, from observations, which are known. We will press probability theory into the service of modeling our uncertain knowledge about the student's proficiencies. Probability theory is one of the most studied models for uncertainty and the choice of probability theory brings with it all of the tools of Bayesian statistics.

Although we assume the reader has had at least one course in probability and statistics at the college level, typically such courses only dwell briefly on the Bayesian formulation of statistics (and that unit is often omitted for time reasons). This chapter, therefore, reviews some of the key differences between Bayesian and Classical statistics. Section 3.1 discusses the basic definition of probability and its use in representing states of information. Section 3.2 reviews conditional probability and Bayes' theorem, tools we will use again and again. Section 3.3 looks at the concepts of independence and conditional independence, which will form the basic building blocks of our models; Chap. 4 on graphical representation will build heavily on this section. Section 3.4 provides a quick review of random variables and Sect. 3.5 looks at how Bayes' theorem can become a paradigm for learning about unknown quantities.

A short chapter such as this one cannot cover all of the probability theory and statistics necessary to understand in detail all of the models explored in this book. We hope that this chapter will provide enough background in Bayesian ideas of probability so that an educational researcher can at least follow the arguments at a high level. Readers wishing to follow the more mathematical parts in greater detail will need to be familiar with Bayesian statistics at the level of Gelman et al. (2013a).

3.1 Probability: Objective and Subjective

Although all statisticians agree that a probability is a number between 0 and 1, there are two main schools for interpreting that probability. Perhaps the best

R. G. Almond et al., *Bayesian Networks in Educational Assessment*,
Statistics for Social and Behavioral Sciences, DOI 10.1007/978-1-4939-2125-6_3

known uses the *Relative Frequency* definition (Sect. 3.1.1), based on objective properties of repeatable experiments. However, Bayesian statistics owes a large debt to the *Subjective* school of probability (Sect. 3.1.2). For educational testing, it is necessary to synthesize the two views into an *Objective–Subjective* approach (Sect. 3.1.3).

3.1.1 Objective Notions of Probability

A random experiment is one whose outcome for a single trial is unknown before the experiment begins, but for which we can make reliable predictions about the collection of outcomes that comes from repeating the experiment many times. To make this more definite, consider the following experiment.

Example 3.1 (Balls in an Urn). *Consider an urn*[1] *that contains b black balls and w white balls all of the same size and weight, thoroughly mixed together (Fig. 3.1). Somebody reaches into the urn and draws out a ball without looking. If the experiment is repeated many times, replacing and mixing the balls each time, the proportion of black balls drawn will be $\frac{b}{b+w}$. We say the* probability *that a drawn ball is black is $\theta = \frac{b}{b+w}$.*

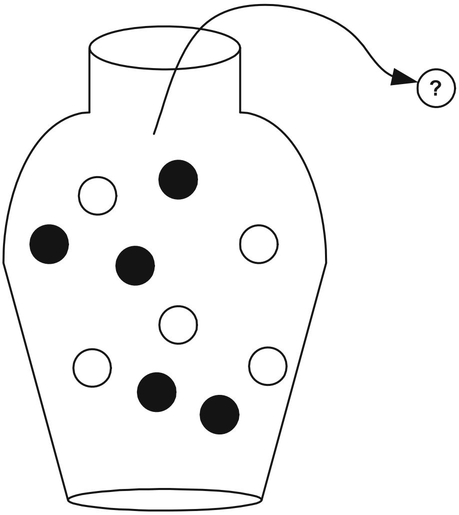


Fig. 3.1 Canonical experiment: balls in an urn
Reprinted with permission from ETS.

[1] The use of an urn instead of another type of container that might be easier to draw from is a hallowed tradition in statistics.

Example 3.1 is a canonical example of a *Bernoulli experiment.* While we cannot say much about the outcome of a single experiment, if we have a series of independent Bernoulli trials, we can say quite a lot about the behavior of the series. Suppose we have n independent draws from our urn, and let Y be the number of black balls we obtain in those draws. Combinatorial arguments (see Ross 1988) show that there are $\binom{n}{y} = \frac{n!}{y!\,(n-y)!}$ distinct sequences of n balls with y blacks and $n-y$ whites. Therefore, we have,

$$p(Y = y|\theta, n) = \begin{cases} \binom{n}{y}\theta^y(1-\theta)^{n-y} & \text{for } y = 0, \ldots, n \\ 0 & \text{otherwise.} \end{cases} \tag{3.1}$$

Equation 3.1 is the well-known *binomial distribution* with parameters θ and n. The mean of the binomial—the expected number of events in n trials—is $n\theta$ and the variance is $n\theta(1-\theta)$. As test items are often scored dichotomously—1 for right, 0 for wrong—the binomial distribution (and its close cousin the multinomial distribution) will play a large role in the sequel.

Y/n is the proportion of black balls in a given sample of size n. The mean of Y/n in repeated experiments is θ, and the variance is $\frac{\theta(1-\theta)}{n}$. Note that as n gets larger and larger, the variance of Y/n gets smaller and smaller. In other words, the distribution of the proportion of black balls obtained in very large samples is clustered closely around the true urn proportion θ. Thus, as n goes to infinity Y/n goes to θ. This is known as the *Law of Large Numbers.* In the frequency school of probability (the one most often taught in elementary statistics courses) this limit is taken as the definition of probability.

Definition. **Probability (Frequency Definition).** *Let Ω be the set of outcomes from an experiment that can be repeated many times (such as black ball and white ball). Let $A \subset \Omega$; that is, A is a subset of the outcomes (such as a black ball). Then the* probability of A, *$P(A)$, is the limiting proportion of the times for which the outcome lies in A in an arbitrarily long sequence of trials.*

Some important properties of probability come out of this definition. First, probability is a *measure* in the same sense that we can measure the length of a line segment in geometry: the probability of two disjoint (nonoverlapping) sets is the sum of the probabilities. Second, the smallest possible probability is 0 and the largest is 1. It would be nice to say that probability 1 means definitely certain and probability 0 means definitely will not occur. This is indeed the case when Ω consists of a finite set of possible outcomes. However, when the set of possible outcomes is continuous, such as all the points on the real line, there are certain pathological cases (such as when A consists of a single point) that can have probability 0. Therefore, probability 1 means "practically certain" and probability 0 means "does not occur except in pathological cases."

3.1.2 Subjective Notions of Probability

The definition of probability given above presents two problems when applied to educational measurement. First, many of the experiments we wish to describe are not repeatable. If we present the same item to a student twice we expect some learning will have taken place, which renders the two trials no longer independent. Second, many of the quantities we wish to make probabilistic statements about are not observable at all. For example, we cannot directly observe a student's calculus proficiency; we can only indirectly observe the outcomes when she is asked to solve calculus problems.

Bayesian statistics often talks about probability distributions over unknown parameters. It is customary in Bayesian statistics to take as the fundamental definition of probability not the relative frequency but a degree of belief in unknown events.

Definition. Probability (Subjective Definition). *Let Ω be the set of outcomes from an experiment which may or may not be possible to carry out. Let $A \subset \Omega$. Then* Your probability of *A, $P(A)$, is Your belief that the outcome will be in A; numerically it is equal to Your belief that a black ball will be drawn from an urn with a proportion $P(A)$ of black balls.*

This phrasing defines probability in terms of analogies to simple, repeatable experiments. Furthermore, it emphasizes probability as a degree of belief of some person. People with different information or different models for the underlying situation can therefore have different probabilities for the same event.

The definition of subjective probability can be derived from a set of axioms describing properties of rational belief of an ideal decision maker[2](Savage 1972; de Finetti 1990). These axioms provide the standard properties of probability as a measure between 0 and 1. They also can be used to show that in the case of a repeatable experiment, a reasonable person's subjective probability will converge to the relative frequency.

The use of this subjective definition of probability has been one of the reasons for the slow adoption of Bayesian statistical ideas (the second has been computational difficulties). It has not helped that the axioms are often stated in terms of "fair bets"; people's attitudes both in favor and against games of chance has hindered the appeal of the derivation. However, the ability to make statements of probability about experiments that are only theoretical opens up a great number of applications. In particular, using Bayesian statistics we can make direct statements of probability about unknown parameters rather than indirect statements about the property of estimators (take a look at the definition of a confidence interval in most any statistics textbook).

When using the subjective definition of probability, assessing probability distributions that model real-world phenomena can be a challenge. The number of techniques for this particular task is large, and many of them are somewhat controversial. Morgan and Henrion (1990), Walley (1991), and Berger (1985) provide reviews. One problem in this field is that the lay perception of probability is not consistent and is subject to heuristic biases, such as

[2] In this literature the ideal decision maker is often called "You."

ignoring base rates and mistaking representativeness for likeliness (Kahneman et al. 1982). The fact that many subject matter experts are not also statistical experts presents a major challenge when basing probabilities on expert opinion.

One method that is universally agreed upon is the principle of an equal probability space.

Definition. Principle of Equal Probability Space. *Let Θ be a sample space with N elements that are judged to be equally likely. Let A be an event in Θ. Then $P(A) = \frac{\#(A)}{N}$, where $\#(A)$ is the number of elements in A.*

Example 3.2 (Playing Cards). *Consider an ordinary deck of 52 playing cards with ace, two, three, …, ten, jack, queen, and king in each of the four suits spades, hearts, diamonds, and clubs. The probability of drawing a spade on a single draw is* $13/52 = 1/4$. *The probability of drawing an ace is* $4/52 = 1/13$. *The probability of drawing the ace of spades is* $1/52$.

This principle works well for simple games of chance like cards and dice. However, in many experiments the elements of the underlying space are not equally likely. Consider the experiment of giving a calculus exam to early elementary students. It is quite unreasonable to assume that all scores are equally likely! This same problem shows up in subtler ways when trying to form a noninformative distribution for a binomial proportion (Sect. 3.5.5). Notwithstanding these difficulties, this principle often provides a reasonable starting place.

3.1.3 Subjective–Objective Probability

In educational testing, we want to be able to make objective statements about learners. On the other hand, the subjectivist approach offers possibilities for making interesting statements about unobservable proficiencies. Clearly we need some kind of synthesis.

Good (1976) points out that even seemingly objective models have a subjective component. For example, we may make a subjective judgment that scores from a given test follow a normal distribution. Then we gather data and create an "objective" estimate of the population mean. The "objective" estimate of this model parameter is conditioned on the subjective choice of the model. Good (1983) points out that the sensitivity of inferences to modeling choices, such as the assumption of normality, is often much larger than the sensitivity to obviously subjective prior opinion about the parameters in the models.

Good's philosophical approach essentially states that all models are subjective. They become *objective* when many people agree on the model. In many cases, the model may not be known precisely, or different peoples' models may differ in minor ways. In such cases a sensitivity analysis can reveal whether critical decisions are dependent on these differences between candidate models.

Dempster (1990) mixes the subjectivist and objectivist ideas in a different fashion. He states that all probability judgments are subjective in the sense that they are relative to a given body of evidence. However, he allows only probabilities that are objective in the sense that they come from a readily identifiable and objective data source.

This book takes an approach somewhere between Good's and Dempster's blend of objectivism and subjectivism. Objective models come from consensus between a group of decision makers on a relatively identifiable body of information. The key here is the identification and publication of the sources of information. It is critical that reviewers be able to examine and critique all of the information sources. We therefore define probability as representing a state of information about an unknown event.

Definition. Probability (Objective–Subjective Definition). *Let Ω be the set of outcomes from an experiment which may or may not be possible to carry out. Let $A \subset \Omega$. Suppose that according to our best information, the outcome A is analogous to drawing a black ball from an urn with a proportion $P(A)$ of black balls. Then* our probability of *A is $P(A)$.*

This definition differs from the Subjective definition of probability by positing an agreement among people about the information and the models that will be used to ground probabilistic inferences. Inferences in educational testing can be consequential decisions such as employment and instruction. In this context, fairness is an important value that goes beyond the statistical property of objectivity. Candidates should know the criteria on which they are being judged. If expert opinion is used, candidates and test users must be able to learn who the experts are, what their qualifications are, how they were selected, and what rationale the criteria are based on. In terms of the evidence-centered design (ECD) models, this includes in particular the evidence rules. Transparency ensures that aspects of the model are open to challenge and eventually improvement.

3.2 Conditional Probability

The key to using subjective probability objectively is to identify the body of information on which our probability judgments are based. It follows that if that body of information changes, then so may our probabilities. The concept of conditional probability formalizes this idea.

Definition. Conditional Probability. *Let A and B be any two events such that* $\mathrm{P}(B) \neq 0$*. Define the* conditional probability of *A* given *B (written* $\mathrm{P}(A|B)$*) as follows:*

$$\mathrm{P}(A|B) = \frac{\mathrm{P}(A \cap B)}{\mathrm{P}(B)} \ . \tag{3.2}$$

An important use of conditional probability in the physical and social sciences is describing the probability of hypothetical situations. In particular, the conditional probability of A given B, $\mathrm{P}(A|B)$, answers the question, “If we knew B, what would we believe about A?” In the medical context, we might let A represent a particular disease state and B represent a particular symptom. $\mathrm{P}(B|A)$ says how likely we think the symptom would be if a person actually had the disease. $\mathrm{P}(A|B)$ says how likely it would be that a person has the disease if we were to observe the symptom.

In many respects, all probabilities are conditioned on our experience. Some authors even write $\mathrm{P}(A|e)$, where e represents the body of our experience so far.

Example 3.3 (Remedial Reading). *Let R represent the proposition that a student will benefit from a remedial reading program. Let e represent a teacher's experience with that student's reading ability. Then $\mathrm{P}(R|e)$ would represent the teacher's probability that the student would benefit from the program. If T represents the proposition that the student does well on a reading test, then $\mathrm{P}(R|T, e)$ represents the teacher's probability that the student would benefit from the program given both the test score and the experience.*

If we wish to be objective about our definition of probability, we must be explicit about what we can and cannot include in e. From a purely scientific perspective, the more information we have the better decision we can make. However, there may be social reasons why we wish to restrict what information we have. Suppose the experience of the teacher is that students from certain racial groups are more likely to benefit from the reading program. Should the teacher include race in e? Primary language spoken at the student's home? These are not easy questions to answer, and they involve complicated personal and societal values about fairness.[3] The answer may be different if the decision is admission to college rather than placement into a reading program.

The *joint probability* of $A \cap B$ can be recovered from the conditional probability of A given B and the *marginal* probability of B. In this context, *marginal* should be read as *unconditional*. We will see that it helps to think of this probability as the margin of a table. The *multiplication rule* does this recovery.

Definition. Multiplication Rule. *Let A and B be two events such that $\mathrm{P}(B) > 0$. Then*

$$\mathrm{P}(A \cap B) = \mathrm{P}(A|B)\mathrm{P}(B) \ . \tag{3.3}$$

[3] Let R represent the proposition that Jim robbed the convenience store, and F be the fact that to the question “Did you rob the convenience store?” Jim responded “I refuse to answer that question on the grounds that my answer may tend to incriminate me.” Even though F empirically supports R—that is, $\mathrm{P}(R|F) > \mathrm{P}(R)$—an American judge instructs the jury to ignore it in their deliberations. The Fifth Amendment provides citizens this right in order to reduce the government's incentive to elicit confessions unscrupulously.

By applying this idea repeatedly, it is always possible to write a joint probability as the product of conditional probabilities, with each event in the list conditional on those earlier in the list. That is,

$$\begin{aligned} &\mathrm{P}\left(A_n, A_{n-1}, \ldots, A_2, A_1\right) \\ &= \mathrm{P}\left(A_n|A_{n-1}, ..., A_2, A_1\right) \times \mathrm{P}\left(A_{n-1}|A_{n-2}, \ldots, A_2, A_1\right) \times \cdots \times \\ &\quad \mathrm{P}\left(A_2|A_1\right) \times \mathrm{P}\left(A_1\right) \\ &= \prod_k \mathrm{P}\left(A_k|A_{k-1}, \ldots, A_1\right), \end{aligned}$$

where the final term is understood to be simply $\mathrm{P}(A_1)$. This is called a *recursive representation.* Such a representation holds for any ordering, but we will see in Sect. 3.3.1 that some orderings are more useful than others.

Conditional probability provides ways of calculating probabilities that would be otherwise difficult to judge. Two principal tools for doing this are the *Law of Total Probability* and *Bayes' Theorem.*

Definition. Law of Total Probability. *Let $A_1, \ldots, A_n$ be a partition of an outcome space Ω and let B be another event in Ω. Then*

$$\mathrm{P}(B) = \sum_{i=1}^{n} \mathrm{P}(B|A_i)\mathrm{P}(A_i) \ . \tag{3.4}$$

This relationship is valuable because the conditional probabilities are often easier to assess than the unconditional probability. The following theme comes up over and over again in educational testing.

Example 3.4 (A Discrete Item Response Model). *Suppose that the students in a classroom can be divided in thirds on their mastery of a new skill, S. One-third have completely mastered the skill, $S =$ `high`; one-third have partially mastered the skill, $S =$ `medium`, and one-third have not mastered the skill at all, $S =$ `low`. Let X represent the event that a student (chosen at random from the class) is able to solve a particular problem that uses the skill in question. Further, we say that there is a 90 % chance a master can solve the problem, a 50 % chance a partial master can solve it, and a 20 % chance a nonmaster will stumble upon a solution. Then the expected proportion of correct responses in the classroom as a whole is the weighted average of the expected proportion for each mastery state, with the weights being the proportion of students at each state:*

$$\begin{aligned} \mathrm{P}(X) &= \mathrm{P}(X|S = \texttt{high})\mathrm{P}(S = \texttt{high}) + \mathrm{P}(X|S = \texttt{medium})\mathrm{P}(S = \texttt{medium}) \\ &\quad + \mathrm{P}(X|S = \texttt{low})\mathrm{P}(S = \texttt{low}) \\ &= 0.9 \cdot 1/3 + 0.5 \cdot 1/3 + .2 \cdot 1/3 \approx 0.533. \end{aligned}$$

Suppose that in another classroom, the conditional probabilities of a correct solution are the same but the distribution of students in mastery states is *different: 3/6 at* $S = \texttt{high}$*, 2/6 at* $S = \texttt{medium}$*, and 1/6 at* $S = \texttt{low}$*. The expected percentage in this classroom is obtained by the same formula with the same conditional probabilities, but the mastery distribution for the second classroom:*

$$\begin{aligned}\mathrm{P}(X) &= \mathrm{P}(X|S=\texttt{high})\mathrm{P}(S=\texttt{high}) + \mathrm{P}(X|S=\texttt{medium})\mathrm{P}(S=\texttt{medium}) \\ &\quad + \mathrm{P}(X|S=\texttt{low})\mathrm{P}(S=\texttt{low}) \\ &= 0.9\cdot 3/6 + 0.5\cdot 2/6 + .2\cdot 1/6 = 0.650.\end{aligned}$$

This example demonstrates an assumption that conditional probabilities for item responses given latent proficiencies are stable, but distributions of latent proficiencies can vary across groups or before and after instruction. It is a hallmark of psychometric models such as latent class analysis and item response theory.

We have just seen how we can build up the population proportion of correct response from the conditional probabilities of students at the different levels of mastery. The reasoning is in the opposite direction of the problem we face in educational testing, however. It would be useful to reverse the direction of the conditioning, that is, to calculate the probability of the skill state given the result from solving the problem. Bayes' Theorem allows us to do just that.

Theorem 3.1 (Bayes' Theorem). *Let* $A_1,\ldots,A_n$ *be a partition and* B *be an event such that* $\mathrm{P}(B)>0$ *and* $\mathrm{P}(A_i)>0$ *for all* i*. Then:*

$$\mathrm{P}(A_i|B) = \frac{\mathrm{P}(B|A_i)\mathrm{P}(A_i)}{\sum_{i=1}^{n}\mathrm{P}(B|A_i)\mathrm{P}(A_i)} = \frac{\mathrm{P}(B|A_i)\mathrm{P}(A_i)}{\mathrm{P}(B)}\ . \tag{3.5}$$

Example 3.5 (Diagnostic Testing; Example 3.4 Continued). *Suppose that one of the students from the first classroom in Example 3.4 is able to solve the problem. What is the probability that that student has completely mastered the skill; that is,* $\mathrm{P}(S=\texttt{high}|X)$*?*

$$\begin{aligned}\mathrm{P}(S=\texttt{high}|X) &= \frac{\mathrm{P}(X|S=\texttt{high})\mathrm{P}(S=\texttt{high})}{\mathrm{P}(X)} \\ &\approx \frac{0.9\cdot 1/3}{0.533} \approx 0.5629\end{aligned}$$

This application of Bayes' Theorem is so useful that its parts are given special names. The unconditional probability, $\mathrm{P}(S=\texttt{high})$, is called the *prior* and the conditional probabilities $\mathrm{P}(X|S=s_i)$ for $s_i = \{\texttt{high,medium,low}\}$ are called the *likelihood*. The final (conditional) probability for the quantity of interest, $\mathrm{P}(S=\texttt{high}|X)$ is called the *posterior* because it represents our information about S after observing X. In general, both the likelihood and the prior have an influence on the posterior. This next example shows this dramatically.

Example 3.6 (HIV Test; Almond (1995)). *A common test for the HIV-1 virus (believed to be a principal cause of AIDS) is the Western Blot Test. In 1988, the* Morbidity and Mortality Weekly Report *reported the analytic sensitivity and specificity of the Western Blot test as reported by the Center for Disease control in a 1988 evaluation. The analytic* sensitivity *is the conditional probability of obtaining a positive test result from a positive sample; it was 99.3 %. The analytic* specificity *is the conditional probability of obtaining a negative result from a negative sample; it was 97.8 %. As a rough guess, about 5 persons per 10,000 had HIV in the state of Washington in 1991. (Note: These figures were obtained by multiplying the AIDS prevalence rate reported in the November 8, 1991* Seattle Times *by 5. This fudge factor should probably be increased for urban areas or other high risk populations. For a discussion of more accurate methods for estimating HIV infection, see Bacchetti et al. 1993.)*

Define the following events:

HIV_+*—subject has HIV*
HIV_-*—subject does not have HIV*
T_+*—subject tests positive*
T_-*—subject tests negative*

The Western Blot test's performance can be summarized by the following two conditional probabilities: $\mathrm{P}(T_-|HIV_-) = 0.978$ *(specificity) and* $\mathrm{P}(T_+|HIV_+) = 0.993$ *(sensitivity). In both cases higher values are preferred, because specificity is the probability of a negative test result when the disease is not actually present and sensitivity is the probability of a positive result when it is.*

If the hospital blood bank uses this test to screen blood donations, it wants to know the probability that a randomly chosen sample of blood will have HIV given that it tests negative with the Western Blot test.

$$\begin{aligned}\mathrm{P}(HIV_+|T_-) &= \frac{\mathrm{P}(T_-|HIV_+)\mathrm{P}(HIV_+)}{\mathrm{P}(T_-|HIV_+)\mathrm{P}(HIV_+) + \mathrm{P}(T_-|HIV_-)\mathrm{P}(HIV_-)}\\ &= \frac{.007 \times .0005}{.007 \times .0005 + .978 \times .9995} \approx 4 \times 10^{-6}\end{aligned}$$

If a doctor administers the test to patients to diagnose them for AIDS, she wants to know the probability that a randomly chosen patient has HIV given that he tests positive with the Western Blot test.

$$\begin{aligned}\mathrm{P}(HIV_+|T_+) &= \frac{\mathrm{P}(T_+|HIV_+)\mathrm{P}(HIV_+)}{\mathrm{P}(T_+|HIV_+)\mathrm{P}(HIV_+) + \mathrm{P}(T_+|HIV_-)\mathrm{P}(HIV_-)}\\ &= \frac{.993 \times .0005}{.993 \times .0005 + .022 \times .9995} \approx .022\end{aligned}$$

Or about 1 in 50! How could there be such a low probability of a correct result when the test seems so reliable in terms of its sensitivity and specificity? The picture starts to become clearer when we realize that the chance of test failure, even though it does not happen often, is still much larger than that of the disease (at least for low-risk populations).

Many people find this example, often called the *Rare Disease Problem*, counterintuitive. Kahneman et al. (1982) talk about people's heuristic biases in evaluating probabilities. The result is puzzling if one ignores or discounts the effect of the low background rate of occurrence of the phenomenon, and puts too much weight on the test results. A false reading from the Western Blot test is a rare occurrence; but so is having HIV (unless the patient belongs to a high-risk population). In this case, a false positive is less rare than the disease itself. That is why doctors do not recommend HIV tests unless the patient is believed to be at risk a priori (before the test). Other information that increases the background probability of HIV would reduce the probability that a positive reading is false. Most doctors would not regard a positive result on the Western Blot test as a definitive positive diagnosis; they would follow it up with more specific (and expensive) tests.

Contrast this to the blood screening test done by the hospital blood bank. Here the two rare events must occur together in order for the undesirable outcome (HIV-positive blood put in the blood bank) to occur. The blood bank is happy to throw out the blood on the "better safe than sorry" principle, and the overall risk to the resulting blood supply is very small (about 4 in 1 million).

3.3 Independence and Conditional Independence

In the previous section we discussed how the knowledge about whether or not B occurred can affect our knowledge about A. When there is no effect, we say that the events are independent. Because B provides no information about A, $P(A|B) = P(A)$ and the multiplication rule reduces as in Eq. 3.6. We take this as our definition.

Definition. Independence. *Let A and B be two events. Then we say A and B are* independent *if and only if*

$$\mathrm{P}(A \cap B) = \mathrm{P}(A) \cdot \mathrm{P}(B) \; . \tag{3.6}$$

This is also called *Marginal Independence* to distinguish it from the related concept of *Conditional Independence* we will introduce shortly.

If $\mathrm{P}(A) > 0$ and $\mathrm{P}(B) > 0$, independence can be defined in terms of conditional probability as shown below (Pearl 1988). We use the standard definition because it works for events with zero probability.

Definition. Alternative Definitions of Independence. *If A and B are two events such that $\mathrm{P}(A) > 0$ and $\mathrm{P}(B) > 0$, then the following three statements are equivalent:*

$$A \text{ and } B \text{ are independent (Eq. 3.6)},$$

$$\mathrm{P}(A|B) = \mathrm{P}(A) = \mathrm{P}(A|\overline{B}) \ ,$$

and

$$\mathrm{P}(B|A) = \mathrm{P}(B) = \mathrm{P}(B|\overline{A}) \ ,$$

where $\overline{A}$ is the complement of the event A in Ω.

Independence corresponds to the cases in which we can simplify the calculation of the probability of complex events. This is a key result which we will use over and over again when building complex models for educational assessment. Often it is easier to determine that certain events are independent than to assess their joint probabilities. Once we have laid out the pattern of independence, we can simplify the probabilities we need to assess and compute. This becomes more interesting when more events are involved, so we need to expand our notion of independence to more events.

Definition. Mutual Independence. *Let $A_1, \ldots, A_n$ be a set of n events. These events are* mutually independent *if* $\mathrm{P}(A_1 \cap \cdots \cap A_n) = \prod_{i=1}^{n} \mathrm{P}(A_i)$ *and any smaller subset of those events is mutually independent.*

Pairwise independence does not imply mutual independence. The following example demonstrates the difficulty.

Example 3.7 (Agreement of Two Random Statements). *Consider a psychological survey designed to test the attitude of subjects towards certain topics. To see if the subjects' attitudes are consistent, the survey asks two questions about the same topic at different points in the survey. Let S_1 and S_2 be the events that a subject agrees with the two statements respectively and let C be the event that a subject's attitudes on the topic are consistent, either agreeing to both or disagreeing to both. Suppose one subject is answering the survey by flipping a coin for every statement. In this situation, S_1 and S_2 are independent. Also, $\mathrm{P}(S_1) = \mathrm{P}(S_2) = \frac{1}{2}$.*

Now C is functionally determined by S_1 and S_2; specifically, $C = (S_1 \cap S_2) \cup (\overline{S_1} \cap \overline{S_2})$. Therefore:

$$\begin{aligned}
\mathrm{P}(C) &= \mathrm{P}(S_1)\mathrm{P}(S_2) + (1 - \mathrm{P}(S_1))(1 - \mathrm{P}(S_2)) \\
&= \frac{1}{2} \cdot \frac{1}{2} + \frac{1}{2} \cdot \frac{1}{2} = \frac{1}{2} \\
\mathrm{P}(C \cap S_1) &= \mathrm{P}(S_1)\mathrm{P}(S_2) = \frac{1}{4} \\
&= \mathrm{P}(S_1)\mathrm{P}(C) = \frac{1}{4} \\
\mathrm{P}(C \cap S_2) &= \mathrm{P}(S_2)\mathrm{P}(C) = \frac{1}{4} \ .
\end{aligned}$$

Therefore, C and S_1 are pairwise independent, as are C and S_2. But when we look at all three events:

$$\begin{aligned} \mathrm{P}(C \cap S_1 \cap S_2) = \mathrm{P}(S_1 \cap S_2) &= \frac{1}{4} \\ &\neq \mathrm{P}(C)\mathrm{P}(S_1)\mathrm{P}(S_2) \ . \end{aligned}$$

Thus C, S_1, and S_2 are not mutually independent.

Recall that we defined probability as a state of knowledge. Knowledge about the response to one statement alone does not provide any knowledge about the other statement or the agreement between the two. But knowledge about one statement and the agreement between the two conveys exact information about the other statement. Thus, every pair of events is pairwise independent but the three together are not mutually independent.

3.3.1 Conditional Independence

As Example 3.7 shows, situations with more than two events can get quite complex. Fortunately, the opposite situation sometime occurs. Learning that an event C occurred can render two other events independent. We call this *conditional independence.*

Definition. Conditional Independence. *Let A, B, and C be three events. Then we say A and B are* conditionally independent *given C if and only if*

$$\mathrm{P}(A \cap B|C) = \mathrm{P}(A|C) \cdot \mathrm{P}(B|C) \ . \tag{3.7}$$

If A and B are conditionally independent given C, we write $I(A|C|B)$. This notation is from Pearl (1988). The intuition is that C separates A from B (Chap. 4 develops this intuition). The standard statistical notation (Dawid 1979) is $A \perp\!\!\!\perp B \mid C$. The notation $I(A|\emptyset|C)$, or $A \perp\!\!\!\perp B$, refers to marginal (unconditional) independence.

Conditional independence is a powerful tool for building models. By conditioning on the right things, we can often build quite complex models from smaller independent pieces. Consider for example the joint probability $\mathrm{P}(A, B, C, D, E, F)$ under the following set of independence relationships: $I(F|D, E|A, B, C)$, $I(E|C|A, B, D)$, $I(D|C|A, B)$, and $I(A|\emptyset|B)$. The recursive representation of the joint probability simplifies to the product of smaller factors as follows:

$$P(A, B, C, D, E, F) = P(F|D, E)P(E|C)P(D|C)P(C|A, B)P(B)P(A) \ .$$

The graphical models we develop in Chap. 4 combine conditional probability with representations and results from graph theory to support inference in even large collections of variables, if theory and experience suggest conditional independence relationships among them. (Moreover, the independence relationships are much easier to see in graphs than in the symbolic notation of the preceding example!)

It will be helpful to review a few examples that give us more intuition into how conditional independence works and where these conditional independence relationships come from. The next few subsections provide some illustrations that arise in educational testing.

3.3.2 Common Variable Dependence

Conditional independence is not the same as mutual independence. The following illustration, adapted from the "accident proneness" example of Feller (1968), illustrates the difference.

Example 3.8 (Accident Proneness). *Imagine a population with two types of individuals: N, normal, and $\overline{N}$, accident prone. And suppose that 5/6 of these people are normal, so that if we randomly select a person from this population, the probability that the chosen person is normal is $P(N) = 5/6$.*

Let A_i be the event that an individual has an accident in year i. For each individual, A_i is independent of A_j whenever $i \neq j$. Thus for each individual, whether or not that person has an accident, a Bernoulli process is followed. The accident probability, however, is different for the two classes of individuals.

$$P(A_i|N) = .01 \qquad\qquad P(A_i|\overline{N}) = .1$$

The chance of a randomly chosen individual having an accident in a given year follows from the Law of Total Probability, as a weighted average of the probability of an accident for normal individuals and for accident-prone individuals:

$$\begin{aligned} P(A_i) &= P(A_i|N)P(N) + P(A_i|\overline{N})P(\overline{N}) \\ &= \frac{.05}{6} + \frac{.1}{6} = \frac{.15}{6} = .025 \ . \end{aligned}$$

That is, $P(A_1) = P(A_2) = .025$.

The probability that a randomly chosen individual has an accident in both the first and second year follows from the Law of Total Probability and the fact that A_1 and A_2 are independent for a given individual. *It too is a weighted average, now of the probability of an accident in both years for normal individuals and for accident-prone individuals:*

$$\begin{aligned} P(A_1 \cap A_2) &= P(A_1 \cap A_2|N)P(N) + P(A_1 \cap A_2|\overline{N})P(\overline{N}) \\ &= P(A_1|N)P(A_2|N)P(N) + P(A_1|\overline{N})P(A_2|\overline{N})P(\overline{N}) \\ &= .01 \times .01 \times \frac{5}{6} + .1 \times .1 \times \frac{1}{6} \\ &= \frac{.0005}{6} + \frac{.01}{6} = \frac{.0105}{6} = .00175 \ . \end{aligned}$$

Note that

$$P(A_2|A_1) = \frac{P(A_1 \cap A_2)}{P(A_2)} = \frac{.00175}{.025} = .07 \; .$$

But $P(A_2) = .025$, so $P(A_2) \neq P(A_2|A_1)$. Therefore, A_1 and A_2 are not (unconditionally) independent!

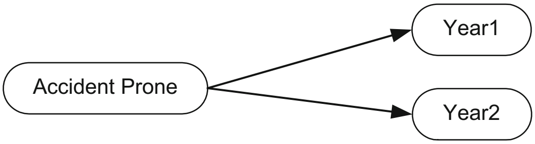


Fig. 3.2 Graph for Feller's accident proneness example
Reprinted with permission from ETS.

The explanation for this phenomenon lies with the interpretation of probability as a state of information. When we learn that the individual in question has had an accident during the first year, it provides information about whether or not he is accident prone, which in turn provides information about what will happen during the next year. In general, whenever a variable (in Feller's example, accident-proneness) that determines the distribution of a set of observations (whether an individual has an accident in each year) is unknown, information about one sample value (accident in Year i?) provides information about the others (accident in Year j?) through the variable (accident-proneness). This is the essence of common variable dependence.

A common example from educational testing is unidimensional item response theory (IRT). Here the latent trait θ accounts for all of the dependence between the observations. This structure, shown as Fig. 3.3, is sometimes called the "naïve Bayes" model. It does not always work well, like when the underlying interrelationships among observable variables are complex, such as medical symptoms that tend to appear in clusters. In assessment, though, tasks can be engineered to make this model fit pretty well (Almond and Mislevy 1999).

3.3.3 Competing Explanations

Conditioning on multiple events requires only a straightforward generalization of the notation. It is worth exploring an example of a situation that arises in diagnostic assessment.

Example 3.9 (Conjunctive Skills Model). *Suppose θ_1 and θ_2 represent two skills (e.g., reading and writing) and X represents performance on a task which requires both (e.g., document-based writing task). Poor performance on the task could be a sign of lack of either of the skills. Suppose we learned*

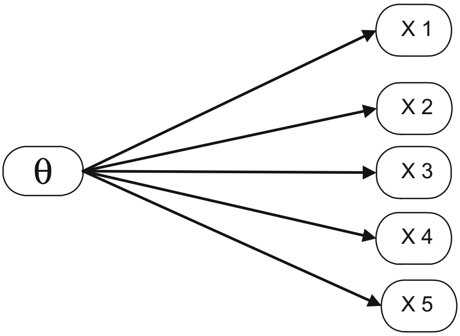


Fig. 3.3 Unidimensional IRT as a graphical model
Reprinted with permission from ETS.

(from an earlier reading test) that the reading skills of the examinee were high; we would then conclude that there was a deficiency in writing. Thus, observing the performance on the task induces a dependency in our knowledge about the skill variables. (See Problem 3.6.)

Figure 3.4 shows this example graphically. Knowing the state of the agreement allows us to "complete the knowledge circuit" between the two statements. Conditioning on common descendents induces dependencies. This is the *Competing Explanation* phenomenon. This is the intuition behind the concept of D-separation, defined in Chap. 4.

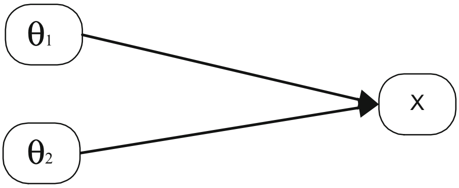


Fig. 3.4 Variables θ_1 and θ_2 are conditionally dependent given X. Even though θ_1 and θ_2 are marginally independent, if X is known they become dependent
Reprinted with permission from ETS.

3.4 Random Variables

Setting a variable on the basis of a random event produces *random variables.* Random variables are very convenient to work with, and we will see numerous examples in educational testing. For example, if we perform an experiment that consists of selecting a student from a school and giving that student a test and a questionnaire, we could define numerous random variables associated with that event: the age of the student, the response given to item 3, an outcome variable representing whether the response was correct or not. Naturally, these variables will not be independent and characterizing that dependence in educationally useful ways will be the subject of most of the rest of this book.

There are generally three types of random variables: categorical random variables whose outcomes are members of a category, possibly ordered; integer–valued random variables whose outcomes are members of a subset of the integers; and real–valued random variables whose outcomes are members of a subset of the real line. Categorical and integer-valued random variables are called *discrete* and real-valued random variables are called *continuous*.

The topic of random variables is usually well covered in basic statistics texts. This section provides some basic definitions to support the discussion of Bayes theorem in Sect. 3.5.

3.4.1 The Probability Mass and Density Functions

For a discrete random variable, the probability of each *atom*—outcome with nonzero probability—of the distribution completely characterizes the distribution. If the random variable X has range $\{x_1, \ldots, x_n\}$, then we can reconstruct the probability measure from:

$$p(x_i) = \mathrm{P}(X = x_i) \ . \tag{3.8}$$

This is known as the *probability mass function* or *p.m.f.*, and is usually written $p(\cdot)$. We can think of the random variable as being generated by an urn filled with balls with numbers printed on the side. The p.m.f. $p(x_i)$ indicates the proportion of balls with x_i written on them.

Consider any set A of possible outcomes for a discrete random variable X. We can calculate the probability that the outcome will fall into that set as follows:

$$\mathrm{P}(X \in A) = \sum_{x_i \in A} p(x_i) \ . \tag{3.9}$$

All p.m.f.s can be characterized by two properties:

1. $1 \geq p(x) \geq 0 \qquad \forall x$
2. $\sum_{\text{all } x} p(x) = 1 \qquad$ (normalization)

Any function $p(\cdot)$ satisfying these two properties is a p.m.f. for some random variable. The second property is particularly important. It is known as the *normalization* constraint. Any non-negative function $g(x)$ defined on $\{x_1, \ldots, x_n\}$ for which the *normalization constant*, $\sum_{\text{all } x} g(x)$, is finite can be *normalized* to produce a p.m.f. by dividing through by the normalization constant. That is, we can obtain a p.m.f. $p(x)$ from $g(x)$ as

$$p\left(x_i\right) \equiv g\left(x_i\right) \Big/ \sum\nolimits_{\text{all} x} g\left(x\right).$$

Continuous random variables present us with an additional problem. For one thing, our canonical example of objective probability, balls from an urn, no longer works. Example 3.10 presents a new canonical example for continuous distributions.

Example 3.10 (Random Point on a Line Segment). *Let $\Theta = [0,1]$ be the unit line segment, and consider an experiment that consists of randomly selecting a point from that line. Let $A \subseteq \Theta$ be any set consisting of a collection of disjoint intervals. The* probability *of the Event A is the total of the length of all the intervals making up the set A.*

Note that this example only defines probability for collections of disjoint intervals. This allows us to avoid some pathological cases. At the same time, the collection of disjoint intervals covers the most practically useful cases, so we lose little by doing this.

For continuous random variables, the mass associated with any specific outcome is always zero (think of the length of a single point). But since the set of outcomes is dense, we can consider the *density* of the probability in a small region around the outcome of interest. Thus we define the *probability density function* or *p.d.f.* by

$$f(x) = \lim_{\Delta x \to 0} \frac{\mathrm{P}(x \leq X \leq x + \Delta x)}{\Delta x} \ . \tag{3.10}$$

The p.d.f. behaves very much like the p.m.f, except that in most of the formulas, sums are replaced by integrals. Thus if X is a continuous random variable and A is a set of possible outcomes for X then:

$$\mathrm{P}(X \in A) = \int_A f(x)\, dx \ . \tag{3.11}$$

Similarly, the normalization constant is defined by

$$\mathrm{P}(X \in \Omega) = \int_\Omega f(x)\, dx \ , \tag{3.12}$$

where Ω is the set of all possible values for X which have nonzero values of $f(\cdot)$ (sometimes called the *support* of f). Normalizing a p.d.f. is analogous to normalizing a p.m.f. Thus, if $g(\cdot)$ is a nonnegative function whose integral over the whole real line exists and is equal to m, the normalized probability density function is $f(x) = g(x)/m$.

A third useful representation for probability measures is the (cumulative) *distribution function* or *d.f.* (or *c.d.f.*). It is defined the same way for continuous and discrete random variables:

$$F(x) = \mathrm{P}(X \leq x) \ . \tag{3.13}$$

$F(x)$ is thus the probability of all outcomes less than or equal to x. For a discrete random variable

$$F(x) = \sum_{y \leq x} p(y) \ , \tag{3.13a}$$

while for a continuous random variable

$$F(x) = \int_{-\infty}^{x} f(y)\, dy \ . \tag{3.13b}$$

The distribution function uniquely defines the probability measure. The term *distribution* is often used to represent any function (d.f., p.d.f., p.m.f., or a probability measure) that uniquely defines the probability measure.

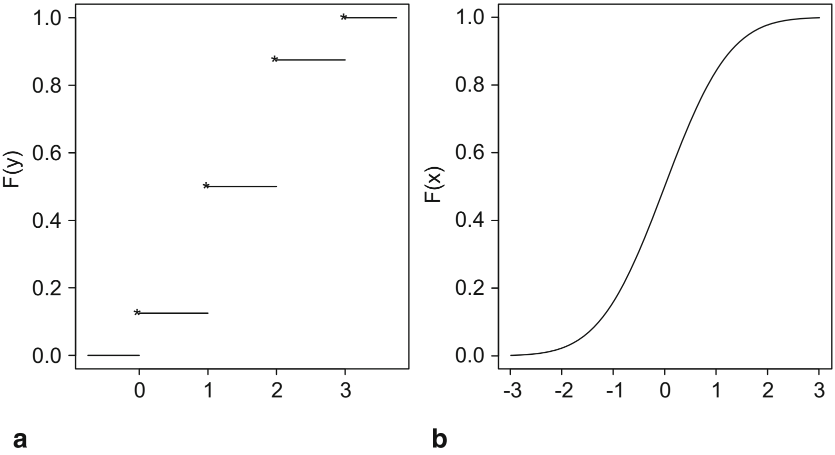


Fig. 3.5 Examples of discrete and continuous distributions. **a** Discrete distribution. **b** Continuous distribution

Reprinted with permission from Almond (1995) with permission from Taylor & Francis Group.

Figure 3.5a, b shows an example of both a discrete and a continuous distribution function. We can see some common features, in particular:

1. Distribution functions are always nondecreasing: That is, $x \leq y$ implies $F(x) \leq F(y)$.
2. They range between zero and one, that is, $0 \leq F(x) \leq 1$ with $F(-\infty) = 0$ and $F(+\infty) = 1$.

The discrete distribution (Fig. 3.5a) is a step function that takes jumps at the atoms (points of nonzero probability) of the distribution. The jump of each step is the probability associated with that particular atom, and the height is the probability of observing a value less than or equal to the value of that atom. For example, the mass associated with the atom 2 for the distribution pictured in Fig. 3.5a is .375. Thus, there is a one-to-one relationship between the p.m.f. and the distribution function for discrete probability relationships.

The distribution function of the continuous distribution (Fig. 3.5b) is continuous at every point in its domain (this is where it gets its name; this property is called being absolutely continuous). We can recover the p.d.f. by:

$$f(x) = dF(x)/dx \ . \tag{3.14}$$

Finally, independence can be characterized in terms of the probability mass or density function.

Definition. Independence of Random Variable. *A series of random variables $X_1, \ldots, X_n$ are* independent *if and only if*

$$\begin{aligned} \textit{discrete case} \quad & p_{\mathbf{X}}(\mathbf{x}) = p_{X_1}(x_1) \cdot p_{X_2}(x_2) \cdots p_{X_r}(x_r) \\ \textit{continuous} \quad & f_{\mathbf{X}}(\mathbf{x}) = f_{X_1}(x_1) \cdot f_{X_2}(x_2) \cdots f_{X_r}(x_r) \ . \end{aligned} \tag{3.15}$$

3.4.2 Expectation and Variance

One of the most useful features of random variables is that we can calculate expected values for functions of those variables.

Definition. **Expected Value.** *Let X be a random variable and let $h(x)$ be a function defined on the range of that random variable. Define the* expected value of $h(X)$, *denoted* $\mathrm{E}[h(X)]$, *to be:*

$$\mathrm{E}[h(X)] = \int_{\text{all } X} h(x)\, dF(x) \ , \tag{3.16}$$

if the integral exists. This so-called Lebesque–Stiltjes[4] *integral is a compact way of writing analogous expressions for discrete and continuous random variables.*

[4] A Lebesque–Stiltjes integral is a generalization of ordinary integration that is performed with respect to a *measure* (e.g., a probability distribution). If the measure is continuous, then it becomes an ordinary integral with respect to the density. If the measure is a counting measure (like a discrete probability) it becomes a sum. Thus, it handily unifies a lot of statistical formulas that are integrals for continuous distributions and sums for discrete distributions.

If X is discrete random it expands to the sum

$$\sum_{\text{all } X} h(x)p(x)$$

and if X is continuous it is the integral

$$\int_{\text{all } X} h(x)f(x)\,dx.$$

In the special case where $h(x) = x$, $\mathrm{E}[X]$ *is the* expected value of the random variable *X.*

$\mathrm{E}[X]$ is also called the *mean* of X and is often written $\overline{X}$. The mean is a measure of location, the "center of gravity" of the distribution.

Example 3.11 (Resampling Distribution). *We can create a probability distribution corresponding to an observed sample as follows: Let $x_1, \ldots, x_m$ be the unique values observed in a sample of size N. Let n_i be the number of times x_i is observed. Define the following p.m.f.:*

$$p(x_i) = \frac{n_i}{N} \; .$$

This is the resampling p.m.f. for "sampling from a sample." Let X be a random variable corresponding to the experiment. We draw a value at random from the set of values in the sample. Then the expected value of X is the average of the sample. The laws of probability theory say that if the original sample size (N) was large enough, the resampling (bootstrap) distribution should approach the original distribution function.

A second special expected value is the variance, which measures the amount of uncertainty associated with a random variable (or a process whose outcome is expressed in terms of the value of a random variable).

Definition. Variance. *Let X be a random variable that has a finite mean* $\mathrm{E}[X] = \mu$*. Then the* variance of variance of *X is the expectation of $(X - \mu)^2$ (if it exists), written* $\mathrm{Var}(X)$*.*

The variance is a measure of spread of a distribution, specifically the expected value of the squared distance from the mean. Its value is always greater than or equal to zero. As the variance gets closer to zero, the state of information about the random variable becomes more certain. The units of the variance are the squared units of the original random variable, which is not natural to think about. For that reason the *standard deviation*, which is the square root of the variance, is often used instead of the variance to describe the dispersion of a distribution. The reciprocal of the variance is called the *precision*. The smaller the variance, the larger the precision of the distribution. The precision turns out to be useful in calculations involving the normal distribution (especially using Bayes theorem).

Example 3.12 (Normal Distribution). *Let X be a random variable with the following probability density function:*

$$f(x) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{1}{2}\left(\frac{X-\mu}{\sigma}\right)^2} \;. \tag{3.17}$$

Then $\mathrm{E}[X] = \mu$ *and* $\mathrm{Var}(X) = \sigma^2$*. We say that X follows a normal distribution with parameters μ and σ^2, and write $X \sim \left(\mu, \sigma^2\right)$.*

A normal distribution is completely defined by its mean and variance. This means we can approximate any distribution function by a normal with the same mean and variance. Sometimes this is a good approximation and sometimes it is a bad one, but it turns out to be quite good in a large number of important situations. In particular, the central limit theorem implies that the totals and averages of reasonably well-behaved random variables are approximately normally distributed. For a normal distribution, approximately 2/3 of the outcomes lie within 1 standard deviation of the expected value and approximately 95 % of the outcomes lie within 2 standard deviations.

Example 3.13 (Monte Carlo Integration). *Let X be a random variable with distribution $F(X)$. Let $h(X)$ be a function whose expectation and variance over the distribution $F(X)$ we would like to know. In general* $\mathrm{E}[h(X)]$ *may be difficult to calculate. However, if we can take a sample $X_1, \ldots, X_n$ from $F(X)$ we can get an approximation to the expectation and variance:*

$$\begin{aligned}
\mathrm{E}[h(X)] &\approx \frac{\sum_{i=1}^{n} h(X_i)}{n} \\
\mathrm{Var}(h(X)) &\approx \frac{\sum_{i=1}^{n} (h(X_i) - \mathrm{E}[h(X)])^2}{n-1}
\end{aligned}$$

With this expectation and variance we can find the closest approximating normal distribution for $h(X)$.

This Monte Carlo Integration is a useful trick which we will use when trying to learn parameters for educational testing models.

3.5 Bayesian Inference

Although Bayesian statistics centers around Bayes theorem, it really represents a statistical philosophy (or philosophies, see Good 1971). The central pillar of this philosophy is that a state of information about an unknown event, variable, or parameter can be represented with a probability distribution. Although initially controversial because of this extra assumption, Bayesian statistics has proved quite powerful. It provides a guiding principle for building and reasoning about complex models, and provides correct solutions to problems that were not tractable under the classical approach (treating parameters as fixed but unknown quantities). Furthermore, modern computing technology has made possible techniques like Markov Chain Monte Carlo (MCMC) which can solve quite complex problems as long as they can be cast in the Bayesian framework.

This book builds up a Bayesian discipline of psychometrics, with a particular focus on discrete observable variables and proficiency variables. In the process of building that discipline it will use the fundamental ideas of Bayesian statistics over and over. This section provides a brief review of those fundamentals. More thorough treatments can be found in Lee (1989), DeGroot (1970), Box and Tiao (1973), Gelman et al. (2013a) or for a more mathematical treatment, Berger (1985).

3.5.1 Re-expressing Bayes Theorem

We introduced Bayes theorem in Sect. 3.2 in terms of probabilities of events. Here is how it looks when written in terms of densities:

$$p\left(y\left|x^{*}\right.\right)=Kf\left(x^{*}\left|y\right.\right)p\left(y\right), \tag{3.18}$$

where $p(y)$ is the density of the random variable y, $f\left(x \mid y\right)$ is the conditional density of another random variable x given y, x^{*} is a particular value of x, $p\left(y \mid x^{*}\right)$ is the conditional density of y given that $x = x^{*}$, and K is the normalization constant needed to make $p\left(y \mid x^{*}\right)$ integrate or sum to one. That is,

$$K^{-1}=\int_{\text{all }y} f\left(x^{*}\mid y\right)d\,P\left(y\right)=p\left(x^{*}\right) \ . \tag{3.19}$$

The value of K is not directly relevant to inference about x or y under the Bayesian paradigm (although it is a consideration in calculations). Writing Bayes theorem only up to proportionality focuses attention on the important pieces:

$$p\left(y\mid x^{*}\right)\propto p\left(y\right)f\left(x^{*}\mid y\right) \tag{3.20}$$

or

$$\text{posterior}\propto\text{prior}\times\text{likelihood}\ .$$

The next section walks through these pieces in detail.

3.5.2 Bayesian Paradigm

Example 3.14 (Propensity to Make Free Throws). *Suppose that a certain individual has a probability θ for making a basketball free throw.*[5] *Suppose further that each time the individual attempts the shot, the outcomes*

[5] A free throw or foul shot in basketball is an attempt to throw the ball from a fixed line into the basket. It is awarded for a penalty, so the player can always attempt the shot without the interference of opponents. Thus, each attempt is made under reasonably repeatable conditions.

are independent given the probability of success and that the probability of success remains unchanged. Our problem is to characterize what we believe about θ if we observe this individual have S successes in n attempts.

This situation is mathematically equivalent to the binomial model for test scores: An assessment consists of n tasks, each student is characterized by a single propensity variable θ that gives probability of a correct answer to each task, and all the responses are conditionally independent given θ.

Let us look first at how this problem would be handled under the "classical statistical" approach. Recall that the probability of observing S successes in n attempts for a given value of θ is given by the binomial distribution:

$$Pr(S|\theta, n) = \binom{n}{S} \theta^S (1-\theta)^{n-S} \ , \tag{3.21}$$

and its mean and variance are $n\theta$ and $\theta\left(1-\theta\right)/n$. The most common way to estimate θ is by the observed proportion of successes, or $\hat{\theta} = S/n$. The observed proportion $\hat{\theta}$ is the least squares estimate of θ, and it is also an unbiased estimate. That is, if increasingly many samples of size n were taken, the mean of the $\hat{\theta}$s would be θ. Their variance would be $\theta\left(1-\theta\right)/n$—which would not be known in practice because it depends θ, so is usually approximated by $\hat{\theta}\left(1-\hat{\theta}\right)/n$. The approximated standard error of estimation $\widehat{\mathrm{SEM}}$ is the square root of this quantity. When $n = 10$ and $S = 7$, $\hat{\theta} = .7$ and $\widehat{\mathrm{SEM}}\left(\hat{\theta}\right) = .145$. An approximate 95 % confidence interval, obtained by adding and subtracting 1.96 times $\widehat{\mathrm{SEM}}$ around $\hat{\theta}$, is $(.416, .984)$.

One obvious shortcoming with this approach is that it breaks down when the observations are all failures or all successes. In these cases $\widehat{\mathrm{SEM}}$ is zero, which implausibly suggests there is no uncertainty associated with 0 or 1 as an estimate of θ.

A bigger problem is that the reasoning is in the wrong direction. It tells us what the distribution of the estimator $\hat{\theta}$ would be in repeated samples, given that we know the true value of θ. But we are not taking repeated samples; we generally have only the one realized sample. And we do not know θ; that is what we want to make inferences about in the first place. The classical approach does allow us to make some rather indirect statements such as "If θ were .7 then the probability of observing an $S \geq s$ would be ..." There is a natural tendency of statistical consumers to misinterpret such statements as probabilistic statements about θ.

Maximum likelihood estimation, developed by R. A. Fisher in the 1920s, is a more sophisticated classical approach. Once we actually observe a particular value of S for a given number of attempts, Eq. 3.21 is reinterpreted as a function of θ given the observed value of S. This is the likelihood, which corresponds to the piece $f\left(x^* \mid y\right)$ in Eq. 3.20 with θ playing the role of y and the observed value of S playing the role of x^*.

From a Bayesian point of view, computing the likelihood function based on the realized sample is a step in the right direction: It conveys the evidence that the sample we actually observed holds about θ. Figure 3.6 shows the likelihood that is induced for θ when $n = 10$ and $S = 7$. Seven successes could occur if θ is any value of than 0 or 1, but it is more likely at some values than others. The figure shows that seven successes are very unlikely to occur for low values of θ (we would usually see fewer successes) and also unlikely for high values of θ as well (we would usually see more successes). The relative heights of the likelihood indicate just how likely 7 of 10 would be at each possible value of θ. It is about three times as great at .5 as it is at .4, for example, and it takes its highest value when $\theta = .7$. The observed proportion of successes is in fact the maximum likelihood estimate (MLE) of θ under the binomial distribution. These are all statements that can be read directly from the likelihood function. They concern only the observed sample, not the distribution of S or of estimates of θ in repeated samples, and not an unknown true value of θ.

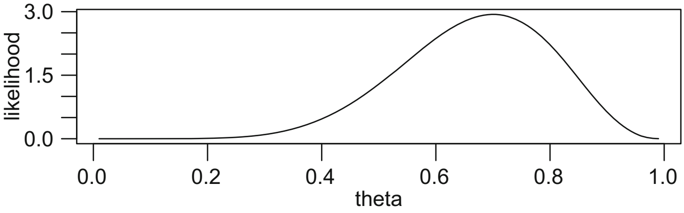


Fig. 3.6 Likelihood for θ generated by observing 7 successes in 10 trials Reprinted with permission from ETS.

Of course MLEs can also be interpreted in terms of sampling distributions. This is how Fisher used them, and most people do today. Their properties under repeated sampling of S for known parameter values of θ are derived, and the problematic interpretation of estimates and standard errors noted above return in full force.

The Bayesian paradigm uses the evidence in the likelihood function in a different way. It allows us to coherently express our belief about θ, conditional on the observed sample, in terms of a probability distribution for θ. We might plot a variable's posterior density to give a full picture of what we believe after obtaining the new information (especially useful if the shape is unusual or it has multiple modes), or we might summarize the information in terms of its mean or its mode, and its standard deviation, or an interval that contains 95 % of the posterior posterior. The posterior mean, sometimes used a point estimate, is called an EAP or expectation a posteriori estimate. The posterior mode is called an MAP or maximum a posteriori estimate.

The key is that in order to express belief about θ in terms of a probability distribution *after* experiment, we must also express our belief about it *before* the experiment. Classical statisticians are reluctant to do this, but as the Bayesian statistician Jimmie Savage said, you do not get to enjoy the Bayesian omelet unless you break the Bayesian eggs.

Let us take the gentlest possible step in the Bayesian direction: A uniform distribution over the interval $[0, 1]$ as the prior distribution; that is, $p(\theta) = 1$. This says that before we see the player take the free throws, we have no reason a priori to think that any possible value of θ is more probable than any other. (This would not be the case if we knew something about the basketball player, and we will discuss this in the next section.)

Example 3.15 (Propensity to Make Free Throws; Example 3.14 continued). *Substituting the uniform prior for θ into Eq. 3.20, the proportionality form of Bayes theorem, gives*

$$\begin{aligned} p\left(\theta \mid n=10, S=7\right) &\propto p\left(\theta\right) \mathrm{P}\left(S=7 \mid n=10, \theta\right) \\ &\propto 1 \times \theta^{7}\left(1-\theta\right)^{3} \\ &= \theta^{7}\left(1-\theta\right)^{3} \ . \end{aligned} \tag{3.22}$$

The normalizing constant is an instance of a general form called the beta function. The resulting posterior distribution is an member of a family of distributions called beta distributions—specifically, a Beta$(8, 4)$ *distribution. Beta distributions are central in Bayesian inference about binary data, and will be discussed in greater detail in the following section. With a uniform prior, the posterior for θ has exactly the same shape as the likelihood function (Fig. 3.6), although it is rescaled so it can be interpreted as a probability density. For example, the mode of the posterior is .7. A calculator for the Beta distribution tells us further that the mean is .667, the standard deviation is .131, and the region containing the most probable 95 % of the posterior is* $(.390, .891)$.

We can use the posterior density to express our belief about θ after having observed S. A person who misinterprets a classical 95 % confidence interval to mean that there is a 95 % probability that theta is between its bounds is implicitly making a Bayesian inference assuming a uniform prior. But this is the correct way to interpret the Bayesian posterior credibility interval, $(.390, .891)$.

We used the uniform prior for expository reasons, but trying to understand just where it comes from, and why it rather than something else, is a deeper question. Rev. Bayes himself had so much difficulty with the uniform assumption that he did not publish the paper from which the Bayes theorem takes its name during his lifetime (it was published posthumously). The remaining sections in this chapter say more about where priors come from and how to construct them.

One simple technique often used when there are questions about the prior distribution is to perform a *sensitivity analysis* on the choice of priors. To perform a sensitivity analysis, one simply redoes the analysis with several different choices of prior. (See Exercise 3.2). If the results change substantially, then they need to be viewed with some skepticism: Your inference depends notably on Your choice of prior, which means that the data are not (through the likelihood) providing enough information for a solid answer.

A thorough sensitivity analysis will check assumptions about the likelihood as well as in the prior. The prior is not the only assumption in Example 3.14. The independent, identically distributed assumption that underlies the binomial distribution can also be questioned. We are assuming that the individual is not learning how to perform the task better during the experiment. This might be approximately true if the experiment is short in duration, but if the experiment goes on for a long time it will become increasingly dubious. (Modeling learning on the part of the test taker is a more difficult problem, which we will not explore in great depth in this book.) Both the Bayesian and classical approach to this problem share the i.i.d. assumption, and the Bayesian analysis is much more sensitive to this assumption than to reasonable choices of prior.

3.5.3 Conjugacy

The distribution in Eq. 3.22 is an example of a *beta distribution*. The beta distribution has the following p.d.f.:

$$f(\theta|a,b) = \left[\frac{1}{B(a,b)}\right] \theta^{a-1}(1-\theta)^{b-1} \ , \tag{3.23}$$

where the normalizing constant $B(a,b)$ is the *beta function*[6],

$$B(a,b) = \int_0^1 t^{a-1}(1-t)^{b-1}\, dt. \tag{3.24}$$

The mean of the beta distribution is $a/(a+b)$, the mode is $\frac{a-1}{a+b-2}$ and its variance is $\frac{ab}{(a+b)^2(a+b+1)}$. Both a and b must be greater than zero for this to be a proper distribution (otherwise the beta integral is infinite). Note that the uniform distribution is a special case of the beta distribution, corresponding to a $\text{Beta}(1,1)$ distribution (i.e., with $a=1$ and $b=1$).

Figure 3.7 shows several beta distributions. We see that when a is equal to b, $\text{Beta}(a,b)$ is symmetric and is centered at .5. When $a > b$, the distribution shifts lower, and when $a < b$, the distribution shifts higher. The greater $a+b$, the more concentrated it is (that is, the higher the central peak). When a or b is less than 1, then it shoots up at the upper and/or lower tail. (This would look flat if the x-axis was plotted on a logistic scale, as is often done for probabilities.)

[6] A beta function can also be written in terms of gamma functions: $B(a,b) = \Gamma(a)\Gamma(b)/\Gamma(a+b)$.

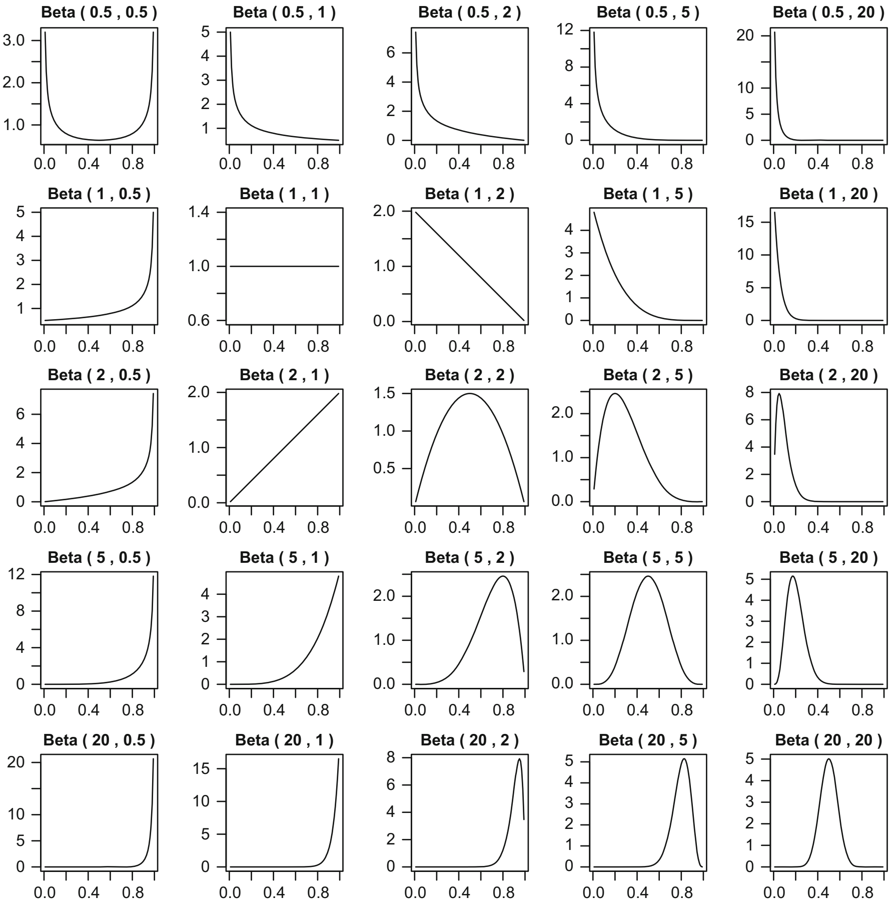


Fig. 3.7 A panel of sample beta distributions
Reprinted with permission from ETS.

The beta distribution is handy for representing knowledge about probabilities because its range is restricted to the interval $[0, 1]$. It is even handier when the likelihood is a binomial distribution, because if the prior distribution is a beta and the likelihood is a binomial distribution, then the posterior distribution will be a beta as well. Specifically, if the prior is Beta(a, b) and the data consist of S successes in n trials, then the posterior is:

$$\begin{aligned} f(\theta|a,b,S,n) &\propto \mathit{prior} \times \mathit{likelihood} \\ &\propto \theta^{a-1}(1-\theta)^{b-1} \times \theta^{S}(1-\theta)^{n-S} \\ &\propto \theta^{a+S-1}(1-\theta)^{b+n-S-1} \ , \end{aligned}$$

which is also a beta distribution, Beta$(a + S, b + n - S)$.

Note the similarity in the functional forms of the beta prior and the binomial likelihood. The difference is that in the beta distribution, θ is variable and a and b are the parameters, while the likelihood comes from the binomial distribution with S is the observed variable and n and θ are its parameters. It follows that when the beta distribution is used as a prior for observations that follow a binomial or Bernoulli distribution, it expresses information equivalent to a hypothetical experiment with $a + b - 2$ observations, of which $a - 1$ were successes and $b - 1$ were failures.

The beta and binomial distribution share a special relationship with each other in Bayesian inference: When the prior is a beta distribution and the likelihood is a binomial distribution, then the posterior is always a beta distribution too. We will see that the normal distribution shares a similar relationship with itself: If both the likelihood and the prior distribution for the mean are normal (and the variance is known), then the posterior distribution for the mean will be normal too. Distribution families for prior and likelihood with this special property are known as *conjugate families.* The *Beta-Binomial* and *Normal–Normal* we will see shortly are well-known examples.

Example 3.16 (Propensity to Make Free Throws; Example 3.14 continued). *Consider the same individual from Example 3.14, and suppose that individual will make another m attempts at the same free throw. Based on the previous data, our prior distribution for success on the new set of attempts is now a* $\mathrm{Beta}(S, n - S)$*. Suppose we observe T successes in the second set of m attempts. Then our posterior for the individual's success after the second set will be a* $\mathrm{Beta}(S + T, n - S + m - T)$ *distribution.*

Note that even if we had reversed the orders of the two sessions of testing (the one with n trials and the one with m trials), we still would have reached the same conclusion at the end. We also would have reached the same conclusion if we did the testing in one large session and observed $S + T$ successes in $n + m$ trials.

This property has an interesting application. It means that we can use Bayesian inference as a model for our learning about the individual from Examples 3.14–3.16. We start with a prior distribution for θ representing our knowledge about this individual if he "dropped out of the sky"—a very weak prior such as $\mathrm{Beta}(1, 1)$, for example. We then observe some data about the individual and update our beliefs (using Bayes theorem) to generate a new posterior distribution. That posterior then becomes the prior for new data.

Thus Bayesian inference is a powerful tool for keeping track of our state of knowledge about the student as the student interacts with an assessment task by task, or in a series of assessments.

There are other conjugate families beside the Beta-Binomial. Suppose our task attempt produces a categorical outcome rather than the success/failure of Example 3.14. An example would be rolling a possibly loaded die, with possible outcomes $\{1, ..., 6\}$. Let $\boldsymbol{\theta} = \{\theta_1, \ldots, \theta_K\}$ be a vector of k probabilities of observing each of the K outcomes on a single trial. We observe a multinomial outcome $\mathbf{S} = \{S_1, \ldots, S_K\}$ where S_k is the number of outcomes in category k in n trials. The likelihood induced by $\mathbf{S}$ would now be a multinomial distribution:

$$p\left(\mathbf{S}\,|n, \boldsymbol{\theta}\right) \propto \prod_k \theta_k^{S_k}$$

In this case, the conjugate prior is a generalization of the beta distribution called the *Dirichlet distribution.* The parameter is a vector of K random values between zero and one, $\boldsymbol{\theta} = \{\theta_1, \ldots, \theta_K\}$, with the restriction that $\sum_{k=1}^{K} \theta_k = 1$. The Dirichlet distribution then has the following density function:

$$f(\boldsymbol{\theta}|\boldsymbol{\alpha}) = C \prod_{k=1}^{K} \theta_k^{\alpha_k - 1}, \tag{3.25}$$

with $\sum_{k=1}^{K} \theta_k = 1$, and normalizing constant C.

The mean and variance of each component of θ in the Dirichlet are simple functions of the αs. Let $m = \sum_k \theta_k$. Then $\mathrm{E}[\theta_i] = \alpha_i/m$ and $\mathrm{Var}[\theta_i] = \left[\alpha_i\left(m - \alpha_i\right)\right] / \left[m^2\left(m+1\right)\right]$. In analogy to the beta distribution, we can think about the Dirichlet as the amount of information about the vector of multinomial probabilities $\boldsymbol{\theta}$ conveyed by observing a total of m trials, with α_k occurring in each category k.

If we then use a Dirichlet prior with parameters $\boldsymbol{\alpha} = \{\alpha_1, \ldots, \alpha_K\}$ and observe a multinomial outcome $\mathbf{S} = S_1, \ldots, S_K$ where S_k is the number of outcomes in category k in n trials, then the posterior will be a Dirichlet$(\alpha_1 + S_1, \ldots, \alpha_K + S_K)$ distribution:

$$f(\boldsymbol{\theta}|\boldsymbol{\alpha}, \mathbf{S}) \propto \prod_{k=1}^{K} \theta_k^{\alpha_k + S_k - 1}.$$

This distribution will come back again in Chap. 8 where we try to learn Bayesian networks (which consist mainly of multinomial distributions) from data.

Another convenient conjugate family is the Normal–Normal family. Suppose our prior distribution for an unknown proficiency variable, θ, for an individual is a normal distribution with mean μ and variance τ^2, written $\mathrm{N}(\mu, \tau^2)$. Let X be the score of that individual on an assessment designed to measure that proficiency. Under classical theory with normal errors, the probability function for a student's observed score X given her true score θ is $\mathrm{N}(\theta, \sigma^2)$, where σ^2 is the error variance, which we will assume is known. In this situation the likelihood for θ induced by observing X is also normal, $\mathrm{N}(X, \sigma^2)$. Then the posterior distribution for θ is

$$\boldsymbol{\theta}|X, \mu, \sigma^2, \tau^2 \sim \mathrm{N}\left(\frac{\mu/\tau^2 + X/\sigma^2}{1/\tau^2 + 1/\sigma^2}, \frac{1}{1/\tau^2 + 1/\sigma^2}\right).$$

(In this context, the symbol $\sim$ should be read, "is distributed as.")

To avoid taking all those reciprocals, Bayesians often work with the *precision* (the reciprocal of the variance) rather than the variance. Let $U = 1/\tau^2$ and $V = 1/\sigma^2$. Then

$$\theta | X, \mu, U, V \sim \mathrm{N}\left(\frac{U\mu + VX}{U+V}, \frac{1}{U+V}\right).$$

The posterior mean is thus a precision-weighted average of the mean of the mean and the mean of the likelihood. The posterior precision is the sum of the prior precision and the precision from the likelihood; i.e., the information about θ from the prior and from the observations.

To extend this reasoning, suppose that we have another assessment, Y, which measures the same proficiency. Let the error variance of Y be σ_Y^2 and its reciprocal, or precision, be W. Then the posterior from observing both X and Y will be

$$\theta | X, Y, \mu, U, V, W \sim \mathrm{N}\left(\frac{U\mu + VX + WY}{U+V+W}, \frac{1}{U+V+W}\right).$$

Thus the precision of the two assessments taken together is the sum of the precision of the individual instruments added together. Continuing in this fashion, we can see what happens as we gather more and more data (more and more assessments). First, the precision of our posterior will get larger and larger (i.e., its variance will get smaller and smaller). Second, the weight of the prior will get smaller and smaller with respect to the weight of the data, so if we have enough data it will eventually overwhelm the prior. (See Exercise 3.9).

What if we do not know the variance for our assessment in the above example? If both the mean and variance are unknown, we need to integrate out over the unknown variance to draw inferences about the mean. The resulting marginal posterior distribution for the mean will be a Student t distribution. If we knew the mean but not the precision, the natural conjugate prior family for the precision would be the gamma distribution, which for certain parameter values is also a chi-squared distribution. Gelman et al. (2013a) develop these cases in greater detail.

The known-mean unknown-precision situation appears in MCMC estimation for models that use the normal distribution such as estimating conditional probabilities in Bayes nets (Chap. 9). The parameterization for the gamma distribution that (Gelman et al. 2013a) and the `WinBUGS` (Lunn et al. 2000) computer program use has the following p.d.f.:

$$\frac{b^a}{\Gamma(a)} x^{a-1} e^{-bx} \tag{3.26}$$

with both a and b positive. The support of gamma is the positive half-line, as is appropriate for a prior for precision. Its mean is $\frac{a}{b}$ and the variance is $\frac{a}{b^2}$. The mode is $\frac{a-1}{b^2}$ for $a > 1$, but it asymptotes at zero when $a \leq 1$.

Figure 3.8 shows several gamma distributions. All the examples in the left column have a mean of 1, as $a = b$. Similarly, the examples in the right column have a mean of 5. As mentioned, when $a \leq 1$ the gamma shoots up to zero as x decreases. For $a > 1$, gamma distributions have a mode greater than zero and are positively skewed. As a increases, the gamma looks more like a normal distribution.

As with the Beta and Dirichlet distributions, one can interpret the parameters of the gamma as results from a previous hypothetical experiment, in which the sample size was $2a$ and the sum of squares around the mean was $2b$. For people who do lots of analyses of variance, this may be an intuitive way to set a prior for precision. At least thinking about $2a$ as sample size helps. Working backwards from a mean and standard deviation and looking at graphs of representative gamma distributions is probably more helpful. Reasonable mild priors for precision in psychometric applications are $(.5, 1)$ or $(1, 1)$—not much weight, means around values that error variances and population variances have when the latent scale is set so that populations are roughly centered around zero and have variances that don't differ dramatically from 1. for example, the 95 % credibility interval for gamma$(1, 1)$ is $(.025, 3.691)$.

3.5.4 Sources for Priors

The Bayesian paradigm has several advantages over the classical statistical paradigm: Bayesian credibility intervals are easier to interpret than classical confidence intervals, and the paradigm often gives us guidance for computational strategies in cases where the classical paradigm has difficulties. The price we must pay for those advantages is specifying the prior distribution.

There are generally two approaches to specifying the prior information. The *strong prior Bayesian* approach tries very hard to model the state of information before data is observed, often eliciting priors from experts. The *weak prior Bayesian* approach instead tries to minimize the impact of the prior, building up a collection of noninformative priors. Section 3.5.5 describes noninformative priors; the remainder of this section looks at eliciting informative priors.

Consider what we know about the individual in Example 3.14. Because the probability of making a free throw is a probability, it must be between zero and one. However, anyone growing up in a culture where basketball is routinely played has access to better information then that. Children on playgrounds often make free throws, so a probability of 1 in 1 million would obviously be too small, even 1 in 100 seems on the low side. On the other hand, even professional players routinely miss free throws, so 999 times out 1000 seems very high; even 90 % would seem like a remarkable feat. Using this knowledge we would build a prior distribution that is high in the middle probabilities, but tails off near the upper and lower ends, such as Beta$(5, 5)$.

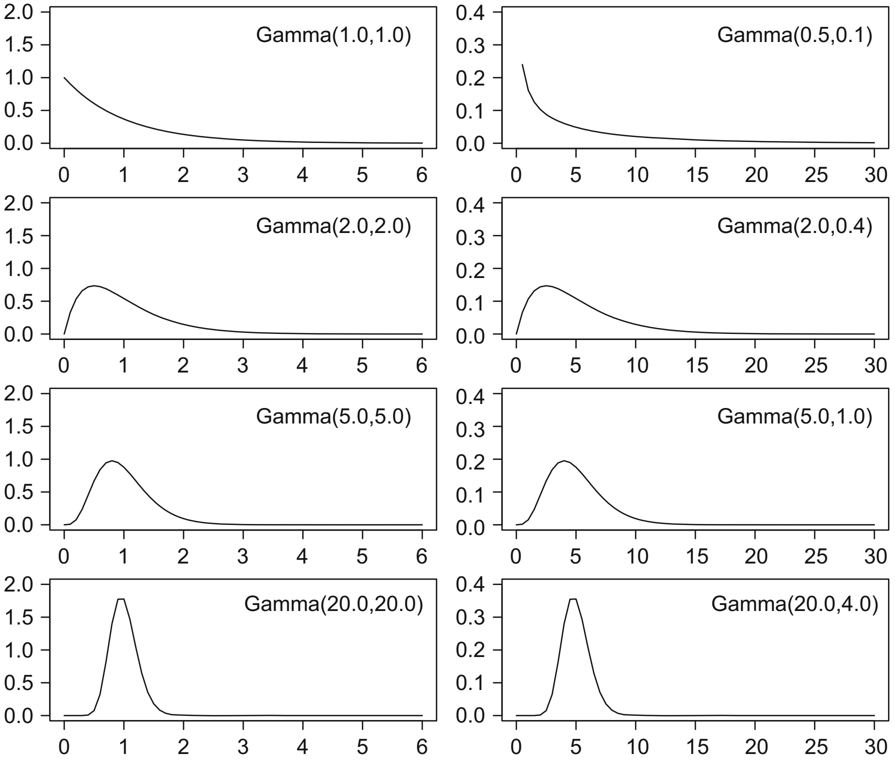


Fig. 3.8 A panel of sample gamma distributions
Reprinted with permission from ETS.

However, if we know the population that the individual comes from, we might be able to say a lot more. For example, if the individual were a female college student who was a member of her school team, we would have access to a large collection of data about the performance of other student athletes in her division on which to base our prior. Suppose the distribution of the previous year's players' percentages across the division was .60 and had a standard deviation of .15. We can work backwards to find the beta distribution that has this mean and standard deviation. Let $\bar{x}$ be the sample mean of a set of proportions and v be its variance. The method of moments estimates of the *beta* parameters are

$$a = \bar{x}\left(\frac{\bar{x}\,(1-\bar{x})}{v} - 1\right) \quad \text{and} \quad b = (1-\bar{x})\left(\frac{\bar{x}\,(1-\bar{x})}{v} - 1\right) .$$

For our college player, this translates to a prior of about Beta$(6, 4)$. If we then observed her make 7 of 10 attempts, our posterior would be Beta$(13, 7)$, which has a mean of .65. and a standard deviation of .10. A teammate who made only 2 of 10 would lead to a posterior of Beta$(8, 12)$, with a mean and standard deviation of .40 and .11, noticeably higher than the observed proportion of .2. If she continued to shoot at this rate, however, and made 20 of 100 the posterior would be Beta$(26, 84)$, with a mean and standard deviation of .24 and .04.

When the prior for an individual case comes from a population distribution, the Bayesian paradigm has an interesting interpretation. The posterior distribution is a mixture of the population information and the information from the data specific to that individual. This is easiest to see in the case of the Normal–Normal conjugate family. In this case, the posterior mean is a weighted average of the population mean and the mean of the data (and the posterior precision is the sum of the prior precision and the precision associated with the data). Thus, the estimate of the mean from the data is "shrunk" towards the population mean. As the amount of data goes up, the amount of shrinkage decreases. Such shrinkage estimators are generally more stable than estimates taken purely from data, especially if the amount of data is low. (Another common example of a shrinkage estimator is a linear regression, where the information we get from the predictor X is "regressed" towards the mean of the dependent variable Y.)

Population distributions are an ideal source for priors when they are available. In educational testing, the reference population is usually clear from the specification of the assessment. However, when substantial prior data on this population or assessment are not available, prior opinion may be the only possible source of information about the unknown values.

The *elicitation* of prior information can be a difficult and time-consuming task. Morgan and Henrion (1990), Walley (1991), Box and Tiao (1973) and Berger (1985) review a number of methods. One problem in this field is that the lay perception of probability is not consistent and is subject to heuristic biases (Kahneman et al. 1982).

Our favorite method is to ask the expert for a mean for the unknown and an "observational equivalence" for the expert's information, or the number of observations which would be equivalent to this expert's opinion. We saw that with Beta and Dirichlet priors (and less intuitively, the gamma) this can be thought of in terms of the results of a hypothetical experiment. There is, however, no evidence that experts are particularly better calibrated on this scale than any other. Furthermore, this approach requires a choice of baseline for zero information (noninformative priors).

3.5.5 Noninformative Priors

To build a model based mainly on data we would like a prior distribution that has a minimal impact on the posterior. The weak prior Bayesian analysis uses *noninformative priors* distributions—priors that, according to some criteria, contain no information about the parameter. Such noninformative priors also play an important role in Bayesian data analysis. Even if the final analysis uses stronger priors, the noninformative prior may be useful for sensitivity analysis or in eliciting expert opinion.

The idea of the equal probability space introduced in Sect. 3.1.2 is the most commonly invoked principle for construction probability distributions. It in fact underlies the Canonical Examples 3.1 and 3.10. The result is a *uniform distribution* over the primitive outcomes.

Applying this principle requires a subjective judgment, namely that each primitive outcome is equally likely. This is not always reasonable in practice. While it may be a good assumption for a simple game of chance, such as tossing a coin or rolling a die, it obviously fails for more complex phenomena, such as whether or not Los Angeles will experience a major earthquake next year. Just because there are two outcomes does not mean, in and of itself, it is reasonable to think they are equally likely.

Applying the principle of equal probability requires a second judgement when assigning a distribution to a continuous random variable: which space we should take to be uniform? Consider the beta-binomial model for the "propensity to succeed" parameter, θ, in Example 3.14. In that example, we used a uniform prior over the space, that is $\theta \sim \text{Beta}(1, 1)$. If instead we take θ to be uniform in the logistic scale, we get $\theta \sim \text{Beta}(1/2, 1/2)$, and taking the distribution to be uniform in the natural exponential family parameter (Jaynes 1968) gives us $\theta \sim \text{Beta}(0, 0)$. The first two priors give a marginal prediction of 1/2 for the probability that the first observation will be a "success" event. The third is not a proper probability distribution because it cannot be normalized. This is a fair amount of information for a "noninformative" prior. Dempster (1968) proposes a system of using upper and lower bounds on the prior, however the resulting distributions are not probabilities but rather belief functions (see Almond 1995).

Jeffreys (1961) argues that the noninformative prior should be invariant under transformations of the variables. Using an information theoretic argument, he winds up with the principle that the prior should be proportional to the reciprocal of the Fisher information. In the beta-binomial case, this yields a $\text{Beta}(1/2, 1/2)$ prior. In the normal–normal case, it yields a uniform distribution over the real line, which is not a proper probability distribution.

The use of improper priors (priors for which the normalization integral is infinite) is a matter of controversy in Bayesian statistics. If we have a fair amount of data, the choice of noninformative prior will not make much difference. For example, if used with a binomial likelihood, the improper Jaynes prior Beta(0,0), for example, amounts to reducing the impact of the observed data by one success and one failure. The posterior does not change much if many successes and failures have been observed. However, the improper prior could get us into trouble if we do not have enough data. The resulting posterior distribution will be improper unless there is at least one success and one failure in the observed data.

One can get many of the benefits of noninformative priors without the problems on noninformative priors by specifying a weak proper prior. Such a prior should be flat or nearly so throughout much of the space of interest, but can rapidly shrink outside the part of the sample space which is thought to be likely. For example, you can take a normal distribution with a "large" variance (where large is taken such that the 95 % interval will have a high probability of covering all meaningful values of the unknown quantity).

Sometimes stronger measures are needed. For example, in an unconstrained IRT model, the person ability variable, θ is usually taken to be normally distributed. However, there is nonidentifiability in this model: You can add a constant to all the abilities and item difficulties, and you can multiply the population standard deviation by another constant and divide all of the item slopes by the same constant, and get an equivalent model. Taking the prior for the ability variable to be $\mathrm{N}(0,1)$ resolves this ambiguity.

3.5.6 Evidence-Centered Design and the Bayesian Paradigm

Evidence-Centered Design is built around the Bayesian paradigm. Although it will work with non-Bayesian measurement models (such as counting up item scores to get the number right), it is at its best when the Proficiency and Evidence Models (Chap. 2) are designed according to the Bayesian paradigm. The proficiency model plays the role of the prior and the evidence model plays the role of the likelihood. The Summary Scoring Process then simply applies Bayes theorem to absorb the evidence coming in from the various task results. Chapter 12 will explore this relationship more formally.

Calibrating the ECD Models (i.e., estimating the conditional probability distributions) requires another application of the Bayesian paradigm. First, we write prior distributions for the parameters of the Proficiency and Evidence models. Then we apply Bayes theorem to update those parameters based on the pretest data. At the same time we can do model checking to refine both our mathematical and cognitive models.

A fundamental principal of ECD is that the mathematical model of the assessment should reflect the cognitive theory of the domain being assessed from a perspective and at a grainsize that suits the purpose of the assessment. As a consequence, applications of ECD can use strong priors based on the cognitive theory. This use of strong priors does not mean that the models are subjective. The ECD process mandates documenting the sources of information and decisions which go into the design of an assessment (Mislevy et al. 2003b) of the knowledge that went into making decisions about both the priors and, more importantly, the likelihoods is disclosed for others to view, critique, and object to. More importantly, as we learn from our data what parts of the model do and do not work well, we can refine the corresponding parts of our cognitive model as well as our mathematical one.

Exercises

3.1. (*Subjective Probability*) Bob, David, Duanli, and Russell are sitting in Russell's office. Russell takes a silver dollar out of his desk drawer and flips it. For each step of this story, write down if Bob's, David's, Duanli's, and Russell's probability that the coin has landed heads side up, is (a) 0, (b) between 0 and 1/2, (c) 1/2, (d) between 1/2 and 1, or (e) 1.

1. Russell has not yet flipped the coin.
2. Russell flips the coin where nobody can see, but does not look at the result.
3. Russell looks at the result, and sees that it is tails. He does not show it to anybody else.
4. Duanli remembers that Russell has a two-headed silver dollar he bought at a magic shop in his desk.
5. Duanli asks Russell about what coins he had in his desk. He replies that he has two normal dollars, a two-headed coin, and a two-tailed coin.
6. Russell shows Bob the result, but does not let anybody else see.
7. Bob announces that the result is tails. Duanli believes Bob always tells the truth, but David remembers that Bob likes to occasionally lie about such things, just to make life a little more interesting.
8. David tells Duanli that Bob sometimes lies.
9. Russell shows everybody the coin.

3.2. (*Sensitivity Analysis*) Example 3.6 (Almond 1995) is mostly based on fairly reliable, if dated, numbers, except for the factor of 5 which is used to inflate the number of reported AIDS cases to the number of HIV-positive individuals. This could charitably be called a wild guess. Perform a sensitivity analysis to this guess by using several different values for this fudge factor (e.g., 1, 5, 10, 25, 50) and calculating the chance that a patient who tests positive on the Western Blot test has HIV. How sensitive are the results to the prior?

3.3. In Example 3.6, the true rate of HIV infection is unknown. Suppose we use a uniform distribution, $\mathrm{P}(HIV_{+}) = .5$, as a "noninformative" prior. Calculate the chance that blood that passes the screening actually contains HIV. Comment on the appropriateness of the uniform distribution as a prior.

3.4. (*Subtest Independence*) Suppose we have a 50-item assessment that follows the Unidimensional IRT model (Fig. 3.3). In particular, assume that all of the item responses, X_i, are conditionally independent given the latent trait, θ. Consider the score on two subtests, $S_1 = \sum_{i=1}^{25} X_i$ and $S_2 = \sum_{i=26}^{50} X_i$, consisting of the first and second halves of the test. Are S_1 and S_2 independent? If not, how could they be made independent? You may use the fact that if $X_1, \ldots, X_n$ are (conditionally) independent of Y, then $\sum_i X_i$ is independent of Y.

3.5. (*Conjunctive Model*) A math "word problem" requires students to read an English-language description of a problem, translate it into a mathematical problem, and solve it. To solve a word problem, a student generally needs both sufficient English reading proficiency, E, and the math skill, M. Let S be the score (right or wrong) for a given word problem, and assume the probability of a correct answer is 85 % for students who have mastered both skills and 15 % for students who lack E, M, or both. (This is a *conjunctive model.*) Assume that in a particular class 90 % of the students have sufficient English proficiency to solve word problems of this type. Of the students that have sufficient English proficiency 75 % of them have M. Of the students that lack E only 50 % have M. Calculate the probability of mastery for the math skill for the following students from this class:

a. A student for which we have not yet observed the performance on the word problem.
b. A student who solves the problem correctly and is known to have sufficient English proficiency.
c. A student who solves the problem incorrectly and is known to have sufficient English proficiency.
d. A student who solves the problem correctly and is known to lack sufficient English proficiency.
e. A student who solves the problem incorrectly and is known to lack sufficient English proficiency.
f. A student who solves the problem correctly and whose English proficiency is unknown.
g. A student who solves the problem incorrectly and whose English proficiency is unknown.

What is the effect of a student's lack of English proficiency on our ability to measure her math skill?

3.6. (*Competing Explanation*) Presume the same situation described in Problem 3.5, except with E and M marginally independent, and $P(E) = P(\overline{E}) = .5$ and $P(M) = P(\overline{M}) = .5$. Show that E and M are not independent conditional on $\overline{S}$.

3.7. Suppose that we are trying to determine the ability of several students to solve a particular kind of problem. Call the probability the student will get the answer right on any particular problem θ. Use the Jeffreys prior (Beta$(1/2, 1/2)$), and calculate the posterior mean and variance for the following students:

a. A student who got 7 items right on a 10-item test.
b. A student who got 9 items right on a 10-item test.
c. A student who got 15 items right on a 20-item test.
d. A student who got 18 items right on a 20-item test.
e. A student who got 30 items right on a 40-item test.

f. A student who got 36 items right on a 40-item test.

Repeat this exercise with the uniform prior (Beta$(1,1)$). How sensitive are the conclusions to the choice of prior?

3.8 (*True Score Test Theory*). Suppose that a student's score on a test $X = T + E$, where T is the students *true score* (the score the student would have obtained if the student did not make any mistakes) and E is the error. Suppose that for a particular assessment instrument, the error is known to be $\mathrm{N}(0, 5^2)$. Assume that the distribution of T for the student's true score is known to be $\mathrm{N}(70, 10^2)$. Calculate the mean and variance of the posterior for the following students:

a. A student who got a score of 75.
b. A student who got a score of 90.
c. A student who got a score of 50.

What happens to those posteriors if the population variance gets larger? smaller?

3.9 (*Test Length*). Suppose that an assessment is assembled from a collection of short tests. Let the score on Test i be $X_i = T + E_i$, where T is the true score and the error $E_i \sim \mathrm{N}(0, \sigma^2)$; that is each short test has the same measurement-error variance. Assume that the population distribution for the true score is $\mathrm{N}(\mu, \tau^2)$. Let $X = \sum_{k=1}^{K} X_k$ be a student's score on an assessment consisting of K tests. Calculate the posterior mean and variance for the true score for that student. What happens to these values as K increases?

4

Basic Graph Theory and Graphical Models

One of the underlying principles in our approach to assessment design is that the psychometric model should reflect the cognitive model, at a grain size and in a manner that suits the job at hand (Mislevy 1994). This answers the fundamental question from the previous chapter, "where do we get the knowledge to construct prior distributions?" It comes from the experts in the domain being modeled. However, experts in cognition, learning, and substance of an assessment area will rarely be comfortable with mathematical notation for expressing their ideas. To work with them, the psychometrician needs a representation which is rigorous, but intuitive enough for the substantive experts to be comfortable.

Enter the graphical model depicting variables with nodes in a graph and patterns of dependency with edges drawn between them; the graphical model is a representation of the joint distribution over all of the variables. However, because this representation is graphical, domain experts can provide feedback or even help construct the corresponding model. Bayesian networks share the idea of using graphs to communicate with experts with other statistical techniques, in particular, structural equation models. The difference is that the graphical representation that structural equation models use is tuned to building systems of simultaneous equations that represent functional relations among variables, while the graphs used with Bayes nets are tuned to expressing conditional independence relationships, within a representation of their joint distribution.

This difference is subtle but has important implications. First, the conditional independence conditions lead directly to efficient computational algorithms, in particular those discussed in Chaps. 5 and 9. Second, the factorization properties of the graph lead to strategies for eliciting consistent probability distributions. These advantages have led to a rapid rise of graphical model techniques in the artificial intelligence community. In particular, Bayesian networks—graphical models in which all of the variables are discrete—are very popular. This book arises out of our work in applying those techniques to problems in educational testing.

R. G. Almond et al., *Bayesian Networks in Educational Assessment*,
Statistics for Social and Behavioral Sciences, DOI 10.1007/978-1-4939-2125-6_4

This chapter provides a basic foundation of graph theory and graphical models to support the application to educational testing that is developed in the rest of the book. Section 4.1 provides a brief introduction to graph theory, providing definitions of all of the necessary terms. The focus is on ideas rather than technical details. The chapter provides enough background for the reader to work with experts to build directed graphs to express the substantively important relationships, and to understand the key ideas behind how this representation is transformed to a representation that supports efficient computational methods. Section 4.2 explores the relationship between the graph and factorization and Sect. 4.3 explores the relationship between separation in the graph and conditional independence. As Bayesian networks are frequently built using causal arguments, Sect. 4.4 reviews the relationship between graphical and causal modeling. Finally, Sect. 4.5 contrasts Bayesian networks to a number of other techniques which use similar graphs.

4.1 Basic Graph Theory

The key feature of *graphical models* (including Bayesian networks) is that they represent probability distributions that factor according to a graph. That is, the joint probability distribution can be expressed as the product of factors (such as conditional probability distributions) that each involve only subsets of the variables, and those subsets correspond to the topology of the graph. This graph thus provides a picture representing key aspects of the structure of the distribution. The previous chapters have already used these pictures informally. Defining graphical models more formally requires some terms from graph theory.

A *graph* is a pair $\mathcal{G} = \langle \mathcal{N}, \mathcal{E} \rangle$ where $\mathcal{N}$ is a set of objects called *nodes* or *vertices* and $\mathcal{E}$ is a set of *edges* with each edge being a nonempty set of nodes. Usually, edges contain exactly two nodes, in which case the resulting graph is a *simple graph* (Sect. 4.2 describes *hypergraphs* which allow more than two nodes in an edge). In this book, we draw nodes with a label (a letter, word, or abbreviation) in a circle or a round box. Simple edges are drawn as a line connecting two nodes.

For graphical models in statistics, the nodes represent variables in a probability model. Each *variable*, A_i, is associated with an outcome space, or set of possible values. For most of the models in this book the outcome space will be a finite set $\{a_{i,1}, \ldots, a_{i,n}\}$, although for more general graphical models, the outcome space can be an dense set such as the real line. For educational applications, it helps to add an icon to the node to indicate the type of variable represented. We will use node labels with circles to indicate proficiency model variables and triangles to indicate evidence model variables.

4.1.1 Simple Undirected Graphs

In a undirected graph all of the edges are unordered pairs (In this book, the word *graph* will be used to refer to simple, undirected graphs unless another meaning is clear from the context). Fig. 4.1 shows an example.

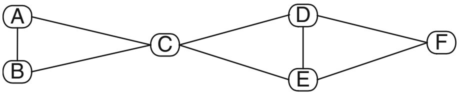


Fig. 4.1 A simple undirected graph
Reprinted from Almond et al. (2007) with permission from ETS.

For a graph $\mathcal{G} = \langle \mathcal{A}, \mathcal{E} \rangle$, two nodes, A_1 and A_2 are *neighbors* if there exists an edge $\{A_1, A_2\}$ (curly braces are used to represent unordered sets, so this is equivalent to $\{A_2, A_1\}$) in $\mathcal{E}$. That is, there is a line between A_1 and A_2 in the representation. The set of all neighbors of a node A_i is called the *neighborhood* of A_i in $\mathcal{G}$. In Fig. 4.1, A and C are neighbors, but C and F are not. The neighborhood of C is $\{A, B, D, E\}$.

Let $\mathbf{C}$ be a set of nodes such that for all A_i, A_j in $\mathbf{C}$, A_i and A_j are neighbors, i.e., there is a line between every two nodes in the set. Such as set is called *complete*. $\{C, D, E, F\}$ in Fig. 4.1 is not complete, because it lacks a $\{C, F\}$ edge. $\{A, B\}$ is a complete set, but we notice that we could include C and get a larger complete set. A maximal complete set is called a *clique*. The cliques of Fig. 4.1 are $\{A, B, C\}$, $\{C, D, E\}$, and $\{D, E, F\}$. Cliques will be important when we get to calculation, because they are subsets of variables that we need to work with simultaneously.

There are two ways that one graph can be smaller than another: it can have a smaller set of edges or it can have a smaller set of nodes. Let $\mathcal{G}_1 = \langle \mathcal{A}_1, \mathcal{E}_1 \rangle$ and $\mathcal{G}_2 = \langle \mathcal{A}_2, \mathcal{E}_2 \rangle$ be two graphs. If they have the same nodes but $\mathcal{G}_1$ lacks some of the edges in $\mathcal{G}_2$ (i.e., $\mathcal{A}_1 = \mathcal{A}_2$ and $\mathcal{E}_1 \subset \mathcal{E}_2$) then $\mathcal{G}_1$ is a *partial graph* of $\mathcal{G}_2$. If $\mathcal{G}_1$ lacks some of the nodes and possibly some edges of $\mathcal{G}_2$ and does not have any additional edges (i.e., $\mathcal{A}_1 \subset \mathcal{A}_2$ and $\mathcal{E}_1 \subseteq \mathcal{E}_2$), then $\mathcal{G}_1$ is a *subgraph* $\mathcal{G}_2$.

4.1.2 Directed Graphs

A *directed graph* (sometimes called a *digraph*) extends the idea of a undirected graph in that its edges are ordered pairs. By convention the edges are drawn as arrows. Figure 4.2 shows a typical directed graph. The arrow from D to F, as an example, represents the directed edge (D, F) (parenthesis are used to indicate ordered sets, so this is not equivalent to (F, D)).

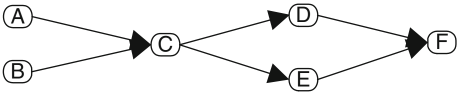


Fig. 4.2 A directed graph
Reprinted from Almond et al. (2007) with permission from ETS.

Let $\mathcal{G} = \langle \mathcal{A}, \mathcal{E} \rangle$ be a directed graph. Let A be a node in $\mathcal{G}$. All of the nodes that have an arrow from them to A are called *parents* of A. More formally, they are defined as $\{A^* : (A^*, A) \in \mathcal{E}\}$, and they are denoted $\mathrm{pa}(A|\mathcal{G})$ or more simply $\mathrm{pa}(A)$ when the context is clear. Similarly, the nodes that A has an edge going to are the *children* of A, or $\{A_* : (A, A_*) \in \mathcal{E}\}$. In a directed graph, two nodes are neighbors if one is a parent of the other. In Fig. 4.2, C is a parent of D. This language is often extended in the obvious way. For example, A is an *ancestor* of D and F is a *descendant* of C. This terminology comes from early applications in animal breeding, where nodes represent characteristics of animals that are literally parents and children, ancestors and descendants.

4.1.3 Paths and Cycles

Let $A_0, A_1, \ldots, A_n$ be a series of nodes such that A_i and A_{i+1} are neighbors. Such a series is called a *path* of *length* n. A path is *simple* if no node is repeated. Two nodes in a graph are *connected* if there exists a path between them. A graph is *connected* if all its nodes are connected.

A path whose first and last node are the same is a *cycle*. In the undirected graph shown in Fig. 4.1, (C, D, F, E, C) is a cycle.

For directed graphs, what was defined as a path is called a *chain*, but there is an additional condition: a path on a directed graph requires that for each i, the ordered pair (A_i, A_{i+1}) is an edge. That is, all the directed edges must point in the direction of travel. In the directed graph shown in Fig. 4.2, (C, D, F, E, C) is not a directed cycle because the directions do not allow a trip from C back to C again. We sometimes call (C, D, F, E, C) an undirected cycle, meaning that it would be a cycle if direction were ignored.

An undirected connected graph that contains no cycles is said to be *acyclic*, and is called a *tree*. A node of a tree that is a member of only one edge is a *leaf*. Figure 4.3 is a tree (in fact, it is a spanning tree of Fig. 4.1, which means it is a subgraph that is a tree). In this graph nodes A, B, and F are leaves. The idea of a tree will also be important when we consider updating beliefs. This is because in a tree, moving from an initial node to each of its neighbors, then to each of their neighbors in turn which have not yet been visited, we are ensured that each node will be visited exactly once.

Acyclic directed graphs—directed graphs containing no directed cycles—play a special role in the construction of models. Figure 4.4b is an acyclic directed graph (note that these graphs may contain undirected cycles).

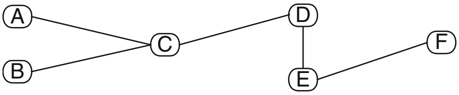


Fig. 4.3 A tree contains no cycles
Reprinted from Almond et al. (2007) with permission from ETS.

Fig. 4.4a is cyclic. Acyclic directed graphs are often called by the euphonious misnomer *DAG*. Technically speaking a directed acyclic graph is an acyclic graph, a tree, whose edges are directed. However, most authors who use the abbreviation DAG are talking about acyclic directed graphs, and we will do so too.

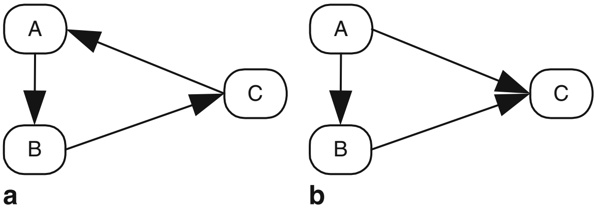


Fig. 4.4 Examples of cyclic and acyclic directed graphs. **a** Cyclic, **b** acyclic
Reprinted from Almond et al. (2007) with permission from ETS.

Acyclic directed graphs play a key role in the theory of Bayesian networks. As the direction of the edges represents the direction of statistical conditioning, the acyclic condition prevents the modeler from specifying the distribution using circular dependencies, ensuring that the distribution is well defined from the graph (In contrast, the graphing conventions used with structural equations models allow directed cycles, for example to convey reciprocal causation).

Let $A_0, A_1, \ldots, A_n, A_0$ be a undirected cycle. A pair of nonadjacent nodes that are contained in a common edge are called a *chord* of the cycle. If a cycle contains no chords it is called *chordless*. Recall that (C, D, F, E, C) is a cycle in Fig. 4.1; $\{D, E\}$ is a chord in this cycle. An undirected graph that has no simple chordless cycles of length greater than three is called *triangulated*. If a graph is not triangulated, additional edges can be *filled in* until it is triangulated.

Figure 4.5 shows an untriangulated graph, and one fill-in that will make it triangulated. We will see in Chap. 5 how triangulation is used in the computation algorithms to avoid double counting evidence. Suppose that in Fig. 4.5 we are propagating evidence from D to A. There are two paths by which the

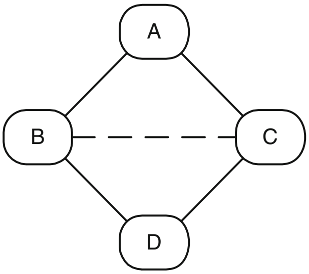


Fig. 4.5 Filling-in edges for triangulation. Without the dotted edge, this graph is not triangulated. Adding the dotted edge makes the graph triangulated
Reprinted from Almond et al. (2007) with permission from ETS.

evidence flows, one through B and one through C. The triangulation reminds us that the two evidence flows are not independent (they both come from D) and we will have to take the joint effect into account.

4.2 Factorization of the Joint Distribution

Armed with our knowledge of graph theory, we can now define a graphical model. A graphical model combines graphs and probability in such a way that features of graphs help us better understand and work effectively with probability models.

Recall that an integer can be written as the product of smaller integers, such as $360 = 5 \times 3^2 \times 2^3$. Basically, a graphical model is a probability distribution that can be factored into the product of pieces involving smaller sets of variables, according to structure of the graph. However, the nature of the pieces and the exact rules varies with the type of graph. Section 4.2.1 describes models using directed graphs and Sect. 4.2.3 describes models using undirected graphs. Section 4.2.2 describes the factorization hypergraph which links the two representations. As described in Sect. 4.3, the different types of graphs also have different rules for reading conditional independence constraints, and it is often useful to work back and forth between the directed and undirected representations. In particular, directed graphs are better for working with experts, building models around substantive knowledge, and eliciting initial estimates of probabilities, while undirected graphs support key computational efficiencies.

4.2.1 Directed Graph Representation

We saw in Chap. 3 that a probability distribution can be written in a recursive representation, and that terms simplify when conditional independences let some variables drop out of the conditioning lists. This section shows how this phenomenon can be expressed in terms of directed graphs. Consider a probability distribution over six variables, A, B, C, D, E, and F, that can be factored as follows:

$$\mathrm{P}(A,B,C,D,E,F) = \mathrm{P}(A)\mathrm{P}(B)\mathrm{P}(C|A,B)\mathrm{P}(D|C)\mathrm{P}(E|C)\mathrm{P}(F|E,D) \ .$$

To draw the directed graph that corresponds to this distribution, start with a set of nodes, one for each variable. For every conditional distribution, draw a directed edge from each conditioning variable to the consequence variable in the distribution; for example, for $\mathrm{P}(C|A,B)$, draw edges from A to C and from B to C. The result is Fig. 4.6. This is the basic idea of the directed graphical model: the distribution for each variable is defined conditionally on its parents in the graph. Note that this representation does not say anything about the nature or the functional form of the dependence—just that the dependence exists. This correspondence between direct dependence and conditional probabilities is the starting point for all that follows. Here is the formal definition.

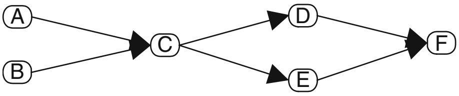


Fig. 4.6 Directed Graph for $\mathrm{P}(A)\mathrm{P}(B)\mathrm{P}(C|A,B)\mathrm{P}(D|C)\mathrm{P}(E|C)\mathrm{P}(F|E,D)$ Reprinted from Almond et al. (2007) with permission from ETS.

Definition. **Directed Graphical Model.** *Let $\mathcal{A}$ be a set of variables that describe some problem space. Let $\mathcal{G} = \langle \mathcal{A}, \mathcal{E} \rangle$ an acyclic directed graph whose nodes correspond to the variables in $\mathcal{A}$. The probability function* $\mathrm{P}_{\mathcal{G}}$ *is called the total probability and is defined by:*

$$\mathrm{P}_{\mathcal{G}} = \prod_{A \in \mathcal{A}} \mathrm{P}(A|\mathrm{pa}(A)) \ . \tag{4.1}$$

If $\mathrm{pa}(A)$ *is empty, then the conditional probability is taken as an unconditional (marginal) probability.*

The key idea is to use the law of total probability to break a big joint probability distribution up into many small factors. Although a joint probability distribution can always be factored according to Eq. 4.1 (using a complete graph—one where there is an edge between every possible pair of nodes—as the base), more often than not the modeler gets a break from conditional independence conditions. In fact, Pearl (1988) claims it is a characteristic of human reasoning to organize our knowledge in terms of conditional independences, and where they do not exist, invent variables to create them—syndromes in medicine, for example, or in our case, latent variables in educational measurement. At any rate, if the edges are sparse in a graph, then the factors in Eq. 4.1 will be small. This condition can be exploited to produce efficient algorithms both for eliciting probabilities and carrying out computations.

4.2.2 Factorization Hypergraphs

It would be nice if the edges were in one-to-one correspondence with the factors of the joint distribution, but this only happens in special cases such as chains. Graphs with more than one edge per distribution, as in the preceding example, are the rule rather than the exception. It happens whenever a variable has more than one parent. We can extend our graphical tool kit to express these relationships with *hypergraphs*. If graph edges are allowed to be arbitrary nonempty sets of nodes, not just pairs, then the resulting graph is a hypergraph and the edges of the hypergraph are called *hyperedges*.

Using hypergraphs we can represent distributions with one hyperedge for each factor in Eq. 4.1. This is a key step for moving from a directed graph, which is easiest to build working with experts, to an undirected graph that supports efficient calculation algorithms. To see the steps by which we move from a directed graph that represents a joint distribution to an undirected graph that supports computation on that distribution, we will need to define hypergraphs, directed hypergraphs, and 2-sections.

A hypergraph is drawn with the nodes represented by points, numbers, or letters and the edges represented by closed curves enclosing the elements of the edges. Figure 4.7a shows an undirected hypergraph. $\{F\}$, $\{F, W\}$, $\{A, D, S\}$, and $\{D, L, F, M\}$ are some of its hyperedges.

For a hypergraph $\mathcal{G} = \langle \mathcal{A}, \mathcal{E} \rangle$, two nodes, A_1, A_2 are *neighbors* if there exists a hyperedge that contains them both. F and W are neighbors in Fig. 4.7a, and the neighbors of S are A, D, W, and R.

If $\mathcal{H}$ is a hypergraph, then there exists a simple (undirected) graph $\mathcal{G}$ with the same set of nodes such that every node A has the same neighbors in $\mathcal{G}$ as it has in $\mathcal{H}$. This graph is called the *2-section* of $\mathcal{H}$. We can construct a hypergraph's 2-section by starting with its set of nodes, and drawing a simple edge between every pair of nodes that are neighbors in the hypergraph. Figure 4.7b is the 2-section of Fig. 4.7a. A given hypergraph has a unique 2-section, but many hypergraphs can have the same 2-section. The idea of a 2-section is a step in moving from a directed graph representation of a joint probability distribution to a computing representation based on a simple graph.

A *directed hypergraph* is made by partitioning each hyperedge into two parts: a set of *parents* and a set of *children*. These are *directed hyperedges*. For the most part, we will restrict ourselves to hypergraphs with one child node per edge, and associate the directed hyperedges with marginal and conditional probability distributions. This gives what is called the *factorization hypergraph* of the directed graphical model—a representation that connects

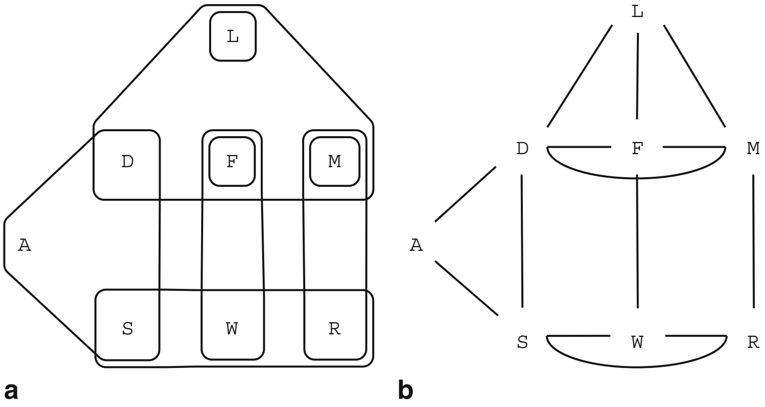


Fig. 4.7 Example of a hypergraph **(a)** and its 2-section **(b)**
Reprinted from Almond (1995) with permission from Taylor & Francis.

the factors of the probability distribution with the features of the graph. We will draw directed hyperedges as rectangles, and later we will annotate them with icons that signify the type of distribution they represent. *Tentacles* link the hyperedge icons to the nodes; they look like arrows. Figure 4.8 shows an example of a directed hypergraph.

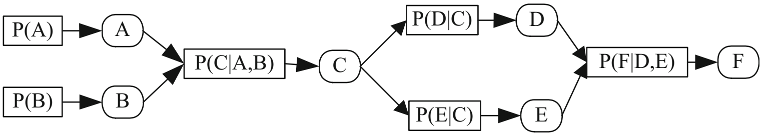


Fig. 4.8 Hypergraph representing $\mathrm{P}(A)\mathrm{P}(B)\mathrm{P}(C|A,B)\mathrm{P}(D|C)\mathrm{P}(E|C)\mathrm{P}(F|E,D)$
Reprinted from Almond et al. (2007) with permission from ETS.

In this representation, variables are nodes in the graph, and hyperedges are distributions. We see a directed hyperedge for each node, which is labeled by the conditional or marginal probability distribution that is associated with it. That is, for each variable X in Fig. 4.8 there is a directed hyperedge $(X, pa(X))$. For example, $\mathrm{P}(F|E,D)$ is represented in the figure by the box with that label and the tentacles from the parents D and E, through the box, to the child F. Recalling that a single node can be a hyperedge in a hypergraph, we also see a directed hyperedge associated with A, which has no parents; accordingly, the box representing this hyperedge is labeled with the marginal distribution $\mathrm{P}(A)$.

Shenoy and Shafer (1990) call this representation a *valuation based system*, where "valuation" refers to the probability distributions and conditional distributions that add a layer of quantitative information to the structural relationships depicted in the hypergraph. The term valuation is actually broader than just probabilities, and a number of kinds of relationships between variables can be modeled with valuation based systems (Sect. 4.5 lists a few).

As with the undirected hypergraph, we make an undirected 2-section by connecting all nodes that are in the same hyperedge, for all hyperedges. Figure 4.9 shows the undirected 2-section of Fig. 4.8.

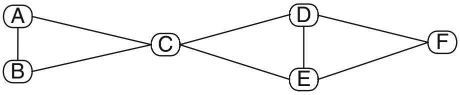


Fig. 4.9 2-section of Fig. 4.8
Reprinted from Almond et al. (2007) with permission from ETS.

Note what happens as we go from a directed graphical model, Fig. 4.6, through its factorization hypergraph, Fig. 4.8, to its 2-section, Fig. 4.9. Nodes that are the parents of common children are joined in the undirected version. This process is called *moralization* and the resulting graph is the *moral graph* because the parents are "married." This is the principle way to go from the directed to the undirected graphical representation of a probability distribution.

4.2.3 Undirected Graphical Representation

In an *undirected* graphical model, the factors of the joint probability distribution are associated with the cliques of the graph. The graph in Fig. 4.9 has three cliques: $\{A, B, C\}$, $\{C, D, E\}$, and $\{D, E, F\}$. The factor associated with each clique is the product of the component distributions defined over the variables in the clique, which we read from Fig. 4.8. The rule is to include the conditional probability distribution for each variable in the clique that has parents that are also in the clique, and the marginal probability distribution for each variable in the clique that has no parents. Thus, the factor associated with the clique $\{A, B, C\}$ is the joint probability obtained as $\mathrm{P}(C|A, B)\mathrm{P}(A)\mathrm{P}(B)$. The factor associated with $\{D, E, F\}$ is the conditional probability $\mathrm{P}(F|D, E)$, and the factor associated with $\{C, D, E\}$ is the product of conditional probabilities $\mathrm{P}(D|C)\mathrm{P}(E|C)$.

As this example shows, the factors associated with the cliques can be either a probability distribution, a collection of conditional probability distributions, or complex combinations of probabilities and conditional probabilities. We call such objects *potentials*. They are what we will use in Chap. 5 for updating probability distributions when values are obtained from a subset of variables. Although potentials may or may not represent probability distributions, they can usually be normalized and interpreted as probabilities. The collection of variables over which a potential is defined is called its *frame of discernment* (This term is adapted from Dempster-Shafer theory where it originally was used to mean outcome space. See Almond 1995).

Whether the graph is directed or undirected, the key idea is the same. The joint probability distribution is broken up into a collection of factors, $\mathcal{C}$.

- Directed Graphs—Sets $\mathcal{C}$ correspond to each node A and its parents.
- Undirected Graphs—Sets $\mathcal{C}$ correspond to cliques in the graph.

Factorization, and the corresponding conditional independence conditions, can be exploited when calculating probabilities for subsets of variables. Chapter 5 explores some of the methods for making routine Bayesian updating calculations much more efficient by using these ideas.

4.3 Separation and Conditional Independence

As mentioned previously, one important feature of the graphical model is that separation of the variables in the graph implies conditional independence of the corresponding variables. To formalize this intuition, we need to formally define separation. Section 4.3.1 provides a definition for separation in both undirected and directed graphs. Section 4.3.2 explores the relationship between separation and independence. Finally, Sect. 4.3.3 describes the important Gibbs–Markov equivalence theorem which states that factorization implies independence and *vise versa*.

4.3.1 Separation and D-Separation

Directed and undirected graphs encode conditional independence conditions differently. Consequently, the definition of separation is different in the two different types of graph. For the undirected graph, separation corresponds nicely to the intuitive notion of the concept. Fortunately, separation in undirected graphs is not only easier to understand, it is the one that matters in computing algorithms.

Definition. **Separation.** *Let* $\mathbf{X}$*,* $\mathbf{Y}$*, and* $\mathbf{Z}$ *be sets of nodes in an undirected graph,* $\mathcal{G}$*.* $\mathbf{Z}$ separates $\mathbf{X}$ *and* $\mathbf{Y}$*, if for every* A_x *in* $\mathbf{X}$ *and for every* A_y *in* $\mathbf{Y}$*, all paths from* A_x *to* A_y *in* $\mathcal{G}$ *contain at least one node of* $\mathbf{Z}$*.*

In Fig. 4.9, C separates $\{A, B\}$ from $\{D, E\}$. Taken together, $\{D, E\}$ separates C from F.

An equivalent way to think about separation is that deletion of the nodes $\mathbf{Z}$ from the graph disconnects the nodes of $\mathbf{X}$ from the nodes of $\mathbf{Y}$. The intuition is that if we remove the nodes $\mathbf{Z}$ by conditioning on the corresponding variables, this renders the variables in $\mathbf{X}$ and $\mathbf{Y}$ independent.

For directed graphs, the "competing explanation" phenomenon complicates the notion of independence. Recall from Example 3.9 that sometimes making an observation can render two previously independent variables dependent. These competing explanation cases will have a graphical structure that looks like $X \rightarrow Z \leftarrow Y$, with *converging arrows* both pointing at Z (these are sometimes called *colliders*). As there is no directed path from X to Y, they are separated. However, when Z is observed there is still a dependence. For that reason, reading independence conditions from a directed graph requires definition of d-separation.

Definition. **d-Separation.** *(Pearl 1988) Let* $\mathbf{X}$*,* $\mathbf{Y}$*, and* $\mathbf{Z}$ *be sets of nodes in a acyclic directed graph,* $\mathcal{G}$*.* $\mathbf{Z}$ d-separates $\mathbf{X}$ *and* $\mathbf{Y}$*, if for every* A_x *in* $\mathbf{X}$ *and for every* A_y *in* $\mathbf{Y}$*, there is no chain from* A_x *to* A_y *in* $\mathcal{G}$ *along which the following conditions hold: (1) every node with converging arrows is in* $\mathbf{Z}$ *or has a descendant in* $\mathbf{Z}$ *and (2) every other node is outside* $\mathbf{Z}$*.*

This somewhat obscure definition captures the fact that observing the value of a common descendant can make ancestors dependent. Simply looking at where there are edges is no longer sufficient, as it was with separation in undirected graphs, because the same pattern of edges can lead to d-separation in some cases and not in others, depending on their directions.

The intuition is as follows: A_x and A_y are d-connected if information can "flow" from A_x to A_y (or the other way). Assume we "know" the values for the variables corresponding to nodes in $\mathbf{Z}$; in some cases this blocks the flow and in other cases it opens the flow. (1) Knowing intermediate steps along a path from A_x to A_y (or A_y to A_x) blocks the flow of information along that path. (2) Knowing common (direct) ancestors blocks the flow of information from A_x and A_y through that ancestor (see Example 3.8). (3) Knowing common descendants unblocks the flow of information from A_x to A_y through the common descendant (see Example 3.9), although if the common descendant is not known, then that path is still blocked. Figure 4.10 shows some examples.

The upshot of d-separation for our purposes is this: It is easiest to construct directed graphs that reflect the local relationships that are cognitively and substantively important. The competing explanations phenomenon, however, can introduce some relationships that are not apparent in this representation, and have important implications for updating beliefs from observations. Induced dependencies are important in two ways: conceptually, for sorting out evidence in correct but subtle ways, and computationally, for making sure the updating to all other variables is coherent. Because under some circumstances knowledge about D renders B and C dependent, then, B and C must be connected in the undirected graph to represent the same pattern of dependencies.

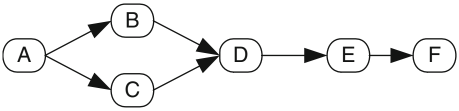


Fig. 4.10 D-Separation (Pearl, 1988). Here, $\{E\}$ d-separates D and F (intermediate step in chain). $\{A\}$ d-separates B and C (common ancestor), but $\{D\}$ does not d-separates B and C (common descendant). Furthermore, $\{A, F\}$ does not d-separate B and C even though $\{A\}$ does by itself (common descendants must not be included in the separator set)

Reprinted from Almond et al. (2007) with permission from ETS.

This is why, to go from the directed to undirected representations, B and C must be married, producing the moral graph (A formal way of saying this is that we need to work with the 2-section of the factorization hypergraph).

4.3.2 Reading Dependence and Independence from Graphs

Ideally, the separation properties of the graph should show all of the conditional independence relationships in the probability model. This is seldom possible. The terms *I-Map* and *D-Map* (Pearl 1988) categorize the relationship between a model and a graph. The formal definitions are stated in terms of separation in the graph and independence in the model:

Definition. **D-Map, I-Map.** *Let $\mathcal{M}$ be a probability model and let $\mathcal{G}$ be a graph with a one-to-one correspondence between the nodes of $\mathcal{G}$ and the variables of $\mathcal{M}$. $\mathcal{G}$ is a* dependency map (or D-map) *of $\mathcal{M}$ if for all disjoint subsets $\mathbf{X}$, $\mathbf{Y}$, and $\mathbf{Z}$ of the variables such that $\mathbf{X}$ is independent of $\mathbf{Y}$ given $\mathbf{Z}$ in $\mathcal{M}$, $\mathbf{Z}$ separates $\mathbf{X}$ and $\mathbf{Y}$ in $\mathcal{G}$. Similarly, $\mathcal{G}$ is an* independence map (or I-map) *of $\mathcal{M}$ if $\mathbf{Z}$ separates $\mathbf{X}$ and $\mathbf{Y}$ in $\mathcal{G}$ implies $\mathbf{X}$ is independent of $\mathbf{Y}$ given $\mathbf{Z}$ in $\mathcal{M}$.*

What this means is that in a D-map, wherever there is an edge there is a dependence relationship between those variables in the probability model. However, there may be dependencies that are not shown. This is the issue we discussed in the previous section, where observing a descendant can induce a dependency among ancestors that was not depicted in the original directed graph. If we are thorough, the directed graphs we create with experts to depict cognitive and substantive relationships will usually be D-maps.

An I-map, in contrast, can miss independence conditions. Everywhere there can a dependency between variables in the model that cannot be removed by conditioning on other variables, there will be an edge in the graph—but there may be edges in the graph where there are not in fact dependencies. The 2-section of a factorization hypergraph, like Fig. 4.9, is an

I-map. The additional edges ensure that any induced dependencies will be taken into account, but they might not have been needed. A key result is that if the graph $\mathcal{G}$ is an I-map of the probability model $\mathcal{M}$, then $\mathcal{M}$ is *Markov* with respect to $\mathcal{G}$, i.e., variables that are separated in the graph are conditionally independent, given the variables that separate them.

If $\mathcal{G}$ is both a D-map and and I-map, it is a *perfect map.* Perfect maps are unfortunately rare (again, chains provide examples). However, I-maps and D-maps always exist. For example, the complete graph (all components connected) is a trivial I-map, and the disconnected graph (no edges) is a trivial D-map. Minimal I-maps (maximal D-maps) capture as many of the independence (dependence) conditions as possible (see Pearl 1988).

In sum, directed graphs are good for making maximal D-maps, and undirected graphs are good for making minimal I-maps. One reason to transform the directed graph to the undirected graph is that it is easier to read the conditional independence relationships from the undirected graph. However, the added moral edges represent dependencies which are only realized in certain circumstances. For the purpose of eliciting distributions we prefer D-maps, and hence prefer to work with the directed graphs. The computational algorithms of Chap. 5 require I-maps, and hence we transform the graph to the undirected representation for calculation.

4.3.3 Gibbs–Markov Equivalence Theorem

Which comes first, the factorization or the conditional independence? We do not need to think hard about this question because the two are equivalent under fairly mild restrictions. This problem was addressed early in the world of statistical physics under the name *Gibbs–Markov equivalence.* The term *Gibbs* refers to the ability to factor the distribution into Gibbs potentials, whose frames are given by the cliques of the graph. As noted above, the term *Markov* means that variables which are separated in the graph are conditionally independent. Moussouris (1974) provides sufficient conditions for Gibbs–Markov equivalence in probabilistic models.

Theorem 4.1 (Gibbs–Markov Equivalence). *Let $\mathcal{G}$ be a graphical model for which the probability is strictly positive (there is no combination of input variables which has zero probability). The model graph is an I-map of its graphical model, or equivalently, a graphical model is Markov with respect to its model graph. (For a proof, see Moussouris 1974)*

In practice we usually use the Gibbs → Markov direction, i.e., given the factorization we derive the conditional independence properties. Logical relationships[1] have zero probabilities and cause the Markov → Gibbs direction to break down. However, such cases usually do not cause a problem because we start with the factorization and go to the conditional independence statements.

Practical model building usually goes back and forth between the two representations. If we have a factorization, we ask experts about conditional independence implications to verify the factorization. If we have independence conditions, we draw an a appropriate factorization and try to elicit factors. Eventually, we exploit the Markov conditions to derive efficient computational techniques (Chap. 5).

4.4 Edge Directions and Causality

The directed graphical representation is often easier to use than the undirected representation for building models. In the undirected graph, one must ensure that all of the factors (associated with the cliques) make up a consistent probability distribution. For the directed graph, if the graph is acyclic and each factor is a consistent probability or conditional probability, then the resulting distribution will be a proper joint probability.

When building such a graph the question, "what do the arrows mean," inevitably arises. The direction of an edge means the "direction of statistical conditioning"—as a formal property, simply what is on the right and left of the conditioning bar. Things become more interesting when we link graphs and pictures to real-world relationships.

Edges can be drawn in either the "causal" (Fig. 4.11) or "diagnostic" (Fig. 4.12) direction. Both are representations of the same joint distribution, so formally they are equivalent.

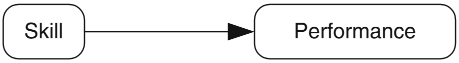


Fig. 4.11 Directed graph running in the "causal" direction: P(Skill) P(Performance|Skill)

Reprinted from Almond et al. (2007) with permission from ETS.

We can use Bayes theorem to translate between the two representations. This is called *arc reversal* in influence diagram literature (Howard and Matheson 1984). However, arc reversal sometimes results in a more complex graph. For example, if we were to reverse the edge between *Language Proficiency*

[1] For example, let A, B, and C be variables that take the values `true` and `false`. Defining $\mathrm{P}(C|A, B)$ to be `true` with probability 1 if both A and B are `true` and `false` with probability 1 otherwise is a logical, rather than stochastic relationship.

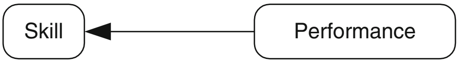


Fig. 4.12 Directed graph running in the "diagnostic" direction: P(Performance) P(Skill|Performance)
Reprinted from Almond et al. (2007) with permission from ETS.

and *Reading* in Fig. 4.13, then we would wind up drawing an edge between *Reading* and the other three variables as well.

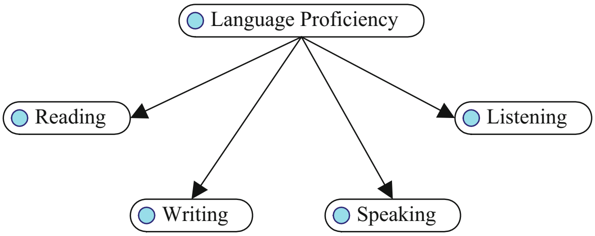


Fig. 4.13 A graph showing one level of breakdown in language skills
Reprinted with permission from ETS.

Graphs whose edges run in the "causal" direction are generally simpler than graphs whose edges run in the "diagnostic" direction. That is because relationships are understood as events or observations that are conditionally independent, given underlying factors of a situation or prior events, perhaps from a substantively grounded understanding of the situation at hand. Pearl (1988) and others have used that to build a theory of causal modeling (see Sect. 10.7). Following this tradition, many authors call Bayesian networks causal models. The influence diagram school, on the other hand, tends to avoid the word "causal" because it is often misinterpreted by the lay public (Sect. 10.7).

Causality is not necessary for building graphical models. Weaker notions like "tendency to cause," "influence," or "knowledge dependence" are sufficient. In Fig. 4.13 the meaning of the edges is a "part-of" relationship—*Reading* is *a part of Language Proficiency*. Ultimately, the meaning of the edges is knowledge dependence. An edge from A to B means that knowledge about A affects our beliefs about B.

This does not mean that if our domain experts have a well established causal theory, we should throw it away. Rather, the causal theory can help build efficient models. In particular, cognitive theory about factors which go into performance on an assessment task can be used to build a mathematical model of that performance (Mislevy 1994). Psychometric models in general posit latent variables that characterize aspects of students' knowledge and skill, which are modeled as parents of aspects of their behavior. This is as true for cognitive diagnosis models as it is for more traditional psychometric models such as classical test theory, item response theory, and factor analysis. But we should not wait until we have a universally agreed upon causal model before building a graph. Any theory of the domain should be enough to get us started.

In evidence centered design , domain experts and assessment designers are encouraged to draw preliminary versions of the future proficiency and evidence models before settling down to built the final statistical model of the test. During this domain modeling phase, the edges in the graph are often labeled with the type of relationship (e.g., prerequisite, part-of, induces, inhibits). These labels are not used in the final Bayes nets, but help the experts think about how to model the conditional probability tables.

4.5 Other Representations

One of the principle advantages of using graphical models for educational assessment is that they provide a useful representation for discussing mathematical models with domain experts who may be less familiar with mathematical notation. However, graphical models are not the only system to use graphs as a means of conveying mathematical models. This section discusses several of these other representations and their relationships to graphical models. *Influence diagrams* (Sect. 4.5.1) are a specific generalization of graphical models which add decision variables to the mix. *Structural equation models* (Sect. 4.5.2) also use graphs to represent complex models, but with some notational differences which are important to point out. Section 4.5.3 briefly lists some other related models.

4.5.1 Influence Diagrams

Influence diagrams (Howard and Matheson 1984; Shachter 1986; Oliver and Smith 1990) have had a strong influence on the modern development of graphical models, especially the application in artificial intelligence. Influence diagrams differ from Bayesian networks in that they use both probabilities and utilities which represent preferences among outcomes. Influence diagrams also use three classes of nodes, one to represent random variables, one to represent decisions (under the control of the decision maker), and one to represent utilities. The "solution" to an influence diagram is a strategy for making the decisions involved in the problem to maximize the expected utility.

Figure 4.14 shows a typical influence diagram.

- Square boxes are *decision variables*. Arrows going into decision variables represent information available at the time when the decision is to be made.

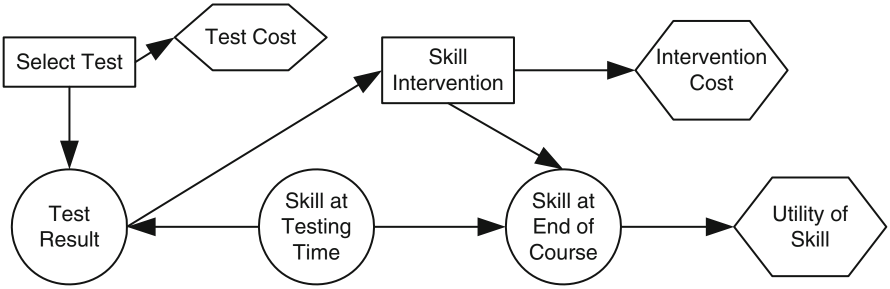


Fig. 4.14 Influence diagram for skill training decision
Reprinted from (Almond 2007b) with permission from ETS.

- Round boxes are *chance nodes* (random variables).
- Hexagonal boxes are *utilities.* Costs are negative utilities.

The example in Fig. 4.14 brings up some interesting issues so it is worth exploring in a little more detail.

Example 4.1 (Value of Testing). *Suppose that we are trying to teach a certain skill to a certain student. We have a utility associated with this student knowing the skill at the end of the course. The student's probability of knowing the skill at the end of the course will depend on both the student's skill level at the beginning of the course and what kind of instruction the student receives. The instruction has certain costs (both monetary and student's time) associated with it (as does no instruction, but we can scale our utilities so that that cost is zero). We do not know the student's ability at the beginning of the course, but we can give the student a pretest whose outcome will depend on the student's ability. This pretest also has a cost associated with it. We can observe the outcome of the pretest when we make the decision about what instruction to give.*

The decision of what instruction to give depends not only on whether or not the student seems to have the skill from the pretest, but also the value of the skill and the cost of the instruction. If the instruction is very expensive and the skill not very valuable, it may not be cost effective to give the instruction even if we know the student does not have the skill. Similarly, the decision about whether or not to test will depend on the cost of the test and the cost of the instruction. If the instruction is very inexpensive (for example, asking the student to read a short paper or pamphlet), it may be more cost effective to just give the instruction and not bother with the pretest.

This example brings up the important concept of *value of information* (Matheson 1990). This will come up again, along with its close cousin *weight of evidence* when we discuss explanation and task (item) selection in Chap. 7.

An influence diagram with all chance nodes is called a *relevance diagram* and is a Bayesian network. This fact, along with the efficient algorithms for Bayesian networks (Chap. 5) has caused most of the current influence diagram research to be cast in terms of Bayesian networks. If the number of decision variables is low, then it is often efficient to represent them as random variables in a Bayesian network and simply assign them distributions with probability 1 for a particular choice.

4.5.2 Structural Equation Models

Before graphical models, Wright (1921) (also Wright 1934) used graphs to represent statistical models in his pioneering development of *path analysis*. This technique has been a strong influence on many of the early developers of Bayesian networks (Pearl 1988). Path analysis has been popular in the social sciences because the pictorial representation of the model is often easier to use than mathematical notation.

Bayesian networks are not the only models using a graphical representation to descend from path analysis. *Structural equation models* (Bollen 1989; Joreskog and Sorbom 1979; Kaplan 2000), or SEMs, have been quite popular in psychological and educational testing applications. Although they concentrate on modeling associations in populations rather than behavior of a single individual, there are many similarities.

This book will not cover structural equation models, as they are covered by many other authors. But it will be helpful to notice a few key differences between structural equation models and Bayesian networks.

1. SEMs most often work with continuous variables which are assumed to have a multivariate normal distribution, and Bayesian networks most often use discrete variables that have a multinomial distribution. There are exceptions on each side (see Whittaker 1990; Lauritzen 1996), but this rule holds for many applications.
2. SEMs model the covariance matrix, while graphical models model the inverse of the covariance matrix (Whittaker 1990). Zeros in the covariance matrix imply marginal independence, while zeros in the inverse covariance matrix imply conditional independence. This drives the next difference.
3. SEMs and Bayesian networks use slightly different graphical notations. Some of these are obvious: SEMs allow bidirectional or undirected edges to model correlations, while all edges in Bayesian networks must be directed and the directed graph must be acyclic. Perhaps more subtle is what a missing edge means. “The missing links in those statistical models [graphical models] represent conditional independencies, whereas the missing links in causal diagrams [SEMs] represent absence of causal connections … that may or may not imply conditional independencies in the distribution” (Pearl 1998, p. 237).

4. SEMs frequently model error terms as nodes in the graph, while in Bayesian networks, they are often implicit in the distribution locked in the edges.
5. In practice, SEM modeling usually focuses on modeling the distribution of a population, while Bayesian network modeling often focuses on calculating probabilities for an individual (see Part I). However, when estimating the parameters of the Bayesian network, the population distribution must be considered (Part II).

In short, SEMs and Bayesian networks are both rich notations for describing complex multivariate distributions and using graphs to visualize the relationships. However, the rules are slightly different, so not all SEMs and Bayesian networks are equivalent (Pearl 1988, notes that the ones that are equivalent are called *recursive models* in the SEM literature). So, when faced with a graph, it is important to know which representation is implied. Anderson and Vastag (2004) provide a side-by-side comparison of SEMs and Bayesian networks.

4.5.3 Other Graphical Models

Although Bayesian networks are one of the most frequently used graphical models, there exist other kinds of graphical models as well. Generally, a graphical models is a representation of a probability distribution that factors according to a graph. So a set of rules for associating graphs (or hypergraphs) of various types with factorizations, and conditional independence conditions produces a new kind of graphical model. This section provide a few pointers into the rich literature on this topic.

The term *graphical model* comes from Darroch et al. (1980) where it is used for modeling contingency tables. According to the definition, a model is graphical if for every set of variables for which all two-way interactions are included in the model (this would be a clique in the graph), all higher order interactions are included as well. The implication of this definition is that the joint probability distribution factors according to the cliques of the graph. These models are particularly convenient computationally, and usually quite easy to interpret.

The restriction to discrete variables is convenient because the integral which is the denominator of Bayes' rule turns into a sum. If that restriction is lifted, then the denominator becomes, in some cases, an integral that usually cannot be solved analytically. One exception is if all of the variables are normal. In this framework, directed edges behave like regression models. Whittaker (1990) provides a general reference for both discrete and multivariate normal graphical models, as well as some cases of mixed models.

Edwards (1990) describes one class of mixed graphical model which is convenient to work with under the name *hierarchical interaction model.* Edwards (1995) describes a system for fitting these models to data, realized in the software package MIM. Lauritzen (1996) describes a general case of *conditional Gaussian* models in which normal variables are allowed to be children of discrete parents, but discrete variables are not allowed to have continuous parents (Contrast this to item response theory (IRT) models in which the continuous latent trait, θ, is a parent of discrete observable outcome variables). Lauritzen (1996) shows how conditional Gaussian models support the algorithms of Chap. 5.

Cox and Wermuth (1996) describe an extension of these ideas called *chain graph models*. Chain graphs use a mixture of directed and undirected edges. The directed edges represent conditional relationships, i.e., the distribution of the child is given conditioned on the parent. The undirected edges represent correlational relationships, i.e., the variables in question are given a joint distribution given common parents.

To unify very similar work in Bayesian networks, influence diagrams, discrete dynamic programming, and graphical belief functions, Shenoy and Shafer (1990) developed a general framework they called *valuation-based systems*. Valuation-based systems associate quantitative information with relationships among variables. The probability potential introduced above is an example of a valuation, as are utilities in influence diagrams. To support this framework, valuations need to support a couple of operations and properties. First, there needs to be some notion of conditional independence related to the factorization of the model. Second, the valuations must support a combination and a projection (changing the frame for the valuation) operation. Finally, it must be possible at least under some circumstances to interchange the combination and projection operators. Given these conditions the algorithms of Chap. 5 can be used to solve problems.

Some examples of valuation based systems include: *Bayesian Networks* (Shenoy and Shafer 1990; Almond 1995), *Influence diagrams* (Shenoy 1991), *Discrete Dynamic Programming* (Bertelè and Brioschi 1972), *Graphical Belief Functions* (Shenoy and Shafer 1990; Almond 1995), and *Mixed Graphical Models* (Cobb and Shenoy 2005). This last paper shows that if the distribution of the continuous variables can be described through a mixture of exponential distributions, then the Shenoy and Shafer (1990) algorithm can be used to get exact solutions. Mixtures of exponentials can often provide good approximations to complex functional forms.

The computation schemes of Chap. 5 rely on the conditional independence properties of Bayesian networks (or, by extension, the other graphical models described here). In particular, Bayesian networks are suitable for the purpose of gathering data, drawing inferences, and making decisions about individuals as data arrive for each of them. This is the reason for our choice of Bayesian networks as the basis of our models for educational testing.

Exercises

4.1. The graph for the item response theory (IRT) model in Chap. 3 (Fig. 3.3) has all of the arrows pointing from θ to the observable item outcomes, X_i. Why did we choose to draw the arrows in that direction?

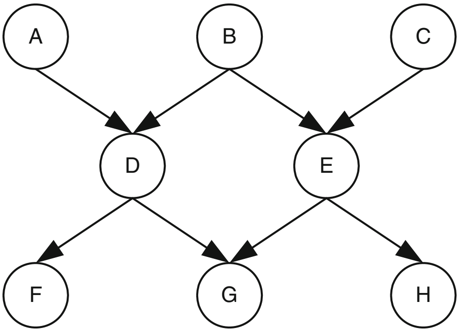


Fig. 4.15 Graph for use in Exercise 4.2
Reprinted from Almond et al. (2007) with permission from ETS.

4.2. Refer to Fig. 4.15. In each of the following scenarios state whether the variables associated with nodes A and C are independent. In each case, if they are not independent, indicate a set of additional variables such that conditioning on these variables would render A and C independent again.

a. No variable values have been observed.
b. Values for the variables F and H are observed, but no other variables are known.
c. A value for the variable G has been observed, but all other variables are unknown.

4.3. In developing an assessment for algebraic sequences, the domain experts identified four proficiency variables: *overall sequence proficiency*, *arithmetic sequence proficiency* (sequences like $2, 4, 6, 8$), *geometric sequence proficiency* (sequences like $2, 4, 8, 16$), and *other recursive proficiency* (sequences that do not follow arithmetic or geometric rules, like the Fibonacci sequence). According to the experts the last three proficiencies are "part of" overall proficiency. What direction should the edges representing this relationship be drawn? Why? Are other edges needed between the remaining variables? What question could be asked of the experts to see if additional edges are needed?

4.4. Start with the language proficiency model on Fig. 4.13. Now add nodes representing the scored outcomes from the following tasks:

a. A pure Reading Task.
b. A pure Listening Task.
c. A task which requires a written response based on textual stimulus which the candidate must read.
d. A task which requires a spoken response based on a audio stimulus which the candidate must listen to, with written instructions.
e. A task which requires a candidate to write a transcript of an audio stimulus, with spoken instructions.

Connect the new outcome variables nodes to the proficiency variables nodes already in the graph. Now connect the parents of each outcome variable to form the moral graph. What is the size of the largest clique? What happens to the conditional independence of the skills? (Mislevy 1995c).

4.5. Suppose we have a proficiency model consisting of an overall proficiency skill and several subskills. The experts tell us that we can model the subskills as conditionally independent given the parent. Suppose further that our test consists of a collection of items which tap each pair of the subskills, so that the moral graph for the proficiency model will be saturated (there will be an edge between every pair of variables). Given that the moral graph will be saturated, why is it still better to have the arrows go from the overall proficiency to the subskills? Hint: Assume all variables have four levels. The expert must then specify three probability values per combination of parent states (because the four probabilities in each conditional distribution must sum to 1). Thus, if a node has two parents, the expert must specify $3 \times (4 \times 4)$ probability values.

4.6. Consider a dance performance competition in which there are three performers and three judges. Draw a Bayesian network to represent this structure in each of the following scenarios:

a. Each dancer gives a single performance which receives a single rating agreed upon by all three judges.
b. Each dancer gives a single performance and receives a separate performance from each judge.
c. Each dancer gives three performances, each performance is rated by a different judge.
d. Each dancer gives three performances, each performance is rated by all three judges.

For the models with multiple performances, should there be arrows between the performances? For the models with multiple ratings, should there be arrows between their ratings? Justify your answers.

5

Efficient Calculations

The previous chapters have laid a foundation for building probability models and embedding them in a graphical structure. The graphical structure reflects our knowledge bout interrelationships among the variables. Once we have expressed all of the interrelationships in terms of a joint probability distribution, it is always possible in principle to calculate the effect of new information about any subset of the variables on our beliefs about the remaining variables (i.e., to propagate the evidence).

For even relatively small numbers of variables, however, the cost of updating the full joint distribution using the definitional expression of Bayes theorem becomes prohibitive. A model with 15 variables with four values each already means working with a joint probability table with over a trillion entries. We have intimated that when the topology of the graph is favorable, we will be able to carry out calculations more efficiently. Mathematically speaking, topologies that are favorable for computing are those that have small cliques, that is, low treewidth. Using the algorithms described in this chapter, the cost of the computation only grows linearly in the total number of variables, but grows exponentially with the size of the largest clique. More informally, if we have structured our understanding of the domain so that the important interactions take place among small subsets of variables, then we can exploit that structure to create an efficient calculation algorithm.

This chapter introduces efficient calculation in networks of discrete variables. The objective is to ground intuition with a simplified version of a basic junction-tree algorithm, illustrated in detail with a small numerical example. More complete descriptions of this so-called *fusion and propagation algorithm* are available in Jensen (1996), Cowell et al. (1999), and Neapolitan (2004). Practically, such calculations are done with computer programs rather than by hand, and Appendix A describes how to get several commercial and free research programs that support these calculations. The basic belief-updating algorithm presented here is just one of a large number of variants on the message-passing algorithm described in Pearl (1988); Sect. 5.6 describes some of them.

R. G. Almond et al., *Bayesian Networks in Educational Assessment*,
Statistics for Social and Behavioral Sciences, DOI 10.1007/978-1-4939-2125-6_5

Section 5.1 begins by reexamining probability updating in a simple two-variable model, which corresponds to a graph with one variable Y depending on a parent variable X. However, this simple example defines two operations, *combination* and *marginalization*, which are key to the *belief-updating algorithm* for more general graphs. Section 5.2 describes how the belief-updating algorithm works in undirected graphs that have no loops, such as chains and trees. This section adds the *message* operation. Section 5.3 notes that this method of propagation breaks down when a graph contains a cycle, or multiple paths from one variable to another. It then describes how the method can be generalized to propagating at the level of cliques of variables in what is called a *junction tree* or *join tree*, rather than at the level of individual variables. The construction of a junction tree and propagation of evidence with a junction tree are illustrated with a simple example of how a junction tree is constructed. Section 5.4 discusses implications of this approach for assessment, including the idea of distinguishing fragments of Bayes nets that correspond to proficiency models and evidence models. These pieces can be assembled dynamically to absorb evidence about students' knowledge in applications such as computerized adaptive testing. Section 5.5 presents an alternative representation for the structure of an assessment, the Q-Matrix. Section 5.6 briefly surveys alternative updating techniques, both exact and approximate, for use in situations when the algorithm of Section 5.3 is not viable.

5.1 Belief Updating with Two Variables

In its definitional form, Bayes theorem (Theorem 3.1) can be applied in principle to any number of variables. However, the calculations soon become intractable. The first steps toward efficient computation in large networks of variables were developed for special structures such as chains and trees, in which globally coherent updating could be accomplished by working through the structure two variables at a time.

It is easy to understand probability updating and Bayes theorem in terms of definitional, or what might be called "brute force," calculation. In discrete Bayes nets, one enumerates the probabilities of all the possible combinations of values of all the variables, reduces the set to reflect news about the values of some variables, and carries out some simple arithmetic to calculate the probabilities of all the remaining possibilities. This is easy to understand and easy to demonstrate, as we will do in this section—as long as there are not too many variables and not too many combinations of values. But the basic steps in the brute force calculation are the building blocks of more efficient calculation in more complex networks.

Consider the case of just two variables, X and Y, and suppose that our initial knowledge consists of marginal probabilities $p(x)$ for the values of X

and conditional probabilities $p(y|x)$ for values of Y given values of X. The graph takes the now-familiar form shown as Fig. 5.1.

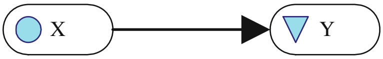


Fig. 5.1 Graph for the distribution of X and Y
Reprinted from with permission from ETS.

As discussed in Chap. 4, this directed representation of the relationship between the two variables supports updating our belief about either X or Y given the values of the other. If we learn the value of X, we update belief about Y directly via the conditional probability distribution $p(y|x)$. If we learn the value of Y, we update belief about X via Bayes theorem. The following example begins by reviewing Bayesian updating in the context of a numerical illustration.

We will then recast the problem in terms of operations on potentials on this (very simple!) graphical structure. In general, we call a multiway array of numbers where each dimension corresponds to a variable in our model, a *potential*. Potential tables are probability distributions for which we have relaxed the normalization constraint. That is, the numbers are nonnegative, and they are in the right proportions to reflect relative probabilities of outcomes and events. A potential could represent a probability distribution, a product of a set of conditional probability distributions or another intermediate quantity in a probability calculation. If we want to interpret a potential as a probability, we often must normalize it.

Example 5.1 (Dental Hygienist Assessment). *Let X represent a dental hygiene student's proficiency in examining patients, and Y represent whether a student takes an adequate patient history in a particular simulation task. The single proficiency variable, X, can take two values, $x_1 = \texttt{expert}$ and $x_2 = \texttt{novice}$, and the observable variable, Y, can also take two values, $y_1 = \texttt{yes}$ and $y_2 = \texttt{no}$. In this assessment the work product is the examinee's sequence of actions in taking a patient history. Assume we can determine unambiguously whether a given sequence is adequate or inadequate.*

Suppose that it is equally likely that a student is an expert or a novice, so $p(\texttt{expert}) = p(\texttt{novice}) = .5$. Suppose further that an expert's probability of taking an adequate history in such tasks is .8, and a novice's is .4. Thus $p(y_1 \mid x_1) = .8$ and $p(y_2 \mid x_1) = .2$, and $p(y_1 \mid x_2) = .4$ and $p(y_2 \mid x_2) = .6$.

In an operational assessment we would want to observe an examinee's performance, evaluate its adequacy, and update our belief about the examinee's expert/novice status. We observe Jay take an adequate history in the task; that is, $Y = y_1 = \texttt{yes}$. Bayes theorem updates our initial beliefs of (.50,.50) *probabilities for expert and novice as follows:*

$$p\,(x_1 \mid y_1) = \frac{p\,(y_1 \mid x_1)\,p\,(x_1)}{p\,(y_1 \mid x_1)\,p\,(x_1) + p\,(y_1 \mid x_2)\,p\,(x_2)} = \frac{.8 \times .5}{.8 \times .5 + .4 \times .5} = .67$$

and $p\,(x_2 \mid y_1) = 1 - p\,(x_1 \mid y_1) = .33.$

Now let us see how this example can be expressed in terms of potential tables and operations on them. We will first work through the symbolic representation, then show the corresponding numbers from Example 5.1.

There is one clique in this undirected graph, $\{X, Y\}$. From the information about $p(x)$ and $p(y|x)$, we can construct a two-way table of the joint probabilities $p(x, y)$ for all possible (x, y) pairs. The results are shown as the top panel in Table 5.1. This is a potential table for $p(x, y)$, which at this point conveys the initial beliefs. The margins for X and Y are shown at either side of the joint distribution; both are consistent with the joint probabilities in the center.

We now learn that $Y = y_1$. Because Y and X are dependent, the value we learned for Y provides *evidence* for X. This evidence can be expressed as the vector $[1, 0]$. In order to combine this new belief about Y with our initial belief about the joint distribution of X and Y, we first replicate the information about Y into a potential table for X and Y. This is shown in the second panel of Table 5.1.

To instantiate the observed value of Y, multiply each cell in the initial table for X and Y by the contents of the corresponding cell for the new evidence. The result is the third panel in Table 5.1. The states that now have zero probability (i.e., are impossible) are colored gray. Note that the contents of the table are no longer a proper probability distribution as the values do not sum to one, but rather sum to $p(y_1)$. Interpreting the values as a probability requires normalizing the values in the table by dividing by the sum of the elements. This yields the final panel of Table 5.1, a potential table that represents the conditional probability distribution for X given that $Y = y_1$.

Example 5.2 (Dental Hygienist Assessment, Example 5.1, Continued). *Table 5.2 gives the numbers for the Dental Hygienist example that correspond to the symbolic representation of the potentials in Table 5.1. We see that the operations on the potentials reflect the calculations we carried out in the definitional application of Bayes theorem. Learning that a student takes an adequate history means focusing attention on the* `yes` *column. The ratio of the values is 2:1, and normalizing, we find that we have updated our initial (.50,.50) probabilities for expert and novice to (.67, .33).*

Table 5.1 Updating probabilities in response to learning $Y = y_1$

	y_1	y_2	P(X)
x_1	$p(y_1\|x_1)p(x_1)$ $= p(x_1, y_1)$	$p(y_2\|x_1)p(x_1)$ $= p(x_1, y_2)$	$p(x_1)$
x_2	$p(y_1\|x_2)p(x_2)$ $= p(x_2, y_1)$	$p(y_2\|x_2)p(x_2)$ $= p(x_2, y_2)$	$p(x_2)$
P(Y)	$p(y_1)$	$p(y_2)$	1

a) Table for initial joint and marginal probabilities.

	y_1	y_2
x_1	1	0
x_2	1	0

b) Potential table representing evidence, $Y = y_1$.

	y_1	y_2	P(X)
x_1	$p\,(x_1, y_1) \times 1$ $= p\,(x_1, y_1)$	$p\,(x_2, y_1) \times 0$ $= 0$	$p(x_1, y_1)$
x_2	$p\,(x_2, y_1) \times 1$ $= p\,(x_2, y_1)$	$p\,(x_2, y_2) \times 0$ $= 0$	$p(x_2, y_1)$
P(Y)	$p(y_1)$	0	$p(y_1)$

c) Combine initial probabilities with evidence.

	y_1	y_2	P(X)
x_1	$p(x_1, y_1)/p(y_1)$ $= p(x_1\|y_1)$	0	$p(x_1\|y_1)$
x_2	$p(x_2, y_1)/p(y_1)$ $= p(x_2\|y_1)$	0	$p(x_2\|y_1)$
P(Y)	1	0	1

d) Normalize by dividing by the grand total, $p(y_1)$.

This example introduced most of the key operations on potentials we will need to update Bayes nets in light of new evidence. They are *projecting* a potential for one set of variables into a potential for an overlapping set of

Table 5.2 Numerical example of updating probabilities

	yes	no	P(X)
expert	$.8 \times .5 = .40$	$.2 \times .5 = .10$	.5
novice	$.4 \times .5 = .20$	$.6 \times .5 = .30$	.5
P(Y)	.6	.4	1

a) Table for initial joint and marginal probabilities.

	y_1	y_2
x_1	1	0
x_2	1	0

b) Potential table representing evidence, $Y = y_1$.

	yes	no	P(X)
expert	$.4 \times 1 = .4$	$.1 \times 0 = 0$	.4
novice	$.2 \times 1 = .2$	$.3 \times 0 = 0$	.2
P(Y)	.6	0	.6

c) Combine initial probabilities with evidence.

	yes	no	P(X)
expert	$.4/.6 = .67$	0	.67
novice	$.2/.6 = .33$	0	.33
P(Y)	$.6/.6 = 1$	0	1

d) Normalize by dividing by the grand total, .6.

variables (both up to a larger set and down to a subset), and combining two potentials over the same set of variables. (All we still need is an operation to pass messages from one potential to another to update beliefs, which we will get in Sect. 5.2.)

To go from a bigger space (like $\{X, Y\}$) to a smaller one (like $\{X\}$), sum across the values of the unused variables. The resulting distributions are often written in the margins of the joint table and hence they are called *marginal distributions*. We already ran into the marginal distribution when talking about joint probability distributions in Chap. 3. The process of calculating the marginal distribution from the joint distribution is called *marginalization*. We use the symbol $\Downarrow_x$ to denote marginalizing over the variable X, and define it as follows:

$$p\,(x, y) \underset{x}{\Downarrow} = \sum_y p\,(x, y) = p(x) \ , \tag{5.1}$$

where the sum is taken over all possible values Y.

To go from a smaller space (like $\{Y\}$) to a larger one (like $\{X, Y\}$), simply replicate the distribution over new dimensions. We did this in the example because as the potential representing $Y = y_1$, namely $[1, 0]$, is a potential defined only over the variable Y. We just replicated it over X to get the second panel in Table 5.2.

To *combine* the information in two potentials over the same variables, multiply them together element by element. This is how we got the third panel in Table 5.2, as we combined the potential representing the initial distribution with the potential representing $Y = y_1$. Note that the combination operation was also used in constructing the initial table. This was the combination of the potentials representing $p(x)$ and $p(y|x)$. The potential over $p(x)$ needed to be projected onto the larger space $\{X, Y\}$ before the combination could occur. The symbol $\otimes$ is used to denote combination of potentials.

Finally, to interpret the potential as a probability, *normalize* the potential by dividing by the sum of the elements. This was done as the last step of the calculation. This calculation is often done last because (a) normalization is only needed to interpret the results, and (b) delaying normalization as long as possible increases the numerical stability of the results.

In principle, belief updating and marginalization as done in this section can be carried out with an arbitrarily large number of discrete variables. All the calculations are done with a large potential table over all the variables in the problem. While the procedure is straightforward conceptually, the problem is feasibility. Even with only ten dichotomous variables, there is a table of size 2^{10} to work with. The cost grows exponentially with the number of variables in the table associated with the largest clique; increasing the number of variables beyond six or seven (or even fewer if each variable has many states) makes computation impractical. The remainder of this chapter discusses a strategy that exploits the conditional independence conditions in the model graph to ensure all computations happen over tables of a manageable size.

5.2 More Efficient Procedures for Chains and Trees

Kim and Pearl (1983) introduce an updating algorithm for a chain of variables, in which computation only grows linearly in the number of variables. Almost all of the various algorithms for performing calculations in Bayesian networks are variations on the basic Kim and Pearl approach. The variation presented here is based on the junction-tree algorithm of Cowell and Dawid (1992).

Section 5.2.1 presents the basic algorithm on a very undirected graph, a chain of variables. Section 5.2.2 extends the algorithm to polytrees, which are basically directed graphical structures whose undirected graphs would be trees. Finally, Sect. 5.2.3 talks about handling evidence that is not certain; this will have some interesting applications later in the chapter. The simple approach described in this section can be extended to more complex graphical structures. Section 5.3 will take up the case of more complex models.

5.2.1 Propagation in Chains

A chain is a set of variables whose joint distribution takes a form like this:

$$\mathrm{P}\left(W,X,Y,Z\right)=\mathrm{P}(Z|Y)\times\mathrm{P}(Y|X)\times\mathrm{P}(X|W)\times\mathrm{P}(W)\ . \qquad (5.2)$$

That is, each variable except for the first depends directly on only the one variable preceding it. The acyclic digraph for such is system is also a chain in the graph-theory sense, as shown at the left of Fig. 5.2.

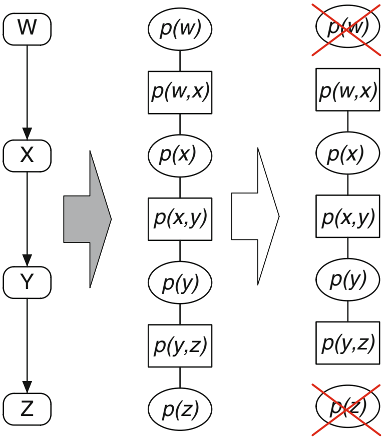


Fig. 5.2 The acyclic digraph and junction tree for a four-variable chain, corresponding to Eq. 5.2. In the junction tree (in the *middle right*) the *square boxes* correspond to cliques in the digraph (on the *left*). The *round boxes* correspond to intersections between cliques. In the final version on the *far right*, the two "intersection" nodes that only connect to a single node are dropped

Reprinted with permission from ETS.

Moving from the right to the left in Fig. 5.2 is a transformation of the original graph called a *junction tree*. This junction tree has several notable properties. First, its structure contains nodes for both variables themselves and pairs of adjacent variables. The nodes for the pairs are where interrelationships among variables that directly influence one another are manipulated. These are called *clique nodes* and they correspond to the cliques in the digraph (more specifically, as we shall see below, the cliques of the undirected graph corresponding to the digraph). The nodes for individual variables are intermediate areas where information common to adjacent cliques, necessary for updating from one clique to the next, are stored. These are called *intersection nodes*. The junction tree in the middle contains two "intersection" nodes, $p(w)$ and $p(z)$ which do not join two clique nodes. As these are not needed for computation, they are commonly dropped (far right in the figure). Section 5.3 describes the properties of junction trees in more detail.

In each node of the junction tree we will store a *potential table* defined over the variables in the node. We would like this table to be proportional to the joint probability distribution over the variables in the node. There are several ways to initialize the values in a junction tree. In the case of the chain, the easiest way is to follow the recursive definition of the joint distribution, Eq. 5.2.

Start with the clique node $\{W, X\}$. From Eq. 5.2 the joint distribution $\mathrm{P}(W, X) = \mathrm{P}(X|W)\mathrm{P}(W)$. Construct potentials corresponding to $\mathrm{P}(X|W)$ and $\mathrm{P}(W)$ and combine them to create a potential corresponding to $\mathrm{P}(X, W)$. Store this potential in the node $\{W, X\}$. Marginalize out the variable W to get the potential $\mathrm{P}(X)$ and put that in the corresponding intersection node. We will call the potential stored in the Node $\{W, X\}$, $p(w, x)$, and the potential stored in the Node $\{X\}$, $p(x)$.

The next clique node is $\{Y, X\}$; it has the joint distribution $\mathrm{P}(X, Y) = \mathrm{P}(Y|X)\mathrm{P}(X)$. The first term was specified in Eq. 5.2, the second term was the value we just calculated and stored in the intersection node $\{X\}$. This node is conveniently connected to the new clique node making it easy to find. Combining these two potentials, we calculate the potential for the node $\{Y, X\}$ and marginalize it to get the potential for the intersection node $\{Y\}$. A similar procedure produces the initial potential for the last clique node $\{Z, Y\}$.

Even though the largest table contains only two variables, the junction tree contains all the information necessary to reconstruct the full joint distribution of all four variables. The conditional independence relationships that allowed us to simplify the joint distribution in Eq. 5.2 also allow us to work with smaller tables, rather than one large table for all possible combinations of all variables.

Updating the distribution in response to new information can be carried out with a generalization of the updating approach described in Sect 5.1. The remainder of this section shows how to do this first in symbols, then with a numerical example.

The initialization process stored a potential with each node of the junction tree given in the right side of Fig. 5.2. Call these potentials $p_{old}(w, x)$, $p_{old}(x)$, $p_{old}(x, y)$, $p_{old}(y)$ and $p_{old}(y, z)$. Now suppose that evidence, e_1, arrives about X. We represent this new information as a potential over $\{X\}$, and denote it $\phi(x)$. We now enter this into the system in the following steps:

1. Pick any clique node containing $\{X\}$, and update the potential in that node by combining it with $\phi(x)$. We will use $\{X, Y\}$ because this choice allows us to illustrate updating both up and down the chain. Call the new potential in that node $p_{new}(x, y) = p_{old}(x, y) \otimes \phi(x)$ and note that it represents $\mathrm{P}(X, Y|e_1)$. This is done just as in Sect. 5.1. At this point, our potential for $\{X, Y\}$ correctly reflects our new belief about X, as obtained directly in the form of e_1, and about Y, as obtained by projection and combining into the $\{X, Y\}$ potential. However the nodes for all of the

other cliques and clique intersections still contain the initial beliefs and are inconsistent with the new state of information.

2. Next, update the potentials in the neighboring *clique nodes* by passing messages from clique node to clique node. Note that the clique nodes are connected through intersection nodes, and hence the messages will be passed through intersection nodes. We will move down the chain first, from $\{X, Y\}$ to $\{Y, Z\}$. Denote the message from $\{X, Y\}$ to $\{Y, Z\}$ as "$\{X, Y\} \Rightarrow \{Y, Z\}$." It takes the form of a potential over the clique intersection $\{Y\}$. It is calculated as follows. Call the potential over $\{Y\}$ already in the intersection node $p_{old}(y)$. Then calculate $p_{new}(y) = p_{new}(x, y) \Downarrow_y$, the marginal distribution over $\{Y\}$ of the new potential at $\{X, \}$. The message sent to the $\{Y, Z\}$ node will be ratio of the new intersection potential divided by the old:
$$\{X, Y\} \Rightarrow \{Y, Z\} = p_{new}(y) \oslash\ p_{old}(y)\ , \tag{5.3}$$
where $\oslash$ is element by element division of potentials.
3. Adjust the potential in the node receiving the message by combining the message with the potential currently in that node (after suitably extending it to the set of variables in the node being updated):
$$p_{new}(y, z) = p_{old}(y, z) \otimes (p_{new}(y) \oslash p_{old}(y))\,. \tag{5.4}$$
This essentially scales the potential in the node $\{Y, Z\}$ by the amount the information about $\{Y\}$ changed because of the added evidence.
4. Now, pass messages up the chain from the Node $\{X, Y\}$ where the new evidence was entered. In this case, calculate the message $\{X, Y\} \Rightarrow \{W, X\}$ in the same manner as Eq. 5.3:
$$\{X, Y\} \Rightarrow \{W, X\} = \left(p_{new}\left(x, y\right) \underset{x}{\Downarrow} \oslash p_{old}\left(x\right)\right) = p_{new}\left(x\right) \oslash p_{old}\left(x\right)$$
.
5. Calculate the new value for Node $\{W, X\}$ by combining the potential already in the node by the just-received message as in Eq. 5.4:
$$p_{new}(w, x) = p_{old}(w, x) \otimes (p_{new}(x) \oslash p_{old}(x))\,.$$
6. If either Node $\{W, X\}$ or $\{Y, Z\}$ had additional neighbors then the process would be repeated. In for each neighbor, an analogue to Eq. 5.3 would be used to calculate the effect of the evidence on the variables in the intersection node. That would be combined with the information already in the clique node using an analogue of Eq. 5.4. This process would be repeated until all nodes in the junction tree have received a message containing the information from the newly entered evidence.

If the evidence arrived about the Variable W, then the procedure would have started with the node $\{W, X\}$ and the messages flowed down the chain.

Evidence about Z is entered in $\{Y, Z\}$ and updating flows up the chain. In all cases, other than the obvious changes of variable, the procedure is the same. As with X, there are two choices of where to enter evidence about Y. The messages flow outward from there to the edges of the junction tree, and the same result is obtained with either choice.

After this procedure, the potentials in the tree now represent the joint distribution $\mathrm{P}(W, X, Y, Z|e_1)$. If additional evidence, e_2 were to arrive about another variable, the same procedure would be applied a second time, now using the current potentials as $p_{old}(\cdot)$. Following Pearl (1988) we will call this the *belief-updating algorithm*, although the algorithm given here is a variant of Pearl's algorithm. This variant is sometimes called the *Hugin algorithm* because of its use in the software package HUGIN (Appendix A). A numerical illustration of this algorithm is given in Example 5.8.

Note that it is not necessary that the values in the potential tables be normalized to sum to one for the message passing; it is only necessary that they reflect the correct proportions in the final joint probability. Normalizing is necessary only when one wants to interpret the marginal distribution for one or more variables. Delaying the normalization until just before the results are interpreted improves both the speed and numerical stability of the algorithm.

Actually, the normalization constant may be of interest in its own right. Recall that in Table 5.1 the normalization constant was $p(y_1)$, that is the probability of the evidence. This holds with more complex patterns of evidence as well. In particular, the normalization constant is the probability (likelihood) of the observed pattern of evidence. This is useful when evaluating how well the data fit the model (see Chap. 10 for more about this).

Example 5.3 (Updating in a Chain). *Let the variables W, X, Y, and Z in Fig. 5.2 all be dichotomous random variables defined on $\{0,1\}$. Let* $\mathrm{P}\,(W = 1) = .6$, *and*

$$\begin{aligned} P\,(X = 1 \mid W = 1) = P\,(Y = 1 \mid X = 1) = P\,(Z = 1 \mid Y = 1) = .9 \\ P\,(X = 1 \mid W = 0) = P\,(Y = 1 \mid X = 0) = P\,(Z = 1 \mid Y = 0) = .2. \end{aligned}$$

This information produces the initial potentials in the junction tree shown in Table 5.3a. Now suppose we learn that $X = 1$; thus, $\phi(x) = [1, 0]$. This evidence is entered into the potential $p_{old}\,(x, y)$ to produce $p_{new}\,(x, y)$, as shown in Table 5.3b. At this point, we have updated the potential for $\{X, Y\}$, but the other potentials remain at their initial values and inconsistent with our new beliefs as shown in Table 5.3c.

We obtain $p_{new}(y)$ as $p_{new}(x, y) \Downarrow_y$, namely $[.558, .062]$. From the initial status of the junction tree, $p_{old}(y) = [.634, .366]$. The message to be passed from $\{X, Y\}$ to $\{Y, Z\}$ is thus calculated as in Table 5.3d. This message is interpreted as a signal to shift belief about Y in the ratio .880/.185, and whatever this implies for the other variables in the receiving clique (in this case, just Z) through their association with Y, as executed in Table 5.3e.

Table 5.3f shows the potentials after e_1 has been propagated down the chain but not yet up the chain.

Propagating evidence up the chain from $\{X, Y\}$ requires calculating the message $\{X, Y\} \Rightarrow \{W, X\}$, which again takes the form of a potential over the clique intersection, here $\{X\}$. The message is calculated in Table 5.4b. Table 5.4c uses it to update the potential for $\{W, X\}$. Table 5.4d shows the potentials after e_1 has been propagated both up and down the chain. At this point, the junction tree is ready to receive and propagate evidence in the form of values for W, Y, or Z (for example, e_2 that $Y = 0$; see Exercise 5.4). Normalizing the potentials in every node gives Table 5.4e.

5.2.2 Propagation in Trees

The approach of updating belief in chains can be easily generalized for updating belief about a set of variables when the undirected graph for their relationships is a *tree*. A tree is a *singly connected* graph—there is never more than one path from any variable to any other. A chain is a particularly simple tree. (In Sect. 5.3.2, we will address the general question of moving from an acyclic digraph, which represents a probability distribution as a *directed* graph, to an *undirected* graph that will serve as the vehicle for updating belief. As we saw in Chap. 4, an acyclic digraph that is singly connected does not necessarily give rise to a singly connected undirected graph for computing purposes.)

A singly connected undirected graph supports a junction tree representation for updating, similar to the one the chain depicted in Fig. 5.2. An example is shown in Fig. 5.3, for a joint distribution that factors as follows:

$$P(U, V, X, Y, Z) = P(Z \mid X)\, P(Y \mid X)\, P(X \mid V)\, P(U \mid V)\, P(V)\,.$$

Suppose new information arrived about X, in the form of $p_{new}(x)$. This information would be propagated through the rest of the network in the update-and-marginalize steps in Sect. 5.2.1, but now in three directions: Down and left to Y, down and right to Z, and up to V, and from V to U in the same manner.

When Kim and Pearl (1983) first defined the belief-updating algorithm, they restricted it to a kind of graph called a *polytree*. A polytree is basically a directed graph that is a tree after the direction of the edges is dropped. Figure 5.4 provides an example. Figure 5.4 gives the junction tree for this graph. Note that two of the clique nodes, $\{T, W, U\}$ and $\{W, X, Y\}$, have size three, so this graph has a treewidth of three. The updating algorithm described for chains in Sect. 5.2.1 works in essentially the same way. All the clique intersections are still single variables, but the projections into cliques and marginalizations down from them now sometimes involve more than two variables.

Table 5.3 Updating probabilities down a chain

$$
\begin{array}{c|c|c|}
\multicolumn{3}{c}{p_{old}(w,x)} \\
\multicolumn{3}{c}{\text{W}} \\
\text{X} & 1 & 0 \\ \hline
1 & .540 & .080 \\ \hline
0 & .060 & .320 \\ \hline
\end{array}
\quad
\begin{array}{c|c|}
\multicolumn{2}{c}{p_{old}(x)} \\
\multicolumn{2}{c}{} \\
 & \text{X} \\ \hline
1 & .620 \\ \hline
0 & .380 \\ \hline
\end{array}
\quad
\begin{array}{c|c|c|}
\multicolumn{3}{c}{p_{old}(x,y)} \\
\multicolumn{3}{c}{\text{X}} \\
\text{Y} & 1 & 0 \\ \hline
1 & .558 & .076 \\ \hline
0 & .062 & .304 \\ \hline
\end{array}
\quad
\begin{array}{c|c|}
\multicolumn{2}{c}{p_{old}(y)} \\
\multicolumn{2}{c}{} \\
 & \text{Y} \\ \hline
1 & .634 \\ \hline
0 & .366 \\ \hline
\end{array}
\quad
\begin{array}{c|c|c|}
\multicolumn{3}{c}{p_{old}(y,z)} \\
\multicolumn{3}{c}{\text{Y}} \\
\text{Z} & 1 & 0 \\ \hline
1 & .571 & .073 \\ \hline
0 & .063 & .293 \\ \hline
\end{array}
$$

a) Potential tables for initial joint and marginal probabilities.

$$
p_{new}(x,y) = p_{old}(x,y) \otimes \phi(x) = \begin{array}{|c|c|}\hline .558 & .076 \\ \hline .062 & .304 \\ \hline \end{array} \otimes \begin{array}{|c|c|}\hline 1 & 0 \\ \hline 1 & 0 \\ \hline \end{array} = \begin{array}{|c|c|}\hline .558 & 0 \\ \hline .062 & 0 \\ \hline \end{array}.
$$

b) Evidence $X = 1$ entered into $p_{old}(x,y)$ to produce $p_{new}(x,y)$.

$$
\begin{array}{c|c|c|}
\multicolumn{3}{c}{p_{old}(w,x)} \\
\multicolumn{3}{c}{\text{W}} \\
\text{X} & 1 & 0 \\ \hline
1 & .540 & .080 \\ \hline
0 & .060 & .320 \\ \hline
\end{array}
\quad
\begin{array}{c|c|}
\multicolumn{2}{c}{p_{old}(x)} \\
\multicolumn{2}{c}{} \\
 & \text{X} \\ \hline
1 & .620 \\ \hline
0 & .380 \\ \hline
\end{array}
\quad
\begin{array}{c|c|c|}
\multicolumn{3}{c}{p_{new}(x,y)} \\
\multicolumn{3}{c}{\text{X}} \\
\text{Y} & 1 & 0 \\ \hline
1 & .558 & 0 \\ \hline
0 & .062 & 0 \\ \hline
\end{array}
\quad
\begin{array}{c|c|}
\multicolumn{2}{c}{p_{old}(y)} \\
\multicolumn{2}{c}{} \\
 & \text{Y} \\ \hline
1 & .634 \\ \hline
0 & .366 \\ \hline
\end{array}
\quad
\begin{array}{c|c|c|}
\multicolumn{3}{c}{p_{old}(y,z)} \\
\multicolumn{3}{c}{\text{Y}} \\
\text{Z} & 1 & 0 \\ \hline
1 & .571 & .073 \\ \hline
0 & .063 & .293 \\ \hline
\end{array}
$$

c) Potential tables after having updated only $\{X, Y\}$.

$$
\{X,Y\} \Rightarrow \{Y,Z\} = p_{new}(y) \oslash p_{old}(y) = \begin{array}{|c|}\hline .558 \\ \hline .062 \\ \hline \end{array} \oslash \begin{array}{|c|}\hline .634 \\ \hline .336 \\ \hline \end{array} = \begin{array}{|c|}\hline .558/.634 \\ \hline .062/.336 \\ \hline \end{array} = \begin{array}{|c|}\hline .880 \\ \hline .185 \\ \hline \end{array}.
$$

d) Calculating the message from $\{X, Y\}$ to $\{Y, Z\}$.

$$
\begin{aligned}
p_{new}(y,z) &= p_{old}(y,z) \otimes (\{X,Y\} \Rightarrow \{Y,Z\}) \\
&= \begin{array}{|c|c|}\hline .571 & .073 \\ \hline .063 & .293 \\ \hline \end{array} \otimes \begin{array}{|c|c|}\hline .880 & .185 \\ \hline .880 & .185 \\ \hline \end{array} = \begin{array}{|c|c|}\hline .592 & .014 \\ \hline .055 & .054 \\ \hline \end{array}.
\end{aligned}
$$

e) Passing the message from $\{X, Y\}$ to $\{Y, Z\}$.

$$
\begin{array}{c|c|c|}
\multicolumn{3}{c}{p_{old}(w,x)} \\
\multicolumn{3}{c}{\text{W}} \\
\text{X} & 1 & 0 \\ \hline
1 & .540 & .080 \\ \hline
0 & .060 & .320 \\ \hline
\end{array}
\quad
\begin{array}{c|c|}
\multicolumn{2}{c}{p_{old}(x)} \\
\multicolumn{2}{c}{} \\
 & \text{X} \\ \hline
1 & .620 \\ \hline
0 & .380 \\ \hline
\end{array}
\quad
\begin{array}{c|c|c|}
\multicolumn{3}{c}{p_{new}(x,y)} \\
\multicolumn{3}{c}{\text{X}} \\
\text{Y} & 1 & 0 \\ \hline
1 & .558 & 0 \\ \hline
0 & .062 & 0 \\ \hline
\end{array}
\quad
\begin{array}{c|c|}
\multicolumn{2}{c}{p_{new}(y)} \\
\multicolumn{2}{c}{} \\
 & \text{Y} \\ \hline
1 & .558 \\ \hline
0 & .062 \\ \hline
\end{array}
\quad
\begin{array}{c|c|c|}
\multicolumn{3}{c}{p_{new}(y,z)} \\
\multicolumn{3}{c}{\text{Y}} \\
\text{Z} & 1 & 0 \\ \hline
1 & .592 & .014 \\ \hline
0 & .055 & .054 \\ \hline
\end{array}
$$

f) Status after e_1 has been propagated down the chain from $\{X, Y\}$.

Table 5.4 Updating probabilities up a chain

$p_{old}\,(w,x)$

	W	
X	1	0
1	.540	.080
0	.060	.320

$p_{old}\,(x)$

	X
1	.620
0	.380

$p_{new}\,(x,y)$

	X	
Y	1	0
1	.558	0
0	.062	0

$p_{new}\,(y)$

	Y
1	.558
0	.062

$p_{new}\,(y,z)$

	Y	
Z	1	0
1	.592	.014
0	.055	.054

a) Status after e_1 has been propagated down the chain from $\{X,Y\}$.

$$\{X,Y\} \Rightarrow \{W,X\} = p_{new}\,(x) \oslash p_{old}\,(x) = \begin{array}{|c|}\hline .620 \\ \hline 0 \\ \hline \end{array} \oslash \begin{array}{|c|}\hline .620 \\ \hline .380 \\ \hline \end{array} = \begin{array}{|c|}\hline 1 \\ \hline 0 \\ \hline \end{array}.$$

b) Calculating the message from $\{X,Y\}$ to $\{W,X\}$.

$$p_{new}\,(w,x) = p_{old}\,(w,x) \otimes (\{X,Y\} \Rightarrow \{W,X\})$$
$$= \begin{array}{|c|c|}\hline .540 & .080 \\ \hline .060 & .320 \\ \hline \end{array} \otimes \begin{array}{|c|c|}\hline 1 & 1 \\ \hline 0 & 0 \\ \hline \end{array} = \begin{array}{|c|c|}\hline .540 & .080 \\ \hline 0 & 0 \\ \hline \end{array}.$$

c) Updating the potential for $\{W,X\}$.

$p_{new}\,(w,x)$

	W	
X	1	0
1	.540	.080
0	0	0

$p_{new}\,(x)$

	X
1	.620
0	0

$p_{new}\,(x,y)$

	X	
Y	1	0
1	.558	0
0	.062	0

$p_{new}\,(y)$

	Y
1	.558
0	.062

$p_{new}\,(y,z)$

	Y	
Z	1	0
1	.592	.014
0	.055	.054

d) Status after propagating e_1 both up and down the chain.

$p_{new}\,(w,x)$

	W	
X	1	0
1	.871	.120
0	0	0

$p_{new}\,(x)$

	X
1	1
0	0

$p_{new}\,(x,y)$

	X	
Y	1	0
1	.9	0
0	.1	0

$p_{new}\,(y)$

	Y
1	.9
0	.1

$p_{new}\,(y,z)$

	Y	
Z	1	0
1	.810	.020
0	.090	.080

e) Status after normalizing.

This leads us to an important property of the junction tree. Look at the three junction tree examples we have seen so far, Figs. 5.2, 5.3, and 5.4. In each case pick out the nodes (both clique and intersection) in the junction tree that contain X. Note that they are all connected. This is true for each variable in the model. This so-called running intersection property is critically important because every time we pass a message from a node that contains the variable X to one that does not, we marginalize out the variable X. This means that we will not pass messages about X through a section of the tree that does not contain X. This property will be discussed further in Sect. 5.3.

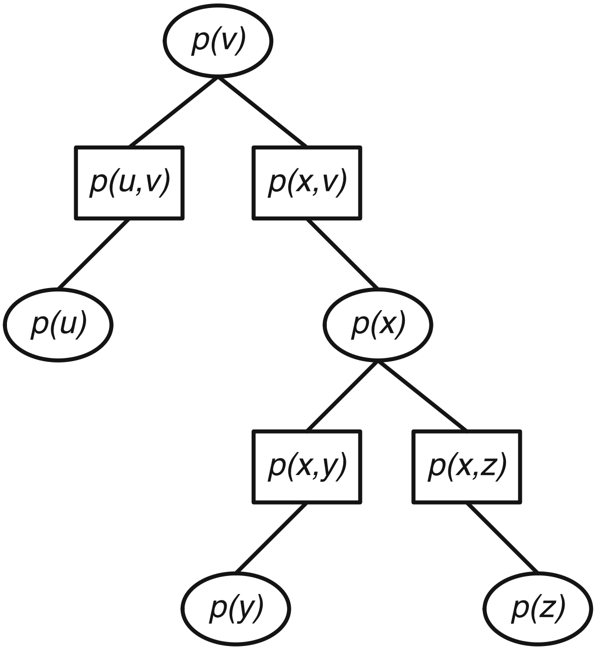


Fig. 5.3 A junction tree corresponding to a singly connected graph. As with Fig. 5.2 the intersection nodes that attach to only one clique node, $p(u)$,$p(y)$ and $p(z)$ are usually dropped for computational purposes
Reprinted with permission from ETS.

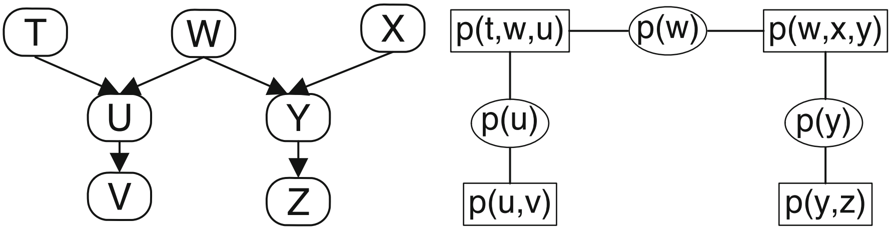


Fig. 5.4 A polytree and its corresponding junction tree
Reprinted with permission from ETS.

5.2.3 Virtual Evidence

Before we move to more complicated graphs, it is worth mentioning what to do when the evidence is uncertain. This situation arises all the time when human (or even computer) raters make a judgment about a performance. It is well known that raters are not perfect. Suppose that we have information that the raters on a particular assessment give the correct rating say 90 % of the time. How do we enter this information into the Bayes net?

The key to understanding how to incorporate this uncertain evidence in our model is to think about the expert's rating as another, implicit, node in our graphical model. If in Fig. 5.1, X represents the examinee's proficiency variable and Y represents the true quality of the performance, we could add Node Z to represent a rater's noisy report about the quality of the performance. Let us suppose that if the performance is truly adequate, or $Y = \texttt{yes}$, the probability of the rater correctly judging it as adequate, or $Z = \texttt{yes}$, is $p\,(z_1|y_1)$. If the performance is truly inadequate, or $Y = \texttt{no}$, the probability of the rater erroneously judging it as adequate, or $Z = \texttt{yes}$, is $p\,(z_1|y_2)$.

Now Node Z is a child of Node Y. This produces a simple chain $X \rightarrow Y \rightarrow Z$. The junction tree would have two clique nodes, $\{X, Y\}$ and $\{Y, Z\}$, with a single intersection node, $\{Y\}$. We can use the conditional probabilities for z given y and the marginal probability of y, $p\,(y)$ to construct the initial potential table for $\{Y, Z\}$:

$$\begin{array}{|c|c|}\hline p\,(z_1, y_1) & p\,(z_2, y_1) \\ \hline p\,(z_1, y_2) & p\,(z_2, y_2) \\ \hline \end{array} = \begin{array}{|c|c|}\hline p\,(z_1 \mid y_1)\,p\,(y_1) & p\,(z_2 \mid y_1)\,p\,(y_1) \\ \hline p\,(z_1 \mid y_2)\,p\,(y_2) & p\,(z_2 \mid y_2)\,p\,(y_2) \\ \hline \end{array}$$

A judgment from the rater takes the form of $Z = z_1$ or $Z = z_2$. Let us use $Z = z_1$ as an illustration, and examine the message $\{Y, Z\} \Rightarrow \{X, Y\}$. The denominator $p_{old}(y)$ is the original marginal distribution for Y, $[p(y_1), p(y_2)]$. The numerator $p_{new}(y)$ will have the the column from the $\{Y, Z\}$ potential that corresponds to observing $Z = 1$, namely $[p(z_1|y_1)p(y_1), p(z_1|y_2)p(y_2)]$. Thus the message to $\{X, Y\}$ will be

$$\left[\frac{p(z_1|y_1)p(y_1)}{p(y_1)}, \frac{p(z_1|y_2)p(y_2)}{p(y_2)}\right] = \left[p(z_1|y_1), p(z_1|y_2)\right].$$

This message is precisely the likelihood of the observed evidence under the various states of Node Y. These likelihoods form a potential that multiplies the potential in Node $\{X, Y\}$ to make the new distribution.

Pearl (1988) calls uncertain evidence, like the judgment from the rater, *virtual evidence* and notes that statements of certainty about the evidence should usually be treated as likelihood information. By the reasoning illustrated above, we can add virtual evidence directly into the Bayes net without needing to explicitly add the node representing the statement (Node Z) into the model. Form the likelihood vector, $[p(z_1|y_1), p(z_1|y_2)]$, as a potential over the variables $\{Y\}$. This can be combined with any clique node in the junction tree which contains Y. The belief-updating algorithm can then propagate this information to the other tables in the junction tree just as before.

This calculation extends smoothly to cases with more than one rater. We assume that given the performance, all of the ratings are independent. Therefore, we can simply multiply the likelihoods for the individual ratings together to get the combined likelihood for the whole set of ratings, then update the

Table 5.5 Updating with virtual evidence

	yes	no	P(X)
expert	$.8 \times .5 = .40$	$.2 \times .5 = .10$	.5
novice	$.4 \times .5 = .20$	$.6 \times .5 = .30$	.5
P(Y)	.6	.4	1

a) Table for initial joint and marginal probabilities.

	yes	no	P(X)
expert	$.4 \times .9 = .36$	$.1 \times .05 = .005$	.365
novice	$.2 \times .9 = .18$	$.3 \times .05 = .015$	.195
P(Y)	.54	.02	.56

b) Information (likelihood) arrives for Y: .9 for yes and .05 for no.

	yes	no	P(X)
expert	.643	.009	.652
novice	.321	.027	.348
P(Y)	.964	.036	1

c) Normalize by dividing by the grand total, .56.

proficiency variable. If the ratings arrive one at a time, then we can apply the belief-updating algorithm for virtual evidence sequentially, once for the first rater, again for the second rater, and so on.

Example 5.4 (Dental Hygienist Assessment, Example 5.1, Continued). *For a numerical example, suppose performances are evaluated by a human judge who has the following characteristics. If the patient history is truly adequate, the judge marks it* yes *90 % of the time. If the history is not adequate, then the judge marks it* no *95 % of the time. (Looking ahead to Chap. 7, these numbers are called the* sensitivity *and* specificity *of the rater.) A rater marking a performance as* yes *produces a virtual evidence likelihood vector of* $[.9, .05]$*. This is a potential over* Y*, which will be combined with the potential representing the prior distribution over* X *and* Y *given in the first panel of Table 5.5.*

The first panel of Table 5.5 is identical to the first panel of Table 5.2, as the prior distributions are identical. The difference comes with the arrival of the evidence in the second panel. In the middle panel the virtual evidence is combined with the prior distributions to produce the posterior potential over $\{X, Y\}$*. In order to interpret this potential as probability distributions we* *need to normalize it. The sum of the entries in middle panel is* .56*, the prior probability of this particular bit of evidence (judge gives a rating of* `yes`*). The final panel shows the normalized distribution.*

One way to think about how virtual evidence works is that the junction tree is broken into two parts. The virtual evidence is the message passed from the $\{Y, Z\}$ part of the tree to the $\{X, Y\}$ part of the tree. We could actually make the split at any intersection node of the junction tree. Section 5.4 shows an application of this idea that is important in educational assessment.

5.3 Belief Updating in Multiply Connected Graphs

As the graphical structure becomes more complex, so does the belief-updating algorithm. However, we already saw that if the graph is shaped like a tree, the belief-updating algorithm was just a simple extension of the algorithm for chains. Thus, we can use the algorithm in any graphical structure if only we could transform it into a tree: in this case, a tree where the nodes represent groups of variables.

The tree we are looking for is the tree of cliques, a more general version of the junction tree we have seen in chains and trees. By transforming the graph into a tree whose nodes represent cliques in our original graph, we can deal with much more complex graphical structures. There are a number of variants on this basic algorithm (See Sect. 5.6). This section describes one approach to doing this, and illustrates the procedures with a simple numerical example.

5.3.1 Updating in the Presence of Loops

A multiply connected graph is a graph with at least one loop, for instance, where there is more than one chain (undirected path) from one variable to another. Figure 5.5 shows an example of a loop. As a directed graph this graph is acyclic. However, if we drop the directions of the edges, the underlying undirected graph has a cycle, V to X to Y to U and back to V.

Had there been only the path from V to X to Y, we could have built a junction tree for pairs of variables that enables coherent updating across the three variables. Likewise had there been only a path from V to U to Y. But with two paths, trying to use both of these variable-level structures to update from information on Y does not generally provide the correct posterior for U, V, or X.[1] The "competing explanations" phenomenon described in Sect. 3.3.3 (Example 3.9) is a clear example of the failure. Seeing an incorrect response

[1] Weiss (2000) attempts to characterize situations in which the algorithm will converge and produce proper marginal distributions. This seems to depend on both the network and the evidence (Murphy et al. 1999). Weiss (2000) does note that the loopy-propagation algorithm will always produce the proper Maximum A Posteriori (MAP) estimate, even if the margins are incorrect.

means that at least one of the two required skills is probably lacking, and learning that one was present or missing would influence our belief about the other. This is a strong finding about the joint distribution of the two skills that cannot be captured by updating belief about each of them them independently.

Lauritzen and Spiegelhalter (1988) broke beyond the barrier of single connectedness. The essential idea is this: One can carry out the coherent propagation of information by passing messages between the cliques. It is common practice to express this algorithm as message passing in a *tree of cliques*. The *junction tree*, introduced in the previous section, takes the tree of cliques and adds nodes corresponding to the intersections between the cliques. The junction-tree algorithm described here is a variant of the Lauritzen–Spiegelhalter algorithm.

As a look ahead, the cliques—maximal sets of connected nodes—for the loopy digraph in Fig. 5.5 are $\{U, V, X\}$, $\{U, Y, X\}$, and $\{X, Z\}$, and the junction tree of cliques and clique intersections is shown as Figure 5.6. The key implication is that if information arrives about Y, its implications for our belief about X must deal with U and V jointly. Coherent reasoning around the loop is achieved, although at the cost of working with groups of variables rather than single variables.

Every acyclic digraph can be represented as a tree of cliques (or junction tree). A key feature is its *treewidth*, or the size of the largest clique. The brute force algorithm requires constructing a giant table with all of the variables in the model, but the biggest table in the message-passing algorithm is the size of the biggest clique, the treewidth. If the topology of the graph is favorable, meaning that the treewidth is small, calculation can be feasible for even networks with very many variables. The muscle and nerve inference network (MUNIN) Bayes net (Andreassen et al. 1987), for example, comprises about a thousand variables, yet can support real-time calculation because of its sparse treewidth. The sparse treewidth is achieved largely through conditional independence assumptions from the substantive domain of the application. For MUNIN, the domain is diagnosing neuromuscular diseases. Medical knowledge and characteristics of test procedures imply independence of certain tests given physical states, of symptoms given syndromes, and of syndromes given disease states. If the topology of a graph is not favorable, large cliques cannot be avoided; treewidth, and therefore computational burden, increases. At worst, all variables are in a single clique and computation through the Lauritzen–Spiegelhalter algorithm reduces to the brute force calculation with all combinations of values of all variables.

5.3.2 Constructing a Junction Tree

This section walks through the steps from a joint distribution of variables to a junction tree for efficient calculation, in the context of a simple example: an

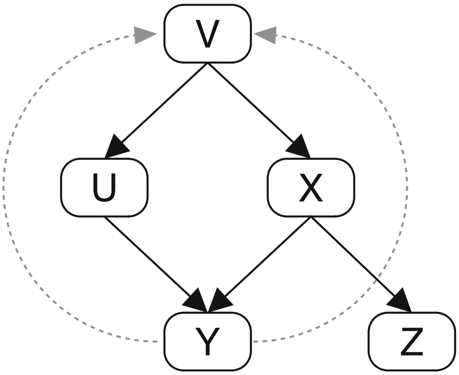


Fig. 5.5 A loop in a multiply connected graph
Reprinted from Almond et al. (2006a) with permission from ETS.

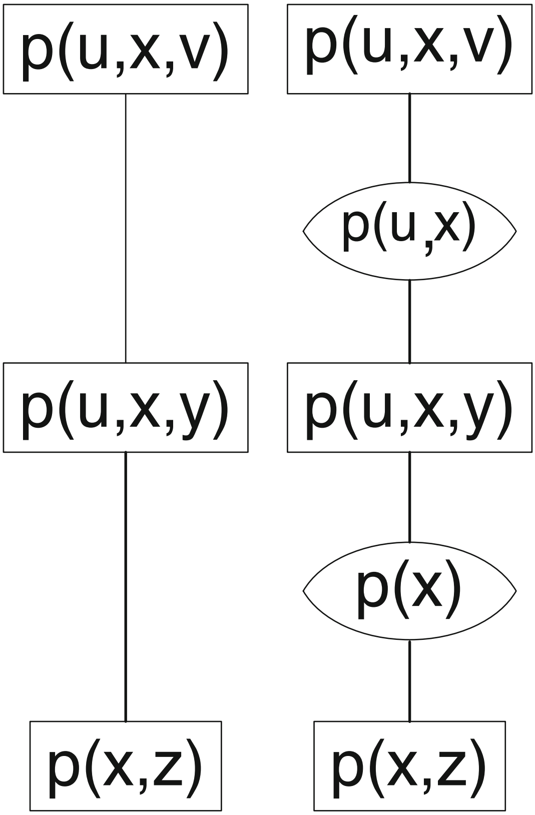


Fig. 5.6 The tree of cliques and junction tree for Figure 5.5. Clique nodes are represented with rectangles and intersection nodes are represented with ovals
Reprinted with permission from ETS.

adaptation of a medical diagnosis example from Jensen (1996) to the context of cognitive diagnosis.

Example 5.5 (Two Skills and Two Tasks). *We are interested in learning the proficiency that a student Pat has with literary terms. Skill A is literary vocabulary, which we suppose can take only the two values* `High` *(*`H`*) and* `Low` *(*`L`*). Skill B is the ability to infer the meaning of such words in context, which again we suppose can take the values of* `High` *(*`H`*) and* `Low` *(*`L`*). Denote these proficiency variables by θ_A and θ_B.*

There are two sources of evidence, Task 1 and Task 2, both of which yield a response that is evaluated as `Right` *(1) or* `Wrong` *(0). Denote these observable variables by X_1 and X_2. Task 1 asks the meaning of "simile" and provides a short essay about how poets use similes and metaphors to open readers' eyes to unexpected connections. A student is more likely to answer Task 1 correctly if either she already is familiar with the word, or can infer its meaning from the passage. Task 2 asks the meaning of "anaphora" with a sparse text illustrating its use, so it is mainly prior familiarity with the term that will be likely to produce a correct response. Figure 5.7 shows the graphical structure of this problem.*

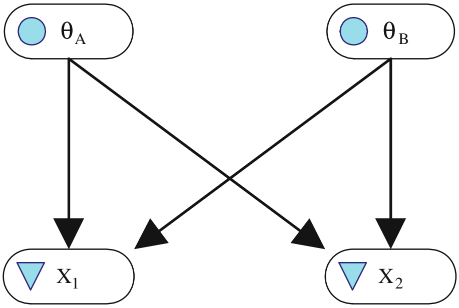


Fig. 5.7 Acyclic digraph for two-skill example (Example 5.5) Reprinted from Almond et al. (2006a) with permission from ETS.

Using this information about Skills, Tasks, and their relationships, we will build a Bayes net, a junction tree, and a computational representation in the form of an interconnected set of potential tables. Then we will observe Pat's response to Task 1, and use this machinery to update our beliefs about Skill A and Skill B and to predict whether we will see a correct response to Task 2 as well.

The following sections describe six steps to building a computing representation for problems such as these. The steps address these topics:

1. Recursive representation of the joint distribution of variables.

2. Acyclic digraph representation of the probability distribution.
3. Representation as a “moralized” and triangulated undirected graph.
4. Determination of cliques and clique intersections.
5. Junction tree representation.
6. Potential tables.

At this point, the potential tables are ready to support calculations that propagate the effect of new evidence.

Recursive Representation of the Joint Distribution

By repeatedly applying the definition of conditional probability, it is always possible to write the joint probability distribution of a number of variables as the product of conditional probabilities, each variable in the list conditional on those earlier in the list. That is,

$$\begin{aligned}&\mathrm{P}\left(A_n, A_{n-1}, \ldots, A_2, A_1\right) \\ &\quad= \mathrm{P}\left(A_n|A_{n-1}, ..., A_2, A_1\right) \times \mathrm{P}\left(A_{n-1}|A_{n-2}, \ldots, A_2, A_1\right) \times \cdots \times \\ &\qquad \mathrm{P}\left(A_2|A_1\right) \times \mathrm{P}\left(A_1\right) \\ &\quad= \prod_k \mathrm{P}\left(A_k|A_{k-1}, \ldots, A_1\right),\end{aligned} \tag{5.5}$$

where the final term is understood to be simply $\mathrm{P}(A_1)$. This is a *recursive representation* of the distribution. Such a representation holds for any ordering of the variables, but some orderings are more useful than others; a variable A_k may be conditionally independent of some of the variables with lower indices, and they drop out of its conditioning list. As in Chap. 4, we call those that remain its *parents* and denote them by $\mathrm{pa}(A_k)$. Thus,

$$\mathrm{P}\left(A_n, A_{n-1}, ..., A_2, A_1\right) = \prod_k \mathrm{P}\left(A_k|\,\mathrm{pa}\left(A_k\right)\right). \tag{5.6}$$

We have already seen this equation in Sect. 4.2.1 as Eq. 4.1. There the parents refer to the parents in the graphical structure. In general, exploiting the conditional independence relationships modeled in the graph lead to an efficient recursive representation. Examples of such relationships are effects that are conditionally independent given causes, observations that are conditionally independent given parameters, and current events that are conditionally independent given past events. In educational assessment, we typically model observable variables from different tasks as conditionally independent given proficiency variables.

The joint distribution we are interested in for the running example in this section is $\mathrm{P}\left(\theta_A, \theta_B, X_1, X_2\right)$. The description of the setup asserts that the observable variables X_1 and X_2 are conditionally independent given the proficiency variables θ_A and θ_B, and that the two proficiency variables are independent with respect to one another in the absence of any observations. This suggests the following recursive expression of the distribution:

$$\mathrm{P}\left(\theta_A, \theta_B, X_1, X_2\right) = \mathrm{P}\left(X_1|\theta_A, \theta_B, X_2\right) \mathrm{P}\left(X_2|\theta_A, \theta_B\right) \mathrm{P}\left(\theta_A|\theta_B\right) \mathrm{P}\left(\theta_B\right)\ .$$

Note that the order of the nodes in the recursive decomposition (Eq. 5.5) follow the graph in Figure 5.7 in the sense that the parents of each node in the graph are always ahead of the child node in the order. Exploiting the conditional independence relationships in the graph yields the following factorization:

$$\mathrm{P}\left(\theta_A, \theta_B, X_1, X_2\right) = \mathrm{P}\left(X_1 | \theta_A, \theta_B\right) \mathrm{P}\left(X_2 | \theta_A, \theta_B\right) \mathrm{P}\left(\theta_A\right) \mathrm{P}\left(\theta_B\right). \tag{5.7}$$

Example 5.6 (Numbers for Example 5.5). *In order to illustrate calculations later in the section, we propose numerical values for the probability distributions in the recursive representation. Part II addresses the issue of where these numbers come from, but it suffices to say at this point that the structures for the probability distributions and initial numerical values can be provided by experts, and both the structures and the numerical values can be refined in light of data that bear on the problem as an exercise in Bayesian estimation. For now we will work with the values in Table 5.6.*

Table 5.6 Probabilities for Example 5.5

$$\theta_A\colon \begin{cases} \mathrm{P}\left(\theta_A = \mathtt{H}\right) = .11 \\ \mathrm{P}\left(\theta_A = \mathtt{L}\right) = .89 \end{cases}$$

$$\theta_B\colon \begin{cases} \mathrm{P}\left(\theta_B = \mathtt{H}\right) = .11 \\ \mathrm{P}\left(\theta_B = \mathtt{L}\right) = .89 \end{cases}$$

$$X_1\colon \begin{cases} \mathrm{P}\left(X_1 = 1 \mid \theta_A = \mathtt{H}, \theta_B = \mathtt{H}\right) = .99;\ \mathrm{P}\left(X_1 = 0 \mid \theta_A = \mathtt{H}, \theta_B = \mathtt{H}\right) = .01 \\ \mathrm{P}\left(X_1 = 1 \mid \theta_A = \mathtt{H}, \theta_B = \mathtt{L}\right) = .90;\ \mathrm{P}\left(X_1 = 0 \mid \theta_A = \mathtt{H}, \theta_B = \mathtt{L}\right) = .10 \\ \mathrm{P}\left(X_1 = 1 \mid \theta_A = \mathtt{L}, \theta_B = \mathtt{H}\right) = .90;\ \mathrm{P}\left(X_1 = 0 \mid \theta_A = \mathtt{L}, \theta_B = \mathtt{H}\right) = .10 \\ \mathrm{P}\left(X_1 = 1 \mid \theta_A = \mathtt{L}, \theta_B = \mathtt{L}\right) = .01;\ \mathrm{P}\left(X_1 = 0 \mid \theta_A = \mathtt{L}, \theta_B = \mathtt{L}\right) = .99 \end{cases}$$

$$X_2\colon \begin{cases} \mathrm{P}\left(X_2 = 1 \mid \theta_A = \mathtt{H}, \theta_B = \mathtt{H}\right) = .99;\ \mathrm{P}\left(X_2 = 0 \mid \theta_A = \mathtt{H}, \theta_B = \mathtt{H}\right) = .01 \\ \mathrm{P}\left(X_2 = 1 \mid \theta_A = \mathtt{H}, \theta_B = \mathtt{L}\right) = .05;\ \mathrm{P}\left(X_2 = 0 \mid \theta_A = \mathtt{H}, \theta_B = \mathtt{L}\right) = .95 \\ \mathrm{P}\left(X_2 = 1 \mid \theta_A = \mathtt{L}, \theta_B = \mathtt{H}\right) = .90;\ \mathrm{P}\left(X_2 = 0 \mid \theta_A = \mathtt{L}, \theta_B = \mathtt{H}\right) = .10 \\ \mathrm{P}\left(X_2 = 1 \mid \theta_A = \mathtt{L}, \theta_B = \mathtt{L}\right) = .01;\ \mathrm{P}\left(X_2 = 0 \mid \theta_A = \mathtt{L}, \theta_B = \mathtt{L}\right) = .99 \end{cases}$$

We can make some observations in passing on the probative, or evidentiary, value of X_1 and X_1 for inferences about θ_A and θ_B. They are based on examining the conditional probability distributions of the observables given their proficiency model parents. Suppose for example it is observed that $X_1 = 1$. This could occur no matter what the values of θ_A and θ_B are, since there are nonzero conditional probabilities for $X_1 = 1$ at each combination. But this is a more likely occurrence under some combinations of θ_A and θ_B than others; for example, the conditional probability of $X_1 = 1$ is .99 if both skills are high and only .01 if both are low. The column of conditional probabilities for $X_1 = 1$ at each combination of its parents θ_A and θ_B is the likelihood function induced for these proficiency variables by a realized observation of $X_1 = 1$. In this case, the likelihood is .9 or above for all combinations with at least one skill at `High`, and only .01 when both skills are `Low`. Conversely, observing $X_1 = 0$ induces a relatively high likelihood for $\{\theta_A = \mathtt{Low}, \theta_B = \mathtt{Low}\}$ and a low likelihood for all the other skill combinations. To borrow a term from medical diagnosis, X_1 provides good differential diagnosis for distinguishing between "both skills at `Low`" and all other proficiency states. On the other hand, it has no value whatsoever for differential diagnosis between the states $\{\theta_A = \texttt{High}, \theta_B = \texttt{Low}\}$ and $\{\theta_A = \texttt{Low}, \theta_B = \texttt{High}\}$. By similar reasoning, X_2 has differential diagnostic value for distinguishing between states with $\theta_B = \texttt{High}$ from those with $\theta_B = \texttt{Low}$, but has little value for distinguishing states that differ with respect to θ_A.

Acyclic Digraph Representation

As discussed in Sect. 4.2, we can draw an acyclic digraph to represent the joint probability distribution of a set of variables straight from a recursive distribution. Each variable is a node in the digraph, and for each node A_k, there is a directed edge coming to it from each of its parents, or the variables in $\text{pa}(A_k)$. The digraph for our running example is shown as Fig. 5.7. The digraph depicts the structure of the joint probability distribution with regard to the conditional independence relationships that can be read from the recursive representation of Eq. 5.7.

Moralized and Triangulated Undirected Graph

Starting with the acyclic digraph, we drop the directions of the edges and add additional edges as necessary to meet two requirements. First, all the parents of every given child must be connected by edges (i.e., the parents of children must be "married"—hence the term *moralized graph*). Looking ahead, we need to assign the factor in the recursive representation corresponding to this child and its parents to one of the clique nodes in the junction tree. Connecting the parents ensures that the child and its parents will either be a clique or a subset of a clique in the moralized graph. It also ensures that any dependencies caused by the competing explanations phenomenon (Sect. 3.3.3) will be handled coherently, as they will be dealt with jointly in the potential table for a clique in the junction tree. Figure 5.8 is the moralized undirected graph for our example.

Note that it is the loops in the moralized graph that cause problems for computation. Even our simple example now has loops; for example, one can start a path at X_1, follow a connection to θ_A, then to X_2, then to θ_B, and finally return to X_1. This loop did not count as a cycle in the directed graph because it did not follow the direction of the arrows. However, it is the loops in the undirected moral graph that cause problems for the simple updating algorithm.

Another way to understand moralization is to think about the factorization hypergraph. Each child variable and its set of parents in the recursive representation corresponds to a hyperedge in the hypergraph. When we construct the 2-section of the factorization hypergraph, the parents of the child variable

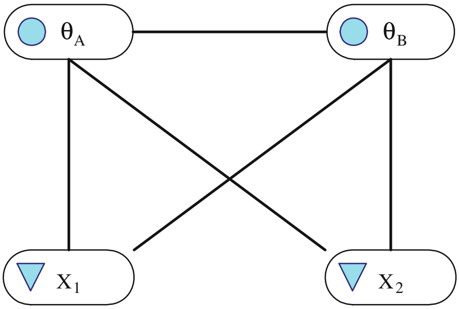


Fig. 5.8 Moralized undirected graph for two-skill example (Example 5.5) Reprinted from Almond et al. (2006a) with permission from ETS.

are in the same hyperedge and thus must be joined. Thus, the moralized graph is the 2-section of the factorization hypergraph.

In addition to being moralized, the graph must be *triangulated*; that is, any cycle (in the moral graph) consisting of four or more variables must have a *chord*, or "short cut." The graph in Fig. 5.8 is already triangulated; the edge between θ_A and θ_B induced by moralization is a chord. The leftmost graph in Fig. 5.9 is an example of a graph that is not triangulated. Although triangulation is not a problem in our simple example, it can be a big issue in larger problems.

Triangulation is necessary to express probability relationships in a way that lends itself to coherent propagation of information under Lauritzen–Spiegelhalter propagation and its variants. Without triangulation, the cliques may form a graph with cycles. For example, the cliques of the leftmost graph in Fig. 5.9 are $\{A_1, A_2\}$, $\{A_2, A_3\}$, $\{A_3, A_4\}$, $\{A_4, A_5\}$, and $\{A_5, A_1\}$. These make a cycle. Recall from Chap. 4 that an acyclic hypergraph was defined as one whose 2-section is triangulated. Triangulating the moral graph guarantees that a singly-connected clique representation can be constructed (Jensen 1988).

Although a given moral graph may not be triangulated, new edges can be "filled in" to make it so. There can be more than one way to fill in a graph to make it triangulated. Figure 5.9 shows two different ways to triangulate the untriangulated graph.

Finding the optimal triangulation for a graph is a hard problem. Different fill ins will create different sized cliques and hence will affect the treewidth of the final graph. Almond (1995) summarizes some of the heuristics which are commonly used to find the best triangulation.

Cliques and Clique Intersections

From the triangulated graph, we next determine *cliques,* or biggest subsets of variables that are all linked pairwise to one another. Cliques overlap, with

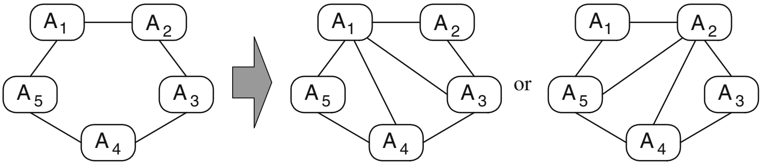


Fig. 5.9 Two ways to triangulate a graph with a loop
Reprinted from Almond et al. (2006a) with permission from ETS.

sets of overlapping variables called *clique intersections.* In the next step these cliques and clique intersections will become the nodes of the junction tree. Figure 5.10 shows the two cliques in our example, $\{\theta_A, \theta_B, X_1\}$ and $\{\theta_A, \theta_B, X_2\}$. The clique intersection is $\{\theta_A, \theta_B\}$.

Although there can be multiple ways to produce a triangulated graph from a given digraph, there is only one way to define cliques from a triangulated graph. There can be multiple ways to arrange them in a tree, but the computational cost is dominated by the size of the largest clique, that is, the treewidth. For this reason a triangulation that yields many small cliques is preferred to one that yields fewer but larger cliques. The HUGIN Bayes net compiler (Appendix A) offers several alternatives for triangulation, and on request reports the resulting cliques to the user. Strategies for increased computational efficiency include adding variables to break loops, redefining variables to collapse combinations of variable values that are not meaningfully distinct, and dropping associations when the consequences are benign.

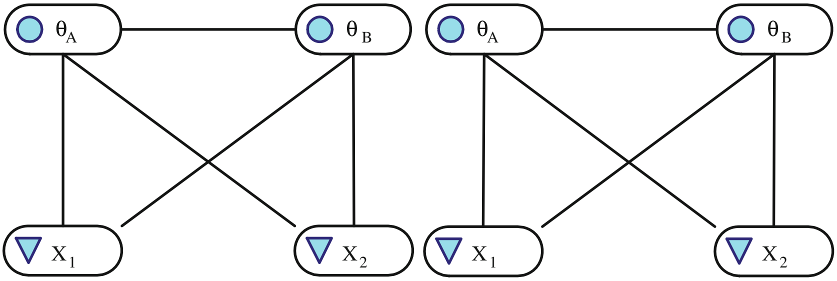


Fig. 5.10 Cliques for the two-skill example. The graph on the left shows the clique $\{\theta_A, \theta_B, X_1\}$; the graph on the right shows the clique $\{\theta_A, \theta_B, X_2\}$
Reprinted from Almond et al. (2006a) with permission from ETS.

Junction Tree Representation

Once we have the cliques and clique intersections, creating a junction tree is straightforward. Start with any clique node, and connect it to clique intersection nodes it contains. Taking each of these clique intersection nodes one at a time, connect it to clique nodes that also contain it and have not yet been addressed. Having done this with all the intersection nodes from the starting clique, take each of the cliques that were added to the junction tree one at a time and repeat the same process, in each case bringing in clique nodes that have not yet been addressed. When no more cliques can be connected through intersections in this way, either all the cliques are now connected or some remain unconnected. Variables in cliques that are unconnected are independent of the variables in the cliques that were connected thus far, and will be in separate junction trees. The connecting process begins anew, starting with one of the remaining cliques. When multiple junction trees result, evidence about variables associated with one tree has no impact on belief about variables associated with another tree, and they can be treated as separate problems. In all, this process ensures that the graph(s) so constructed will be trees and have the running intersection property.

Definition. **Running Intersection Property.** *A junction tree (or other tree containing sets of variables) has the* running intersection *property if for every variable the subgraph which contains that variable is connected. A tree with the running intersection property is called a* Markov Tree*.*

The key to the efficiency of the belief-updating algorithm is that we can marginalize out information that is no longer needed. The running intersection property tells us when information can be safely marginalized out. When we pass a message from a clique node which contains X to an intersection node which does not, we can be sure because of the running intersection property that there will be no nodes containing X on the other side of that intersection. The running intersection property is a key part of the proof of correctness of this algorithm (Shenoy and Shafer 1990; Almond 1995).

Figure 5.11 gives the junction tree for our example. Note that the two clique nodes correspond to the two tasks and the clique intersection corresponds to the proficiency variables. Thus the junction tree reflects our understanding of the conditional independence assumptions on which this model is based, namely that the two observable outcome variables are independent given the proficiency variable.

Potential Tables

As described in Sect. 5.2, each clique or clique intersection node in the junction tree has a *potential table*, which is related to the joint probability distribution of the nodes in that clique or intersection. Each of the factors in the recursive representation is expressed as a potential and allocated to one of these tables. As implied by the preceding steps, the variables in each factor will be together in some clique, but depending on the topology, a clique may contain the variables for multiple factors. The allocated tables are combined to make the initial potential associated with that node.

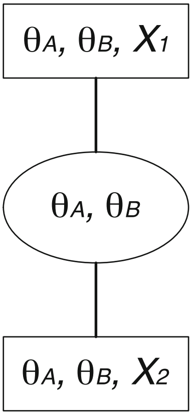


Fig. 5.11 Junction tree for the two-skill example
Reprinted from Almond et al. (2006a) with permission from ETS.

To initialize the junction tree version of the belief-updating algorithm requires that the table in each clique or clique intersection reflect the joint distribution of those variables (i.e., before adding evidence). The potential tables in Table 5.7 indicate the initial status of the network for Example 5.5; that is, before specific knowledge of a particular individual's observable variables states becomes known.

Table 5.7 Potential tables for the two-skill example

θ_A	θ_B	$\mathrm{P}(X_1 = 1)$	$\mathrm{P}(X_1 = 0)$
H	H	.012	.000
H	L	.088	.010
L	H	.088	.010
L	L	.008	.784

θ_A	θ_B	Probability
H	H	.012
H	L	.098
L	H	.098
L	L	.792

θ_A	θ_B	$\mathrm{P}(X_2 = 1)$	$\mathrm{P}(X_2 = 0)$
H	H	.011	.001
H	L	.005	.093
L	H	.088	.010
L	L	.008	.784

There are a number of algorithms for initializing the junction tree. The following example shows one.

Example 5.7 (Potential Tables for Example 5.5). *Constructing potential tables can be accomplished in a number of ways. Starting with a recursive representation of the probability distribution, it is easiest to work from root nodes, or the one or more variables that have no parents, and successively use their marginal distributions and the conditional distributions of their children, as they appear in cliques further down the list. For example, the potential table for the clique* $\{\theta_A, \theta_B, X_1\}$ *was calculated as follows:* θ_A *and* θ_B *are both root nodes. Because they are independent, their joint distribution is calculated by multiplying the prior probabilities of .11 for* `High` *and .89 for* `Low` *for all four* `High`/`Low` *combinations the two variables can take:*

$$\begin{aligned}
\mathrm{P}\left(\theta_A = \mathtt{H}, \theta_B = \mathtt{H}\right) &= \mathrm{P}\left(\theta_A = \mathtt{H}\right)\mathrm{P}\left(\theta_B = \mathtt{H}\right) = .11 \times .11 = .012\\
\mathrm{P}\left(\theta_A = \mathtt{H}, \theta_B = \mathtt{L}\right) &= \mathrm{P}\left(\theta_A = \mathtt{H}\right)\mathrm{P}\left(\theta_B = \mathtt{L}\right) = .11 \times .89 = .098\\
\mathrm{P}\left(\theta_A = \mathtt{L}, \theta_B = \mathtt{H}\right) &= \mathrm{P}\left(\theta_A = \mathtt{L}\right)\mathrm{P}\left(\theta_B = \mathtt{H}\right) = .89 \times .11 = .098\\
\mathrm{P}\left(\theta_A = \mathtt{L}, \theta_B = \mathtt{L}\right) &= \mathrm{P}\left(\theta_A = \mathtt{L}\right)\mathrm{P}\left(\theta_B = \mathtt{L}\right) = .89 \times .89 = .792.
\end{aligned}$$

X_1 *is the child of* θ_A *and* θ_B*. Its conditional for each combination of values of its parents was given in the recursive definition of the distribution. For example,* $\mathrm{P}\left(X_1 = 1 \mid \theta_A = H, \theta_B = L\right) = .90$*, so*

$$\begin{aligned}
&\mathrm{P}\left(X_1 = 1, \theta_A = \mathtt{H}, \theta_B = \mathtt{L}\right)\\
&\quad = \mathrm{P}\left(X_1 = 1 \mid \theta_A = \mathtt{H}, \theta_B = \mathtt{L}\right)\mathrm{P}\left(\theta_A = \mathtt{H}, \theta_B = \mathtt{L}\right)\\
&\quad = .9 \times .098 = .010.
\end{aligned}$$

In a similar manner, the joint probability for every combination of $\{\theta_A, \theta_B, X_1\}$ *values can be calculated, and becomes the entry for that combination in the potential table.*

Once the potential table for a clique has been calculated, the table for any clique intersection connecting it to another clique is obtained by marginalizing with respect to whatever variables are in the intersection. In this simple example, the only clique intersection is $\{\theta_A, \theta_B\}$*. It can be obtained by collapsing the* $\{\theta_A, \theta_B, X_1\}$ *potential table over* X_1*, or* $p\left(\{\theta_A, \theta_B, X_1\}\right)\Downarrow_{X_1}$*. We already know the result since we obtained the joint* $\{\theta_A, \theta_B\}$ *along the way to building the table for* $\{\theta_A, \theta_B, X_1\}$*, but this does not generally happen.*

Having started from root nodes, moving from a clique intersection to a successive clique means that the variables in the clique intersection come in with their joint marginal distribution. The new variables will have conditional distributions given those in the clique intersection and possibly on other variables in the clique. One computes joint distributions using the rule of marginal times conditional distribution as above, variable by variable, in the order they appear in the recursive representation. The second clique in our two-skill example is $\{\theta_A, \theta_B, X_2\}$*. We already have the joint distribution for* θ_A *and* θ_B*. The conditional distribution of* X_2 *given* θ_A *and* θ_B *is given* *in the recursive representation of the full joint distribution. For example,* $\mathrm{P}(X_2 = 0 \mid \theta_A = \mathtt{L}, \theta_B = \mathtt{L}) = .99$*, so*

$$\begin{aligned} &\mathrm{P}(X_2 = 0, \theta_A = \mathtt{L}, \theta_B = \mathtt{L}) \\ &\;= \mathrm{P}(X_2 = 0 \mid \theta_A = \mathtt{L}, \theta_B = \mathtt{L})\,\mathrm{P}(\theta_A = \mathtt{L}, \theta_B = \mathtt{L}) \\ &\;= .99 \times .792 = .784. \end{aligned}$$

The six steps used to move from the digraph representation of the probability model to the junction tree are sometimes called *compiling* the Bayesian network (this term is used by many of the software packages described in Appendix A, even if those packages do not use exactly the algorithm described here). The digraph representation is most convenient for defining the conditional independence relationships that will define the shape of the graph and eliciting the conditional probabilities that will define the joint distribution (or will serve as priors for distributions to be learned from data, as in Part II). The junction tree is more convenient for answering queries, such as what is the probability distribution of θ_A after observing X_1. Just like compiling a computer program, compiling a digraph into a junction tree makes it ready to go to work for us.

5.3.3 Propagating Evidence Through a Junction Tree

To absorb new evidence about a single variable, first express the evidence as a potential. Pick a clique node containing the variable, and combine the potential in that node with the potential representing the evidence. Now apply the belief-updating algorithm described in Sect. 5.2.1 to propagate that information throughout the tree. The only additional wrinkle is that for clique intersections with more than one variable, we work with entries for the joint combinations of all the variables in the clique intersection, rather than just for the values of a single variable. When messages containing the evidence have reached all clique nodes in the tree, then the posterior distribution of any variable in the model given the evidence can be found by looking at the potential of any node in the junction tree that contains that variable. The single-connectedness and running intersection properties of the junction tree assure that coherent probabilities result.

Example 5.8 (Evidence Propagation in Example 5.5). *Suppose Pat answers Item 1 correctly; that is,* $X_1 = 1$*. How does this change our beliefs about the other variables?*

The process begins with the potential table for the clique $\{\theta_A, \theta_B, X_1\}$*. In the initial condition, we had a joint probability distribution for the variables in this clique, as shown in the top table of Table 5.7. We now know with certainty that* $X_1 = 1$*, so the column for* $X_1 = 0$ *is zeroed out (Table 5.8). The remaining columns (in this case there is just one of them) reflect the proportion of our revised belief about the values for the other variables in the clique,* θ_A *and* θ_B*.*

That first column in the top table of Table 5.7, or $[.012, .088, .088, .008]$, *is the updated potential in the clique intersection, or* $p_{new}\{\theta_A,\ \theta_B\}$. *The initial potential that was stored in* $\{\theta_A, \theta_B\}$ *was* $[.012, .098, .098, .792]$, *which is* $p_{old}\{\theta_A,\ \theta_B\}$. *This is the information we need to calculate the message* $\{\theta_A, \theta_B, X_1\} \Rightarrow \{\theta_A,\ \theta_B, X_2\}$.

	p_{old}	p_{new}	Message (p_{new}/p_{old})
$\theta_A = \mathtt{H}$, $\theta_B = \mathtt{H}$	.012	.012	1.00
$\theta_A = \mathtt{H}$, $\theta_B = \mathtt{L}$	.098	.088	.90
$\theta_A = \mathtt{L}$, $\theta_B = \mathtt{H}$	.098	.088	.90
$\theta_A = \mathtt{L}$, $\theta_B = \mathtt{L}$	.720	.008	.01

The values in the potential table for $\{\theta_A, \theta_B, X_2\}$ *are obtained with the belief updating operation as shown in Table 5.7. The resulting values are proportional to the new probabilities for the variables in this clique. The final panel of Table 5.8 shows the values after normalizing. The highest probabilities are for the combinations in which only one of the skills is High (a consequence of the low prior probabilities for the skills) and* X_2 *being right or wrong in accordance with whether it is* θ_A *or* θ_B, *that is at High.*

5.4 Application to Assessment

Chapter 2 described a general framework for assessments in terms of a number of models. Two of those models, the *proficiency model* and the *evidence model* have components that describe a probabilistic relationship among the variables. (That is not to say that the other models do not have a strong influence on the statistical properties of the assessment, rather that these are the two parts of the model that are conventionally modeled with direct statements of probability). In this book, we are interested in assessments for which those probabilistic parts of the model are expressed with Bayesian networks.

The *proficiency models* consist of proficiency variables—latent variables that characterize the knowledge, skills, or other attributes of students—and their distribution in a population of interest. The *measurement component* of evidence models addresses the relationship of these proficiency variables to observable variables—characterizations of the qualities of things students say, do, or make. The proficiency variables are of persistent interest in an assessment application. They are the level at which we conceive of the effects of learning and the locus of decisions about instruction. The observable variables that appear in evidence models are of interest mainly insofar as they provide information about proficiency variables.

The *total graphical model* for an assessment consists of a Bayesian network with all of the proficiency variables and all of the observable outcome variables

Table 5.8 Updating the potential tables for $\{\theta_A, \theta_B, X_2\}$

Before belief updating

θ_A	θ_B	$\mathrm{P}(X_2 = 1)$	$\mathrm{P}(X_2 = 0)$
H	H	.011	.001
H	L	.005	.093
L	H	.088	.010
L	L	.008	.784

Belief updating (i.e., multiplication by message)

θ_A	θ_B	$\mathrm{P}(X_2 = 1)$	$\mathrm{P}(X_2 = 0)$
H	H	.011 × 1.00	.001 × 1.00
H	L	.005 × .90	.093 × .90
L	H	.088 × .90	.010 × .90
L	L	.008 × .01	.784 × .01

After belief updating

θ_A	θ_B	$\mathrm{P}(X_2 = 1)$	$\mathrm{P}(X_2 = 0)$
H	H	.011	.001
H	L	.004	.084
L	H	.080	.009
L	L	.000	.008

After normalizing

θ_A	θ_B	$\mathrm{P}(X_2 = 1)$	$\mathrm{P}(X_2 = 0)$
H	H	.056	.005
H	L	.020	.426
L	H	.406	.046
L	L	.000	.041

from any task which could conceivably be given to a student. In an ongoing assessment system, hundreds or thousands of test items are developed and used, all providing information about the same small set of proficiency variables; the tasks (and the task model variables) are relevant during the time they are used, but they are retired and replaced continually.

In such an environment, it is obviously of benefit to be able to update proficiency models without having to build a single huge computational representation for every student using all items that have been and may ever be presented. Computation using a representation using only the tasks that are or may be used in the present test would be preferable. Such a scheme would be an example of what Breese et al. (1994) call *knowledge-based model construction*: dynamic assembly of computational or representational models from preassembled fragments, according to the unfolding nature of the problem. Computerized adaptive testing (CAT; Wainer et al. 2000) with item response theory (IRT) is a familiar example from psychometrics.

The key idea is that the Bayesian networks associated with the proficiency model and the measurement component of the evidence models are only fragments of the total graphical model for the assessment. These fragments can be stored in a library and assembled on demand. This is related to the object-oriented Bayesian network models of Koller and Pfeffer (1997) and Laskey and Mahoney (2000). This section expresses this modular measurement framework in the context of Bayes nets, defining proficiency model and evidence model Bayes nets fragments. The focus is on the implications for assessment design and analysis, with an eye toward adaptive applications.

5.4.1 Proficiency and Evidence Model Bayes Net Fragments

Two fundamental properties of psychometric models hold important implications for the recursive representation of the variables in psychometric models, and consequently for the Bayes nets and junction trees they induce. Let $(\theta_1, \ldots, \theta_m)$ be proficiency variables and $(X_1, \ldots, X_n)$ be observable variables. The two properties are as follows:

Property 5.1. Observable variables are always children, and never parents, of proficiency variables. Proficiency variables may be parents of other proficiency variables, and generally will be when there are multiple proficiency variables and they are associated in the examinee population. Proficiencies in Reading, Writing, Speaking, and Listening, for example, tend to be correlated, and we would probably model θ_R, θ_W, θ_S, and θ_L as either directly related among themselves or as children of a common language proficiency, say θ_{LP}.

Property 5.2. (**Local Independence**) *Observable variables from distinct tasks are conditionally independent, given proficiency variables.* Observable variables from the same complex performance or from the same multipart task may be parents of other observable variables in the same task, in addition to their student-model parents, as when an answer to a multiple-choice question is followed by "explain your answer." (In graphical terminology, the proficiency variables in the digraph d-separate the sets of observable variables from different tasks.)

This second property is often called *local independence*. Yen (1993) describes a number of situations in which local independence breaks down at the level of individual observables; that is, local *de*pendence occurs. Ratings of multiple aspects of the same complex performance and items whose response depends on previous responses are two examples. One of the most common testing situations with local dependence is a *testlet* (Wainer and Kiely 1987) in which several discrete items share a common stimulus, such as a reading passage or a graph. But if observable outcome variables that exhibit local dependence are placed within a single task and hence are scored by a single evidence model, then Property 5.2 is not violated at the level of tasks and we can use the method in this section. This gives Bayesian network models a expressive power to model situations that can be difficult to model with other methods.

The properties together imply first that the joint distribution of $(\theta_1, \ldots, \theta_m)$ and $(X_1, \ldots, X_n)$ can be written in terms corresponding to the joint distribution of θs and the conditional distribution of Xs from distinct tasks. We refer to the joint distribution of the θs as the *proficiency model Bayes net fragment,* or *PMF* for short; that is, $\mathrm{P}\left(\theta_1, \ldots, \theta_m\right)$. To allow for conditional dependence among observable variables from the same task, we introduce the index j for such interrelated groups of observables, denote the observables corresponding to Task j as $\left(X_{j1}, \ldots, X_{jn_j}\right)$, and refer to the conditional distribution of $(X_{k1}, \ldots, X_{kn_k})$ given its student-model parents as the *evidence model Bayes net fragment*, or *EMF*, for Task j; that is, $\mathrm{P}\left(X_{j1}, \ldots, X_{jn_j} \mid \Theta_j\right)$ where Θ_j is the subset of θs that are parents of any of the Xs in Task k. We refer to Θ_k as the *footprint* of Task j in the proficiency model. Thus, the joint distribution can be written as

$$\mathrm{P}\left(X_{j1}, \ldots, X_{jn_j}, \theta_1, \ldots, \theta_m\right) = \mathrm{P}\left(\theta_1, \ldots, \theta_m\right) \prod_j \mathrm{P}\left(X_{j1}, \ldots, X_{jn_j} \mid \Theta_j\right). \tag{5.8}$$

An acyclic digraph corresponding to a recursive representation in this form may have edges connecting θs to one another, and Xs have as parents only some set of θs and possibly other Xs from the same task. Figure 5.12 represents these relationships in terms of a Venn diagram for variables in the PMF and EMFs for two hypothetical tasks. Note that $\Theta_1 = \{\theta_2, \theta_3\}$ and $\Theta_2 = \{\theta_2, \theta_5, \theta_6\}$.

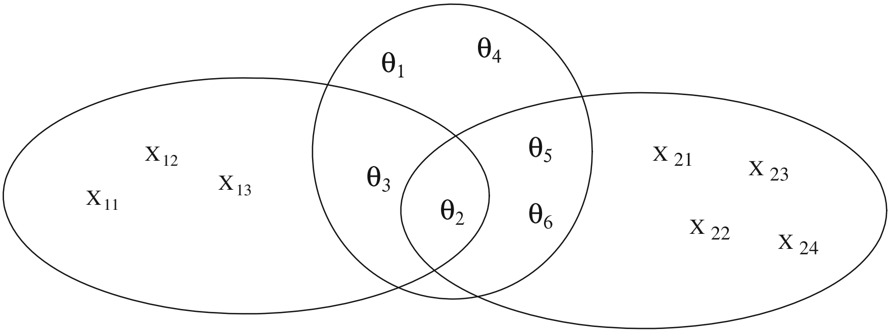


Fig. 5.12 Relationship among proficiency model and evidence model Bayes net fragments

Reprinted with permission from ETS.

5.4.2 Junction Trees for Fragments

Consider the total graphical model for an assessment consisting of a single task. It is formed by joining the PMF to the EMF by connecting the footprint variables. Now form the junction tree for this graph. We can arrange it so there will always be a node in the junction tree corresponding to the footprint (see Exercise 5.11), because moralization and triangulation will force edges among at least some of them, and we can add edges if we need to in order to connect them all. Split the junction tree at that node, producing two junction trees, one for the proficiency model and one for the evidence model. The virtual-evidence algorithm (Sect. 5.2.3) can then be used to pass information between the two trees. Almond et al. (1999) use this as the basis of an efficient algorithm for working with large assessments.

According to the Local Independence Property (Property 5.2), any impact on belief about the observables of one task on the observables of another task is mediated strictly through the influence on proficiency variables. This separation of tasks by the proficiency variables allows us to precompute junction trees and potential tables for PMFs and EMFs. These can be stored in a large pool and only the PMFs and EMFs relevant to a particular assessment situation (the form of the assessment the examinee actually sees) need be consulted to draw inferences.

More formally, the *Proficiency Model–Evidence Model* (PMEM) algorithm (Almond et al. 1999) requires special construction procedures for the PMF and EMF.

- **For the PMF**: The PMF is a Bayes net in itself, and potential tables could be built following the procedure described above in Sect. 5.3. But doing so from whatever structure of dependencies happen to reside in the recursive representation for $\mathrm{P}\left(\theta_1, \ldots, \theta_m\right)$ does not guarantee that the footprint of each EMFs will appear in a clique. In addition to edges added to moralize and triangulate the PMF, one must also add edges among proficiency variables to ensure that the footprint of each EMF that will be used to update the θs appears in at least one clique. After adding the additional edges joining the proficiency variable, the junction tree and potential tables for the PMF are then constructed as usual, starting with the triangulation step.
- **For the EMF for each Task** j: The essential element of an evidence model Bayes net fragment is a conditional distribution of observable variables given the proficiency variables in its footprint, namely $\mathrm{P}\left(X_{j1}, \ldots, X_{jn_j} | \Theta_k\right)$. The EMF does not have information about the marginal distribution of the proficiency variables, Θ_j. To produce the full joint distribution to initialize the EMF, say $\mathrm{P}^*\left(X_{j1}, \ldots, X_{jn_j}, \Theta_j, \right)$ assign independent uniform distributions for the θs in Θ_j. This implies the unit potential, **1**, in which every element is 1 (or, since only proportionality matters in potentials, equal values at any other nonnegative value), over

all possible combinations of values of the proficiency variables in Θ_j. Starting from the acyclic digraph for this augmented EMF, produce an undirected graph that first adds edges between all pairs of θs in Θ_j, as well as whatever additional edges are needed to moralize and triangulate the graph.
The junction tree and potential tables for the EMF for Task j are then constructed as usual.

This procedure guarantees that every EMF k will share an identical clique with the PMF, namely Θ_k.

Example 5.9 (Bayes net fragments for Example 5.5).
The total graphical model in Example 5.5 can be expressed as one PMF, over $\{\theta_A, \theta_B\}$, and two EMFs, one for each task, over $\{\theta_A, \theta_B, X_1\}$ and $\{\theta_A, \theta_B, X_2\}$. As before, $\{\theta_A, \theta_B\}$ is initialized at $[.012, .098, .098, .792]$ based on the the marginal distributions for $\{\theta_A\}$ and $\{\theta_B\}$ and the fact that they are independent. The EMF for Task 1 is initialized using the conditional probabilities for X_1 given θ_A and θ_B (as shown in the middle of Table 5.6) and the unit potential over all possible combinations of the values of θ_A and θ_B. The resulting initial potential in EMF_1 is thus

$$\begin{bmatrix} 1 & 1 \\ 1 & 1 \\ 1 & 1 \\ 1 & 1 \end{bmatrix} \otimes \begin{bmatrix} P(X_1 = 1 \mid \theta_A = \mathtt{H}, \theta_B = \mathtt{H}) & P(X_1 = 0 \mid \theta_A = \mathtt{H}, \theta_B = \mathtt{H}) \\ P(X_1 = 1 \mid \theta_A = \mathtt{H}, \theta_B = \mathtt{L}) & P(X_1 = 0 \mid \theta_A = \mathtt{H}, \theta_B = \mathtt{L}) \\ P(X_1 = 1 \mid \theta_A = \mathtt{L}, \theta_B = \mathtt{H}) & P(X_1 = 0 \mid \theta_A = \mathtt{L}, \theta_B = \mathtt{H}) \\ P(X_1 = 1 \mid \theta_A = \mathtt{L}, \theta_B = \mathtt{L}) & P(X_1 = 0 \mid \theta_A = \mathtt{L}, \theta_B = \mathtt{L}) \end{bmatrix}$$
$$= \begin{bmatrix} .99 & .01 \\ .90 & .10 \\ .90 & .10 \\ .01 & .99 \end{bmatrix} . \tag{5.9}$$

Note that the two columns of this result are the likelihood induced for θ_A and θ_B by observing $X_1 = 1$ and $X_1 = 0$ respectively.

By similar calculation, the initial potential in EMF_2 is

$$\begin{bmatrix} .99 & .01 \\ .05 & .95 \\ .90 & .10 \\ .01 & .99 \end{bmatrix} . \tag{5.10}$$

To illustrate PMFs and EMFs in a more interesting example, consider the case of five proficiency variables, $\theta_1, \ldots, \theta_5$, and three tasks. Task 1 and Task 3 contain one observable each, X_{11} and X_{31} respectively. Task 2 contains two observable variables, X_{21} and X_{22}, and also an unobservable evidence model variable X_{23} to account for conditional dependence between X_{21} and X_{22} (more about this idea in Sect. 6.2. The acyclic digraph is shown as Fig. 5.13.

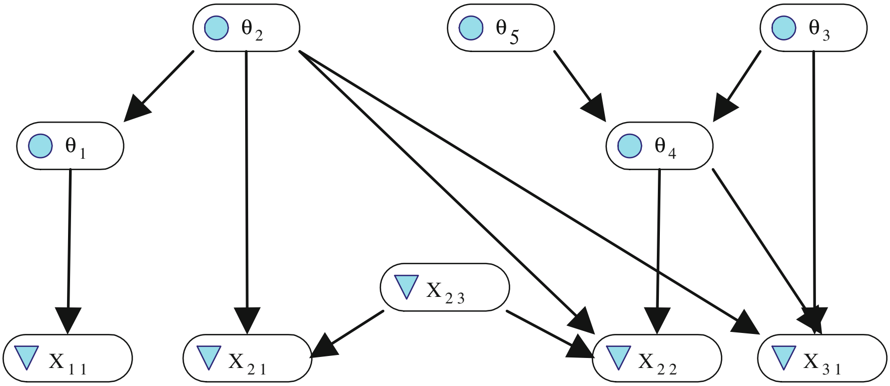


Fig. 5.13 Total acyclic digraph for three-task test
Reprinted from Almond et al. (2010) with permission from ETS.

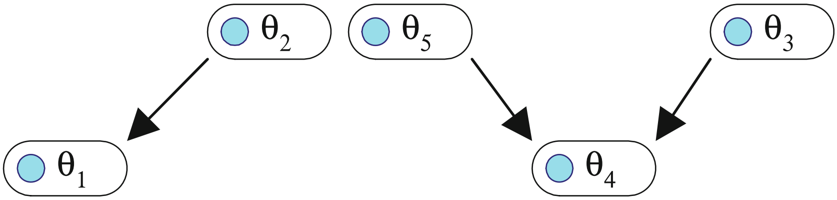


Fig. 5.14 Proficiency model fragments for three-task test
Reprinted from Almond et al. (2010) with permission from ETS.

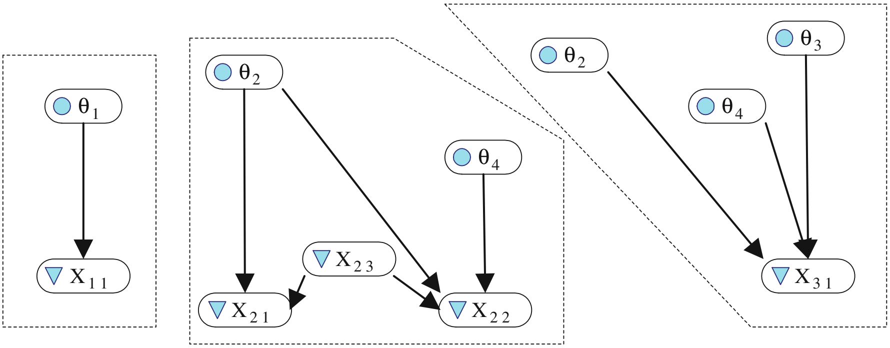


Fig. 5.15 Evidence model fragments for three-task test
Reprinted from Almond et al. (2010) with permission from ETS.

The digraphs that corresponds to the PMF is shown as Fig. 5.14, and the EMFs are shown as Fig. 5.15.

Because of independence and conditional independence relationships, the digraph for the proficiency model fragment (Fig. 5.14) is quite sparse. A junction tree for this digraph by itself consists of two cliques, $\{\theta_1, \theta_2\}$ and $\{\theta_3, \theta_4, \theta_5\}$. The footprints of the three tasks are these: $\Theta_1 = \{\theta_2\}$, $\Theta_2 = \{\theta_1, \theta_4\}$, and $\Theta_3 = \{\theta_2, \theta_3, \theta_4\}$. Θ_1 requires no new edges in the proficiency model fragment, but both Θ_2 and Θ_3 do. For example, Task 2 demands an edge between θ_2 and θ_4, which were independent in the original digraph. We refer to this phenomenon as an *induced dependency*. The moralized and triangulated undirected graph for the proficiency model fragment, with additional edges required to conform with the footprints of the three evidence mode fragments, is shown in Fig. 5.16. The moralized, triangulated, and conformable undirected graphs for the evidence model fragments are shown in Fig. 5.17.

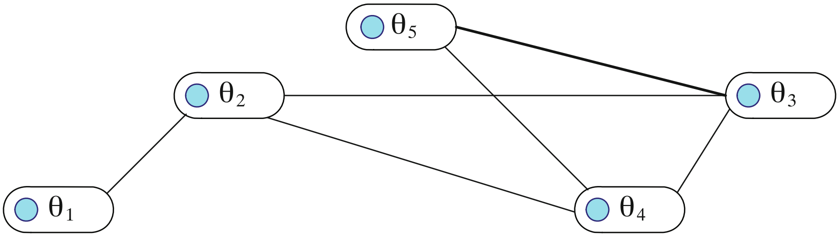


Fig. 5.16 Moralized proficiency model graph for three-task test Reprinted from Almond et al. (2010) with permission from ETS.

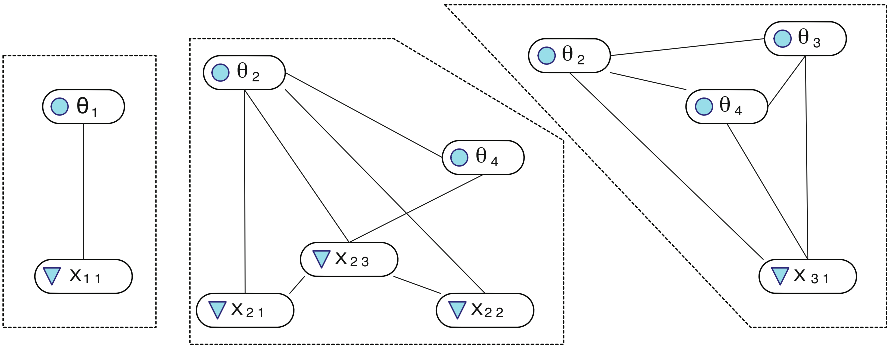


Fig. 5.17 Moralized evidence model fragments for three-task test Reprinted from Almond et al. (2010) with permission from ETS.

5.4.3 Calculation with Fragments

Once junction trees and potential tables have been constructed for each fragment in the manner described in the preceding sections, the PMEM algorithm (Almond et al. 1999) can update the PMF with evidence coming from any Task j in five steps:

Update Step 1: Start with the junction tree for the evidence model for Task j. Calculate the marginal distribution over Θ_j, $p_{old}(\Theta_j)$. If the junction tree was calculated according to the method of the previous section, this should be the unit potential, **1**.

Update Step 2: Cast the obtained evidence in the form of potentials over the observable variables and combine this evidence with the existing potentials over the observable nodes. This produces $p_{new}\left(x_{j1}, \ldots, x_{jn_j}\right)$.

Update Step 3: Apply the belief-updating algorithm to obtain the new joint distribution over the observables and footprint of Task j:

$$\left\{x_{j1}, \ldots, x_{jn_j}, \Theta_j\right\} \Rightarrow \Theta_k = p_{new}\left(\Theta_j\right) \ .$$

Update Step 4: The message EMF ⇒ PMF will be $p_{new}\left(\Theta_j\right) \oslash p_{old}\left(\Theta_j\right) = p_{new}\left(\Theta_j\right)$ (as the denominator is the unit potential). Enter this value as virtual evidence in any clique node in the junction tree for the PMF which contains the footprint, Θ_j.

Update Step 5: Apply belief updating to update the remaining proficiency variables outside Θ_j, if there are any.

At the end of the PMEM updating, the PMF contains the posterior distribution over the proficiency variables given the evidence from Task k. The EMF is no longer needed and can be discarded (or recycled for use with another student). In fact this gives us a simple Computer Adaptive Testing (CAT) engine. To start, the PMF contains the prior distribution over the proficiency variables for a student. As each observation arrives about an examinee, the engine fetches the appropriate EMF for that task from a database. The PMEM algorithm is then used to update the PMF. At any time the PMF can be queried to give our current state of knowledge about the student's proficiency. Chapter 13 expands on this idea.

Occasionally, we want to be able to forecast the values for observable outcome variables we have not yet seen. (Chapter 7 describes several applications of this capability.) We can run the PMEM algorithm backwards to obtain the predictive distribution for observables in another not-yet-administered task, Task j':

Predict Step 1: Start with the distribution of the proficiency variables after evidence from previous tasks has been entered. In the update steps above, this is $p_{new}\left(\theta_1, \ldots, \theta_m\right)$. Marginalize down to $\Theta_{j'}$ to obtain $p_{new}\left(\Theta_{j'}\right)$.

Predict Step 2: There will be a clique node in EMF j' corresponding to $\Theta_{j'}$. It will be the unit potential (unless some other source of evidence has already been used to update it). Marginalize down to $\Theta_{j'}$ to obtain $p_{old}\left(\Theta_{j'}\right)$. Combine the result with $p_{new}\left(\Theta_{j'}\right)$.

Predict Step 3: Apply the belief updating algorithm to update the remaining variables in the EMF junction tree, including in particular the observable variables $\left\{X_{j'1},\ldots,X_{j'n_{j'}}\right\}$. Marginalizing down to them gives the predictive distribution of the not-yet-administered observable variables in Task j'.

This use of the PMEM algorithm has a big advantage over the procedure of producing the total graphical model for the assessment that contains every observable for every task. In the latter configuration, the belief-updating algorithm propagates messages to all of the clique nodes to update these predictive distributions even when they are not needed (although modern Bayes net software usually uses lazy-propagation algorithms that only calculate messages in response to queries). In the PMEM algorithm, only the PMF is updated by default. The EMFs are only updated on demand, to answer a particular question.

Example 5.10 (Example 5.9, continued). *By the way it was constructed in Example 5.9, Update Step 1 of marginalizing the initial potential in* EMF_1 *with respect to* θ_A *and* θ_B *does yield the unit potential for* $p_{old}(\theta_A,\theta_B)$*. When evidence arrives that* $X_1=1$*, Update Step 2 tells us to zero out the right column (for* $X_1=0$*) in Eq. 5.9. Update Step 3 gives the left column as* $p_{new}(\theta_A,\theta_B)$*. In Update Step 4, the message* $p_{new}(\theta_A,\theta_B)\oslash old(\theta_A,\theta_B)$ *is simply* $p_{new}(\theta_A,\theta_B)$ *since* $p_{old}(\theta_A,\theta_B)=\mathbf{1}$*. It is combined with the initial potential for for* $\{\theta_A,\theta_B\}$ *to produce the new belief about* θ_A *and* θ_B*:*

$$[.012,.098,.098,.792]\otimes[.99,.90,.90,.01]=[.012,.088,.088,.008].$$

This is the same updated potential for $\{\theta_A,\theta_B\}$ *we obtained in Example 5.9 using the message passing algorithm. There are no other proficiency variables in the PMF, so nothing further needs to be done in Update Step 5.*

We are now in a position to forecast the response to Item 2 in light of having observed a correct response to Item 1. Predict Step 1 is to marginalize the new status of the PMF down to the footprint of the task in question. Nothing really needs to be done, since in this small example $\Theta_1=\Theta_2=\{\theta_A,\theta_B\}$*. In Predict Step 2, we marginalize the potential in* EMF_2 *down to the footprint and again obtain* $p_{old}(\theta_A,\theta_B)=\mathbf{1}$*. We combine this with* $p_{new}(\theta_A,\theta_B)$ *to obtain* $[.012,.088,.088,.008]$*; this is the message to pass to* EMF_2*. Predict Step 3 combines the message* $[.012,.088,.088,.008]$ *with the initial potential in* EMF_2 *(Eq. 5.10) to produce an updated potential, which reweights expec-*

tations about X_2 in accordance with our new belief about θ_A and θ_B:

$$\begin{bmatrix} .012 & .012 \\ .088 & .088 \\ .088 & .088 \\ .008 & .008 \end{bmatrix} \otimes \begin{bmatrix} .99 & .01 \\ .90 & .10 \\ .90 & .10 \\ .01 & .99 \end{bmatrix} = \begin{bmatrix} .011 & .005 \\ .004 & .426 \\ .080 & .046 \\ .000 & .041 \end{bmatrix} .$$

Marginalizing down to X_2 gives $[.095, .518]$ *and normalizing gives* $[.155, .845]$. *That is,* $P(X_2 = 1 \mid X_1 = 1) = .155$ *and* $P(X_2 = 0 \mid X_1 = 1) = .845$.

5.5 The Structure of a Test

In the previous section, we moved from viewing an assessment as a giant Bayesian network to viewing it as a library of network fragments: a central fragment based on the proficiency model and a collection of evidence models for each task that could be potentially used. Another evidence-centered design (ECD) model, the *assembly model* controls which tasks an examinee actually sees, and thus, what constitutes a valid form of the assessment.

This view of the assessment is helpful for task design. As the test designers focus on each task, they concentrate on the evidence provided by that task. By the local independence property, they only need to worry about evidence from one task; the evidence from other tasks should be conditionally independent, given proficiency variables.

However, the library of fragments view is not as useful for considering the evidence from an assessment as a whole. Assessment designers need to know if a given form is properly balanced so as to provide evidence about all of the claims to be made. To answer this question, the test designer must look across many fragments all at once.

An alternative way of viewing a graph is to use a matrix. Each row represents an observable variable and column represents a proficiency variable. Put a one in the matrix everywhere there is an edge in the graph; that is, for each instance where a given proficiency variable is a direct parent of a given observable. Doing that with the total graphical model of an assessment leads in a straightforward way to the *Q-matrix*, a representation that has been popular with many ways of modeling diagnostic assessment (Fischer 1973; Haertel 1989; Haertel 1984; Leighton and Gierl 2007; Rupp et al. 2010; Suppes 1969; Tatsuoka 1983). The Q-matrix provides the kind of view of the assessment that does help the designer answer questions about the balance and evidentiary properties of the assessment as a whole, or of one particular form.

Section 5.5.1 defines the Q-matrix more formally for an important subset of assessments, those which consist solely of discrete items (e.g., multiple choice), that each have a single, conditionally independent, observable outcome. Section 5.5.2 talks about how to expand the basic Q-matrix notation for assessments consisting of more complex tasks with multiple observables. Chapter 7 will look at how to use this new notation to assess the amount of evidence in a particular assessment.

5.5.1 The $\boldsymbol{Q}$-Matrix for Assessments Using Only Discrete Items

This book has been careful to use the term *task* rather than *item* to remind readers that assessment tasks can be more than just a collection of multiple-choice items. However, tasks which yield only a single observable outcome variable are easy to work with from a lot of different perspectives. In this section, we will restrict our attention to tasks which have a single observable outcome variable. The next section will talk about how to lift that limit.

Let $\Theta = \{\theta_1, \ldots, \theta_M\}$ be the set of proficiency variables in our assessment. Let $X_1, \ldots, X_j$ be the set of observable outcome variables associated with the items. Note that by our assumption above each observable is associated with a different task. In particular, that means by the local independence property X_j is independent of $X_{j'}$ given Θ.

Now consider the footprint for Task j, that is, the parents of X_j. Let $q_{jm} = 1$ if θ_m is a parent of X_j and 0 if it is not. Tatsuoka (1983) calls the matrix of $\boldsymbol{q}_j$s for a set of items the *Q-matrix* of a assessment. This matrix provides an at-a-glance view of the assessment. For example, the column sum $\sum_j q_{jm}$ is the number of tasks that tap the proficiency variable θ_m. Section 6.4 shows an example. An assessment that has only a single proficiency variable will have a Q-matrix that consists of just a single column of ones.

Consider a Q-matrix for which the sum of each row is 1; that is, each observable has only a single proficiency parent. Such a test is said to have *simple structure*. In particular, it can be thought of as a collection of unidimensional tests, one for each proficiency variable, that are combined in some way. Adams et al. (1997) refers to this as *between items multidimensionality*, in contrast to *within items multidimensionality* where at least some observable variables depend on more than one proficiency variable.

According to the construction algorithm in Sect. 5.4.2, the proficiency variables indicated in each row of Q must appear in a common clique in the final undirected graph for the proficiency model for the assessment. This is forced at the level of the PMF digraph by drawing an edge between all indicated θs in a row, where the edge is from the θ earlier (i.e., closer to the root) in the recursive representation to the one later. These are the induced dependencies mentioned earlier.

An interesting if unpleasant consequence is that even EMFs with simple structures in themselves can force many edges to be added to the graph for the PMF, and increase the treewidth of the junction tree for the proficiency model. For example, suppose there are five independent proficiency variables and five items, each of which has only two parents, as indicated by the Q-matrix in Table 5.9.

Table 5.9 Q-Matrix for design leading to saturated model

	θ_1	θ_2	θ_3	θ_4	θ_5
$x_{1,1}$	1	1	0	0	0
$x_{2,1}$	1	0	1	0	0
$x_{3,1}$	1	0	0	1	0
$x_{4,1}$	1	0	0	0	1
$x_{5,1}$	0	1	1	0	0
$x_{6,1}$	0	1	0	1	0
$x_{7,1}$	0	1	0	0	1
$x_{8,1}$	0	0	1	1	0
$x_{9,1}$	0	0	1	0	1
$x_{10,1}$	0	0	0	1	1

This design induces an edge between every pair of θs, and forces all the θs into a single large clique. This is called the *saturated model*, and the belief-updating algorithm offers no computational advantage over the brute-force algorithm for such models. This example is small enough that computational demand of the brute force example would not be an issue. But with a larger proficiency model, it is easy to imagine that allowing an unconstrained number of EMFs with unconstrained patterns of proficiency variable parents could easily lead to cliques and thus potential tables of an intractable size. Some ways to avoid this problem include the following:

- Limit the size of the proficiency model. (This option is often quite practical as limited test time forces test designers to concentrate on a few variables of interest.)
- Neglect minor θ-to-X edges.
- Predetermine a set of EMF structures and their footprints (i.e., *motifs*), ensure its computability, and constrain task development to those structures.

If designs that would require large saturated or nearly saturated junction trees for the PMF are desired nevertheless, the alternative approximations discussed in Sect. 5.6 can be pressed into service.

Recall from Chap. 2 that many tasks can be generated from a single task model. Usually, all of those tasks are scored using the same, or similarly structured, evidence models. This implies that the rows in the Q-matrix corresponding to the tasks from the same evidence model will be identical. For the purposes of determining the induced dependencies in the proficiency model, it is sufficient to have a single row in the Q-matrix for each motif.

5.5.2 The Q-Matrix for a Test Using Multi-observable Tasks

The Q-matrix provides a nice compact view of the entire assessment when all of the tasks are discrete items with only a single observable outcome variable.

However, much of our interest in Bayes nets stems from their ability to model more complex tasks with multiple observables. How do we extend the Q-matrix notation to include tasks with multiple observables? There are basically two options: (1) make the rows of the Q-matrix correspond to tasks/evidence models, and (2) make the rows of the Q-matrix correspond to observables.

Building a Q-matrix of the first type is straightforward. Each row of the Q-matrix becomes the representation of the footprint for that task. This version of the Q-matrix is particularly useful for scanning through the dependencies induced by a collection of tasks on a test form. The test designer can scan through the list quickly and estimate the treewidth of the final model.

Using the one row per task representation, the Q-matrix for the network depicted in Fig. 5.13 is:

Table 5.10 Q-Matrix for Fig. 5.13, one row per task

	θ_1	θ_2	θ_3	θ_4	θ_5
Task 1	1	0	0	0	0
Task 2	0	1	0	1	0
Task 3	0	1	1	1	0

The alternative is to use one row for each observable in the task, giving only the proficiency variables of that observable in the matrix. Table 5.11 depicts the Q-Matrix for the model in Fig. 5.13 using the one row per observable representation. However, the variable X_{23}, which is used to model local dependence between X_{21} and X_{22} presents some problems. First it produces a row with no "1"s, which is somewhat odd. Second, neither the relationship between X_{23} and X_{21} nor the relationship between X_{23} and X_{22} are captured in the Q-matrix.

Table 5.11 Q-Matrix for Fig. 5.13, one row per observable

	θ_1	θ_2	θ_3	θ_4	θ_5
x_{11}	1	0	0	0	0
x_{21}	0	1	0	0	0
x_{22}	0	1	0	1	0
x_{23}	0	0	0	0	0
x_{31}	0	1	1	1	0

Thus, we can see that in the tasks with multiple observables, we are bumping up against the limits of the Q-matrix representation. Bayesian networks have a richer set of tools to describe complex relationships among variables. That is a large part of our interest in using them. Still, the Q-matrix is good for providing an at-a-glance summary of an entire assessment form. For this reason, we often compute the Q-matrix for a given Bayesian network model as a way of evaluating the assessment.

5.6 Alternative Computing Algorithms

The preceding sections introduced a method for efficient calculation in Bayesian networks. This is by no means an exhaustive treatment of the topic, but it does lay a foundation for the use of Bayesian networks in the following chapters. A large and active research community has grown up around the use of Bayesian networks. This last section provides some pointers into the body of literature that community has developed. In particular, Jensen (1996) and Cowell et al. (1999) provide recent tutorial references. Pearl (1988) remains a classic in the field and has many chapters that anticipated current research trends. Lauritzen (1996) describes how to extend the algorithm to graphical models which include normally distributed continuous variables and mixtures of discrete and continuous variables.

The algorithm presented in the previous sections is a variant of the propagation algorithm described in Lauritzen and Spiegelhalter (1988). That paper spawned a large family of algorithms for propagating in Markov trees. The variant presented here first appeared in Cowell et al. (1993).

Shenoy and Shafer (1990) show how the algorithm can be extended to models which are not purely probabilistic, including influence diagrams and graphical belief functions. They define a general version of the belief-updating algorithm called the fusion and propagation algorithm, which is based on *valuations.* A valuation describes the relationship about the states of a set of variables in a *frame of discernment*, the prime example being the probability potential we have been working with. Valuations support the *combination* and *projection* operations that are the basis of message passing. They, in combination with the running intersection property, provide for efficient updating algorithms.

Section 5.6.1 notes variants on the basic algorithm and Sect. 5.6.2 explores variations of the algorithm and approximation techniques that can be used when the treewidth of the model gets too large.

The *Uncertainty In Artificial Intelligence (UAI)* community is constantly doing research into better algorithms for computation with Bayesian networks, and recent copies of the UAI proceedings are a good source for keeping up with the latest developments Many software packages exist which implement both variations on the basic Markov tree propagation algorithm and many of the variants described here. Appendix A contains some pointers to online resources for both software and articles.

5.6.1 Variants of the Propagation Algorithm

By making small changes in the fusion and propagation algorithm we can adapt it for a number of special purposes. In particular, we can use it to find the most likely *proficiency profile* and to sample from the joint distribution of all variables.

Most Likely Configuration

Pearl (1988) notes that by simply using maximization instead of summation in the belief-updating algorithm, one can find the most likely configuration of a set of nodes. The idea is that any time we perform a marginalization operation (going from a clique node to an intersection node in the Junction tree, we pick the value of of the eliminated variable(s) that maximizes the probability. When one reaches the end of the Markov tree, pick the most likely configuration for the remaining variables.

In the educational setting, we are usually interested in configurations of proficiency variables. Thus the output of this algorithm is usually the *proficiency profile* which provides the most likely explanation for the observed evidence. Section 7.1.2 takes up this idea.

Sampling Algorithm

A junction tree representation can be used to sample from the joint distribution of all of the variables in the model. First, pick any node in the Markov tree as a root node. Draw a sample configuration from those variables, in accordance with the probabilities its potential implies. Now condition the model on the sampled values and start propagating out from the root. At each new node in the junction as we move out, we have the correct conditional distribution based on the sample so far. Sample the remaining variables in this node and continue propagating outwards until all values are sampled.

This can be very useful for calculating the accuracy of a proposed assessment design (Sect. 7.5). In a typical experiment, we start by simulating a set of proficiency variables for a simulee. This produces a proficiency profile for the simulee. Next, for each task in the assessment design we can sample a set of observed outcomes given the conditional distribution of the observables given the proficiency variables. The result is an observation vector for that simulee. This kind of simulation experiment has a number of different uses, such as calculating traditional reliability indices, item statistics, and expected score distributions, and fitting alternative models to the generated data.

5.6.2 Dealing with Unfavorable Topologies

The Peeling Algorithm

If we are only interested the marginal distribution of a particular set of variables, we can sequentially eliminate the remaining variables one by one. At each step, we combine all of the potentials involving the eliminated variable and project the result onto the remaining variables. The procedure is called *peeling*. Hilden (1970) and Cannings et al. (1978) first developed the peeling algorithm to answer questions about genetic probabilities in complex pedigrees.

The disadvantage of peeling compared to the Markov tree propagation algorithm is that it can produce only one marginal distribution at a time; producing other marginal distributions requires repeeling the model, possibly many times. In such cases, it usually makes sense to transform to the Markov tree. On the other hand, if we know precisely which query we want to make peeling can take advantage of special structures in the model (Li and D'Ambrosio 1994).

Cut Set Methods

In both the Markov tree propagation algorithm and the peeling algorithm, the cost of the computation is largely determined by the treewidth of the graph. What can we do when that largest clique gets too large for these algorithms to be practical? One suggestion presented in Pearl (1988) is to choose a variable or set of variables to "condition out" of the model. Because these variables are typically chosen to cut loops, they are called a *cut set*. The best cut variables are usually roots of the original directed graph (or close to the roots).

For example, suppose we have a proficiency model consisting of an *Overall Proficiency* variable which has three levels and several subskills. We choose the *Overall Proficiency* variable (usually the root in the model graph) as our cut set. We make three new graphical models by conditioning on the three possible values of the overall proficiency variable. Because we have conditioned on this variable, it can be eliminated from the model. If, as typically happens, the overall proficiency variable is the apex of a number of loops, the resulting conditional models will have lower treewidth. We now build a junction tree for each of those models. We weight the resulting trees by the original probabilities of the overall probability model. The resulting model is a weighted mixture of trees.

Not only do the conditional models eliminate the cut variable, but they also could have different graphical forms (Heckerman 1991). For example, there may be a dependency between certain skills at high levels of proficiency which is not typically observed at lower levels of proficiency.

The update algorithm for the mixture of trees is straightforward. We start by updating each tree in the normal fashion. We next update the weights. This is done in the normal fashion using Bayes rule with the likelihood of the particular observation. Any query we make is a weighted average of the queries from each of the conditional models.

Loopy Belief Propagation

Pearl (1988) suggests that one could simply apply the propagation for trees in loopy graphs iteratively with every node send a message to its neighbors every cycle. In the case of polytrees, this algorithm always converges. In the case of graphs with loops, the algorithm may or may not converge. Weiss (2000) attempts to characterize situations in which the algorithm will converge and produce proper marginal distributions, and Murphy et al. (1999) attempt to validate these situations empirically. Murphy suggests that when loopy belief propagation does not converge it oscillates between two or more states. This might correspond to the explaining away problem mentioned earlier, where the system would oscillate between two modes indicating alternative explanations for the observed evidence.

Variational Approximations

Jordan (1998) and Jaakkola (2001) describe an approximate inference method for Bayes nets based on calculus of variations. The basic idea is that the Bayes net is approximated by another Bayes net with a lower treewidth, and calculus of variations is used to find the optimal approximation. The usual approximation described is the mean field method, in which the approximating graph assumes that all of the variables are independent. This approximation is usually pretty close on the marginal distribution of the variables but sacrifices information about the interactions.

Particle Filtering

Sequential importance sampling (Liu 2001), sometimes known as *particle filtering* can also be used to calculate approximate posterior distributions. Particle filtering can be done in the following steps:

1. Sample a collection of skill profiles, configurations of proficiency variables, according to the prior distribution (this is given by the proficiency model). These are the “particles.” Assign each of them equal weight.
2. As each new piece of evidence arrives, adjust the weights by multiplying by the likelihood of that evidence given the proficiency state described by the particle.
3. After a period of time, the weights of some particles will become very small. At this point in time, resample from the existing particles using their current weights.

Koller and Learner (2001) and Murphy and Russell (2001) describe particle filtering to handle calculations in dynamic Bayesian networks (Bayesian networks which capture changes to a system over time). There is an additional step at each time point as we put the particles through a random growth operator. Thus in the dynamic Bayes net model, the particles represent trajectories through the proficiency states at various points in time.

Exercises

5.1. Assume that we have built a Bayesian network with proficiency variables $S_1, \ldots, S_K$ and observable outcome variables $X_1, \ldots, X_m$. For simplicity, assume that all variables are binary. Assume that a learner about whom we do not have information other than the fact that this learner is from the population for whom the test is designed, has just walked into the test center. For each of the following questions, write a symbolic expression that answers the question.

a. *Marginal Belief* "What is the probability that the learner has skill S_1?"
b. *Marginal Belief 2* "What is the probability that the learner will have a good outcome on observable X_1?"
c. *Conditional Belief* "What is the probability that the learner has skill S_1 given we have made observations on tasks $\mathbf{X} = \{X_1, \ldots, X_n\}$?"
d. *Hypothetical Belief* "Given that we have made observations on tasks $\mathbf{X} = \{X_1, \ldots, X_n\}$, if we observed a performance on Task Y, how would our beliefs change?"

5.2. Reverse the direction of the arrow between W and X in Fig. 5.2. How do the calculations in Sect. 5.2.1 change?

5.3. Reverse the direction of the arrow between Z and Y in Fig. 5.2. How do the calculations in Sect. 5.2.1 change?

5.4. Starting from the final state of the junction tree in Example 5.3, calculate the effect of evidence e_2 that $Y = 0$. Does the marginal distribution for W change? Why or why not?

5.5. Figure 5.3 gives the junction tree that corresponds to the factorization of $P(U, V, X, Y, Z)$. Draw the intermediate steps of a directed graph for the distribution, the factorization hypergraph, and its 2-section.

5.6. Change the performance rating procedure for the Dental Hygienist assessment in Example 5.4, by adding a second rater. Assume that the two raters work independently, but all rate the same performance. What does the graph look like (include nodes for the ratings)? How about for N raters?

5.7. Suppose that the rater in Example 5.4 instead of calling a history adequate or not, gives a probabilistic rating, such as "the probability of this history being adequate is 2/3." How should that evidence be entered into the model?

5.8. Consider the Dental Hygienist Exam of Example 5.4. Suppose that through training we can increase the sensitivity and specificity of the raters, so that we have a perfect rater that always rates adequate performances as `adequate` and inadequate performances as `inadequate`. What is the maximum value for the posterior probability of the dental hygiene skill after the patient history task? Note that one way to increase the accuracy of the rating is to have more and more raters rate the same performance. What is the maximum change in the posterior probability we can get by increasing the number of raters?

5.9. Imaging that we want to add a task to the assessment described in Example 5.6 (Table 5.6) that had good differential diagnosis for θ_A, but not θ_B. What would its conditional probability table need to look like?

5.10. In Fig. 5.4, replaced the directed edge $T \rightarrow U$ with $U \rightarrow T$ and replace $X \rightarrow Y$ with $Y \rightarrow X$. What is the size of the largest clique in the moral graph for the modified graph? What is the treewidth of the junction tree? Why is the one smaller than the other?

5.11. When the PMFs and EMFs are constructed separately, after moralization and before triangulation, additional edges are added to join together proficiency variables that appeared together in the footprint of one or more evidence fragments. This step was not included when constructing the total graphical model for the assessment (that is the proficiency and evidence model fragments are together in one big graph). Why was it not necessary there?

5.12. A design committee for a new assessment identifies six proficiencies, $\theta_1, \ldots, \theta_6$, which it wants to measure with a new assessment. The committee structures the proficiency model so that all six proficiencies are independent given an overall proficiency labeled θ_0. They propose a collection of nine tasks, each of which tap two proficiencies as show in Table 5.12.

a. Draw the graph for this example.
b. Calculate the treewidth for this assessment.
c. How would you recommend the committee reduce the treewidth of the assessment? Hint: Are all of the tasks necessary?

Table 5.12 Q-Matrix for proposed assessment (Exercise 5.12)

	θ_1	θ_2	θ_3	θ_4	θ_5	θ_6
Task 12	1	1	0	0	0	0
Task 23	0	1	1	0	0	0
Task 34	0	0	1	1	0	0
Task 45	0	0	0	1	1	0
Task 56	0	0	0	0	1	1
Task 61	0	0	0	0	0	1
Task 14	1	0	0	1	0	0
Task 25	0	1	0	0	1	0
Task 36	0	0	1	0	0	1

5.13. Suppose that we have an assessment described by a PMF and a small set of EMFs. Suppose further that the assessment design calls for each examinee to see each task. Section 5.6 presents a simple algorithm for sampling from a junction tree. Explain how that sampling algorithm can be extended to work when the PMF and EMFs are maintained separately.

5.14. Let $\mathcal{C}$ be the set of clique nodes in a junction tree, and let $\mathcal{I}$ be the set of intersection nodes. Let $p_n(\cdot)$ be the potential over Node n. The condition for the junction-tree algorithm can be expressed as

$$\mathrm{P}(\cdot) = \left(\prod_{C in \mathcal{C}} p_C(\cdot) \right) \oslash \left(\prod_{I in \mathcal{I}} p_I(\cdot) \right) \ .$$

That is, the joint probability of all the variables is the product of all of the potentials over the clique nodes, divided by the product of the potentials over the intersection nodes.

Use the simple chain graph in Fig. 5.2 and demonstrate that this is correct. Hint: Use the fact that $p(x, y)/p(x) = p(y|x)$ and use that to recover the original recursive definition of the joint probability.

6

Some Example Networks

This chapter will illustrate the calculations shown in Chap. 5 with specific examples. It shows the basic ways that belief updating, or propagation, is used in discrete Bayes nets applications for assessment. The conditional probabilities are taken as known for now, as subsequent chapters will address just where they come from (to anticipate: theory, design, expert opinion, and data).

In educational assessment, the objective is to uncover the relationships between the students' unobservable characteristics and the observable outcomes from students' performance on tasks. The methods in the previous chapters allow us to answer these questions. Once the probability models are built and embedded in a graphical structure that reflects our knowledge about interrelationships among the variables, we can propagate evidence through the model and make inferences about a student.

As we discussed in Chap. 5, updating for the full joint distribution using the definitional expression of Bayes theorem is often prohibitively expensive, even for relatively small numbers of variables. A model with 15 variables and 4 values for each variable already means working with a joint probability table with over a trillion entries. Several computer software packages exist to help with this task. Appendix A presents a brief summary of several commonly available software application for doing this task as well as instructions for where to download some of the examples used in this chapter.

The goal of this chapter is to describe how, with the help of Bayes net software, to build and use Bayesian networks as the scoring engine for an assessment. Section 6.1 begins with a simple item response theory (IRT) model example translated into its Bayesian network equivalent. Section 6.2 expands the example in Sect. 6.1 to include a context effect that is similar to the testlet-effect IRT model (Bradlow et al. 1999). In this model, some items in a particular context are correlated with each other beyond their joint dependence on the proficiency variable. Section 6.3 illustrates three combination distributions for use when an observable outcome variable has more than one parent: the compensatory, conjunctive, and disjunctive distributions. Parallel models using the three distribution types provide a mechanism for studying

R. G. Almond et al., *Bayesian Networks in Educational Assessment*,
Statistics for Social and Behavioral Sciences, DOI 10.1007/978-1-4939-2125-6_6

the behavior of the three distributions in application. Section 6.4 shows a more complicated educational assessment example drawn from real data, a binary-skills measurement model with multiple, intercorrelated, proficiency variables.

6.1 A Discrete IRT Model

A very common case in educational testing is an assessment designed to assess a single proficiency. Usually the domain of tasks is restricted to the kind of simple items that can be unambiguously scored as `right` or `wrong`. Multiple-choice items are natural in this context, but short constructed-response tasks such as fill in the blank items are often used as well (especially, if the evidence rules are to be applied by a human rater). In this case, the rules of evidence are very simple and most of the work in building an evidence model goes into determining the strength of the relationship between the observable outcome variable and the proficiency model.

As this simple case is very common, a large number of psychometric approaches have been developed to model it. The most widely used is IRT (Hambleton et al. 1991; Thissen and Wainer 2001). By convention, the single proficiency variable[1] in the IRT model is called θ. The observable outcome variables, one for each task or item[2], are called X_j and usually take on the values 0 (for incorrect responses) and 1 (for correct responses).

A fundamental assumption of the IRT model is the local item independence property, that is, $X_j \perp\!\!\!\perp X_{j'} | \theta$. Using this assumption, we can write the joint probability distribution over both the proficiency and evidence variables as:

$$\mathrm{P}(X_1, \ldots, X_J, \theta) = \mathrm{P}(\theta) \prod_{j=1}^{J} \mathrm{P}_j(X_j|\theta) \ . \tag{6.1}$$

From the previous chapters, it is readily apparent that this factorization structure can be represented with a graphical model, such as the one in Fig. 6.1.

The IRT model is an example of the more general graphical model rather than a discrete Bayesian network. This is because, typically, in IRT the proficiency variable θ is continuous. Not only that, but the direction of the arrows goes from the continuous to the discrete variables, so it does not fall into the computationally convenient class of conditional Gaussian networks (Lauritzen 1992; Lauritzen 1996), for which all integrals involved in the the fusion and propagation algorithm (described in Chap. 5) can be solved in closed form. Of course this is old news in the field of psychometrics, where a large number of approximate methods have been developed for working with the IRT model.

[1] This is sometimes called a *proficiency parameter* in the IRT literature, but in this book we use *variable* to emphasize that the value is specific to a person.

[2] In this context each task consists of a single discrete item and so the terms *task* and *item* can be used interchangeably.

Approximating the continuous proficiency variable, θ, with a discrete variable, for example, restricting $\theta \in \{-2, -1, 0, 1, 2\}$, makes all variables discrete and creates a Bayesian network. As all variables are discrete, there are no integrals that need to be solved numerically when scoring students. This approximation may not even be that bad. Haberman (2005a) compares a five level ordered latent class model (the simple Bayesian network model posed here is essentially an ordered latent class model) to a unidimensional IRT model and notes that both models fit the chosen data sets equally well. Furthermore, an instructor using the inferences from the network may not be concerned with finer distinctions. A student for whom $\theta = 0$ is doing about as well as expected. A student for whom $\theta = -2$ is clearly in need of extra attention and a student for whom $\theta = +2$ could benefit from extra curricular work. Students for whom $\theta = -1$ should be watched closely to make sure they do not slip further down and students for whom $\theta = +1$ could be stimulated to try and move them further up the scale.

There are a number of different variants of the basic IRT model depending on how the evidence model, $\mathrm{P}_j(X_j|\theta)$, is parameterized. Technically speaking, the evidence model also contains the rules of evidence, but those are often quite simple, e.g., matching the key with the selected option. They are typically left in the background in the IRT literature, and attention focuses on the more interesting part of the evidence model, namely, the probability of the observable, given the proficiency variable. The example below uses the Rasch model (Rasch 1960), which uses the following probability function:

$$P(X_j|\theta, \beta_j) = \frac{1}{1 + e^{-(\theta-\beta_j)}} \ . \tag{6.2}$$

The parameter β_j is called the difficulty of the item. Note that the difficulty paralmeter and the proficiency variable are on the same scale. A person whose proficiency exactly equals the difficulty of the item would have a 50–50 chance of getting that item correct.

Although the Rasch model was built to work with continuous proficiency variables, we can use it to fill out the conditional probability tables (CPT) which drive the Bayes net approximation to the IRT model. In particular, by plugging the values $\theta = -2, -1, 0, 1, 2$ into Eq. 6.2 we get the values for each row of the tables for Fig. 6.1. For Item 3, for example, $\beta_3 = 0$, so probabilities of a correct response at the five θ levels are .1192, .2689, .5000, .7311, and .8088. The following example illustrates this idea.

Example 6.1 (Classroom Math Quiz). *Consider a simple math quiz consisting of five items scored to yield observable values of* `Right` *and* `Wrong`*. Let θ represent a student's proficiency in the math knowledge and skills that this set of items taps. The teacher is interested in drawing inferences about the math proficiency of each student based on the observed score patterns from the quiz. Figure 6.1 shows the Bayesian network for this five item quiz.*

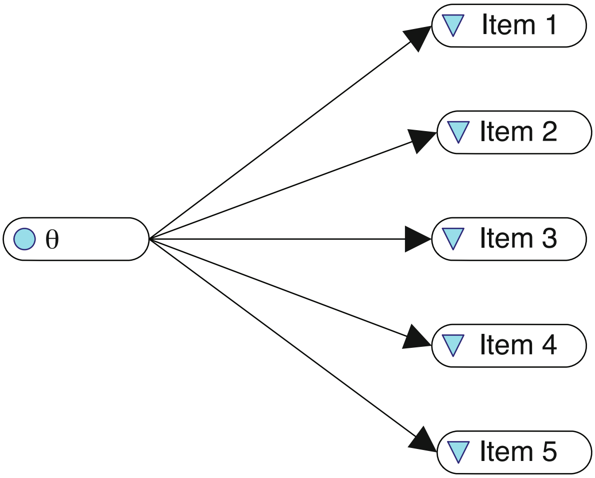


Fig. 6.1 Model graph for five item IRT model
Reprinted with permission from ETS.

To make the model discrete, restrict the proficiency variable to fall in the set $\theta \in \{-2, -1, 0, 1, 2\}$. Further assume that the proficiency of the students in the class is distributed with a triangular distribution, with 40 % of the students at the 0 *level, 20 % of the students at both the* $+1$ *and* -1 *levels, and 10 % of the students at both the* -2 *and* $+2$ *levels. Suppose further the chance of a student answering the item correctly follows the Rasch model (Eq. 6.2) and that the items and range from easy (Item 1) to hard (Item 5) with* $\beta = -1.5, -0.75, 0, 0.75, 1.5$. *Table 6.1 contains the conditional probabilities of a correct response for this Bayes net.*

Table 6.1 Conditional probabilities of a correct response for the five-item IRT model

θ	Prior	Conditional Probability				
	θ	Item 1	Item 2	Item 3	Item 4	Item 5
-2	0.1	0.3775	0.2227	0.1192	0.0601	0.0293
-1	0.2	0.6225	0.4378	0.2689	0.1480	0.0759
0	0.4	0.8176	0.6792	0.5000	0.3208	0.1824
1	0.2	0.9241	0.8520	0.7311	0.5622	0.3775
2	0.1	0.9707	0.9399	0.8088	0.7773	0.6225

6.1.1 General Features of the IRT Bayes Net

Figure 6.1 and Table 6.1 provide enough information to specify the Bayesian network for Example 6.1. Although this model is small enough that one could do all of the calculations in this section by hand, it is much more convenient to use a computer. As the software can react quickly to new data and other changes in the model, properly designed Bayesian network software encourages the modeler to explore the model by posing a large number of hypothetical questions. Appendix A describes how to obtain many of the more popular Bayes net software packages, many of which have free student or research versions which are suitable for following along with this example. The appendix also lists where copies of the example network can be downloaded (although this example is small enough to enter by hand).

Most Bayesian network software operates in two modes: a model construction/editing mode and a model manipulation or inference mode. In the model construction mode, the analyst performs the following steps:

1. Construct a *node* for every *variable* in the model. The number and names of the variables possible states must be specified. Various software packages provide places for specifying other details about the node (e.g., a definition).
2. Draw *edges* between the nodes to represent the conditional dependence relationships inherent in the model.
3. Specify a CPT for each node in the model, given its *parents* in the graph. If a node has no parents, an unconditional probability table is used.

Although the modeling software offers considerable freedom in what order those three steps are completed, they must be completed for every node in the model before the model itself is “complete.” Once the model is complete, the model is *compiled*, i.e., a *junction tree* is built from the completed model. This junction tree is then used to support inference. If the model is later edited (e.g., another node is added), the compilation step must be repeated for the revised model.

After the compilation step, most Bayesian network software packages immediately display the marginal distributions for all nodes (some packages require you to specifically query the nodes you want to display). Figure 6.2 shows the result of compiling the Bayes net from Example 6.1 in the Netica (Norsys 2004) software package. Table 6.2 gives the conditional probabilities in a slightly more legible format.

These marginal probabilities represent our best predictions of how a student who we know nothing about, other than that the student is representative of the population for which the model was built, would do on this quiz. (They are calculated from the conditional probabilities of correct response given θ and the initial marginal distribution for θ, using the law of total probability, Eq. 3.4). These are the expected proportions correct (P+) on each item in a population of students for whom this quiz is designed.

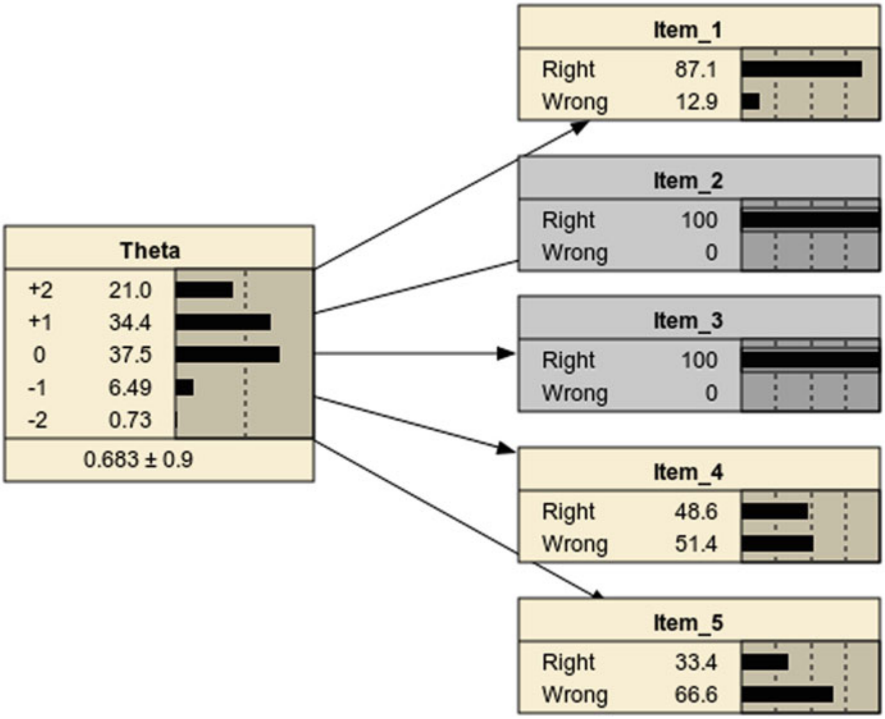


Fig. 6.2 The initial probabilities for the IRT model in Netica. The numbers at the bottom of the box for the Theta node represent the expected value and standard deviation of Theta

Reprinted with permission from ETS.

Table 6.2 Initial marginal probabilities for five items from IRT model

	Item 1	Item 2	Item 3	Item 4	Item 5
Right	0.77	0.65	0.49	0.35	0.23
Wrong	0.23	0.35	0.51	0.65	0.77

6.1.2 Inferences in the IRT Bayes Net

One important reason for using Bayesian network software, instead of simply doing these calculations by hand, is that the software will nearly instantly (especially with such a small model) update the distributions to take the new evidence into account. The basic operation is called *instantiating* the value of a variable. To do this, select the variable in the model and chose a value for that variable (The details of how this is done are similar but different for different Bayes net packages. In particular, many packages have an "auto-update" mode which will immediately propagate evidence to other nodes in the model so that they immediately reflect the changes, and a "batch" mode in which propagation of evidence must be requested manually). This basic facility can be used to both enter data and play "what-if" games for various hypotheses.

The most basic application of this idea is the scoring procedure for Bayesian networks. Out of the evidence identification process (item level scoring) will come the values of a collection of observables (in this case, right/wrong judgments) for the variables in our model. To score the student, simply instantiate the nodes in the graph corresponding to the observable variables with the values from evidence identification. The updated graph shows the posterior distribution for the proficiency variables, now conditional on the responses that have been observed.

Example 6.2 (Absorbing Evidence from Simple Math Quiz, Example 6.1 continued). *Suppose that Ali, a student in this class, has observed outcomes of Item 2 =* `Right` *and Item 3 =*`Right`*. Instantiating these value in the network and propagating the information using the methods of Chap. 5 produces the posterior distribution for Ali's θ (Fig. 6.3*a*). It has shifted toward higher probabilities for higher values of θ. The probabilities for Ali to get the other items "Right" also increase as a consequence.*

*Ali's classmate Yuri also answers two items correctly, this time Items 3 and 4. The state of the network after instantiating the values of these observables is shown in Fig. 6.3*b*. The shift toward higher values of θ is stronger because the two items that Yuri answered correctly were more difficult than the ones Ali answered. Although the story here is much the same as in Ali's case, we will see an interesting difference in the next section.*

Although in practice, the proficiency variables are unobservable, the Bayes net software allows us to hypothesize values for the latent variables in order to play out various scenarios. The following example illustrates this idea.

Example 6.3 (Calculating Conditional Probabilities for Simple Math Quiz (Example 6.1 continued)). *In the class for which the quiz was designed, consider the groups of students corresponding to the five proficiency levels (We don't actually ever know students' proficiencies, of course, so we can not form these groups for, say, small group instruction. The best we could do is to form groups based on what we do know about their proficiencies—for example, the modal value of their posterior distributions after they have taken a quiz). Suppose that Group A corresponds to $\theta = 2$, Group B corresponds to $\theta = 1$, and so forth. What is the expected performance on each of the items for a member of Group B?*

In the IRT Bayes net, instantiate $\theta = 1$, then propagate the probabilities through the Bayes net. This yields probabilities as in Fig. 6.4). This is the fourth row of Table 6.1. Group A corresponds to the fifth row of that table and Group C to the third row.

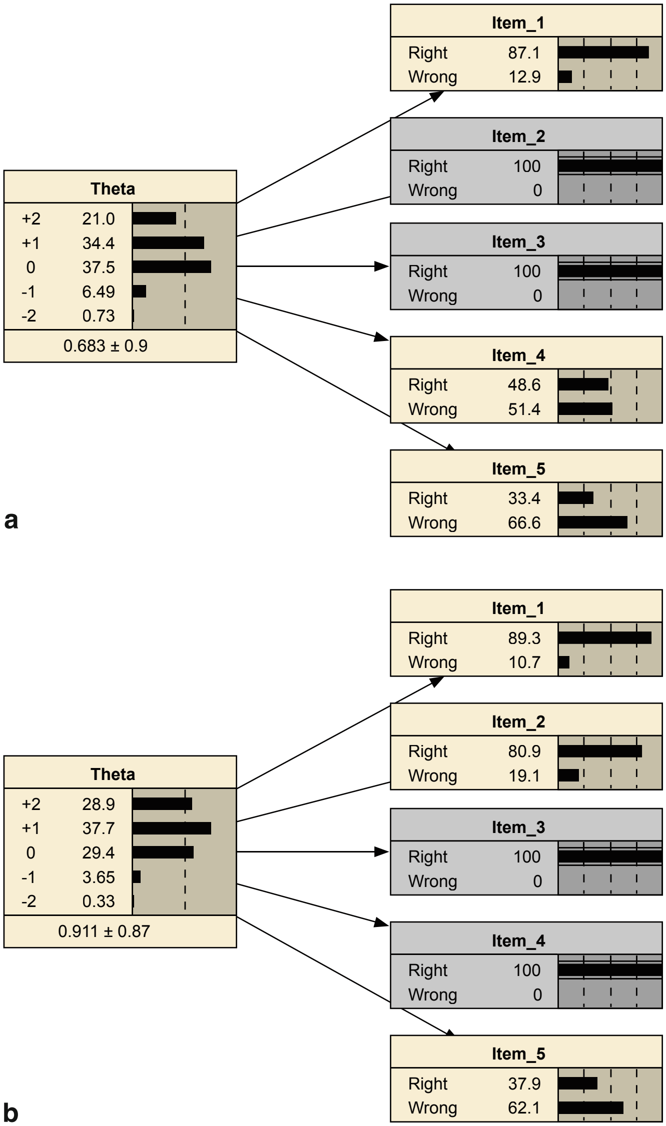


Fig. 6.3 a Student with Item 2 and 3 correct. **b** Student with Item 3 and 4 correct Reprinted with permission from ETS.

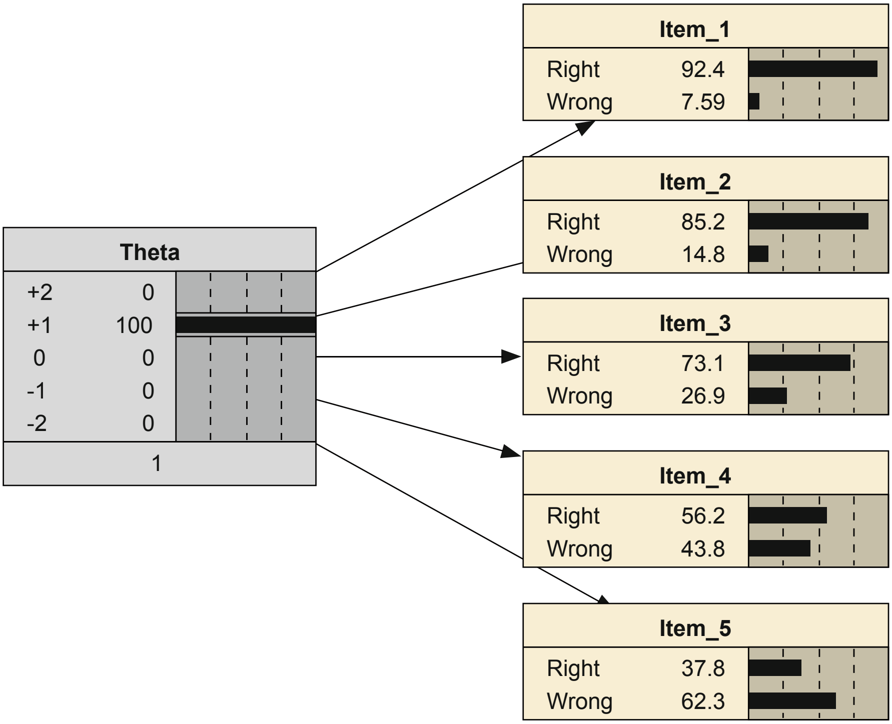


Fig. 6.4 Probabilities conditioned on $\theta = 1$
Reprinted with permission from ETS.

Example 6.3 is a rather trivial application of this conditioning idea; Exercise 6.3 extends the model a bit further. This technique is more interesting in models with more than one proficiency variable. Here, conditioning on a single proficiency variable will have subtle influence on both the values of other proficiency variables and the implications of evidence from integrated tasks that tap more than one proficiency.

One application of this technique is in validating a Bayesian network. In this simple example, it can be used to verify that the probabilities from Table 6.1 were correctly entered. In more complex models it can be used with domain experts to validate properties of the model. For a given proficiency profile—that is, instantiating values of all of the proficiency variables—look at the predicted observables on each task and verify with the experts that this distribution is reasonable.

Examples 6.2 and 6.3 also show how Bayesian networks support both *deductive* and *inductive* reasoning. Example 6.3 is an example of deductive reasoning: The network reasons from the known value of the proficiency variable to deduce the likely consequences. Example 6.2, on the other hand, is an example of inductive reasoning: From the known values of the observed outcomes, the network induces a posterior distribution for the unknown proficiency.

Another important application of conditioning on the unknown proficiency variable is calculating the expected weight of evidence for an unobserved item or task. Section 7.2 defines this idea formally, but in the context of the simple discrete IRT example, there is a simple short cut. In this case, we can lean on a result from IRT theory that says that under the Rasch model, an item for which a student has a 50–50 chance of getting right will provide the most information.[3] Therefore, looking for the item whose predictive probability, given the evidence so far, is closest to 50–50 produces a simple adaptive testing mechanism (The situation become much more complex when there are multiple proficiency variables, see Chap. 7).

Example 6.4 (Item Selection for Simple Math Quiz, Example 6.2 continued). *Recall Ali from Example 6.2, who got Items 2 and 3 correct. Now, what items should we give Ali next? The responses Item 2 =* `Right` *and Item 3 =* `Right` *provide some evidence about Ali's proficiency. Entering this information into the Bayes net (Fig. 6.3a) has updated not only our posterior distribution over theta, but also the predictive distributions for Items 1, 4, and 5. The updated probability of getting a* `Right` *outcome is .87 for Item 1, .48 for Item 4, and .33 for Item 5. Thus for what we currently know about Ali, Item 4 appears to be the best choice; Item 1 is too easy and Item 5 is too hard.*

6.2 The "Context" Effect

The example in the previous section recreates results from IRT, or because θ was made discrete, from a comparable structured latent class model. It shows how results from more familiar testing models can be expressed in terms of Bayesian networks. It does not yet highlight the strength of Bayesian networks, which is their flexibility.

One way to put that flexibility to use is in modeling more complex tasks, not just simple discrete items. Natural examples of complex tasks are plentiful: reading passages followed by several multiple choice (or short answer) items, multistep tasks, and simulation tasks which are scored on multiple aspects of the same performance. In such contexts, it is more natural to think of a

[3] It may seem counterintuitive that an item for which a student has 50–50 chances of responding correctly provides the most information about her θ. The way to think of it is, "Which item provides the biggest difference between the posteriors that result if she gets it right or gets it wrong?"

defined collection of material presented to the examinee and the work products obtained in the performance as a single task, with multiple observable outcomes, as opposed to a collection of items that is somehow bound together.

A problem with such complex tasks is the observable outcome variables are likely to be dependent, even after conditioning on the relevant proficiency variables. In the usual IRT framework, where all items are assumed to be locally independent, given the proficiency variables, the observable variables from these complex tasks violate that assumption (Yen 1993). However, because the evidence-centered design (ECD) framework calls for a single evidence model to score all observables coming from a single task, a more complex Bayes net model can be built to account for the local dependence.

One source of local dependence is common stimulus material shared by several items. A common example is a reading passage followed by several multiple-choice items. Although the items are typically authored to minimize the dependency among the items, the topic of the passage still might produce an effect on the overall performance of the examinee. An examinee who is unfamiliar with the topic of the passage is likely to have a more difficult time with all of the items, where an examinee who is familiar with the topic will be able to read it more quickly and retain more details in working memory.

One trick that can be used in such situations is to introduce a variable, often called *Context*, that represents familiarity with the topic of the passage (or other stimulus material). This variable is made a parent of all of the observable outcome variables in just that task. Thus, even after conditioning on the proficiency variable, the observable outcomes within this task are still dependent. As the dependent outcomes are all within a single evidence model, the local independence property in the Bayes net model is not violated at the level of tasks (Note that *Context* is associated with the student, much as a proficiency variable. We will discuss shortly why it is not included in the proficiency model). The *Context* variable approach can also be used as a mathematical device to induce dependency among observables even when the cognitive model of topic familiarity is less appropriate. Examples are ratings of the same performance by multiple judges, ratings of several aspects of the same complex performance, and observables that share the same format or come from the same work product.

Example 6.5 (Story Problem in the Math Quiz (Example 6.1 continued)). *Story problems are popular with Math instructors because they test the student's ability to recognize and apply the mathematical principle being tested in real world circumstances. On the other hand, the story of the story problem also introduces irrelevant information, namely, the topic and the details of the story, into the problem solving experience. If multiple problems depend on the same story, then how well the student comprehended the background story can have an effect on all of the items related to the story.*

Suppose that in the math quiz , Item 3 and Item 4 are story problem items that depend on the same story. In this case, we can expect that if the *student has trouble understanding the story, then both Item 3 and Item 4 will be affected. To model this relationship, we add a node Context to the graph of Fig. 6.1, and make it a parent of both Item 3 and Item 4. Figure 6.5 shows the results. The Context variable takes two possible values* `U`*, or* `Unfamiliar`*, for students who have difficulty relating to the content of the story, and* `F`*, or* `Familiar`*, for students who do not. The CPT for Item 3 and Item 4 need to be modified to take the new parent variable into account. The difference between the probabilities for* `F` *and* `U` *for given θ is the strength of the context effect. For each level of θ, we made the conditional probability of a correct response about .10 higher if Context =* `F` *and about .10 lower if Context =* `U`*. Table 6.3 shows the modified values. Finally, we assign a 50-50 probability to the two states of the Context variable.*

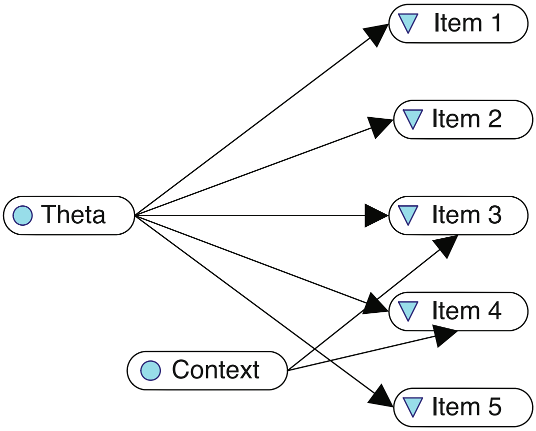


Fig. 6.5 Five item IRT model with local dependence
Reprinted with permission from ETS.

Call the original version of the math quiz with no story problem (as described in Example 6.1) Form A. Call the variant version with the added story problem Form B. It is interesting to compare what the inference about a student would be on the two forms.

*Consider again the responses of the student Ali, who got Item 2 (not in the set) and Item 3 (in the set) correct. Figure 6.6*a *shows the inferences graphically. Note that when only one item from the set is observed, that the effect of the Context variable is averaged out and the inferences are the same on both Forms.*

*Now consider the response vector of Yuri, who got Item 3 and Item 4 correct. Figure 6.6*b *shows the inferences on Form B. Comparing the posterior distribution of θ here with the one from Form A with the same responses (Fig. 6.3*b*), the posterior distribution in Form B is a little less concentrated*

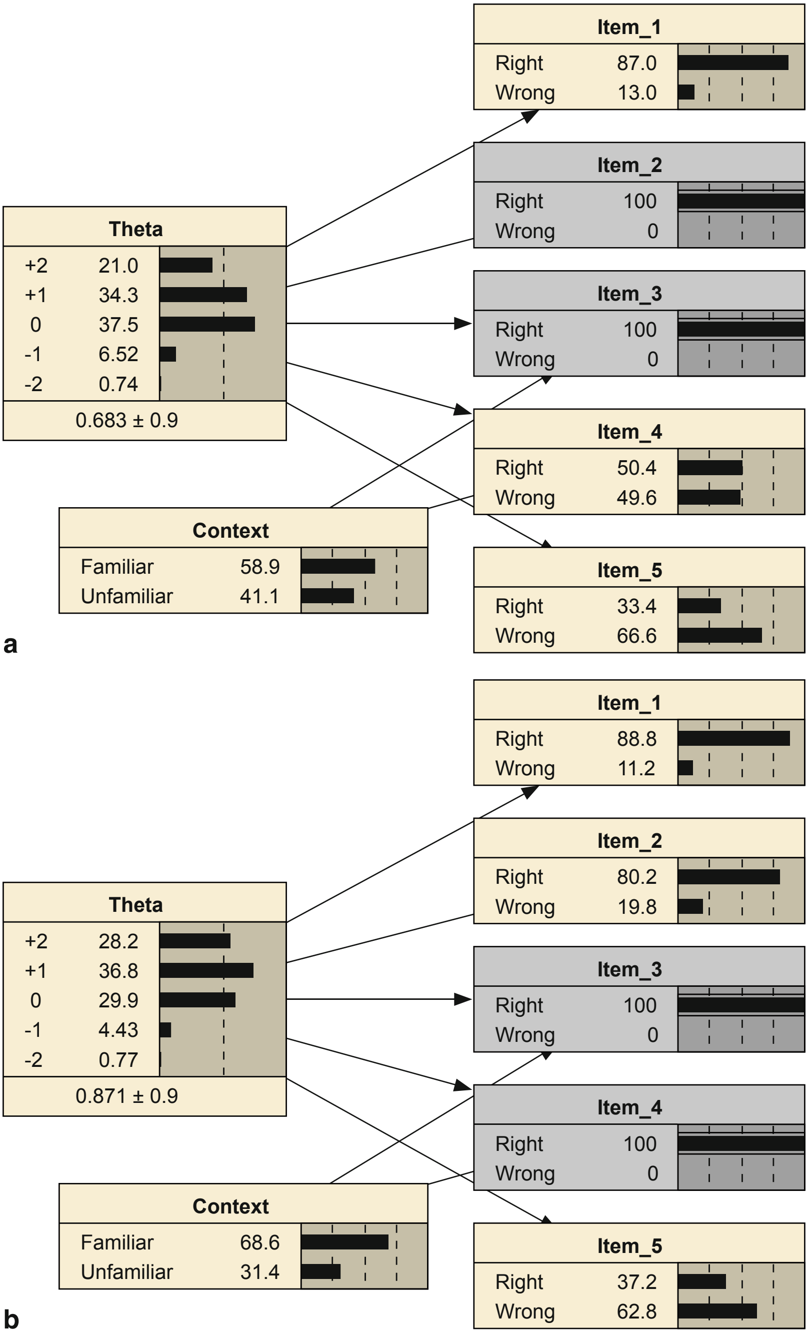


Fig. 6.6 **a** Student with Item 2 and Item 3 correct with context effect. **b** Student with Item 3 and Item 4 correct with context effect

Reprinted with permission from ETS.

Table 6.3 New potentials for Item 3 and Item 4, conditioned on *Context*

Parents		Item 3		Item 4	
θ	Context	`Right`	`Wrong`	`Right`	`Wrong`
−2	F	0.2192	0.7808	0.1601	0.8399
	U	0.0192	0.9808	0.0001	0.9999
−1	F	0.3689	0.6311	0.2480	0.7520
	U	0.1689	0.8311	0.0480	0.9520
0	F	0.6000	0.4000	0.4208	0.5792
	U	0.4000	0.6000	0.2208	0.7792
1	F	0.8311	0.1689	0.6622	0.3378
	U	0.6311	0.3689	0.4622	0.5378
2	F	0.9088	0.9120	0.8773	0.1227
	U	0.7088	0.2192	0.6773	0.3227

and has not moved quite as far from the prior distribution. The Context variable here offers an alternative explanation for Yuri's performance, thus the amount of evidence that the two observations provide jointly about θ is less than it is in Form A.

This example makes sense from an evidentiary perspective. With the two independent items, the only possible explanation for correct answers is increased proficiency (higher θ). Therefore all of the evidence goes toward the proficiency. When the two items are part of the same larger task (in this example, the story problem and its two items form a task), then the task specific *Context* variable forms an alternative explanation. This diverts some of the evidence and so the joint evidence from dependent observations is less than that from independent observations. Chapter 7 talks about evidence in a more formal way. For context effects specifically, Chap. 10 discusses testing for their presence, and Chaps. 14 and 15 show how they arose naturally from task design and are estimated from pilot data in the Biomass example.

The exact degree to which evidence is diverted depends on the relative sizes of the influences from the proficiency (θ) and *Context* variables on the observable outcomes. If the *Context* variable has a large influence on the outcomes then the decrease in evidence will be large. If the *Context* variable has a more modest effect, then the decrease in evidence will be more modest as well.

This example illustrates some of the flexibility that makes Bayesian networks attractive as a model for assessments. Grouping observables that depend on common stimulus together into a task supports more complex models of dependencies among those observables. Recall that standard IRT requires the rather strong local independence property that all items (observables) are conditionally independent given the single proficiency variable. The Bayes net model illustrated here, like the IRT testlet model (Bradlow et al. 1999) uses the weaker local independence property that observables (items) from different tasks are conditionally independent given the proficiency variable(s).

Within the evidence model for a single task, there is considerable freedom as to how to model the dependence among observables. The Context variable model described here is just one possibility; Almond et al. (2006b) compare several possible models. Ideally, domain experts and psychometricians should pick an evidence model for each task which is based on a theory about how students solve the problem. In practice, the Context variable model is often used because it is easy to articulate, and it has roughly the right effect of decreasing the joint evidence from observables from the same task.

The *Context* variable has a peculiar place in the evidence-centered assessment design (ECD) framework because it is neither a proficiency variable residing in the proficiency model nor is it an observable outcome variable residing in the evidence model for a particular task. Instead it represents “proficiency” variable that is local to a particular task. As such it resides in the evidence model for that task. Unlike conventional proficiency variables, we are usually not interested in its value; rather it is a nuisance variable whose value must be estimated along the way, then marginalized out, in order to estimate the proficiency variables whose values form the basis of the assessment’s claims.

The interpretation of the *Context* variable as a proficiency specific to the task highlights the problems that can arise when the distribution of this variable is not uniform across the tested population. For example, if the story in the story problem was about boat racing, then students who lived near large bodies of water and students from wealthy families who vacation near large bodies of water would be expected to have `F` *Context* more often than poor students living in inland communities. This would be a poor choice of story for a large, standardized test because it raise issues of fairness. On the other hand, a classroom teacher might reasonably expect all of the students to be familiar with the story context because they had just completed reading a nautical adventure story in their literature class.

In the math quiz example, there is another possible cognitive explanation for the *Context* variable, namely that it represents reading comprehension. If all the other items on the math quiz are expressed algebraically, then reading comprehension would be a common skill between these two items. While familiarity with boats is not a proficiency that is related to the claims of the math quiz, the ability to recognize mathematical terms embedded in natural language could be. As such, it is probably better placed as a second variable in the proficiency model, which will be reported on, than as a variable local to the task. Both interpretations have the same effect on the evidence for the overall mathematics proficiency, but the latter supports inferences about the specific skill of extracting mathematical information from natural language.

6.3 Compensatory, Conjunctive, and Disjunctive Models

The simple IRT example was restricted to a single proficiency variable parent. Part of the flexibility of the Bayesian network is that it allows for multiple proficiency variable parents. However, as soon as there are more than one parent variable, the question arises "how do the skills required for this task interact to make the probability of a successful outcome?" There are three commonly used models for CPT where there are more than one proficiency variable as the parent of an observable outcome:

- *Conjunctive Distribution*—This is the case where all skills are necessary for a high probability of a successful outcome. Because this corresponds to a logical "and" of the input proficiencies, this model is sometimes called a *noisy-and* model. The "noise" comes because the relationship is not perfect but probabilistic. If the proficiency variables have more than two levels, then the conjunctive model assumes that the student behaves at the level of the weakest skill. For this reason, the model is also sometimes called a *noisy-min* model.
- *Disjunctive Distribution*—This is the case where the parent skills represent alternative ways to solve the task. Presumably examinees choose the approach based on their strongest skills and the probability of success is determined by the strongest skill. This is sometimes called a *noisy-or* or *noisy-max*.[4]
- *Compensatory Distribution*—In this model having more of one skill will "compensate" for having less of another. In this case the probability of success is determined by the sum of the skills, possibly a weighted sum. This is sometimes called an *additive model*.

When using Bayesian networks, there is no need to choose a single model for skill interaction that holds across all observable outcome variables. The choice of model is determined for each obervable by how the CPT for that variable is set up. The analyst can mix and match any of the three types of distribution in a single Bayesian network, or build other distributions for special situations. Chapter 8 describes some possible parameterizations for this kind of model. This section compares simple examples of each type of model to help develop intuition for when and where they should be used.

Figure 6.7 shows the three kinds of conditional probability distribution in a series of simple parallel models. In this directed hypergraph notation the variables are shown as rounded rectangles, and the CPT are shown as square boxes. Each of the boxes is labeled with an icon depending on the type of

[4] Although in educational testing conjunctive models are more common than disjunctive model, in other applications of Bayes nets noisy-or models are more common than noisy-and, and there is a considerable literature on the topic (for example, Díez 1993; Srinivas 1993; Pearl 1988). Fortunately, the two models are symmetric so translating the results is fairly straightforward.

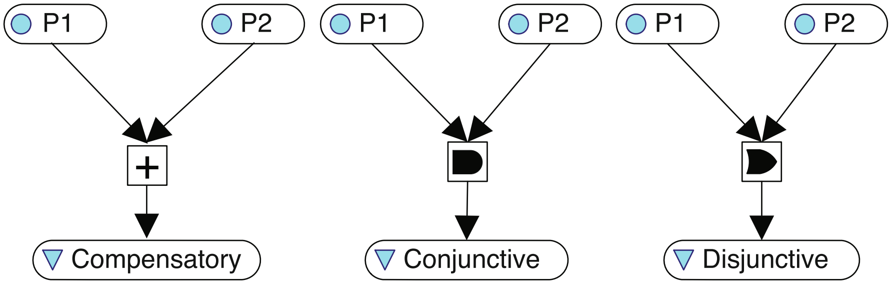


Fig. 6.7 Three different ways of modeling observable with two parents Reprinted with permission from ETS.

distribution. The plus sign is used for the compensatory (additive) model. The symbols in the boxes for the conjunctive and disjunctive distributions are the symbols used for AND-gates and OR-gates in logical diagrams. The advantage of this directed hypergraph notation is that the type of relationship is obvious from the picture; in the more usual directed graph notation, one needs to open the CPT to determine the type of distribution.

The three models are designed to be close parallels of each other. They have the following characteristics in common:

- There are two proficiency variables as parent nodes (*P1* and *P2*), and the two proficiencies are independent of each other (before making observations).
- The priors for the proficiency nodes are the same for the three models with a probability of 1/3 for each of the high (`H`), medium (`M`), and low (`L`) proficiency states.
- The initial marginal probability for observable variable *Obs* is the same for the three models (50/50). (Fig. 6.8)

The difference comes in how the conditional probability distribution $P(Obs|P1, P2)$ is set up. Table 6.4 gives the probabilities for the three distributions. The easiest way to approach this table is to start in the middle with the row corresponding to both parent variables in the middle state. For the compensatory distribution when either skill increases, the probability of success increases by .2, and when either skill decreases, the probability of success decreases by a corresponding amount. For the conjunctive distribution both skills must increase before the probability of success increases, but a drop in either skill causes a decline in probability. The opposite is true for the disjunctive distribution. The probability of the middle category needs to

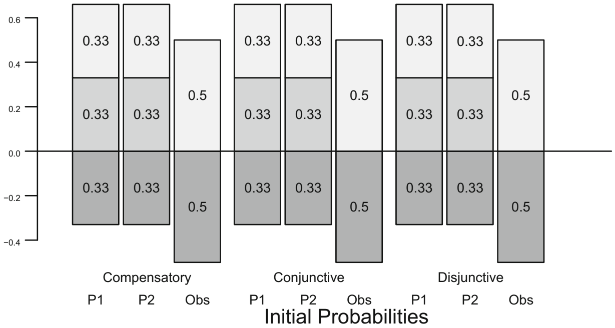


Fig. 6.8 This figure shows the probabilities for all three models side by side. Each *bar* represents the marginal probability of one of the variables in one of the models. The length of the fragment give the probability of a particular state from best (*highest* and *lightest*) to worst (*lowest* and *darkest*). The *bars* are *offset* so that the extent below the *line* gives the probability of being in the lowest category and the extent above the line give the probability of being above the lowest category. The $y-$axis shows amount of probability of being below the line as negative and the probability of being above as positive

Reprinted with permission from ETS.

be adjusted slightly to get the marginal probability of success to be .5 for all three distributions.

Table 6.4 Conditional probabilities for the three distributions.

Parent state		P(*Obs* = `Right`)		
P1	*P2*	Compensatory	Conjunctive	Disjunctive
`H`	`H`	0.9	0.9	0.7
`H`	`M`	0.7	0.7	0.7
`H`	`L`	0.5	0.3	0.7
`M`	`H`	0.7	0.7	0.7
`M`	`M`	0.5	0.7	0.3
`M`	`L`	0.3	0.3	0.3
`L`	`H`	0.5	0.3	0.7
`L`	`M`	0.3	0.3	0.3
`L`	`L`	0.1	0.3	0.1

Obs is the observable outcome variable in each of the three models

Effects of Evidence

Suppose we observe the value `Right` for the outcome variable *Obs* in all three models. Figure 6.9a shows the posterior probabilities after adding this evidence. In all three cases, the probability mass shifts toward the higher states, however, more mass remains at the `L` level in the disjunctive model. While the compensatory and conjunctive models have the same probability for the low state, the effect is slightly different for the highest state, here the compensatory model shifts slightly more probability mass toward the highest state. These minor differences are as much a function of the adjustments to the probabilities needed to get the difficulties to match as they are differences in the way the three distribution types behave.

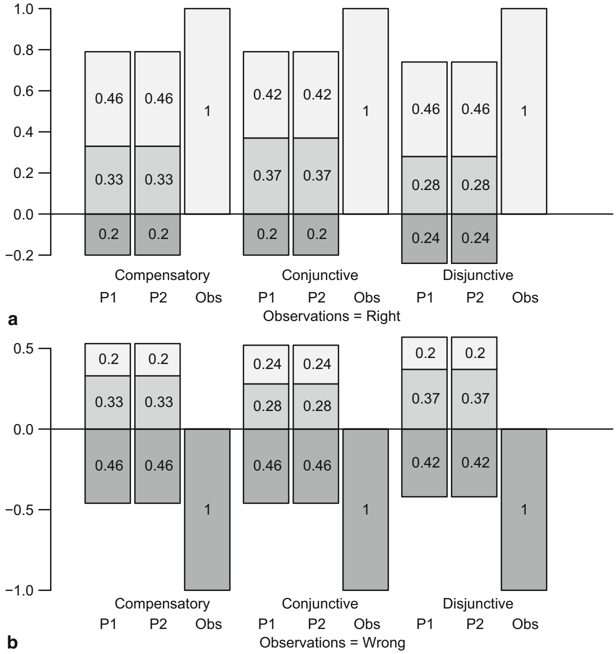


Fig. 6.9 **a** Updated probabilities when *Observation* = `Right`. **b** Updated probabilities when *Observation* = `Wrong`
Reprinted with permission from ETS.

If the observed outcome value is `Wrong` instead of `Right` similar effects work in the opposite directions. Figure 6.9b shows the posterior probabilities for this case. Now the conjunctive model has the highest probability for the `H` high states. Other conclusions follow as well with the `H` and `L` low proficiency states and conjunctive and disjunctive distributions switching roles.

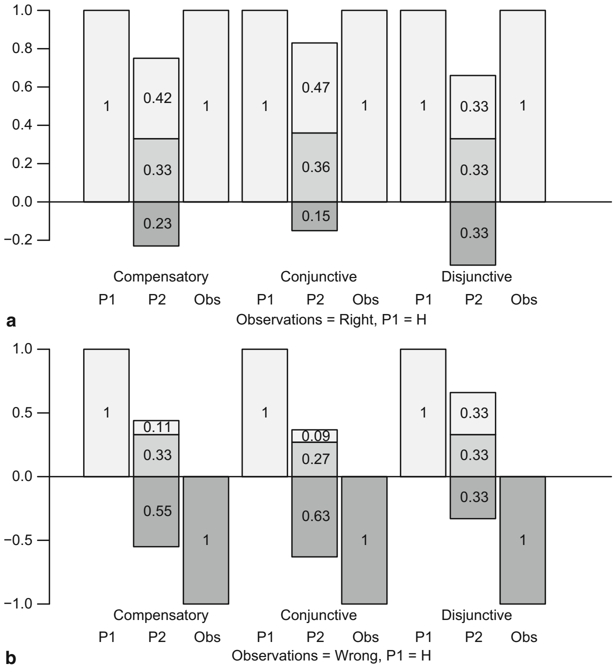


Fig. 6.10 a Updated probabilities when *P1* = `H` and *Observation* = `Right`. **b** Updated probabilities when *P1* = `H` and *Observation* = `Wrong` Reprinted with permission from ETS.

Effect of Evidence When One Skill is Known

When there are two parent proficiencies for an observable outcome variable, what is known about one proficiency will affect inferences about the other. Suppose that *P1* is easy to measure and its state can be determined almost exactly by an external test. How does knowledge about *P1* affect inferences about *P2* under each of the three types of distribution?

Assume that we know (through other testing) that *P1* is in the `H` state. Figure 6.10a shows the posterior distribution when the observable is `Right` and Fig. 6.10b shows the posterior distribution when the observable is `Wrong`. The most startling effect is with the disjunctive distribution. The fact that *P1* is at the `H` is a perfectly adequate explanation for the observed performance. As can be seen from Table 6.4, when *P1* is at the `H` state, the probability of success is the same no matter the value of *P2*. Therefore, if *P1* = `H` the task provides no information whatsoever about *P2*.

The effect of the additional information about *P1* in the conjunctive distribution is the opposite of its effect in the disjunctive distribution. Given that

P1 is at the highest state, the second proficiency *P2* governs the probability of success. Therefore the distributions in Fig. 6.10a and b are very different. The compensatory distribution shows a more moderate change, lying between the posteriors of the conjunctive and disjunctive distributions.

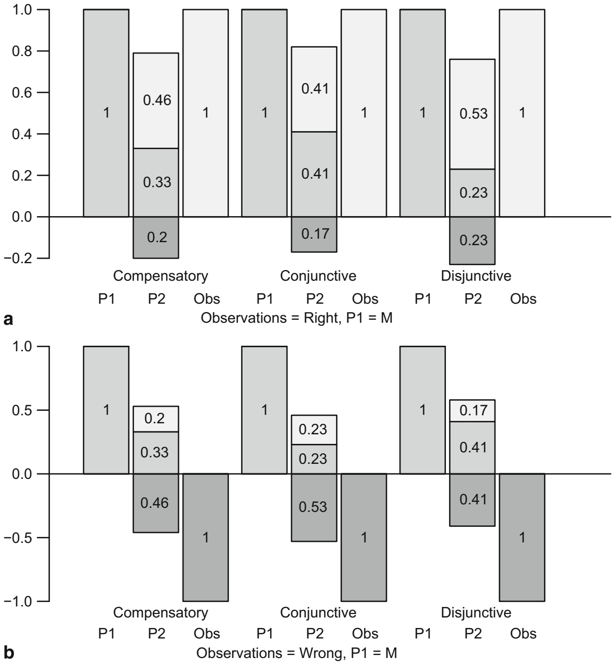


Fig. 6.11 a Updated probabilities when *P1* $=$ `M` and *Observation* $=$ `Right`. **b** Updated probabilities when *P1* $=$ `M` and *Observation* $=$ `Wrong`
Reprinted with permission from ETS.

Now assume that we know (through other testing) that *P1* is only in the `M` state. Figure 6.11a shows the posterior distribution when the observable is `Right` and Fig. 6.11b shows the posterior distribution when the observable is `Wrong`. Starting with the compensatory distribution, note that the effect is similar to when the value of *P1* was `H`, only shifted a bit toward high values of *P2*. The conjunctive distribution gives a big swing (between the posteriors after the two different observable values) for the lowest state, but provides no information to distinguish between the two higher states of *P2*. This is because the state of `M` for *P1* provides an upper bound on the ability of the student to perform the task. Similarly, in the disjunctive distribution the evidence can distinguish between the highest state of *P2* and the others, but provides no information to distinguish between the lower two states.

6.4 A Binary-Skills Measurement Model

The examples in this chapter so far have been completely artificial. The final section in this chapter explores a real example. Any real example starts with a cognitive analysis of the domain, which is a lot of work. For this example we will borrow an extensive cognitive analysis of the domain of mixed-number subtraction found in Tatsuoka (1984) and Tatsuoka et al. (1988). This example was used by Tatsuoka (1990) as part of the development of the rule space method, but the description shown here comes from the Mislevy (1995b) adaptation of this problem to Bayesian networks.

Section 6.4.1 describes the results of the cognitive analysis of this domain (Tatsuoka 1984; Tatsuoka et al. 1988). Section 6.4.2 derives a Bayes net model based on the cognitive analysis. Section 6.4.3 describes how the model is used to make inferences about students.

6.4.1 The Domain of Mixed Number Subtraction

Tatsuoka (1984) begins with cognitive analyses of middle-school students' solutions of mixed-number subtraction problems. Klein et al. (1981) identified two methods that students used to solve problems in this domain:

- *Method A:* Convert mixed numbers to improper fractions, subtract, and then reduce if necessary.
- *Method B:* Separate mixed numbers into whole number and fractional parts; subtract as two subproblems, borrowing one from minuend whole number if necessary; then simplify and reduce if necessary.

The cognitive analysis mapped out flowcharts for applying each method to items from a universe of fraction subtraction problems. A number of key procedures appear, which any given problem may or may not require depending on the features of the problem and the method by which a student might attempt to solve it. Students had trouble solving a problem with Method B, for example, when they could not carry out one or more of the procedures an item required. Tatsuoka constructed a test to determine which method a student used to solve problems in the domain[5] and which procedures they appeared to be having trouble with.

This analysis concerns the responses of 325 students, whom Tatsuoka (1984) identified as using Method B, to 15 items in which it is not necessary to find a common denominator. These items are a subset from a longer 40-item test, and are meant to illustrate key ideas from Bayes nets analysis in a realistic, well-researched cognitive domain. Instructional decisions in operational work were based on larger numbers of items. Figure 6.12 shows the proficiency model for the following skills:

[5] Their analyses indicated their students tended to use one method consistently, even though an adult might use whichever strategy appears easier for a given item.

Skill 1 Basic fraction subtraction.
Skill 2 Simplify/reduce fraction or mixed number.
Skill 3 Separate whole number from fraction.
Skill 4 Borrow one from the whole number in a given mixed number.
Skill 5 Convert a whole number to a fraction.

All of these skills are binary, that is a student either has or does not have the particular skill. Furthermore, there is a prerequisite relationship between *Skills 3* and *4*: a student must acquire *Skill 3* before acquiring *Skill 4*.

In the rule space method (Tatsuoka 1984; Tatsuoka 1990) it is traditional to express the relationship between the proficiency variables and the observable outcome variables (in this case, whether each problem was correct or not), through the use of a Q-matrix (Sect. 5.5). Table 6.5 shows the Q-matrix for the mixed-number subtraction test. All of the models in this example are conjunctive—all skills are necessary to solve the problem. Note that several groups of items have identical patterns of required skills. Following ECD notation, we call a common pattern an *evidence model*. The column in the table labeled EM shows the items' associations with the six evidence models that appear in the example.

Table 6.5 Q-Matrix for the Tatsuoka (1984) mixed number subtraction test

		Skills required					
Item	Text	1	2	3	4	5	EM
6	$\frac{6}{7}-\frac{4}{7}$	x					1
8	$\frac{3}{4}-\frac{3}{4}$	x					1
12	$\frac{11}{8}-\frac{1}{8}$	x	x				2
14	$3\frac{4}{5}-3\frac{2}{5}$	x		x			3
16	$4\frac{5}{7}-1\frac{4}{7}$	x		x			3
9	$3\frac{7}{8}-2$	x		x			3
4	$3\frac{1}{2}-2\frac{3}{2}$	x		x	x		4
11	$4\frac{1}{3}-2\frac{4}{3}$	x		x	x		4
17	$7\frac{3}{5}-\frac{4}{5}$	x		x	x		4
20	$4\frac{1}{3}-1\frac{5}{3}$	x		x	x		4
18	$4\frac{1}{10}-2\frac{8}{10}$	x		x	x		4
15	$2-\frac{1}{3}$	x		x	x	x	5
7	$3-2\frac{1}{5}$	x		x	x	x	5
19	$7-1\frac{4}{3}$	x		x	x	x	5
10	$4\frac{4}{12}-2\frac{7}{12}$	x	x	x	x		6

With five binary skills there are 32 possible proficiency profiles—assignment of values to all five skills. However, the prerequisite relationship reduces the number of legal profiles to 24, since combinations with *Skill 3* = `No` and *Skill 4* = `Yes` are impossible. Not all 24 profiles can be identified using the data from the test form described in Table 6.5. For example, there are no tasks which do not require *Skill 1*, therefore this form provides no evidence for distinguishing among the twelve proficiency profiles which lack *Skill 1*. This does not make a difference for instruction, as a student lacking *Skill 1* would be tutored on that skill and then retested. The test was designed to determine which of the more advanced skills a student might need further instruction in.

Up to this point, the analysis for the Bayesian network model is the same kind of analysis that is done for the rule space method (Tatsuoka 1990). It is in accounting for departures from this ideal model that the two methods differ. Rule space looks at ideal response vectors from each of the 24 skill profiles and attempts to find the closest match in the data. The Bayesian method requires specifying both a probability distribution over the possible proficiency profiles (a proficiency model) and a probability distribution for the observed outcomes given the proficiency parents. It is then in a position to calculate a posterior distribution over each examinee's proficiencies given their observed responses. The next section describes how that is done in this example.

6.4.2 A Bayes Net Model for Mixed-Number Subtraction

The ECD framework divides the Bayes net for this model into several fragments. The first is the proficiency model fragment (PMF) containing only the variables representing the skills. Then there are 15 separate evidence model fragments (EMFs), one for each item (task) in the assessment. In order to specify a Bayes net model for the mixed-number subtraction assessment, we must specify both the graphical structure and the condition probability tables for all 16 fragments.

We start with the proficiency model. There are only five binary proficiency variables, making the total number of possible skill profiles 32. As this is a manageable size for a clique, we will not worry about asking the experts for additional conditional independence statements to try to reduce the treewidth of the proficiency model. Instead, we will just choose an ordering of the proficiency variable and use that to derive a recursive representation for the joint probability distribution.

Mislevy (1995b) chose the order: *Skill 1, Skill 2, Skill 5,* and finally *Skill 3* and *Skill 4*. We leave those two for last because of the prerequisite relationship between them which requires special handling. Putting *Skill 1* first makes sense because normally this skill is acquired before any of the others. This is a kind of a soft or probabilistic prerequisite, as opposed to the relation

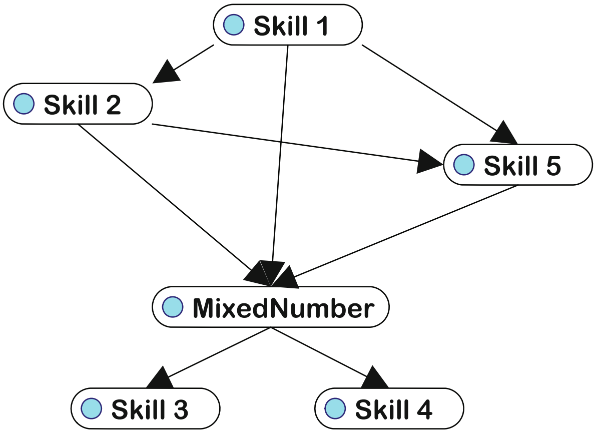


Fig. 6.12 Proficiency model for Method B for solving mixed number subtraction Reprinted with permission from ETS.

between *Skill 3* and *Skill 4* which is a hard prerequisite; there are no cases where *Skill 4* is present and *Skill 3* is absent.

This means that there are only three possible states of the two variables *Skill 3* and *Skill 4*. To model this, we introduce a new variable *MixedNumber* which has three possible states: (0) neither *Skill 3* nor *Skill 4* present, (1) *Skill 3* present but *Skill 4* absent, and (2) both *Skill 3* and *Skill 4* present. The relationship between the *MixedNumber* variable and *Skill 3* and *Skill 4* are logical distributions which consist solely of ones and zeroes.

Figure 6.12 gives the graphical structure for the proficiency model. The structures of the EMFs are given by the rows of Table 6.5. First note that several rows in that table are identical, in that they use exactly the same skills. Items 9, 14, and 16, for example, all requires Skills 1 and 3. We have assigned each unique row an *evidence model* . Thus, we really only need to create six EMFs to build the complete model for this short assessment. Items 9, 14, and 16 will all use EMF 3. Later on, we will assign different probability tables to the EMFs for different tasks. When we do that we will create individual *links*—task specific versions of the evidence model—for each task (see Chap. 13 for details).

The *Q*-Matrix (Table 6.5) provides most of the information necessary to build the EMFs. In particular, the parents of the observable outcome variable (correct/incorrect for the item) are variables checked in the *Q*-Matrix. The one additional piece of information we need, supplied by the cognitive experts, is that the skills are used conjunctively, so the conjunctive distribution is appropriate. Figure 6.13 shows the EMFs for evidence models 3 and 4.

After constructing the six different evidence model fragments and replicating them to make links for all 15 items in the test, we have a collection of 16 Bayes net fragments: one for the proficiency model and 15 (after repli-

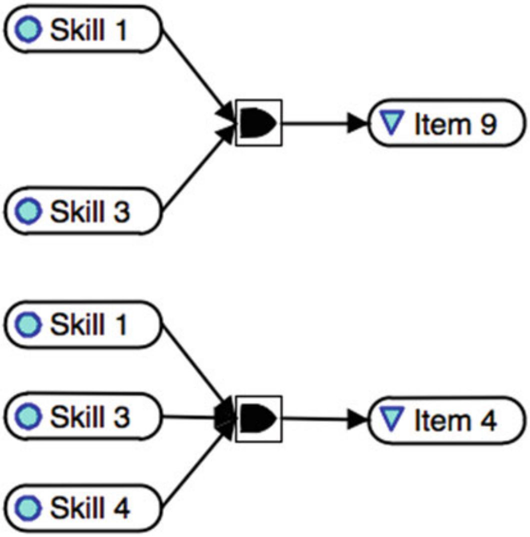


Fig. 6.13 Two evidence model fragments for evidence models 3 and 4 Reprinted with permission from ETS.

cation)for evidence model fragments. We can catenate them to produce the *full Bayesian model* for the mixed number subtraction test. This is shown in Fig. 6.14. Although we could use the computational trick described in Sect. 5.4.1 to calculate probabilities in this model just joining one EMF at a time to the PMF, the full Bayesian model is small enough to be easily handled by most Bayes net programs.

All that remains to complete the model is to build a CPT for each variable in the model. First we must build a CPT for each variable in the proficiency model. Then we must build a CPT for each observable variable in the Evidence Model Fragments. (In this example, all variables in the evidence models are either borrowed from the proficiency model, and hence do not require a CPT, or are observable. If we had other evidence model variables, like the *Context* variable above, they would require CPTs as well.)

There are basically two sources for the numbers, expert opinion and data. In this particular case, Tatsuoka (1984) collected data on 325 students. As mentioned above, Chap. 11 (see also Mislevy et al. (1999a)) tells that part of the story. The numbers derived from those calculations are the ones used below.

However, even with only the expert opinion to back it up, the model is still useful. In fact, the version used in Mislevy (1995b) uses only the expert numbers. At first pass, we could simply assign a probability of .8 for success on an item if a student has all the prerequisite skills, and a probability of .2 for success if the student lacks one or more skills. Similarly, we could assign

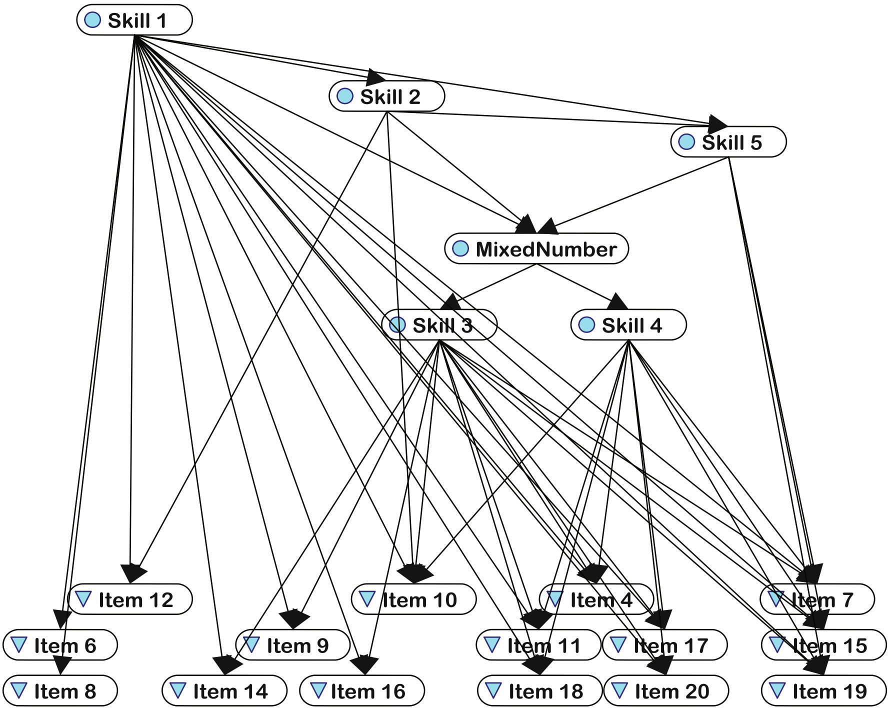


Fig. 6.14 Full Bayesian model for Method B for solving mixed number subtraction Reprinted from Almond et al. (2010) with permission from ETS.

a prior probability of around 0.8 for students having all the parent skills and a probability of around 0.2 when they lack one or more of the parent skills. When there is more than one parent, or more than two states for the skill variable (e.g., the *MixedNumber* variable) we interpolate as appropriate.

While such a Bayes net, built from expert numbers, might not be suitable for high stakes purposes, surely it is no worse than a test scored with number right and a complicated weighting scheme chosen by the instructor. In fact, it might be a little better because at least it uses a Bayesian scheme to accumulate the evidence (Exercise 7.13). Furthermore, if there are several proficiency variables being estimated, the Bayes net model will incorporate both direct evidence from tasks tapping a particular proficiency and indirect evidence from tasks tapping related proficiencies in providing an estimate for each proficiency. This should make estimates from the Bayes net more stable than those which rely on just subscores in a number right test.

To give a feel for the structure and the contents of the CPTs behind the following numerical illustrations, let us look at two tables based on the analysis in Chap. 11, one for a proficiency variable and one for an observable outcome variable.

Recall that there are associations among the proficiency variables. It suffices here to say that the skills are positively associated—having high values of parent skills makes it more likely that a student will have a high value for a child skill as well—and to present one example. *Skill 5* is modeled as depending on *Skill 1* and *Skill 2*. This means that there are four conditional probability distributions for *Skill 5*, one for each combination of *Skill 1* and *Skill 2* values. We estimated these conditional probabilities under the constraint that the distributions would be the same for both combinations in which one prior skill is mastered but the other is not. The CPT for *Skill 5* that is built into the network is shown in Table 6.6.

Table 6.6 Conditional probability table for *Skill 5*

Skill 1	*Skill 2*	P(*Skill 5* = `Yes`)	P(*Skill 5* = `No`)
`Yes`	`Yes`	0.748	0.252
`Yes`	`No`	0.469	0.531
`No`	`Yes`	0.469	0.531
`No`	`No`	0.129	0.871

Item 16 is one of three items that uses evidence model 3: The parents of the observable outcome are *Skill 1* and *3*, and the relationship is conjunctive. Again the CPT is composed of four conditional probability distributions. We estimated them under a constraint common in binary skills models, that there would be one distribution when both *Skill 1* and *Skill 3* are `Yes` and a different distribution common to all combinations in which one one or both required skills are `No`. We expect the conditional probability of a correct response to be higher in the first case (The equality constraint of probabilities across the three profiles with a `No` can be examined using methods discussed in Part II). Table 6.7 shows the CPT used in the example.

Table 6.7 Conditional probability table CPT for *Item 16*

Skill 1	*Skill 3*	P(*Item 16* = `Right`)	P(*Item 16* = `Wrong`)
`Yes`	`Yes`	0.910	0.090
`Yes`	`No`	0.170	0.830
`No`	`Yes`	0.170	0.830
`No`	`No`	0.170	0.830

6.4.3 Inferences from the Mixed-Number Subtraction Bayes Net

Regardless of how the numbers get into the Bayesian network, the procedure used to draw inferences from the Bayesian network is the same. There are two common cases: drawing inferences about proficiency variables given the observed outcomes, and making predictions about observable outcomes given the hypothesized proficiency levels. These are both described below. In both cases, the first step is to enter the conditional probability tables into the Bayesian network software and compile the network. The network will then produce the prior probability for a student from this population having each of the proficiencies as well as predictions for each observable variable. Figure 6.15 shows how this looks in the software package Netica (other packages have similar displays).

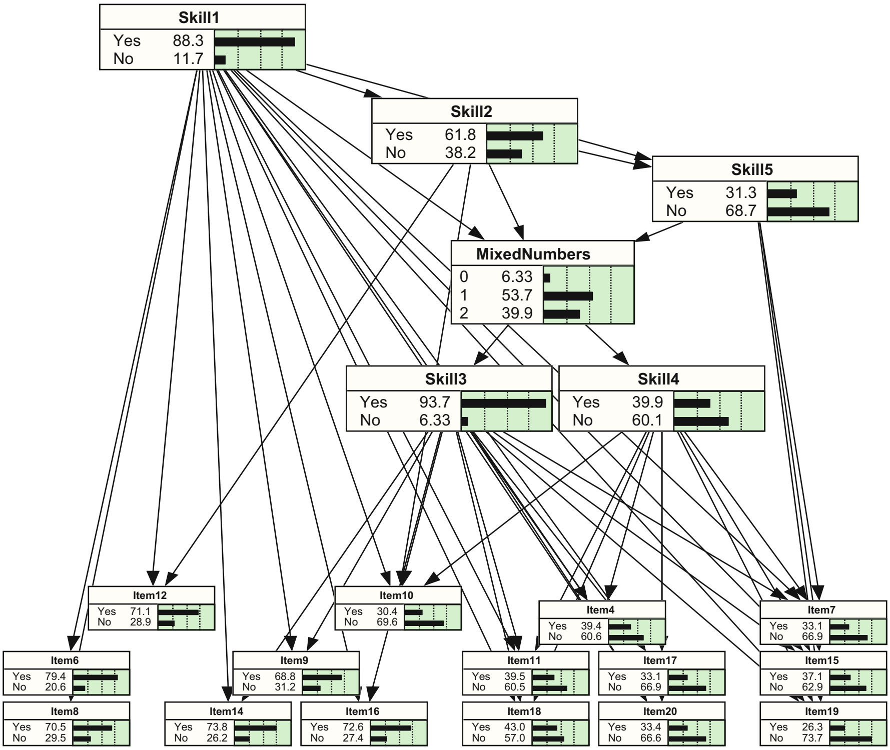


Fig. 6.15 Prior (population) probabilities
Reprinted with permission from ETS.

Scoring a Test

The most basic use of the Bayes net can be described as follows. For each observed outcome, find the corresponding node in the Bayes net and instantiate its value (The details of how to do this differ according to the Bayes net software, but it generally involves clicking on the node and selecting a value from a list of possible states for the variable). The instantiated values are then propagated to the nodes representing the proficiency variables whose marginal distributions are then calculated with some variation of the algorithm in Chap. 5 (Again, depending on the software and chosen options, the propagation could be manual or automatic).

Example 6.6 (Mixed-Number Subtraction Complete Response). *Suppose that a student (whose class uses Method B) takes the mixed-number subtraction test, and gets Items 4, 6, 8, 9, 11, 17, and 20 correct and Items 7, 10, 12, 14, 15, 16, 18, and 19 incorrect (Items 1, 2, 3, 5 and 13 were dropped from the shortened 15 item version of the test). Which skills does this student have?*

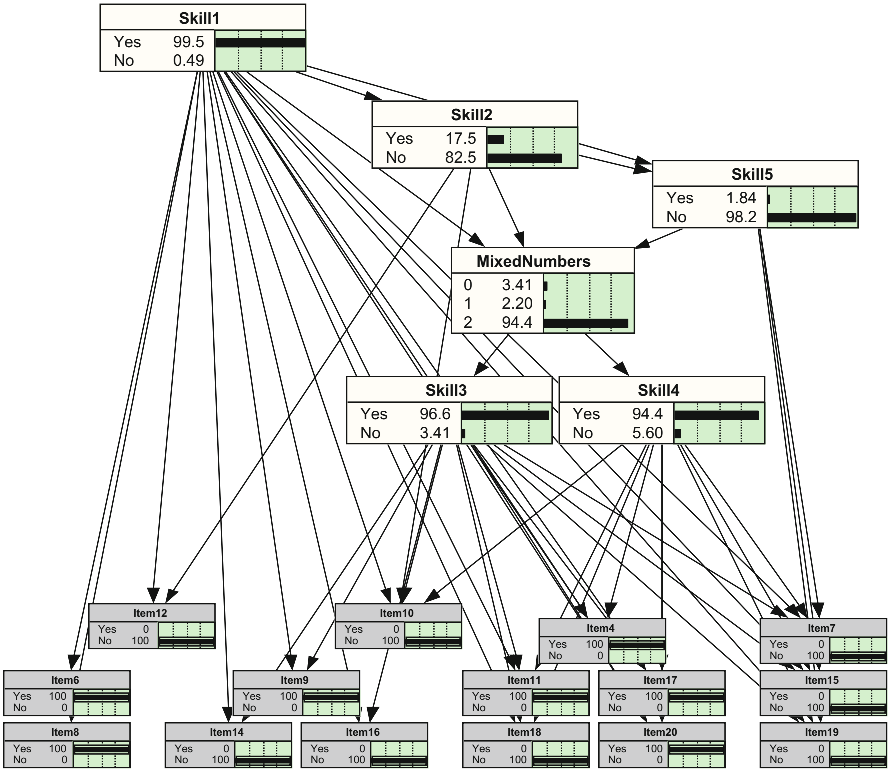


Fig. 6.16 Mixed number subtraction: a sample student
Reprinted with permission from ETS.

Figure 6.16 shows the network after instantiating these observed variables into this network. From this picture, there is a high probability that the student has Skills 1, 3, and 4, but low probability that the student has Skills 2 and 5. The second column of Table 6.8 summarizes the results.

Table 6.8 Posteriors after two sets of observations

Node	Initial probability	Example 6.6	Example 6.7
Skill 1	0.883	0.995	0.999
Skill 2	0.618	0.172	0.587
Skill 3	0.921	0.950	0.991
Skill 4	0.396	0.929	0.839
Skill 5	0.313	0.032	0.215
Item 6	0.794	1.000	1.000
Item 8	0.705	1.000	1.000
Item 12	0.711	0.000	0.712
Item 14	0.731	0.000	0.849
Item 16	0.719	0.000	0.835
Item 9	0.686	1.000	1.000
Item 4	0.392	1.000	1.000
Item 11	0.393	1.000	1.000
Item 17	0.330	1.000	0.584
Item 20	0.332	1.000	0.606
Item 18	0.428	0.000	0.668
Item 15	0.369	0.000	0.356
Item 7	0.329	0.000	0.000
Item 19	0.262	0.000	0.251
Item 10	0.304	0.000	0.000

The test results need not be complete in order to draw inferences; any subset of the observables can be used to draw inferences about the proficiencies (including the empty set, but that will just reproduce the prior population levels for the skills). For observable variables whose values are not observed, we simply leave the node uninstantiated (or select the special value `unknown` depending on the software package). The Bayes net updating algorithm automatically uses only observed values. In fact, it will even provide predictions for the remaining unknown values.

Example 6.7 (Mixed-Number Subtraction, Partial Data). *Suppose that the test is administered on a computer and that a student is midway through completing the test. Suppose further, that the outcomes observed so far are correct results for Items 6, 8, 9, 4, and 11 and incorrect results for Items 7 and 10. Figure 6.17 shows the state of the network after entering this information.*

Even the first half of the test is sufficient to drive the probability of Skill 1 to close to 1.00. The probability for Skills 3 and 4 have increased over the prior values, but the probabilities for Skills 2 and 5 have dropped slightly. The third column of Table 6.8 summarizes the results.

Note that the software automatically produces predictive distributions for the as–yet–unobserved observable variables. Section 7.2 shows an important application of this idea in test selection.

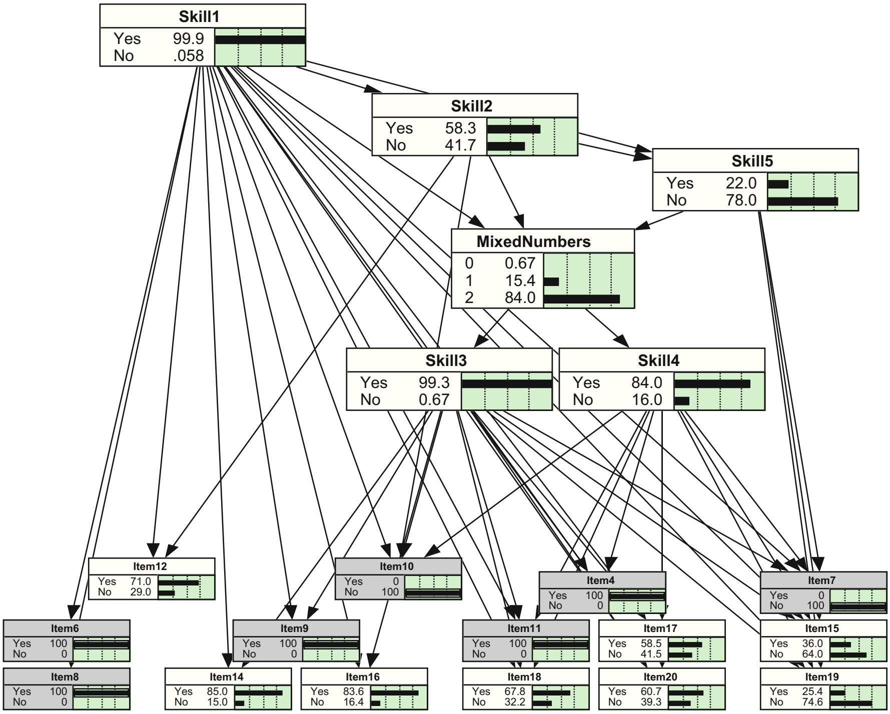


Fig. 6.17 Mixed number subtraction: posterior after 7 Items
Reprinted with permission from ETS.

Predicting Test Results

Although the proficiency variables are not directly observable, we can use the Bayes net software to instantiate them in order to answer hypothetical questions. This is actually a good method for validating a Bayesian network. Show the predictions made from the model to an expert in the subject of the assessment and ask if they are reasonable. Any predictions which seem unusual are potential problems with the model (The techniques discussed in the next chapter can help debug the network).

Turning to the mixed-number subtraction problem which has been the focus of this section, instantiate the value of *Skill 1* to `Yes`. The result can be seen in Fig. 6.18 and the second column of Table 6.9. The probabilities of a correct response to all items increases. This is so for two reasons. First, we know that a requirement for all tasks, *Skill 1*, has been met. Second, the high value of *Skill 1* increases our belief that the student also has the other skills because the skills are positively associated in the proficiency model.

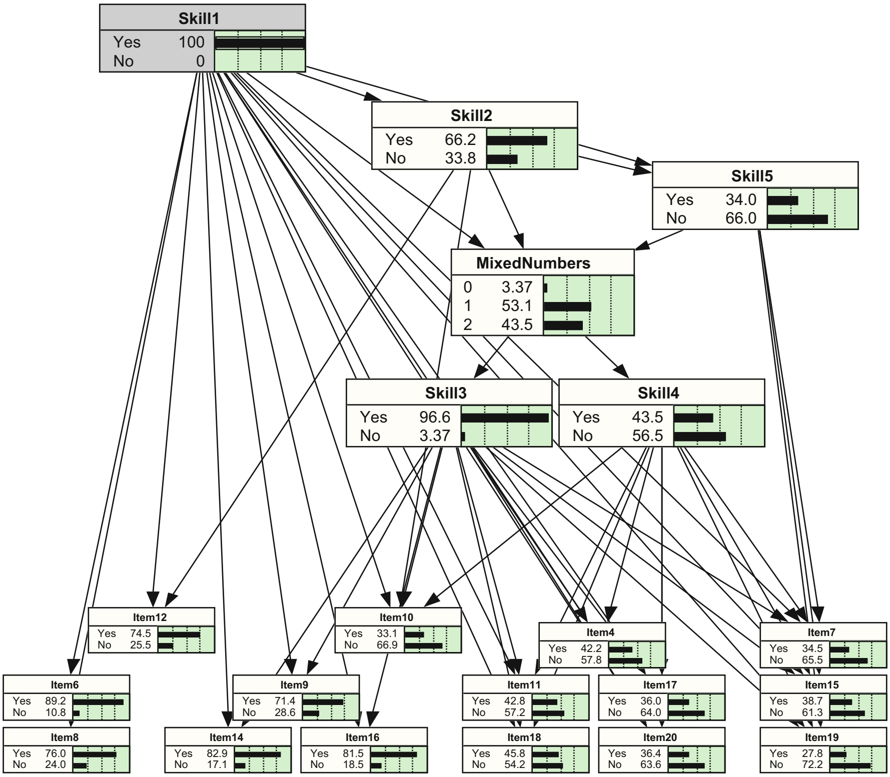


Fig. 6.18 Mixed number subtraction: *Skill 1* = `Yes`
Reprinted with permission from ETS.

Unsurprisingly, conditioning on *Skill 1* = `No` has the opposite result. The result can be seen in Fig. 6.19 and the third column of Table 6.9. Not only does the predictive probability for each item drop, but so does the predictive probability for the remaining skills.

It is possible to condition on any number of proficiency variables (or a complete proficiency profile). As a simple example, consider a student who has *Skill 1* but lacks *Skill 2*. This result can be seen in Fig. 6.20 and the last column of Table 6.9. The probabilities of getting correct responses on Items 6 and 8 increase, as do the probabilities for Items 14 and 17. However, the predictive probability drops for Items 19, 20, and 10. The probability that the student has *Skill 3* drops slightly from the value when conditioning only on *Skill 1*=`Yes`, but there is a substantial drop in the predictive probabilities for *Skill 4* and *Skill 5* decrease.

Table 6.9 Predictions for various skill patterns Subtraction assessment

Node	Initial	*Skill 1*=`Yes`	*Skill 1*=`No`	*Skill 1*=`Yes` & *Skill 2*=`No`
Skill 1	0.883	1.000	0.000	1.000
Skill 2	0.618	0.662	0.289	0.000
Skill 3	0.921	0.956	0.650	0.921
Skill 4	0.396	0.432	0.117	0.156
Skill 5	0.313	0.340	0.110	0.100
Item 6	0.794	0.892	0.053	0.892
Item 8	0.705	0.760	0.289	0.760
Item 12	0.711	0.745	0.452	0.452
Item 14	0.731	0.821	0.049	0.792
Item 16	0.719	0.807	0.049	0.779
Item 9	0.686	0.712	0.488	0.703
Item 4	0.392	0.421	0.178	0.265
Item 11	0.393	0.426	0.140	0.243
Item 17	0.330	0.358	0.117	0.204
Item 20	0.332	0.362	0.103	0.196
Item 18	0.428	0.456	0.220	0.305
Item 15	0.369	0.385	0.246	0.258
Item 7	0.329	0.343	0.223	0.234
Item 19	0.262	0.276	0.154	0.165
Item 10	0.304	0.331	0.098	0.098

6.5 Discussion

By this point in the book, we hope you have acquired enough understanding of how Bayes nets work to try something for yourself. Many of the concepts that are difficult to explain in words and equations are easy to understand with the help of Bayes net software that allows you to manipulate the data and rapidly see the results. We urge you to look at some of the software described in Appendix A and try out the extended example presented in Appendix A.2.

The next chapter looks at some more advanced uses of the basic computation algorithms presented in the previous chapter: explaining scores, selecting tasks in an adaptive test, and test construction. Part II turns to the more difficult question of how to get the numbers into the Bayesian network.

Exercises

6.1. Suppose that a teacher has a classroom of 25 students, all of whom are members of the population for which the model in Example 6.1 was built. Answer the following questions about that model:

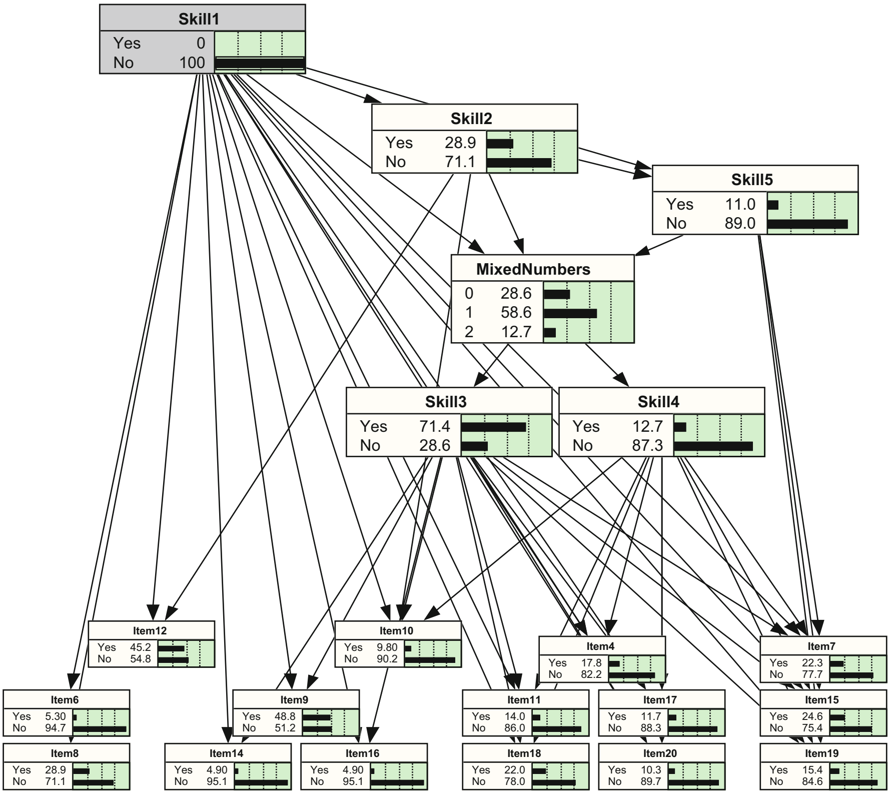


Fig. 6.19 Mixed number subtraction: *Skill 1* = `No`
Reprinted with permission from ETS.

1. For each of the five items on that quiz, how many students are expected to get that item correct?
2. For each item, give a range of values for the number of students who should get that item right such that the teacher should be surprised if the number of right answers falls outside that range.

Hint: The variance of the binomial distribution with parameters n and p is $np(1-p)$.

6.2. Following Examples 6.2 and 6.3, suppose that Ali is a member of Group B. Start with the network instantiated to $\theta = 1$ and enter the data that Ali got Item 2 and Item 3 correct. What changes in this network? Why?

6.3. Suppose that for Example 6.1 the school district defines a student who has a θ value of 0 or higher as meeting the district standards. Add a node to the model for Example 6.1 with two values, which represents whether or not

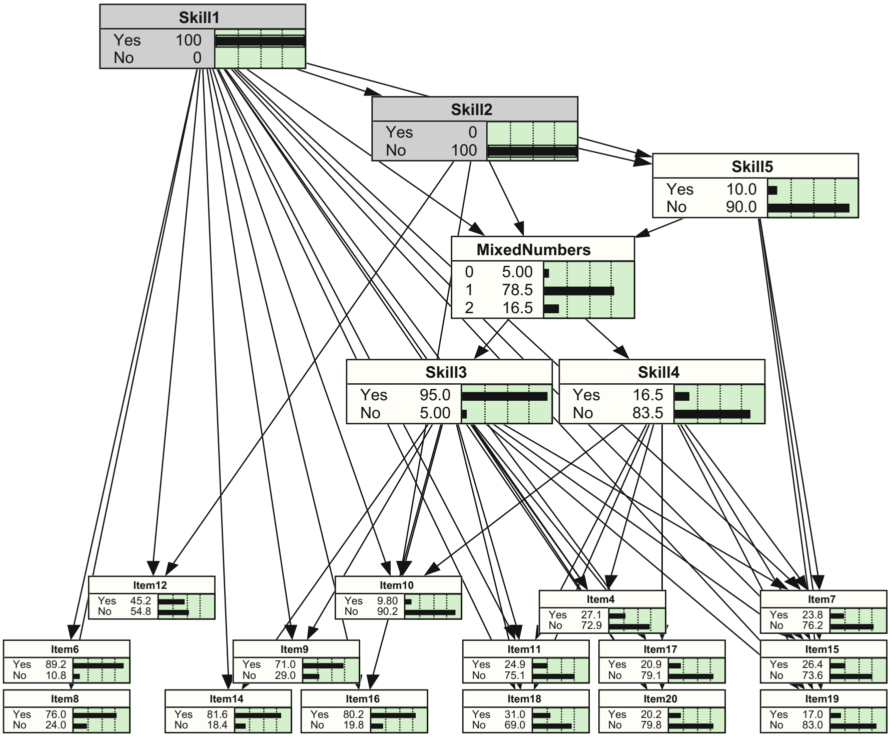


Fig. 6.20 Mixed number subtraction: *Skill 1* = `Yes`, *Skill 2* = `No`
Reprinted with permission from ETS.

the student meets the standards. What are the expected response probabilities for students who are meeting the standards?

6.4. A common practice in tests scored with IRT is to provide the maximum likelihood estimate (MLE) for θ; that is, to find the value of θ that maximizes $\mathrm{P}(\mathbf{X}|\theta)$. Using the assessment and Bayesian network model of Example 6.1, how can you compute the maximum likelihood estimate for θ under the assumption that $\theta \in \{-2, -1, 0, 1, 2\}$? Hint: You will need to change the probability distribution in one of the nodes.

6.5. Suppose in Example 6.5 the teacher covers very similar story problems in class before the quiz. The teacher therefore believes that 95 % of the students should have the value `High` for *Context*. Modify the Bayesian network for Example 6.5 to reflect this. First adjust the CPT for the *Context* node, and then adjust the CPTs for *Item 3* and *Item 4* so that there marginal distribution matches what is shown in Fig. 6.2. You can do this by adding the difference between the observed marginal probability and the desired one to the numbers in Table 6.3. Compare the evidence from this new model to the complete independence model and the model of Example 6.5.

6.6. Section 6.2 noted that *Context* variables are usually placed in the evidence model because they are local to a specific task. Suppose that common background material would render observables from multiple tasks dependent (even after conditioning on the proficiency variables). How can this be modeled without violating the local dependence property?

6.7. (van der Gaag et al. 2004) experts often believe that a Bayesian network should be *isotonic*: as the observable variables move into better states then the probability of a good outcome should increase. In particular, if $\mathbf{e}$ and $\mathbf{e}'$ are two instantiations of observable outcome variables such that for each component $e_k \succeq e_k{}'$ then $\mathrm{P}(S|\mathbf{e}) \succeq \mathrm{P}(S|\mathbf{e}')$. In networks for educational assessment, this means that if the "score" on a task (or item) increases then the probability of proficiency variable being in a high state should also increase. Verifying the monotonicity properties can be an important check that a model is build correctly and accounts for all of the necessary latent variables.

Comparing two probability distributions is a bit tricky, but in the case of the simple IRT Bayes net of Example 6.1 we can simply compare the expected value for *Theta*. We can verify that this network is isotonic by picking a couple of increasing sequences of probability assignments and verify that the expected value increases for each assignment. Using 0 and 1 to represent `Incorrect` and `Correct` observed outcomes, consider the two sequences:

1. $(0,0,0,0,0)$ $(1,0,0,0,0)$ $(1,1,0,0,0)$ $(1,1,1,0,0)$ $(1,1,1,1,0)$ $(1,1,1,1,1)$
2. $(0,0,0,0,0)$ $(0,0,0,0,1)$ $(0,0,0,1,1)$ $(0,0,1,1,1)$ $(0,1,1,1,1)$ $(1,1,1,1,1)$

Calculate the expected value for *Theta* for each response pattern in both sequences. Is this network isotone in θ?

6.8. Repeat Exercise 6.7 for the network with the context variable in Example 6.5.

6.9. Section 6.3 described a simple example of each of three types of distributions for tasks involving three skills. Suppose that through an external test we have established that a particular student is low in *P1*. Figure 6.21 gives the posterior distribution if the value of the task observable is `Wrong`. What does the posterior distribution for *P2* look like for each distribution when the observed outcome is `Right`?

6.10. Recall that in Fig. 6.5 each of the items which has the *Context* variable has a parent. Therefore it is necessary to choose a type for the CPT linking *Theta* and *Context* to the observable. For each of the following scenarios, tell whether it would be more appropriate to model the relationship with a conjunctive, disjunctive, or compensatory distribution:

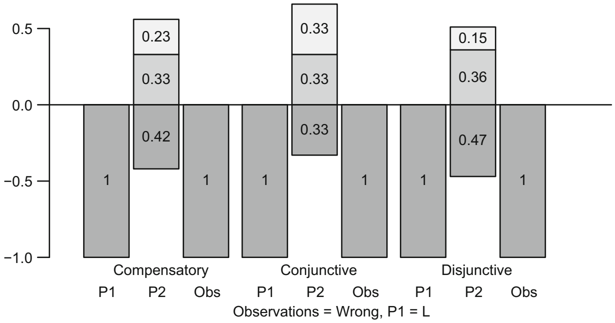


Fig. 6.21 Updated probabilities when *P1* = `L` and *Observation* = `Wrong` Reprinted with permission from ETS.

1. There is an alternative fast method for solving Items 3 and 4 which was taught on a day in which many students were absent. Students who were present that day, or did their make-up homework, should be able to use either the usual or alternative method to solve the problem. The *Context* variable represents knowledge of the alternative solution.
2. Items 3 and 4 are part of an extended complex task with complex instructions. Students who do not understand the instructions are likely to be off task for both problems. The *Context* variable represents understanding the instructions.
3. Items 3 and 4 are both story problems with the subject taken from the field of physics. Students who have studied physics are likely to have previously studied these problem types. Here *Theta* represents the student's math ability and *Context* represents the student's physics ability.

6.11. Section 6.4.2 introduced the artificial *MixedNumber* variable to model the relationship between *Skill 3* and *4*. What must the CPT linking *MixedNumber* and *Skill 3* look like? The CPT linking *MixedNumber* and *Skill 4*? Hint: As this is a logical probability all of the entries must be either 0 or 1.

6.12. In the mixed-number subtraction example, suppose that *Skill 1* was a hard prerequisite for the other skills instead of a soft one. Specifically, suppose that in the CPT for that network we set the probability of having any of the other skills given that *Skill 1* to zero, but make no other changes. Describe how the new model would differ from the one presented in Sect. 6.4 with respect to the following queries:

1. The probability that a student has *Skill j*, $j > 1$, given no other information.
2. The probability that a student will get *Item i* correct, given no information about proficiencies (or no inferences about other items).
3. The probability that a student has *Skill 5*, given that the student got Items 7, 15, and 19 correct.

4. The probability that a student has *Skill 5*, given that the student got Items 7, 15, and 19 incorrect.

6.13. In Example 6.7, why do the predictive probabilities for Items 15 and 19 decrease slightly after observing the first seven items, while the predictive probabilities for Items 14 and 16 increase?

6.14. Consider once again the student who has *Skill 1* but lacks *Skill 2* (Fig. 6.20). Why does the predictive probability for Item 16 go up, but the predictive probability for Item 20 go down?

6.15. What is the expected number of correct score on the mixed-number subtraction test for somebody who has *Skill 1*? Who lacks *Skill 1*? Hint: Use the values in Table 6.9.

6.16. Pick one of the Bayesian network packages listed in Appendix A and use it to build the accident proneness example, Example 3.8. Verify the computations in that example.

6.17. Use your favorite Bayesian network package to build the following simple IRT Bayes net. Assume a single proficiency variable, θ with which takes on the values $\{-1.5, -0.5, 0.5, 1.5\}$ with prior probabilities $\{.125, .375, .375, .125\}$. Let the test have three items with the conditional probabilities given in Table 6.10.

Table 6.10 Potentials for Exercise 6.17

	Item 1		Item 2		Item 3	
θ	Right	Wrong	Right	Wrong	Right	Wrong
-1.5	0.378	0.622	0.182	0.818	0.076	0.924
-0.5	0.622	0.378	0.378	0.622	0.182	0.818
0.5	0.818	0.182	0.622	0.378	0.378	0.622
1.5	0.924	0.076	0.818	0.182	0.622	0.378

Use that network to answer the following questions:

1. What is the probabilities of getting a correct outcome for each item for a student for whom $\theta = -1.5$?
2. What is the most likely level of θ for a student whose answer to *Item 3* is `Right`?
3. Is it possible for the student who got *Item 3* `Right` to answer *Item 1* `Wrong`?

6.18. Appendix A.2 shows where to obtain a complete description of the language assessment. Build this network using your favorite Bayes net package and use it to score some possible response patterns.

7

Explanation and Test Construction

For Bayesian network models to be useful in educational applications, they must not only provide belief estimates for important proficiencies and claims about the learner, but they must also explain the basis of those estimates. Explanation transforms the model from a black box that pontificates an answer to a question into a glass box, whose reasoning methods and assumptions can be evaluated. Contrast this to a neural network model that classifies a learner without being able to explain the rationale behind its conclusion. Usually, a preliminary model makes several unrealistic assumptions, which result in unrealistic inferences. Models must be "debugged" like computer programs, to correct errors in assumption or specification (Almond et al. 2013). The mechanisms used for explanation aid in the process of model validation, criticism, and debugging.

For assessments constructed using evidence-centered design (ECD; Chap. 2), it is only natural that the explanation would be in terms of evidence. Each observed outcome from an assigned task provides "evidence" for or against a claim represented by one or more proficiency variables. But how much? The *weight of evidence* quantifies the evidence provided by a single observation, a complete task, or a complete test. There is a close connection between the weight of evidence and the *reliability* of an assessment.

If we have not yet seen the results from a task, we can calculate the *expected weight of evidence* (EWOE) for that task. This gives a guide to test construction for both adaptive and fixed form tests. EWOE is always calculated with respect to a hypothesis, so we can use it as a spot meter to determine where an assessment has the most power. We can use expected weight of information to make cost/benefit trade-offs and focus on the assessment for particular purposes, even on the fly in adaptive tests.

Section 7.1 reviews some of the literature on explanation in graphical models, describing some simple textual and coloring techniques. Section 7.2 formally defines weight of evidence and provides some of its key properties. Section 7.3 describes EWOE as a metric for Activity Selection—selecting the next task in an adaptive test. Section 7.4 expands on this idea to explore

R. G. Almond et al., *Bayesian Networks in Educational Assessment*,
Statistics for Social and Behavioral Sciences, DOI 10.1007/978-1-4939-2125-6_7

issues of test design and construction. Finally, Sect. 7.5 explores the connection between EWOE and the reliability of the test.

7.1 Simple Explanation Techniques

An expert system is a computer program that takes in information and produces predictions, solutions, or diagnoses. A key feature of an expert system is its ability to explain its findings to a human decision maker. For example, a rule-based system could “explain” itself by running through the chain of rules it had followed to reach a conclusion. As Bayes nets are often thought of as “statistical expert systems,” the explanation problem has been explored in this literature as well. Suermondt (1992) offers a relatively comprehensive discussion of metrics for influential findings and conflicts of evidence, arriving at Kullback–Leibler as his metric for explanation. Henrion and Druzdzel (1990) also looked at qualitative propagation through a graph. Both of these authors looked at natural language as a tool for communicating their findings to users.

This section briefly looks at two proposed techniques. First, Sect. 7.1.1 looks at the technique of Madigan et al. (1997) for coloring the nodes in the graph to provide an explanation. Second, Sect. 7.1.2 looks at an algorithm for finding the most likely explanation for a given pattern of outcomes (Pearl 1988).

7.1.1 Node Coloring

One of the simplest explanation techniques is to simply color the nodes according to the probability of occurrence (Madigan et al. 1997). Thus nodes with a high probability of a noteworthy event would have a different appearance. For proficiency variables, one would color them according to the probability of mastery. For dichotomous observable outcome variables, one would color them according to the probability of getting a correct outcome.

Example 7.1 (Simplified Language Test). *Mislevy (1995c) creates a simple language test to illustrate sorting out evidence from integrated tasks in language testing. In this test, reporting is on the basis of the four modalities of language:* Reading, Listening, Writing, *and* Speaking. *There are four kinds of tasks: a pure reading task; a pure listening task; two kinds of integrated tasks, one involving reading and writing and another involving reading, speaking, and listening. We increase the test length of this illustrative example by making replications of the tasks. In this case, we have five replicates each of the reading and listening tasks and three replicates each of the speaking and writing tasks. Figure 7.1 shows the model graph. Appendix A.2 describes where complete networks for this example can be found online.*

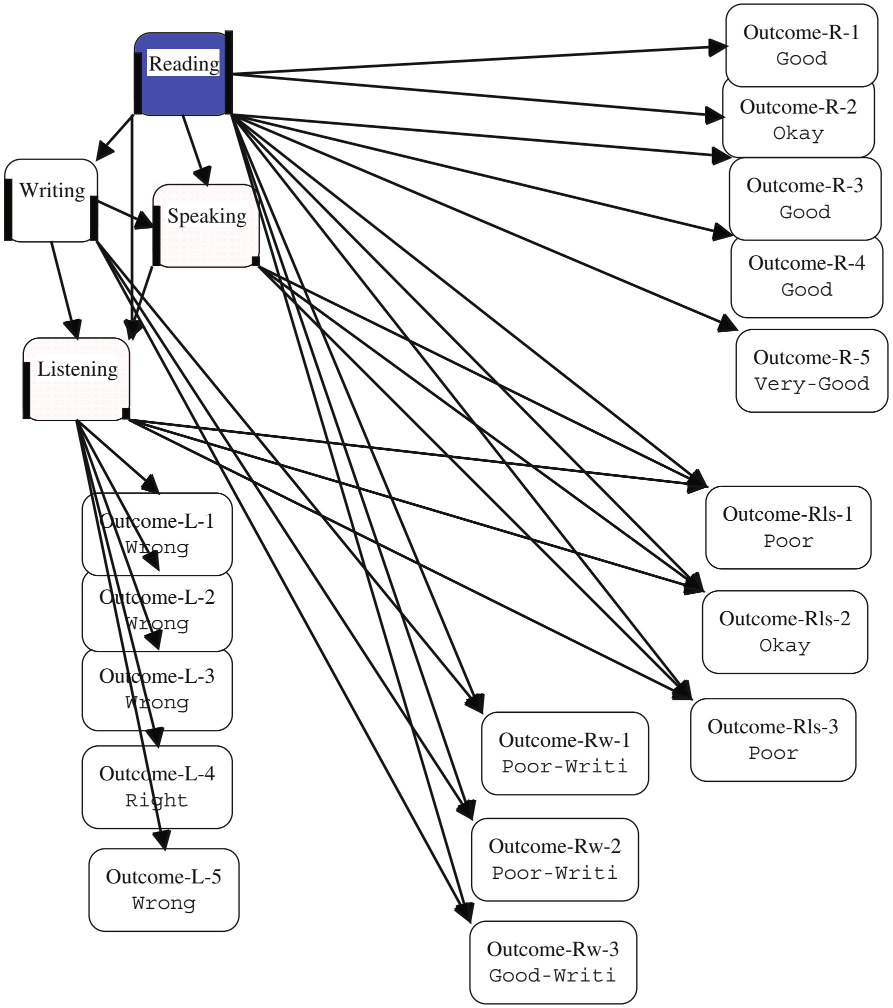


Fig. 7.1 Colored graph for an examinee who reads well but has trouble speaking.

This is a set of "typical" numbers taken from Example 7.1. The color of the node depends on the probability that the student is in a high (*blue*) or low (*red*) state of master, with the *darkness* indicating the strength of the belief. The *black bars* on the sides of the nodes indicate belief of mastery before (*left side*) and after (*right side*) observing the evidence from this test. This figure was generated by GRAPHICAL-BELIEF.

Figure 7.1 shows an example using the model of Example 7.1. The graph is colored according to the trouble spots. The figure shows that this student is doing fairly well in reading, but is having trouble with speaking and listening.

The program Graphical-Belief (Almond 1995) produced this graph using the idea behind so-called heat maps. On a color screen, it uses a "temperature" going from bright red (high probability of negative state) to bright blue (high probability of positive state). It uses nine color levels rather than trying to make fine distinctions with colors. The bar on the right side of the nodes provides a more detailed estimate of the probability.

Graphical-Belief will display the colors for any binary hypothesis. In this example, as in many others, the nodes can take on more than two states. For each variable, we must pick one state (or set of states) as the positive state (blue color or light gray); the rest of the states become the negative states (red color or dark gray). Interactively changing which states are defined to be positive provides a more complete picture of the model.

Daniel et al. (2003) uses this kind of graphical representation to facilitate interaction between the teacher and the student. Both the teacher and the student are given a view of the graph. The teacher can enter data through assessment nodes and the student through self-assessment nodes (the data entry form contains additional fields for justifications). The display shows the joint information. Initial field trials in classrooms have yielded positive responses.

7.1.2 Most Likely Scenario

Pearl (1988) suggests dividing the variables of the model into three categories: *observable variables* whose values may or may not be observed; unobservable *hypothesis variables* which cause the particular configuration of the observation variables; *intermediate variables* whose results are important only in calculating the beliefs of the other variables. He considers the problem of finding a pattern of the hypothesis variables that best (highest probability) explains the configuration of the observable variables. Pearl (1988) calls this task *belief revision*, as the idea is to possibly revise a current "best explanation" when new evidence arrives. This approach contrasts with *belief updating* using the fusion and propagation algorithm in Chap. 5 that produces marginal probability distributions for the hypothesis variables.

Applying this approach to ECD terminology, the hypothesis variables correspond to proficiency variables. Let $\mathbf{S} = \{S_1, \ldots, S_K\}$ be the set of proficiency variables. A given assignment of values to all of the proficiency variables $\mathbf{s} = \{s_1, \ldots, s_K\}$ is a *proficiency profile*. The goal of belief revision is then to find the proficiency profile that is most likely to produce the given pattern of observed outcomes. This is the most likely "explanation" for the observed outcomes.

Belief revision is simple to carry out computationally using the same Bayes net structures. One simply replaces a summation in the algorithm of Chap. 5

with a maximization. Pearl (1988) gives the details (also Almond 1995; Shenoy 1991). Belief revision is an option on almost all of the free and commercial software for manipulating Bayesian networks (Appendix A.1.1).

The most likely scenario or proficiency profile consistent with a particular set of observations and hypotheses can provide insight into the behavior of the model. At any point, the best explanation is just the pattern of proficiencies that best fits the pattern of observations. Henrion and Druzdzel (1990) advocate this approach and suggest that scenario-based explanations mimic the way one person would explain a model to another.

7.2 Weight of Evidence

Each outcome that we observe from each task provides evidence for whether the learner has the proficiencies we are trying to assess. An important part of explanation is understanding which observations were most influential in estimating those proficiencies. The *weight of evidence* provides a metric for influential findings.

When an observation is used to update beliefs in Bayes net built for a profile score assessment, it usually changes the belief about all of the proficiency variables. However, the same observation will cause a different strength and even direction of change for each proficiency variable. Any collection of proficiency variables defines a universe of possible proficiency profiles. We will call any split of the set of all possible profiles into two groups a *hypothesis*, H, and its negation, $\overline{H}$. Typical hypotheses have to do with the mastery of one of the skills, for example, $S_k \geq$ `proficient`. More complex hypotheses may also be of interest, for example, whether a given instructional program is appropriate for a given learner might depend on several proficiency variables.

Good (1985) derives the weight of evidence as a measure of the amount of information a piece of evidence E provides for a hypothesis H. The *weight of evidence for H vs $\overline{H}$* is:

$$W(H{:}E) = \log \frac{\mathrm{P}(E|H)}{\mathrm{P}(E|\overline{H})} = \log \frac{\mathrm{P}(H|E)}{\mathrm{P}(\overline{H}|E)} - \log \frac{\mathrm{P}(H)}{\mathrm{P}(\overline{H})} \ . \tag{7.1}$$

Thus, the weight of evidence is the difference between the prior and posterior log odds for the hypothesis. (Good recommended taking the logarithms to base 10, and multiplying the result by 100. He calls the resulting units "centibans." All the comparisons are the same; he just found the units easier to work with).

Just as our hypothesis may be compound, our evidence may be compound too. In a typical testing situation, the evidence is in fact made up of the observed outcomes from many tasks. In this case, Eq. 7.1 refers to the joint evidence from all tasks. If we can partition the evidence into two sets of observations, E_1 and E_2, we can define the *conditional weight of evidence*:

$$W(H{:}E_2|E_1) = \log \frac{\mathrm{P}(E_2|H, E_1)}{\mathrm{P}(E_2|\overline{H}, E_1)} \ . \tag{7.2}$$

These sum in much the way that one would expect:

$$W(H{:}E_1, E_2) = W(H{:}E_1) + W(H{:}E_2|E_1) \ . \tag{7.3}$$

In general, $W(H{:}E_2|E_1) = W(H{:}E_2)$ only when E_1 and E_2 are independent given both H and $\overline{H}$. As typically either H or $\overline{H}$ is usually a compound (consists of several different skill profiles), this independence usually does not hold. Instead, if E_1 and E_2 both favor H, then typically $W(H{:}E_2|E_1) < W(H{:}E_2)$. This makes intuitive sense. Suppose that we do not know whether a learner has a particular skill. The first time we see the learner solve a problem that requires that skill, we will get a great deal of evidence that the learner has the skill. The second and third time we make that kind of observation, we are confirming what we observed from the first observation, so, we expect the evidentiary value of the replications to be lower.

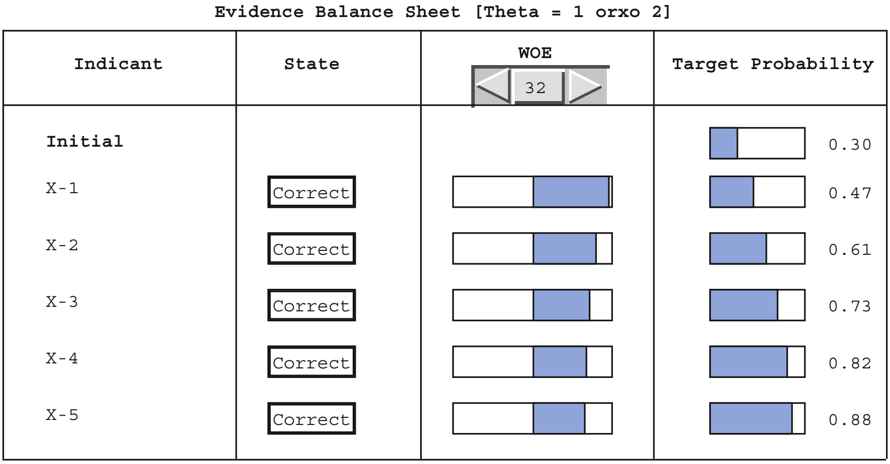
7.2.1 Evidence Balance Sheet

Spiegelhalter and Knill-Jones (1984) present the weights of evidence in an *evidence balance sheet* in simple logistic regression models. Madigan et al. (1997) adapt the evidence balance sheet for graphical models. Figures 7.2, 7.3, 7.4, and 7.5 show a possible graphical interpretation from GRAPHICAL-BELIEF.[1] The evidence is ordered according to when it arrives (for example, the order of test items in a booklet, or presentation of tasks in a CAT). At each point of time, the weight of evidence conditioned on all previous evidence is displayed along with the current estimate of probability.

Example 7.2 (IRT with identical items). *This example looks at a five item test, where all of the items have identical item parameters. The model used is essentially a discretized IRT model. The proficiency model has a single variable θ with five levels: $\{-2, -1, 0, 1, 2\}$. The prior distribution is a triangular distribution (similar to the normal, but with fatter tails). The evidence models are made by calculating the 2PL likelihood with discrimination 1.0 and difficulty 0.0, and then filling the entries in the table. Figure 7.2 shows the evidence balance sheet for the hypothesis $\theta \geq 1$ for a person who got all five items right.*

Note that in the example of Fig. 7.2, the only difference between observations X_1 and X_5 is the order in which they arrive. When X_1 arrives, we know little about the student, and the observation has a relatively large evidentiary value. However, when later observations arrive, they are confirming what we learned from the first observation. Their evidentiary value is smaller, and it decreases as the number of replications go up. It is important to remember this order effect as we look at the evidence balance sheet. If we had asked for

[1] On a color screen, this rendering uses pale blue for positive evidence, and red for negative evidence.



Evidence Balance Sheet [Theta = 1 orxo 2]

Indicant	State	WOE 32	Target Probability
Initial			0.30
X-1	Correct		0.47
X-2	Correct		0.61
X-3	Correct		0.73
X-4	Correct		0.82
X-5	Correct		0.88

Fig. 7.2 Evidence balance sheet for $\theta \geq 1$ (Example 7.2)

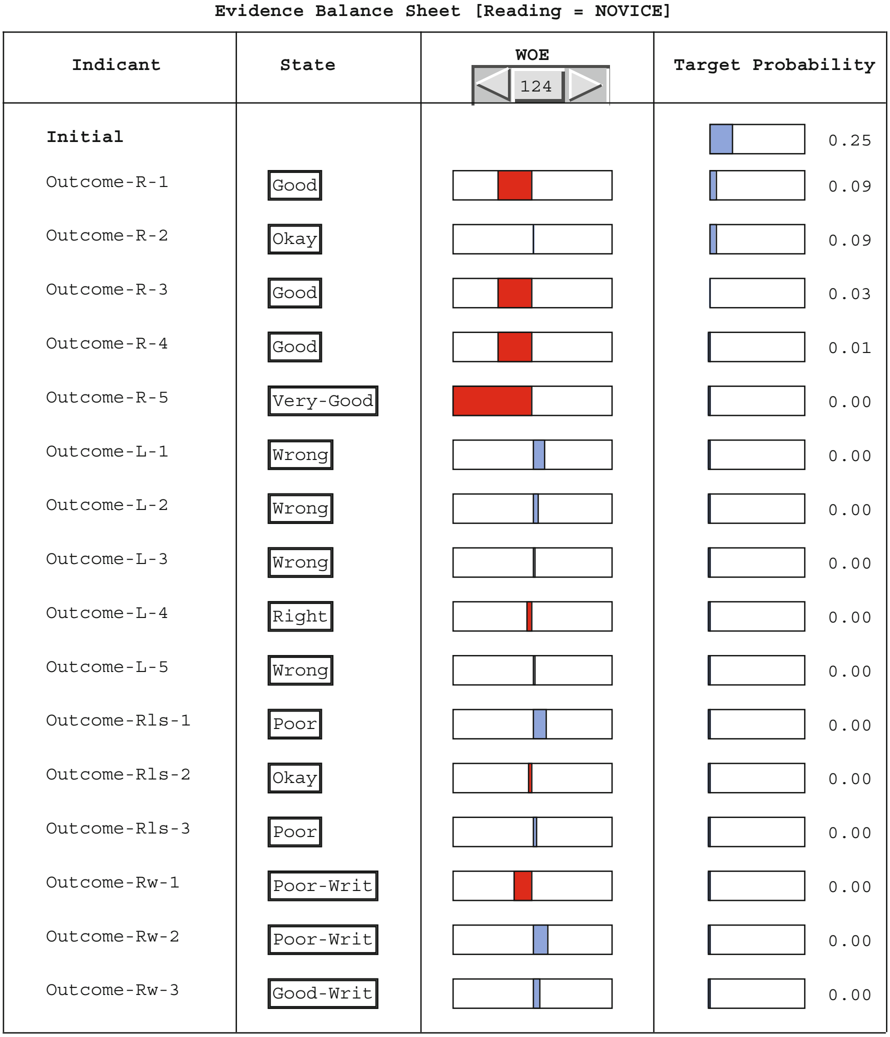
This shows the progressive influence of the five identical (in parameters) items on the running probability that θ is at the highest level. The column marked "WOE" displays the conditional weight of evidence (given the previous evidence). (The number "32" at the top of the column indicates that the weight of evidence bar runs from -32 to $+32$ centibans). The column marked target probability shows the cumulative probability that $\theta \geq 1$ after each finding. This figure was generated by GRAPHICAL-BELIEF.

Task 5 first instead of last, it would have had the biggest evidentiary value, not Task 1. As the conditional weights of evidence are order sensitive, Madigan et al. (1997) suggest interactively ordering the observations to promote better understanding of sensitivity to the findings.

This order effect does not change the total evidence across all tasks; this remains constant. What is changing is which variables are conditioned on when we calculate the conditional weight of evidence observation by observation. It is always the case that $W(H{:}E_1)+W(H{:}E_2|E_1) = W(H{:}E_2)+W(H{:}E_1|E_2)$. If E_1 and E_2 point in the same direction, it is usually, but not necessarily, the case that $W(H{:}E_1) \geq W(H{:}E_1|E_2)$.

For a richer model, we return to Example 7.1. The proficiency model has four reporting variables. This give us the chance to observe the effects of both *direct evidence*—evidence about proficiency variables that are parents of the task—and *indirect evidence*—evidence about tasks which are children of other correlated proficiency variables.

Figure 7.3 shows that the evidence against low reading ability is mostly direct. The reading tasks are first, and relatively good performance on those tasks (averaging at about the second highest state) quickly establishes a low probability of Reading being at the `novice` state. Even though poor perfor-



Evidence Balance Sheet [Reading = NOVICE]

Indicant	State	WOE 124	Target Probability
Initial			0.25
Outcome-R-1	Good		0.09
Outcome-R-2	Okay		0.09
Outcome-R-3	Good		0.03
Outcome-R-4	Good		0.01
Outcome-R-5	Very-Good		0.00
Outcome-L-1	Wrong		0.00
Outcome-L-2	Wrong		0.00
Outcome-L-3	Wrong		0.00
Outcome-L-4	Right		0.00
Outcome-L-5	Wrong		0.00
Outcome-Rls-1	Poor		0.00
Outcome-Rls-2	Okay		0.00
Outcome-Rls-3	Poor		0.00
Outcome-Rw-1	Poor-Writ		0.00
Outcome-Rw-2	Poor-Writ		0.00
Outcome-Rw-3	Good-Writ		0.00

Fig. 7.3 Evidence balance sheet for *Reading* = `Novice` (Example 7.1)

As the five reading tasks are first in this assessment, the assessment quickly establishes that this person reads fairly well (above the novice level). The other items have relatively little influence. Note that hypothesis here is negative; it is a hypothesis that reading is in its low state, so that evidence against this hypothesis is strong.

mance on the later tasks provides some direct and indirect evidence in favor of Reading being low, it is not enough to overwhelm the initial direct evidence. In particular, there is an alternative explanation (poor Writing and Speaking skills) that explains away the direct evidence from the latter tasks.

The story with Listening, shown in Fig. 7.4, is quite different. First, the good performance on the five reading tasks provides a small bit of indirect evidence against Listening being at the lowest level. Second, the poor performance on the Listening tasks (4 out of 5 wrong) provides strong direct evidence that the Listening proficiency is in the lowest state. Note that one listening task that has a correct outcome has an effect in the opposite direction, but it is quickly countered by the next task whose outcome is once again wrong. Also, the first integrated Reading–Listening–Speaking task provides evidence that Listening is at the `Novice` level. This is a combination of direct and indirect evidence. Relatively strong evidence that Reading skills are good makes poor listening skills a much more likely explanation for poor performance on this task.

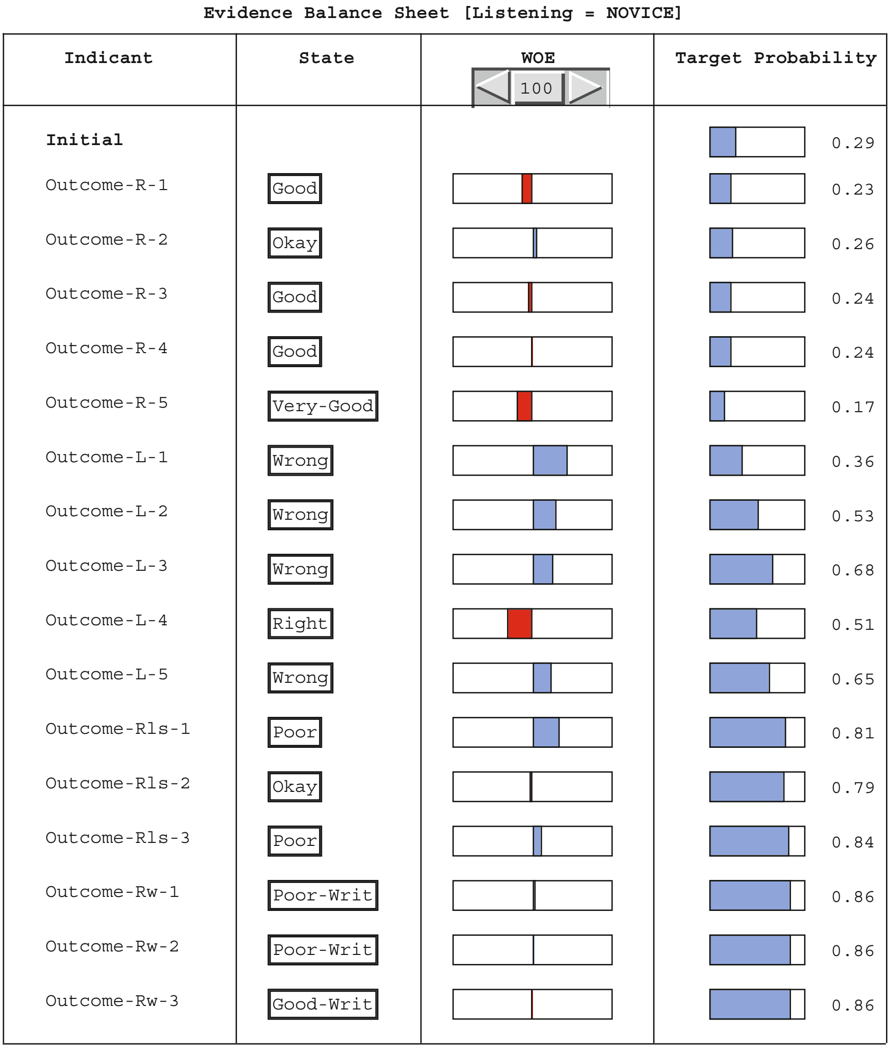
Figure 7.5 shows how the inferences about the Speaking variable progresses. First, the good performance on the Reading items provides indirect evidence against Speaking being low (through its correlation with Reading). Next, in contrast, the weak performance on the Listening items provides indirect evidence that Speaking may in fact be low. However, the poor performance on the three integrated tasks involving speaking provides much stronger direct evidence than the preceding indirect evidence. Finally, the indirect evidence provided by the Reading–Writing tasks is quite small.

A related way that the weight of evidence could be used is to select a few tasks for more detailed feedback to the test taker. By giving feedback on tasks that have the biggest negative weight of evidence for a skill mastery hypothesis, one can focus the learner's attention on problem spots. By giving feedback on tasks that have the biggest positive weight of evidence, one can reinforce students' appropriate use of skills.

7.2.2 Evidence Flow Through the Graph

Madigan et al. (1997) suggest using the model graph to provide a picture of the flow of information through the model. In particular, they suggest coloring the edges of the graphical model to encode the strength of information flow through the model. Working with probabilistic models, they create a hollow edge whose width displays the strength of influence from a node to its neighbor. Several metrics can be used to measure this strength; Madigan et al. (1997) recommend the weight of evidence.

Madigan et al. (1997) demonstrate weight of evidence-based edge coloring in simple chain graphs. For example, consider a model with three binary variables: A, B, and C. Suppose that we know that A is true and want to know what impact this information has on our belief that C is true. For each

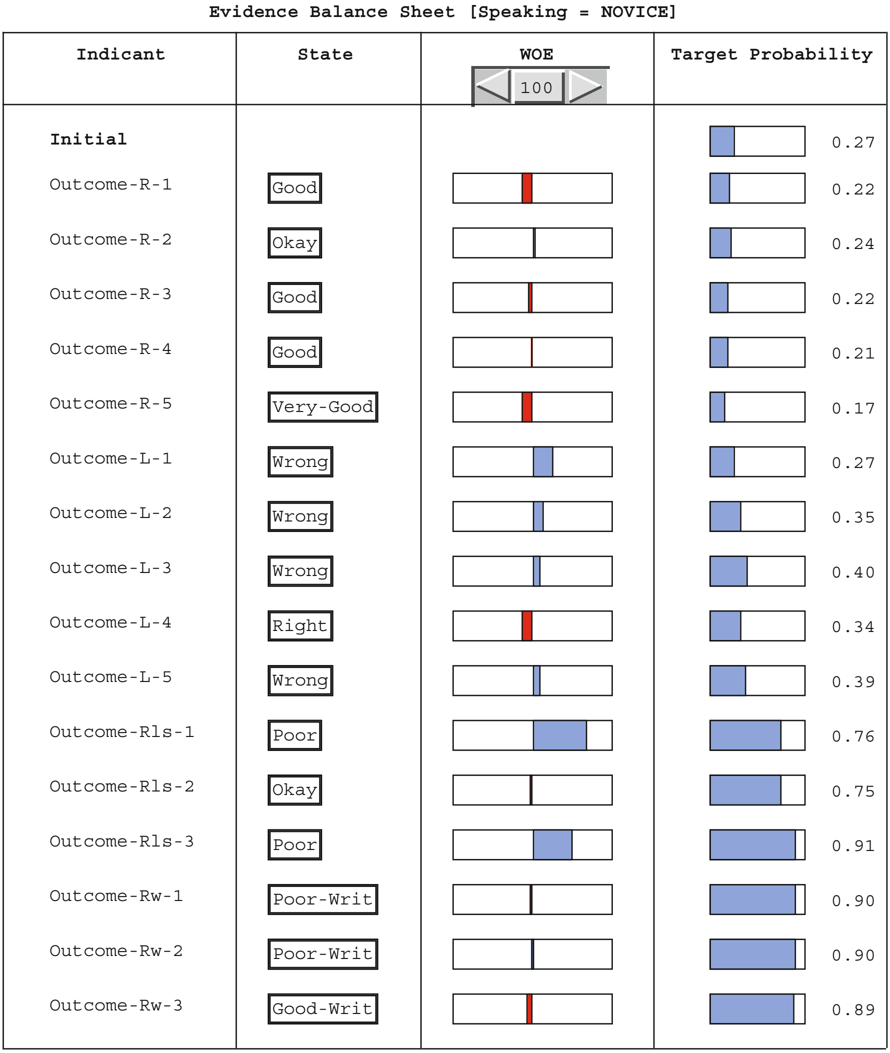


Evidence Balance Sheet [Listening = NOVICE]

Indicant	State	WOE 100	Target Probability
Initial			0.29
Outcome-R-1	Good		0.23
Outcome-R-2	Okay		0.26
Outcome-R-3	Good		0.24
Outcome-R-4	Good		0.24
Outcome-R-5	Very-Good		0.17
Outcome-L-1	Wrong		0.36
Outcome-L-2	Wrong		0.53
Outcome-L-3	Wrong		0.68
Outcome-L-4	Right		0.51
Outcome-L-5	Wrong		0.65
Outcome-Rls-1	Poor		0.81
Outcome-Rls-2	Okay		0.79
Outcome-Rls-3	Poor		0.84
Outcome-Rw-1	Poor-Writ		0.86
Outcome-Rw-2	Poor-Writ		0.86
Outcome-Rw-3	Good-Writ		0.86

Fig. 7.4 Evidence balance sheet for *Listening* = `Novice` (Example 7.1)

The first five tasks are reading tasks which have good outcomes. They provide a little bit of evidence against low listening ability. The next five tasks are listening tasks with poor outcomes. They provide evidence for low listening ability, except for L-4 which has a correct outcome. The first integrated Reading–Listening–Speaking task (which has a poor outcome) also provides evidence for low Listening ability.



Evidence Balance Sheet [Speaking = NOVICE]

Indicant	State	WOE 100	Target Probability
Initial			0.27
Outcome-R-1	Good		0.22
Outcome-R-2	Okay		0.24
Outcome-R-3	Good		0.22
Outcome-R-4	Good		0.21
Outcome-R-5	Very-Good		0.17
Outcome-L-1	Wrong		0.27
Outcome-L-2	Wrong		0.35
Outcome-L-3	Wrong		0.40
Outcome-L-4	Right		0.34
Outcome-L-5	Wrong		0.39
Outcome-Rls-1	Poor		0.76
Outcome-Rls-2	Okay		0.75
Outcome-Rls-3	Poor		0.91
Outcome-Rw-1	Poor-Writ		0.90
Outcome-Rw-2	Poor-Writ		0.90
Outcome-Rw-3	Good-Writ		0.89

Fig. 7.5 Evidence balance sheet for *Speaking* = `Novice` (Example 7.1)

The first ten tasks are the reading and listening tasks. They provide a small amount of indirect evidence for the Speaking skill being at the novice level. However, the poor performance on two of the three integrated Reading–Listening–Speaking tasks provides rather stronger direct evidence for low Speaking ability.

possible value of B, say b_i, the quantity $W(C/\overline{C}{:}B = b_i)$ is the *potential weight of evidence* for b_i. The largest potential weight of evidence is the *relevant potential weight of evidence*. As B is a binary variable, only one of $W(C/\overline{C}{:}B)$ and $W(C/\overline{C}{:}\overline{B})$ is positive; that one will be the relevant potential weight of evidence.

The following scheme encodes the weight of evidence (for the evidence chain) via the width of the edge of a graphical model. Figure 7.6 displays this idea for a simple graphical model. The arrow between nodes A and B shows the weight of evidence A provides for B. As A is known, the actual weight of evidence equals the potential and the edge is shown as a filled arrow. The outer arrow between nodes B and C shows the relevant potential weight of evidence, that is the maximum evidence B could provide for C if it were known. The inner arrow shows the actual weight of evidence all findings upstream of B (i.e., A) provides for C.

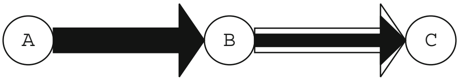


Fig. 7.6 Evidence flows using weight of evidence

Although edge coloring is an effective technique for tree-shaped graphical models with binary variables, extending it beyond those special cases presents some difficulties. In particular, if the intermediate variable B has many possible outcomes it may be difficult to show how each outcome contributes to our overall beliefs about C. Clustering variables to form a Markov tree presents the same difficulty: the clustered nodes are effectively nonbinary variables. Madigan et al. (1997) suggest selecting a set of *positive states* for each node in the graph and using the positive states for determining color in the displays. All weight of evidence calculations are made with respect to the binary proposition “The variables in the node take on one of the positive states.” Interactively selecting the marked state (or set of states) for each node should allow the modeler to build up a good picture of evidence flow. Nicholson and Jitnah (1998) use mutual information instead of weight of evidence to similar ends. This has the advantage of not requiring that the nodes be reduced to binary ones.

The evidence-flow graphs are primarily useful in understanding how indirect evidence flows through a system. In particular, they can help identify evidence bottlenecks, places where no matter how much evidence we get downstream, the evidence has little effect on our inferences upstream. This could mean that we need to redesign our test, to provide more direct evidence about the skills of interest. It also could mean that there is an inhibitor effect, some alternative explanation of the observed evidence that does not allow us to make inferences about the quantity of interest.

7.3 Activity Selection

The ECD framework is intended to describe both pure assessments (activities whose goal is to assess the current proficiencies of the learner) and assessments embedded in the more complex setting of a tutoring system, where students can also be administered activities meant to increase their proficiencies. In the latter case, we would have a large number of activities to choose from, including activities whose purpose is primarily instruction, or *instructional tasks*, and activities whose primary purpose is assessing the learner's current state, or *assessment tasks*. The process responsible for selecting the next activity would also have the responsibility for deciding when to switch between "instruction" and "assessment" modes, as well as criteria for terminating the activities.

A teacher making an instructional plan for a student is very much like a physician diagnosing a disease and prescribing treatment for a patient. Success in both cases typically depends on the degree to which the teacher/doctor has valid knowledge of the student/patient. As such, instructional planning goes hand-in-hand with student modeling. In general, instructional planning refers to the process of specifying instructional goals and actions that provide for a learning experience that is coherent, complete, and valid.

Blending assessment and instruction is a big topic and this book cannot do it justice. Instead, this section looks at the smaller problem of just selecting assessment tasks. In particular, it develops an approach based on the weight of evidence. Section 7.3.1 defines the related concept of value of information, and Sect. 7.3.2 defines the expected weight of evidence. Section 7.3.3 provides an analogous measure called mutual information that can be used in more complicated situations. Section 7.4 goes on to show how these metrics can be used in both fixed-form and adaptive test construction.

7.3.1 Value of Information

For a licensing agency, it is presumably more regrettable to award somebody a license who is not fit to practice, than it is to fail to award the license to somebody who is fit. The latter person only needs to take the test again, while the former could do harm to the public. On the other hand, for a classroom teacher, it is more regrettable to fail to identify a student who is having difficulty than it is to mistakenly identify a student as needing extra help who does not. Presumably this mistake will be uncovered and corrected during the remediation.

This discussion gets at the idea of *utility*. In these examples, we have a decision we need to make: for instance, whether or not to license the candidate, or whether or not to assign the student to remediation. The utility of a decision depends on both the decision, d, and the true values of the proficiency variables $\mathbf{S}$, as well as the relative costs of decision alternatives and the expected outcomes. We can write the utility as $u(d, \mathbf{S})$. This utility now can be traded-off against the cost of the test, as in Example 4.1.

The decision theoretic approach to this problem would be to calculate the value of information (Matheson 1990; Heckerman et al. 1993) with respect to the instructional choices that are available. As the true proficiency state $\mathbf{S}$ is unknown, it is considered a random variable under the Bayesian framework. Thus, our expected utility is $\mathrm{E}_{\mathbf{S}}u(d, \mathbf{S})$, where $\mathrm{E}_{\mathbf{S}}$ is the expectation taken with respect to the marginal distribution $\mathrm{P}(\mathbf{S})$ before any evidence is observed. The *Bayes decision* is to take the value of d that maximizes the expected utility.

Suppose, we had the result of some test T that is related to $\mathbf{S}$. Then the decision we would take is the one that maximizes expected utility with respect to the posterior distribution $\mathrm{P}(\mathbf{S}|T = t_k)$. The *expected value of information* is the expected difference between the best decision we could make with the test result and the best decision we could make without the test result. Thus,

$$\mathrm{VoI}(T) = \mathrm{E}_T \left[\max_d \mathrm{E}_{\mathbf{S}|T} u(d, \mathbf{S}) - \max_d \mathrm{E}_{\mathbf{S}} u(d, \mathbf{S}) \right] , \qquad (7.4)$$

where $E_{\mathbf{S}|T}$ is the expectation taken with respect to the conditional distribution $\mathrm{P}(\mathbf{S}|T)$ and E_T is the expectation with respect to the marginal distribution of the evidence, $\mathrm{P}(T)$—that is, before T is observed.

Now, we have a rule for deciding whether or not to test: If the expected value of information exceeds the cost of the test, testing is worthwhile. Consider Example 4.1. There the goal is to maximize the student's ability at the end of whatever instructional strategy is chosen. If almost all students need the same instruction, it may not make sense to test because the test results will not affect the optimal decision. Similarly, it does not make sense to test if test is very expensive compared with the cost of the instruction. Only if the test helps us make better instructional decisions is testing worthwhile.

If we have multiple tests, we would pick the one that maximizes the expected value of information. Using the ECD framework, we can regard each task as a "test" (in the sense we have been using the term in this section), so the expected value of information gives us a principle for task selection in adaptive testing. We can also consider the collection of tasks $\{T_1, \ldots, T_n\}$ that maximizes the expected value of information.

Suppose, during an adaptive test, we pick the task T_* which maximizes the expected value of information at every step, stopping when there is no test whose expected value of information (given the results seen so far) exceeds the cost of the test. This is called myopic search. It is sometimes possible to find cases where there is a pair of tests T_1 and T_2 that together give more information than T_*. Thus, myopic search is not guaranteed to find the best sequence of tests. Heckerman et al. (1993) explore this issue in more detail.

Example 7.3 (Value of Information for a Dental Hygienist Assessment). *This illustration extends Example 5.1. To be consistent with the notation in this chapter, S represents a dental hygiene student's proficiency in* *examining patients, which can take two values, $s_1 = \texttt{expert}$ and $s_2 = \texttt{novice}$. The proficiency model* **S** *consists simply of the single variable S. Initially $p(\texttt{expert}) = p(\texttt{novice}) = .5$. Now there are three tasks available to administer:*

- *Task 1 is a hard task. It yields an observable variable X_1, which can take two values, $x_{11} = \texttt{yes}$ and $x_{12} = \texttt{no}$ corresponding to whether an examinee takes an adequate patient history. An expert's probability of taking an adequate history in such tasks is .6 and a novice's is .2, so $p\left(x_{11} \left| s_1 \right.\right) = .6$ and $p\left(x_{12} \left| s_1 \right.\right) = .4$, and $p\left(x_{11} \left| s_2 \right.\right) = .2$ and $p\left(x_{12} \left| s_2 \right.\right) = .8$.*
- *Task 2 is a medium task. It yields the observable variable X_2, with an expert's and novice's probabilities of taking an adequate history .8 and .4; that is, $p\left(x_{21} \left| s_1 \right.\right) = .8$ and $p\left(x_{21} \left| s_2 \right.\right) = .4$. This is the task that appears in Example 5.1.*
- *Task 3 is easy. It yields X_3, where $p\left(x_{31} \left| s_1 \right.\right) = .95$ and $p\left(x_{31} \left| s_2 \right.\right) = .65$.*

Denote the possible decisions by $d_1 = \texttt{expert}$ and $d_2 = \texttt{novice}$, and let the utilities $u(d, s)$ for deciding an examinee is an expert or novice given true proficiency be as shown below. There are high utilities for deciding an expert is an expert and a novice is a novice; there is a lower utility for deciding an expert is a novice and the lowest utility for deciding a novice is an expert.

Decision	*Proficiency*	*Utility*
`expert`	`expert`	100
`novice`	`expert`	40
`expert`	`novice`	0
`novice`	`novice`	100

We can now use Eq. 7.4 to calculate the value of information of any of the tasks. We will focus on Task 2.

Starting from the initial .5 probabilities for novice and expert, we first determine the term $\max_d \mathrm{E}_{\mathbf{S}} u(d, \mathbf{S})$. This is the expected utility of the decision that gives the greatest expected utility before administering any tasks. This expression is needed and is the same for calculating VoI for all tasks in the pool at this point. If the decision is `expert`, the expected utility is the average of the utility for `expert` if the true proficiency is `expert`, or 100, and the utility for `expert` if the true proficiency is `novice`, or 0, weighted by their respective current probabilities of .5. That is,

$$\mathrm{E}_S u(\texttt{expert}, S) = u(d_1, s_1)p(s_1) + u(d_1, s_2)p(s_2) = 100 \times .5 + 0 \times .5 = 50.$$

By similar calculations,

$$\mathrm{E}_S u(\texttt{novice}, S) = u(d_2, s_1)p(s_1) + u(d_2, s_2)p(s_2) = 70.$$

The decision maximizing expected information is therefore d_2, `novice`, and $\max_d \mathrm{E}_S u(d, S) = 70$.

We next consider the expected value of the maximum-decision utility if Task 2 were administered. Calculations similar to those above are first determined to evaluate $\max_d \mathrm{E}_{\mathbf{S}|T} u(d, \mathbf{S})$, *where* X_2 *plays the role of* T. *We need to consider the cases of when* $X_2 = x_{21}$ *and* $X_2 = x_{22}$, *or when an adequate or inadequate performance is observed, and determine the maximum-decision utility in each case. For* $X_2 = x_{21}$, *recall from Example 5.1 that the posterior probabilities for* `expert` *and* `novice` *given an adequate performance are .67 and .33. For the decision of* `expert`, $\mathrm{E}_{\mathbf{S}|T} u(d, \mathbf{S})$ *becomes in this case*

$$\begin{aligned} \mathrm{E}_{S|x_{21}} u(\texttt{expert}, S) &= u(d_1, s_1)p(s_1|x_{21}) + u(d_1, s_2)p(s_2|x_{21}) \\ &= 100 \times .67 + 0 \times .33 = 67. \end{aligned}$$

For the decision of `novice`,

$$\begin{aligned} \mathrm{E}_{S|x_{21}} u(\texttt{novice}, S) &= u(d_2, s_1)p(s_1|x_{21}) + u(d_2, s_2)p(s_2|x_{21}) \\ &= 40 \times .67 + 100 \times .33 = 60. \end{aligned}$$

Thus, when $X_2 = x_{21}$ *deciding* `expert` *produces the maximal utility, 67.*

If the performance to Task 2 is inadequate, or $X_2 = x_{22}$, *the posterior for* S *is .25 for* `expert` *and .75 for* `novice`. *By calculations similar to those above,* $\mathrm{E}_{S|x_{22}} u(\texttt{expert}, S) = 25$ *and* $\mathrm{E}_{S|x_{22}} u(\texttt{novice}, S) = 85$. *Deciding* `novice` *produces the maximal utility of 85.*

The final step is the outer expectation. This is the weighted average of the maximizing utilities for the decisions when X_2 *takes each possible value, weighted by the current marginal probabilities for those outcomes before the observation – from Table 5.2, .6 and .4 respectively. Thus*

$$\begin{aligned} \mathrm{VoI}(X_2) &= \mathrm{E}_{X_2}\left[\max_d \mathrm{E}_{S|X_2} u(d, S) - \max_d \mathrm{E}_S u(d, S)\right] \\ &= \left[\max_d \mathrm{E}_{S|X_2=x_{21}} u(d, S) - \max_d \mathrm{E}_S u(d, S)\right] \times .6 \\ &\quad + \left[\max_d \mathrm{E}_{S|X_2=x_{22}} u(d, S) - \max_d \mathrm{E}_S u(d, S)\right] \times .4 \\ &= [67 - 70] \times .6 + [85 - 70] \times .4 = 4.2. \end{aligned}$$

Applying the same steps to Tasks 1 and 3, the hard and easy tasks, we find their Value of Information to be 8 and 0. Task 1, the hard task, provides the greatest expected VoI. If the cost of administering Task 1 were greater than 8, however, it would be best to decide `novice` *without testing.*

Task 3 has zero VoI. The best decision before testing was deciding `novice`. *Test 3 is very easy, so practically all experts and even most novices get it right. With the utilities favoring caution in deciding expert status, the expected utilities for both an adequate and an inadequate performance to this task lead to deciding* `novice`. *However, different utilities or different initial probabilities for proficiency could produce circumstances under which administering Task 3 would increase expected utility.*

If a test is carried out, the same steps can be repeated with not-yet-administered tests to see if a subsequent test provides sufficient VoI to then apply, and if so which had the largest VoI. All of the calculations carried out above would start with beliefs conditional on the observed value of the tests already administered.

7.3.2 Expected Weight of Evidence

The decision-making approach described in the previous section requires explicating utility functions for various states of proficiency on the same scale as the costs of instructional and assessment tasks, in addition to establishing the probabilities. Establishing such a mapping can be difficult, and different stakeholders can disagree. In these cases, the optimizing machinery of decision theory can be pressed into service nevertheless by substituting a *quasi-utility* (Glasziou and Hilden 1989) for the true utility. Quasi-utilities gauge how close our estimated proficiency is to the actual proficiency. For example, Lord's (1980) "expected information" computerized adaptive testing (CAT) algorithm in item response theory uses Fisher information as a quasi-utility. Henson and Douglas (2005) suggest using a weighted sum of Kullback–Leibler distances, and Madigan and Almond (1995) recommend the use of the weight of evidence as a quasi-utility. This section explores the idea.

When discriminating between a single hypothesis H and its negation $\overline{H}$, Good and Card (1971) recommend the *Expected Weight Of Evidence (EWOE)* as a quasi-utility:

$$EW(H{:}E) = \sum_{j=1}^{n} W(H{:}e_j)\mathrm{P}(e_j \mid H) \tag{7.5}$$

where $\{e_j, j = 1, \ldots, n\}$, represent the possible outcomes of the observation E. $W(H{:}e_j)$ is the weight of evidence concerning H provided by the evidence e_j, $\log[\mathrm{P}(e_j \mid H)/\mathrm{P}(e_j \mid \overline{H})]$ (Eq. 7.1). Informally, $EW(H{:}E)$ is the weight of evidence that will be obtained from E "on the average," when the hypothesis H is true.

Example 7.4 (Expected Weight of Evidence for a Dental Hygienist Assessment). *This example is also based on the dental hygienist assessment of Example 5.1, with the probabilities and the additional items introduced in Example 7.3. We consider the hypothesis that an examinee is an expert, so $H : S = s_1$ and $\overline{H} : S = s_2$, and initially focus on Task 2, so X_2 plays the role of E. Equation 7.5 then takes the form*

$$EW(s_1{:}X_2) = \sum_{j=1}^{n} W(s_1{:}x_{2j})\mathrm{P}(x_{2j} \mid s_1).$$

Now

$$W(s_1{:}x_{21}) = \log \frac{\mathrm{P}(x_{21}|s_1)}{\mathrm{P}(x_{21}|s_2)} = \log \frac{.8}{.4} = .69.$$

Similarly, $W(s_1{:}x_{22}) = \log[\mathrm{P}(x_{22}|s_1)/\mathrm{P}(x_{22}|s_2)] = \log[.20/.60] = -1.10.$ *Then since* $p(x_{21}|s_1) = .8$ *and* $p(x_{22}|s_1) = .2$,

$$EW(s_1{:}X_2) = .69 \times .8 + -1.10 \times .2 = .332.$$

Similar calculations for Task 1 and Task 3 give $EW(s_1{:}X_1) = .384$ *and* $EW(s_1{:}X_3) = .259$. *Thus the hard task provides the greatest expected evidence about whether an examinee is an expert, followed by the medium task, then the easy task.*

If applied directly to H and $\overline{H}$, EWOE does not distinguish the values of different outcomes, and thus does not use utility in a formal way. It is however possible to adjust it for misclassification costs. Breiman et al. (1984) have proposed the following approach (see also Glasziou and Hilden 1989). Suppose that to misclassify as candidate for whom H holds as $\overline{H}$ is w times as regrettable as the reverse. An effectively weighted EWOE formulation can be applied using an artificial hypothesis H', where a case of H is considered to be w cases of H'; that is,

$$\mathrm{P}(H') = \frac{w\mathrm{P}(H)}{w\mathrm{P}(H) + \mathrm{P}(\overline{H})}.$$

A quasi-utility based on EWOE and valuing correct classifications of H as w times as much as correct classifications of $\overline{H}$ is then obtained as

$$\sum_{j=1}^{n} W(H'{:}e_j)\mathrm{P}(e_j|H') \tag{7.6}$$

with

$$\mathrm{P}(e_j|H') = \mathrm{P}(e_j|H)\frac{\mathrm{P}(H)(w-1)+1}{\mathrm{P}(H|e_j)(w-1)+1}$$

and

$$W(H'{:}e_j) = W(H{:}e_j).$$

Incorporating the misclassification costs into the EWOE means that it is assessing improvement in *risk* (the glass-half-empty view of expected utility) instead of just probabilities. If the probability of a certain disease is relatively low, but missing it would be serious and a simple test is available, a doctor will usually order the test. A risk-based metric for test selection that includes the costs of misclassification enables the procedure to reflect this reasoning.

Heckerman et al. (1993) note a close connection between the EWOE and value of information. If our hypothesis is binary and we have two possible actions we can take, it is possible to define a value p^* such that if $p(H) > p^*$ then taking one particular action is always the best decision. This is called the *Bayes decision* and is discussed in most standard texts on decision theory (DeGroot 1970; Berger 1985). The formulation can be re-expressed in term of the log odds. Because the EWOE represents changes to the log odds, it is a metric for determining how much evidence is needed to reach the Bayes decision.

7.3.3 Mutual Information

One problem with EWOE is that the hypothesis must be binary. Pearl (1988) suggests using Mutual Information (MI) to select the best next observation. The *mutual information* of two variables X and Y is defined as:

$$MI(X,Y) = \sum_{x,y} \mathrm{P}(x,y) \log \frac{\mathrm{P}(x,y)}{\mathrm{P}(x)\mathrm{P}(y)} \ . \tag{7.7}$$

This is the Kullback–Leibler distance between the joint distribution and the distribution in which X and Y are independent. It can also be expressed as

$$\sum_{x} \mathrm{P}(x) \sum_{y} \mathrm{P}(y|x) \log \frac{\mathrm{P}(y|x)}{\mathrm{P}(y)} \ . \tag{7.8}$$

Suppose, for example, that S is a proficiency variable with multiple values and $X_1, \ldots, X_n$ are observable variables from n tasks that could be administered. A test selection procedure based on MI would choose the task that maximizes $MI(S, X_j)$, say X_{j^*}. After the outcome x_{j^*} is observed, the process is repeated by finding the maximizing value of $MI(S, X_j|x_{j^*})$, where all of the probabilities in Eq. 7.7 are calculated conditional on $X_{j^*} = x_{j^*}$:

$$MI(S,X|x_{j^*}) = \sum_{s,x} \mathrm{P}(s,x|x_{j^*}) \log \frac{\mathrm{P}(s,x|x_{j^*})}{\mathrm{P}(s|x_{j^*})\mathrm{P}(x|x_{j^*})} \ . \tag{7.9}$$

Note that MI is a symmetric measure; Y provides as much information about X as X provides about Y. A number of Bayesian network software packages will calculate mutual information (Appendix A.1.1).

7.4 Test Construction

What has gone before has almost given us a theory of adaptive test construction. Simply maximizing value of information, or in the absence of true utilities, a quasi-utility such as weight of evidence, produces a test form that is useful for a particular purpose. Indeed, Madigan and Almond (1995) propose such a theory of test selection, and much of the material both here and in the previous section is an adaptation of the methods described there to the educational setting. Section 7.4.1 uses this perspective to explore relationships between item response theory computer adaptive testing (IRT-CAT) and Bayesian network CAT. Section 7.4.2 picks up on a technique called critiquing recommended by Madigan and Almond (1995) to improve the coherence of the test forms generated by such a procedure.

7.4.1 Computer Adaptive Testing

There is a large body of theory and practice associated with item response theory (IRT) computer adaptive testing (CAT) (Wainer et al. 2000 and van der Linden and Glas 2010, present overviews). We briefly describe a basic version of IRT-CAT to provide a feel for key ideas in adaptive testing. The proficiency model consists of a single real-valued variable, θ. At any point in time, our state of knowledge about a student taking the test consists of a posterior distribution over θ. A simple way to make a selection of the next item is to calculate a MAP estimate (posterior mode) estimate for θ and pick the item that maximizes Fisher information at that value of θ.

In this framework, CAT is essentially myopic search using Fisher information as a quasi-utility. The search is *myopic* because we are only searching for the single best task or item at each point of time. There may well be a combination of tasks or items which does better than any single task. Furthermore, the best combination may not include the best single task. Generally speaking, finding a single best task is a simpler search problem than finding the best combination. One faces a time versus optimality problem.

As Fisher information is defined for continuous variables only, it is not directly applicable when the proficiency model is a Bayesian network consisting of discrete variables. However, we could use other measures, for example weight of evidence on some hypothesis of interest, perhaps an overall proficiency variable. Another suggestion by Henson and Douglas (2005) is to try to minimize entropy over the proficiency model. This should have the effect of trying to move the probability mass toward a single proficiency profile.

One issue with adaptive testing is when to stop. In IRT-CAT the stopping rule can be based on the number of items presented or a target posterior variance (in practice, time limits, maximum and minimum test lengths, content constraints, and exposure rates for items are considered jointly with an information metric; van der Linden and Glas 2010). In a profile scored assessment, a stopping rule analogous to target posterior variance can be based on the target classification probability.

If we know the potential actions that could be taken on the basis of an assessment, it would be even better to base the stopping rules on the consequences of the decisions. For example, if it is known that assigning a student a particular remedial treatment is an optimal decision if the probability that she has mastered a skill is below 25 %, then as soon as her posterior probability for that skill falls below 25 % we can stop testing and start providing that instruction.

These kinds of stopping rules are particularly useful in the context of an Intelligent Tutoring or e-Learning system. Such a system moves between two modes: an assessment mode and an instructional mode. Presumably such a system would start in assessment mode and continue assessing the learner's state of knowledge until the probability that a particular piece of instructional material is useful exceeds a certain threshold. At this point, it would switch to instructional mode until the learner completes that unit. It would then switch back to the assessment mode to gauge the state of the student's progress. Such a system could be enhanced by including a model for student growth and learning in the assessment model (Sect. 16.2.2).

7.4.2 Critiquing

Standard IRT-CAT is unidimensional, although the ideas have been extended to multidimensional proficiency models in the context of both IRT (Mulder and van der Linden 2010) and cognitive diagnosis (Cheng 2009). Bayes nets are well-suited to multivariate proficiencies and amenable to adaptive testing as well. Sometimes there is a node in the model for overall proficiency, but in some domains, there is no single dominant proficiency. Even if there is a node for overall proficiency, presumably the multidimensional model has been chosen because the other dimensions are of interest in addition to the overall proficiency. Thus the activity selection algorithm must support multiple purposes.

EWOE is attractive as a measure of test quality because it focuses on one potential purpose at a time, as operationally defined by a particular H. The EWOE is like a spot meter for a test, looking at how much power the test has with respect to a particular purpose. An assessment design with multiple conceivable Hs must then balance the design over each potential purpose of interest.

Picking a single, main purpose has a marvelous focusing effect. In particular, choosing a quasi-utility focused on that purpose and maximizing it produces a test design optimized for that purpose. This produces a universal rule for test construction. Maximizing weight of evidence for an overall proficiency variable is one way to achieve this. The mutual information and expected value of information principles are two others.

As an alternative, one could use a task selection strategy based on a global measure of information calculated across all proficiencies. Suppose for example there are n dichotomous proficiency variables and $H_1, \ldots, H_n$ are hypotheses corresponding to mastery of each of the corresponding skills. A plausible task selection strategy is to choose whichever task provides the greatest expected information, looking across all items and all proficiencies. A criticism of this kind of automated test selection strategies is that they tend to meander. The sequence of selected tasks can cycle rapidly through different topics, an item for one hypothesis and the next item for a different hypothesis according to where the maximum information happens to appear (Barr and Feigenbaum 1982, p. 82). This slows down the student who must mentally switch topics for each new task.

Example 7.5 (Adaptive Content with the Evidence-Based Diagnosis (ACED) Sequences Test). *Shute (2006) and Shute et al. (2008) describe an assessment system designed for a sequences unit of an eight grade algebra system (also see Shute 2004; Shute et al. 2005). The proficiency model was a Bayesian network with 42 nodes (Fig. 7.7). At the top level is a variable for overall proficiency in sequences. Just below that are three variables measuring proficiency in the three types of subsequences,* arithmetic, geometric *and* other recursive, *as well as a proficiency for recognizing sequences. The lower level nodes are skills corresponding to proficiency with tasks based on various ways sequences can be presented and manipulated. Appendix A.2 shows where the ACED models can be accessed online.*

Suppose we used an activity selection model for ACED (Example 7.5) that maximizes EWOE for the overall proficiency node. Suppose that the first item chosen provided direct evidence about arithmetic sequences. If arithmetic, geometric, and other recursive sequences provide equal input into the overall proficiency construct, the next best item will come from one of the other two branches. The third item will come from the remaining branch. Only at the fourth item, we will potentially return to the arithmetic sequences. This will force the student to switch contexts between each task. The minimum entropy principle will exhibit this tendency even more strongly as it will attempt to make sure we have roughly equal information about all corners of the model.

To counter this problem, Madigan and Almond (1995) suggest using the critiquing approach (Miller 1983). First, elicit a suggested hypothesis from the tutor, say $S = s_0$, where S denotes a node associated with a skill, and s_0 is one particular state of S. The system then selects tasks to maximize the probability of quickly accepting or rejecting s_0. Thus, once the tutor has suggested a hypothesis, the system only selects activities that are of high relevance (expected weight of evidence) to that hypothesis. If the hypothesis is rejected, the tutor is prompted for an alternative suggestion, and so on. Once a hypothesis has been chosen, EWOE for the chosen hypothesis becomes the criteria for selecting a task. The sequence of hypothesis would come from the instructional design of the course and could be related to the natural order of learning.

The adaptive algorithm currently used in the ACED prototype of Example 7.5 (Shute et al. 2008) consists of a two stage decision-making process: (1) selection of a target node (what to assess), then (2) selection of a specific task tied to that node (how the target is assessed). The first stage determines

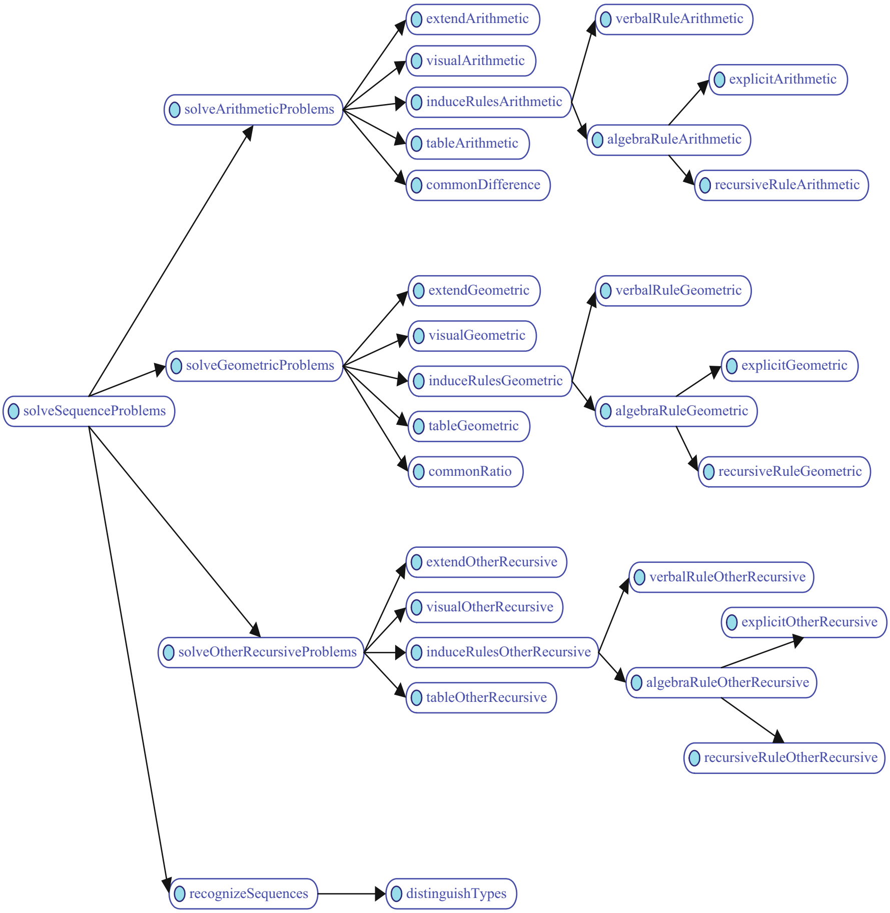


Fig. 7.7 Proficiency model for ACED assessment
Overall proficiency in sequences is divided into three branches and then further divided into nodes corresponding to various skills related to series. To simplify the presentation, only 32 of 42 nodes are displayed.

the appropriate proficiency, represented as a Bayesian network variable, as the target of the proficiency for the adaptive process. Given the hierarchical nature of the cognitive model, the main target variables map to one of the parent nodes of three branches; i.e., solving/understanding (1) arithmetic, (2) geometric, and (3) other recursive sequences. The highest node can also serve as a general target variable: i.e., understanding sequences (which subsumes the three branches). Once the target node (proficiency variable) is selected, a cut point ("High" vs. "Medium or Low" or "High or Medium" vs. "Low") is selected as well. Together the target node and cut point make up a target hypothesis. The idea is that once the value of a target hypothesis is learned to a given threshold of accuracy, then a new target will be selected.

The second stage commences once a target hypothesis (proficiency variable and cut point) is identified as the assessment goal for the next cycle of task administration. The second stage then selects the task from the pool which maximizes the EWOE provided for the hypothesis represented by the target node and cut point. After the outcome from a particular task is received from the learner, we update our beliefs about the learner's proficiency as represented in the Bayes net. If the target hypothesis has been learned to the desired degree of accuracy, the selection process returns to stage one and selects a new target hypothesis. Otherwise, it continues selecting tasks with maximum EWOE (in the updated Bayes net) about the current target hypothesis.

The critiquing strategy can be combined with a strategy of switching between assessment mode and instructional mode mentioned above, to make an activity selection engine for a intelligent tutoring or e-Learning system. Here the instructional goals of the system would be described as a series of goals which the system would try to achieve. Each goal would be formulated as a hypothesis described as a variable and a cut state for that variable. For each goal in turn, the system would first attempt to assess that the hypothesis was true, and if it uncovered sufficient evidence that the hypothesis was false, it would switch to instructional mode and attempt to remedy the problem. The system would continue until all goals are met. Having models for student growth and learning would be useful in this application (Sect. 16.2.2).

7.4.3 Fixed-Form Tests

In some ways, producing a form for an adaptive test seems easier than finding a good form for a fixed-form test. In the adaptive test, choosing the next task requires that at each step we maximize the EWOE for one item based on the observations we have seen so far. Thus we are doing each step of maximization with respect to the conditional distribution, and only calculating the weights of evidence for the observations from one task at a time. This is the myopic search strategy, which is not optimal, but is computationally simple.

For a fixed-form test, we need to maximize the joint expected weight of evidence over all of the variables in the test. As we noted above, the joint evidence as not necessarily the same as the sum of the marginal evidence for each task. This is the nonmyopic search problem, and in general it is hard (at a worst case involving iterations over both proficiency profiles and outcome vectors). Thus, we need approximation methods to tackle this problem.

The problem of assembling optimal test forms from a bank of items has been successfully addressed in the context of item response theory and classical test theory, using the machinery of combinatorial optimization (van der Linden 2005). This approach is readily adapted to assembling optimal collections of tasks for inference about proficiency variables in Bayes nets. This section describes how to express form assembly as a 0/1 linear programming problem.

In a traditional test assembly framework, we have a collection of items which can be selected for the use on a particular form. Let $u_j = 1$ if item j is selected to appear on the form, and let $u_j = 0$ if it does not appear on the form. Linear programming is a mathematical technique which seeks to find a set of 0/1 values for the u_j's that gives the maximum value of an *objective function* given a series of *constraints*.

In ECD, the rules for building a form of the assessment are determined by the assembly model. First, the assembly model is a container which tells the assessment assembler which student, evidence, and task models are available to be used in this assessment. Second it contains a number of rules for the optimization routine: *target rules*—rules that define the objective function,—and *constraints*—rules that define other aspects of the tests.

Using the instructions in the assembly model, the assessment is assembled from a collection of tasks and their corresponding links (Sect. 13.2.3)—versions of the evidence model whose parameters are adjusted for a specific task. (For high-stakes assessments, this usually involves pretesting and calibration, c.f. Part II. For low-stakes assessments, often the default parameter values in the evidence model are good enough.) Note that there is also a Q-matrix corresponding to this collection of tasks. Let $q_{jk} = 1$ if Task j taps Proficiency k (i.e., Proficiency j is in the footprint of the link for Task j).

The objective function is the heart of an optimization problem. It is the quantity the designer wants to minimize or maximize, subject to constraints. We will define objective functions in terms of the amount of information tasks provide for hypotheses about skill profiles. The target rules of the assembly model represent a series of hypotheses. Let $H_1, \ldots, H_M$ be hypothesis concerning the proficiency variables, $\mathbf{S}$. For example, if there are K dichotomous proficiency variables, we could define K hypotheses $H_k : S_k = 1$. We could define additional hypotheses about particular combinations of skills, such as $\{S_k = 1 \text{ and } S_{k'} = 1\}$. If proficiencies have multiple states, we could define hypotheses such as $S_k > \texttt{novice}$. In addition to the hypotheses, $H_1, \ldots, H_M$, we need a corresponding set of weights, $w_1, \ldots, w_M$ indicating the relative importance of the hypothesis. An objective function for assembling a test form is defined as the weighted sum of EWOE (or value of information), the tasks in the assessment. In other words, the objective is to *maximize*

$$\sum_{j=1}^{J} \sum_{m=1}^{M} u_j w_m EW(H_m{:}\mathbf{X}_j), \tag{7.10}$$

where $\mathbf{X_j}$ are the observable outcome variables associated with Task j.

Equation 7.10 has a trivial maximum: simply include every single task in the collection on the form. This is not a very interesting solution, however. For one thing, it is likely to require that the examinee spend far too much time on the test. For another thing, it means that all of the items in the collection will be exposed and none will be saved for later use. A far more realistic solution would be to maximize Eq. 7.10 subject to a constraint that no more than N tasks be used.

In practice, there are a wide variety of constraints that the designers would like to put on the test form. The constraints of the assembly model are how these are recorded. Some typical constraints are:

- *Minimum and Maximum number of tasks.* Constraints of this type are straightforward. Suppose the test design calls for between 30 and 60 tasks. This can be represented with inequalities: $30 \leq \sum u_j \leq 60$.
- *Should be able to complete in specified time.* Usually there is either a fixed time limited or an expected time to complete an assessment. Let t_j be the 75th percentile of the distribution of times required to complete the task in pilot test attempts. The constraint that most students should complete the assessment in 120 min is expressed with the inequality $\sum u_j t_j \leq 120$.
- *Minimum or maximum number of tasks from a certain task model.* Although proficiency variables are defined by subject matter experts based on claims, unless the set of tasks in the form supports the targeted set of claims, the effective meaning of the proficiency variable might be different from the intended meaning. Consider a proficiency variable with the interpretation "Understands Tables and Graphs." If the form consisted of all table tasks, and no graph tasks, the effective meaning of that variable would differ from its label. To avoid such problems, the assembly model could require a minimum number of tasks from a certain task model. To avoid weighting a claim or set of claims too heavily, it could also include a maximum from that task model.

  To express this as an inequality, let $TM_{jn} = 1$ if Task j comes Task Model n and 0 otherwise. Let $\underline{N}_n$ be the minimum number of tasks from Task Model n and let $\overline{N}_n$ be the maximum. This adds one constraint for each task model type of the form $\underline{N}_n \leq \sum_j u_j TM_{jn} \leq \overline{N}_n$.
- *Spanning Contexts.* Often the assessment needs to span content at a finer grain size than that of the task model. Here, task model variables can be used to describe that finer detail of content. For example, a language test might require that there are a mixture of tasks spanning both formal academic and informal non-academic contexts. Let Y_j be a task model variable that represents the context of the task and let Y_{jv} be an indicator variable that takes on the value 1 if task model variable $Y_j = v$ and 0 otherwise. The constraint can be written in the form of a series of inequalities for each possible value v of Y_j, $\underline{NY}_v \leq \sum_j Y_{jv} \leq \overline{NY}_v$.

  Note that we can also consider constraints on task model variables nested within tasks. For example, we could require an easy, medium and hard variant of each task model by appropriately constraining a task model variable in each task corresponding to difficulty.
- *Number of tasks tapping a given proficiency.* Suppose that the proficiency model has several proficiency variables, $S_1, \ldots, S_K$. Almost certainly the

form should achieve some sort of balance among the amounts of evidence gathered about each variable. If the objective function is written as the sum of hypotheses about a number of proficiency variables, the optimization will balance across proficiencies according to the provided weights. A second approach is to write the objective function in terms of some overall ability variable, but to constrain the form so that there is a minimum number of items with direct evidence about a particular proficiency. The second approach has a certain advantage in that it supports reporting in terms of number right scales on subtests. For example, a report that an examinee got 7 out of 10 possible points in tasks which address Skill S_k would provide a good explanation of a score report that said a student's probability of mastery of S_k is 65 %. For that reason, the assembly model may constrain the minimum (or maximum) number of tasks that provide direct evidence of a particular skill.

This is done through the Q-matrix. In particular, $\sum_j u_j q_{jk}$ indicates the number of tasks providing direct evidence for Skill S_k. This sum then forms the object of the constraint. Note that it might be better to work with the version of the Q-matrix that is focused on observables rather than tasks. Then the constraint would be on the number of observations relevant to Skill S_j.

- *Simple versus complex tasks.* Simple structure tasks—tasks that tap exactly one skill variable—hold an important place in form design. Tasks that tap multiple skills usually support competing explanations for poor performance. Adding a few simple structure tasks usually helps disambiguate the causes of the poor performance. On the other hand, complex tasks that tap multiple proficiencies are important because they are often closer to the kinds of authentic tasks that constitute valued work in the domain of the assessment.

  For each Task j, let $QR_j = \sum_k q_{jk}$ be the sum of 1s in the row of the Q-matrix for Task j, indicating the number of proficiency variable parents it has. This is a measure of task complexity. Let $QR_j == c$ be an expression which evaluates to 1 if $QR_j = c$ and 0 if not. Then an constraint on the number of simple structure tasks could be expressed through the expression $\sum_j u_j(QR_j == 1)$. Similarly, the number of tasks providing direct evidence for 2, 3 or more proficiency variables could be constrained. Also, the overall complexity of the the assessment could be constrained by putting upper and lower bounds on the expression $\sum_j u_j QR_j$.
- *Incompatible tasks.* There are a number of cases where we might not want two tasks to appear on the same form. For example, one task may contain the solution to another in its background material. Or two variant tasks from the same task model may be so close that they should not appear in the same form. Or maybe both tasks have a similar context and familiarity with that context might provide an unmodeled dependence among the observables from the two tasks.

This can be also handled with constraints. Often what is done is to create an *enemy list*, E, of tasks that should not appear on the same form. The constraint is then $\sum_{j \in E} u_j \leq 1$. Assessment designs often require many such enemy lists. Note that enemy lists can be defined through task model variables: Any task which has a particular task model variable set to a certain value might appear in an enemy list.

- *Item sets.* One problem that has long been difficult for conventional test design is how to accommodate item sets: items that must appear together on the same form because the share common stimulus material (e.g., a reading passage). Optimization algorithms that use a greedy function to optimize the objective function (Eq. 7.10) can easily get into trouble because once they pick one item from the set, the algorithm must then pick several others from that set. This can lead to bad forms if the remaining items in the set are too hard, too easy, or do not meet other constraints. Although the problem is more noticeable in adaptive testing, it makes the optimization problem more difficult in fixed form tests as well.

  The ECD model avoids many of the difficulties by assembling forms from tasks instead of items. Usually, an item set can be modeled as a single task. As the selection algorithm considers the joint evidence from the task, it is harder to get stuck by making a poor initial choice. However, there may be other considerations here: tasks may be bound together in scenarios, or some but all items from a set might be needed. Additional constraints can be written to meet these conditions.

The assembly model must contain a target rule and at least one constraint. There can be as many or as few constraints of each type as are needed to express the intent of the designers and to ensure that sufficient evidence is gathered for all of the claims. By expressing the target rule as a function of the task indicators, u_j, and the constrains as inequalities using the task indicators, one can use standard 0/1 linear programming to assemble a test form that resembles a previous form or to assemble multiple parallel test forms. Standard optimization theory and software (e.g., Nocedal and Wright 2006) can be applied. Alternative approaches and many insights to particular challenges and kinds of constraints in test-assembly more generally are found in van der Linden (2005).

7.5 Reliability and Assessment Information

When we build the evidence model $\mathrm{P}(\mathbf{X}|\mathbf{S})$ we are acknowledging that the relationship between the proficiency variables, $\mathbf{S}$ and the observed outcomes, $\mathbf{X}$ is a probabilistic one. In other words, the outcome pattern $\mathbf{X}$ is not a pure measure of the proficiency variable $\mathbf{S}$, but contains some noise that is irrelevant to what we are trying to measure. In engineering terms, we could think about the signal to noise ratio for the assessment; in psychometrics we speak of the *reliability*.

Note that not all sources of noise are actually irrelevant to the construct we are trying to measure. Take for example an assessment of communicative competence in a given language. By restricting the setting to the academic environment, we remove one source of variability. However, other settings may also be relevant to the kinds of inferences we are trying to make. For example, if we are trying to understand how well a potential student is likely to be able to get by living in a foreign country, settings related to shopping and interacting with the local bureaucracy may be equally important. In assessment, reliability is usually taken as a measure of the irrelevant sources of variability given a specified domain of tasks and test procedures (Brennan 2001).

Our treatment of reliability with Bayes nets differs from that of classical test theory in two important respects. First, if our proficiency model is expressed as a Bayesian network, then our scores will typically be either classifications of participants according their proficiency variables or posterior distributions over one or more proficiency variables. The majority of the literature on reliability is devoted to continuous or integer valued scores. Even when authors do talk about classifications, it is usually in the context of a cut score on a continuous variable. Second, classical test theory relies on the concept of a *true score*. Typically, the distribution of the true score in the population is unknown and must be estimated from data. In our case, the true score corresponds to the skill profile, $\mathbf{S}$. The proficiency model provides the population distribution for the skill profile, $\mathrm{P}(\mathbf{S})$.

For simplicity, we start with purely discrete scores, where the student is classified as having the skill profile with the highest posterior probability (the MAP estimate). Section 7.5.1 looks at some measures of accuracy, and Sect. 7.5.2 looks at some measures of consistency between two test forms. However, the Bayes net score is not just a single best proficiency profile, but rather a probability distribution over possible profiles. These contain more information than the point estimates, and hence are usually better scores. Section 7.5.3 extends the discrete accuracy and consistency measures to this continuous world.

7.5.1 Accuracy Matrix

We start with a classification score. We partition the space of skill profiles into a series of disjoint hypotheses, $H_1, \ldots, H_K$, which span the space of possible skill profiles. A common case is to look at the value of one proficiency variable ignoring the others; that is $H_j : S_j = \texttt{expert}$. A more sophisticated model might look at a number of possible courses a student could be placed in and what kinds of student would benefit from which course, yielding a partitioning of all possible vectors in $\mathbf{S}$ according to placement based on proficiency profiles. When there are exactly two hypotheses, this corresponds to the setup in the weight of evidence calculation above. But when the students are to be classified into more than two categories, a new measure is needed which extends to multiple categories.

Suppose that we observe a pattern of outcomes $\mathbf{X}$ from the collection of tasks that appears on one form of the assessment. By Bayes' theorem we obtain the posterior distribution $\mathrm{P}(\mathbf{S}|\mathbf{X})$, and the implied posterior probability for each hypothesis in the partition, or $\mathrm{P}(h_k|\mathbf{X})$. We can then define a point estimate for H by $\hat{H} = \max_{h_k} \mathrm{P}(h_k|\mathbf{X})$. This is the *Maximum A Posteriori* or *MAP* estimate for H. It will be a function of $\mathbf{X}$ so we can write $\hat{H}(\mathbf{X})$.

Doing this assumes that the utility function associated with misclassification is relatively symmetric. That might not always be the case. Again in a licensure test it is more regrettable to license somebody who is not qualified than to make the opposite mistake. Similarly, it may be much more regrettable to fail to identify a student who needs remediation than the opposite. In such case, instead of choosing the value of $\hat{H}$ which maximizes the posterior probability, we would take the one which maximizes expected utility. This is called the *Bayes decision* and is covered in standard texts on decision theory (e.g., DeGroot 1970; Berger 1985).

We define the elements of the *accuracy matrix*[2] A as follows:

$$a_{ij} = \mathrm{P}(H = h_i, \hat{H} = h_j) = \sum_{\mathbf{x}:\hat{H}(\mathbf{x})=h_j} \mathrm{P}(\mathbf{x}|H = h_i)\mathrm{P}(H = h_i) \ . \tag{7.11}$$

This is the probability that when h_i is the correct hypothesis, a response vector $\mathbf{x}$ will be observed for which $\hat{H} = h_j$ is the decision. The diagonal of this matrix corresponds to the cases where the decision agrees with the true classification. Perfect agreement would result in a diagonal matrix. Thus, we can define the *accuracy* as the trace of the matrix, that is, $\sum_k a_{kk}$.

The accuracy matrix A may be difficult to calculate analytically, especially for a long assessment. In general, evaluating Eq. 7.11 involves iteration over both the set of possible skill profiles and the set of possible outcome patterns. This becomes prohibitively expensive as the number and complexity of the tasks in the assessment increases. However, it can be easily estimated by a simple simulation experiment. First randomly select a skill profile according to the distribution of the proficiency model, $\mathrm{P}(\mathbf{S})$. We can then assign the value of H based on the selected skill profile $\mathbf{S}$. Next, we randomly select an outcome pattern $\mathbf{X}$ according to the distribution of the evidence model $\mathrm{P}(\mathbf{X}|\mathbf{S})$. We can then classify the simulated outcome with an estimated value of the hypothesis $\hat{H}(\mathbf{X})$. If we repeat this experiment many times, the observed frequencies will converge to the accuracy matrix.

This experiment contains two important assumptions: The first is that the model is correct, i.e., the model used to generate the data is the same as the one used in the classification. In practice we can never know the true data generation model. The second is that there is no accounting of the uncertainty

[2] This is sometimes called a *confusion matrix*, referring to its off-diagonal elements.

about the probabilities (or parameters from which they are obtained) used to generate the **S**s and the **X**s. Part II takes up the issue of uncertainty about the parameters. Taking these problems into consideration, calculating the accuracy matrix estimate in this fashion is really an upper bound on the true accuracy of the assessment.

Example 7.6 (Language Test Accuracy Matrix). *Consider once more the simplified language test from Mislevy (1995c) described in Example 7.1, (see also Appendix A.2). Suppose we perform the following experiment. First, we simulate 1000 possible proficiency profiles from the proficiency model. Next, we generate a response vector over 63 geometric sequence tasks for each of the 1000 simulees. Finally, we score the test for all 1000 simulees (ignoring their actual proficiency profiles). For each simulee, we should now have both their "true" (from the simulation) value of the* Reading, Writing, Speaking *and* Listening *nodes and the most likely (MAP) estimate for each node from the scored response.*

We can calculate the accuracy matrix as follows: First, set $a_{ij} = 0$ for all i and j. Next, for each simulee, if the true value of Reading *is i and the MAP estimate is j, add one to the value of a_{ij}. Repeating this for all 1000 simulees and dividing by 1000 (the number of simulees) yields a matrix like Table 7.1.*

Table 7.1 Accuracy matrices for *Reading* and *Writing* based on 1000 simulated students

	Reading			*Writing*		
	Novice	Intermediate	Advanced	Novice	Intermediate	Advanced
Novice	0.229	0.025	0.000	0.163	0.097	0.000
Intermediate	0.025	0.445	0.029	0.053	0.388	0.065
Advanced	0.000	0.040	0.207	0.000	0.051	0.183

Some authors take the accuracy defined in this way as a measure of validity rather than one of reliability. However, this contains the implicit assumption that the set of hypotheses $H_1, \ldots, H_K$ and by extension the proficiency variables **S** represent the construct on which we wish to base our decisions. We prefer to think of the accuracy as a measure of internal consistency under the model, that is, of reliability, and reserve the term "validity" for measures which take into account the use and consequences of the classification (Sect. 16.4).

Many other important measures of agreement can be derived from the accuracy matrix. In defining those measures, it will be helpful to have notation for the row and column sums. Let $a_{i+} = \sum_j a_{ij} = \mathrm{P}(H_i)$ be the sum of all of the elements in Row i. This is the marginal (population) probability of the hypothesis. Let $a_{+j} = \sum_i a_{ij} = \mathrm{P}(\hat{H}_j)$. This is the marginal probability of the classifications.

Simply normalizing the accuracy matrix by dividing by the row sums produces interesting results. The matrix normalized in this way, $a_{ij}/a_{i+} = \mathrm{P}(\hat{H} = h_j|H = h_i)$, produces the operating characteristics of the assessment. If the hypothesis is binary, then a_{11}/a_{1+} is the *sensitivity* of the test, the probability of asserting that the hypothesis holds when it is in fact true. Similarly, a_{22}/a_{2+} is the *specificity* of the test, the probability of asserting that the hypothesis is false when in fact it is false. These terms are used frequently in medical testing.

Often the multiple measures of the operating characteristics are more useful than a single measure describing accuracy. This is particularly true because most of the single number summaries depend on the population distribution of H. The operating characteristics specifically condition out this distribution. They are still a useful measure of the strength of evidence in the assessment even when all of the members of the sample have the same value for the hypothesis.

Normalizing by the column sums also has another interesting interpretation. Now we are conditioning on observed decision, $a_{ij}/a_{+j} = \mathrm{P}(H = h_i|\hat{H} = h_j)$. The resulting conditional probability distributions answer the question, “If the assessment classifies a participant as $\hat{H}$, what is the probability that this is the true classification?” This is the question that end users of the assessment scores would very much like answered. As in the rare disease problem (Example 3.6), the probability of true classification depends on both the operating characteristics of the test and the population distribution of the hypothesis.

The accuracy, $\sum_i a_{ii}$, answers the question “What is the probability that the classification assigned on the basis of this assessment will agree with truth?” Note that it is possible to get a fairly large agreement by chance, even if the classification and truth are independent. Consequently, some authors recommend adjusting the accuracy for chance agreement.

One such adjusted agreement is *Cohen's* κ. Fleiss et al. (2003) note that adjusting for chance agreement unifies a number of different measures of agreement in a 2 by 2 table. If the true value of the hypothesis, H and the estimated hypothesis, $\hat{H}$ were two raters acting independently, the probability of agreement by chance would be $\sum_i a_{i+}a_{+i}$. The coefficient κ is expressed as a ratio of the obtained accuracy corrected for chance to the ideal accuracy:

$$\kappa = \frac{\sum_i a_{ii} - \sum_i a_{i+}a_{+i}}{1 - \sum_i a_{i+}a_{+i}} \ . \tag{7.12}$$

The chance term is based on the idea that the two classification mechanism are independent. Thus Cohen's κ answers the question “How much better is the classification given by this assessment than what we would expect if the assessment was independent of truth?” Sometimes when the categories are ordered, κ is weighted so that classifications that are one category away are worth more that classifications that are multiple categories away. In either case, κ is easier to interpret as a measure of the consistency of two classifiers (Sect. 7.5.2) than the accuracy of one classifier.

Goodman and Kruskal (1954) offer a different statistic, λ, that using a different baseline, $\max(p_H)$, for adjusting the agreement statistic (see also Brennan and Prediger 1977). This is the agreement level that would be achieved by simply classifying everybody at the most likely state.

$$\lambda = \frac{\sum_n a_{n,n} - \max_n a_{n,+}}{1 - \max_n a_{n,+}}.$$

The metric λ corresponds to the question, "How much better do we do with this assessment than simply classifying each person at the population mode?" This index relates directly to the decision of whether or not to use the test.

Although less well known that Cohen's κ, Goodman and Kruskal's λ is often a better choice when talking about the accuracy of an assessment (regardless of the method used to obtain the estimates). In particular, the question answered by λ is often more interesting. While κ answers how much better is the agreement (between the truth and the classifier), λ answer the question how much better is it to use the classifier than not. In fact, neither measure may be the ideal measure; Goodman and Kruskal (1954), offer a number of alternatives that could be explored.

Example 7.7 (Simplified Language Test Accuracy Matrix, Kappa and Lambda). *Using the estimated accuracy matrix for the simplified language test (Table 7.1, Example 7.6), we can calculate Cohen's κ and Goodman and Kruskal's λ. To begin, we find the diagonal of the Reading portion of Table 7.1; this is 0.881. In other words, this form of the assessment classifies slightly almost 90 % of the examinees correctly on reading. Next, we sum over the rows and columns to produce marginal distributions for the true proficiency levels,* $(.254, .499, .247)$*, and the estimated proficiency levels,* $(.254, .510, .236)$.

To calculate κ, we need to calculate the probability of chance agreement. We get this by multiplying the two vectors of marginal probabilities and taking the sum, which yields 0.377. Thus, about 1/3 of the time we are likely to get the correct classification just by chance. The adjusted agreement is now $\kappa = (0.881 - 0.377)/(1 - 0.377) = 0.809$, *which means we are getting approximately a 80 % improvement over the agreement we would have gotten if we just assigned everybody a label randomly.*

To calculate λ, we note that the modal category is `Intermediate`*, and that the probability that a randomly chosen simulee has* `Intermediate` *ability is .499. In other words, if we rated everybody as* `Intermediate` *we would get approximately 1/2 of the ratings correct. The adjusted agreement is now* $\lambda = (0.881 - 0.499)/(1 - 0.499) = 0.763$, *which means we are getting approximately a 75 % improvement over the agreement we would have gotten if we just assigned everybody the label* `Intermediate`.

Turning to the `Writing` *variable, similar calculations show $\kappa = .567$ and $\lambda = .462$. These numbers are smaller as a consequence of the test design. In particular, of the 16 tasks in this assessment all but the 5 listening tasks involve at least some Reading and hence provide evidence for Reading. Only the three Writing tasks provide direct evidence for Writing, and because those are integrated tasks that also involve Reading, their evidence is weaker. Thus, both λ and κ are smaller.*

Note that the Bayesian network does not actually assign each student to a category, rather it gives a probability distribution over the categories. We can do slightly better if we look at the probabilities rather than just the most likely category. Section 7.5.3 explores this. In fact, this is one example of a scoring rule for Bayesian networks; Chap. 10 explores other scoring rules.

7.5.2 Consistency Matrix

Suppose that we have two parallel forms of the assessment, Form X and Form Y. We could produce two accuracy matrices A_X and A_Y, one for each form. The *Consistency Matrix* is the product of those two accuracy matrices, $C = A_X^t A_Y$. The normalized rows and columns represent conditional probabilities which describe what we expect to happen when a person who takes Form X later takes Form Y and *visa versa.* This is of practical importance to testing program where the same assessment (with alternative forms) is given over and over again to a similar population of examinees. In this case, large shifts in the classification is likely to produce confusion among the test-takers and score-users.

The consistency matrix can be estimated with a simulation experiment as described above, it can also be estimated by giving both Form X and Form Y to a sample of examinees. If the test is long enough, it could also be used to form split-half estimates of reliability. However, this may be tricky with a diagnostic assessment. In particular, there may only be a few tasks providing direct evidence for each proficiency of interest. Hence, the half-tests may be very unbalanced or have very low reliability.

The *consistency* is the sum of the diagonal elements of the consistency matrix, $\sum_i c_{ii}$. This answers the question, "What fraction of examinees who take both Form X and Form Y will get the same classification with respect to hypothesis H?" Cohen's κ is frequently used with the consistency matrix as well. It answers the question "How much better do the two forms agree than two form which are independent, that is measuring different aspects of proficiency?" Again, it may be worth exploring some of the other measures described in Goodman and Kruskal (1954) as well.

7.5.3 Expected Value Matrix

One source of error in both the accuracy matrix and classification matrix is that we are assigning a person to a class on the basis of the MAP estimate for the hypothesis. This gives equal weight to someone who we believe with high confidence is in one category and someone who is on the border between two categories. By reporting the marginal probability of classification rather than the MAP estimate, we should better convey our uncertainty about the truth.

Suppose we simulate *proficiency profiles* and outcome vectors from N simulees. Let $\mathbf{S}_n$ be the proficiency profile for Simulee n and let $\mathbf{X}_n$ be the outcome vector. Then $\mathrm{P}(H|\mathbf{X}_n)$ is the probabilistic classification that we would assign to Simulee n on the basis of the outcome vector $\mathbf{X}_n$. We can define a *probabilistic classification matrix for Hypothesis H* by summing over these classifications.

$$z_{ij} = \sum_{n:H(\mathbf{S}_n)=h_i} \mathrm{P}(H = h_j|\mathbf{X}_n). \tag{7.13}$$

Here, z_{ij} is the weighted number of individual whose true classification is h_i who are classified as h_j, where their weights are the posterior probability of classification in that class. In other words, in the simulation to estimate the accuracy matrix we place a simulee from the ith category into the cell for the jth decision category; to estimate the expected value matrix, we distribute that simulee across all n cells in the jth column, according to its posterior probabilities for each.

The sum of the diagonal elements, $\sum_i z_{ii}$, is another measure of accuracy for the assessment. We can also look at Cohen's κ and λ for the probabilistic classification as well. In general, these should do at least as well as their non-probabilistic counterparts.

Another way to treat the probabilistic scores, $\mathrm{P}(H|\mathbf{X}_n)$ is to regard them as predictions of the true value of the hypothesis. In this case within the confines of the simulation experiment, we can use the scoring rules in Chap. 10 to evaluate the quality of the assessment for making this particular prediction.

Example 7.8 (Simplified Language Test Expected Accuracy Matrix). *The procedure is similar to the one used in Example 7.6. The initial simulation proceeds in the same way. It differs at the scoring step, instead of calculating the MAP score for Reading we calculate its marginal probability. This score will be a vector of three numbers over the possible classifications (`Novice`, `Intermediate`, and `Advanced`). The "true" value is still a single state.*

We can calculate the expected accuracy matrix as follows: First, set $z_{ij} = 0$ for all i and j. Next, for each simulee, let the true value of SolveGeometricProblems is i and the marginal estimate be $\{p_1, p_2, p_3\}$, which is a probability vector. We update the values of z_{ij} by adding the probability vector to Row i, that is, let $z_{ij} \leftarrow z_{ij} + p_j$ for $j = 1, 2, 3$. Repeating this for all 1000 simulees yields a matrix like Table 7.2. We divide the entries in this matrix by 1,000 to put all of the numbers on a probability scale.

The agreement measures $\tilde{\kappa}$ and $\tilde{\lambda}$ are calculated in the same way. In this case, $\tilde{\kappa}_{Reading} = .73$, $\tilde{\lambda}_{Reading} = .66$, $\tilde{\kappa}_{Writing} = .43$, and $\tilde{\lambda}_{Writing} = .28$. These are lower than the agreement rates based on the modal classifications (Example 7.7), but are more honest about the uncertainty in the classifications.

Table 7.2 Expected accuracy matrices based on 1000 simulations

	Reading			*Writing*		
	Novice	Intermediate	Advanced	Novice	Intermediate	Advanced
Novice	0.220	0.034	0.000	0.162	0.092	0.007
Intermediate	0.037	0.413	0.050	0.091	0.331	0.084
Advanced	0.000	0.049	0.198	0.003	0.076	0.154

7.5.4 Weight of Evidence as Information

The preceding discussion has introduced many different possible measures for reliability, not just one. That is because when a test user asks about the reliability of an assessment, there are a number of possible motivations for that question. She might be asking about how the results from the assessment varies when sampling tasks from a pool of possible tasks. In this case, consistency is the most appropriate answer. She might be asking about how well the assessment captures the variability in the population; in this case accuracy, perhaps as measured by Cohen's κ is a reasonable choice. She might be asking whether or not it is worthwhile to give the assessment to learn about a hypothesis, H. In this case, λ seems appropriate.

Smith (2003) presents another possible meaning for reliability, namely "Sufficiency of Information." Smith points out that a teacher may give an end of unit quiz expecting all of the students to get all of the items correct. After all, having finished the unit, the students should have mastered the material. This quiz serves several important purposes: (1) it helps the student's self-assessment of their progress on this material, and (2) it identifies for the teacher any students who have not yet mastered the material as predicted. This assessment has value, even though by many of the reliability measures posed above may have trivial values because all of the students are expected to have mastered the material.

Note that the EWOE does not depend on the population distribution of the hypothesis. The calculations for the EWOE are done with the conditional distribution given the hypothesis. Thus, if an assessment has a high EWOE for the hypothesis that the students have mastered the material of the unit it will be appropriate. (It still may be difficult to estimate the task specific parameters from the classroom population as there is little variability with respect to the proficiency variables of interest, but that is a separate problem.)

We have seen that weight of evidence provides a useful mechanism for explaining how certain patterns of evidence influence our conclusions about certain proficiency variables. Furthermore, its ability to act like a spot meter for specific hypotheses helps us to evaluate how much information is provided by a proposed assessment design for a specific purpose. If the assessment does not provide enough information, we could consider altering the assessment design, that is the Q-Matrix associated with the collection of tasks in an assessment form, to obtain more information about the proficiency variables of interest. Lengthening the test is one mechanism for altering the Q-Matrix; replacing simple structure tasks that tap just one proficiency variable with complex tasks that tap multiple variables, or vice versa, is another. However, careful test design requires balancing the *cost* (the biggest component of which is usually the time the examinee spends taking the test) with both the information gained and the complexity of the calculations (Sect. 7.4.3).

Exercises

7.1 (Order dependence of WOE). Recall the 5-item math quiz from Example 6.1. Suppose that a student gets items 1, 2, and 4 correct and items 3 and 5 incorrect. Calculate the weight of evidence for *Theta* > 0 provided by each item under the following assumptions:

1. The student works through the problems in order from item 1 to item 5.
2. The student works through the problems in reverse order from item 5 to item 1.

7.2 (Context variable and WOE). To see what effect the *Context* variable has on the weight of evidence, compare the five item IRT model without the context variable (Sect. 6.1) to the model with the *Context* variable (Sect. 6.2). Calculate the weight of evidence for *Theta* > 0 provide by the following evidence for both models:

1. *Item 3* and *item 4* are both correct.
2. *Item 3* and *item 4* are both incorrect.

For the following exercises, consider a simplified version of the ACED model (Shute 2004) given in Fig. 7.8. This simplified model uses the same conditional probability table given in Table 7.3 for all proficiency variables except for *SolveSequenceProblems*, which has a uniform distribution. Also, attached to all nodes except for *SolveSequenceProblems* are three tasks meant to tap that proficiency. There are three variants of the tasks, a *Hard*, *Medium* and *Easy* version. Each evidence model fragment adds one observable variable with a conditional probability table given by one of the columns in Table 7.4.

7.3 (Direct and indirect evidence). Suppose that a person is assigned a single medium difficulty task attached to the *SolveArithmeticProblems* proficiency and the person gets that item correct. Calculate the weight of evidence provided for both *SolveArithmeticProblems* $=$ `H` and *SolveGeometricProblems* $=$ `H`. Why is the first higher than the second?

7.4 (Chaining weight of evidence). Consider two medium difficulty tasks, one attached to *AlgebraRuleGeometric* and one attached to *InduceRulesGeometric*. Calculate the weight of evidence that getting a correct score on

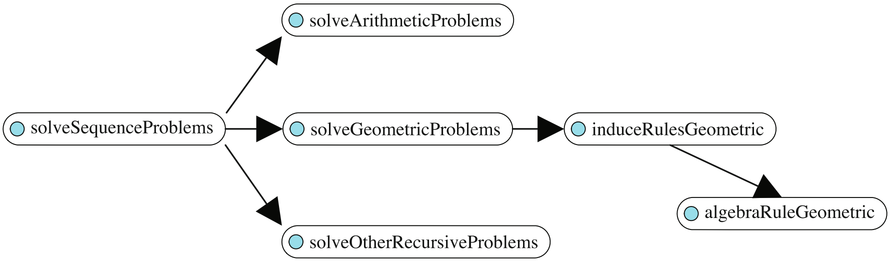


Fig. 7.8 Subset of ACED proficiency model for exercises

Table 7.3 Conditional probabilities for ACED subset proficiency model

Generated using regression distribution (Sect. 8.5.4) with correlation of .8. All of the conditional probability tables in the proficiency model use this same table

	Subskill		
Skill	H	M	L
High	0.83	0.09	0.08
Medium	0.40	0.20	0.40
Low	0.08	0.09	0.83

Table 7.4 Conditional probabilities for evidence models for ACED subset

Evidence model fragments that consist of a single dichotomous observable variable attached to one of the proficiency nodes (except the overall proficiency). Depending on the difficulty of the task, one of the three columns of this table indicates the conditional probability for a correct response

Skill	P(*Easy* = 1)	P(*Moderate* = 1)	P(*Hard* = 1)
High	0.88	0.72	0.49
Medium	0.73	0.50	0.27
Low	0.51	0.28	0.12

each of those tasks (with on other evidence) provides for the proposition *SolveGeometricProblems* = H. Why is the one smaller than the other?

7.5 (Correlation and weight of evidence). Make an alternative model for the ACED subset problem by substituting the conditional probabilities given in Table 7.5 for the conditional probabilities for the nodes *AlgebraRuleGeometric* and *InduceRulesGeometric*. (Both the original and alternative conditional probabilities tables were produced using the Regression Distribution, Sect. 8.5.4. The original has a correlation of 0.8, while the alternative has a correlation of 0.9).

Table 7.5 Alternative conditional probabilities for ACED subset proficiency model

Generated using regression distribution (Sect. 8.5.4) with correlation of .9.

	Subskill		
Skill	H	M	L
High	0.90	0.07	0.03
Medium	0.39	0.22	0.39
Low	0.03	0.07	0.90

7.6 (Compensatory, conjunctive and disjunctive distributions). Consider the compensatory, conjunctive, and disjunctive distribution models from Sect. 6.3. Calculate the weight of evidence for the proposition $P1 = \mathtt{H}$ for all three models under the following conditions:

1. The observable is Right.
2. The observable is Wrong
3. The observable is Right conditioned on the proposition $P2 = \mathtt{H}$.
4. The observable is Wrong conditioned on the proposition $P2 = \mathtt{H}$.

7.7 (Task difficulty and EWOE). Consider a test which assesses a single proficiency, *Skill*, which can take on the values High, Medium and Low. Assume that all tasks are scored with one of three different evidence models with observable variables *Easy*, *Moderate*, and *Hard* (one observable per evidence model) which take on the values 0 (wrong) and 1 (right). Let *Skill* have a uniform distribution and the conditional probability tables for the observables follow the distributions given in Table 7.4. (Note, these distributions were produced by using the DiBello-Samejima models, Sect. 8.5, with difficulty parameters of −1, 0 and +1 for easy, moderate and hard tasks.)

1. Calculate the EWOE for $\mathit{Skill} = \mathtt{High}$ versus $\mathit{Skill} \in \{\mathtt{Medium}, \mathtt{Low}\}$ for each kind of task. Which kind of task provides the best evidence for this distinction?
2. Calculate the expected weight of evidence for $\mathit{Skill} = \mathtt{Low}$ versus $\mathit{Skill} \in \{\mathtt{Medium}, \mathtt{High}\}$ for each kind of task. Which kind of task provides the best evidence for this distinction?

7.8 (Effect of prior on EWOE). Consider the assessment described in Exercise 7.7 using only the medium difficulty task. However, this time consider three different prior distributions for *Skill*: $(.33, .33, .33)$, $(.6, .2, .2)$, and $(.6, .3, .1)$, for the states High, Medium, and Low respectively.

1. Calculate the weight of evidence for the proposition $\mathit{Skill} = \mathtt{High}$ when the observable is correct under all three priors.
2. Calculate the weight of evidence for the proposition $\mathit{Skill} = \mathtt{High}$ when the observable is incorrect under all three priors.

3. Calculate the expected weight of evidence the medium task provides for the proposition $\mathit{Skill} = \texttt{High}$ under all three priors.

How do the weight of evidence and the expected weight of evidence change with the change in prior?

7.9 (Value of information). Consider the assessment described in Exercise 7.7. Assume that a certain school district has determined that having a student in the `High` state for *Skill* at the end of the year is worth \$ 1200, having a student at the `Medium` state is worth \$ 1000, and having a student at the `Low` state is worth zero (we can always make one particular state worth zero by subtracting a constant from all of the values). Calculate the value of information for a Hard, Medium and Easy task.

7.10 (Additivity of WOE and EWOE). Consider a situation in which neither H nor $\overline{H}$ is compound and hence $X_1 \perp\!\!\!\perp X_2 | H, \overline{H}$. Demonstrate under those conditions that:

1. $W(H{:}x_1, x_2) = W(H{:}x_1) + W(H{:}x_2)$
2. $EW(H{:}X_1, X_2) = EW(H{:}X_1) + EW(H{:}X_2)$

7.11 (Test length and WOE). Consider a test with a single proficiency variable, *Skill*, which takes on two values, `High` and `Low`, and let the prior (population) probability for that task be $(.5, .5)$. Suppose that there is a pool of tasks to assess that skill, all of which have a single observable outcome X which takes on values `correct` and `incorrect`. Assume that the link models are identical for all the tasks in the pool and that $\mathrm{P}(X = \texttt{correct} | \mathit{Skill} = \texttt{High}) = .8$ and $\mathrm{P}(X = \texttt{correct} | \mathit{Skill} = \texttt{Low}) = .2$. Calculate the EWOE for $\mathit{Skill} = \texttt{High}$ provided by a 5 task test, a 10 task test, and a 25 task test. Hint: use the results from the Exercise 7.10.

7.12 (Reading passage topic). For a reading comprehension test for graduate students, the design team intends for there to be between 4–6 tasks calling for a student to read a passage and then answer questions. The design team would like the passages to be reasonably balance among topic chosen from the natural sciences, the social sciences and the humanities. Describe how one might set up constraints so that all forms will meet this criteria.

7.13 (Bayes net versus number right). Consider the assessment described in Exercise 7.7 and a form that consists of 10 medium difficulty tasks. Suppose we take this same assessment and score it with number right instead of the Bayes net. How will the Bayes net and number right scores differ?

7.14 (Accuracy and task difficulty). Consider the assessment described in Exercise 7.7 and two test forms, one consisting of 10 easy tasks and one consisting of 10 hard tasks. Use a simulation experiment to calculate the accuracy matrix for both forms. What can be said about the difference between the two forms?

7.15 (Accuracy of language assessment). Modify the simplified language assessment (Appendix A.2) by doubling the number of tasks of each type. Use a simulation experiment to calculate the accuracy matrix for the modified test.

7.16 (Kappa and Lambda for Speaking and Listening). Calculate Cohen's κ and Goodman and Kruskal's λ for *Speaking* and *Listening* proficiencies using the accuracy matrices in Table 7.6. Compare them to the numbers from Example 7.7, and interpret what they say about the relative information in the test for the four skills.

Table 7.6 Accuracy matrices for *Speaking* and *Listening* based on 1000 simulated students

	Speaking			*Listening*		
	`Novice`	`Intermediate`	`Advanced`	`Novice`	`Intermediate`	`Advanced`
`Novice`	0.243	0.027	0.000	0.242	0.054	0.000
`Intermediate`	0.044	0.390	0.030	0.054	0.290	0.059
`Advanced`	0.000	0.033	0.233	0.000	0.087	0.214

Part II

Learning and Revising Models from Data

8

Parameters for Bayesian Network Models

While Part I concentrated on models for one student at a time, Part II expands our horizons to include data from a population of similar students. The most important result of this transition is that we can use experiential data to improve our model. In particular, we can learn about the parameters and structure of the model. Chapter 9 describes a method for learning parameters from data, Chap. 10 introduces some measures of how well our model fits the data and surveys a number of techniques for learning model structure from data. Finally, Chap. 11 illustrates these ideas with an extensive analysis of a single data set.

This chapter talks about various approaches to parameterizing graphical models. There is a large and growing literature on parameterizing Bayes nets and eliciting probabilities from experts more generally (e.g., Díez and Druzdzel 2006; Laskey and Mahoney 2000; O'Hagan et al. 2006; Zapata-Rivera 2002). We bring some of these ideas to bear on the particular context of educational assessment. Section 8.1 introduces basic notation for graphical parameters. Section 8.2 discusses the hyper-Markov Laws, conditional independence relationships among parameters. Section 8.3 introduces the hyper-Dirichlet distribution, the natural conjugate distribution of the Bayesian network. As we will see, the hyper-Dirichlet has many parameters as table size increases, and it is often difficult to assess hyper-Dirichlet priors. The chapter thus explores two different approaches to reducing the number of parameters in the model. Section 8.4 describes models that add a layer of probabilistic noise to logical functions. Section 8.5 describes a suggestion by Lou DiBello to model probability tables using Samejima's graded response model.

8.1 Parameterizing a Graphical Model

The standard directed graphical representation does a good job of hiding the complexity of the graphical model. The edges in the model represent probability tables that may have some internal structure. This hides the work

R. G. Almond et al., *Bayesian Networks in Educational Assessment*,
Statistics for Social and Behavioral Sciences, DOI 10.1007/978-1-4939-2125-6_8

we need to do in parameterizing the model. This chapter will look more closely at the internal structure of conditional probability tables. The next chapter, on estimation, will introduce a representation called plate notation to clarify the independence and dependence relationships across tasks, parameters, and multiple subjects.

The directed hypergraph representation starts to make the internal structure of conditional probability tables more explicit.

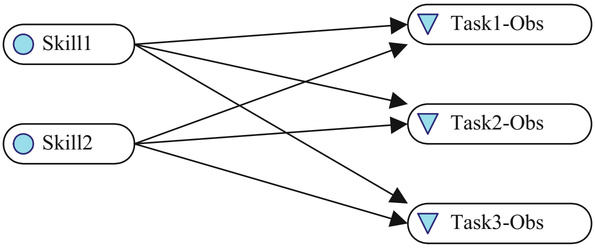


Fig. 8.1 A simple latent class model with two skills
Reprinted from Almond et al. (2006a) with permission from ETS.

We'll start with a simple latent skill model with two skills and observable responses from three tasks, all dichotomous, shown in Fig. 8.1. The symbol ◯ is used to denote Proficiency Model variables that are shared across evidence models and the symbol ▽ is used to denote evidence model variables that are specific to a particular task.

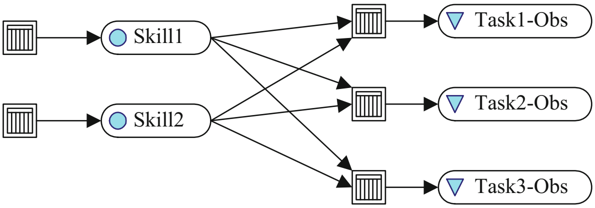


Fig. 8.2 Hypergraph for a simple latent class model with two skills. *Square boxes with tables* represent probability tables that must be learned or elicited from experts
Reprinted from Almond et al. (2006a) with permission from ETS.

Next we add icons to represent hyperedges for factorization to produce Fig. 8.2. Each square box in Fig. 8.2 represents a conditional probability table we must specify. To do learning, we use parametric forms for these tables. The parameters "float above" the model in a second layer. This is shown in Fig. 8.3.

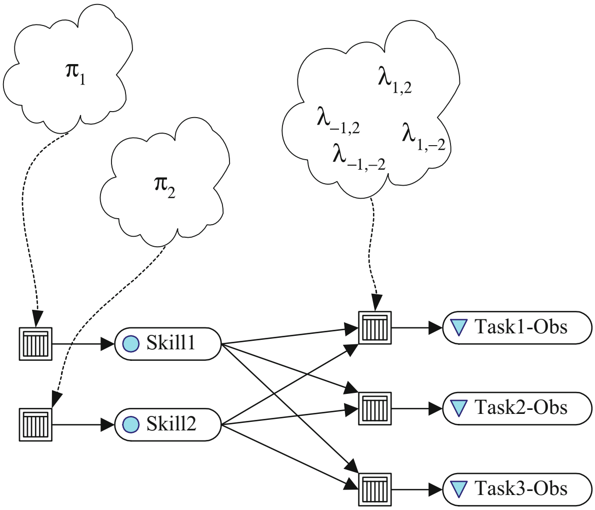


Fig. 8.3 Second layer of parameters on *top* of graphical model. Parameters float above the model. When we want to do calculations for an individual learner with the model, parameter values drop down into the model. Reprinted from Almond et al. (2006a) with permission from ETS.

The parameters for the model in Fig. 8.3 are as follows:

$$\begin{aligned}
\pi_1 &= \mathrm{P}(\bigcirc \mathit{Skill1}) \\
\pi_2 &= \mathrm{P}(\bigcirc \mathit{Skill2}) \\
\lambda_{1,2} &= \mathrm{P}(\triangledown\, \mathit{Task1\text{-}obs} | \bigcirc \mathit{Skill1}, \bigcirc \mathit{Skill2}) \\
\lambda_{-1,2} &= \mathrm{P}(\triangledown\, \mathit{Task1\text{-}obs} | \neg\bigcirc \mathit{Skill1}, \bigcirc \mathit{Skill2}) \\
\lambda_{1,-2} &= \mathrm{P}(\triangledown\, \mathit{Task1\text{-}obs} | \bigcirc \mathit{Skill1}, \neg\bigcirc \mathit{Skill2}) \\
\lambda_{-1,-2} &= \mathrm{P}(\triangledown\, \mathit{Task1\text{-}obs} | \neg\bigcirc \mathit{Skill1}, \neg\bigcirc \mathit{Skill2})
\end{aligned}$$

where $\neg\bigcirc \mathit{Skilli}$ represents the absence of *Skill i*.

When we are making inferences about a single learner using the techniques of Part I, we generally do not worry about the parametrization. In this case, our best estimates for the parameters “drop down” into the model and give us the values in the conditional probability tables in an ordinary Bayes network. Only when we want to account for uncertainty about the parameters do we need to worry about the distribution of the parameters. (For example, we may want to gauge the impact of this uncertainty on inferences about individuals, to see if we need to collect more data to estimate them more precisely.)

The directed hypergraph representation allows us to annotate each distribution with the parametric form that it takes. In particular, the table icons seen in Figs. 8.2 and 8.3 represent generic conditional multinomial distributions (Sect. 8.3). We can use distinct icons to indicate particular structures. For a noisy-or distribution (Sect. 8.4), we would use an OR-gate icon. For a conjunctive distribution (Sect. 8.5), we would use an AND-gate icon. For a compensatory distribution (Sect. 8.5), we use a plus sign.

As we go forward, we will need to discuss both distributions over variables in our models (like *Skill1* and *Task1-obs*) and distributions over parameters of those distributions (like π_1 and $\lambda_{1,2}$). We will call distributions over parameters, *laws*[1] and reserve the word *distribution* for distributions over variables. Naturally, *laws* will have *hyperparameters*. If we choose to give the hyperparameters distributions (instead of fixing their values) we will call these *laws* as well.

Similarly, we will reserve the term *variable* for a quantity that refers to a particular learner: *observable outcome variables*, for variables that describe the outcomes from scoring responses to presented tasks; *proficiency variables* for variables that describe knowledge, skills, and abilities of the learner; and *demographic variables* for variables that describe fixed characteristics of the learner. We will use the term *parameter* to refer to quantities that are constant across all learners: *population parameters* associated with the proficiency model and *link parameters* associated with the evidence model for a specific task. (In the item response theory (IRT) literature, the latent variable θ is called a person parameter. In our usage, this would be a latent variable associated with the learner, not a parameter. Nothing different conceptually—just terminology to help keep straight what is happening at different levels in the model.)

8.2 Hyper-Markov Laws

Just as the graphical model entails certain probability conditions on the variables, we need to be able to make independence assumptions about the parameters of the model. Spiegelhalter and Lauritzen (1990) define two different types of independence.

Definition. **Local Parameter Independence.** *If parameters within a single probability table are independent, then the model is said to have local parameter independence.*

For example, if $\lambda_{1,2}, \lambda_{1,-2}, \lambda_{-1,2}$, and $\lambda_{-1,-2}$ in Fig. 8.3 are all independent, then we have local parameter independence.

Definition. **Global parameter Independence.** *If parameters for different probability tables are independent, then the model is said to have global parameter independence.*

[1] Steffen Lauritzen (private communication) suggested this language to distinguish the various kinds of distributions in a complex model.

For example, in the graph for Fig. 8.3 under global parameter independence, π_1 and π_2 are independent.

***Definition.* Hyper-Markov Law.** *Let $\mathcal{G}$ be a graphical model and $\mathcal{P}$ be a set of parameters for that model. A distribution for $\mathcal{P}$ which has both the local and global parameter independence property is called a* Hyper-Markov Law*.*

Both local and global parameter independence are assumptions of convenience. They allow us to prove certain properties of the models we learn from data. Even if local and global parameter independence hold a priori, though, they may not hold a posteriori. If any of the variables have missing data for one or more cases, global parameter independence will not hold in the posterior model. We can get a posteriori local parameter dependence even without missing data (York 1992).

Global parameter independence also will not hold if there is some common cause for the variables that is not modeled. Example 8.1 illustrates this situation.

Example 8.1 (Breakdown of Global Parameter Independence). *Suppose the population for our latent skill model was from a mixture of grades. Figure 8.4 illustrates this model. As both Skill 1 and Skill 2 are more likely to be acquired with advancing instruction, π_1 and π_2 likely to be correlated.*

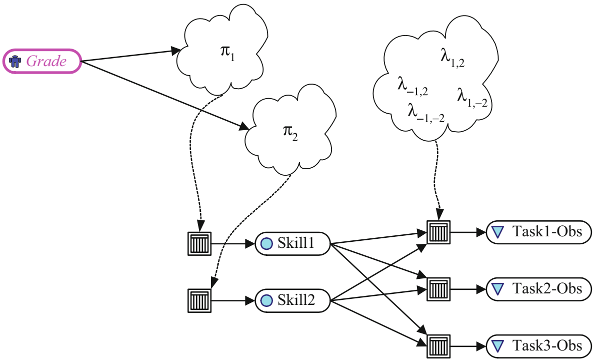


Fig. 8.4 Introducing the demographic variable Grade breaks global parameter independence

Reprinted from Almond et al. (2006a) with permission from ETS.

Adding explicit dependence on grade to our model is one way to deal with this type of correlation. In point of fact, we could always add the parameters directly to the model; the "layers" are merely a convenient way of simplifying the distribution under certain circumstances.

Another common reason for breakdown of global parameter independence is that our probability assessments might share a common source of information. Suppose that we generate a collection of items for a test from the same task model. Because they are all similar items, we may use a common prior distribution for the parameters of the evidence model for that task. However, there should really be some dependence among those parameters because they are all based on the same assessment. Ignoring this dependence will cause us to understate our uncertainty about the final conclusions of the model.

While the global parameter independence assumption breaks down easily, the local parameter independence assumption is suspect from the start. We can often place meaningful constraints on the parameters of the table.

Example 8.2 (Breakdown of Local Parameter Independence). *Suppose that we believe that the probability that a student will get an item right in the model of Fig. 8.3 is strictly increasing in both Skill 1 and Skill 2. It is almost certainly true that $\lambda_{1,2} \geq \lambda_{1,-2} \geq \lambda_{-1,-2}$. This violates local parameter independence.*

We could reparameterize the table for *Task 1-Obs* given *Skill 1* and *Skill 2* as:

$$\begin{aligned}
\lambda_{-1,-2} &= \phi_0 \ ; \\
\lambda_{1,-2} &= \phi_0 + (1 - \lambda_{-1,-2}) * \phi_1 \ ; \\
\lambda_{1,2} &= \phi_0 + (1 - \lambda_{-1,-2}) * \phi_1 + (1 - \lambda_{1,-2}) * \phi_2 \ .
\end{aligned}$$

Here the ϕ's can be modeled as independent variables on [0,1] but the λ's increase in the way we expect.

Despite these problems, global and local parameter independence play a major part in the theory of learning graphical models. In particular, they are used in building the natural conjugate distribution for the Bayesian network, the hyper-Dirichlet distribution.

8.3 The Conditional Multinomial—Hyper-Dirichlet Family

In order to completely specify our parameterized model, we now need a set of prior laws for our parameters. When trying to build priors it is often helpful to look at the conjugate family (Sect. 3.5.3), if one is available. This section shows how to build the conjugate family for Bayesian networks. One of the most studied conjugate families, the beta-binomial family, is the starting point. The Dirichlet-multinomial family is a natural generalization of it. This will lead naturally to the hyper-Dirichlet—conditional multinomial family.

8.3.1 Beta-Binomial Family

We already explored the beta-binomial family in Example 3.14. This section reviews the key ideas.

Let Y be a binomial random variable, i.e., it has the following distribution:

$$p(y|\theta, n) = \begin{cases} \binom{n}{y}\theta^{y}(1-\theta)^{n-y} & \text{for } y = 0, \ldots, n \\ 0 & \text{otherwise.} \end{cases} \tag{8.1}$$

Suppose that n is known and θ is unknown, and we choose the beta prior law for θ, that is,

$$f(\theta|a, b) \propto \theta^{a-1}(1-\theta)^{b-1} \; . \tag{8.2}$$

After Y is observed, applying Bayes theorem produces the posterior distribution:

$$f(\theta|a, b, y, n) \propto \theta^{a+y-1}(1-\theta)^{b+n-y-1} \; . \tag{8.3}$$

Thus, the posterior is also a beta distribution. Therefore, the beta distribution is the natural conjugate of the binomial distribution.

For certain parameter values, the beta distribution has interesting interpretations. The Beta$(1,1)$ distribution is uniform between 0 and 1. Beta$(.5,.5)$ is uniform on the logistic scale. This prior is generated by applying Jeffrey's rule to the binomial likelihood to produce a prior that is invariant under reparameterization, in terms of the amount of information it provides at different values of the variable (Berger 1985). Finally, although Beta$(0,0)$ is not a true probability distribution and is therefore an "improper" prior, the posterior we get when we combine it with a binomial likelihood is a probability distribution as long as we have observed at least one instance of both the event and its converse. The posterior mode using this prior will correspond to the maximum likelihood estimate. All three of these beta distributions have been put forward as noninformative priors for the beta distribution, which we might use if our information about θ were equivalent to no observations at all.

We can interpret the a and b hyperparameters of the beta law as the number of positive and negative cases the expert has seen, i.e., the posterior distribution after seeing a positive and b negative cases. This makes $a+b$ the "effective sample size" of our prior distribution. In particular, it is the size of a sample whose weight is equal to the prior in Bayes theorem. Often a good way to approach experts about a beta prior is to ask for a mean, p^*, and an effective sample size, n^*. This expresses their best guess about θ and their degree of confidence in it. The parameters of the corresponding beta distribution are then $a = p^*n^*$ and $b = (1-p^*)n^*$.

8.3.2 Dirichlet-Multinomial Family

Just as the multinomial distribution is a natural generalization of the binomial distribution, the *Dirichlet law* is a natural generalization of the beta law.

Let $\mathbf{Y}$ be a multinomial random variable that can take on one of K categories. Let Y_k be number of observations that fall in category k in n experiments. The multinomial likelihood is then:

$$p(\mathbf{y}|\boldsymbol{\theta}, n) = \begin{cases} \binom{n}{y_1 \cdots y_K} \theta_1^{y_1} \cdot \cdots \cdot \theta_K^{y_K} & \text{for } y_k \in \{0, \ldots, n\} \text{ and } \sum y_k = n \\ 0 & \text{otherwise.} \end{cases} \tag{8.4}$$

The natural generalization of beta law for $\boldsymbol{\theta}$ is the *Dirichlet law:*

$$f(\boldsymbol{\theta}|\mathbf{a}) \propto \theta_1^{a_1 - 1} \cdot \cdots \cdot \theta_K^{a_K - 1} \ , \tag{8.5}$$

where $\theta_1 + \cdots + \theta_K = 1$. The Dirichlet law is the natural conjugate for the multinomial distribution. The Dirichlet posterior given data $\mathbf{y}$ is Dirichlet$(a_1 + y_1, \ldots, a_K + y_K)$. The beta distribution is a special case of the Dirichlet when $K = 2$.

Setting all $a_k = 0$ produces an improper prior similar to the Beta$(0, 0)$ prior, where the observed proportions will be both maximum likelihood estimates and posterior modes. Applying Jeffrey's rule to produce a noninformative prior that is invariant with respect to variable transformations yields a Dirichlet distribution with all parameters set to $1/2$.

As with the beta law, we can interpret the sum of the parameters of the Dirichlet law as the "effective sample size." Again we can elicit Dirichlet parameters by eliciting $\boldsymbol{\theta}^*$ and n^* (effective sample size of the expert's opinion). Then set $a_k = n^* \theta_k^*$.

8.3.3 The Hyper-Dirichlet Law

Suppose we have a Bayesian network, and a sample of individuals for whom we have observed all the variables in our Bayes Net. This is unrealistic in many of the examples discussed in the previous chapters as most of them contain latent variables. Chapter 9 will discuss methods for getting around that problem. Still, an understanding of how the hyper-Dirichlet law behaves in the complete data case will be helpful in designing tools to deal with the missing data that latent variables effectively constitute.

Next assume that the global parameter independence assumption holds. This allows us to build the prior law one probability table at a time. In the Bayes net, we have two types of probability tables we must parameterize:

Unconditional tables; i.e., variables with no parents. The data for this kind of table will be multinomial, so the Dirichlet law is the natural conjugate.

Conditional tables, that is, variables with one or more parents. In this case the data will be a multinomial distribution for each configuration of the parent variables. We call this distribution the *conditional multinomial distribution* (sometimes it is called the *product multinomial*). By the local parameter independence assumption, the parameters of these multinomials are independent. Thus, the natural conjugate prior is a collection of Dirichlet priors.

Constructing Dirichlet prior laws for every table in this way produces a *hyper-Dirichlet law.* Spiegelhalter and Lauritzen (1990) show that under the global and local parameter independence assumptions, the posterior law is also a hyper-Dirichlet law (in which the global and local parameter independence assumptions continue to hold). Thus, the hyper-Dirichlet law is the natural conjugate of the Bayes net. Furthermore, the hyper-Dirichlet forms a natural noninformative prior for the Bayes net. This property is exploited in many algorithms for discovering models from data.

Although Dirichlet priors are generally straightforward to elicit, for example by the mean and effective sample size method, in practice the work is tedious for a large size network, even for experts who are comfortable with the mathematics. The hyper-Dirichlet prior has as many free parameters as their were in the original network–every probability in every table. All of them must be elicited.

The large number of parameters also means that without strong priors, a large amount of data may be necessary to reliably learn the parameters of the network. This is especially problematic as some configurations of parent variables might be quite rare. For example, consider a distribution with two parent variables *Skill 1* and *Skill 2*. Suppose each have five states and are moderately correlated in the population. Individuals who are `Very High` on *Skill 1* and `Very Low` on *Skill 2* are likely to be very rare in the population at large. Consequently, we are unlikely to do much better than our prior estimates for the probabilities in this row.

We have two ways to get around these difficulties. First, tying together several rows in the table that we believe have the same values (Almond 1995) allows us to borrow strength for seldom-observed rows. Second, parametric models can describe the relationships between the parent and child variables with just a few parameters. The remainder of this chapter concentrates on developing parametric models.

In building Bayes nets, we make the choice of what distribution to use on a table by table basis. Consequently, we tend to use the term *hyper-Dirichlet* distribution[2] for just one table, i.e., for the collection of independent Dirichlets for each table. We use a "table" icon in the hypergraph to represent hyper-Dirichlet distributions (Fig. 8.3).

[2] Technically, it is a conditional-multinomial distribution with a hyper-Dirichlet law, but call it a hyper-Dirichlet distribution for short.

8.4 Noisy-OR and Noisy-AND Models

To reduce the number of parameters in Bayes nets, many authors have looked at a class of models known as noisy-OR, or disjunctive models (Pearl 1988; Díez 1993; Srinivas 1993). Noisy-OR-type models have readily interpretable parameters. They also separate the influences of the parent variables, allowing factorizations of the probability distributions that can be exploited for efficient computation. But because the models suggested in an educational context are more often conjunctive (Junker and Sijtsma 2001; Rupp et al. 2010), we will start with the noisy-AND model and return to the noisy-OR later.

Consider a task that requires mastery of two skills for correct performance. If "*Skill 1* is mastered" and "*Skill 2* is mastered" then the "Response is correct" should be true, otherwise it should be false. This is an *conjunctive model* as both skills are necessary to solve the item. The following truth table represents the conjunctive model:

Conditions		Observed outcome	
Skill 1	*Skill 2*	Right	Wrong
`Yes`	`Yes`	1	0
`Yes`	`No`	0	1
`No`	`Yes`	0	1
`No`	`No`	0	1

A distribution with this kind of truth table is known as an AND-gate in the engineering world and is represented with the symbol, ∩ Consequently, Fig. 8.5 represents the conjunctive model.

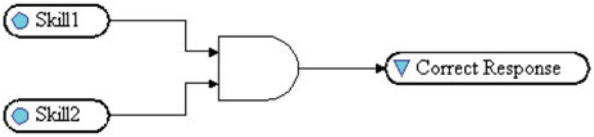


Fig. 8.5 A conjunctive model, with no "noise"
Reprinted from Yan et al. (2004) with permission from ETS.

This deterministic model is not realistic in educational testing. A student who has not mastered one of the required skills may be able to guess the solution to a problem, or solve it via a different mechanism than the one modeled, giving a false-positive result. Let π_- be the probability that a learner without the skills gets the item correct in some other way:

Conditions		Observed outcome	
Skill 1	*Skill 2*	Right	Wrong
`Yes`	`Yes`	1	0
`Yes`	`No`	π_-	$1-\pi_-$
`No`	`Yes`	π_-	$1-\pi_-$
`No`	`No`	π_-	$1-\pi_-$

People who have the skills will occasionally get the item wrong anyway, or slip (Tatsuoka 1983), whether through failure to apply the skills correctly, failure to recognize the correct solution path, or carelessness. In this case, we get a false-negative result. Let π_+ be the probability that the learners who have the requisite skill get the item correct, so $1-\pi_+$ is the probability of a slip. We then have the model:

Conditions		Observed outcome	
Skill 1	*Skill 2*	Right	Wrong
`Yes`	`Yes`	π_+	$1-\pi_+$
`Yes`	`No`	π_-	$1-\pi_-$
`No`	`Yes`	π_-	$1-\pi_-$
`No`	`No`	π_-	$1-\pi_-$

With just two additional parameters we have made a more realistic model. We could simplify things further by positing $1-\pi_+ = \pi_-$; that is, the probability of a false negative is the same as the probability of a false positive. Junker and Sijtsma (2001) call this structure a DINA (Deterministic Input Noisy AND) model. Figure 8.6 portrays it by adding a probabilistic inversion gate to the conjunctive model (Fig. 8.5).

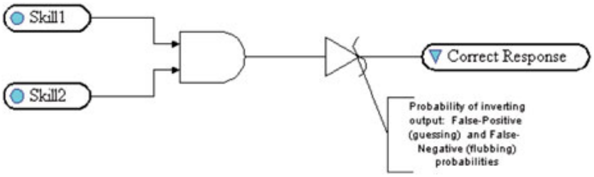

Skill1
Skill2
Correct Response
Probability of inverting output: False-Positive (guessing) and False-Negative (flubbing) probabilities

Fig. 8.6 A conjunctive model with noisy output (DINA)
Reprinted from Yan et al. (2004) with permission from ETS.

The classic approach to building a noisy-logic model is to look at inversions of the inputs. To solve the example problem, the student must either have mastered *Skill 1*, or find a way to work around that lack of mastery. We call the probability of finding that skill workaround r_1. Similarly, let r_2 be the probability of finding a workaround for *Skill 2*. Then, we have:

Conditions		Observed outcome	
Skill 1	*Skill 2*	Right	Wrong
Yes	Yes	1	0
Yes	No	r_2	$1 - r_2$
No	Yes	r_1	$1 - r_1$
No	No	$r_1 r_2$	$1 - r_1 r_2$

Figure 8.7 shows this model. Junker and Sijtsma (2001) call this model NIDA (Noisy Input Deterministic AND). Note that each of the inputs is a combination of two factors. A person can solve the problem if the person has *Skill 1* OR can find a workaround for *Skill 1* AND the person has *Skill 2* OR can find a workaround for *Skill 2*.

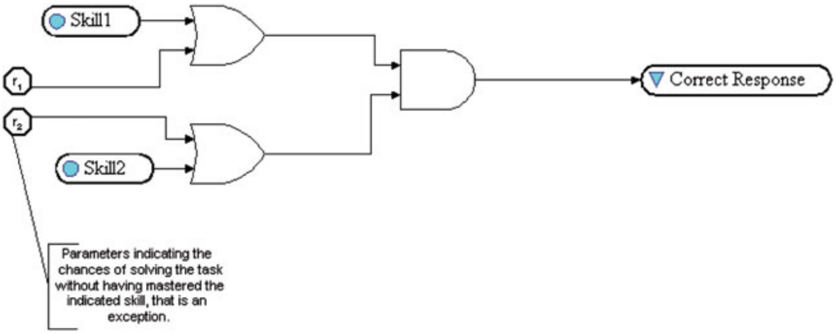

Skill1
r_1
r_2
Skill2
Correct Response
Parameters indicating the chances of solving the task without having mastered the indicated skill, that is an exception.

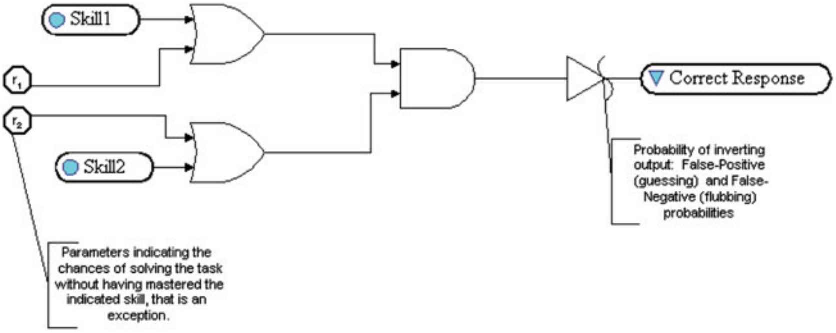
Fig. 8.7 A conjunctive model with noisy inputs (NIDA)
Reprinted from Yan et al. (2004) with permission from ETS.

To extend this to an arbitrary number of parents, let **S** represent the set of skills required for a particular observable outcome variable. Let $S_k = 1$ if the learner has mastered the k^{th} skill to the necessary level to solve the task, and let $S_k = 0$ otherwise. Let T represented the observed outcome from the task. Then, the distribution for the outcome variable is:

$$\mathrm{P}(T = \text{Right}|\mathbf{S}) = \prod_{S_k \in \mathbf{S}} r_k^{1-S_k} \ . \tag{8.6}$$

We can put the two different types of “noise” together to make a full noisy-AND distribution (Fig. 8.8). To reduce the number of parameters for this model, we eliminate the false-positive parameter, π_-, i.e., we set it to 0. The false-positive parameter and the skill workaround parameters, r_k, are measuring the same thing anyway. The final probability model is then:

Conditions		Observed outcome	
Skill 1	*Skill 2*	Right	Wrong
`Yes`	`Yes`	π_+	$1 - \pi_+$
`Yes`	`No`	$\pi_+ r_2$	$1 - \pi_+ r_2$
`No`	`Yes`	$\pi_+ r_1$	$1 - \pi_+ r_1$
`No`	`No`	$\pi_+ r_1 r_2$	$1 - \pi_+ r_1 r_2$



Probability of inverting output: False-Positive (guessing) and False-Negative (flubbing) probabilities

Parameters indicating the chances of solving the task without having mastered the indicated skill, that is an exception.

Fig. 8.8 A noisy conjunctive model, with noisy inputs and outputs Reprinted from Yan et al. (2004) with permission from ETS.

The noisy-AND model as developed here assumes that all of the proficiency variables are binary. If the variables have an arbitrary number of ordered states, we can extend the noisy-and model to a noisy-min model.

Similarly, if only one of the k skills is needed to successfully solve the problem, we can use a noisy-OR, or *disjunctive*, model instead. It is a straightforward translation of the noisy-AND model described above. For instance, the probability table for a noisy-OR with true positive and false positive parameters π_+ and π_- is

Conditions		Observed outcome	
Skill 1	*Skill 2*	Right	Wrong
`Yes`	`Yes`	π_+	$1 - \pi_+$
`Yes`	`No`	π_+	$1 - \pi_+$
`No`	`Yes`	π_+	$1 - \pi_+$
`No`	`No`	π_-	$1 - \pi_-$

In certain application areas, such as failure diagnosis, noisy-OR models are much more common than noisy-AND models. The noisy-OR model generalizes to a noisy-MAX model.

Note that the number of parameters in these models are linear in the number of input variables, not exponential! For a conditional multinomial distribution with k binary parent variables, we must specify 2^k parameters.

However, for a noisy-AND model, we need only $k + 1$ parameters, the r_k's and π_+. Easier to elicit, easier to estimate, and often better suited to the real-world problem at hand.

8.4.1 Separable Influence

Pearl (1988) develops a model for the noisy-OR case, but does not parameterize it with true positive and false positive probabilities. In his model, q_i is the probability that S_i behaves like it is absent even though it is present. Pearl's noisy-OR model looks like this for two skills:

Conditions		Task	
Skill 1	*Skill 2*	Right	Wrong
`Yes`	`Yes`	$1 - q_1q_2$	q_1q_2
`Yes`	`No`	$1 - q_1$	q_1
`No`	`Yes`	$1 - q_2$	q_2
`No`	`No`	0	1

This generalizes to k skills as follows:

$$\begin{aligned} \mathrm{P}(T = \mathit{Wrong}|\mathbf{S}) &= \prod_k q_k^{S_k} \ , \\ \mathrm{P}(T = \mathit{Right}|\mathbf{S}) &= 1 - \prod_k q_k^{S_k} \end{aligned} \tag{8.7}$$

again with $S_k = 1$ for `Yes` and $= 0$ for `No`.

Notice how Eq. 8.7 factors according to the input variables. Thus, the hypergraph for this model fragment effectively simplifies to Fig. 8.9b instead of Fig. 8.9a—but only if the student gets the item wrong. If the learner gets the task right the parents are dependent. This is exactly the competing explanation phenomenon discussed in Chap. 3.

Distributions that can be factored according to the parent variables are said to have *separable influences* (also called "causal independence," a term we avoid because of the dissonance between a technical sense of "causal" and its everyday meaning). The NIDA model (Eq. 8.6) has the separable influences property too, for just when the response is correct. Computation algorithms can exploit this special structure (Li and D'Ambrosio 1994).

8.5 DiBello's Effective Theta Distributions

For the Biomass project (Chap. 14), Lou DiBello (Almond et al. 2001) developed a method based on IRT models to produce a class of reduced parameter distributions for use in the evidence models. The central idea was to map

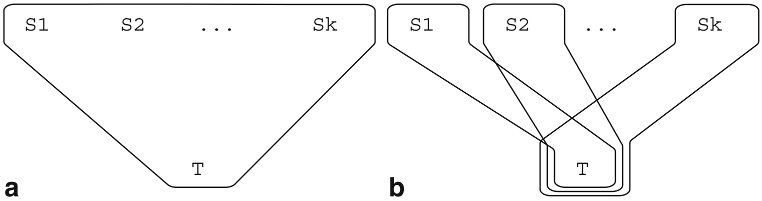


Fig. 8.9 Separable influences. If the conditional probability table for P(T/S_1, $\ldots, S_K$) has the separable influence property, the Graph (a) becomes Graph (b). Reprinted with permission from ETS.

each configuration of skill variables into an *effective theta*—a real number representing the student's propensity to be able to perform tasks of that type. The assumption was that even though the test is multidimensional, the proficiency combination associated with scoring well on any given observable within a task represents a single direction within that multidimensional space. Once we represent our beliefs about the student's state as a real number θ (a distance along that direction), we can press familiar IRT models into service to determine the probabilities for the distribution tables which drive the Bayes nets. This reduces the number of parameters, and furthermore relates the parameters to concepts like difficulty and discrimination that are already familiar to psychometricians. The approach is akin to that of structured or located latent class models (Almond et al. 2001; Formann 1985; von Davier 2008).

The DiBello effective theta method proceeds in three steps:

1. For each input variable, designate a real number to serve as an *effective theta* for the contribution of that skill (Sect. 8.5.1).
2. The inputs each represent a separate dimension. Use a *combination function* to combine them into an effective theta for the task (Sect. 8.5.2). There are a number of combination functions that can be used, to model for example compensatory, conjunctive, and inhibitor relationships.
3. Apply a *link function* to calculate probabilities for the dependent variable from the combined effected theta. DiBello proposed using Samejima's graded response model as the link function, i.e., the DiBello–Samejima model (Sect. 8.5.3).[3] For representing relationships among proficiency variables, for example, we recommend a different method, based on cut points of the normal distribution (Sect. 8.5.4).

A key feature of this class of models is the combination function (Step 2). The choice of this function dictates the type of relationship (e.g., sum for compensatory, min for conjunctive). In typical modeling situations, the experts

[3] Other models could be used to parameterize link functions at this point, such as the generalized partial credit model (Muraki 1992) and the model for nominal response categories (Bock 1972).

provide not only which variables are parents of a given variable but also what the type of relationship is. They also indicate the relative importance of each variable in the combination. Section 8.6 describes the technique we use to elicit the information from the experts. Note that the generalized diagnostic model (von Davier 2008) also uses combination and link functions, although it does not use the name "combination function."

8.5.1 Mapping Parent Skills to θ Space

For the moment, we assume that each parent variable lies in its own separate dimension. Each category for a parent variable represents an interval along this dimension. It is common in IRT to assume that proficiency follows a standard normal distribution. We will build effective theta values from standard normal distributions.

Suppose that one of the parent variables in the relationship is *Skill 1*. Suppose further that it has three levels: `Low`, `Medium`, and `High`. Positing for the moment that the skill labels reflect an equal-probabilities partitioning of the distribution, we can break the area under the normal curve into three segments (Fig. 8.10).

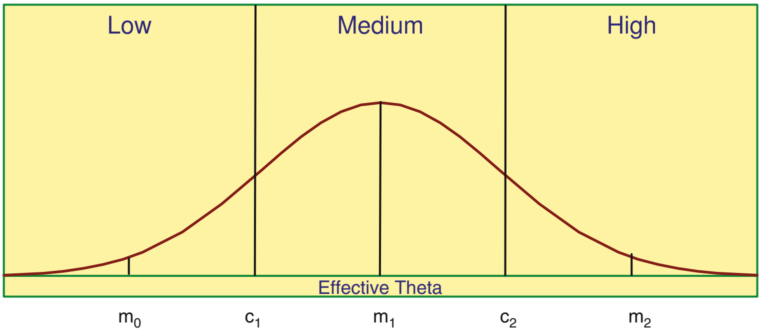


Fig. 8.10 Midpoints of intervals on the normal distribution Reprinted from Yan et al. (2004) with permission from ETS.

This is fairly easy to do in practice. Assume that the parent variable has K-ordered levels, indexed $0, \ldots, K-1$. Let m_k be the effective theta value associated with Level k. Let $p_k = (2 * k + 1)/2K$ be the probability up to and including this level. Then, $m_k = \Phi^{-1}(p_k)$, where $\Phi(\cdot)$ is the cumulative normal distribution function. For two levels, for example, the values are $-.67$ and $+.67$. For three levels, the values are $-.97$, 0, and $+.97$.

When an observable has just one proficiency parent, these are the effective theta values we use to model conditional probabilities for it. Section 8.5.3

shows how this is done. First, we discuss how to combine effective thetas across proficiencies when there are multiple parents.

8.5.2 Combining Input Skills

Each mapping from a parent variable to its effective theta value is done independently. That is, each knowledge, skill, or ability represented by a parent variable is assumed to have a separate dimension. The task observation (or dependent proficiency variable) also has its own dimension. As in regression, the individual parents' effective thetas are independent variables and the response variable is a dependent variable. The idea is to make a projection from the space of the parent dimensions to a point on the child dimension. This projection is the effective theta, synthesized accordingly across parents, as it applies to this particular child. The value will then be used in the Samejima graded response IRT model to produce probabilities for the response categories of the child.

The easiest way to do this is to use a *combination function*, $g(\cdot)$, that produces a single theta value from a list of others, and the easiest combination function to understand, and therefore the one we will start with, is the sum. When effective thetas are summed, having more of one parent skill can compensate for having less of another. For this reason, we call the distribution we build using this combination rule the *compensatory distribution.* If $\theta_1, \ldots, \theta_K$ are the effective thetas for the parent dimensions, then the effective theta for the child dimension is

$$\tilde{\theta} = g(\theta_1, \ldots, \theta_K) = \sum_{k=1}^{K} \frac{\alpha_k}{\sqrt{K}} \theta_k - \beta \; . \tag{8.8}$$

This model has one slope parameter, α_k, for each parent variable and one intercept parameter. Following IRT terminology, we sometimes call these the discrimination and difficulty parameters; in fact, β is given a negative sign so that it will be interpreted as a difficulty (higher values of β mean that the problem is "harder"—the probability of getting it right is lower). The factor $1/\sqrt{K}$ is a variance stabilization term. If we assume that the variance of each of the θ_k's is 1 (unit normal assumption), then the variance of $\tilde{\theta}$ will be $\sum \alpha_k^2 / K$. In other words, the variance of the effective theta will not grow with the number of parent variables. Table 8.1 gives a simple example for two choices of the αs and βs.

Equation 8.8 does not include a subscript to indicate which probability table we are defining. Technically, there should be another subscript on all the parameters in that equation to indicate the child variable. We have suppressed that subscript to simplify the exposition. However, it is worth noting that we are assigning a separate set of slope and difficulty parameters to each observable outcome variable. Sometimes the values of the slope parameters are constrained to be the same across multiple observable variables. In particular,

Table 8.1 Effective thetas for a compensatory combination function

Skill 1	θ_1	*Skill 2*	θ_2	Effective thetas	
				$\alpha_1 = \alpha_2 = \beta_1 = 1$	$\alpha_1 = 1$, $\alpha_2 = 0.5$,$\beta_1 = 0$
`High`	$+0.97$	`High`	$+0.97$	$+0.37$	$+1.03$
`High`	$+0.97$	`Medium`	0.00	-0.32	$+0.68$
`High`	$+0.97$	`Low`	-0.97	-1.00	$+0.34$
`Medium`	0.00	`High`	$+0.97$	-0.32	$+0.34$
`Medium`	0.00	`Medium`	0.00	-1.00	0.00
`Medium`	0.00	`Low`	-0.97	-1.68	-0.34
`Low`	-0.97	`High`	$+0.97$	-1.00	-0.34
`Low`	-0.97	`Medium`	0.00	-1.68	-0.68
`Low`	-0.97	`Low`	-0.97	-2.37	-1.03

the Rasch IRT model uses a single slope parameter for all items. A constrained model is possible, but if the slope parameter is estimated rather than fixed, the constrained model does not have the global parameter independence property.

The compensatory distribution produces a considerable savings in the number of parameters we must elicit from the experts or learn from data. The total number of parameters is $K + 1$, where K is the number of parent variables. Contrast this with the hyper-Dirichlet model which requires 2^K parameters when all variables are binary, and more if any variable is not binary.

We can change the combination function to model other relationships among the parent and child variable. The most common is to replace the sum with a minimization or maximization. Thus, the combination function

$$\tilde{\theta} = g(\theta_1, \ldots, \theta_K) = \left[\min_{k=1}^{K} \alpha_k \theta_k \right] - \beta \ , \tag{8.9}$$

produces a *conjunctive distribution* where all the skills are necessary to solve the problem. Equation 8.9 posits that the examinee will behave with the weakest of the skills as the effective theta level. Similarly, the combination function

$$\tilde{\theta} = g(\theta_1, \ldots, \theta_K) = \left[\max_{k=1}^{K} \alpha_k \theta_k \right] - \beta \ ,$$

produces a *disjunctive distribution* where each parent variable represent an alternative solution path. The examinee will behave with the strongest of the skills as the effective level. Table 8.2 gives examples for the conjunctive and disjunctive combination functions; for simplicity, $\alpha_1 = \alpha_2 = 1$ and $\beta = 0$.

So far, all of the combination functions have been symmetric (or, more properly, the asymmetry has been modeled by different values for the slope parameters). One interesting type of asymmetric combination function is the *inhibitor distribution* (or more optimistically, the *enabler distribution*). Here, we assume that a minimal level of the first skill is required as a prerequisite,

Table 8.2 Effective thetas for the conjunctive and disjunctive combination functions

Skill 1	θ_1	*Skill 2*	θ_2	Effective thetas	
				Conjunctive	Disjunctive
`High`	$+0.97$	`High`	$+0.97$	$+0.97$	$+0.97$
`High`	$+0.97$	`Medium`	0.00	0.00	$+0.97$
`High`	$+0.97$	`Low`	-0.97	-0.97	$+0.97$
`Medium`	0.00	`High`	$+0.97$	0.00	$+0.97$
`Medium`	0.00	`Medium`	0.00	0.00	0.00
`Medium`	0.00	`Low`	-0.97	0.00	0.00
`Low`	-0.97	`High`	$+0.97$	-0.97	$+0.97$
`Low`	-0.97	`Medium`	0.00	-0.97	0.00
`Low`	-0.97	`Low`	-0.97	-0.97	-0.97

but once the examinee have reached the minimal level in the first skill, the second skill takes over.

Example 8.3 (Story Problem). *A typical example is a "story problem" in a math assessment. Usually the skills we wish to assess are related to mathematics, i.e., the ability to translate the verbal representation of the problem into the symbolic one, then solve the symbolic problem. However, in order to have any hope of solving the problem the examinee needs a minimal competency in the language of the test. This is an inhibitor relationship.*

Assume that only two skills are necessary, and that the first one is the inhibitor with the prerequisite that this skill is at least at level r. Also, let θ_{km} be the effective theta value associated with the mth level of *Skill k*. Then we can write the inhibitor relationship as follows:

$$\tilde{\theta} = g(\theta_1, \theta_2) = \begin{cases} \alpha_2\theta_2 - \beta & \text{for } \theta_1 \geq \theta_{1r}, \\ \alpha_2\theta_{2,0} - \beta & \text{otherwise.} \end{cases} \tag{8.10}$$

The numerical example in Table 8.3 shows two skills, both with three levels, and θ_1 acting as an inhibitor: If $\theta_1 <$ `Medium`, the combined effective theta is assigned to the lowest level of θ_2. If $\theta_1 \geq$ `Medium`, the combined effective theta is that of θ_2. For simplicity, $\alpha_2 = 1$ and $\beta = 0$. Contrast this with the conjunctive combination function in Table 8.2

The inhibitor distribution has an interesting evidentiary interpretation. Return to the English "story problem" example (Example 8.3). If the examinee's English proficiency is above the threshold, then the story problem can provide evidence for the mathematical skills it is designed to measure. On the other hand, if the English proficiency is below the threshold, the distribution produces no evidence for the mathematical skills; the lack of a prerequisite inhibits any evidence from flowing from the observable to the proficiency (see Exercise 8.10).

Table 8.3 Effective thetas for inhibitor combination functions

Skill 1	*Skill 2*	θ_2	Effective thetas
High	High	+0.97	+0.97
High	Medium	0.00	0.00
High	Low	−0.97	−0.97
Medium	High	+0.97	+0.97
Medium	Medium	0.00	0.00
Medium	Low	−0.97	−0.97
Low	High	+0.97	−0.97
Low	Medium	0.00	−0.97
Low	Low	−0.97	−0.97

It is possible to make more complex combination functions by mixing the combination functions given above. For example, a mix of conjunctive and disjunctive combination functions could model a task that admits two solution paths, one using *Skills 1* and *2* and the other using *Skills 3* and *4*. In fact, mixing the inhibitor effect with compensatory and conjunctive combination functions is quite common.

Distributions can also be chained together by introducing dummy latent variables as stand-ins for combinations of skills. In the example above, new variables would be defined for having the conjunction of *Skills 1* and *2*, say *Skill 1&2*, and similarly defining *Skill 3&4*. Then a disjunctive combination would follow, of having *Skill 1&2* or *Skill 3&4*. Even though there are more variables, computation is simplified because the conditional probability matrices are smaller. On the other hand, there is a loss of precision with this approach. An unrestricted conditional probability matrix can have different probabilities for each possible combination of parents, whereas the stage-wise combinations collapse categories. Whether this is a good trade-off depends on whether the collapsed parent combinations are substantively different. There also may be some other issues noted in Sect. 8.5.4.

8.5.3 Samejima's Graded Response Model

Applying the combination function to the effective thetas for each of the parent variable produces one effective theta for each row of the conditional probability table (CPT). Completing the CPT requires turning that effective theta into a set of conditional probabilities, one for each possible state of the child variable. This is the role of the link function. This section explores a class of link functions based on the logistic function (commonly used in IRT models). Now that we have projected the examinee's skill state (i.e., on the parent variables), we will calculate probabilities with, the graded response model of Samejima (1969). Section 8.5.4 describes an alternative link function based on the normal distribution.

The most common IRT models posit a single continuous skill variable, θ, and binary response variables. The relationship between the observation and the skill variable is a logistic regression. Various parameterizations are found in the literature. The simplest is the one parameter logistic (1PL) or Rasch model (same function, different conceptual underpinnings):

$$\mathrm{P}(X=1|\theta) = \mathrm{logit}^{-1}(\theta - d) = \frac{\exp(D(\theta-d))}{1+\exp(D(\theta-d))}, \tag{8.11}$$

where b is an item difficulty parameter and X is a binary observed outcome, 1 for correct and 0 for incorrect. The constant $D = 1.7$ is sometimes used to scale the logistic function so that it looks like a normal CDF (Sect. 8.5.4). The 2PL additionally has a slope or discrimination parameter for each item, so $\mathrm{logit}^{-1}\, Da(\theta - d)$.

Samejima's graded response model extends the this model to an observable X that can take one of the ordered values $x_0 \prec \cdots \prec x_{M-1}$. It is usually developed from the 2PL, but we will build from the 1PL because the 2PL's discrimination parameter is redundant with the slope parameters in the effective theta combination function. For $m = 1, \ldots, M-1$, we first define cumulative conditional probabilities for the response categories:

$$\mathrm{P}(X \geq x_m|\theta) = P^*_m(\theta) = \mathrm{logit}^{-1}\, D(\theta - d_m), \tag{8.12}$$

where d_m is a category difficulty parameter. There is always one fewer cumulative probability curve than the number of possible outcomes, as $\mathrm{P}(X \geq x_0) = 1$. The category response probabilities $\mathrm{P}(X = x_m|\theta)$ can be calculated from the differences of the cumulative probabilities given by Equation 8.12, with $\mathrm{P}(X = x_0) = 1 - \mathrm{P}(X \geq x_1)$. That is,

$$\mathrm{P}(X = x_m|\theta) = P^*_m(\theta) - P^*_{m+1}(\theta). \tag{8.13}$$

Figure 8.11 illustrates response category probabilities for a three-category task, $d_1 = -1$, and $d_2 = +1$. For very low values of θ, the lowest level of response is most likely. As θ increases, probabilities increase for higher-valued responses in an orderly manner. Given the item parameters, a single value of θ specifies the full conditional distribution for all possible responses.

All that remains is to specify the values for d_m. As we have a difficulty parameter, β, additional constraints are needed for the ds in the effective theta combination step. One way to do this is to require $\sum_{m=1}^{M-1} d_m = 0$. For three categories, this means that $d_1 = -d_2$. Or we can set $d_1 = -1$ and $d_{M-1} = 1$, or $d_1 = 0$ and $d_{M-1} = 1$ (as we do in Chap. 15). Furthermore, as the categories of the output variable are ordered with respect to difficulty, the d_m's should be increasing. Other than that we can specify any values we like for these parameters. When there are more than three categories, $d_2 < \ldots < d_{m-1}$ can be estimated, or set at equal spacing, or, when M is large, determined as a function of some smaller number of parameters (e.g., Andrich 1985).

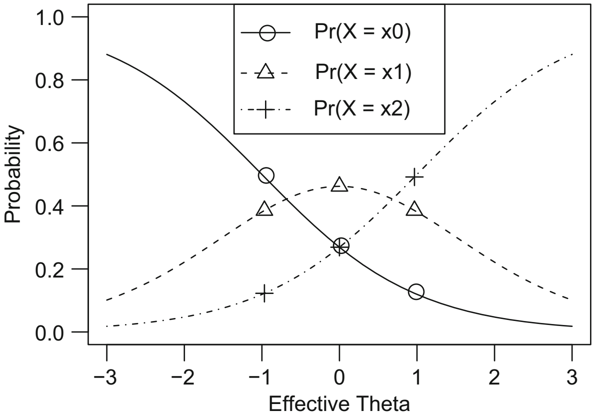


Fig. 8.11 Probabilities for Graded Response model
Reprinted from Almond et al. (2006a) with permission from ETS.

Almond et al. (2001) took them to be equally spaced in the interval $[-1, 1]$. The intercept β controls the overall difficulty. The d parameters control how spread out the probabilities for the various child variable values are, with the slope parameter(s) α_k determining how much varying each of the parents changes the probabilities.

Figure 8.11 illustrates the idea for a simple conditional probability table for a child variable with a single parent, both of which have three levels. The combination function is $\alpha_1\theta_1 - \beta$ (as there is only one parent, it does not matter if it is compensatory or conjunctive), and for simplicity, set $\alpha_1 = 1$ and $\beta = 0$, also let $d_0 = -1$ which forces $d_1 = 1$. Figure 8.11 shows the three graded response curves, and the points are the values of those curves evaluated at the effective thetas. Putting this in tabular form yields the table shown in Fig. 8.4.

Example 8.4. *Compensatory Graded Response Distribution. As a more complete example, consider the CPT for an observable variable Credit which represents three possible scoring levels for a short constructed response task* `Full`*,* `Partial`*, or* `No` *credit. This task taps two measured skills which are labeled Skill* 1 *and Skill* 2 *each of which have three possible levels:* `High`*,* `Medium`*, and* `Low`*. The domain experts have determined that the relationship between these skills is compensatory and that they are equally important, and that the task should be of average difficulty. The combination function is therefore the one given in Eq. 8.8, with $\alpha_1 = \alpha_2 = 1$ and $\beta = 0$. The effective thetas are given in Table 8.1. The graded response link functions are given in Eqs. 8.12 and 8.13. Table 8.5 shows the final conditional probability table.*

Even though we may use equally-spaced values of d_m for elicitation from the experts, we may not want to preserve the equal spacing when we have a lot

Table 8.4 Conditional probability table for simple graded response model

Skill	Effective theta	Full	Partial	None
High	+0.967	0.656	0.278	0.066
Medium	0.000	0.269	0.462	0.269
Low	−0.967	0.066	0.278	0.656

Table 8.5 Compensatory combination function and graded response link function

Skill 1	*Skill 2*	Effective thetas	Full	Partial	No
High	High	+1.03	0.678	0.262	0.060
High	Medium	+0.68	0.541	0.356	0.103
High	Low	+0.34	0.397	0.433	0.171
Medium	High	+0.34	0.397	0.433	0.171
Medium	Medium	0.00	0.269	0.462	0.269
Medium	Low	−0.34	0.171	0.433	0.397
Low	High	−0.34	0.171	0.433	0.397
Low	Medium	−0.68	0.103	0.356	0.541
Low	Low	−1.03	0.060	0.262	0.678

of data. Instead, we could estimate distinct d_m's for each category, free to vary independently as long as they remain in increasing order. In this approach, the model will fit the marginal proportions in each category exactly.

Following the advice of Kadane (1980), we should try to elicit priors in terms of quantities the experts are used to observing. Here, the best parameters to elicit would be the marginal proportions of observed outcome (Kadane et al. 1980 and Chaloner and Duncan 1983 describe applications of this idea for regression models and binomial proportion models, respectively). In a Bayes net, marginal proportions depend on both the conditional probability distributions described above and population distribution for the parent variables (usually proficiency variables). If the population distribution for the parent variables have already been elicited, it should be possible to pick b_j's and $d_{j,m}$'s to match the marginal proportions.

8.5.4 Normal Link Function

The IRT-based link functions shown in the preceding section work fairly well when the child variable is an observable outcome and the parent variable is proficiency variable. This makes sense as IRT models were designed to model the relationship between a latent ability and an item outcome. When the child variable is a proficiency variable, a different link function—one based on the normal distribution—works better (Almond 2010a). This link function is motivated by many users' familiarity with basic regression with normally distributed residuals.

Assume again that the population distribution on the effective theta space for the child variable is normally distributed, and that (before taking into account information from the parent variables) it is equally divided between the categories. We can establish cut points, c_i, between the intervals. This is the solid curve in Fig. 8.12). (Actually, equal divisions are not necessary, Almond 2010a extends the method to use any desired marginal distribution for the child variable. This section will stick to the equal divisions, as they are easier to explain.)

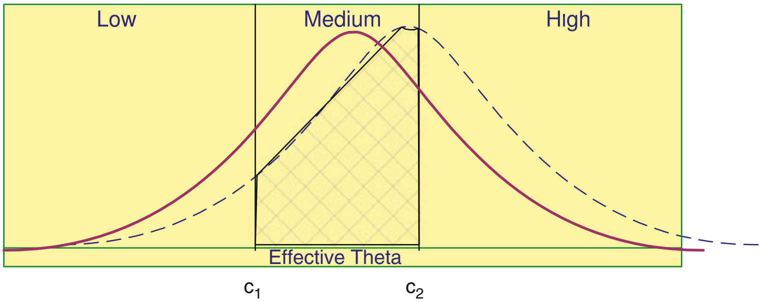


Fig. 8.12 Output translation method. The *solid curve* represents standard normal reference curve for child variable; cut points are established relative to equal probability intervals on this curve. The *dashed curve* represents a displaced curve after taking parent variables into account. Probabilities for child variables are areas between cut points under the *dashed curve*. Reprinted with permission from ETS.

Next, we consider how the parent variables would shift our beliefs about the child. We can think about this as a kind of a regression where the effective theta for the child variable is predicted from the effective theta for the parent variables, so that:

$$\tilde{\theta} = \sum_{k \in \text{parents}} \frac{\alpha_k}{\sqrt{K}} \theta_k - \beta, \tag{8.14}$$

where the θ_ks on the right are the effective thetas of the parent variables and $\tilde{\theta}$ on the left is the combined effective theta of the child. As in the compensatory combination function, the factor of $1/\sqrt{K}$ stabilizes variance of the linear predictor the compensatory distribution. Equation 8.14 gives the mean of the dashed curve in Fig. 8.12. Changing the value of β shifts this curve to the right or the left, and changing the value of the α_k's changes the amount that the curve shifts with the change in the parent variables.

A given individual will not have an effective theta precisely at the mean of this curve, but rather somewhere around it. Let σ be the residual standard deviation. As the effective thetas for the parent variables are scaled to have variance one, $\mathrm{Var}(\tilde{\theta}) = \sigma^2 + \sum_{k=1}^{K} \alpha_k^2/K$. When σ^2 is small compared to the α_k's, then the distributions will be tightly clustered around the value predicted by Eq. 8.14; in other words the parents will have more predictive power. When σ^2 is large, then there will be more uncertainty about the state of the child variable given the parents.

Continuing the analogy to multiple regression, we can define

$$R^2 = \frac{\sum_{k=1}^{K} \alpha_k^2/K}{\sigma^2 + \sum_{k=1}^{K} \alpha_k^2/K}.$$

We need to be careful in interpreting this R^2 value though. In the typical effective-theta application, it is describing the squared correlation between latent variables. Latent variable correlations are considerably higher than the observed variable correlations that experts are used to seeing. An observed variable correlation of 0.5 might correspond to an latent correlation of 0.8 or higher (depending on the reliability of the measures of the latent variables.)

The final probabilities come from the area between the cut points in the prediction (dashed) curve. The probability for the lowest state is the area below c_1. The probability for the second state is the area between c_1 and c_2, and so on up to the highest state which is the area above the highest cut point. Thus, if the child variable is X, then:

$$\mathrm{P}\left(X \geq x_m | \,\mathrm{pa}(X)\right) = \Phi\left(\frac{c_m - \tilde{\theta}}{\sigma}\right) \tag{8.15}$$

where $m > 1$. Naturally, $\mathrm{P}(X \geq x_1) = 1$, so the individual probabilities can be calculated from the differences:

$$\mathrm{P}\left(X = x_m | \,\mathrm{pa}(X)\right) = \Phi\left(\frac{c_{m-1} - \tilde{\theta}}{\sigma}\right) - \Phi\left(\frac{c_m - \tilde{\theta}}{\sigma}\right) \tag{8.16}$$

The value of $\tilde{\theta}$ is given by Eq. 8.14.

There is a strong similarity between Eqs. 8.12 and 8.15. First, the shape of the logistic curve and the normal CDF is very similar; the factor $D = 1.7$ commonly used in IRT models makes the two curves almost the same shape. In particular, the normal link function described here is really the probit function, and the graded response link function described in the previous section is the logistic link function.

There are four differences between the two link functions. The first is the metaphor that motivates the curve: Eq. 8.12 is based on IRT while Eq. 8.15 is based on factor analysis and regression. The second is the role of the parameter σ, the residual standard deviation. In the normal model, this effectively scales the α_k's and β, so that the equivalent values using the graded respond link function would be α_k/σ and β_k/β_0. The residual standard deviation, σ, also scales the constants c_m so that $d_m = c_m/\sigma$. The third difference is that with the normal link function, the c_m's are fixed and σ must be elicited from experts or learned from data. With the graded-response link function, the d_m's must be elicited or learned. Fourth, although the normal and logistic distributions have similar shapes, the normal is more dispersed. If comparable parameters are desired as in some IRT applications, one can rescale the logistic parameters using the approximation $\Psi(1.7z) \approx \Phi(z)$.

Example 8.5. *Correlation between two proficiency variables. Assume that the subject matter expert has suggested modeling Skill 1 as a parent of Skill 2, and that the correlation between the two skills is* 0.8 *(remember, this is a correlation between the two latent variables, so it will be higher than an observed score correlations). Setting $\alpha_1 = 0.8$ and $\sigma = 0.6$ preserves this correlation. Further assume that the expert asserts that Skill 2 is slightly more common in the population (this is equivalent to having lower difficulty, except that "difficulty" is not really appropriate for a proficiency variable). This can be modeled by setting $\beta = -0.5$. As there are three categories, so we need to divide the effective theta range into three equal categories. This gives the values $c_1 = \Phi^{-1}(1/3) = -0.43$ and $c_2 = \Phi^{-1}(2/3) = +0.43$. Table 8.6 gives the conditional probability table.*

Table 8.6 Conditional probability table with normal link function, correlation = 0.8

	Skill 2		
Skill 1	High	Medium	Low
High	0.920	0.078	0.002
Medium	0.546	0.394	0.060
Low	0.120	0.483	0.397

Example 8.6. *Conditional Probability Table built from Path Analysis Results. Assume that the subject matter expert has recommended modeling Skill 3 as the child of Skill 1 and Skill 2, and furthermore, the expert has the results of a path analysis, which shows path coefficients of 0.58 and 0.47 for Skill 1 and Skill 2 to Skill 3, an a residual standard deviation of 0.67. To model this, set $\alpha_1 = 0.58$, $\alpha_2 = 0.47$ and $\sigma = 0.67$. Path analysis centers all variables, so it does not provide us with a source of information for β. Assume that the expert believes that Skill 3 will be less common in the population than Skill 1 and Skill 2. On this basis set $\beta = 0.75$. As path analysis corresponds to the compensatory combination function, Eq. 8.8 is used to calculate the effective thetas. The final conditional probability table is given in Table 8.7.*

Even though the number of parameters in conditional probability tables created in this way grows linearly with the number of parents, the size of

Table 8.7 Conditional probability table for path coefficients 0.58 and 0.47

			Skill 3		
Skill1	*Skill2*	Effective theta	High	Medium	Low
High	High	−0.032	0.245	0.479	0.276
Medium	High	−0.428	0.100	0.401	0.499
Low	High	−0.825	0.030	0.248	0.722
High	Medium	−0.353	0.121	0.425	0.454
Medium	Medium	−0.750	0.039	0.278	0.683
Low	Medium	−1.147	0.009	0.133	0.857
High	Low	−0.675	0.049	0.308	0.642
Medium	Low	−1.072	0.012	0.157	0.831
Low	Low	−1.468	0.002	0.058	0.939

the conditional probability table grows exponentially as the number of parents increases. Therefore, models in which every node has a limited number of parent variables, will be generally more efficient. There are two ways to accomplish this: First, one can try to identify conditional independencies to simplify the model. Almond (2010a) notes that path analyses and factor analyses often produce correlation matrixes, and that zeroes in the inverse of the covariance matrix (Dempster 1972; Whittaker 1990) correspond to conditional independencies. Second, often a latent variable can be added to the Bayesian network to capture additional dependencies in a way that makes can both add insight and make computation more efficient.

It is possible to use the normal link function with the conjunctive, disjunctive and inhibitor combination functions, too. This would allow modelers to combine compensatory, conjunctive, disjunctive, or inhibitor combination functions with logistic or probit link functions. However, because the linear regression paradigm is so strongly associated with the normal link function, the compensatory combination function is almost always used with the normal link function.

8.6 Eliciting Parameters and Laws

Let us shift our focus back to the big picture of where the Bayes net models we use in this book come from. Ideally, it is a combination of expert opinion and data. As all the variables in the proficiency model are latent, we will always need to lean on subject matter experts to define the variables and the relationships between the latent and observable variables. On the other hand, when we get data back from students taking the assessment, we want to revise our models to be more consistent with the data. The following chapters will describe techniques for doing just that.

The evidence-centered design (ECD) process (described briefly in Chap. 2 and in more detail in Part III) is a method for eliciting (and documenting) a model of the assessment from the subject matter experts. Although many of the steps of ECD are aimed at the equally important task of designing tasks to provide the evidence to update the model, all of the core steps required to build the Bayes nets for the proficiency and evidence models are part of the ECD process. In particular, the assessment designer (working with the subject matter expert) needs to perform the following steps:

1. *Define all of the variables in the proficiency and evidence models.* This step can be tricky because of the need to balance the purposes and constraints of the assessment and competing views of the domain held by various groups of experts.
2. *Define the relationships among the variables.* The most important part of this task is specifying which proficiency variables each observed outcome variable depends on, i.e., defining the Q-Matrix for the assessment. These first two steps may be 90 % of the problem of specifying the mathematical structure of the assessment. Relationships among proficiency variables can be learned from correlation matrixes built as part of factor analyses or other structural equation models (Almond 2010a). Once the model structure is set, the remaining steps involve establishing the conditional probability tables, which are much easier to learn from data.
3. *For each variable, choose a distribution type that defines the relationship between it and its parents.* This chapter provides a number of examples, and there are many others. The challenge is translating the expert's description of the relationship among variables (e.g., compensatory or conjunctive) to a mathematical model.
4. *For each distribution, establish the values of any parameters.* In practice, an operational assessment will usually use the mean of the law established in Step 5. In other words, we will estimate the posterior distributions for the parameters that determine conditional probability matrices, then use the means of their posteriors to provide a working approximation of the conditional probabilities in the operational assessment.
5. *For each distribution establish a law (and appropriate hyperparameters) that describe our uncertainty about the parameters.* By this point we are dealing with a level of abstraction in which the experts are likely to feel uncomfortable. However, this step is necessary if the parameters are to be updated from experiential data as described in the later chapters.[4]

[4] As the number of parameters increases, more priors are required. If the same strength of priors is imposed parameter by parameter, the effective influence of the priors is greater on the model as a whole. We think this is generally reasonable for estimation; for a model that is more complicated and probably less stable, it is conservative to require more evidence from the data to move parameter values

As we progress through the steps, the level of abstraction goes up and the comfort level of the subject matter experts goes down. At the first step, the experts are very comfortable identifying skills which are important in a domain. Typically the role of the psychometrician is to keep the experts focused on what can be measured within the resource constraints (particularly time) for the assessment.

Somewhere around Step 3, the tide turns. The experts are comfortable describing the relationships of the variables in qualitative terms and the psychometrician must help them translate this into quantitative terms. The following sections describe some of the techniques we have used for the distributions described in previous sections.

8.6.1 Eliciting Conditional Multinomial and Noisy-AND

The key to eliciting the parameters for the conditional multinomial distribution and hyper-Dirichlet law is to use the observational equivalence property that comes out of the conjugacy. The examples in this section illustrate a practical way to do this. The first step in each case is to ask the subject matter expert about the skills and the anticipated performance of 100 fictitious students. This will give the mean of the law. The second step is to ask the expert about how many student's worth of observation their opinion is worth. This will give the scale factor.

Example 8.7 (Unconditional Multinomial Elicitation). *Suppose that the proficiency model the experts provide has a node for OverallProficiency which has no parents. (This is fairly typical). Suppose further that we are allowing four levels for this variable:* `minimal`, `basic`, `proficient`, *and* `advanced`. *We ask the experts "Of 100 typical students, among those for whom the assessment is to be used, how many will fall in each category?" Suppose that the expert tells us that about* $1/6$ *will be in the lowest category,* $1/3$ *will be in each of the two middle categories, and* $1/6$ *will be advanced. If the expert is fairly confident of those numbers, we might say this is worth a pretest sample size of 100, so the Dirichlet hyperparameters for this distribution would be:* $(16.67, 33.33, 33.33, 16.67)$. *If the expert was not so confident, we might assign it an effective sample size of 10. This would give the following hyperparameters:* $(1.667, 3.333, 3.333, 1.667)$. *The mean has not changed, but the variance is considerably larger. It could change faster as data arrive and we update the parameters accordingly by the methods in discussed in the next chapter.*

A prior distribution with an effective sample size of 100, will have more information than 100 pretest subjects. That is because the variable we are

away from the inferential structure suggested by experts and theory. Of course model checking needs to be done along the way, as per Chap. 10.

building the distribution for is latent, and hence not measured perfectly in real data. (In effect, we have to work with posterior distributions for each of the students.) Depending on the design of the assessment, in particular, on the Q-Matrix, we may not receive any information at all about certain parameters, as when there are multiple parent variables and certain combinations of their states are rarely observed.

The procedure for elicitation in the conditional multinomial case is similar.

Example 8.8 (Conditional Multinomial Elicitation). *Suppose that the variable Skill 1 is modeled as conditional on* Overall Proficiency. *Suppose that both variables have four levels (as in Example 8.7). Then, we would ask the experts in turn, "If we had 100 students who were at the* `advanced` *level in OverallProficiency, how many of them would be* `advanced` *in Skill 1? How many would be* `proficient`, `minimal`, *and* `basic`*?" Same set of questions repeated, for when the parent values is* `proficient`, *then* `minimal` *and* `basic` *in turn. Again, we would then ask the expert the number of students their prior opinion is worth and use that to scale the Dirichlet parameters for each row of the table.*

Suppose further, that Skill 2 had both OverallProficiency and Skill 1 as parents, again with four levels each. We would now have 16 conditional distributions for the child variable to elicit from the expert, one for each combination of possible values for OverallProficiency and Skill 1.

The amount of work required to support this unconstrained distribution goes up exponentially with the number of parent variables. Consequently, eliciting the parameters is a lot of work. Even though the distribution is extremely flexible, and the "100 students" method is reasonably compact, once there are very many variables or they have many possible values, experts may not feel that their knowledge is detailed enough to be comfortable specifying all of those parameters. For this reason, the experts seem to prefer the other distribution types in the ways discussed below.

One final note, even though the probability of a cell may be small, one should be careful about setting any Dirichlet hyperparameters to zero. A zero hyperparameter means that this outcome could never occur. Any data we see in this cell of the table is an error. This makes sense if the combination is logically impossible. But otherwise we should put a small positive value in the cell instead of a zero.

Eliciting parameters for the Noisy-AND family of distributions is similar. The true-positive parameter, π_+, is a binomial proportion. The natural conjugate prior is a beta law, and we can use the same hypothetical data questions to elicit the parameters. We would ask the subject matter expert, "Of 100 students who had all of the necessary skills, how many would get the problem right?" We would then ask how many observations the expert's opinion should be worth and use this to set the hyperparameters of the beta law. The elicitation method for the false-positive parameter, π_-, is similar, with the first question now being, “Of 100 students who lacked some of the necessary skills, how many would get the problem right anyway?”

However, if we use the NIDA distribution, we need a slightly different method for the skill workaround parameters, r_k. Each r_k represents the probability that a student who has all of the required skills except *Skill k* will be able to solve the problem correctly. This is what we must elicit from the expert. There is one r_k for each parent skill, but we can make the process simpler by asking for one value that we’d use for all of them, unless the expert has substantive reasons for making some of them different. Of course when data arrives, we can use it to update all of these parameters.

Things get more complicated with the full noisy-AND distribution, with both the skill workaround and true-positive parameter. In this model, r_k represents the probability that a student who lacks only *Skill k* will behave like a student who has all of the required skills. The beta law is still the appropriate conjugate prior, but putting the question just like this is rather abstract. It is probably better to ask the expert about the probability that a student who had all of the required skills except *Skill k* would get the question right. This is $\pi_+ r_k$. If we have previously elicited π_+ we can work backward to get a prior mean for r_k.

Another possible approach is put a relatively weak prior (one with a large variance) on the r_k parameters and try to learn them from data. (This is more or less the approach taken in the Fusion model Hartz 2002; Roussos, DiBello, et al. 2007a.) However, we cannot use a completely noninformative prior here, as there is a potential problem with identifiability. Suppose that all of our skills are binary, with 1 representing mastery and 0 representing non-mastery. There is another “solution” to our model with 0 representing mastery, so we need a way to distinguish between the two.

Technically speaking, identifiability is not a Bayesian problem. As long as the prior law is a proper probability distribution, the posterior law will be a proper probability distribution as well. But in this case, we want to use our prior law to gently guide us towards one solution (the one with 1 representing mastery) as opposed to the other.

Even though identifiability is not a technical issue, we still want to pay attention to it. If a parameter is not identifiable in our model, then its prior and posterior law will be virtually identical in some aspect. This means that the posterior distribution is very sensitive to changes in the prior. This may or may not be a problem depending on the purpose of the assessment. Nonidentifiability and approximate nonidentifiability often result in technical problems (e.g., slow convergence of iterative algorithms) even in the Bayesian framework, so it is worth spending time trying to build models that avoid it.

8.6.2 Priors for DiBello's Effective Theta Distributions

Unlike the other two types of distribution, the effective theta distributions have no natural conjugate prior for the parameters. Therefore, we use a normal law to describe the distribution of the parameters. At this level of abstraction, we have no particular reason to believe that the parameters are truly normal. However, the normal law is represented by exactly two moments, the mean and the variance, and thus it serves as a workable form for "approximating" the truth. Previous experience with IRT models suggest that there may be some posterior dependence among the parameters, so we will use a multivariate normal law. However, for elicitation, we will use a diagonal covariance matrix (i.e., assume local parameter independence of the parameters). Later, the cross-parameter relationships that are hard to think about can be estimated from data.

This section discusses eliciting priors for the DiBello–Samejima model from experts who are familiar with the IRT models they are based on. The next section proposes an alternative that is based on experts' perceptions of items in terms that are more familiar to them.

One problem with the choice of a normal prior is that the normal law has infinite support. That is, any possible value of the parameters can occur. However, not all possible choices of parameter are legal in our problem. The discrimination parameters, α_k should be positive. And the level difference parameters, d_m for the Samejima distribution need to be in increasing order and must sum to zero. We need to apply some transformations to ensure we end up with appropriate priors.

For the α_k, we again follow a common practice of IRT software (such as BILOG, Zimowski et al. 2003) and model $\log \alpha_k$ as normal. However, we still elicit the mean μ_θ and "standard deviation" σ_θ of the prior law on the natural scale. We can take $\log(\mu_\theta)$ as the mean of our log-normal prior. The variance of the corresponding log–normal is $\left[\exp\left(\sigma_\theta^2 - 1\right)\right] \exp\left(2\mu_\theta + \sigma_\theta^2\right)$.

We can elicit the values for difficulty parameters, β, in the natural scale, as long as our experts are used to IRT discrimination and difficulty parameters. However, the level difficulty parameters for the Samejima graded response model, d_m, need to be transformed on this scale. When we use the constraint set $\sum_{m=1}^{M-1} d_m = 0$, there are only $M-2$ free parameters. We are better off working with the *difficulty increment* parameters: $d'_m = d_m - d_{m-1}$ for $m = 2, \ldots, M-1$. These are differences between the effective difficulties for the Samejima curves. As the center of the distribution is given by the difficulty parameter, β, we arrange the d_m parameters symmetrically around zero by setting $d_1 = -\sum_{m=2}^{M-1} d'_m/2$.

Later, when we estimate the d'_ms, we will need to make sure they are strictly positive. We will do this by giving these parameters a truncated normal law. This is a normal distribution with the probability of any value below zero set to zero. Dividing by the cumulative normal probability of being above zero (given our current mean and variance) normalizes the truncated normal law.

The regression distribution does not have this problem. We treat the c_m parameters as fixed. The $1/\sigma$ parameter is effectively another discrimination parameter, so we use the lognormal distribution for this parameter as well.

8.6.3 Linguistic Priors

Eliciting prior laws for the DiBello–Samejima Effective Theta distributions requires a fairly intimate knowledge of the IRT models that motivate them. An expert may be comfortable in stating that a given problem is easier or harder than average, but usually will be uncomfortable in assigning an exact value to its difficulty parameter. The preceding section discussed eliciting priors from experts who are familiar with IRT, so they bring the required knowledge with them. This section discusses eliciting priors in a structure that has the IRT knowledge built into it, so the expert only needs to think about familiar properties of items.

We use *linguistic priors* to accomplish this. The psychometrician creates three possible priors and labels them with appropriate terms. For example, for difficulty we have three normal priors, one centered at -1, one centered at 0, and on centered at 1. They are labeled "easier," "average," and "harder," respectively. A fourth choice label-led "unknown" also has a mean of zero, but a wider variance. The expert picks one of the labels and we use the corresponding prior. The expert or psychometrician can also enter the hyperparameters manually to produce for example a prior that represents beliefs about a task halfway between "easier" and "average" difficulty, or a prior that reflects the belief that a particular item will be "really hard"—say with a mean of 2.

Ideally, these linguistic priors would be calibrated by observing how the expert's predictions bore out over a long period of time. If that information is not available, one should make sure that the prior has a sufficiently large variance that the data will overwhelm the expert's opinion when they don't agree.

Another approach is to use hierarchical modeling, especially for the *links*, the task specific versions of the evidence models. As the task model can be used to control features of the task that influence difficulty and evidentiary focus (discrimination), we can think of the task-specific link parameters as coming from a hierarchical model (Mislevy et al. 1998). Test data from similar tasks could tell us quite a bit about new types of tasks.

Although we have not done this in the context of Bayes net models, there has been some work in this area in the context of IRT models. The techniques of Johnson and Sinharay (2003) and Glas and van der Linden (2001) should be straightforward to extend to this case. Note that these hierarchical models tie the parameters of different distributions together through shared hyperparameters and hence they do not satisfy the global parameter independence property.

Currently, the experience with the models presented in this chapter, especially Sect. 8.5, is limited. Models that seem reasonable for the cognitive phenomenon they are trying to address may prove difficult to calibrate to real data. Practical experience will bring strategies and structures for efficient use.

Further, the models described in this chapter represent only a few of the ways we might parameterize the relationship among several variables. Drawing on the rich tradition of models for psychological testing would produce many more possible parameterizations for probability tables, and for data other than discrete response values, such as response time or counts of various kinds of actions. In particular, since Bayes net proficiency variables are essentially latent classes, many of the models used in latent class analysis could be adapted to this framework. There is ample opportunity for research in this field.

Before going on to talk about how we will use these complicated models, it is worth injecting a note of caution. As we are by Step 5 well beyond the comfort level of our experts; we are building castles in latent variable space. We need to be careful how much faith we are putting in our complicated models for the hyperparameters of laws for parameters of distributions of latent variables. To a large extent, we are leaning heavily on the experts.

Remember that any model we build will not be "true," it will only be an approximation of the patterns in the real world. A given model may capture some aspects of real life well and others not. If it models the aspects of real life that are important sufficiently well, then it will be useful. A useful model that is easy to compute is more important than one which is true but intractible.

Naturally, we would like to validate our model with data. If the model is good, then it will be able to accurately predict future data. The next chapters describe mechanisms for fitting our model to data and critiquing it against data. This model checking is an important part of validating our model.

In an ECD assessment, the ultimate utility of an assessment is measured by how well claims made on the authority of the assessment are borne out in the real world. The strongest kind of evidence for determining the validity of such an assessment means finding an independent mechanism for measuring the claims, and gauging the extent to which the assessment reflects the patterns that more comprehensive "gold standard" evidence conveys. This can be challenging in practice, but it is a challenge that all assessments must ultimately meet.

Exercises

8.1 (Noisy-AND Model). Suppose that in a given population *Skill 1* and *Skill 2* are both present roughly half the students, and that furthermore, the two skills are acquired independently, so that the presence of *Skill 1* and *Skill 2* are independent in the population. Now suppose that solving a particular problem requires both *Skill 1* and *Skill 2*. Suppose further that students who lack Skill 1 have a 10 % chance of stumbling on a way to solve the problem without applying the skill and students who lack *Skill 2* have a 20 % chance of finding a workaround for the missing skill. Finally, assume that 5 % of all students who solve the problem make careless mistakes which render an otherwise correct solution incorrect.

Build a noisy-AND model for this problem. What is the joint posterior probability of the two skills for a student from this population who got the problem right? What is the joint posterior probability of the two skills for a student who got the problem wrong? Under what circumstances are the two skills independent a posteriori?

8.2 (Noisy-OR Model). Assume that *Skill 1* and *Skill 2* are distributed in the population as in the previous exercise. Now suppose we have a problem which admits two possible solution paths, one of which requires just *Skill 1* and the other of which requires just *Skill 2*. Suppose that 10 % of students who have *Skill 1* make some kind of mistake when applying this strategy and get the problem wrong, and 20 % of students who have *Skill 2* and apply it get the problem wrong.

Build a Pearl's noisy-OR model for this problem (Sect. 8.4.1). What must be assumed about the behavior of a student who has both skills? What is the joint posterior probability of the two skills for a student who got the problem right? What is the joint posterior probability of the two skills for a student who got the problem wrong? Under what circumstances are the two skills independent a posteriori?

8.3 (Hyper-Dirichlet Update). Suppose that we have a task with an observable outcome variable which can take on values `Right` and `Wrong`, and that this outcome requires *Skill 1* and *Skill 2* in some combination. Let the prior beliefs about the parameters of the conditional probability table linking *Skill 1* and *Skill 2* to the outcome variable be given by the hyper-Dirichlet distribution shown in the following table:

Skill 1	*Skill 2*	`Right`	`Wrong`
`High`	`High`	9.75	0.25
`Medium`	`High`	8.75	1.25
`Low`	`High`	5.00	5.00
`High`	`Medium`	8.75	1.25
`Medium`	`Medium`	5.00	5.00
`Low`	`Medium`	1.25	8.75
`High`	`Low`	5.00	5.00
`Medium`	`Low`	1.25	8.75
`Low`	`Low`	0.25	9.75

Suppose further that we have a sample of 1000 students who have been rated by experts on *Skill 1* and *Skill 2*. We administer the task to these students and obtain the results given in the table below:

Skill 1	*Skill 2*	Right	Wrong
High	High	293	3
Medium	High	112	16
Low	High	0	1
High	Medium	14	1
Medium	Medium	92	55
Low	Medium	4	5
High	Low	5	1
Medium	Low	62	156
Low	Low	8	172

Calculate the posterior distribution for these parameters. How much have we learned about the case in which both skills are High? How much have we learned about the case where *Skill 1* is High and *Skill 2* is Low?

8.4 (DiBello Models). Consider a skill, *Skill 1*, with three levels: High, Medium, and Low. Consider a task which uses this skill and is scored with an evidence rule which yields one of three values: No Credit, Partial Credit, or Full Credit. Build the conditional probability table for a DiBello–Samejima compensatory distribution for the relationship between the proficiency variable and the observable variable with the following parameters: discrimination of 1.0, difficulty of 0.0 and difficulty increment of 1.0.

What happens to the probability table when we increase or decrease the difficulty? When we increase or decrease the discrimination? When we increase or decrease the difficulty increment? What would happen if the type of the relationship was changed from compensatory to conjunctive or disjunctive?

8.5 (Difficulty and Weight of Evidence). Consider a skill, *Skill 1*, with three levels: High, Medium, and Low, which are evenly distributed in the population of interest. Consider a task which uses this skill and is scored Right or Wrong. Build the conditional probability table for this model with a compensatory distribution with a discrimination of 1.0 and a difficulty of 0.0. What is the expected weight of evidence for the distinction between High and Medium or Low and the weight of evidence for the distinction between Low and Medium or High.

Repeat this exercise for various difficulty values between -3 and $+3$. What happens to the weight of evidence as the difficulty changes?

8.6 (Difficulty and Population Distribution). Repeat the previous exercise with a population distribution of $\{.5, .3, .2\}$ for the states High, Medium, and `Low`. How does the relationship between difficulty and expected weight of evidence change?

Repeat with a population distribution of $\{.2, .3, .5\}$.

8.7 (Discrimination and Mutual Information). Consider a skill, *Skill 1*, with three levels: `High`, `Medium`, and `Low`, which are evenly distributed in the population of interest. Consider a task which uses this skill and is scored `Right` or `Wrong`. Build the conditional probability table for this model with a compensatory distribution with a discrimination of 1.0 and a difficulty of 0.0. Calculate the mutual information between the proficiency variable and the observable outcome variable.

Repeat this exercise for discrimination values of 0.25, 0.5, 0.75, 1.25, 1.5, and 2.0. How is the discrimination related to the mutual information?

8.8 (Reading Skill and Limited Visual Acuity). Consider a reading test where the skill *Reading* is defined as having four levels: `Advanced`, `Proficient`, `Basic`, and `Below Basic`. Assume that their distribution in the population is $\{.2, .3, .3, .2\}$. Assume that the rules of evidence for the task are such that an outcome variable has possible values: `Full Credit`, `Partial Credit,` and `No Credit`. Naturally, a reading test requires sufficient visual ability to read the words printed on the computer screen. Use a second proficiency variable *VisualAcuity* with possible values `Sufficient` and `Insufficient` to model this.

Build the conditional probability table for an inhibitor distribution to model this. Use a discrimination of 1.0 and a difficulty of 0.0. What is the mutual information between the observable and the *Reading* proficiency with the *VisualAcuity* is `Sufficient`? When it is `Insufficient`? What does this say about the appropriateness of this task for use with individuals with low *VisualAcuity*?

8.9 (Decoding Skill and Limited Visual Acuity). Consider the same model as before, but now a read aloud accommodation is offered for individuals with limited visual acuity. Assume further that the Reading construct is primarily aimed at measuring the ability to decode the words in the document. The appropriate model is now a kind of reverse inhibitor model in which persons with the accommodation act as if they are in the highest possible category of the *VisualAcuity* skill.

Build the conditional probability table for this model using a discrimination of 1.0 and a difficulty of 0.0. What is the mutual information when the accommodation is offered? Not offered? What does this say about the appropriateness of this task with the accommodation for use with individuals with low Visual Acuity?

8.10 (Weight of Evidence for a Story Problem). Consider a story problem from an algebra test. The test is designed to assess *Algebra* skill, but unless the student's *Reading* level is at least at the `Basic` level, they will not be able to perform well on the problem no matter what their *Algebra* skill is. Assume that the rules of evidence are such that there is a single observed outcome which takes on values `Right` and `Wrong`. Also assume that the Algebra variable takes on values `High` and `Low`.

Build an inhibitor model for this task. Use a discrimination and a difficulty of 1.0. Assume that the Algebra skill is distributed 50/50 in the population, and that 20 % of the population below the basic level in Reading. What is the expected weight of evidence of this problem for the Algebra proficiency? What happens if the proportion below basic on Reading drops to 10 %? Rises to 30 %?

9

Learning in Models with Fixed Structure

The preceding chapters have described an approach to assessment design and analysis that exploits the advantages of Bayesian networks. Chapter 5 showed how to update beliefs about individual students' proficiencies from the information in their responses. Carrying out these calculations in practice, however, requires the conditional probability distributions that are treated as known in Chap. 5, and may be structured as in Chap. 8. This chapter addresses the problem of estimating these distributions or parameters they are modeled in terms of.

The following section begins by introducing representational forms for the models that underlie Bayes nets, in terms of formulas and plate diagrams (Buntine 1994). It lays out a general framework for expressing educational measurement models in these terms. The next section is a brief sketch of approaches to estimating the parameters in these kinds of models, including Bayes modal estimation via the expectation–maximization (EM) algorithm and Markov Chain Monte Carlo (MCMC) estimation. Both methods are then discussed in additional detail, and illustrated with data from a simple latent class example.

9.1 Data, Models, and Plate Notation

As we have seen repeatedly, conditional independence is central to probability-based reasoning in situations with many variables. Models for large problems are easier to understand and compute when we can structure them in terms of conditionally independent instances of relatively simple and structurally similar models. The dependencies that make their joint distribution complex are attributed to their dependence on shared values of higher level parameters, which may in turn have their own simplifying hierarchical structures.

Buntine (1994) introduced a set of graphical conventions that are useful for depicting these kinds of models. They are a natural extension of the acyclic-

R. G. Almond et al., *Bayesian Networks in Educational Assessment*,
Statistics for Social and Behavioral Sciences, DOI 10.1007/978-1-4939-2125-6_9

directed graphical notation discussed in Chap. 4, with the major elaboration being for structurally identical replications of models and variables. Plate notation is introduced below with some simple examples that show the correspondence between plate diagrams and probability models. A probability model and a plate diagram for a general educational measurement model are then presented.

We then discuss inference about structures like these—parameter estimation, as statisticians typically call it, or “learning,” as it is more often called in the artificial intelligence community. A natural starting point is estimating the structural parameters of the model, namely the population proficiency distributions and conditional distributions for observable variables, when examinees’ proficiency variables are known. This so-called *complete data problem* is a straightforward application of the Bayesian updating in the binomial and multinomial models. It is also at the heart of the more realistic *incomplete data problem*, where neither examinees’ proficiencies nor structural parameters are known. Two approaches are discussed: the EM algorithm (Dempster et al. 1977) and MCMC estimate (Gilks et al. 1996). Many good references for these techniques are available, so the focus here is on the key ideas and their application in a simple problem. The special problem of bringing new tasks into an established assessment, a recurring activity in ongoing assessment systems, is also addressed.

9.1.1 Plate Notation

Plate notation (Buntine 1996) extends the notation for directed graphs we developed in Chap. 4. As before, there are nodes that represent parameters and variables, and directed edges that represent dependency relationships among them. The new idea is an efficient way to depict replications of variables and structures by displaying a single representative on a “plate” that indicates multiplicity.

As a first simple example, consider the outcomes X_j of four draws from a Bernoulli distribution with known parameter θ. The joint probability distribution is $\prod p(x_j|\theta)$. The usual digraph is shown in the top panel of Fig. 9.1. The bottom panel represents the same joint distribution using a plate for the replicated observations. The node for θ lies outside the plate—it is not replicated—and influences all observations in the same manner. That is, the structure of the conditional probability distributions for all four X_js given θ is the same. The double edge on the θ node indicates that this parameter is known, while the single edge on the X_j node indicates that their values are not known.

Plates can be nested. Consider a hierarchical model extending the previous example such that four responses are observed each of N students. For each student i, the response variables X_{ij} are Bernoulli distributed with probability θ_i[1]. These success probabilities are not known, and prior belief about them is

[1] This is a variable not a parameter as it is student specific.

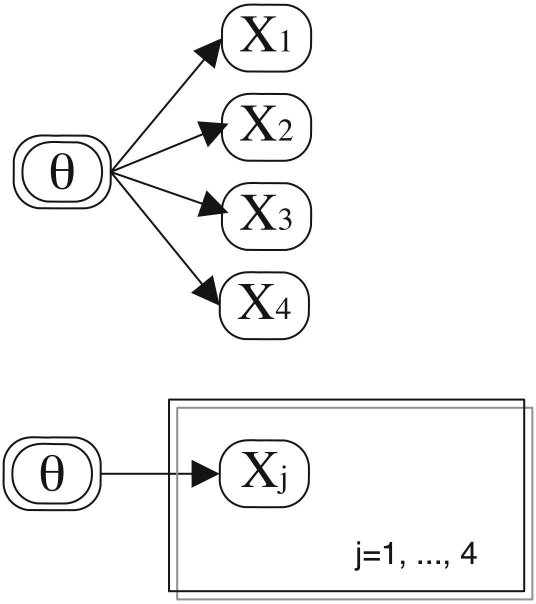


Fig. 9.1 Expanded and plate digraphs for four Bernoulli variables
Upper figure shows the full graph, *lower* figure shows the same structure with the plate notation. Reprinted with permission from ETS.

expressed through a beta distribution with known parameters α and β. The joint probability distribution is now

$$\prod_i \prod_j p\,(x_{ij} \mid \theta_i) p\,(\theta_i \mid \alpha, \beta) \ ,$$

and the digraph using plate notation is as shown in Fig. 9.2. The replications of responses for a given examinee are conditionally independent and all depend in the same way on the same Bernoulli probability θ_i, as implied by $\prod_j p\,(x_{ij} \mid \theta_i)$. Similarly, all the θ_is depend in the same way on the same higher level parameters α and β, as implied by $\prod_i p\,(\theta_i \mid \alpha, \beta)$.

A final introductory example is the Rasch model for dichotomous items, shown in Fig. 9.3. At the center of the digraph, where plates for the proficiency variables θ_i for students and difficulty parameters β_j for items overlap, is the probability p_{ij} of a correct answer by student i to item j:

$$p_{ij} = \mathrm{P}\,(X_{ij} = 1 \mid \theta_i, \beta_j) = \Psi\,(\theta_i - \beta_j) \equiv \frac{\exp\,(\theta_i - \beta_j)}{1 + \exp\,(\theta_i - \beta_j)} \ ,$$

where $\Psi\,(\cdot)$ denotes the cumulative logistic distribution. This probability is known if θ_i and β_j are known, through the functional form of the Rasch

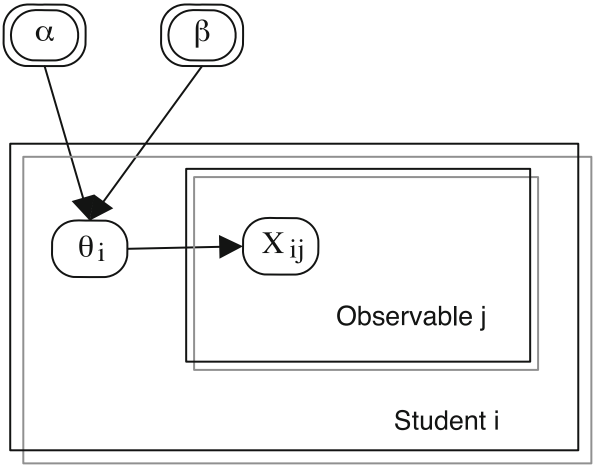


Fig. 9.2 Plate digraph for hierarchical Beta-Bernoulli model Reprinted with permission from ETS.

model; this functional relationship rather than the stochastic relationship is indicated in the digraph by a double edge on the node. That is, the double edge indicates that the value of a variable is *known* or that it is *known conditional on the values of its parents*. Logical functions such as an AND gate have this property too, but not a noisy-AND because its outcome is probabilistic.

Examinee proficiency parameters are posited to follow a normal distribution with unknown parameters mean μ_θ and variance σ^2_θ. Higher level distributions for the examinee proficiency distribution are $\mu_\theta \sim N(\mu_w, \sigma^2_w)$ and $\sigma^2_\theta \sim \text{Gamma}(a_\theta, b_\theta)$, with the parameters of the higher level distributions known. Item parameters are also posited to follow a normal distribution, with a mean fixed at zero to set the scale and an unknown variance σ^2_β, and $\sigma^2_\beta \sim \text{Gamma}(a_\beta, b_\beta)$.

9.1.2 A Bayesian Framework for a Generic Measurement Model

In educational and psychological measurement models, observable variables are outcomes of the confrontation between a person and a situation, or more specifically, a task. In particular, observable variables X are modeled as independent given unobservable, or latent, person variables[2] $\boldsymbol{\theta}$ and task

[2] In the context of maximum likelihood estimation, these are called person parameters because they must be estimated, but this book is following the convention of calling person-specific values variables rather than parameters.

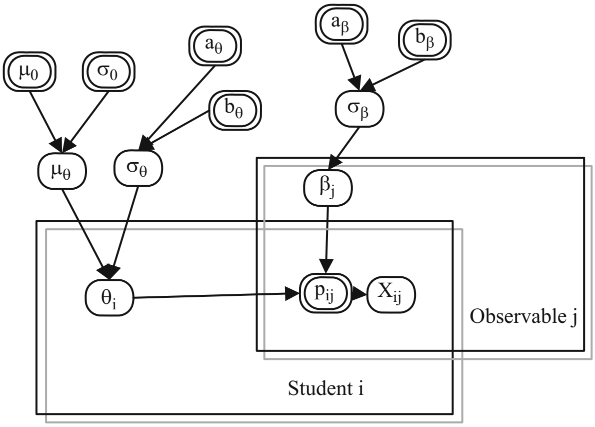


Fig. 9.3 Plate digraph for hierarchical Rasch model
Reprinted with permission from ETS.

parameters $\boldsymbol{\beta}$. In the Rasch model, for example, examinees have more or less proficiency in the same amount with respect to all items, and items are more or less difficult for all examinees. (Section 9.1.3 gives extensions where this is no longer technically true, including differential item functioning (DIF) and mixture models.)

The discrete Bayes nets that are useful in cognitive diagnosis exhibit this character. It is useful, however, to cast estimation in these models in a more general framework in order to connect it with the broader psychometric and statistical literature. The same characterization applies to the models of item response theory (IRT), latent class analysis, factor analysis, latent profile analysis, and, at the level of parallel tests, parametric classical test theory. Examples of familiar models that often have their own history, notation, and terminology are shown in Table 9.1[3]. All of these models differ only in the nature of the observable variables and student-model variables—discrete vs. continuous, for example, or univariate vs. multivariate—and the form of the link model, or probability model for observables given person variables and task parameters. We will write $p\left(x_j \mid \boldsymbol{\theta}, \boldsymbol{\beta}_j\right)$, and include a subscript i as in $p\left(x_{ij} \mid \boldsymbol{\theta}_i, \boldsymbol{\beta}_j\right)$ if we need to focus on the responses of a particular examinee i.

In discrete Bayesian models in cognitive diagnosis, $p\left(x_j \mid \boldsymbol{\theta}, \boldsymbol{\beta}_j\right)$ takes the form of a collection of categorical distributions: For a given value of the proficiency variable(s) $\boldsymbol{\theta}$, there is a categorical probability distribution over the possible values of x_j, the value of the observable variable. The (vector valued)

[3] The table is not exhaustive, as for example, classical test theory is readily applied to vectors, factor analysis can be carried out without assuming normality, and there are mixture models that include both discrete classes and classes with probabilities structured by IRT models (e.g., Yamamoto 1987).

parameter $\boldsymbol{\beta}_j$ comprises the probabilities for each response category for each given value of $\boldsymbol{\theta}$.

A full Bayesian model also requires distributions (or laws) for $\boldsymbol{\theta}$s and $\boldsymbol{\beta}$s (as they are unknown). Treating examinees as exchangeable means that before seeing responses, we have no information about which examinees might be more proficient than others, or whether there are groups of examinees more similar to one another than to those in other groups. It is appropriate in this case to model $\boldsymbol{\theta}$s as independent and identically distributed, possibly conditional on the parameter(s) $\boldsymbol{\lambda}$ of a higher level distribution or law. That is, for all examinees i,

$$\boldsymbol{\theta}_i \sim p\left(\boldsymbol{\theta} \mid \boldsymbol{\lambda}\right).$$

In the normal-prior Rasch model example in the previous section, $\boldsymbol{\lambda} = (\mu_\theta, \sigma^2_\theta)$ (the mean and variance of the normal law). If θ is the categorical variable, $\boldsymbol{\lambda}$ will be category probabilities or a smaller number of parameters that imply category probabilities in a more parsimonious way (e.g., the models discussed in Chap. 8).

Similarly, if we have no prior beliefs to distinguish among item difficulties, we can model $\boldsymbol{\beta}$s as exchangeable given the parameter(s) of their law:

$$\boldsymbol{\beta}_j \sim p\left(\boldsymbol{\beta} \mid \boldsymbol{\xi}\right).$$

In the Rasch example, $\xi = \sigma^2_\beta$. When $\boldsymbol{\beta}$s are probabilities in categorical distributions, as they are in discrete Bayes nets, then either $p\left(\boldsymbol{\beta} \mid \boldsymbol{\xi}\right)$ is a beta or Dirichlet prior on the probabilities or a function of smaller number of variables that imply the conditional probabilities. The DiBello–Samejima distributions in the previous chapter are an example of the latter.

Both $\boldsymbol{\lambda}$ and $\boldsymbol{\xi}$ may be specified as known, or as unknown with higher level distributions, or laws, $p(\boldsymbol{\lambda})$ and $p(\boldsymbol{\xi})$.

The full Bayesian probability model for educational measurement is thus

$$p\left(\mathbf{X}, \boldsymbol{\theta}, \boldsymbol{\beta}, \boldsymbol{\lambda}, \boldsymbol{\xi}\right) = \prod_i \prod_j p\left(x_{ij} \mid \boldsymbol{\theta}_i, \boldsymbol{\beta}_j\right) p\left(\boldsymbol{\theta}_i \mid \boldsymbol{\lambda}\right) p\left(\boldsymbol{\beta}_j \mid \boldsymbol{\xi}\right) p\left(\boldsymbol{\lambda}\right) p\left(\boldsymbol{\xi}\right) \ . \tag{9.1}$$

The corresponding graph, using plate notation, is shown in Fig. 9.4.

9.1.3 Extension to Covariates

We briefly note three common extensions of the basic measurement model that involve covariates, or collateral information, about examinees, denoted $\mathbf{Z}_i$, or about tasks, denoted $\mathbf{Y}_j$. (Note that the entries in a Q-matrix are collateral information about tasks.) The first is the extension to conditional exchangeability about examinee or task parameters. The second concerns interactions between examinee covariates and response probabilities, or DIF. The third concerns interactions as well, but when examinee covariates are not fully known. These are mixture models.

Table 9.1 Special cases of the generic measurement model

Model	*Student variables (θ)*	*Observables (X)*	*Link function*	*Task parameters (β)*
Parametric classical test theory	Unidimensional continuous (true score)	Continuous observed scores	Normal	Error variance
Dichotomous IRT	Unidimensional continuous	Dichotomous responses	Bernoulli	Item difficulty, discrimination, etc.
Graded response IRT	Unidimensional continuous	Ordered categorical responses	Categorical	Item step difficulties
Multidimensional dichotomous IRT	Multivariate continuous	Dichotomous responses	Bernoulli	Item difficulty, discrimination, etc.
Latent class	Discrete (class memberships)	Categorical responses (including dichotomous)	Categorical (including Bernoulli)	Conditional probabilities
Cognitive diagnosis	Discrete (attribute masteries)	Categorical responses (including dichotomous)	Categorical (including Bernoulli)	Conditional probabilities (and parameters in functional forms)
Factor analysis	Multivariate continuous	Continuous scores	Normal	Factor loadings
Latent profile analysis	Discrete (class memberships)	Continuous scores	Multivariate normal	Means and covariances for each class
Dichotomous IRT mixture	Unidimensional continuous (proficiencies) and discrete (class memberships)	Dichotomous responses	Bernoulli	Item parameters for each class
(Discrete) Bayes nets	Discrete (proficiencies and class memberships)	Categorical responses	Categorical (including Bernoulli)	Conditional probabilities (and parameters in functional forms)

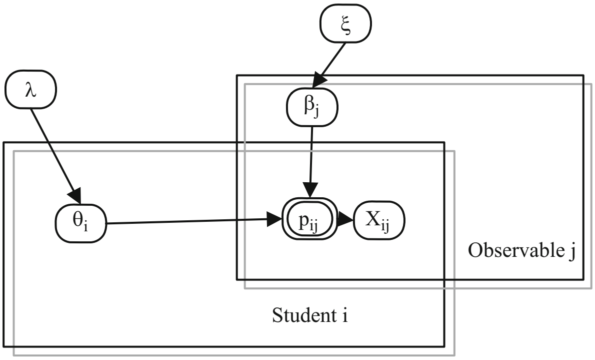


Fig. 9.4 Graph for generic measurement model
Reprinted with permission from ETS.

Z refers to known information about examinees such as demographic group or instructional background. When this information affects our beliefs about examinees' proficiencies, examinees are no longer exchangeable. Eighth grade students are more likely to know how to subtract fractions than fourth grade students, for example. We may however posit conditional exchangeability, or exchangeability among examinees with the same covariates, by making the distributions on $\boldsymbol{\theta}$ conditional on covariates, namely $p(\boldsymbol{\theta}|\mathbf{z},\boldsymbol{\lambda})$. Through $p(\boldsymbol{\theta}|\mathbf{z},\boldsymbol{\lambda})$, we model an expectation that examinees with the same value of **Z** are more similar to one another than those with other values of **Z**, or that eighth graders generally have higher proficiencies than fourth graders.

Similarly, we can incorporate covariates for tasks, through $p(\boldsymbol{\beta}|\mathbf{y},\boldsymbol{\xi})$. A task model variable **Y** can indicate the particular task model used to generate a task (or characteristics of the task defined by the task model variable), so we can use information about the parameters of similar tasks to estimate the parameters of a new task from the same family, or allow for the fact that problems with more steps are usually harder than problems with fewer steps (Mislevy et al. 1993; Glas and van der Linden 2003).

Figure 9.5 depicts the extension to covariates that influence belief about item or student parameters, but do not affect responses directly. That is, responses remain dependent on item parameters and student variables only. If we knew an examinee's $\boldsymbol{\theta}$ and a task's $\boldsymbol{\beta}$, learning **Z** or **Y** would not change our expectations about potential responses.

DIF occurs when learning **Z** or **Y** *would* change our expectations about responses, beyond parameters for the student and task (Holland and Wainer 1993). One example occurred in a reading assessment that surveyed grades 8 and 12: A task that required filling in an employment application was relatively easier for 12th grade students than for 8th grade students. Presumably many 12th grade students had personal experience with employment applica-

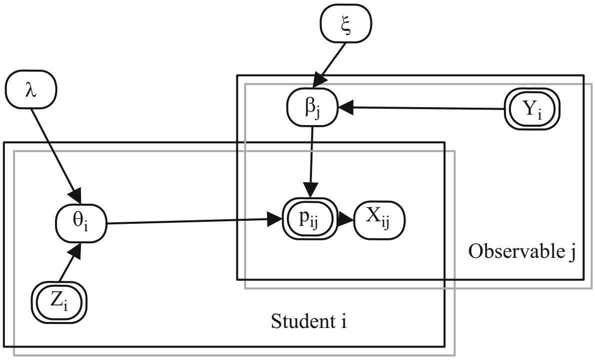


Fig. 9.5 Graph with covariates and no DIF
Reprinted with permission from ETS.

tions, a factor that would make the item easier for such a student than one who had not, but was otherwise generally similar in reading proficiency. In this case, $p\left(x_j \mid \boldsymbol{\theta}, \boldsymbol{\beta}_j, Grade = \texttt{8}\right) \neq p\left(x_j \mid \boldsymbol{\theta}, \boldsymbol{\beta}_j, Grade = \texttt{12}\right)$. In such cases, task parameters would differ for at least some tasks, and be subscripted with respect to covariates as $\boldsymbol{\beta}_{j(\mathbf{z})}$. Figure 9.6 shows how DIF appears in a graph.

DIF concerns interactions between task response probabilities and examinees' background variables $\mathbf{Z}$, when $\mathbf{Z}$ is known. Mixture models posit that such interactions may exist, but examinees' background variables are not observed (e.g., Rost 1990). An example would be mixed number subtraction items that vary in their difficulty depending on whether a student solves them by breaking them into whole number and fraction subproblems or by converting everything to mixed numbers and then subtracting (Tatsuoka 1983). Figure 9.7 depicts the digraph for a generic mixture model. It differs from the DIF model only in that $\mathbf{Z}$ is marked as unobserved rather than observed. Note that this requires specifying a prior (population) distribution for $\mathbf{Z}$. Now the higher level parameter λ for student variables has two components: λ_θ for the law for θ and λ_Z representing the probability parameter in, say, a Bernoulli law for Z if there are two classes, or a vector of probabilities for a categorical law if there are more than two classes.

9.2 Techniques for Learning with Fixed Structure

How, then, does one learn the parameters $\boldsymbol{\beta}$ of conditional probability distributions and $\boldsymbol{\lambda}$ of examinee proficiency distributions? This section addresses inference in terms of the full Bayesian probability model of Eq. 9.1, the basics of which are outlined below in Sect. 9.2.1. Section 9.2.2 discusses the simpler problem in which students' $\boldsymbol{\theta}$s are observed as well as their $\mathbf{x}$s—the "complete data" problem, in the terminology introduced in Dempster et al.

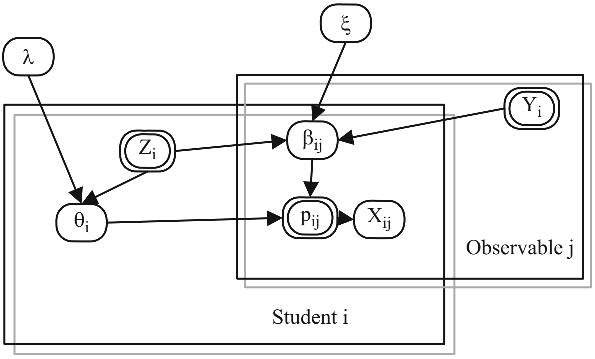


Fig. 9.6 Graph with DIF
Reprinted with permission from ETS.

(1977)'s description of their EM algorithm. Section 9.3 relates the complete data problem to the "incomplete data" problem we face in Eq. 9.1. This chapter introduces two common approaches to solve it, namely the EM algorithm (Sect. 9.4) and MCMC (Sect. 9.5). Although both algorithms are general enough to work with any model covered by Eq. 9.1, the focus of this chapter is on discrete Bayes nets measurement models.

9.2.1 Bayesian Inference for the General Measurement Model

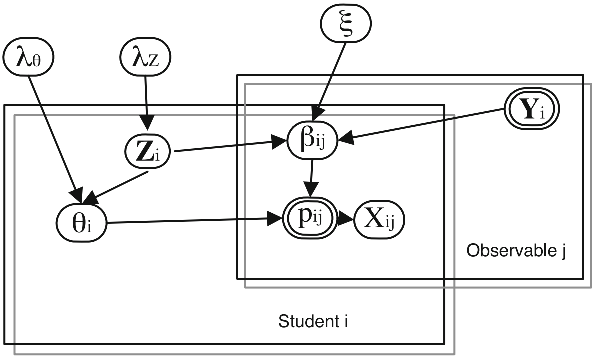


Fig. 9.7 Graph for mixture model
Reprinted with permission from ETS.

The full Bayesian model (Eq. 9.1) contains observable variables X, proficiency variables (sometimes called *person parameters*) $\boldsymbol{\theta}$, task parameters

$\boldsymbol{\beta}$, and higher level parameters $\boldsymbol{\lambda}$ and $\boldsymbol{\xi}$. Equation 9.1 represents knowledge about the interrelationships among all the variables in the generic measurement model without covariates, before any observations are made. The only observations that can be made are those for observable variables. We observe realized values of $\mathbf{X}$, say $\mathbf{x}^*$. In Bayesian inference, the basis of inference about person variables and task parameters and their distributions is obtained by examining the conditional distribution of these variables conditional on $\mathbf{x}^*$. By Bayes theorem, this posterior is easy enough to express. Aside from a normalizing constant K, it has exactly the same form as Eq. 9.1, just with the expressions involving $\mathbf{X}$ evaluated at their observed values:

$$p\left(\boldsymbol{\theta},\boldsymbol{\beta},\boldsymbol{\lambda},\boldsymbol{\xi} \mid \mathbf{x}^*\right) = K \prod_i \prod_j p\left(\mathbf{x}^*_{ij} \mid \boldsymbol{\theta}_i,\boldsymbol{\beta}_j\right) p\left(\boldsymbol{\theta}_i \mid \boldsymbol{\lambda}\right) p\left(\boldsymbol{\beta}_j \mid \boldsymbol{\xi}\right) p\left(\boldsymbol{\lambda}\right) p\left(\boldsymbol{\xi}\right) \ , \tag{9.2}$$

where

$$\begin{aligned} &K^{-1} = p\left(\mathbf{x}^*\right) \\ &= \iiiint p\left(\mathbf{x}^* \mid \boldsymbol{\theta},\boldsymbol{\beta},\boldsymbol{\lambda},\boldsymbol{\xi}\right) \partial\boldsymbol{\theta}\, \partial\boldsymbol{\beta}\, \partial\boldsymbol{\lambda}\, \partial\boldsymbol{\xi} \\ &= \iiiint \textstyle\prod_i \prod_j p\left(\mathbf{x}^*_{ij} \mid \boldsymbol{\theta}_i,\boldsymbol{\beta}_j\right) p\left(\boldsymbol{\theta}_i \mid \boldsymbol{\lambda}\right) p\left(\boldsymbol{\beta}_j \mid \boldsymbol{\xi}\right) p\left(\boldsymbol{\lambda}\right) p\left(\boldsymbol{\xi}\right) \partial\boldsymbol{\theta}\, \partial\boldsymbol{\beta}\, \partial\boldsymbol{\lambda}\, \partial\boldsymbol{\xi} \ . \end{aligned} \tag{9.3}$$

Bayesian inference proceeds by determining important features of the posterior distribution, such as means, modes, and variances of variables, picturing their marginal distributions, calculating credibility intervals, and displaying plots of marginal or joint distributions. If the prior laws are chosen to be conjugate to the likelihood, then simple procedures of posterior inference are available for independent and identically distributed samples, as illustrated in the following section for the complete data problem.

But Eq. 9.2 in its entirety does not always have a natural conjugate, even when all of the factors taken by themselves do (especially if some of the variables are missing; see York 1992). So, while the form of Eq. 9.2 is simple enough to write, the difficulty of evaluating the normalizing constant (Eq. 9.3) renders direct inference from the posterior all but impossible. Approximations must be found.

9.2.2 Complete Data Tables

The population distribution of proficiency variables (the proficiency model) and conditional probability of observable outcome variables given the proficiency variables (the evidence models and link models) in discrete Bayes nets take the form of categorical distributions, the special case being Bernoulli distributions when there are just two categories. Bayesian inference in these models is particularly simple, since they admit to posterior inference via conjugate prior distributions, and the forms of the priors, likelihoods, and posteriors have intuitive interpretations. This section describes Bayesian updating for the multinomial distributions, starting with the Bernoulli distributions and generalizing to categorical distributions. An example of parameter estimation for a complete data version of a simple Bayes net follows. We will start with a quick review of Bayesian inference for Bernoulli and categorical distributions.

A Bernoulli random variable X has two possible outcomes, which we may denote 0 and 1. Let π be the probability that $X = 1$, typically representing a "success" or "occurrence" event, or with dichotomous test items a correct response. It follows that $1 - \pi$ is the probability that $X = 0$, a "failure" or "nonoccurrence," or an incorrect item response. The probability for a single observation is $\pi^x(1-\pi)^{1-x}$. The probability for r successes in n independent replications is

$$p\left(r\,|n,\pi\,\right) \propto \pi^r \left(1-\pi\right)^{n-r} . \tag{9.4}$$

Such counts of successes in a specified number n of independent Bernoulli trials with a common parameter π are said to follow a *binomial distribution.*

Once n trials occur and r successes are observed, Eq. 9.4 is interpreted as a likelihood function, which we denote as $L\left(\pi \mid r, n\right)$. The maximum likelihood estimate (MLE) of π is the value that maximizes the likelihood, or equivalently maximizes its log likelihood $\ell\left(\pi \mid r, n\right) = \log\left[L\left(\pi \mid r, n\right)\right]$. In the case of Bernoulli trials, the MLE is $\hat{\pi} = r/n$, the sample proportion of successes, a sufficient statistic for π. The squared standard error of the mean, or error variance, is $\pi\left(1-\pi\right)/n$.

For Bayesian inference, the conjugate prior for the Bernoulli and binomial distribution is the beta distribution (defined in Sect. 3.5.3). What that means is this: If we can express our prior belief about π in terms of a beta law, then the posterior for π that results from combining this prior through Bayes theorem with a likelihood in the form of Eq. 9.4 is also a beta law. Bayesian inference through conjugate distributions has the advantages of eliminating concern about the normalizing constant (in the case of the beta distribution, the normalizing constant is the beta function evaluated at the parameters that can be looked up in a table), and relegating posterior analysis to features of a well-known family of distributions (conjugate families).

Leaving out the normalization constant, the p.d.f. for the beta distribution is:

$$p\left(\pi|a,b\right) \propto \pi^{a-1}\left(1-\pi\right)^{b-1} \;\; . \tag{9.5}$$

Comparing the form of the beta distribution in Eq. 9.5 with the form of the binomial distribution in Eq. 9.4 suggests that the beta can be thought of as what one learns about the unknown parameter π of a binomial distribution from observing $a-1$ successes and $b-1$ failures or, in other words, from $a+b-2$ trials of which the proportions $(a-1)/(a+b-2)$ are successes. This interpretation is reinforced by the combination of a beta prior and a binomial likelihood through Bayes theorem:

$$p(\pi|a,b) \times L(\pi|r,n) \propto Beta\,(\pi \mid a,b) \times L\,(\pi|r,n) \propto Beta\,(\pi \mid a+r, b+(n-r))\,. \tag{9.6}$$

That is, a posterior state of knowledge about π after updating a $Beta(a,b)$ prior with the information from r successes in n trails is the same as it would have been based on observing $a+r$ successes from $(a+b)+n$ Bernoulli trials. Table 9.2 shows what happens to the prior and posterior.

Table 9.2 Prior and posterior statistics for beta distribution with r successes in n trials

Statistic	*Prior*	*Posterior*
Law	$\text{Beta}(a,b)$	$\text{Beta}\,(a+r, b+(n-r))$
Mean	$\frac{a}{a+b}$	$\frac{a+r}{a+b+n}$
Mode	$\frac{a-1}{a+b-2}$	$\frac{a+r-1}{a+b+r-2}$
Variance	$\frac{ab}{(a+b)^2(a+b+1)}$	$\frac{(a+r)(b+n-r)}{(a+b+n)^2(a+b+n+1)}$

As the observed sample size, n, becomes large compared with the pseudo-sample size embodied in the prior, $a+b$, the posterior mean and mode approach the sample mean and the posterior variance approaches the expression for the maximum likelihood sampling variance evaluated with the MLE.

These results of Bayesian conjugate inference for the Bernoulli and binomial generalize in a straightforward manner to the categorical and multinomial distributions. Now there are K possible outcomes, $1,\ldots,K$. Let p_k be the probability that X takes the particular value x_k, where $\sum p_k = 1$. This is a *categorical distribution*, which simplifies to a Bernoulli distribution when $K=2$. Let R_k be the count of the number of observations in category k from n independent samples of the categorical variable. The collection of random variables $R_1,\ldots,R_K$ follows a *multinomial distribution*:

$$\mathrm{P}\,(R_1 = r_1, \ldots, R_K = r_K \mid \pi_1, ..., \pi_K) \propto \prod_{k=1}^{K} \pi_k^{r_k}\ , \tag{9.7}$$

when $\sum_{k=1}^{K} r_k = n$ and 0 otherwise (Sect. 3.5.3). The MLEs for the category probabilities are the observed proportions r_k/n.

The conjugate prior for the categorical and multinomial distributions is the Dirichlet distribution, with parameters $a_1,\ldots,a_K$:

$$p\,(\boldsymbol{\pi}|a_1,\ldots,a_K) \propto \pi_1^{a_1-1} \times \cdots \times \pi_K^{a_K-1}. \tag{9.8}$$

Letting $n_+ = \sum r_K$, a Dirichlet corresponds to what one knows about π from observing a sample of n_+ independent observations, with proportions a_k/n_+.

With a Dirichlet prior, then, it follows from Eqs. 9.7 and 9.8 that the posterior distribution induced by the observation of $\mathbf{r} = r_1, \ldots, r_K$ is also Dirichlet:

$$p(\boldsymbol{\pi} \mid \alpha) \times L(\pi \mid \mathbf{r}, n) \propto \text{Dirch}(a_1 + r_1, ..., a_K + r_K) \ . \tag{9.9}$$

The posterior can be thought of as the results of combining the results from a real sample of size n and a pseudo-sample of size n_+. The posterior mean for π_k is $(a_k + r_k)/(n + n_+)$.

In discrete Bayes nets, each variable has a categorical distribution given the state of its parent variables in the graph. This is the conditional multinomial distribution from Sect. 8.3. If the parameters of these categorical distributions are unknown, then the conjugate prior law will be a Beta law or Dirichlet law, depending on the number of possible categories for the variable. We require one law for each possible combination of the values of the parent variables (row of the conditional probability table). We refer to such a collection of laws as a *hyper-Dirichlet* prior law. The following example shows how updating the distribution for conditional probabilities in a Bayes net would simply be a matter of adding contingency tables of category counts if examinees' proficiency variables were known.

Example 9.1 (Complete Data Estimation). *Figure 9.8 is a simple discrete Bayes net, which also happens to be a latent class model (Dayton 1999). It contains one proficiency variable, θ, and three conditionally independent observable variables, X_1, X_2, and X_3. (The plate notation in the graph is good for any number of observables, but the example only uses three to keep things simple.) The proficiency variable θ can take two values,* 1 *for mastery and* 0 *for nonmastery. The first two observables, X_1 and X_2, are both scored outcomes from dichotomous test items, with a value of* 1 *representing a correct response and* 0 *an incorrect response. The last observable, X_3, is the scored outcome of a partial-credit task, taking values of* 0*,* 1*, and* 2 *that represent responses of increasing quality. $\boldsymbol{\pi}_j$ represents the conditional probability matrix for X_j. Which row applies to a given student is determined by which class the student belongs to, indicated by θ_i.*

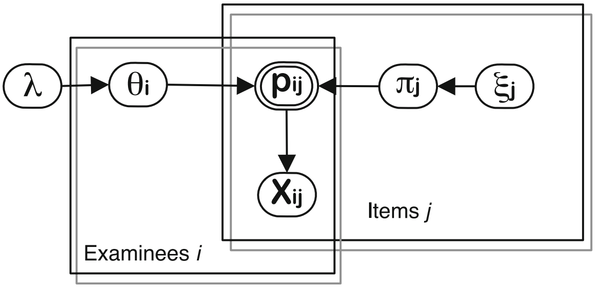


Fig. 9.8 Graph for three-item latent class model
Reprinted with permission from ETS.

Carrying out Bayesian updating for an examinee through a Bayes net requires the probability distributions (and laws) listed below. The Bernoulli distributions are written with the success probability first, while the categorical distributions are written in terms of increasing order. This is a little inconvenient here, but it will make writing the priors, data, and posteriors under Bayesian conjugate updating with the Beta and Dirichlet priors more transparent.

$$\begin{aligned}
p(\theta \mid \lambda): &\quad (\mathrm{P}(\theta=1), \mathrm{P}(\theta=0)) = (\lambda, 1-\lambda) \\
p(x_1 \mid \theta, \pi_1): &\quad \left\{ \begin{array}{l} (\mathrm{P}(X_1=1 \mid \theta=0), \mathrm{P}(X_1=0 \mid \theta=0)) \\ (\mathrm{P}(X_1=1 \mid \theta=1), \mathrm{P}(X_1=0 \mid \theta=1)) \end{array} \right\} = \left\{ \begin{array}{l} (\pi_{10}, 1-\pi_{10}) \\ (\pi_{11}, 1-\pi_{11}) \end{array} \right\} \\
p(x_2 \mid \theta, \pi_2): &\quad \left\{ \begin{array}{l} (\mathrm{P}(X_2=1 \mid \theta=0), \mathrm{P}(X_2=0 \mid \theta=0)) \\ (\mathrm{P}(X_2=1 \mid \theta=1), \mathrm{P}(X_2=0 \mid \theta=1)) \end{array} \right\} = \left\{ \begin{array}{l} (\pi_{20}, 1-\pi_{20}) \\ (\pi_{21}, 1-\pi_{21}) \end{array} \right\}
\end{aligned}$$

$$\begin{aligned}
p(x_3 \mid \theta, \pi_3): &\left\{ \begin{array}{l} (\mathrm{P}(X_3=0 \mid \theta=0), \mathrm{P}(X_3=1 \mid \theta=0), \mathrm{P}(X_3=2 \mid \theta=0)) \\ (\mathrm{P}(X_3=0 \mid \theta=1), \mathrm{P}(X_3=1 \mid \theta=1), \mathrm{P}(X_3=2 \mid \theta=1)) \end{array} \right\} \\
&= \left\{ \begin{array}{l} (\pi_{300}, \pi_{301}, \pi_{302}) \\ (\pi_{310}, \pi_{311}, \pi_{312}) \end{array} \right\}.
\end{aligned}$$

The unknown parameters needed in the Bayes net are λ, $\boldsymbol{\pi}_1$, $\boldsymbol{\pi}_2$, and $\boldsymbol{\pi}_3$. We propose prior laws that are proper but weak, with weights equivalent to six observations in each case:

$$\begin{aligned}
p(\lambda) &= \mathit{Beta}(3,3) \\
p(\pi_{10}) &= \mathit{Beta}(2,4) \\
p(\pi_{11}) &= \mathit{Beta}(4,2) \\
p(\pi_{20}) &= \mathit{Beta}(2,4) \\
p(\pi_{21}) &= \mathit{Beta}(4,2) \\
p(\boldsymbol{\pi}_{30}) &= \mathit{Dirch}(3,2,1) \\
p(\boldsymbol{\pi}_{31}) &= \mathit{Dirch}(1,2,3)
\end{aligned} \tag{9.10}$$

These priors mildly express a belief that masters and nonmasters are equally likely, and that masters are more likely to give better answers than nonmasters. If we had to carry out inference in a discrete Bayes net without accounting for our uncertainty about these probabilities, we would use the means of these priors in the network:

$$\begin{aligned}
E[p(\lambda)] &= .5 \\
E[p(\pi_{10})] &= .33 \\
E[p(\pi_{11})] &= .67 \\
E[p(\pi_{20})] &= .33 \\
E[p(\pi_{21})] &= .67 \\
E[p(\boldsymbol{\pi}_{30})] &= (.500, .333, .167) \\
E[p(\boldsymbol{\pi}_{31})] &= (.167, .333, .500).
\end{aligned}$$

To illustrate Bayesian inference for the parameters λ and $\boldsymbol{\pi}$, we generated 100 draws from the implied joint probability distribution $\mathrm{P}(\theta, X_1, X_2, X_3)$. The "true"[4] *parameter values in the simulation were $\lambda = .7$, $\pi_{10} = .1$, $\pi_{11} = .8$, $\pi_{20} = .3$, $\pi_{21} = .6$, $\boldsymbol{\pi}_{30} = (.5, .3, .2)$, and $\boldsymbol{\pi}_{31} = (.1, .2, .7)$. Table 9.3 shows the counts of patterns for (θ, X_1, X_2, X_3), the data in the complete data problem and Table 9.4 shows the counts of patterns (X_1, X_2, X_3) after collapsing over θ. The latter are what would be observed in the practice, in the incomplete data problem.*

Bayesian inference is straightforward in the complete data case. We can compute the sufficient statistics for λ, the observed counts of nonmasters and masters, say r_0 and r_1. For each item j, we can calculate the counts at each mastery level m of each response k, which we write as r_{jmk}. Writing the data in the same pattern as the priors, we obtain

$$\begin{aligned}
\lambda : &\quad (r_1, r_0) &= (76, 24) \\
\pi_{10} : &\quad (r_{101}, r_{100}) &= (0, 24) \\
\pi_{11} : &\quad (r_{111}, r_{110}) &= (51, 25) \\
\pi_{20} : &\quad (r_{201}, r_{200}) &= (4, 20) \\
\pi_{21} : &\quad (r_{211}, r_{210}) &= (58, 18) \\
\boldsymbol{\pi}_{30} : &\quad (r_{300}, r_{301}, r_{302}) &= (10, 9, 5) \\
\boldsymbol{\pi}_{31} : &\quad (r_{310}, r_{311}, r_{312}) &= (15, 15, 46)\,.
\end{aligned} \tag{9.11}$$

Note that the MLEs in the complete data problem are just the proportions from these observed counts:

$$\begin{aligned}
\hat{\lambda} &= .760 \\
\hat{\pi}_{10} &= 0 \\
\hat{\pi}_{11} &= .671 \\
\hat{\pi}_{20} &= .167 \\
\hat{\pi}_{21} &= .763 \\
\hat{\boldsymbol{\pi}}_{30} &= (.417, .375, .208) \\
\hat{\boldsymbol{\pi}}_{31} &= (.197, .197, .605)
\end{aligned}$$

Because no nonmaster got task 1 right, $\hat{\pi}_{10} = 0$—the value that gives the maximum probability for the observed data. But $\hat{\pi}_{10} = 0$ does not reflect our belief that this result is not likely but it surely is not impossible. We would not want to put this conditional probability into a Bayes net to assess new students.

Returning to the Bayesian solution, let $\mathbf{X}$ denote the complete data, that is the data in Table 9.3. The posterior laws are Beta or Dirichlet, obtained by summing the parameters of the prior from Eq. (9.10) and the observed counts from Eq. (9.11) summarized previously:

[4] In most practical problems, the truth is unknowable, but in a simulation experiment, the truth is the value chosen for the simulator input.

Table 9.3 Response pattern counts with proficiency variable, θ

Proficiencies	Observables			Observed
θ	X_1	X_2	X_3	count
0	0	0	0	6
0	0	0	1	9
0	0	0	2	5
0	0	1	0	4
0	0	1	1	0
0	0	1	2	0
0	1	0	0	0
0	1	0	1	0
0	1	0	2	0
0	1	1	0	0
0	1	1	1	0
0	1	1	2	0
1	0	0	0	0
1	0	0	1	0
1	0	0	2	0
1	0	1	0	4
1	0	1	1	6
1	0	1	2	15
1	1	0	0	6
1	1	0	1	2
1	1	0	2	10
1	1	1	0	5
1	1	1	1	7
1	1	1	2	21

Table 9.4 Response pattern counts collapsing over proficiency variable, θ

Observables			Observed
X_1	X_2	X_3	count
0	0	0	6
0	0	1	9
0	0	2	5
0	1	0	8
0	1	1	6
0	1	2	15
1	0	0	6
1	0	1	2
1	0	2	10
1	1	0	5
1	1	1	7
1	1	2	21

$$\begin{array}{rcl}
\mathrm{P}\left(\lambda \mid \theta, \mathbf{X}\right) = & \mathit{Beta}\left(3+76, 3+24\right) & = \mathit{Beta}\left(79, 27\right) \\
\mathrm{P}\left(\pi_{10} \mid \theta, \mathbf{X}\right) = & \mathit{Beta}\left(2+0, 4+24\right) & = \mathit{Beta}\left(2, 28\right) \\
\mathrm{P}\left(\pi_{11} \mid \theta, \mathbf{X}\right) = & \mathit{Beta}\left(4+51, 2+25\right) & = \mathit{Beta}\left(55, 27\right) \\
\mathrm{P}\left(\pi_{20} \mid \theta, \mathbf{X}\right) = & \mathit{Beta}\left(2+4, 4+20\right) & = \mathit{Beta}\left(6, 24\right) \\
\mathrm{P}\left(\pi_{21} \mid \theta, \mathbf{X}\right) = & \mathit{Beta}\left(4+58, 2+18\right) & = \mathit{Beta}\left(62, 20\right) \\
\mathrm{P}\left(\boldsymbol{\pi}_{30} \mid \theta, \mathbf{X}\right) = & \mathit{Dirch}\left(3+10, 2+9, 1+5\right) & = \mathit{Dirch}\left(13, 11, 6\right) \\
\mathrm{P}\left(\boldsymbol{\pi}_{31} \mid \theta, \mathbf{X}\right) = & \mathit{Dirch}\left(1+15, 2+15, 3+46\right) & = \mathit{Dirch}\left(16, 17, 49\right)
\end{array}$$

The conditional probabilities for nonmasters are estimated less precisely than those for masters, as seen by the smaller sums of parameters in the Beta and Dirichlet posteriors. This is because there were about a third as many nonmasters in the sample from which to estimate them. For example, by the formulae in Table 9.2, we see the posterior standard deviations for π_{20} *and* π_{21} *are .072 and .047. The posterior means for the parameters, which could be used in a Bayes net for calculating posterior distributions for the* θ*s of individual students, are obtained as:*

$$\begin{aligned}
E\left[\mathrm{P}\left(\lambda \mid \theta, \mathbf{X}\right)\right] &= .745 \\
E\left[\mathrm{P}\left(\pi_{10} \mid \theta, \mathbf{X}\right)\right] &= .067 \\
E\left[\mathrm{P}\left(\pi_{11} \mid \theta, \mathbf{X}\right)\right] &= .670 \\
E\left[\mathrm{P}\left(\pi_{20} \mid \theta, \mathbf{X}\right)\right] &= .200 \\
E\left[\mathrm{P}\left(\pi_{21} \mid \theta, \mathbf{X}\right)\right] &= .756 \\
E\left[\mathrm{P}\left(\boldsymbol{\pi}_{30} \mid \theta, \mathbf{X}\right)\right] &= \left(.433, .367, .200\right) \\
E\left[\mathrm{P}\left(\boldsymbol{\pi}_{31} \mid \theta, \mathbf{X}\right)\right] &= \left(.195, .207, .598\right).
\end{aligned}$$

Numerically these point estimates are not much different from the MLEs, which we would expect from using mild priors. But the zero MLE for π_{10} has been replaced by a small nonzero value, namely .067. This value arises from having observed no correct answers from 24 nonmasters when we expressed a prior expectation of .33, with the weight of 6 observations. And the identical MLEs of .197 for π_{310} and π_{311} that came from identical counts of 0 and 1 responses to X_3 from masters are shifted to posterior means of .195 and .207, an ordering that comes from our prior belief that masters' probabilities should be increasing for higher scores on X_3.

The complete data case scales to Bayesian networks of arbitrary complexity. Spiegelhalter and Lauritzen (1990) prove that if all of the prior laws for a Bayesian network are independent Dirichlet distributions (a hyper-Dirichlet law) and for each individual in the sample, there is complete data for every variable in the Bayes net, then the posterior law will be a hyper-Dirichlet law as well. The update procedure simply replicates the calculations shown here for every conditional probability table in the networks. Unfortunately, as Spiegelhalter and Lauritzen (1990) observe, this conjugacy breaks down if any of the variables are missing for any of the individuals in the sample.

9.3 Latent Variables as Missing Data

Observing values of proficiency variables (i.e., θ) would make estimation for the parameters of Bayes nets easy, but by their nature they can never be observed. What can we do? Fortunately, the perspective on missing data developed by Donald Rubin (Rubin 1977; Rubin 1987; Little and Rubin 1987) provides some leverage.

In Rubin's terms, an observation is *missing at random* (MAR) if the mechanism by which the value of a random variable came to be missing does not depend on that variable, conditional on data that are observed. This is a weaker condition than *missing completely at random* (MCAR), where the probability of missingness does not depend on the value of the variable or the values of variables that are observed as well. MCAR implies MAR. Both MAR and MCAR are independence statements. If Y is the variable that may or may not be missing, S_Y is an indicator telling whether or not Y is missing, and $\mathbf{Z}$ is the collection of completely observed data, then MAR is equivalent to the conditional independence statement $Y \perp\!\!\!\perp S_Y|\mathbf{Z}$, and MCAR is equivalent to the marginal independence statement $S_Y \perp\!\!\!\perp Y, \mathbf{Z}$.

Although the MAR assumption is central to most modern thinking about missing data, it does not always hold in practice. For example, missing responses to pain surveys are not MAR if patients do not fill out the form on days when they do not feel up to it. In such cases, the solution often lies in gathering additional fully observed covariates so that the MAR assumption holds at least approximately. The key result is that what one knows about a missing observation that is MAR is appropriately expressed by its predictive distribution, given whatever data have been observed. In other words, we must be able to model $\mathrm{P}(Y|\mathbf{Z})$, or, if necessary, $\mathrm{P}(Y|\mathbf{Z}, S_Y)$.

Mislevy (2015) works through a number of examples of inference in the presence of missing responses in IRT . Suppose an examinee is administered one of several test forms selected at random. Her response values for items on a form not presented are MCAR. In a computerized adaptive test (CAT), items are selected for administration one at a time based on an examinee's previous responses, in order to be maximally informative about that examinee's proficiency. Examinees doing well tend to receive harder items next, while students doing poorly are administered easier ones. The responses to items not administered in a CAT are MAR, but not MCAR. Suppose an examinee is presented a test booklet and decides to omit some of them because he thinks he would probably get them wrong. These responses are neither MCAR nor MAR.

An important result for assessment is that the latent variables $\boldsymbol{\theta}$ in the proficiency models are always MAR, because they are missing for everyone regardless of their values. Proficiency variables are not MCAR, however, because under psychometric models, they are instrumental in determining the probabilities of observable variables $\mathbf{X}$. Hence learning Student i's responses $\mathbf{x}_i$ provides information about $\boldsymbol{\theta}_i$. If the parameters $\boldsymbol{\lambda}$ and $\boldsymbol{\beta}$ were known, the missing value of θ_i would be appropriately replaced in Bayesian and maximum likelihood inference by the predictive distribution:

$$p\left(\boldsymbol{\theta} \mid \mathbf{x}_i, \boldsymbol{\lambda}, \boldsymbol{\beta}\right) \propto p\left(\mathbf{x}_i \mid \boldsymbol{\theta}, \boldsymbol{\beta}\right) p\left(\boldsymbol{\theta} \mid \boldsymbol{\lambda}\right) \ , \tag{9.12}$$

or if covariates $\mathbf{y}_i$ for students were also available,

$$p\left(\boldsymbol{\theta} \mid \mathbf{x}_i, \mathbf{y}_i, \boldsymbol{\lambda}, \boldsymbol{\beta}\right) \propto p\left(\mathbf{x}_i \mid \boldsymbol{\theta}, \boldsymbol{\beta}\right) p\left(\boldsymbol{\theta} \mid \mathbf{y}_i, \boldsymbol{\lambda}\right) \ .$$

These ideas lie at the heart of popular methods for estimating the parameters in Bayes nets models. They are all based on filling in, in one way or another, missing variables based on their predictive distributions given data and information about other variables in the model. Spiegelhalter and Lauritzen (1990), for example, developed an approach for learning the parameters of hyper-Dirichlet distributions in the face of missing data. In general, they are mixtures of Dirichlet distributions. One important result is that parameter estimates for $\boldsymbol{\lambda}$ and $\boldsymbol{\pi}_j$ that would have been independent (global independence) under the complete data problem generally are not independent in the incomplete data problem.

The following two sections address the problem from the point of view of two more generally applicable approaches, namely the EM algorithm and MCMC estimation. The focus is on the underlying concepts and how they play out in Bayes nets with latent proficiency variables. For more in-depth discussions of these and other estimation approaches in psychometric models, see Junker (1999) and Patz and Junker (1999b); Patz and Junker (1999a).

9.4 The EM Algorithm

Until the recent rise in popularity of MCMC estimation, the most popular method of carrying out inference for complex posteriors in measurement models was maximization (e.g., Mislevy 1986). This is because the values of the parameters that maximize Eq. 9.2, or equivalently of its log, can be found without having to calculate the normalizing constant. In many problems, once samples are large enough for the data to swamp the influence of prior distributions, the posterior is essentially normal, the maximizing values are essentially posterior means as well as modes (as well as MLEs), and the negative inverse of the matrix of second derivatives of the log posterior approximates the posterior covariance matrix.

The joint posterior mode does not behave well in certain circumstances however (O'Hagan 1976), and one of these is the case of "infinitely many incidental parameters[5] (Neyman and Scott 1948)". The general measurement model presented as Eq. 9.1 exhibits just this property. The problem is that for a fixed set of tasks, increasing the sample size of examinees also proportionally increases the number of student proficiency variables, or θs. In the language of maximum likelihood estimation, θs are "incidental parameters," in contrast to the "structural parameters" β for tasks and λ for the proficiency distribution, which do not increase with sample size. From the perspective of maximum likelihood estimation, MLEs of the parameters λ and β can be inconsistent. From the perspective of Bayesian inference, the posterior distributions of both proficiency variables (incidental parameters) and parameters (structural parameters) can be markedly nonnormal, so that the posterior modes can be far from means and the normal approximation to the posterior covariance matrix can give a misleading impression of both the shape and the dispersion of the posterior.

Approximation based on maximizing the likelihood or posterior in such cases can be improved, often dramatically so, by marginalizing or integrating over the incidental parameters, here the proficiency variables. The posterior marginal with respect to $\boldsymbol{\theta}$s is

$$p\left(\boldsymbol{\beta},\boldsymbol{\lambda},\boldsymbol{\xi}\mid\mathbf{X}\right)=\left[\int_{\boldsymbol{\theta}}\prod_{i}\prod_{j}p\left(x_{ij}\mid\boldsymbol{\theta}_i,\boldsymbol{\beta}_j\right)p\left(\boldsymbol{\theta}_i\mid\boldsymbol{\lambda}\right)\partial\boldsymbol{\theta}_i\right]p\left(\boldsymbol{\beta}_j\mid\boldsymbol{\xi}\right)p\left(\boldsymbol{\lambda}\right)p\left(\boldsymbol{\xi}\right) .$$

The expression in brackets on the right is called the marginal likelihood function. Maximum marginal likelihood (MML) estimation proceeds by maximizing this factor only with respect to $\boldsymbol{\lambda}$ and/or $\boldsymbol{\beta}$, or equivalently its logarithm, the log marginal likelihood

$$\ell\left(\boldsymbol{\beta},\boldsymbol{\lambda}\mid\mathbf{X}\right)=\log\left[\int_{\boldsymbol{\theta}}\prod_{i}\prod_{j}p\left(x_{ij}\mid\boldsymbol{\theta}_i,\boldsymbol{\beta}_j\right)p\left(\boldsymbol{\theta}_i\mid\boldsymbol{\lambda}\right)\partial\boldsymbol{\theta}_i\right] . \tag{9.13}$$

This is the formal expression of the incomplete data problem. From the perspective of maximum likelihood estimation, consistent estimates of $\boldsymbol{\lambda}$ and $\boldsymbol{\beta}$ are obtained because increasing the size of the student sample does not increase the number of parameters (including proficiency variables as parameters) to be estimated. From the perspective of Bayesian modal inference, the marginal posterior for the parameters is typically much nearer to normal

[5] In this context, the proficiency variables count as "parameters" because they must be estimated from data.

after many poorly determined nuisance variables (i.e., the proficiency variables) have been removed by marginalization.

Maximizing Eq. 9.13 can itself be a challenge, as seen in Bock and Lieberman's (1970) MML solution for IRT. Bock and Aitkin (1981) found however that rearranging the order of computations led to a more tractable iterative solution, in which each iteration presented a facsimile of the easier to solve complete data problem. Bock and Aitkin's solution turns out to be a variant of the EM algorithm. The EM exploits what is known about the missing variables, as seen in Sect. 9.3, to write the expected value of the complete data log likelihood for λ and β at each cycle, conditional on provisional estimates from the previous cycle:

$$Q^{t+1}\left(\boldsymbol{\beta},\boldsymbol{\lambda}\mid\mathbf{X}\right)=E_{\boldsymbol{\theta}}\left[\log\prod_i\prod_j p\left(x_{ij}\mid\boldsymbol{\theta}_i,\boldsymbol{\beta}_j\right)p\left(\boldsymbol{\theta}_i\mid\boldsymbol{\lambda}\right)\mid\boldsymbol{\beta}^t,\boldsymbol{\lambda}^t,\mathbf{X}\right]. \tag{9.14}$$

In Bayesian modal estimation, Eq. 9.14 is additionally multiplied by the factors for the priors, namely $\prod p\left(\boldsymbol{\beta}_j\mid\boldsymbol{\xi}\right)p\left(\boldsymbol{\lambda}\right)p\left(\boldsymbol{\xi}\right)$, before maximizing. An E-step calculates the expectation of the complete data log likelihood. The M-step maximizes the result. The estimates obtained in each such cycle increase the marginal likelihood or posterior, as required.

Some important properties of EM estimates are these (McLachlan and Krishnan 2008). Under identifiability conditions, iterations converge to a local maximum. When this is a unique global maximum, it is the MLE or posterior mode, as required. This is always the case when the problem is in the exponential family—a class of distributions that includes many distributions discussed in this book, including the normal distribution, binomial and multinomial distributions, beta and gamma distributions, and the Dirichlet distribution. Again under identifiability conditions, the rate of convergence near the maximum is geometric. The rate of convergence varies for different parameters depending on the amount of information available about them, and can be very slow. Convergence can be accelerated by methods such as those described by Lange (1995). An asymptotic normal approximation for the sampling variance or posterior covariance matrix (which are the same in the limit) can be obtained as a computational by-product by the methods of Louis (1982) and Cai (2008).

As noted earlier, the multinomial distributions that constitute the conditional distributions in Bayes nets belong to the exponential family. Applying the EM to such problems is known in the categorical analysis literature as iterative proportional fitting (Deming and Stephan 1940; Fienberg 1970; Haberman 1972). The solution takes the intuitively appealing form shown below.

Example 9.2 (An EM Solution). *We saw in Example 9.1 that the sufficient statistics for the population proportion λ in the complete data contingency table problem were the counts of nonmasters and masters, r_0 and*

r_1. *The sufficient statistics for the conditional response probabilities* $\boldsymbol{\pi}$ *in the two classes were the observed counts* r_{jmk} *of examinees in class* m *giving response* k *to item* j. *These counts are all obtained as appropriate collapsing of the information in the complete data table (Table 9.3), which gives the number of students in each mastery class with each possible response pattern. The counts cannot be observed directly in the incomplete data latent class problem, because individual students' class membership is not known.*

If the $\boldsymbol{\pi}$*s and* λ *were known, the expectations of the* r*s could be calculated from the observed response patterns. The EM solution proceeds iteratively as follows. The expectations of the* r*'s, say* $\bar{r}^{(t)}$*, are calculated using provisional estimates* $\boldsymbol{\pi}^{(t)}$ *and* $\lambda^{(t)}$ *of the structural parameters. This is the E-step. Then the facsimile of a complete data problem using an* $\bar{r}^{(t)}$ *in the place of each* r *is solved to obtain improved estimates* $\boldsymbol{\pi}^{(t+1)}$ *and* $\lambda^{(t+1)}$ *(One difference is that the EM algorithm seeks the posterior mode rather than the posterior mean). This is the M-step.*

Consider Bayes modal estimation with the counts of observed response patterns (Table 9.4) and prior distributions (9.10) from Example 9.1. Let the initial estimates of the parameters take the following values: $\lambda^{(0)} = .5$, $\pi_{10}^{(0)} = .2$, $\pi_{11}^{(0)} = .8$, $\pi_{20}^{(0)} = .2$, $\pi_{21}^{(0)} = .8$, $\boldsymbol{\pi}_{30}^{(0)} = (.43, .34, .23)$*, and* $\boldsymbol{\pi}_{30}^{(0)} = (.23, .34, .43)$.

The first E-step begins from here by calculating the expected value of the count of masters given the provisional values of the structural parameters. This can be calculated examinee by examinee (usually more efficient with small samples and long tests), or over response patterns weighted by the number of examinees with that pattern (usually more efficient with large samples and short tests). The expected count of masters, $\bar{r}_1^{(0)}$ *is obtained as the sum of the posterior probability of being a master, over all examinees:*

$$\begin{aligned}
\bar{r}_1^{(0)} &= \sum_i p\left(\theta = 1 \mid \mathbf{x}_i, \lambda^{(0)}, \boldsymbol{\pi}^{(0)}\right) \\
&= \frac{\sum_i p\left(\mathbf{x}_i \mid \theta = 1, \lambda^{(0)}, \boldsymbol{\pi}^{(0)}\right) \mathrm{P}\left(\theta = 1 \mid \lambda^{(0)}, \boldsymbol{\pi}^{(0)}\right)}{\mathrm{P}\left(\mathbf{x}_i \mid \lambda^{(0)}, \boldsymbol{\pi}^{(0)}\right)} \\
&= \frac{\sum_i p\left(\mathbf{x}_i \mid \theta = 1, \lambda^{(0)}, \boldsymbol{\pi}^{(0)}\right) \mathrm{P}\left(\theta = 1 \mid \lambda^{(0)}, \boldsymbol{\pi}^{(0)}\right)}{\sum_i p\left(\mathbf{x}_i|\theta=1,\lambda^{(0)},\boldsymbol{\pi}^{(0)}\right)\mathrm{P}\left(\theta=1|\lambda^{(0)},\boldsymbol{\pi}^{(0)}\right)+\sum_i p\left(\mathbf{x}_i|\theta=0,\lambda^{(0)},\pi^{(0)}\right)\mathrm{P}\left(\theta=0|\lambda^{(0)},\boldsymbol{\pi}^{(0)}\right)},
\end{aligned}$$

where

$$\begin{aligned}
&p\left(\mathbf{x}_i \mid \theta = 1, \lambda^{(0)}, \boldsymbol{\pi}^{(0)}\right) \mathrm{P}\left(\theta = 1 \mid \lambda^{(0)}, \boldsymbol{\pi}^{(0)}\right) \\
&= p\left(x_{i1}, x_{i2}, x_{i3} \mid \theta = 1, \boldsymbol{\pi}^{(0)}\right) \mathrm{P}\left(\theta = 1 \mid \lambda^{(0)}\right) \\
&= \left(\pi_{11}^{(0)}\right)^{x_{i1}} \left(1 - \pi_{11}^{(0)}\right)^{1-x_{i1}} \left(\pi_{21}^{(0)}\right)^{x_{i2}} \left(1 - \pi_{21}^{(0)}\right)^{1-x_{i2}} \\
&\qquad \left(\pi_{310}^{(0)}\right)^{0[x_{i3}]} \left(\pi_{311}^{(0)}\right)^{1[x_{i3}]} \left(\pi_{312}^{(0)}\right)^{2[x_{i3}]} \lambda_1^{(0)} .
\end{aligned}$$

Here, the notation $k\left[x_{i3}\right]$ *in the exponent for the terms concerning the three-category response is an indicator that takes the value* 1 *if* $x_{i3} = k$ *and* 0 *if not.*

For example, the following calculations produce the posterior probabilities for response pattern 000 in the first cycle, namely $\mathrm{P}\left(\theta=0 \mid \mathbf{X}=(000), \lambda^{(0)}, \boldsymbol{\pi}^{(0)}\right)$ *and* $\mathrm{P}\left(\theta=1 \mid \mathbf{X}=(000), \lambda^{(0)}, \boldsymbol{\pi}^{(0)}\right)$*.*

First calculate the likelihood functions for the response pattern 000, given class membership (θ) and provisional estimates of response probabilities given class membership ($\boldsymbol{\pi}^{(0)}$). These are:

$$\begin{aligned}
&\mathrm{P}\left(\mathbf{X}=(000) \mid \theta=0, \lambda^{(0)}, \boldsymbol{\pi}^{(0)}\right) \\
&=\mathrm{P}\left(X_1=0, X_2=0, X_3=0 \mid \theta=0, \boldsymbol{\pi}^{(0)}\right) \\
&=\mathrm{P}\left(X_1=0 \mid \theta=0, \boldsymbol{\pi}^{(0)}\right) \mathrm{P}\left(X_2=0 \mid \theta=0, \boldsymbol{\pi}^{(0)}\right) \mathrm{P}\left(X_3=0 \mid \theta=0, \boldsymbol{\pi}^{(0)}\right) \\
&=(1-\pi_{10})(1-\pi_{20})\pi_{300} \\
&=.8 \times .8 \times .43 \\
&=.275\ .
\end{aligned} \tag{9.15}$$

$$\begin{aligned}
&\mathrm{P}\left(X=(000) \mid \theta=1, \lambda^{(0)}, \boldsymbol{\pi}^{(0)}\right) \\
&=\mathrm{P}\left(X_1=0, X_2=0, X_3=0 \mid \theta=1, \boldsymbol{\pi}^{(0)}\right) \\
&=\mathrm{P}\left(X_1=0 \mid \theta=1, \boldsymbol{\pi}^{(0)}\right) \mathrm{P}\left(X_2=0 \mid \theta=1, \boldsymbol{\pi}^{(0)}\right) \mathrm{P}\left(X_3=0 \mid \theta=1, \boldsymbol{\pi}^{(0)}\right) \\
&=(1-\pi_{11})(1-\pi_{21})\pi_{310} \\
&=.2 \times .2 \times .23 \\
&=.0092\ .
\end{aligned} \tag{9.16}$$

The marginal probability of response pattern 000 is the average of the conditional probabilities from two mastery classes, each weighted by the provisional estimate of the proportions of masters ($\lambda^{(0)}$) and nonmasters in the population:

$$\begin{aligned}
&\mathrm{P}\left(\mathbf{X}=(000) \mid \lambda^{(0)}, \boldsymbol{\pi}^{(0)}\right) \\
&=\mathrm{P}\left(\mathbf{X}=(000) \mid \theta=0, \lambda^{(0)}, \boldsymbol{\pi}^{(0)}\right) \mathrm{P}\left(\theta=0 \mid \lambda^{(0)}, \boldsymbol{\pi}^{(0)}\right)+ \\
&\qquad \mathrm{P}\left(X=(000) \mid \theta=1, \lambda^{(0)}, \boldsymbol{\pi}^{(0)}\right) \mathrm{P}\left(\theta=1 \mid \lambda^{(0)}, \boldsymbol{\pi}^{(0)}\right) \\
&=.275 \times \left(1-\lambda^{(0)}\right)+.0092 \times \lambda^{(0)} \\
&=.275 \times .5+.0092 \times .5 \\
&=.1375+.0046=.1421\ .
\end{aligned} \tag{9.17}$$

Posterior probabilities of mastery and nonmastery classes given response pattern 000 and provisional parameter estimates $\boldsymbol{\pi}^{(0)}$ *and* $\lambda^{(0)}$*, are obtained with Bayes theorem:*

$$\begin{aligned}
\mathrm{P}\left(\theta=0 \mid \mathbf{X}=(000), \lambda^{(0)}, \boldsymbol{\pi}^{(0)}\right) &= \frac{\mathrm{P}\left(\mathbf{X}=(000) \mid \theta=0, \boldsymbol{\pi}^{(0)}\right)\left(1-\lambda^{(0)}\right)}{\mathrm{P}\left(\mathbf{X}=(000) \mid \lambda^{(0)}, \boldsymbol{\pi}^{(0)}\right)} \\
&= \frac{.1375}{.1421}=.9676\ ;
\end{aligned} \tag{9.18}$$

$$\begin{aligned}
\mathrm{P}\left(\theta=1 \mid \mathbf{X}=(000), \lambda^{(0)}, \boldsymbol{\pi}^{(0)}\right) &= \frac{\mathrm{P}\left(\mathbf{X}=(000) \mid \theta=1, \boldsymbol{\pi}^{(0)}\right)\lambda^{(0)}}{\mathrm{P}\left(\mathbf{X}=(000) \mid \lambda^{(0)}, \boldsymbol{\pi}^{(0)}\right)} \\
&= \frac{.0046}{.1421}=.0324\ .
\end{aligned} \tag{9.19}$$

The sum over all examinees of expressions like Eq. 9.19, or posterior probabilities of being in the mastery class given responses and provisional parameter estimates, produces $r_1^{(0)}$, the expected count of masters.

The expected count of correct responses to item 1 from masters is calculated in a similar manner, with the summation over only those examinees who answered correctly:

$$\bar{r}_{111}^{(0)} = \sum_{i:x_{i1}=1} \mathrm{P}\left(\theta = 1 \mid x_i, \lambda^{(0)}, \boldsymbol{\pi}^{(0)}\right) .$$

The same procedure is used to obtain expected values for all the sufficient statistics. In this example, there are only 12 possible response patterns, so computation is more conveniently carried out with respect to response patterns, then taking sums weighted by the counts of the observed patterns. Table 9.5 gives the required values by response pattern; that is, for each response pattern, it gives the observed counts (collapsing across masters and nonmasters), the likelihood of each pattern given the initial values of λ and $\boldsymbol{\pi}$ under each class, and the posterior probabilities for each class. It does so for this first iteration, and also for iterations 10 and 100. We will say more about them shortly, but for now we focus on the column for iteration 1. We can use these posterior probabilities to produce a facsimile of the complete data table, Table 9.3. A given examinee contributed to the count in exactly one line of Table 9.3, namely the line for her response pattern and her class. In the E-step, an examinee contributes in two places, partially, to the expected counts in Table 9.6: for the count in the rows with her observed response pattern for ***both*** *classes, distributed according to her provisional posterior probabilities of being in each class. For example, an examinee with pattern* 000 *contributes* .968 *to the row for* $\theta = 0$ *and* $\mathbf{X} = 000$, *and* .032 *to the row for* $\theta = 1$ *and* $\mathbf{X} = 000$. *Finally, Table 9.7 gives the resulting sums for the sufficient statistics* $\bar{r}^{(t)}$ *in iteration 1.*

The M-step consists of solving the facsimile of the complete data problem, which in this case requires combining the expected counts obtained above with the pseudo-counts given by the Beta and Dirichlet priors:

$$\begin{aligned}
\mathrm{P}^{(1)}(\lambda \mid \theta, \mathbf{X}) &= \mathit{Beta}(3+57.8, 3+42.2) &&= \mathit{Beta}(60.8, 45.2)\\
\mathrm{P}^{(1)}(\pi_{10} \mid \theta, \mathbf{X}) &= \mathit{Beta}(2+10.01, 4+32.19) &&= \mathit{Beta}(12.01, 36.19)\\
\mathrm{P}^{(1)}(\pi_{11} \mid \theta, \mathbf{X}) &= \mathit{Beta}(4+40.99, 2+16.81) &&= \mathit{Beta}(44.99, 18.81)\\
\mathrm{P}^{(1)}(\pi_{20} \mid \theta, \mathbf{X}) &= \mathit{Beta}(2+15.05, 4+27.15) &&= \mathit{Beta}(17.05, 31.15)\\
\mathrm{P}^{(1)}(\pi_{21} \mid \theta, \mathbf{X}) &= \mathit{Beta}(4+46.95, 2+10.85) &&= \mathit{Beta}(50.95, 12.85)\\
\mathrm{P}^{(1)}(\boldsymbol{\pi}_{30} \mid \theta, \mathbf{X}) &= \mathit{Dirch}(3+15.45, 2+12.88, 1+13.87) &&= \mathit{Dirch}(18.45, 14.88, 14.87)\\
\mathrm{P}^{(1)}(\boldsymbol{\pi}_{31} \mid \theta, \mathbf{X}) &= \mathit{Dirch}(1+9.55, 2+11.12, 3+37.13) &&= \mathit{Dirch}(10.55, 13.13, 37.13)
\end{aligned}$$

The EM iteration 1 provisional estimates are the modes of these M-step posteriors: $\lambda^{(1)} = .579$, $\pi_{10}^{(1)} = .264$, $\pi_{11}^{(1)} = .690$, $\pi_{20}^{(1)} = .373$, $\pi_{21}^{(1)} = .786$,

Table 9.5 E-step probabilities for iterations 1, by response pattern

Pattern	Count	$p(\mathbf{x} \mid \theta = 0)$	$p(\mathbf{x} \mid \theta = 1)$	$p(\mathbf{x})$	$\mathrm{P}(\theta = 0 \mid \mathbf{x})$	$\mathrm{P}(\theta = 1 \mid \mathbf{x})$
000	6	0.275	0.009	0.142	0.968	0.032
001	9	0.218	0.014	0.116	0.941	0.059
002	5	0.147	0.017	0.082	0.895	0.105
010	8	0.069	0.037	0.053	0.652	0.348
011	6	0.054	0.054	0.054	0.500	0.500
012	15	0.037	0.069	0.053	0.348	0.652
100	6	0.069	0.037	0.053	0.652	0.348
101	2	0.054	0.054	0.054	0.500	0.500
102	10	0.037	0.069	0.053	0.348	0.652
110	5	0.017	0.147	0.082	0.105	0.895
111	7	0.014	0.218	0.116	0.059	0.941
112	21	0.009	0.275	0.142	0.032	0.968

Table 9.6 E-step expected response pattern counts

Proficiencies	Observables			Expected count		
θ	X_1	X_2	X_3	Iteration 1	Iteration 10	Iteration 100
0	0	0	0	5.806	5.618	5.542
0	0	0	1	8.471	7.921	7.603
0	0	0	2	4.477	1.742	0.010
0	0	1	0	5.212	6.210	6.118
0	0	1	1	3.000	3.803	3.562
0	0	1	2	5.227	1.679	0.008
0	1	0	0	3.909	4.621	4.564
0	1	0	1	1.000	1.251	1.177
0	1	0	2	3.485	1.085	0.005
0	1	1	0	0.523	2.206	2.302
0	1	1	1	0.412	1.979	1.940
0	1	1	2	0.679	0.586	0.003
1	0	0	0	0.194	0.382	0.458
1	0	0	1	0.529	1.079	1.397
1	0	0	2	0.523	3.258	4.990
1	0	1	0	2.788	1.790	1.882
1	0	1	1	3.000	2.197	2.438
1	0	1	2	9.773	13.321	14.992
1	1	0	0	2.091	1.379	1.436
1	1	0	1	1.000	0.749	0.823
1	1	0	2	6.515	8.915	9.995
1	1	1	0	4.477	2.794	2.698
1	1	1	1	6.588	5.021	5.060
1	1	1	2	20.321	20.414	20.997

Table 9.7 E-step iteration 1 expectations of sufficient statistics

Nonmasters							
$\bar{r}_0^{(0)}$	$\bar{r}_{100}^{(0)}$	$\bar{r}_{101}^{(0)}$	$\bar{r}_{200}^{(0)}$	$\bar{r}_{201}^{(0)}$	$\bar{r}_{300}^{(0)}$	$\bar{r}_{301}^{(0)}$	$\bar{r}_{302}^{(0)}$
42.2	32.19	10.01	27.15	15.05	15.45	12.88	13.87
Masters							
$\bar{r}_1^{(0)}$	$\bar{r}_{110}^{(0)}$	$\bar{r}_{111}^{(0)}$	$\bar{r}_{210}^{(0)}$	$\bar{r}_{211}^{(0)}$	$\bar{r}_{310}^{(0)}$	$\bar{r}_{311}^{(0)}$	$\bar{r}_{312}^{(0)}$
57.8	16.81	40.99	10.85	46.95	9.55	11.12	37.13

$\boldsymbol{\pi}_{30}^{(1)} = (.405, .328, .267)$, *and* $\boldsymbol{\pi}_{31}^{(1)} = (.145, .185, .670)$. *These are maximum a posteriori (MAP), or posterior mode, Bayesian estimates in each M-step, maximizing the M-step provisional posterior. Over repeated iterations, the EM algorithm will converge to the MAP estimate with respect to the marginal posterior of the structural parameters.*

EM cycles continue until convergence. Table 9.6 shows the E-step expected counts of response patterns for masters and nonmasters at iterations 10 and 100. Note that expected counts for patterns in which nonmasters give responses of 2 *to task 3 are moving toward 0. Table 9.8 traces the progress of the first 10 iterations, every 10th iteration afterward up to 100, then the* 200*th and* 300*th. Convergence to three decimal places is achieved by the* 200*th, although because the EM algorithm converges only linearly it is prudent to run additional cycles after the algorithm has apparently converged to makes sure that it has truly converged and is not just moving slowly.*

Note the .000 value for π_{302}*. It is not a hard zero, but close to it. It is the counterpart of the nonmasters' expected count for a response of 2 to task 3 converging to 0, as seen in Table 9.6. This is a visible result of the EM algorithm's use of posterior modes rather than posterior means. Section 9.5 will say more about this when we compare the EM results with MCMC results, which provide posterior means. Posterior modes, posterior means, and MLEs are the same asymptotically, but a sample of 100 is small enough to make a difference. We will also see the impact of the missingness of* θ *on the posteriors for* λ *and* $\boldsymbol{\pi}$.

9.5 Markov Chain Monte Carlo Estimation

MCMC estimation takes an alternative approach to Bayesian inference from complex posterior distributions such as Eq. 9.2. Rather than analytically finding a maximum for modal inference, MCMC estimation takes samples from specially constructed distributions that in the long run are equivalent to samples from the posterior. The mean of the sampled values converges to the mean of the posterior, their standard deviation converges to the posterior's, and so on; the collection of sampled points is an empirical approximation of

Table 9.8 Trace of EM parameter estimates

Iteration	λ	π_{10}	π_{11}	π_{20}	π_{21}	π_{300}	π_{301}	π_{302}	π_{310}	π_{311}	π_{312}
0	0.575	0.238	0.712	0.347	0.808	0.386	0.307	0.307	0.157	0.199	0.644
1	0.579	0.264	0.690	0.373	0.786	0.405	0.328	0.267	0.145	0.185	0.670
2	0.583	0.276	0.678	0.387	0.774	0.421	0.342	0.237	0.135	0.176	0.689
3	0.586	0.283	0.671	0.394	0.766	0.435	0.351	0.214	0.127	0.170	0.703
4	0.590	0.287	0.666	0.399	0.761	0.447	0.359	0.194	0.120	0.166	0.714
5	0.593	0.290	0.662	0.402	0.756	0.458	0.365	0.178	0.115	0.163	0.722
6	0.597	0.292	0.658	0.404	0.753	0.467	0.370	0.164	0.110	0.161	0.729
7	0.600	0.294	0.655	0.406	0.750	0.475	0.374	0.151	0.107	0.159	0.734
8	0.603	0.296	0.652	0.407	0.747	0.483	0.377	0.141	0.103	0.158	0.739
9	0.606	0.297	0.650	0.408	0.745	0.489	0.380	0.131	0.101	0.157	0.742
10	0.609	0.298	0.647	0.409	0.743	0.495	0.383	0.122	0.099	0.156	0.745
20	0.630	0.304	0.633	0.412	0.731	0.534	0.400	0.066	0.088	0.153	0.758
30	0.642	0.304	0.626	0.412	0.725	0.553	0.410	0.037	0.087	0.153	0.761
40	0.650	0.303	0.623	0.411	0.722	0.563	0.416	0.021	0.087	0.152	0.761
50	0.656	0.302	0.621	0.409	0.720	0.568	0.420	0.012	0.088	0.152	0.759
60	0.659	0.301	0.620	0.408	0.719	0.570	0.422	0.007	0.089	0.152	0.758
70	0.662	0.300	0.619	0.407	0.718	0.572	0.424	0.004	0.090	0.153	0.757
80	0.663	0.299	0.619	0.406	0.718	0.572	0.425	0.002	0.091	0.153	0.756
90	0.664	0.299	0.619	0.406	0.718	0.573	0.426	0.001	0.092	0.153	0.756
100	0.665	0.298	0.618	0.405	0.718	0.573	0.426	0.001	0.092	0.153	0.755
200	0.667	0.297	0.618	0.404	0.717	0.573	0.427	0.000	0.093	0.153	0.754
300	0.667	0.297	0.618	0.404	0.717	0.573	0.427	0.000	0.093	0.153	0.754

the posterior, as accurate as we like by just making the chain of draws long enough.

It is important to note that increasing the number of (MCMC) samples from the posterior does not add information about the unknown variables per se. The posterior contains all the information to be had from the model, the prior, and the observed data. Running indefinitely many cycles reduces approximation error due to the fact that the run output is a larger sample from the posterior, while the posterior itself, and the uncertainty due to having only a fixed amount of observed data remains the same no matter how long the chain is.

The "Markov[6] chain" part of MCMC refers to the property of drawing from probability distributions in a sequence, where each draw leads to a next distribution to draw from and the procedure works from one step to the next with "no memory" of previous steps that led it to the present state.

[6] This is a different context from which the term "Markov" was introduced in Chap. 4. In graphs the "Markov" property is that two variables are conditionally independent given their separator. In time series, the "present" is the separator which renders the past independent of the future.

"Monte Carlo" integration is a trick often employed to calculate a complex integral whose integrand can be written as a probability distribution times a function of the random variable. This is the expected value of the function (under the probability distribution), and the mean of a random sample of function values is approximately equal to the value of integral. The approximation gets better as the number of values sampled from the probability distribution increases. Thus, if we can sample from the full joint posterior, Monte Carlo integration is capable of calculating features such as means, variances, and percentile points, as well as smoothed approximations of the distribution.

Putting these ideas together, MCMC uses a Markov Chain (a time series) to sample from the joint posterior. This is not an independent sample because each sampled value depends on the one in the previous time point. The lack of independence does not bias the results if the series is long enough—the draws for a given variable are usually correlated with the previous ones, but the marginal distribution is right. It does mean though that a larger number of draws (i.e., longer chains) is necessary for a given level of accuracy. MCMC estimation can take a great deal of computation, especially with large models and large data sets. Much of the art of MCMC estimation comes from trading-off dependency between number of iterations and ease of drawing samples taken from the requisite probability distributions.

The software package `BUGS` (Bayesian analysis using Gibbs sampling Thomas et al. 1992, see Appendix A) has played a special role in the popularity of MCMC estimation. It takes a description of a model and develops a MCMC sampler for that model, so the analyst does not need to write one-off software for each problem. The successor package `WinBUGS` (Lunn et al. 2000) adds a graphical user interface and some of the types of graphics used below. We used WinBUGS in initial work on the examples in this chapter.[7] BUGS has had a huge impact in making Bayesian methods more widely available.

A full treatment of MCMC methods is beyond the current presentation. A number of excellent resources are now available, however; the reader is referred to Gelman et al. (2013a) and Gilks et al. (1996) for a start. This section describes the approach and basic properties of MCMC estimation, with a focus on popular variants called Gibbs sampling (Sect. 9.5.1) and the Metropolis–Hastings algorithm (Sect. 9.5.3). It then continues the running latent class example to show how Gibbs sampling can be applied to estimate the parameters in Bayesian networks.

[7] For the final version, we used our own MCMC software StatShop (Almond, Yan, et al. 2006c). We tested that software by comparing the posteriors it generated to those generated by WinBUGS. The graphics were prepared with CODA (Best et al. 1996), a package of R (R Development Core Team 2007) functions for doing MCMC output analysis originally developed for use with BUGS output.

9.5.1 Gibbs Sampling

The Gibbs sampler is a variety of MCMC estimation with a particularly simple form (Geman and Geman 1984). Each cycle consists of a set of unidimensional draws, one for each unknown variable or parameter, from what is called a "full conditional" distribution. In other words, a draw is taken from a distribution for that variable conditional on a value for every other variable and parameter in the problem—actual values for variables that have been observed, such as observable variables or covariates, and, for unobserved variables, the previous draw for that variable. Iteration $t+1$ in the general measurement model (Eq. 9.1) starts with values for all of the variables, say $\left\{\boldsymbol{\theta}^t, \boldsymbol{\beta}^t, \boldsymbol{\lambda}^t, \boldsymbol{\xi}^t\right\}$. A value is then drawn from each of the following full conditional distributions in turn:

$$\begin{array}{l}
\text{For each person, } i \text{, draw } \theta_i^{t+1} \text{ from } p\left(\theta_i \mid \boldsymbol{\theta}_{<i}^{t+1}, \boldsymbol{\theta}_{>i}^{t}, \boldsymbol{\beta}^t, \boldsymbol{\lambda}^t, \boldsymbol{\xi}^t, \mathbf{X}\right). \\
\text{For each item, } j \text{, draw } \beta_j^{t+1} \text{ from } p\left(\beta_j \mid \boldsymbol{\theta}^{t+1}, \boldsymbol{\beta}_{<j}^{t+1}, \boldsymbol{\beta}_{>j}^{t}, \boldsymbol{\lambda}^t, \boldsymbol{\xi}^t, \mathbf{X}\right). \\
\text{Draw elements } \lambda_k^{t+1} \text{ from } p\left(\lambda_k \mid \boldsymbol{\theta}^{t+1}, \boldsymbol{\beta}^{t+1}, \boldsymbol{\lambda}_{<k}^{t+1}, \boldsymbol{\lambda}_{>k}^{t}, \boldsymbol{\xi}^t, \mathbf{X}\right). \\
\text{Draw elements of } \xi_\ell^{t+1} \text{ from } p\left(\xi_\ell \mid \boldsymbol{\theta}^{t+1}, \boldsymbol{\beta}^{t+1}, \boldsymbol{\lambda}^{t+1}, \boldsymbol{\xi}_{<\ell}^{t+1}, \boldsymbol{\xi}_{>\ell}^{t}, \mathbf{X}\right).
\end{array} \tag{9.20}$$

Run cycles like these long enough, and under broadly satisfied conditions taking a draw for a given parameter follows the same distribution as a draw from its marginal posterior distribution, given the observed data. What is more, the set of draws for all parameters in a given cycle follows the same distribution as a multivariate draw from their joint posterior. The difficult problem of characterizing a complex posterior with many parameters has been reduced to a series of draws from unidimensional distributions that are usually much easier to work with. The key condition for the Gibbs sampler to work is reversibility, which means that it is possible for the chain to move from any point in the parameter space to any other point, so it cannot become "stuck" in some region of the space.[8]

Conditional independence relationships in the joint distribution can make Eq. 9.20 easier to work with. In particular, with a Bayes net, we can often use the conditional independence statements implicit in the network to simplify calculations. In particular, in drawing a sample for a variable θ_i, we usually only need to consider the neighbors of θ_i in the graph. Furthermore, if the global independence condition holds, we only need to consider the parameters for one conditional probability table at a time. The local independence condition brings about further simplifications. Thus Gibbs sampling is a natural method to use with Bayes nets. It can exploit the conditional independence properties of the net to make the calculations more efficient.

[8] For example, consider a model containing variables $X_1, \ldots, X_N$ that are assumed to be normally distributed with unknown mean μ_X and variance σ_X. Suppose the sampler moves to a state with $\sigma_X = 0$. Then all $X_i = \mu_X$ for the next cycle, which will in turn force $\sigma_X = 0$ in the following cycle as well, and ever after. To avoid such problems, the prior for a variance parameter is usually restricted so that the probability of $\sigma_X = 0$ is zero.

9.5.2 Properties of MCMC Estimation

Convergence of a Gibbs sampler is not to a point (a single value for each parameter and unknown variable), as it is in the EM solutions discussed in the previous section, but rather to a stationary distribution. That is, the joint distribution of the draws at t through $t+\ell$ is the same as that of $t+m$ through $t+\ell+m$ for any $m>0$. If a Markov Chain is in its stationary distribution at one point in time, then it will still be in the stationary distribution at every future time point.

The key result from Geman and Geman (1984) is that the stationary distribution of the Gibbs sampler *is* the joint posterior of the unknown parameters and variables. In the limit, draws from Gibbs cycles move around the posterior in proportion to its density. Of course being possible to move from any part of the parameter space to another in a given cycle does not mean it is probable. Although only positive probability is required for long run behavior to be satisfied, there is no guarantee that any finite portion of a chain covers the support of the posterior (the space over which it is defined) representatively. A practical challenge in any application of Gibbs sampling, then, is how long the chain should be.

Draws in cycle $t+1$ depend on values in cycle t, but given them, not on previous cycles. This conditional independence is the Markov property of "no memory." In general, however, values for a given parameter in a chain do tend to be autocorrelated from cycle to cycle, so there is less information about a parameter's posterior in M successive draws from the Markov chain than there would be from M independent draws from the same posterior. Under regularity conditions, dependence on starting values is 'forgotten' after a sufficiently long run. After convergence to a stationary distribution, the empirical distribution and summary statistics of a long series of draws estimate the target posterior distribution and its summary characteristics.

The autocorrelations of a time series are the correlations of a point in the series $x^{(t)}$ with previous points in the series $x^{(t-\ell)}$. The difference between the two points is called the *lag* of the autocorrelation. High autocorrelation means the Markov chain is moving more slowly around the parameter space. This is called slow mixing or poor mixing. It can be caused by lack of information about the parameter in the data, and by high correlations with other parameters; sometimes reparameterizing the model can help. A slow mixing chain takes longer to visit all regions of the posterior, so it needs to be run longer than a series that is mixing well. Slow mixing can be a sign of other trouble as well, so it is worth checking to make sure the chain is visiting all parts of the posterior. If a Gibbs sampler is started from initial points that are in a low density region of the posterior, it can take many cycles before stationarity is reached. Draws from early cycles are discarded for the purpose of approximating the posterior. This is called "burn-in." We will return to this idea again shortly.

Figure 9.9 shows a trace plot (after burn-in) for two chains along with the autocorrelations plotted against the lag. The first (an intercept parameter in a DiBello–Samejima model) is mixing well, while the second parameter (a slope parameter) is not. Different variables in the same problem can have good and poor mixing. A good trace plot should look like "white noise" that is bouncing around with no discernible pattern. If we run the second series long enough, then compression of the time scale will make the second plot look like white noise. That is an indication that we have run the series longer enough to compensate for slow mixing.

The autocorrelation plots for the same series are also plotted in Fig. 9.9. We can see that the autocorrelation for the intercept parameter drops to a fairly low figure by about lag 5, while the autocorrelation for the slope parameter stays high even to lags of 20 and higher. When the autocorrelation for a given parameter is high, more cycles are required for burn-in and more cycles are required for a given number of draws to provide a given degree of accuracy for estimating the posterior. A measure of the latter effect is the *effective sample size* of the Markov chain—this is the sample size of a simple random sample from the posterior which would give a similar accuracy. A correction factor from time series analysis gives a reasonable approximation for the impact of autocorrelation ρ on the estimate of the posterior mean of a variable from a chain of length N:

$$\textit{Effective sample size} \approx N\left(\frac{1-\rho}{1+\rho}\right). \tag{9.21}$$

In the series shown in Fig. 9.9, the intercept parameter has an effective sample size of 1161, which is probably adequate for estimating posterior means, standard deviations, and 95% credibility intervals. The slope parameter has an effective sample size of 80, which indicates more samples are needed. In both cases, the full sample consists of 15,000 draws (5000 each from three independent chains, i.e., three chains started from different initial values).

To check convergence, Gelman and Rubin (1992) recommend running multiple chains from different, widely dispersed, initial values, and monitoring whether they come to approximate draws from the same stationary distribution. Figure 9.10 shows the first 2000 draws from the slope parameter from Fig. 9.9. The three chains were started from three different starting points. There is evidence that the chains have converged at about the 1000th cycle. The values before this point should not be used to approximate the characteristics of the posterior distribution, and are discarded as "burn-in" cycles.

Brooks and Gelman (1998) discuss ANOVA-like indices for a given variable that compare variance between (B) and variance within (W) n chains. One intuitive simple index is

$$R = \frac{B+W}{W}.$$

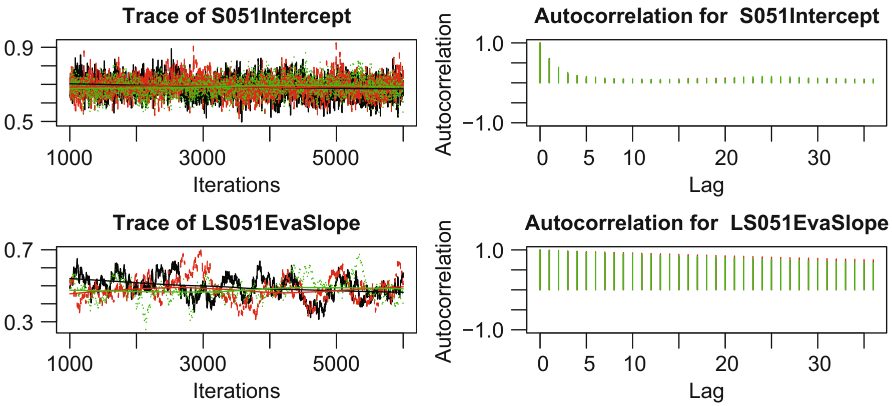


Fig. 9.9 Examples of poor mixing and good mixing

The graph shows trace and autocorrelation plots for two parameters (an intercept or difficulty parameter and log slope or discrimination parameter) from a DiBello-Samejima model for Observable S051. The first 1000 samples (not shown in the plot) were discarded as "burn-in." The intercept parameter is mixing well with the autocorrelation damping down at higher lags, while the discrimination parameter is mixing slowly with the autocorrelation remaining high even at longer lags. Reprinted with permission from ETS.

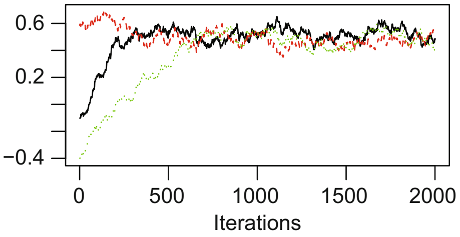


Fig. 9.10 Convergence of three Markov chains

Before cycle 1000, the three *chains* do not overlap, but after this time they substantially overlap. The *chain* has almost certainly not reached its stationary distribution before cycle 1000. Reprinted with permission from ETS.

One looks at values of R as computed in windows (short segments of the chain) of length 50. If the chains have converged to the stationary distribution, R is near 1. But if the chains have not converged, R exceeds 1 as B tends to overestimate the variance of the stationary distribution because the chains were started from overdispersed initial values, while W tends to underestimate the variance of the stationary distribution because the chains have not yet covered the range of the parameter space. Figure 9.11 is the trace of an adjusted version of R, the Brooks–Gelman–Rubin (BGR) index for the same series shown in Fig. 9.10. R has settled to a value near 1 by cycle 1000 suggesting that this is a good choice for burn-in (The *WinBUGS User Guide* suggests looking more closely at additional evidence, such as visual inspection of trace plots and densities from different chains, when values exceed 1.05; Brooks and Gelman (1998) mention values like 1.1 and 1.2 as being satisfactory, but emphasize looking at other sources of evidence rather than relying on any single criterion). Sinharay (2003) reviews convergence diagnostics for MCMC, including multivariate versions of the BGR.

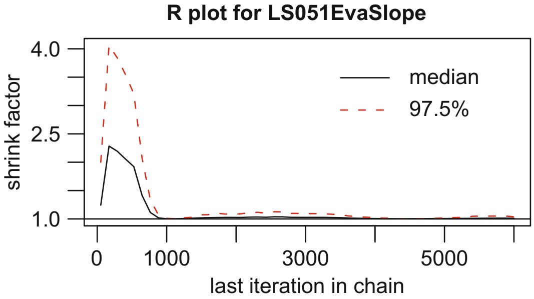


Fig. 9.11 Plot of the Brook–Gelman–Rubin R vs. cycle number
The R statistic is near 1 by iteration 1000 suggesting a burn-in of 1000. Reprinted with permission from ETS.

There are many variations of MCMC that all enjoy the same long-run properties but are more efficient for different specific problems. For example, generalizations of the basic Gibbs sampler include the following: variables can be sampled in different orders in different cycles. Not every variable must be sampled in each cycle; it is sufficient that each variable will be sampled infinitely many times in some mix with all the others. (One might draw ten values of the slope parameter in the example of Fig. 9.9 from the same full conditional, for instance, to help compensate for its large autocorrelation compared to that of the intercept.) Sampling can be carried out drawing from conditionals of blocks of parameters rather than individual parameters (Patz and Junker 1999b, for example, draw from the trivariate distributions of each task's item parameters $\log a$, b, and $\operatorname{logit} c$ in the three-parameter logistic IRT model).

9.5.3 The Metropolis–Hastings Algorithm

There are handy programs to sample from the beta, Dirichlet, normal, and gamma distributions we have seen as forms for full conditionals so far. If the full conditional of a given parameter is difficult to sample from, one can carry out approximations such as Metropolis or Metropolis–Hastings sampling within Gibbs cycles. A Metropolis sampling approximation to a density $p(\cdot)$ (Metropolis and Ulam 1949) requires that one be able to compute the density of the target at any given point, say $p(z)$. Samples are not drawn from $p(\cdot)$, but from a proposal distribution $q(\cdot)$ that has the properties that one can both compute it at any point z and it is easy to draw samples from. In Metropolis sampling, the idea is to draw a value from the proposal distribution, and either accept this draw as the next value in the chain for the parameter or to stay with the previous value. The probability of accepting the draw (Eq. 9.22 or 9.23) is chosen to correct for the difference between $q(\cdot)$ and $p(\cdot)$. Therefore, the stationary distribution of the Metropolis sampler is still the target distribution.

Let p be the target distribution in this case, the full conditional of a given parameter z in the full Bayesian model. Let $z^{(t)}$ be the value for this parameter in the tth cycle. Denote the proposal distribution for cycle $t+1$ by $q(y|z^{(t)})$. Note that the proposal distribution may depend on the previous value $z^{(t)}$. WinBUGS, for example, uses normal proposal distributions when it cannot sample directly from full conditionals, with the mean at the previous value. Proposal distributions can be described as symmetric or nonsymmetric. A symmetric proposal distribution (such as WinBUGS's) has the property that $q(y|z^{(t)}) = q(z^{(t)}|y)$. In this case, the probability of accepting a proposal value y drawn from $q(y|z^{(t)})$ is given by

$$\alpha\left(z^{(t)}, y\right) = \min\left(1, \frac{p(y)}{p(z^{(t)})}\right) \ . \tag{9.22}$$

In other words, the proposed value y can become $z^{(t+1)}$ in two ways. It is accepted with certainty if the density of the target distribution p is higher than that of the previous cycle's estimate. If its density is lower, it may still be accepted, with a probability equal to the ratio of the densities of the target distribution at the proposed value and the previous cycle's value. Figures 9.12 and 9.13 illustrate these two situations.

Metropolis–Hastings sampling pertains to proposal distributions that are not symmetric (Hastings 1970). The probability of acceptance is now

$$\alpha\left(z^{(t)}, y\right) = \min\left(1, \frac{p(y)q(z^{(t)}|y)}{p(z^{(t)})q(y|z^{(t)})}\right) \ . \tag{9.23}$$

Metropolis–Hastings sampling simplifies to Metropolis sampling if $q(\cdot,\cdot)$ is symmetric.

Remarkably, Metropolis and Metropolis–Hastings sampling work for practically any proposal distribution that is positive over the support of the target distribution. Different choices may differ considerably as to their efficiency, however. If the proposal distribution is chosen to take random steps from the current value, then the step size (variance) of the proposal distribution will have a big impact on the efficiency of the chain. If the step size is too large,

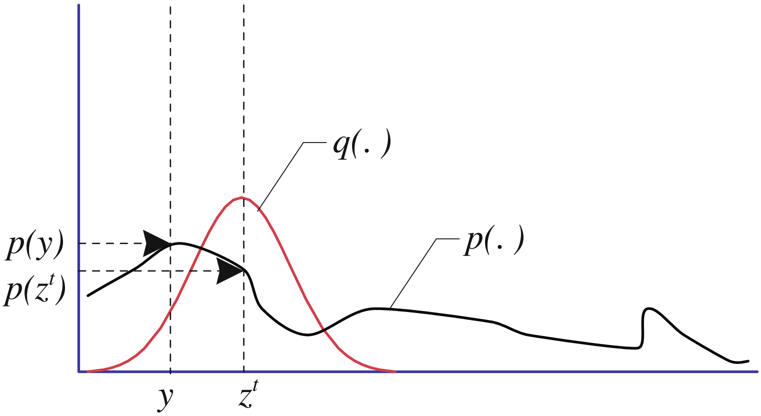


Fig. 9.12 A Metropolis step that will always be accepted
Reprinted from Almond et al. (2006a) with permission from ETS.

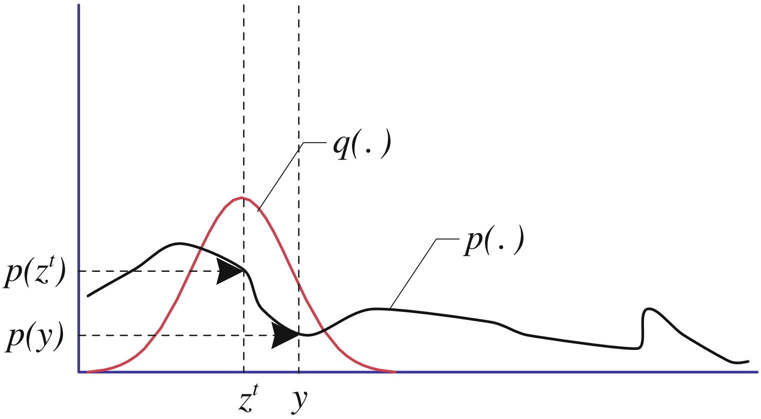


Fig. 9.13 A Metropolis step that might be rejected
Reprinted from Almond et al. (2006a) with permission from ETS.

then the Metropolis algorithm will reject frequently. This leads to slow mixing, since the next value in the chain is too often the same as the previous value. On the other hand, if the step size is too small, the chain will move very slowly through the space because any new accepted value is usually close to the previous step. This also results in slow mixing. Proposal distributions that lead to 30–40 % acceptance are most efficient (Gelman et al. 1996). WinBUGS automatically tunes the proposal distribution (adjusting the variance as it tracks the acceptance rate) during the burn-in cycles using Metropolis sampling. For this reason, WinBUGS requires a longer burn-in for problems in which it uses Metropolis sampling.[9]

Although the original Metropolis and Metropolis–Hastings algorithms sampled from the joint distribution of all the unknown variables and parameters, it is possible to combine this technique with Gibbs sampling. As in Gibbs sampling, one samples from each variable or parameter in turn (or in a randomly chosen order). If the full conditional distribution is convenient to work with (e.g., the conditional independence properties of the Bayes net usually make for relatively straightforward full conditionals), then a draw is taken from the full conditional. If not, then a draw for that variable or parameter is taken from a proposal distribution and it is then accepted/rejected with a Metropolis (Eq. 9.22) or Metropolis–Hastings (Eq. 9.23) rule. WinBUGS uses this strategy, automatically deciding whether to use Gibbs sampling, Metropolis sampling, or some other algorithm, such as the slice sampler (Neal 2003), on a parameter-by-parameter, variable-by-variable basis.

9.6 MCMC Estimation in Bayes Nets in Assessment

In typical operational assessment programs, the latent proficiency variables ($\boldsymbol{\theta}$) of students are of persistent interest. Particular tasks and their parameters ($\boldsymbol{\pi}$) are moved into and out of the system over time, whether for security purposes, to extend the range of ways to collect evidence, or simply to provide variety for students. We would be perfectly well satisfied if a student were assessed using this set of tasks or that one, as long as they provide evidence with respect to the same proficiencies. The latent scale is established in a start-up data collection in which all examinee, task, and structural parameters are estimated. This is called the *initial calibration.* The resulting structural parameters can be used to estimate the $\boldsymbol{\theta}$s of new examinees as they are administered tasks for which good estimates of $\boldsymbol{\pi}$s are available.

As the testing program continues, some tasks are retired or rotated out of operational use. New tasks are created to provide information about the same $\boldsymbol{\theta}$s. We assume that the new items are created in accordance with existing task models and conformable evidence models. In these cases, we estimate the parameters for the evidence models of these new tasks. To accomplish this, new tasks are administered to examinees along with already-calibrated or old tasks, and the $\boldsymbol{\pi}$s of the new tasks are estimated with the $\boldsymbol{\pi}$s of the old tasks setting the scale. This is called *online calibration* (Wainer et al. 2000).

This section describes MCMC estimation using Gibbs sampling for both initial calibration and online calibration in Bayes nets. Additional detail can be found in Hrycej (1990), Mislevy, Almond, et al. (1999a), and York (1992).

[9] In fact, the slow mixing series shown in Fig. 9.9 was produced by artificially shortening the time available for WinBUGS to tune the proposal distribution.

The approach will be illustrated with the latent class example introduced earlier in the chapter.

9.6.1 Initial Calibration

The form of Gibbs sampling for the general measurement model (Eq. 9.20) specializes in the case of Bayes nets as follows:[10]

$$\begin{aligned}&\text{For each person, } i\text{, draw } \boldsymbol{\theta}_i^{(t+1)} \text{ from } p\left(\boldsymbol{\theta}_i \mid \boldsymbol{\theta}_{<i}^{(t+1)}, \boldsymbol{\theta}_{>i}^{(t)}, \boldsymbol{\pi}^{(t)}, \boldsymbol{\lambda}^{(t)}, \mathbf{X}\right).\\ &\text{For each task, } j\text{, draw } \boldsymbol{\pi}_j^{(t+1)} \text{ from } p\left(\boldsymbol{\pi}_j \mid \boldsymbol{\theta}^{(t+1)}, \boldsymbol{\pi}_{<j}^{(t+1)}, \boldsymbol{\pi}_{>j}^{(t)}, \boldsymbol{\lambda}^{(t)}, \mathbf{X}\right).\\ &\text{Draw class proportions } \boldsymbol{\lambda}_k^{(t+1)} \text{ from } p\left(\lambda_k \mid \boldsymbol{\theta}^{(t+1)}, \boldsymbol{\pi}^{(t+1)}, \boldsymbol{\lambda}_{<k}^{(t+1)}, \boldsymbol{\lambda}_{>k}^{(t)}, \mathbf{X}\right).\end{aligned} \tag{9.24}$$

The first line of Eq. 9.24 is drawing values for proficiency variables from posterior distributions for them, student by student, given their response patterns and the previous cycle's draws for conditional response probabilities $\boldsymbol{\pi}$ and $\boldsymbol{\lambda}$. Note that given the model parameters, student proficiencies are independent, so $p\left(\boldsymbol{\theta}_i \mid \boldsymbol{\theta}_{<i}^{(t+1)}, \boldsymbol{\theta}_{>i}^{(t)}, \boldsymbol{\pi}^{(t)}, \boldsymbol{\lambda}^{(t)}, \mathbf{X}\right) = p\left(\boldsymbol{\theta}_i \mid \boldsymbol{\pi}^{(t)}, \boldsymbol{\lambda}^{(t)}, \mathbf{X}_i\right)$ and each student can be considered in isolation. Sampling from this distribution for each student provides an augmented data set containing the actual response data $\boldsymbol{X}$, and for each examinee a provisional value of $\boldsymbol{\theta}$.

The posterior distributions of the $\boldsymbol{\pi}$s and $\boldsymbol{\lambda}$s can be obtained as they were in the complete data solution discussed in Sect. 9.2.2. In particular, the local independence property of the assessment—that the observable outcomes for given tasks are independent given the proficiency variables—often simplifies the required conditional probabilities in Eq. 9.24. For example, the second line of Eq. 9.24 describes how to draw the task specific link model parameters. If in addition the global parameter dependence model holds (in particular, we are not using a hierarchical model for the task parameters), then the parameters from each task are rendered independent by conditioning on the proficiency variables. Thus, we have: $p\left(\boldsymbol{\pi}_j \mid \boldsymbol{\theta}^{(t+1)}, \boldsymbol{\pi}_{<j}^{(t+1)}, \boldsymbol{\pi}_{>j}^{(t)}, \boldsymbol{\lambda}^{(t)}, \mathbf{X}\right) = p\left(\boldsymbol{\pi}_j \mid \boldsymbol{\theta}^{(t+1)}, \boldsymbol{\lambda}^{(t)}, \mathbf{X}\right)$. This allows the sampling to take place one task at a time. (In the case of hierarchical models, the additional parameters $\boldsymbol{\xi}$ usually restore the conditional dependence again supporting working one task at a time.)

When sampling the parameters of the Proficiency Model, the third line of Eq. 9.24, again the independence conditions simplify the required work. A direct consequence of the global parameter independence condition is that

[10] If the πs are given by some parameterization such as DiBello–Samejima models, then the parameters β of those models also appear in the MCMC chains, using exactly the same principles: e.g., drawing from their full conditionals in turn with the other variables in the full model.

$\boldsymbol{\lambda}$ is independent of $\boldsymbol{\pi}$ given $\boldsymbol{\theta}$. Thus the third line of Eq. 9.24 simplifies to $p\left(\lambda_k \mid \boldsymbol{\theta}^{(t+1)}, \boldsymbol{\pi}^{(t+1)}, \boldsymbol{\lambda}_{<k}^{(t+1)}, \boldsymbol{\lambda}_{>k}^{(t)}, \mathbf{X}\right) = p\left(\lambda_k \mid \boldsymbol{\theta}^{(t+1)}, \boldsymbol{\lambda}_{<k}^{(t+1)}, \boldsymbol{\lambda}_{>k}^{(t)}\right)$, and often the dependence on $\boldsymbol{\lambda}$ values from other distribution vanishes as well.

Thus, a complete Gibbs cycle consists of three phases:

1. Draw proficiency variable values, $\boldsymbol{\theta}$, for each student.
2. Draw link model (evidence model) parameters, $\boldsymbol{\pi}_j$ for each task.
3. Draw proficiency model parameters, $\boldsymbol{\lambda}$.

(If there are any missing task responses, these can be filled in by another Gibbs sampling step between Step 1 and Step 2.) When all values for all unknown variables and parameters have been drawn, a new cycle starts drawing new proficiency variables. This process continues until a Monte Carlo sample of sufficient size has been drawn. If multiple chains are run, then the sample is the pooled data from all chains after burn-in. We will take a closer look at just how this works in the running latent class example.

Example 9.3 (Initial Calibration Using Gibbs Sampling, Example 9.1 Continued). *The full conditionals for individual examinee's class memberships shown in the first line of Eq. 9.24 take simpler forms because each examinee's $\boldsymbol{\theta}$ is conditionally independent of the other examinees' responses and $\boldsymbol{\theta}$s:*

$$p\left(\theta_i \mid \boldsymbol{\theta}_{<i}^{(t+1)}, \boldsymbol{\theta}_{>i}^{(t)}, \boldsymbol{\pi}^{(t)}, \lambda^{(t)}, \mathbf{X}\right) = p\left(\theta_i \mid \boldsymbol{\pi}^{(t)}, \lambda^{(t)}, x_i\right). \tag{9.25}$$

In particular, conditional probabilities of examinees' class memberships are calculated just as they were under the EM algorithm in Eqs. 9.15–9.19. In the EM algorithm, however, one accumulates over examinees the posterior probabilities of being a master, say, in order to obtain the expected count of masters. In Gibbs sampling, one draws a value—0 or 1, `nonmaster` or `master`—from the same posterior distribution, examinee by examinee. If there were a 100 examinees with identical response patterns that gave a .70 probability that $\theta = 1$ in a given Gibbs cycle, then for perhaps 60–80 of these examinees, a draw of 1 would be taken as their θ in the next cycle and for the rest a draw of 0 would be passed on.

In each cycle of the EM algorithm, expected counts of masters and nonmasters, and of item response counts among masters and nonmasters, were substituted into the expressions for the posterior distributions of the structural parameters λ and $\boldsymbol{\pi}$. In a Gibbs cycle, counts based on the draws $\theta^{(t)}$ are instead used to obtain facsimiles of the same posteriors. Cycle t counts of masters and nonmasters are counts of corresponding draws from Eq. 9.25 examinee by examinee:

$$r_1^{(t)} = \sum_i \theta_i^{(t)} \quad \textit{and} \quad r_0^{(t)} = \sum_i \left(1 - \theta_i^{(t)}\right) .$$

The cycle t counts of correct responses to item 1 from masters and nonmasters are calculated in a similar manner, with the summation over only those examinees who answered correctly:

$$r_{111}^{(t)} = \sum_{i:x_{i1}=1} \theta_i^{(t)} \quad \textit{and} \quad r_{101}^{(t)} = \sum_{i:x_{i1}=1} \left(1 - \theta_i^{(t)}\right) .$$

Note that in these sums, a given examinee's response is always the same from one cycle to the next, but whether the examinee is accumulated along with masters or nonmasters will generally vary from cycle to cycle. In the long run, the examinee will be accumulated with masters in proportion to the correct marginal posterior probability. Table 9.9 gives the facsimile of the complete data counts based on draws for θ at every 1000 MCMC cycles from 1000 to 6000.

Table 9.9 MCMC cycle response pattern counts

Proficiencies	Observables			Expected count at cycle					
θ	X_1	X_2	X_3	1000	2000	3000	4000	5000	6000
0	0	0	0	6	5	6	6	6	5
0	0	0	1	9	8	7	6	6	8
0	0	0	2	1	0	2	0	5	1
0	0	1	0	8	6	8	6	7	7
0	0	1	1	3	4	1	4	3	5
0	0	1	2	1	0	2	0	5	2
0	1	0	0	6	2	5	2	4	2
0	1	0	1	1	1	1	1	1	0
0	1	0	2	2	0	0	0	7	1
0	1	1	0	5	1	5	1	3	2
0	1	1	1	2	1	0	0	0	0
0	1	1	2	0	0	1	0	0	0
1	0	0	0	0	1	0	0	0	1
1	0	0	1	0	1	2	3	3	1
1	0	0	2	4	5	3	5	0	4
1	0	1	0	0	2	0	2	1	1
1	0	1	1	3	2	5	2	3	1
1	0	1	2	14	15	13	15	10	13
1	1	0	0	0	4	1	4	2	4
1	1	0	1	1	1	1	1	1	2
1	1	0	2	8	10	10	10	3	9
1	1	1	0	0	4	0	4	2	3
1	1	1	1	5	6	7	7	7	7
1	1	1	2	21	21	20	21	21	21

The cycle t full conditional for the population proportion of masters, λ, is a facsimile of the compete data posterior. Using the same Beta$(3,3)$ prior for λ yields

$$\begin{aligned} p\left(\lambda \mid \boldsymbol{\theta}^{(t)}, \boldsymbol{\pi}^{(t)}, \mathbf{X}\right) &= p\left(\lambda \mid \boldsymbol{\theta}^{(t)}\right) \\ &= \mathit{Beta}\left(3 + r_0^{(t)}, 3 + r_1^{(t)}\right). \end{aligned} \tag{9.26}$$

A draw from Eq. 9.26 will be the value for cycle t, $\lambda^{(t)}$. This draw is used in two ways. It is used as one data point in the empirical approximation of the posterior of λ, and is used in turn to compute full conditionals for the proficiency variables, θ_i, in the next cycle.

Similarly, for the conditional probabilities of correct response for masters and nonmasters for task 1, using again the same prior distributions as in the EM solution,

$$\begin{aligned} p\left(\pi_{111} \mid \boldsymbol{\theta}^{(t)}, \lambda^{(t)}, \boldsymbol{\pi}^{(t)}, \mathbf{X}\right) &= p\left(\pi_{111} \mid \boldsymbol{\theta}^{(t)}, \mathbf{X}\right) \\ &= \mathit{Beta}\left(4 + r_{110}^{(t)}, 2 + r_{111}^{(t)}\right) \end{aligned}$$

and

$$\begin{aligned} p\left(\pi_{101} \mid \boldsymbol{\theta}^{(t)}, \lambda^{(t)}, \boldsymbol{\pi}^{(t)}, \mathbf{X}\right) &= p\left(\pi_{101} \mid \boldsymbol{\theta}^{(t)}, \mathbf{X}\right) \\ &= \mathit{Beta}\left(2 + r_{100}^{(t)}, 4 + r_{101}^{(t)}\right). \end{aligned}$$

Values are drawn for each of the $\boldsymbol{\pi}$s as $\pi^{(t+1)}$s.

Because the model only has hyper-Dirichlet distributions, the Gibbs sampler works with this model (more complex parameterizations of the conditional probability tables often require a Metropolis algorithm). A typical procedure is to run three chains of length 6000 each (planning on discarding the first 1000 observations from each chain) starting from three different starting points: one at the prior median or mean, one in the upper tail of the prior, and one in the lower tail of the prior. The choice of starting points is arbitrary and three sets of random draws could be used instead (or three sections of a very long run could be compared).

The first task is to assess whether or not these 18,000 cycles represent an adequate draw from the posterior, and how many of them should be discarded as burn-in. Taking the initial 1000 cycles from each chain as burn-in we calculate the Gelman–Rubin R for each parameter in the model. As these parameters are all probabilities, we can apply a logistic transformation to make them more "normal" before calculating the Gelman-Rubin R values. The maximum value across all parameters is 1.09, so this looks fairly settled. The maximum autocorrelation at lag 5 is 0.7, which is pretty high, but the smallest effective sample size is 444, so this sample is reasonable. If the sample were not adequate, one recourse would be to run the chains out for a longer time. However, often running longer only helps a little bit, and alternative parameterizations of the model should be considered.

Table 9.10 shows the values of parameters from cycles at intervals of 1000 from the first chains. The table also shows some of the summary statistics for the combined data set provided by the `coda` *package (Best et al. 1996). WinBUGS provides similar summary statistics. The mean, standard deviation, and 2.5- and 97.5-percentiles are just the corresponding statistics of the data set combining draws from all three chains. The generating values, the complete data solution, and the EM estimates are given for comparison. Table 9.10 also shows different estimates of the standard error of the mean for this distribution. The "naïve SE" shows what the standard error would be if all of the samples were independent, in this case if we had a pure Monte Carlo sample of size* $N = 10,000$. *The "time-series SE" corrects the standard error for the autocorrelation of the Markov chain. Time-series SE would be equivalent to the naïve SE if the draws from the posterior were independent, but the greater the autocorrelation within a chain, the more time-series SE exceeds the naïve SE (Heidelberger and Welch 1981; Best et al. 1996).*

Table 9.10 MCMC parameter draws from intervals of 1000 and summary statistics

Cycle	λ	π_{10}	π_{11}	π_{20}	π_{21}	π_{300}	π_{301}	π_{302}	π_{310}	π_{311}	π_{312}
1000	0.58	0.39	0.62	0.52	0.73	0.59	0.32	0.09	0.02	0.22	0.77
2000	0.74	0.16	0.64	0.43	0.63	0.47	0.51	0.01	0.20	0.14	0.66
3000	0.51	0.27	0.70	0.54	0.71	0.33	0.35	0.32	0.03	0.22	0.75
4000	0.61	0.15	0.54	0.19	0.63	0.60	0.37	0.02	0.15	0.16	0.68
5000	0.55	0.40	0.77	0.35	0.87	0.46	0.16	0.38	0.12	0.37	0.51
6000	0.65	0.15	0.58	0.53	0.69	0.47	0.41	0.11	0.20	0.21	0.59
Mean	0.633	0.308	0.629	0.399	0.724	0.462	0.377	0.160	0.136	0.177	0.687
SD	0.123	0.110	0.084	0.123	0.073	0.122	0.117	0.112	0.073	0.067	0.092
Naive SE	0.001	0.001	0.001	0.001	0.001	0.001	0.001	0.001	0.001	0.001	0.001
Time-series SE	0.005	0.003	0.002	0.003	0.002	0.003	0.003	0.004	0.002	0.001	0.003
% "missing"	0.95	0.85	0.87	0.87	0.79	0.85	0.84	0.91	0.91	0.79	0.89
2.5 %	0.37	0.09	0.48	0.14	0.58	0.24	0.16	0.01	0.01	0.05	0.52
50 %	0.64	0.31	0.62	0.41	0.72	0.46	0.37	0.14	0.13	0.17	0.68
97.5 %	0.85	0.52	0.82	0.62	0.87	0.72	0.62	0.41	0.28	0.32	0.87
EM MAP	0.667	0.297	0.618	0.404	0.717	0.573	0.427	0	0.093	0.153	0.754
Complete Bayes	0.745	0.067	0.670	0.200	0.756	0.433	0.367	0.200	0.195	0.207	0.598
Complete MLE	0.760	0	0.671	0.167	0.763	0.417	0.375	0.208	0.197	0.197	0.605
"True"	0.700	0.100	0.800	0.300	0.600	0.500	0.300	0.200	0.100	0.200	0.700

Figure 9.14 shows smoothed empirical approximations of the posteriors of selected parameters. This density plot includes small tick marks along the bottom of each graph showing the individual sample points. For most variables, the posteriors are roughly normal. This accounts for the similarity between the posterior means from MCMC and the Bayes modal estimates, or MAPs, from the EM solution.

The outlier is π_{302}, with a highly skewed posterior that peaks near zero. The MAP is therefore near zero, although there is considerable probability for values above zero. This fact is reflected in the posterior mean of .160, and a 95 % posterior credibility interval of .01–.41. An advantage of the MCMC solution is that the particular shape and spread of uncertainty about this parameter is taken into account in the posteriors for the other other variables, and in particular for the θs. No single point estimate can tell the whole story, but if we were to use point estimates of the structural parameters from the initial calibration in a Bayes net to make inferences about new students, the posterior means (or, for that matter, posterior medians) are preferable to the MAPs.

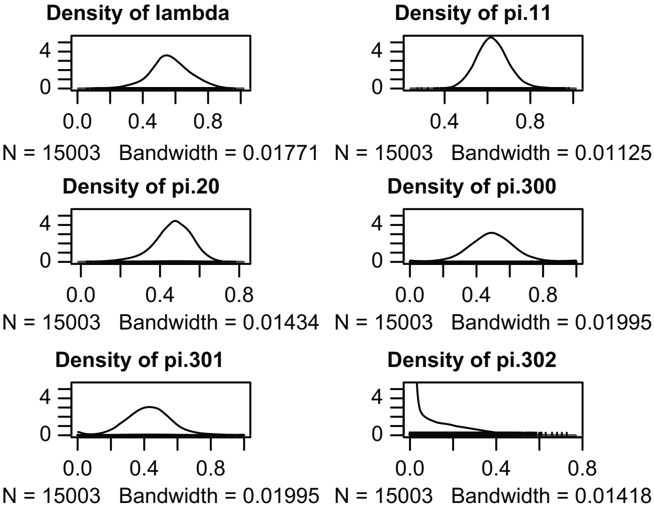


Fig. 9.14 Posteriors for selected parameters
Reprinted with permission from ETS.

9.6.2 Online Calibration

At the beginning of an operational assessment program, one obtains responses from a sample of examinees to estimate the parameters of the population of examinees and a start-up set of tasks. The methods described in the preceding section apply. At a later date one wants to calibrate new tasks into the item pool. Let us use subscripts *old* and *new* to refer to data and parameters from the initial calibration and data and parameters for the new items. The inferential targets of the initial calibration were thus $\boldsymbol{\pi}_{old}$ and $\boldsymbol{\lambda}_{old}$, and the relevant posterior distribution was $p\left(\boldsymbol{\pi}_{old}, \boldsymbol{\lambda}_{old} \mid \mathbf{X}_{old}\right)$. In online calibration, a new sample of examinees is administered both some old items and new items, with the resulting responses denoted $\mathbf{X}_{new}$. The inferential targets are $\boldsymbol{\pi}_{new}$ and $\boldsymbol{\lambda}_{new}$ (we do not need to assume that the distribution in the new examinees is the same as in the old sample as long as we have some old items in common).

Formally, the correct Bayesian model for online calibration is

$$p\left(\boldsymbol{\pi}_{new}, \boldsymbol{\pi}_{old}, \lambda_{new}, \boldsymbol{\lambda}_{old} \mid \mathbf{X}_{new}, \mathbf{X}_{old}\right) \ . \tag{9.27}$$

As a first pass, one might try to fix the parameters $\boldsymbol{\pi}_{old}$ at the point estimate from the initial calibration, to simplify the problem and reduce the number of parameters to be estimated in Eq. 9.27. However, unless the initial calibration is sufficiently large, small errors in the estimation will accumulate and cause the scale to drift over time (Mislevy et al. 1994; Mislevy et al. 1999a).

What we can do instead is to substitute $p_j(\boldsymbol{\beta}_j|\boldsymbol{\xi}, \mathbf{x}_{old})$ for $p_j(\boldsymbol{\beta}_j|\boldsymbol{\xi})$ in Eq. 9.1. Then the output of the calibration will be consistent with the combined data. There is only one catch: the posterior $p_j(\boldsymbol{\beta}_j|\boldsymbol{\xi}, \mathbf{x}_{old})$ does not necessarily have a convenient function form (e.g., beta distribution). In such cases, one can approximate the posterior with something with a convenient functional form.

When the prior law takes the form of a beta distribution, a method of moments approximation is convenient. To do this, one finds a beta distribution with the same mean and variance as the posterior and uses that as the new prior. For example, in Example 9.3, the posterior mean and standard deviation for π_{11} are 0.629 and 0.084. Let $p = a/(a+b) = E[\pi]$ and substitute this into the formula for the variance of the beta distribution, we get $\mathrm{Var}(\pi) = p(1-p)/(a+b+1)$. Solving this for $a+b$, we get $n = a+b = 33.072$; this is the effective sample size of the posterior. We can now get the beta parameter $a = np = 20.803$ and $b = n(1-p) = 12.270$. Dirichlet parameter can be treated like a collection of beta distributions and the resulting as can be summed to produce a single effective sample size for the entire law. Table 9.11 gives the results for $\boldsymbol{\pi}_{31}$.

Table 9.11 Approximating Dirichlet priors from posterior means and standard deviations for $\boldsymbol{\pi}_{31}$

	π_{310}	π_{311}	π_{312}
Mean	0.136	0.177	0.687
SD	0.073	0.067	0.092
Variance	0.0053	0.0045	0.0085
a_k	2.863	5.567	16.766
$\sum a_k$	25.196	25.196	25.196

Example 9.4 (Online Calibration Using Gibbs Sampling, Example 9.3 Continued). *Consider a new sample of 200 (simulated) examinees who are administered tasks 1 and 3 from the running example and a new dichotomous item, task 4. The generating values of the old items 1 and 3 are as before:* $\pi_{10} = .1$, $\pi_{11} = .8$, $\boldsymbol{\pi}_{30} = (.5, .3, .2)$, *and* $\boldsymbol{\pi}_{31} = (.1, .2, .7)$. *The new generating parameters are* $\pi_{40} = .1$ *and* $\pi_{41} = .8$ *for item 4 and* $\lambda_{new} = .6$. *The resulting data are shown in Table 9.12.*

Table 9.12 Response pattern counts for online calibration

			Response
Item 1	Item 3	Item 4	Pattern count
0	0	0	22
0	0	1	17
0	1	0	12
0	1	1	18
0	2	0	11
0	2	1	29
1	0	0	4
1	0	1	6
1	1	0	4
1	1	1	19
1	2	0	12
1	2	1	46

As in the initial calibration, we let the prior distributions for π_{40} *and* π_{41} *be* Beta$(2, 4)$ *and* Beta$(4, 2)$ *respectively, reflecting a mild prior expectation that masters are more likely to answer an item correctly, and nonmasters are more likely to answer incorrectly. For the population proportion of masters,* λ_{new}, *we will use a slightly different prior than in the initial calibration.* Beta$(3, 3)$ *was used there, reflecting a mild expectation that masters and nonmasters would be equally likely. But the posterior mean of* λ_{old} *was .633 and the posterior standard deviation was 0.123. Using the method of moments reveals that a* Beta$(9.72, 5.64)$ *distribution has the same mean and variance. We will use this as a prior for* λ_{new} *(if we thought the population might have shifted, we could downweight the prior by multiplying the parameters by a constant less than one). We apply similar calculations to come up with new beta and Dirichlet prior distributions for* $\boldsymbol{\pi}_1$ *and* $\boldsymbol{\pi}_3$.

The MCMC calibration then proceeds in a similar manner. Again we run three chains of length 6000 and discard the first 1000 observations from each chain as burn-in. Again the Gelman–Rubin statistic and plot shows that the chains have likely reached the stationary distribution by cycle 500, so that the burn-in of 1000 is likely to be conservative.

Table 9.13 Average parameter values from initial and Online calibrations

	λ	π_{10}	π_{11}	π_{300}	π_{301}	π_{302}	π_{310}	π_{311}	π_{312}	π_{40}	π_{41}
Initial mean	0.633	0.308	0.629	0.462	0.377	0.160	0.136	0.177	0.687	–	–
Initial SD	0.123	0.110	0.084	0.122	0.117	0.112	0.073	0.067	0.092	–	–
Online mean	0.649	0.183	0.621	0.567	0.321	0.112	0.073	0.240	0.687	0.410	0.806
Online SD	0.060	0.054	0.047	0.078	0.067	0.064	0.032	0.043	0.050	0.080	0.044
"True"	0.700	0.100	0.800	0.500	0.300	0.200	0.100	0.200	0.700	0.100	0.800

Table 9.13 shows the posterior means from both the initial and subsequent online calibration. For many of the variables that are common across the initial and online calibrations, the online values are closer to the "true" values which were used to simulate the data. This is not only because the online sample is both larger in and of itself (200 as opposed to 100 simulees) but also because the priors for the online calibration include information from the initial calibration. Thus, as more data become available, we can improve the estimation of the various parameters.

Even though the sample size was bigger than Example 9.3, the calibration still does not exactly reproduce the simulation parameters. Two factors are at work here. The first factor is that 200 students is still a relatively small sample. IRT calibrations with the two- and three-parameter logistic models aim for 1000 examinees. The second and perhaps more important factor is that 3 items is an extremely short test. Any individual's proficiency is likely to be estimated quite poorly and varies from chain to chain. Increasing the test length and increasing the sample size are both likely to produce more accurate estimates (although they may slow down the speed at which the Markov chains mix).

9.7 Caution: MCMC and EM are Dangerous!

The EM algorithm (Dempster et al. 1977) was one of the earliest developments in the field of Bayesian computation. Using EM, the posterior mode and variance could be calculated for latent variable models of arbitrary complexity. All that was needed was a complete Bayesian model. The arrival of the Gibbs Sampler (Geman and Geman 1984) and the increasing availability of cheap computer power brought about an explosion in the field. Now Bayesian techniques could be applied to a wide variety of problems (Gilks et al. 1996). All one needed to do was to write down the full Bayesian model (prior as well as likelihood), and some form of MCMC could approximate any statistic of the posterior, not just the mode.

The BUGS program(Thomas et al. 1992) and its successors, WinBUGS (Lunn et al. 2000), OpenBUGS (Lunn et al. 2009), and JAGS (Plummer 2012), have made this new computational power easily accessible, without requiring a great deal of time to develop, code, and test algorithms for particular models. With BUGS, the analyst only needs to specify the model in a language based on the S statistical language (Chambers 2004). BUGS then figures out how to set up a Gibbs sampler, and whether it can calculate the full conditional distributions analytically or whether it needs the Metropolis algorithm. In the latter case, it even automatically tunes the proposal distribution.

This means almost any Bayesian model can be fit to any data set. There is no requirement that the chosen model is sensible. If the data provide no information about a parameter then the parameter's prior and posterior law will be nearly identical. The displays in WinBUGS are designed to help one assess convergence, but they do not always help with the issue of whether or not the model is appropriate.

For this reason, the BUGS manual (Spiegelhalter et al. 1995) bears the warning, "Gibbs sampling is dangerous" on the first page. The warning is not so much about Gibbs sampling as it is to leaping into computation without thinking about whether or not the model is appropriate for the data and problem at hand. To that extent, the warning is equally applicable to blindly applying the EM algorithm. Although both procedures will provide an answer to the question, "What are the parameters of this model?" they do not necessarily answer the question, "Is this a reasonable model?"

Fortunately, Bayesian statistics offers an answer here as well. If we have a full Bayesian model, that model makes a prediction about the data we might see. If this model has a very low probability of generating a given data set, it is an indication that the model may not be appropriate. We can also use this idea to choose between two competing models, or search for the best possible model. The next chapter explores model checking in some detail.

Exercises

9.1 (Stratified Sampling). Example 9.1 used a simple random sample of 100 students with the number of masters in the sample unknown in advance. Suppose instead a stratified sample of 50 masters and 50 nonmasters was used. How would the inference differ, if it was known who the masters and nonmasters are? How about if we do not know who is who, but we know there are exactly 50 of each?

9.2 (Missing At Random). Classify the following situations as MCAR, MAR, or neither:

1. A survey of high school seniors asks the school administrator to provide grade point average and college entrance exam scores. College entrance exam scores are missing for students who have not taken the test.
2. Same survey (mentioned in first point) except now survey additionally asks whether or not student has declared an intent to apply for college.

3. To reduce the burden on the students filling out the survey, the background questions are divided into several sections, and each student is assigned only some of the sections using a spiral pattern. Responses on the unassigned section are missing.
4. Some students when asked their race decline to answer.

9.3 (Missing At Random and Item Responses). Item responses can be missing for a variety of reasons. Classify the following situations concerning a student's missing response to a particular Task j as MCAR, MAR, or neither. Hint: See (Mislevy 2015) or (Mislevy and Wu 1996).

1. John did not answer Task j because it was not on the test form he was administered.
2. Diwakar did not answer Task j because there are linked harder and easier test forms, intended for fourth and sixth grade students; Task j is an easy item that only appears on the fourth grade form; and Diwakar is in sixth grade, so he was administered the hard form.
3. Rodrigo took an adaptive test. He did well, the items he was administered tended to be harder as he went along, and Task j was not selected to administer because his responses suggested it was too easy to provide much information about his proficiency.
4. Task j was near the end of the test, and Ting did not answer it because she ran out of time.
5. Shahrukh looked at Task j and decided not to answer it because she thought she would probably not do well in it.
6. Howard was instructed to examine four items and choose two of them to answer. Task j was one of the four, and not one that Howard chose.

9.4 (Classical Test Theory). Consider the following simple model from classical test theory. Let T_i be a student's true score on a family of parallel tests on which scores can range from 0–10. T_i characterizes Student i's proficiency in the domain, but cannot be observed directly. Instead, we observe noisy scores on administrations of the parallel tests. Let X_{ij} be Student i's score on Test j. Following classical test theory, let

$$X_{ij} = T_i + \epsilon_{ij} \tag{9.28}$$

where $\epsilon_i \sim \mathrm{N}(0, \sigma_\epsilon^2)$ and $T_i \sim \mathrm{N}(\mu, \sigma_T^2)$. The classical test theory index of reliability is $\rho = \frac{\sigma_T^2}{\sigma_T^2 + \sigma_\epsilon^2}$. The following is BUGS code for estimating μ, σ_T^2, σ_ϵ^2, and ρ from the responses of nine students to five parallel test forms:

```
model ctt {
  for (i in 1:N) {
    T[i] ~ dnorm(mu,tauT)
    for (j in 1:ntest) {
      x[i,j] ~ dnorm(T[i],taue);
    }
  }
  mu ~ dnorm(0,.01)
  tauT ~ dgamma(.5,1)
  taue ~ dgamma(.5,1)

  rho <- taue / (taue + tauT)
  varT <- 1/tauT
  varE <- 1 / taue
}

#inits
list(T = c(20,20,20,20,20,20,20,20,20))
list(T = c(-20,-20,-20,-20,-20,-20,-20,-20,-20))

#data
list(N=9,ntest=5,
x=structure(.Data=c(
  2,      3,      2,      5,      3,
  4,      3,      4,      3,      6,
  6,      4,      3,      4,      3,
  4,      7,      5,      4,      5,
  7,      5,      4,      5,      4,
  4,      5,      4,      7,      5,
  6,      5,      6,      5,      8,
  5,      6,      5,      8,      6,
  7,      6,      7,      6,      9 ), .Dim=c(9,5)))
```

Note that in BUGS, the normal distribution is parameterized with the mean, μ, and precision, $\tau = 1/\sigma^2$. The line `varT <- 1/tauT` produces draws for σ_T^2. Run the problem with this setup, and the two chains as initial values for the Ts. Monitor `T`, `mu`, `varT`, `varE`, and `rho`.

1. Run 500 MCMC cycles. There are overdispersed initial values for the Ts. Ask for Stats, history, density, and the BGR convergence diagnostics plot. Does it look like burn-in cycles may be needed for this problem? Which parameters seem to be more or less affected by the overdispersed initial values?
2. Run another 500 cycles, and calculate summary statistics and distributions for the parameters you monitored based on only the last 500 cycles

(hint: beg = 501 on the sample monitor dialog box). Regarding estimates for individual students: What are the posterior means for the Ts (i.e., true scores) of each of the students? What are their maximum likelihood estimates (hint: BUGS does not tell you this—you need to do a little arithmetic). In what directions do they differ, and why?

3. Look at the summary statistics for `T[1]`. What is the meaning of the number in each column?

9.5 (Classical Test Theory and the Effects of Different Priors). Consider the model and data from Exercise 9.4. Start with the original setup. Run just one chain for each of the variations required. In each case, run 2000 cycles, and calculate statistics based on only the last 1000. Monitor `mu`, `varT`, `varE`, `T`, and `rho`.

1. The original setup uses, as a prior distribution for `mu`, $N(0, .01)$ (using the BUGS convention with the precision as the shape parameter). Run the same problem, except with $N(0, 1)$ as the prior for `mu`, and monitor the results. Focusing on posterior means and standard deviations of the parameters listed above, which ones change? How much and why?
2. Repeat the run, except with $N(0, 10)$ as the prior for `mu`. Again focusing on posterior means and standard deviations of the parameters listed above, which ones change? How much and why?
3. The original setup uses, as a prior distribution for both `tauT` and `taue`, $\text{Gamma}(.5, 1)$. Run the same problem, except with $\text{Gamma}(.05, .10)$ as the prior for `tauT`, and monitor the results. Focusing on posterior means and standard deviations of the parameters listed above, which ones change? How much and why?
4. Repeat the run, except with $\text{Gamma}(50, 100)$ as the prior for `tau`. Again focusing on posterior means and standard deviations of the parameters listed above, which ones change? How much and why?

9.6 (Slow Mixing). A researcher runs a small pilot MCMC chain on a particular problem and gets a trace graph for one parameter like Fig. 9.9. The researcher shows it to two colleagues. The first says that the problem is the proposal distribution and suggests that the researcher should try to find for new proposal distributions. The second says that if the researcher just runs the chain for ten times longer than originally planned, then the trace plot will look like “white noise” and the MCMC sample will be adequate. The second colleague further suggests that a long weekend is coming up and the lab computers will be idle anyway. Which advice should the researcher take? Why?

9.7 (MAP and MCMC Mean). In Table 9.10, the MCMC mean frequently appears to be closer to the MAP from the EM algorithm than it does to the “true” value from the data generation. Why is this seen?

9.8 (Latent Class). The following response vectors come from a simple latent class model with two classes and ten dichotomously scored items.

```
1111111010 1110111011 0000010000 1111001011 0010001000
0000101000 0000010000 0010100000 0000000000 0000001000
0011100111 0100000000 0000000000 0000000000 0010000000
1100000000 1111011111 0010100000 0010000010 0111111011
0010000000 0000000000 1011111111 0010000100 1111111011
1110110111 0000000000 0000110010 1111010100 0000000010
0100100000 0111100010 0000000000 0011000000 1111111010
0100100000 0000001100 1111111111 1100000000 0010010000
0000101000 0101011111 0000000000 1111111011 0000010000
0010000000 0000010000 1111110111 0010000100 0010000000
0010010010 1111011111 1110101110 1111011101 0000000000
1111111110 1100010010 0001000001 1111111101 0011111011
0000000100 0010000001 0010000000 0010100000 0111111111
0000000000 0001110000 0000010010 1111011111 0010010000
1110101001 0010110110 1110000000 0000100101 0010100010
0000000110 0000000000 0000100010 1111001011 0000001000
0010000000 0000010001 0010011010 1111111011 1010101100
0010000001 0011000000 1000100010 0000010000 1111100011
0110000100 1001000000 1010000000 0100010110 0010111101
1101111111 0010000000 1111101111 1000100100 1111101111
```

Use a Beta$(1, 1)$ prior for the class membership probability λ and for all tasks use a Beta$(1.6, .4)$ prior for the probability of success for masters, π_{j1}, and a Beta$(.4, 1.6)$ probability of success for nonmasters, π_{j0}. Estimate the parameters from the data using MCMC.

9.9 (Latent Class Prior). In Exercise 9.8, what would have happened if we had used a Beta$(1, 1)$ prior for π_{j1} and π_{j0}?

Hint: Consider three MCMC chains starting from the starting points: $\boldsymbol{\pi}_j = (.2, .8)$, $\boldsymbol{\pi}_j = (.5, .5)$, and $\boldsymbol{\pi}_j = (.8, .2)$ for all j.

9.10 (Latent Class Parameter Recovery). The data for Exercise 9.8 are from a simulation, and the parameters used in the simulation are: $\lambda = 0.379$, plus the values in the following table.

Item	1	2	3	4	5	6	7	8	9	10
Nonmasters	0.22	0.17	0.31	0.07	0.25	0.19	0.15	0.18	0.22	0.11
Masters	0.82	0.84	0.85	0.72	0.81	0.76	0.8	0.7	0.84	0.76

Calculate a 95 % credibility interval for each parameter (this can be done by taking the 0.025 and 0.975 quantiles of the MCMC sample). How many of the credibility intervals cover the data generation parameters? How many do we expect will cover the data generation parameters?

9.11 (EM vs. MCMC). In each of the following situations, tell whether it is better to use the EM algorithm or MCMC to estimate parameters.

1. The posterior mean will be used in an online scoring engine. The posterior variance will be examined briefly as a model checking procedure, but will not be used in scoring.
2. The test specs call for only using items whose *p-plus* — marginal probability of success, $p_j^+ = \mathrm{P}(X_j = 1)$ — is greater than .1 and less than .9 with 90 % credibility, that is $\mathrm{P}(0.1 \leq p_j^+ \leq 0.9) \geq 0.9$.
3. Only the posterior mean will be used in online scoring, but there is strong suspicion that the distribution for the difficulty on several item parameters is bimodal.

9.12 (Improving Posterior Standard Deviation). Consider the calibration in Example 9.3. Which of the following activities are likely to reduce the standard deviation of the posterior law for the proficiency model parameter λ:

1. Increase the size of the MCMC sample.
2. Increase the length (number of tasks) of the test.
3. Increase the number of students in the calibration sample.

Which of the following activities are likely to reduce the standard deviation of the posterior for an evidence model parameter such as π_{10}:

1. Increase the size of the MCMC sample.
2. Increase the length (number of tasks) of the test.
3. Increase the number of students in the calibration sample.

9.13 (LSAT model). The BUGS distribution package (Spiegelhalter et al. n.d.) comes with a sample model called "LSAT" based on an analysis performed by Bock and Aitkin (1981) of responses on five items from 1000 students taking the Law School Admissions Test (LSAT). The data are analyzed using the Rasch model, where if p_{ij} is the probability that Student i gets item j correct, then

$$\mathrm{logit}(p_{ij}) = \theta_i - \alpha_j \tag{9.29}$$

where the proficiency variable θ_i has distribution $N(0, \tau)$ (Spiegelhalter et al. n.d.). Note that this equation can be reparameterized as:

$$\mathrm{logit}(p_{ij}) = \beta\theta_i - \alpha_j \tag{9.30}$$

where $\theta_i \sim N(0, 1)$ and $\beta = \sqrt{1/\tau}$.

Run an MCMC sampler using both the original (Eq. 9.29) and reparameterized (Eq. 9.29) models. What differences are there in the resulting Markov Chains?

10

Critiquing and Learning Model Structure

The previous chapter described how to fit a model to data. The parameter-learning methods described there assumed the structure of the model was fixed. However, often there is as much or more uncertainty about the structure of the model as there is about the values of the parameters. There are basically two approaches to this problem. The first is model checking, or as it is sometimes called, model criticism. Fit indices and graphical displays can help us explore where and how well the model fits the data, and bring to light problems with a model. The second is model search. There are a number of ways to search the model space for one that is "best" in some sense.

While traditional methods of characterizing model fit emphasized overall goodness of fit, we take a more utilitarian perspective. The statistician George Box famously said "All models are false, but some are useful" (Box 1976). We want a Bayes net that captures the key interrelationships between what students know, in terms of proficiency variables, and what they can do, in terms of observables, without having to believe that the model expresses every pattern in the data. We do not, however, want unmodeled patterns that make our inferences about students' proficiencies misleading for the purpose at hand. We are interested in fit indices that highlight particular kinds of model misfit which, from experience, we know can appear in assessment data and distort our uses of the model.

We emphasize exploratory uses of model checking over statistical tests of fit, partly because Bayes nets are often applied with small to medium size data sets, and partly because the techniques we describe fall out almost as a by-product of Markov Chain Monte Carlo (MCMC) estimation, in ways that generate their own reference distributions. The reader interested in large-sample distributions of prediction-based fit indices is referred to Gilula and Haberman (1995), Gilula and Haberman (2001), and to Haberman et al. (2013) for an application to item response theory (IRT) models. Although we do not pursue large sample properties here, the chapter draws in places on their work on prediction-based indices.

R. G. Almond et al., *Bayesian Networks in Educational Assessment*,
Statistics for Social and Behavioral Sciences, DOI 10.1007/978-1-4939-2125-6_10

Section 10.1 introduces some fit indices and describes a simple simulation experiment for using them. Section 10.2 looks at the technique of posterior predictive model checking (PPMC), which goes well with MCMC estimation. Section 10.3 looks at some graphical methods for assessing model fit. Section 10.4 addresses differential task functioning, where the issue is whether a task works similarly across student groups. Section 10.5 then turns to model comparison. Usually simpler models are preferable to more complex ones, but complex ones will fit better just because of the extra parameters. The DIC fit measure discussed in Sect. 10.5.1, which also goes well with MCMC, includes a penalty for model complexity. In Sect. 10.5.2, prediction-based indices are defined and illustrated with the discrete-IRT testlet model. Looking ahead, Chap. 11 will apply several of these techniques to the mixed-number subtraction example.

Given a measure of model fit, one can search for a model that fits the data best. Section 10.6 looks at some literature on automatic model selection. There are, however, some important limitations on learning model structure with Bayes nets. In particular, there can be ways of reversing the direction of some of the edges that have different interpretations but do not change the implied probability distribution. Section 10.7 discusses some of these equivalent models, and highlights pitfalls in attempting to learn "causality" from data.

10.1 Fit Indices Based on Prediction Accuracy

The fact that Bayes nets are probability models gives them a distinct advantage over more ad hoc mathematical models for managing uncertainty, such as fuzzy logic (Zadeh 1965) and certainty factors (Shortliffe and Buchanan 1975). In particular, the probabilities can be regarded as predictions and standard statistical techniques can assess how accurate those predictions are. This assessment yields information about how well the model fits the data.

Cowell et al. (1993) describe several locations in a Bayesian network at which fit can be assessed:

Node Fit: How well the model predicts the distribution of a single variable in the model. These can either be conditional predictions taking the values of other variables into account or marginal predictions ignoring the values of other variables.

Edge Fit: How well the relationship between a parent and child in the graph is modeled.

Global Fit: How well all variables in the data fit the graphical model.

These are not always easy to apply in educational testing because the proficiency variables are latent, and therefore predictive patterns of certain dependencies described in the model cannot be directly assessed. In particular, parent–child relationships often involve at least one latent variable and hence, they cannot be directly tested with only observed data. Thus, the node fit indices can only be applied to observable variables and the global fit indices must calculate how well the model predicts all the observables.

One way to get around this problem is to leave the data out for one observable, and see how well the model predicts that observable based on the remaining values. This is called *leave one out prediction.* (The idea extends readily to leaving out a group of observables, then predicting some summary statistic of them such as a subtest score.) Suppose that a collection of observable outcomes is available from N learners taking a particular form of an assessment. Let Y_{ij} be the value of Observable j for Learner i. Assume that Y_{ij} is coded as an integer and it can take on possible values $1, \ldots, K_j$.

Using the methods of Chap. 5, it is easy to calculate a predictive distribution for Y_{ij} given any other set of data. Let $\mathbf{Y}_{i,-j}$ be the vector of responses for Learner i on every observable except Observable j. Define $p_{ijk} = \mathrm{P}(Y_{ij} = k|\mathbf{Y}_{i,-j})$. In all of the fit indices described below, the idea is to characterize how well the model's prediction p_{ijk}, given all her other responses, predicts the Y_{ij} that was actually observed. Let p_{ij*} denote the value of p_{ijk} for $k = Y_{ij}$, that is, the prediction probability of the event that actually occurred.

Williamson (2000) noted that a number of measures of quality of prediction have historically been applied to evaluate weather forecasting. These can be pressed into service fairly easily to evaluate node fit in Bayesian networks. Williamson (2000) (see also Williamson et al. 2000) evaluated a number of these and found the most useful to be *Weaver's Surprise Index*, the *Ranked Probability Score*, and *Good's Logarithmic Score.* The first two are more traditional indices, which remain useful to alert users to anomalies in collections of predictions. Good's Logarithmic Score, useful enough on its own, leads us to more theoretically grounded techniques based on statistical theory and information theory.

Weaver's Surprise Index (Weaver 1948) attempts to distinguish between a "rare" event and a "surprising" one. A rare event is one with a small probability. A surprising event is one with a small probability relative to the probability of other events. (The definition of events can make a difference: The probability of a Royal Flush in clubs—the ace, king, queen, jack, and ten of clubs—is the same as the probability of any other specified hand, so getting this particular hand in poker is rare but no more surprising than any other specified set of five cards in the deck when each is considered an event in the comparison. It is rare *and* surprising with respect to events defined by sets of hands with the same poker value, such as one pair, two pairs, straight, etc.)

Weaver's surprise index is defined as the ratio of the expected value of prediction probabilities to that of the actual event:

$$W_{ij} = \frac{E(p_{ijk})}{p_{ij*}} = \frac{\sum_{k=1}^{K_j} p_{ijk}^2}{p_{ij*}} \tag{10.1}$$

The expectation here is over Y_{ij} using the predictive probabilities p_{ijk}. The larger the value, the more surprising the result. Weaver suggests that values of 3–5 are not large, values of 10 begin to be surprising and values above 1000 are definitely surprising.

Weaver's surprise index assumes that one wrong prediction is as bad as another. However, frequently the observables in an educational model represent ordered outcomes (e.g., a letter grade assigned to an essay). In those situations, the measure of prediction quality should provide a greater penalty for predictions that are far off than for near misses. The *Ranked Probability Score* (Epstein 1969) takes this into account.

$$S_{ij} = \frac{3}{2} - \frac{1}{2(K_j-1)} \sum_{k=1}^{K_j-1} \left[\left(\sum_{n=1}^{k} p_{ijn} \right)^2 + \left(\sum_{n=k+1}^{K_j} p_{ijn} \right)^2 \right] \\ - \frac{1}{K_j-1} \sum_{k=1}^{k} |k - Y_{ij}| p_{ijk}. \tag{10.2}$$

This index ranges from 0.0 (poor prediction) to 1.0 (perfect prediction). It assumes the states have an interval scale.

Good's Logarithmic Score extends the basic *Logarithmic Scoring Rule.* Recalling that p_{ij*} is the posterior probability of the observed outcome, the basic logarithmic score is

$$L_{ij} = -\log p_{ij*}. \tag{10.3}$$

The lower the probability, the larger the value of the logarithmic score for this observation. Twice the logarithmic score is called the *deviance*, and finding parameter values that minimize the total deviance over a sample gives the maximum likelihood estimates for a model. We will return to the use of deviance in model comparisons in Sect. 10.5.1.

The basic logarithmic score makes no distinction between "rare" and "surprising." To take care of this effect, Good (1952) subtracts the expected logarithmic score, over the values that might have been observed:

$$GL_{ij} = -\log p_{ij*} - -\sum_{k=1}^{K_j} p_{ijk} \log p_{ijk}. \tag{10.4}$$

Values near zero indicate that the model accurately predicts the observation. The average logarithmic score is known in information theory as entropy, or the uncertainty about Y_{ij} before it was observed, and is denoted by $\mathrm{Ent}(Y_{ij})$.

Any of the preceding fit indices measures can be averaged across the sample of examinees to give a measure of node fit for a particular observable. For example, $S_j = \sum_{i=1}^{N} S_{ij}/N$ is the ranked probability score for Observable j. The indices can also be averaged across nodes to get a measure of person fit, e.g., $W_i = \sum_{j=1}^{J} W_{ij}/J$.

The question is how extreme indices above need to be to indicate a problem. Williamson et al. (2000) posed a simple simulation experiment to answer that question for any given model. When assessing model fit, the null hypothesis is that the data fit the model. This suggests a procedure for determining the distribution of the test statistic under the null hypothesis.

1. Generate a sample of the same size as the real data from the posited model.
2. Calculate the value of the fit indices for the simulated data set, using the posited model and conditional probabilities.
3. Repeat Steps 1 and 2 many times to generate the desired reference distribution of any of the fit indices.

Step 2 produces fit index values for all of the simulee*task combinations in the sample, e.g., S^*_{ij}. Reference distributions for task and person fit measures are created by averaging over tasks or persons accordingly, i.e., S_j is computed by averaging S_{ij}. Williamson et al. (2000) note that the reference distribution can be computed more cheaply using a simple bootstrap (Efron 1979), which samples repeatedly from the observed data.

10.2 Posterior Predictive Checks

The Williamson et al. (2000) method ignores one potentially important source of variability, the uncertainty in the predictions due to uncertainty about the parameters (it matters less for problems with large samples). The method of PPMC (Guttman 1967; Rubin 1984) does incorporate uncertainty in parameters. Furthermore, it works very naturally with Markov Chain Monte Carlo estimation. Sinharay (2006) provides a good summary of this technique applied to Bayesian network models. This section describes the approach and gives simple example.

Let $p(\mathbf{y}|\boldsymbol{\omega})$ be the likelihood for data $\mathbf{y}$ given parameters $\boldsymbol{\omega}$. Let $\mathbf{y}^{\mathrm{rep}}$ be a replicate set of data generated by the same process as $\mathbf{y}$ with the same parameters $\boldsymbol{\omega}$. This is sometimes called a shadow data set. The posterior predictive method suggests using the posterior predictive distribution for $\mathbf{y}^{\mathrm{rep}}$ to create a reference distribution for a given fit statistic, similar to the way Williamson et al. (2000) method created an empirical reference distribution in the previous section. The posterior predictive distribution is defined as:

$$p(\mathbf{y}^{\mathrm{rep}}|\mathbf{y}) = \int p(\mathbf{y}^{\mathrm{rep}}|\boldsymbol{\omega})p(\boldsymbol{\omega}|\mathbf{y})d\boldsymbol{\omega} \ . \tag{10.5}$$

Correspondingly, the posterior predictive distribution for a fit index, e.g., Weaver's surprise index for Task j, W_j, is obtained as

$$p(W_j(\mathbf{y}^{\mathrm{rep}})|\mathbf{y}) = \int p((W_j(\mathbf{y}^{\mathrm{rep}})|\boldsymbol{\omega})p(\boldsymbol{\omega}|\mathbf{y})d\boldsymbol{\omega} \ . \tag{10.6}$$

The idea is to repeatedly draw shadow data sets $\mathbf{y}^{\text{rep}}$ from a predictive distribution for data, using the posterior distribution of $\boldsymbol{\omega}$ given the observed data $\mathbf{y}$. In each such replicate, calculate the value of some statistic or index of interest. The resulting distribution is used as a reference distribution to evaluate the value calculated with $\mathbf{y}$ itself. (Exercise 10.11 is a simple problem the reader can do by hand to get a feel for PPMC.)

Although $p(\mathbf{y}^{\text{rep}}|\mathbf{y})$ can be difficult to derive analytically in more complex problems, it is actually straightforward to sample from, especially if an MCMC algorithm was produced to sample from $p(\boldsymbol{\omega}|\mathbf{y})$. In each cycle (or in selected cycles) of the MCMC loop, after the values are drawn for the parameters $\boldsymbol{\omega}$, a shadow data set $\mathbf{y}^{\text{rep}}$ is drawn.

For instance, in the running latent class example in Chap. 9, for the first two tasks the probability of a correct response to Task j by a learner in Class k is Bernoulli(π_{jk}). We can, thus, write the probability for Learner i as $\pi_{j,class(i)}$. When the model is described in the BUGS language, the corresponding line in the model code is

```
y[i,j] ∼ dbern(pi[class[i],j]).
```

In every MCMC cycle, the observed value of y_{ij} contributes to the likelihood function for π_{j0} and π_{j1} and all of the other unobserved variables in the model; in turn, values for each of them are drawn from their full conditionals as described in Chap. 9. To obtain a shadow draw for y_{ij} in each cycle, we merely need to add the line

```
yrep[i,j] ∼ dbern(pi[class[i],j]).
```

Now in every cycle, the variable y_{ij}^{rep} is part of the model. No value is observed for it so the sampler draws a value from its full conditional—which is exactly the same in form as the distribution for y_{ij}. The draw is carried out with $class(i)$ and $\pi_{j,class(i)}$ fixed at the values drawn for them this cycle. These values will vary from one cycle to the next. The posterior distribution for x_{ij}^{rep} , thus, properly takes into account uncertainty about these variables, and all the other unknown variables in the problem.

Any descriptive statistic or analysis that can be run on $\mathbf{y}$ can also be run on the shadow data set $\mathbf{y}^{\text{rep}(t)}$ from MCMC cycle t, and the results compared. Are there far too many zeros for Task j? Does a factor analysis of $\mathbf{y}$ yield factors that $\mathbf{y}^{\text{rep}(t)}$ does not? In diagnostic testing, an interesting statistic is the number right on a subscale. Multiply the outcome vector by one column of a Q-Matrix to get a number right score focused on one particular skill. This will provide a measure of how well the model predicts performance on tasks requiring that skill. The range of features to compare which might shed light on model fit and model improvement is limited only by the analyst's ingenuity. To avoid over-interpreting the results of such comparisons, they can be carried out with multiple shadow data sets, drawn from different MCMC chains or widely spaced cycles in the same chain.

In practice, much posterior predictive checking uses a *test statistic*, or *discrepancy measure*, $D(\mathbf{y}, \boldsymbol{\omega})$. For example, Yan et al. (2003) illustrate the use of Pearson residuals from each person-by-observable prediction (for other choices of residual, see Bishop et al. 1975). Let Y_{ij} be the response of Person i to Observable j. Following the notation of the previous chapter, let $\boldsymbol{\lambda}^{(t)}$ be the estimate of the proficiency model parameters at MCMC cycle t, $\boldsymbol{\beta}_j^{(t)}$ the estimate of the link model parameters at MCMC cycle t, and $\boldsymbol{\theta}_i^{(t)}$ be the estimate of the proficiency variables for Person i at cycle t. Let $p_{ij}^{(t)} = E[Y_{ij}|\boldsymbol{\lambda}^{(t)}, \boldsymbol{\beta}_j^{(t)}, \boldsymbol{\theta}_i^{(t)}]$ be the expected value for Y_{ij} given the values for parameters and imputed variables at MCMC cycle t. The squared *Pearson residual* for Person i, Observable j and cycle t is:

$$V_{ij}^{(t)} = \frac{\left(Y_{ij} - p_{ij}^{(t)}\right)^2}{p_{ij}^{(t)}(1 - p_{ij}^{(t)})} \; . \tag{10.7}$$

Taking the average of this measure over observables yields a measure of person fit and taking the average over persons yields a measure of observable fit. Taking the average over all observables for all people yields a measure of overall goodness of fit. As $V_{ij}^{(t)}$ represents a squared residual, taking the square root of the average provides a root mean squared error (RMSE).

For typical discrepancy measures like these where higher values indicate worse fit, Gelman et al. (1996) suggest comparing $D(\mathbf{y}, \boldsymbol{\omega})$ to the distribution of $D(\mathbf{y}^{\text{rep}}, \boldsymbol{\omega})$. In MCMC algorithms, one can simply count the number of MCMC cycles in which $D(\mathbf{y}^{\text{rep}}, \boldsymbol{\omega})$ is larger than $D(\mathbf{y}, \boldsymbol{\omega})$. This produces a *posterior predictive p-value* (PPP):

$$\text{PPP-value} = P\left(D(\mathbf{y}^{\text{rep}}, \boldsymbol{\omega}) \geq D(\mathbf{y}, \boldsymbol{\omega})|\mathbf{y}\right) \tag{10.8}$$

PPPs around .5 indicate that the observed discrepancies fall in the middle of the distribution of discrepancy measures from the posterior predictive distribution. This suggests adequate data-model fit with respect to whatever characteristic the measure is targeting. Values near zero (or unity) indicate that the observed values fall in the upper (or lower) tail of the distribution, which indicate that the model is underpredicting (or overpredicting) the measure, and the patterns in the observed data depart from those in data generated from the proposed model.

Robins et al. (2000) show that posterior predictive tests can be conservative (fail to detect misfit). That is, when the null model is correct, PPMC indices can tend to some degree to concentrate around .5 rather than having a uniform distribution. Too few would be rejected at, say, at $\alpha = .05$ when the fitted model is correct, and almost certainly too few misfits would be detected when the model is misspecified.[1] However, the ease with which they can be implemented in the MCMC context makes them particularly attractive. The degree to which they are conservative turns out to depend on the choice of discrepancy measure. Gelman et al. (1996) point to a number of studies in which the posterior predictive tests are shown to have reasonable long-run frequency properties, in the sense that they approximate nominal rejection rates (see Example 10.1 for a graphical illustration of what this means).

Thus, the posterior predictive distribution method is conceptually simple and computationally straightforward under MCMC estimation. The challenge lies in finding discrepancy measures that are interesting, useful, and, preferably, have good long-run frequency properties.

The *raison d'être* of latent variables is modeling the associations among observable variables. It stands to reason that discrepancy measures that concern such associations are of particular interest for psychometric models in general. Levy (Levy 2011; Levy et al. 2009) studied the performance of a number of discrepancy indices that focus on the joint distributions of observables, including item correlations, residual covariances, log odds ratios (Sinharay and Almond 2007), and Yen's Q_3 statistic (Yen 1993). These indices can be used to detect violations of local independence from phenomena such as omitted proficiency variables, differential item functioning, item drift, testlet effects (i.e., context effects), rater effects, and method effects. Yen's Q_3 is among the indices Levy found to have good long-run frequency properties. It has the additional appeal of being easy to understand and to calculate. The Q_3 for a pair of observables is the correlation between their residuals from model-based predictions:

$$Q_{3_{jk}} = r_{e_{ij}e_{ik}}, \tag{10.9}$$

where, r is the correlation across persons i of the residuals $e_{ij} = y_{ij} - E(Y_{ij})$. When conditional independence holds, these correlations are near zero (approximately $-(n-1)^{-1}$ for an n-item test). Positive values indicate the variables are more positively associated than the model would predict, and negative values indicate a more negative association than predicted. Originally defined for dichotomous items, the Q_3 has proven useful for ordered response outcomes as well (J. Mislevy et al. 2012).

Example 10.1 (PPMC for discrete IRT). *This example extends the "discrete IRT" models of Sects. 6.1 and 6.2. We will generate data from both unidimensional discrete IRT model and a testlet model with two additional "context" variables for subsets of observables, then look at model-checking statistics assuming the unidimensional model.*

[1] Yes, this is frequentist reasoning in a book about Bayesian inference. Rubin (1984) explains that this is appropriate logic for Bayesians who want to compare features of an observed data set against the corresponding features in a sample of data sets generated from a posited model.

The unidimensional model, Fig. 10.1a, contains a single proficiency variable θ that can take five values. There are ten dichotomous observables, or conditionally independent items. The testlet model, Fig. 10.1b, contains a local proficiency variable $Context1$ that is also a parent of Items 3 and 4, and another $Context2$ that is the parent of Items 6, 7, and 8. As discussed in Sect. 6.2, the context variables make associations within such testlets higher than the θ alone would predict; that is, the items within a testlet are conditionally dependent given θ.

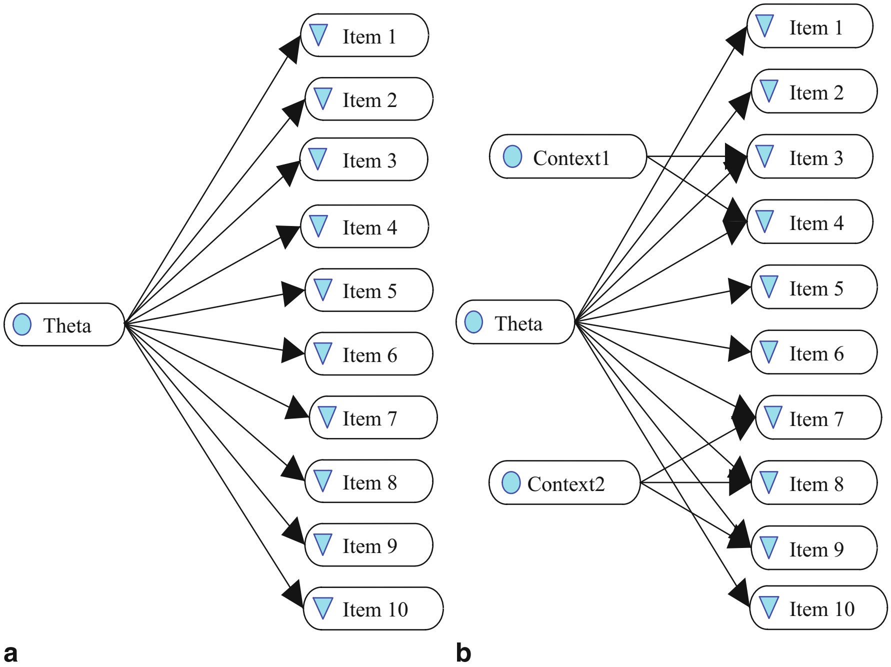


Fig. 10.1 Two alternative discrete IRT models with and without context effect. **a** Conditionally independent IRT model. **b** Testlet model
Reprinted with permission from ETS.

The data were generated in all cases with $\theta \sim$ Categorical$(.1, .2, .4, .2, .1)$. The conditional probabilities for items were given by DiBello–Samejima models (Sect. 8.5). The five levels of θ are assigned the values $\{-2, -1, 0, 1, 2\}$. A Rasch IRT model is used to calculate the probabilities of correct response to each of the items, with item difficulties $\boldsymbol{\beta} = (-1.5, -.75, 0, .75, 1.5, -1.5, -.75, 0, .75, 1.5)$:

$$p_{ij} = \Psi(\theta_i - \beta_j) = \exp(\theta_i - \beta_j)/[1 + \exp(\theta_i - \beta_j)] \,.$$

In the testlet model, students' context variables ϕ_{i1} *and* ϕ_{i2} *can take the values* $\{-1, 1\}$*. The probabilities for items 1, 2, 5, 6, and 10 are the same as in the unidimensional model but*

$$p_{ij} = \Psi\left(\theta_i + c\phi_{i1} - \beta_j\right) \quad \textit{for} \quad j = 3, 4$$

and

$$p_{ij} = \Psi\left(\theta_i + c\phi_{i2} - \beta_j\right) \quad \textit{for} \quad j = 7, 8, 9.$$

The variable c *represents the strength of the context effect. When* $c = 0$*, the testlet model simplifies to the Rasch model described above with conditional independence. If* $c = .5$*, a student familiar with a context has her proficiency increased by .5 for just those items in the testlet. Note that a student could be familiar with one context and not another. In this example, the magnitude of the effect* c *is the same in both testlets.*

Data sets were generated with 500 simulees each, for values of c *= 0, .5, .75, and 1.0. In each case, an MCMC solution was run assuming the unidimensional model, which included* $\boldsymbol{\beta}$*, the population probabilities* $\boldsymbol{\pi}$ *for* θ*, and the* θ *of each simulee. Replicate data sets* $\mathbf{y}^{\text{rep}}$ *were generated in each cycle.* Q_3 *indices for all item pairs were calculated in each cycle for both the real data and the replicate data using that cycle's draws of* $\boldsymbol{\pi}$*,* $\boldsymbol{\beta}$*, and simulees'* θ*s and* ϕ*s. PPP values were calculated for the* Q_3*s.*

Both the Q_3 *values and the PPPs are useful in analyzing the results. Figure 10.2 shows Tukey stem and leaf plots for the PPPs at each level of* c*. A row in the plot indicates the count of indices with a leading decimal. For example, the lowest left row reads* $.0|23$*. This means that the data include the values of .02 and .03. These are the worst-fitting item-residual correlations, as only 2 and 3 % of the model-generated residual correlations have higher values. For the* $c = 0$ *plot, when the unidimensional model is correct, the distribution is roughly uniform. This is what "good long run frequencies" means for an index in PPMC analyses.*

In the subsequent plots for observed data with testlet effects, the PPPs for item pairs in the same testlet are underscored. We are looking for greater-than-expected residual correlations. For $c = .5$*, these items show a tendency to be toward the lower end, but this diagnostic is not good at distinguishing them at this level of conditional dependence. The trend is more pronounced for* $c = .75$*, and by* $c = 1$*, the four item pairs from testlets are the smallest values in the plot. Their values are greater than the residuals from the null model in more than 95 % of the cycles.*

The PPP distribution for $c = 1$ *is no longer rectangular but U–shaped. We noted that the low values are for item pairs in the same testlet. The pile-up of high values indicates item pairs for which the* Q_3 *from the data was lower than the one from the replicate data most of the time, i.e., residual correlations were lower than expected. These tend to be items from different testlets.*

```
.9|2367          .9|0578          .9|05          .9|12255666
.8|246           .8|789           .8|24889       .8|134
.7|12289         .7|014445579     .7|456679      .7|1114566
.6|24            .6|2             .6|24799       .6|23348
.5|022348        .5|04669         .5|344778      .5|4
.4|0223578       .4|1456          .4|01129       .4|0247
.3|0333479       .3|3578888       .3|01255       .3|2228
.2|1459          .2|4668          .2|0012        .2|34
.1|33455         .1|023578        .1|378         .1|02356
.0|23            .0|35            .0|356         .0|00157
```

$c = 0$ $c = 0.5$ $c = 0.75$ $c = 1$

Fig. 10.2 Stem-and-leaf plots of posterior predictive probabilities for Q_3 values of item pairs. *Underscores* indicate values for item-pairs in the same testlet. By $c = .75$, the four pairs from common testlets are among the most discrepant from the fitted conditional independence model. By $c = 1$, they are the most extreme

Reprinted with permission from ETS.

Figure 10.3 shows the values of the Q_3s themselves. We see strong positive residuals for the items within testlets and negative residuals for items from different testlets. When the conditional dependence has reached 1, the Q_3 indices give us useful clues to its structure.

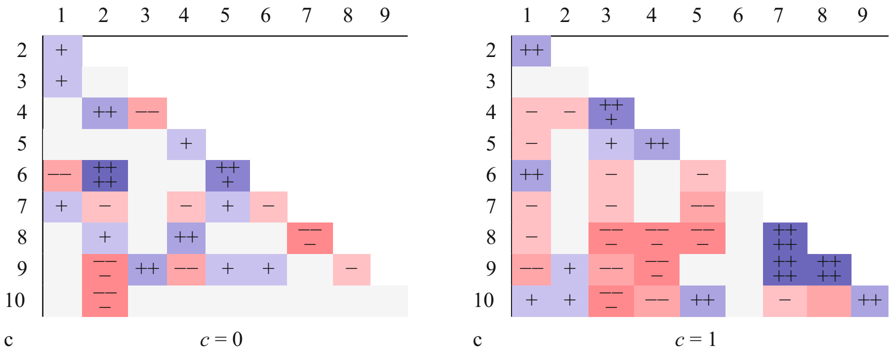


Fig. 10.3 Q_3 values for item pairs, in increments of .025 from −.10 (*bright red*, $\overset{-}{-}\overset{-}{-}$) to +.10(*dark blue*, $\overset{++}{++}$). For the actual conditionally-independent data, where $c = 0$, values closer to zero and randomly distributed. For the testlet data, the item pairs corresponding to testlets show strong residual correlations and the pairs from different testlets show strong negative residual correlations

Reprinted with permission from ETS.

We also calculated item-fit and person-fit RMSEs for the $c = 0$ and $c = 1$ data sets, under the assumption of conditional independence. Histograms for

for both kinds of residual were roughly rectangular, and nearly identical in shape across the two data sets. In contrast to the Q_3 analyses, these marginal residuals for items and for people were not sensitive to conditional dependence from testlet effects.

An examination of person-fit RMSEs did, however, reveal interesting patterns of misfit for individual simulees. Recall that the item difficulties went from easy to hard for items 1–5, and again for items 6–10. The person-fit RMSEs did distinguish simulees whose responses were more consistent or less consistent with the expectation of getting easier items right and harder items wrong. The pairs of response patterns below have the same total score but PPP values that signal whether which items are right are expected or surprising.

The point illustrated here is that different fit indices are better at picking up different kinds of misfit, so we use multiple methods for checking for different problems. The Q_3 indices are better at picking up unexpectedly strong relationships among items that can signal either unintended similarities across items or distinct skills we might want to consider measuring distinctly. They are not sensitive to particular individuals with aberrant response patterns. Person-fit indices can do this well, but tell us little about unmodeled conditional dependencies.

Total	*Pattern*	*PPP*
3	00000 01110	.004
3	11000 10000	.878
5	00111 10001	.004
5	11100 11000	.874
8	11111 01011	.004
8	11110 11110	.897

10.3 Graphical Methods

Graphical methods can be more useful than diagnostic statistics because they can reveal unexpected patterns. One of the more useful diagnostic plots from IRT is the empirical *item characteristic curve* (Lord 1980). This graph plots an estimate of the proportion of students at a proficiency level getting an item correct against the proficiency variable. Yan et al. (2003) develop a variation of the item characteristic curve that is appropriate for Bayes nets.

The basic idea of their *observable characteristic plot*, is to group the students into classes based on their proficiency variables such that all students within a class should have the same probability distribution for the observed outcome variable. The model-based probability for the class is plotted against an estimate based on a sample of individuals from that class, who should have that probability for this observable. We will first describe how the plots would be constructed and used if we knew students' proficiencies.

Consider an observable variable, *Obs*, that has two parents, *Skill1* and *Skill2*, both of which take on two values: `1` and `0`. There are four possible configurations of the two parent variables: $(0, 0)$, $(0, 1)$, $(1, 0)$, and $(1, 1)$. The conditional probability table $\mathrm{P}(\mathit{Obs}|\mathit{Skill1}, \mathit{Skill2})$ gives the probability for *Obs* given each possible configuration of the parents. For example, if $\mathrm{P}(\mathit{Obs}|\mathit{Skill1}, \mathit{Skill2})$ is a DINA (deterministic input noisy-and) model, then configurations $(0, 0)$, $(0, 1)$ and $(1, 0)$ are associated with π_{-} and configuration $(1, 1)$ is associated with π_{+}.

Suppose for the moment that we had a sample of 100 students for which we knew the value of *Obs*, *Skill1* and *Skill2*. It would then be straightforward to produce a graphical test for the values of $\mathrm{P}(\mathit{Obs}|\mathit{Skill1}, \mathit{Skill2})$ predicted by our model. Suppose that n_{00} students had the proficiency profile $(0, 0)$ and that of them, x_{00} got a correct outcome for *Obs*. We can then build a 95 % credibility interval for the proportion $\mathrm{P}(\mathit{Obs}|\mathit{Skill1} = 0, \mathit{Skill2} = 0) = \pi_{00}$ using a beta-binomial model for just that proportion. To ensure that the posterior is proper, we need to use a proper prior distribution for π_{00}. Since we expect this probability will be less than .5, we can use a weak prior that biases the estimates slightly towards small values, say a Beta(.2, .8) distribution. The posterior will then be a $\mathrm{Beta}(x_{00} + .2, n_{00} - x_{00} + .8)$ distribution we can use to form the credibility interval. We can use a similar procedure to form credibility intervals for the remaining three configurations of the parent variables. The intervals for π_{10} and π_{01} are constructed similarly. However, we expect π_{11} will be greater than .5 so we use a Beta(.8, .2) prior. The exact value of the priors are not critical, but it is important that (a) it is a proper prior, so we get a proper posterior, and (b) the sum of the parameters is small, say 1 or less, so the interval depends mainly on the data.

Figures 10.4a, b show the observable characteristic plot (Yan et al. 2003), a graphical realization of these credibility intervals. For each skill profile, a vertical bar gives the credibility interval for the proportion correct for that group. The horizontal lines are the two probabilities π_{-} and π_{+} predicted by the DINA model. The midpoint of each credibility interval is plotted with the symbol '$-$' or '$+$' according to whether π_{-} or π_{+} is the appropriate probability for this group.

Figure 10.4a shows the plot for a task that is working fairly well. All of the credibility bars overlap the probabilities predicted under the model. The first three skill groups lack one or both skills, and the modeled probability is .18; the three "observed" probabilities for these groups are between .15 and .30. The fourth group, with both skills, is modeled as having .85 probability of a correct response, and the "observed" value is .9.

Figure 10.4b shows a plot from a task that is not working according to its evidence model. While the lowest two groups have "observed" probabilities that are a bit high and the group with both skills has a high probability that

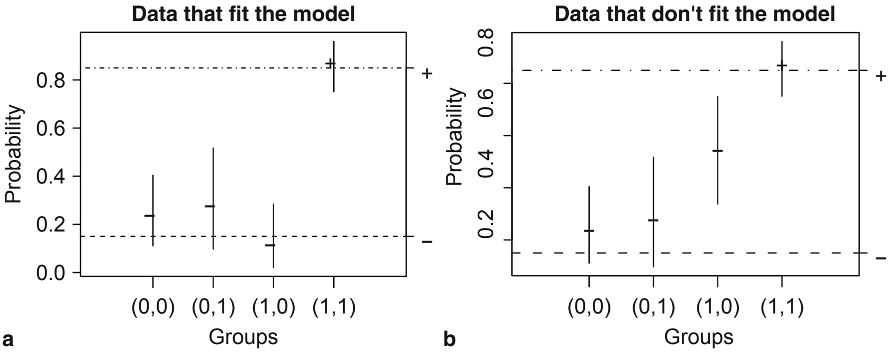


Fig. 10.4 Observable characteristic plot
Reprinted with permission from ETS.

agrees well with the modeled value. But the credibility interval for skill profile $(1, 0)$ lies midway between the π_- and π_+ lines. This calls into question the symmetric treatment of the two skills given by the DINA model. An alternative model, perhaps a compensatory one, should be investigated. Finding a similar pattern on more tasks would reinforce this choice. Alternatively, the problem may lie with the task, which should be investigated to see if it is working as expected—e.g., with think-alouds from students as they solve it.

Of course, constructing observable characteristic plots is not as simple as the above discussion implies, because the states of the proficiency variables, and therefore the proficiency profiles for each person, are not known. One solution is to use an imputed set of proficiency variable states from one cycle of an MCMC sampler (Yan et al. 2003). An alternative is to use the proportion of cycles an individual appears in a given skill profile during the MCMC as a "weight" for that individual (Sinharay et al. 2004; Sinharay and Almond 2007). Thus, if an individual who got correct outcome for the observable and appeared was assigned Skill Profile $(1, 1)$ in 75 % of the cycles, Skill Profile $(1, 0)$ in 14 % of the cycles, Skill Profile $(0, 1)$ in 10 % of the cycles, and Skill Profile $(0, 0)$ in the remaining 1 %, would provide weights of $(.01, .1, .14, .75)$ to the four profiles (in the order given in the plot in Fig. 10.4a). This way of dealing with uncertainty about individuals' latent proficiencies is analogous to the E-Step of the EM algorithm in Chap. 9.

The proficiency profiles do not need to be limited to just the parent variables of the observable in question. Including other variables produces a test of the local independence assumption. Consider what happens when we add another variable, Skill 3, to the two-variable DINA shown in Fig. 10.4a. If the local independence assumption holds, then the augmented graph should look something like Fig. 10.5a with the profiles both with and without Skill 3, giving similar credibility intervals. In Fig. 10.5b, however, Skill 3 appears to give a boost in performance on the observable. In the face of such a plot, the

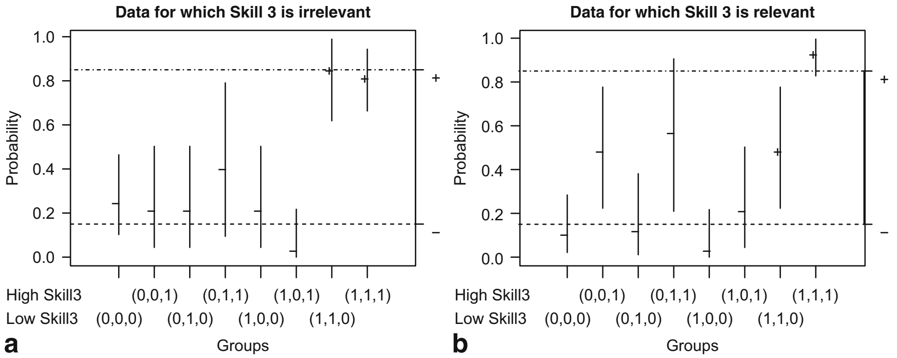


Fig. 10.5 Observable characteristic plot for additional skill
Reprinted with permission from ETS.

assumption of independence between Skill 3 and the observable in the evidence model should be reviewed; Skill 3 might need to be added as a parent to the observable.

As an extreme case, the x-axis of the graph could contain all possible skill profiles. However, all skill profiles may not be distinguishable from the data. For example, in the mixed number subtraction assessment, only nine different groups of skill patterns are identifiable based on the Q-Matrix for this particular set of tasks (Klein et al. 1981; see also Sect. 6.4 and Chap. 11 of this book). In particular, all tasks require Skill 1, therefore, all 12 skill patterns that lack Skill 1 have the same expected outcome pattern (all tasks incorrect). Similar logic reveals that the 24 skill patterns fall into 9 different equivalence classes, or groups of skill patterns that have the same expected outcome pattern (Yan et al. 2003; see also Sect. 11.1 of this book).

Sinharay and Almond (2007) develop an observable fit statistic to accompany this observable characteristic plot. It uses the same data that are used to construct the plot. Divide the proficiency space up into K equivalence classes as described above, and let τ_{ik} represent the probability that Person i is in Equivalence Class k. Define the "number" of people in Equivalence Class k as $N_k = \sum_i \tau_{ik}$. Then (assuming that all observables are binary), the "observed" number correct for Equivalence Class k and Observable j is $O_{kj} = \sum_i \tau_{ik} Y_{ij}$. The "expected" number of correct outcomes is $E_{kj} = \pi_{kj} N_k$, where π_{kj} is the probability (according to the model) of a person in Equivalence Class k getting Observable j correct. Then the goodness of fit statistic is

$$\chi_j^2 = \sum_k \frac{(O_{kj} - E_{kj})^2}{E_{kj}} \ . \tag{10.10}$$

This statistic is inspired by the classical χ^2 test. If proficiencies were known, it would follow a χ^2 distribution with degrees of freedom equal to the number of equivalence classes minus the number of probabilities estimated for this observable. Because the proficiencies are not known, Sinharay and Almond (2007) suggest using the posterior predictive distribution as a reference distribution.

An information-based analogue to Eq. 10.10 is the entropy of the "observed" proportions of correct responses in the equivalence classes with respect to the modeled proportions (Savage 1971):

$$\mathrm{Ent}_M(Y_j) = -\sum_k p_{kj}\log(\frac{p_{kj}}{\pi_{kj}}) \tag{10.11}$$

where $p_{kj} = O_{kj}/N_k$.

A problem occurs when there is a very large number of potential proficiency profiles (assignments of states to all proficiency variables in a model). If each unique proficiency profile is assigned a unique equivalence class, there may be many classes with fewer than three members in the sample. Coarser grouping is then preferable.

Sinharay and Almond (2007) introduce a graphical method called Direct Data Display. In this display, participants are sorted according to some rough measure of overall ability (e.g., number right score) and observables are sorted according to some measure of difficulty (e.g., their marginal distribution in the sample). The entire data set (or a sample, if there are too many cases) is then depicted in a grid, with students as the x–axis, from low to high, and items as the y–axis, from easy to hard. Each observed outcome Y_{ij} is used to assign a gray-scale value to a pixel in the image (white for lowest possible response, black for highest possible response). The result should be a bar that is mostly black toward the left and bottom and white toward the right and top.

The reference distribution for this plot is produced through posterior predictive checks. Some number (say ten) cycles are randomly chosen from the MCMC estimation procedure and a set of shadow data is generated from each. Any feature that appears in the real data plot but not the shadow data represents an unmodeled feature of the data. It is up to the analyst to decide if this feature is important.

Figure 10.6 shows an analysis by Sinharay (2006) of the mixed number subtraction data (Sect. 6.4). Look at the students at the lower end of the scale. In the simulated data, they get a random pattern of tasks correct through guessing. In the real data, they seem to get only the fourth and fifth easiest items right. These turn out to be the items $\frac{3}{4}-\frac{3}{4}$ and $3\frac{7}{8}-2$. Both can be solved by strategies that do not involve fraction subtraction skills: "Something minus itself is always zero," and "3 and something minus 2 is 1 and something."

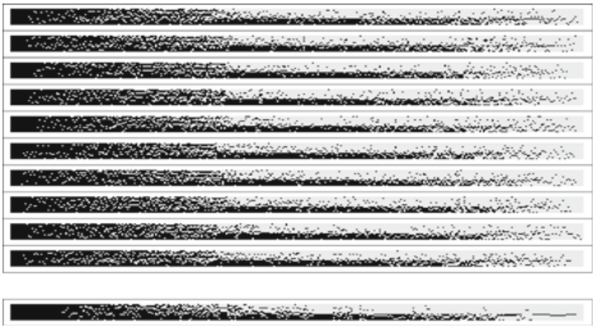


Fig. 10.6 Direct data display for mixed number subtraction test

The plot at the *bottom* is a direct data display of the mixed number subtraction data (Klein et al. 1981). The ten comparable figures above it are posterior predictive replications from an MCMC sampler. Note the difference between the real-data plot at the *bottom* and the replicates above for the last 20 or so individuals: The *replicate data sets* show more random correct answers through guessing spread across the tasks, while the *real data* show that most of the correct answers by low-scoring students are to two particular tasks. Reprinted from Sinharay (2006) with permission of S. Sinharay and Sage Publications.

10.4 Differential Task Functioning

A special kind of model misfit can occur when the population of interest for an assessment consists of distinct subpopulations. Ideally, the assessment should behave in the same way for each subpopulation, i.e., the same evidence models with the same parameters should be appropriate. There has been special concern when the subpopulations are based on gender, racial, or cultural groups. However, the problem also encompasses issues of whether an assessment is suitable for multiple purposes. For example, is the same test (with the same evidence models) suitable for both 2-year community colleges and 4-year universities?

This problem has been well studied under the name *differential item functioning* (DIF) (Holland and Wainer 1993). The basic question of DIF analysis is whether an item behaves differently for different subpopulations, especially if the different behavior is unrelated to the construct of interest. In the context of the more complex assessments supported by Bayes nets, tasks are more naturally the unit of analysis. Therefore, we refer to this phenomenon as *differential task functioning.*

Measuring the equivalence of a task across different subpopulations is complicated by the fact that the populations often have different distributions of proficiency. DIF analyses therefore examine conditional probabilities of task response across groups conditional on some matching criteria, which could be observed scores or latent proficiency variables (Holland and Wainer 1993).

The latter alternative is well suited to Bayes nets proficiency models: If the proficiency model is a complete description of the construct, then the observable outcomes should be conditionally independent of subpopulation membership given the proficiency variables. In other words, the performances of the groups may differ, but all of the differences can be captured as differences in distributions on the proficiency variables. This framing provides a testable definition of differential task functioning: Are task responses conditionally independent given proficiency variables (including context variables when needed)—specifically not requiring group membership as an additional parent? Note that because of the independence assumptions embedded in Bayes nets we have developed, only the proficiency variables that are involved in the evidence model for a given task need to be considered for this determination. Bishop et al. (1975) suggest thinking about this test as the difference between two models, one of which contains the independence condition.

Let D be a demographic variable representing membership in a subpopulation of interest, O be an observable in a task to be tested for differential task function, and P be a variable representing the proficiencies variables that are parents of any observable in the task of interest. Figure 10.7a shows the model without differential task functioning, with P D-separating D and O; Fig. 10.7b shows the model with differential task functioning. If P were known, the difference in deviance between those two models would follow a χ^2 distribution. For unstructured, or hyper-Dirichlet, conditional probabilities, the degrees of freedom would be $(|D|-1)(|O|-1)$, where $|D|$ represents the number of levels of the demographic variable and $|O|$ represents the number of possible states of the observable. PPMC or Williamson simulations (Sect. 10.1) provide a custom-built reference distribution to use in practice.

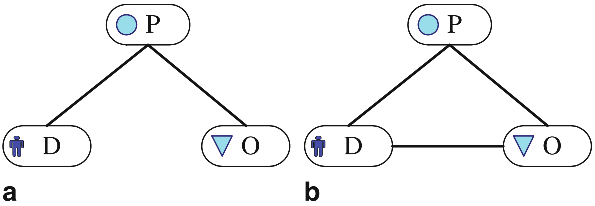


Fig. 10.7 Two graphs showing the effects of differential task functioning. **a** Graph with no DTF **b** Graph with DTF

Reprinted with permission from ETS.

The method illustrated in Fig. 10.4b provides a graphical check for DIF: We examine empirical probabilities of task response, for students grouped by their values on the proficiency variables that are parents of the task *and* their subpopulation membership. As discussed there, since proficiency is not known these charts can be made by assigning students their modal proficiencies or by distributing their responses across groups according to their posterior probabilities.

A statistical test is easiest to implement with a parametric model for conditional probabilities, as in Sects. 8.4 and 8.5.[2] For example, the DINA model for a dichotomous task with two binary skill parents shown in Fig. 8.6 has two parameters, the true-positive probability π_+ of a correct response when a student has both *Skill 1* and *Skill 2*, and a false-negative probability π_- when she lacks one or both skills. To allow for DIF, extend the model with group-by-task interaction parameters δ_{+k} and δ_{-k}, such that the true-positive probability for a student in Group k is $\pi_+ + \delta_{+k}$ and the false-positive probability is $\pi_- + \delta_{-k}$. (To identify the values of the DIF parameters, designate one group, the so-called reference group, for which $\delta_{+k} = \delta_{-k} = 0$.) Fit the model using EM or MCMC, and examine the posterior distribution of the δs. For example, does the 95 % posterior credibility interval for a given δ include 0? If so, the DIF is probably not substantial.

Similarly, suppose the conditional probabilities for a task are given by a DiBello–Samejima model as in Eq. 8.12. The no-DIF cumulative probability of a response at or above level m from a student with effective theta θ is $\mathrm{P}(X \geq x_m|\theta) = \mathrm{logit}^{-1}(\theta - d_m)$, with d_m the category difficulty. A single-parameter shift, toward relative harder or easier for subpopulation k, is affected by fitting $\mathrm{P}(X \geq x_m|\theta, D = k) = \mathrm{logit}^{-1}(\theta + \delta_k - d_m)$. More refined checks are affected by incorporating category DIF parameters, as $\mathrm{P}(X \geq x_m|\theta, D = k) = \mathrm{logit}^{-1}(\theta - d_m + \delta_{mk})$.

In high-stakes assessment, DIF analysis is used with unidimensional IRT or number-correct scores to detect and remove items which show DIF that is both statistically significant and large enough to distort the meaning of scores across groups. In the medium and low stakes uses of Bayes nets in assessment, the goal is understanding students' profiles of proficiency to support further learning. Including tasks that exhibit DIF but building the effect into the model is useful when certain tasks provide evidence but, due to construct irrelevant features, provide less information or conflicting information about students from different backgrounds. For example, a writing task dealing with job applications on the National Assessment of Educational Progress some years ago was found to provide less information for $8th$ grade students than $12th$ grade students, presumably because more of the older students had personal experience with this genre. In a large-scale survey like NAEP, meant to study patterns of proficiency among populations rather than assess individuals, downweighting the evidence from this task for younger students is an appropriate course of action.

[2] In IRT, the most popular method for detecting DIF is the nonparametric Mantel–Haenszel test (Holland and Thayer 1988), which conditions on observed score. See Exercise 10.9 and Sect. 13.2.2.

10.5 Model Comparison

The previous section contained a number of diagnostic tests to see if a proposed model fits. Implicit in the idea of model diagnostics is that if the model does not fit well, a new better model can be found. Suppose that a new model is proposed. How do we know that the new model is better? The two approaches discussed below build around ideas discussed in Sect. 10.1 in connection with Good's logarithmic score. The classical statistical approach is to compare fit from the perspective of likelihood, which revolves around deviance. An alternative approach based on predictive power uses entropy as a metric (Gilula and Haberman 1995; Gilula and Haberman 2001).

10.5.1 The DIC Criterion

The likelihood function is the basis of many model-comparison indices. Intuitively, the better a model fits a given data set $\mathbf{Y}$, the higher the likelihood $\mathrm{P}(\mathbf{Y}|\boldsymbol{\omega})$, where $\boldsymbol{\omega}$ represents the model parameters. The *deviance* of a model compared to a saturated model is defined as

$$D(\boldsymbol{\omega}) = -2\log\{p(\mathbf{Y}|\boldsymbol{\omega})\} + 2\log\{f(\mathbf{Y})\}, \tag{10.12}$$

where $f(\mathbf{Y})$ is a term that depends on the data, but not the parameters. This term usually drops out of the equations through subtraction when values for two competing models are compared (thus a common misuse of terminology is calling just $-2\log\{p(\mathbf{Y}|\boldsymbol{\omega})\}$ the deviance). Note that the first term is just twice the logarithmic score (Eq. 10.3) summed over the sample, evaluated at a particular parameter value $\boldsymbol{\omega}$.

A familiar case is when either the new model or the old one is a simplification, or submodel, of the other model—Model 1 is nested within Model 2, for instance, a linear regression model with two coefficients fixed to zero. Model 2 will always fit better than Model 1, but, is the improvement enough to justify estimating additional parameters? The likelihood ratio test developed by Neyman and Pearson in 1928 can be expressed in terms of deviance: When Model 1 is the true model, $D(\hat{\boldsymbol{\omega}}_2) - D(\hat{\boldsymbol{\omega}}_1)$ should approximately follow a χ^2 distribution with degrees of freedom equal to the difference in their numbers of parameters.

This test does not generalize easily to the case where one model is not a submodel of the other. Several authors have suggested model fit indices that can be used with models that are not nested, and also correct for the number of parameters in the model. The most popular are the AIC (Akaike 1973) and BIC (Schwarz 1978) criteria:

$$AIC = -2\log \mathrm{P}(\mathbf{Y}|\hat{\boldsymbol{\omega}}) + 2d, \quad \text{and} \quad BIC = -2\log \mathrm{P}(\mathbf{Y}|\hat{\boldsymbol{\omega}}) + \log(n)d,$$

where d is the number of parameters in the model, and n is the number of observations. A more complicated model is preferred to a simpler one (one with a smaller p_D) only if the improvement in fit is bigger than the difference in dimensionality. Therefore when comparing two models, the one with the smaller value is preferred. Differences less than less than, say, 4 or 5 are not considered compelling.

AIC, BIC, and similar alternatives require knowing the number of free parameters to be estimated in the model. While counting the number of parameters is straightforward in a regression problem, it is not so straightforward in complex Bayes model, when parameters are constrained through prior distributions or subsets of them are related in hierarchical structures.

Spiegelhalter et al. (2002) introduce a measure called DIC, which includes a Bayesian notion of dimensionality(see also Plummer 2008; Gelman et al. 2013a). The $-2\log\{p(\mathbf{Y}|\boldsymbol{\omega})\}$ part turns out to be very easy to calculate for Bayes net models. Let $\mathbf{Y}_i$ be the vector of outcomes for each person in the sample. Score the students according to the method of Chap. 5, only make sure that when passing messages with a Bayes net, or from the evidence model to the scoring model that the message tables are not normalized. Now, pick any node in the scoring model and calculate the normalization constant for its marginal distribution. This should have the value $p(\mathbf{Y}_i|\boldsymbol{\omega})$. Its log is the person specific contribution to the deviance:

$$D_i(\boldsymbol{\omega}) = -2\log p(\mathbf{Y}_i|\boldsymbol{\omega}) \ . \tag{10.13}$$

The deviance $D(\boldsymbol{\omega})$ is the sum of the person specific contributions across persons for a given value of $\boldsymbol{\omega}$.

Then (Spiegelhalter et al. 2002) propose a measure of model fit based on the posterior distribution of deviance:

$$DIC = \overline{D(\boldsymbol{\omega})} + p_D, \tag{10.14}$$

where $\overline{D(\boldsymbol{\omega})} = \mathrm{E}[D(\boldsymbol{\omega})]$, the average deviance calculated using the draws of $\boldsymbol{\omega}$ across MCMC cycles, and the dimensionality p_D is calculated as either of two asymptotically equivalent ways (Gelman et al. 2013b):

$$p_D = \overline{D(\boldsymbol{\omega})} - D(\bar{\boldsymbol{\omega}}) \tag{10.15}$$

or

$$p_D = \frac{1}{2}\mathrm{Var}[D(\boldsymbol{\omega})]. \tag{10.16}$$

As with the AIC and DIC, a smaller value of DIC is preferred.

Spiegelhalter et al. (2002) showed the values of p_D agree well with parameter counts in straightforward cases like nested regression models. However, DIC assumes the posterior mean is a good estimate of the stochastic parameters in the model, and this assumption is not always reasonable. A case in point is finite mixture models, which includes Bayes nets with finite-valued proficiencies. Suppose, e.g., a proficiency can take three values that represent strategies for solving mixed-number subtraction, where a student is assumed to apply the same strategy on all tasks. It is mechanically possible to label them 1, 2, and 3, then calculate a posterior mean, but the result is not a meaningful quantity in the model. Two work-arounds are useful to compare structurally similar models. First, we can substitute posterior modes for means in $\bar{\boldsymbol{\omega}}$ if the values for such variables are well-determined, that is, with most of the posterior probability on one value. Second, in some problems we can approximate the discrete variable with a continuous one. Example 10.2 illustrates the latter approach.

From the individual contributions to the deviance, $D_i(\boldsymbol{\omega})$, an individual contribution to the dimensionality can be defined:

$$p_{D_i} = \overline{D_i(\boldsymbol{\omega})} - D_i(\bar{\boldsymbol{\omega}}) \ . \tag{10.17}$$

Spiegelhalter et al. (2002) suggest that this is a measure of *leverage*. Leverage is a measure of how influential an individual is determining the parameters of a model. For example, in a linear regression, points with high leverage are often outliers on one or more explanatory variables.

Any individuals with an unusually high leverage value should be examined carefully. Does this individual really belong with the other data, or is something else going on with this person? High values can indicate issues with either the data, the model or the model fitting process.

Example 10.2 (DIC for the Testlet Model). *The definitional application of DIC does not apply directly to the discrete IRT model and context model of Example 10.1. The results of fitting the context model to the $c = 1$ data set showed the posterior modes of more than half of the simulees' two-valued ϕs had probabilities of less than .7. This is not disconcerting because the purpose of a testlet model is to account for conditional dependence in responses, not to estimate individuals' testlet values. But it does mean the work-around of using posterior modes to approximate DIC is not appropriate.*

We can, however, compare analogous models with continuous θ, ϕ_1, and ϕ_2, using normal priors with mean 0 and precision .25. This is more reasonable since the categorical values of the θs in the discrete IRT model are in fact associated with the values $\{-2, -1, 0, 1, 2\}$ in calculating conditional probabilities through the Rasch model. Continuous values for ϕs also accord with the model's conditional probability function, and account better for the diffuse posteriors of most simulees' testlet effects.

Accordingly, DIC values were obtained fitting a conditional independence model, a two-testlet model with a common c across testlets, and a testlet model with a c for each testlet. The results are are shown below. The model with a single c fits much better than the conditional independence model, and the model allowing different cs fits marginally better.

Model	$\overline{D(\boldsymbol{\omega})}$	$D(\bar{\boldsymbol{\omega}})$	p_D	DIC	Difference
Conditional independence	5366.23	5030.71	335.52	5701.75	
Testlets, common c	5013.86	4415.08	598.78	5612.64	89.11
Testlets, different cs	5043.02	4478.51	564.51	5607.52	5.12

10.5.2 Prediction Criteria

One problem with fit measures based on deviance is that they can be overly sensitive to occurrences of observations with small probabilities. (This is the rare vs. surprising difference that Good corrected for in his index for evaluating predictions, by subtracting out the expected log penalty, or entropy.) Gilula and Haberman (2001) propose model comparison metrics based on expected improvement in prediction that are less sensitive to this problem. The indices are based on improving prediction of a new observation: Given the information in the data $\mathbf{Y}$, how much better would Model $M^{(2)}$ be expected to predict a new observation than Model $M^{(1)}$? This question can be answered in terms of how much smaller the expected deviance would be, or entropy reduction.

The entropy for $M^{(r)}$ is

$$\text{Ent}(M^{(r)}) = -\sum_{i=1}^{N} p^{(r)}(\mathbf{y}_i) \log(p^{(r)}(\mathbf{y}_i)), \tag{10.18}$$

where $p^{(r)}(\mathbf{y}_i)$ is the modeled probability of $\mathbf{y}_i$ under $M^{(r)}$. A measure of how much better $M^{(2)}$ predicts a new observation than $M^{(1)}$ is

$$\text{Ent}(M^{(1)}) - \text{Ent}(M^{(2)}). \tag{10.19}$$

The difference is positive if $M^{(2)}$ yields better prediction of Y than $M^{(1)}$, and negative if $M^{(1)}$ yields better prediction. This measure can be used to compare models that are nested or non-nested, and to evaluate the impact of including covariate in a model. If $M^{(1)}$ is nested within $M^{(2)}$, the difference is non-negative. In cases where counting parameters is straightforward, one can evaluate model differences in terms of improvement per parameter by dividing Eq. 10.19 by the difference in number of parameters. For interpretability, it can be rescaled to improved prediction for a single observation by dividing through by N.

Gilula and Haberman (2001) also suggest a criterion for model comparison that is analogous to proportion of variance accounted for in regression analysis. Let $M^{(0)}$ be a base model. The proportional improvement afforded by $M^{(r)}$ is

$$\frac{\text{Ent}(M^{(0)}) - \text{Ent}(M^{(r)})}{\text{Ent}(M^{(0)})}. \tag{10.20}$$

Example 10.3 (Prediction Improvement in the Testlet Model). *This example continues with the ten-task discrete IRT testlet model of Example 10.1, using the data set with* $N = 500$ *and two testlets with a common*

$c = 1$. *The following nested models were fit, and per-person entropy values calculated:*

$M^{(0)}$: *Null model, with item effects only. All responses of all persons modeled as independent, with item probabilities given by sample proportions-correct.*

$M^{(1)}$: *Discrete Rasch model, i.e., person parameters (i.e., latent variables) added, conditional independence assumed.*

$M^{(2)}$: *Testlet model, common c.*

$M^{(3)}$: *Testlet model, possibly different cs for the two testlets.*

Model	Ent($M^{(r)}$)	% improvement over $M^{(0)}$	% improvement over $M^{(r-1)}$
Null model	6.223		
Conditional independence	2.687	56.8	56.8
Testlets, common c	2.645	57.5	1.6
Testlets, different cs	2.519	59.5	4.7

Person parameters improve prediction substantially, but testlet effects do not. This is so even though the fit investigation for conditional dependence in Example 10.1 revealed strong testlet effects. These can be important as feedback to test developers for finding problems with items and for discovering omitted skills in the proficiency model, but in this case they have little impact for estimating the proficiencies of individuals. (The story can be different for assessments consisting of a small number of testlets with many items each, and strong testlet effects.) The remaining entropy is the variation inherent in the Bernoulli distributions of the item responses given person and task parameters.

10.6 Model Selection

It is only a short step from comparing two models to see which is better, to searching for a best model to fit a particular set of data. Heckerman (1998) (reprinted in Jordan 1998), Buntine (1996), and Gelman et al. (2013b) provide good tutorials. Cowell et al. (1999) has several chapters on this topic, and Neapolitan (2004) devotes half the book to both parameter and structure learning. The rest of this chapter will briefly review some of the trends in this rapidly evolving area.

The idea is to search the space of possible Bayes net models that optimize a given criterion. Some example criteria include model likelihood, or the likelihood of the data under the model; posterior model probability, or the posterior probability of the data under the model; various penalized versions of those criteria, such as AIC, BIC, and DIC; and predictive power, as in Sect. 10.5.2. The required computations can be made more efficient if the criterion factors with the graph (Cooper and Herskovits 1992).

In the educational assessment, setting the problem is more complex than in some domains because the proficiency variables are usually latent. A special case of much interest with the kinds of networks we have focused on in this book, and in cognitive diagnosis models more general, is when we are not 100 % sure about the Q-matrix. To tackle this problem specifically, the general techniques described below can be applied by maintaining the set of proficiency variables, and searching over the inclusion or exclusion of edges from proficiency parents to potential observable children (de la Torre 2008). The tasks in a test determine which possible Q-matrices can be distinguished from one another and which cannot; Liu et al. (2012) provide theory for this identification issue.

One problem with model search is over fitting. A model that is over fit will give good predictions of the training data, but may not be very good at future predictions. Simpler models tend to avoid over fitting better than more complex ones. One way to guard against over fitting is *Cross validation* (Kohavi 1995). The basic idea is to divide the data into two groups. The *training data* is used in the model selection algorithm, and the *test data* is used in evaluating which model to select. More complex cross validation schemes use multiple training and testing data sets to set parameters of the model search algorithm.

The remainder of this section summarizes some basic results from the field of model search. Section 10.6.1 reviews simple search strategies and Sect. 10.6.2 looks at stochastic search strategies. Section 10.6.3 considers choosing a set of models, not just a single best model. Section 10.6.4 looks at prior distributions over the space of possible models. Section 10.7 looks at an important technical issue that comes up during model search, the fact that models with different graphical structures can in fact be equivalent.

10.6.1 Simple Search Strategies

In a typical problem, the space of all possible models is too big to perform a "British Museum" search (calculate goodness of fit measure for every possible model). Instead several possible heuristic strategies can be used. The following approaches are familiar from regression analysis and structural equation modeling. They can be applied with, although they require more calculation:

1. *Forward Selection*—Start with a disconnected graph and keep adding edges until the new model fits no better than the current model.
2. *Backward Selection*—Start with the saturated (completely connected) graph and keep removing edges until the new model fits substantially worse than the current model.

3. *Forward and Backward Selection*—At each stage add or remove edges to optimize model fit.

Usually these searches are done using the *greedy* or *myopic* version of the algorithm. That is, at each stage, the best single modification is chosen, without trying to look ahead to the effect of multiple modifications. This strategy often works fairly well, but it can get stuck in local maxima, e.g., the best three single modifications considered one at a time in sequence might not improve model fit as much as the best set of three together. A better strategy for fixed computational cost is greedy search with multiple restarts from random starting graphs (Chickering 1996).

The search can be carried out in either the space of directed or undirected graphs, although the choice has some consequences for model equivalence (Sect. 10.7).

10.6.2 Stochastic Search

Incorporating a probabilistic aspect into a search can help get around the problem of local maxima. Occasionally accepting a modification to the model that decreases the criterion can lead to two- or three-step improvements in the model that a greedy search would miss. Stochastic search methods accept such modifications with a small probability, so they should get to the globally optimum model eventually. The two most popular methods are simulated annealing and model search MCMC.

Simulated Annealing introduces a parameter called "temperature" that controls the rate at which changes that do not improve the model are selected. At each cycle, the algorithm carries out the following steps.

1 Randomly select a change in graph, and evaluate the criterion for both the old and new models. Let δE be the change in the fit measure (the name suggests that it is the change in energy of the system).
2a If change improves the model, always accept.
2b If change does not improve the model, accept with probability based on temperature, often $e^{-\delta E/T}$.

The temperature is started at a high value (almost all changes accepted) and then slowly "cooled" as time goes on (like annealing a metal). Depending on the algorithm, it can be reheated and cooled several times. High temperatures help avoid local maxima, while low temperatures cause it to climb into local maxima. When the temperature is zero, the procedure is a greedy algorithm.

The acceptance criteria in Step 2 looks similar to the formula in the Metropolis–Hastings algorithm, and was in fact derived from the Metropolis formula. *Model Search MCMC* extends the MCMC algorithm with a model selection step. At the beginning of each cycle, propose a “step” (simple modification) in model space. Then use the Metropolis–Hastings rule to accept or reject the modification. There is a technical complication here in that the Metropolis–Hastings algorithm requires a reversible jumping rule, that is, it must be possible to calculate the probability of stepping forward to the new model and backward to the old one. Liu (2001) discusses reversible jumping rules. Madigan, Gavrin, and Raftery (1995a) implement the procedure for Bayes net model search.

10.6.3 Multiple Models

The typical model selection algorithm finds a single best model, and then proceeds to make predictions as if that model were true. However, this ignores an important component of uncertainty, our uncertainty due to not knowing the correct model. Draper et al. (1987) (also Draper 1995) suggest “averaging” predictions across multiple models.

Let $\varDelta$ be some quantity of interest, such as the probability that a student has mastered all of some subset of skills. Let $\mathbf{Y}$ be the data. Let S^h for $h = 1, \ldots, H$ be a collection of models, say with different plausible instantiations of the Q-matrix structure. Then

$$\mathrm{P}(\varDelta|\mathbf{Y}) = \sum_{h=1}^{H} \mathrm{P}(\varDelta|S^h, \mathbf{Y})\mathrm{P}(S^h|\mathbf{Y}) \tag{10.21}$$

is the posterior distribution for $\varDelta$ averaged across the models. Methods that average over several models tend to have better predictive accuracy (when measured using cross validation) than methods that rely on a single “best” model.

Madigan et al. (1996) point out that if model selection is done through model search MCMC, there is no reason to need to select a set of best models before doing model averaging. Simply calculate a value of $\varDelta$ at each cycle of the MCMC loop using the model employed in that cycle, then report on its posterior distribution across cycles—effectively across the space of possible models, each weighted by its posterior probability.

10.6.4 Priors Over Models

Most of the methods search methods above, either explicitly or implicitly, put a uniform prior over models. Maximum likelihood procedures implicitly weight all models equally *a priori*. Bayesian methods (like Bayes factors) have a built-in penalty for model complexity, because the extra parameters in the more complex models lead to lower likelihoods when they are integrated out.

Several methods for more structured priors have been proposed. Heckerman et al. (1995) suggest a prior based on deviations from proposed model, and Madigan et al. (1995a) use imaginary data from experts to construct a prior. Heckerman (1998) reviews proposed priors over models.

10.7 Equivalent Models and Causality

The problem of searching over the space of possible directed graphs to find the best model is complicated by the fact that models with different graphs can in some sense be equivalent. The first possible problem lies with the direction of edges (Sect. 10.7.1). The second problem lies with the effect of unobserved and unmodeled variables (Sect. 10.7.2). The problems are confounded if the goal of the model selection process is to discover causal mechanisms. Section 10.7.3 discusses some of the limitations of procedures with this goal.

10.7.1 Edge Orientation

Putting priors over a collection of directed graphs encounters the problem that certain models are reparameterizations of each other (Andersen et al. 1996), in the sense that they produce the same joint distribution over the variables.

Figure 10.8(a)–(c) represent reparameterizations of each other. Each graph contains the independence condition "A is independent of C given B." However, Fig. 10.8(d) has a different set of conditional independence conditions. In a simple unpenalized search, the first three models would all have the same likelihoods. In a model averaging situation, the first three would get three times the weight of the fourth model even though they are essentially the same.

Spirtes et al. (1997) introduce an extended graphical notation called *partial ancestral graphs (PAGs)* to mark models that are identical in this sense. In this notation, arrows are annotated with circles to show which edges can and cannot be reversed without changing the model. Searching the space of PAGs instead of the space of DAGs avoids counting the same essential model more than once.

The motivation of much of the structure learning research is to learn causal structure from data. If the data are a faithful[3] representation of the underlying process, then the "causal" model should be the one with the fewest arrows.

In certain cases, the direction of the arrows in minimal directed model is apparent from the data. For example, in Fig. 10.8(d), A and C are "causes" of B, because the directed arrows must run in that particular direction to make the independence conditions work. If the best fitting model was Fig. 10.8(a), (b), or (c), then the causal structure would not be apparent from the data.

10.7.2 Unobserved Variables

It is important to keep in mind that causal structure learned from data is limited by which variables are included in the data set. Unobserved or hidden

[3] The term "faithful" has a precise technical definition in the literature of causal discovery: Roughly, the d-separation relationships in the digraph correspond completely to the conditional independencies in the probability distribution.

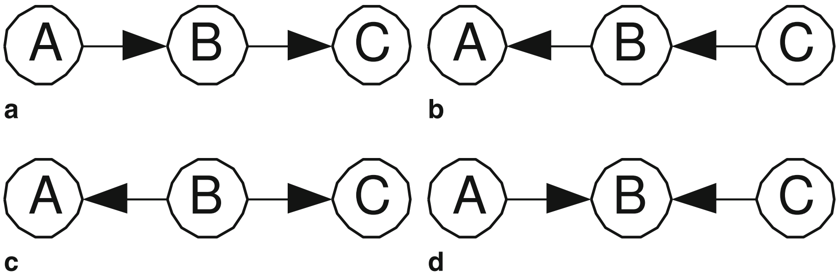


Fig. 10.8 Graphs (a), (b), and (c) have identical independence structures, but Graph (d) does not

Reprinted from Almond et al. (2006) with permission from ETS.

variables can affect the causal conclusions in one of two ways: they can be intermediate or common causes, and they can produce selection effects.

Variables that are not observed can redefine the meaning of directed edge. In Fig. 10.9b, the hidden variable H presents a common cause which accounts for the apparent relationship between A and C. When H is unobserved, the joint distribution of A and C could be identical in the models of Fig. 10.9a, b.

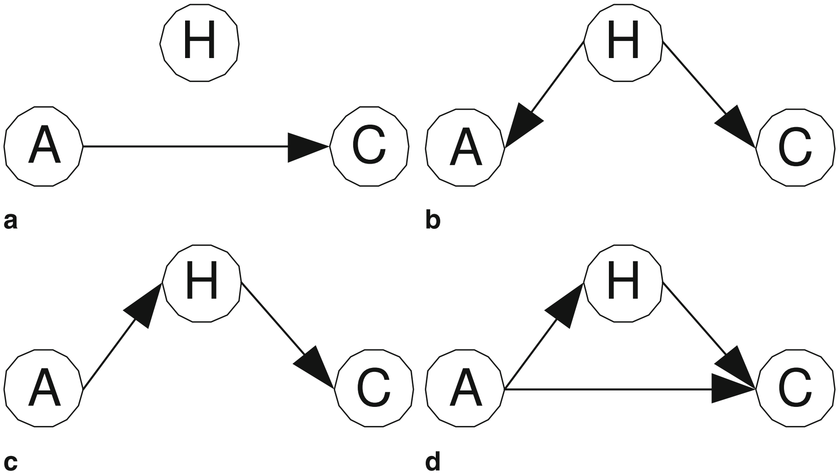


Fig. 10.9 Four graphs which could have identical distributions on the observed variables A and C. **a** No hidden cause, **b** common cause, **c** intermediate cause, **d** partial cause

Reprinted from Almond et al. (2006) with permission from ETS.

Figure 10.9c, d present two other models that can produce the same distribution over $\{A, C\}$ as Fig. 10.9a. In Fig. 10.9c, A causes H which in turn causes C. In Fig. 10.9d, the influence of H is only partial. None of the four are distinguishable when H is unobserved. Thus "cause" can only be defined relative to a universe of variables. The choice of that universe will influence the causal structure that is found by a search procedure and possibly its interpretation.

Selection bias can also limit the conclusions. This can happen when data are obtained in a so-called observational study of an existing population, a convenience sample, or a self-selected sample, as opposed to an experiment. Let the variable S represent selection of a case in the data set that is being used for model selection. Only cases in which S is `true` are observed. For example, if the sample consists of all of the students in a particular classroom and either A or C is a topic of instruction in the classroom, then the relationship between A and C may be different than in a general population, some of whom have received instruction and some of whom have not.

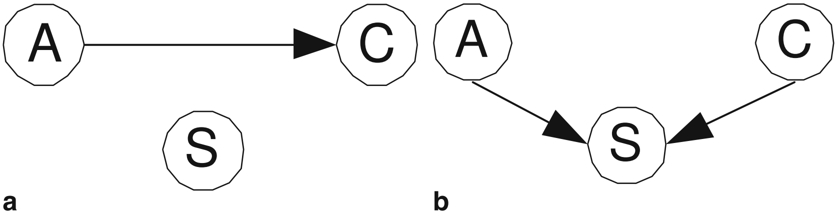


Fig. 10.10 Selection effect produces apparent dependence among observed variables. **a** No selection effect. **b** Selection effect

Reprinted from Almond et al. (2006) with permission from ETS.

Figure 10.10a, b illustrate two models which are once again equivalent. If A and C are both related to the selection mechanism then there might appear to be a relationship between them, even though they are independent in the population at large. This example illustrates the importance of random sampling in surveys. Controlling the selection mechanism explicitly ensures that it is independent from the measured variables.

10.7.3 Why Unsupervised Learning cannot Prove Causality

The literature on causal discovery is filled with some rather precise mathematical definitions of the term "causal," which take into account some of the difficulties described above. These mathematical definitions do not always correspond to the lay definition of causality. This can produce problems if the results are presented or interpreted carelessly. Example 10.4 illustrates some of the problems.

Example 10.4 (Educational Survey). *Consider an educational survey that asks both background questions about a subject (producing demographic variables) and cognitive tasks designed to measure a particular proficiency. For the moment, construct a data set using the background variables Gender and Race and include a proficiency variable and a few cognitive tasks. Applying a causal discovery method to this data set would likely result in a directed graph like the one in Fig. 10.11.*

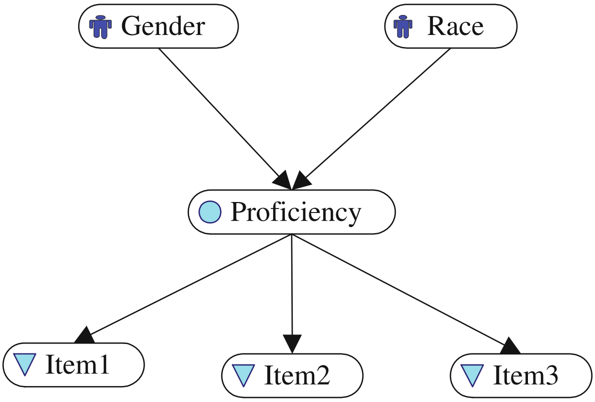


Fig. 10.11 A minimal graph which should not be interpreted as causal Reprinted from Almond et al. (2006) with permission from ETS.

This model says that there is an association (in this sampled population) between Gender and Proficiency and between Race and Proficiency. However, there are far too many factors excluded from the model to hope to make causal conclusions.

Suppose an additional demographic variable, Parent's Education, was included. This would likely result in a model like that of Fig. 10.12. Here, the new variable explains some, but not all, of the association between Race and Proficiency.

This example treats in a superficial way a truly thorny educational problem, the achievement gap. Barton (2003) takes a more thorough look. The research summarized in there includes not only observational studies that characterize the gap and highlight potential causes, but also experimental studies that confirm or refute potential causes.

Causality research is an important motivation for statistical learning, but causality cannot be proven by statistics alone, especially from observational studies. At best an observational study can make us suspect that a factor is a cause of an observed problem and suggest research to follow up on it.

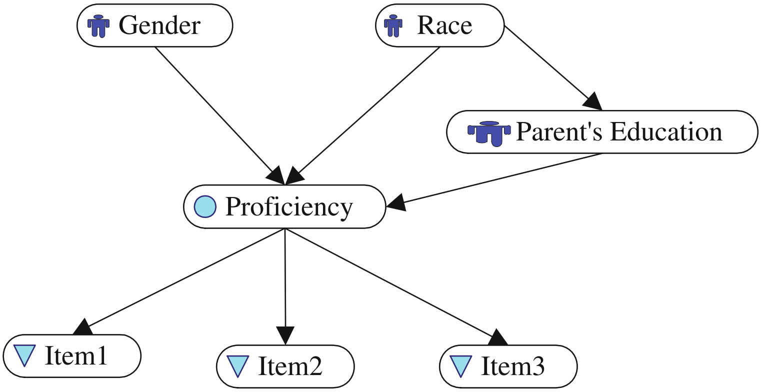


Fig. 10.12 Inclusion of an additional variable changes picture dramatically Reprinted from Almond et al.(2006) with permission from ETS.

Model search can provide abductive evidence suggesting that a factor may be a cause, but before accepting that factor as "causal" we would want confirming evidence and an understanding of the causal mechanism.

Learning causality from data has been a raging debate in the statistics community for as long as there has been data analysis. For those who want to know more about this topic, Holland (1986) on Rubin's model and Shafer (1996) are good places to start—then, on to Pearl (2009)!

10.8 The "True" Model

This chapter started with the idea that model fit statistics could help us explore both how and how well a model fit a given set of data, and progressed to searching for the model that best fits the data. This idea of model search might lead one to believe that there was a "true" model we are searching for. However, unless the data are generated through simulation, the truth will never be known.

We began the chapter with George Box's maxim, "All models are false, but some are useful." Models are useful if they provide an explanation of a complex phenomenon, predict future observations, or support better practical decisions, such as instructional treatment. Generally speaking, simpler models provide better explanations, and often better predictions. For this reason, it is important to evaluate the predictive power of models using cross validation.

Model checking becomes especially important when there is an underlying cognitive model that the graphical model is designed to represent. In this case, feedback about how well the mathematical model fits may supply insight into how well the cognitive model fits. This may in turn lead to new hypotheses about the underlying cognitive structure and new experiments to validate those hypotheses.

Exercises

Williamson (2000) proposed a model, shown in Fig. 10.13a for a hypothetical physician licensure exam using simulated patients. Each simulated patient required a different combination of skills as shown in the graph, and the candidate's care for each simulated patient was judged on a four point scale. Suppose that the model in Fig. 10.13a, Model A, is closer to the true cognitive demands of the simulated patient tasks, but that the model of 10.13b, Model B, is proposed to draw inferences from the assessment. (Model B was created by deleting the *TreatmentPlan* node, shown in gray, from the original model.)

To evaluate the two models, Table 10.1 presents the outcomes from three hypothetical students who took the assessment. Both Bayesian network models were used to predict the probability of each outcome given all of the others. Table 10.1 shows the result. Refer to these models in the following exercises.

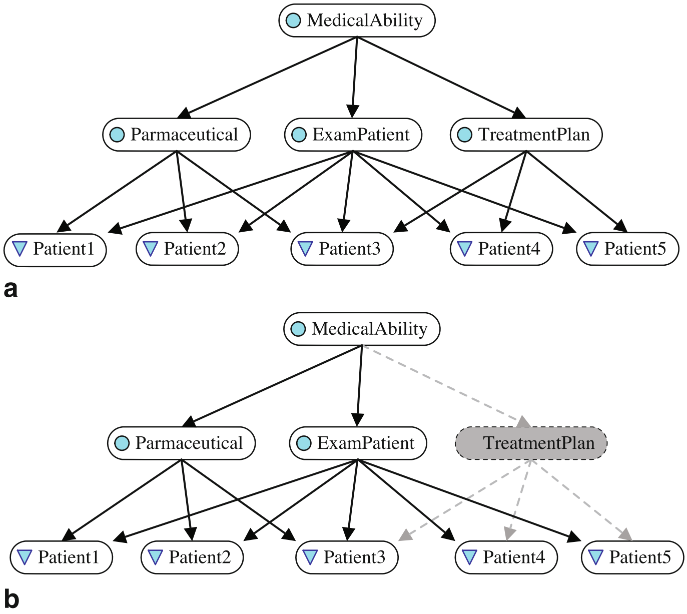


Fig. 10.13 Two candidate models for a hypothetical medical licensure assessment. **a** Model A. **b** Model B

Reprinted with permission from ETS.

Table 10.1 Actual and predicted outcomes for the hypothetical medical licensure exam.

Each row of the table gives (a) the observation for one simulated student on one patient (b) the prediction under Model A given all of the observations except the one in the current row, (c) the prediction under Model B. (Data are simulated from the model used in Williamson (2000))

Student	**Patient**	**Observed**	**Model A**				**Model B**			
	Number	**Outcome**	1	2	3	4	1	2	3	4
1	1	4	.10	.15	.35	.40	.10	.15	.34	.41
	2	4	.17	.22	.31	.30	.17	.23	.31	.29
	3	2	.08	.16	.21	.55	.06	.12	.22	.60
	4	3	.06	.22	.36	.36	.10	.16	.21	.53
	5	3	.26	.18	.26	.30	.15	.25	.25	.35
2	1	1	.24	.24	.35	.17	.18	.23	.39	.20
	2	2	.63	.18	.12	.07	.59	.20	.13	.08
	3	2	.43	.23	.24	.10	.36	.25	.24	.15
	4	1	.10	.38	.34	.18	.12	.35	.21	.32
	5	3	.44	.19	.17	.20	.56	.22	.11	.11
3	1	3	.14	.17	.41	.28	.10	.17	.47	.26
	2	4	.28	.29	.27	.17	.23	.28	.29	.20
	3	3	.13	.21	.30	.36	.09	.18	.35	.38
	4	2	.07	.29	.43	.21	.10	.44	.24	.22
	5	2	.22	.20	.25	.33	.33	.41	.16	.10

10.1 (Person-By-Observable Fit Indicators for Hypothetical Medical Licensure Exam, Model A). Using the data from Table 10.1 calculate Weaver's surprise index, the logarithmic score, Good's logarithmic score and the ranked probability score for each observable (patient) for each simulee using the predictions from Model A. Which observables are flagged by which fit index?

10.2 (Person-By-Observable Fit Indicators for Hypothetical Medical Licensure Exam, Model B). Using the data from Table 10.1 calculate Weaver's surprise index, the logarithmic score, Good's logarithmic score and the ranked probability score for each observable (patient) for each simulee using the predictions from Model B. Which observables are flagged by which fit index?

10.3 (Person Fit Indicators for Hypothetical Medical Licensure Exam). Using the data from Table 10.1 calculate the average Weaver's surprise index, the logarithmic score, Good's logarithmic score and the ranked probability score for each simulee using the predictions from both models Model B. Which sets of results seem the most consistent with which model? [Hint: Use the results from Exercises 10.1 and 10.2.]

10.4 (Task Fit Indicators for Hypothetical Medical Licensure Exam). Using the data from Table 10.1 calculate the average Weaver's surprise index, the logarithmic score, Good’s logarithmic score and the ranked probability score for each task (simulated patient) using the predictions from both models. Which sets of results seem the most consistent with which model? [Hint: Use the results from Exercises 10.1 and 10.2.]

10.5 (Model Fit Indicators for Hypothetical Medical Licensure Exam). Using the data from Table 10.1 calculate the average Weaver’s surprise index, the logarithmic score, Good’s logarithmic score and the ranked probability score across both simulees and tasks using the predictions from both models. Which model seems to fit the observed results best?

10.6 (Calculating Log Score for Configuration). Suppose that we are given a Bayes net for an assessment and a vector of observed outcomes. What is a fast way of calculating the logarithmic score for that observation vector?

10.7 (Bayes Factor Computation). Suppose that ten students take the hypothetical medical exam, and suppose we calculate the logarithm score for each candidate under both Model A and Model B. The results are given in Table 10.2. Calculate a Bayes factor for comparing Model A to Model B. Which model seems to fit the data better?

Table 10.2 Logarithmic scores for ten student outcome vectors

Model	Candidates									
	1	2	3	4	5	6	7	8	9	10
A	3.167	2.252	2.780	2.241	2.152	3.377	2.652	2.423	3.167	2.932
B	2.936	2.117	3.022	2.019	2.097	3.066	2.682	2.564	3.481	2.807

10.8 (DIC Calculation). Shute et al. (2008) performed a small-scale field trial of the Bayesian network-based system ACED. We subsequently calibrated the model. We looked at two potential prior distributions. The “anchored” prior constrained certain sets of tasks to have average difficulty and discrimination parameters of 0 and 1 respectively. The “unanchored” prior did not add these constraints. After a suitable burn-in, we recorded the deviance from every cycle and calculated the deviance at the average parameter values (across all cycles). The results are shown in Table 10.3.

Calculate the effective dimensionality, p_D for both models. Which model has the higher dimensionality? Why? Calculate the DIC model. Which model is a better fit to the data?

Table 10.3 Deviance values for two ACED models

Model	$\overline{D(\omega)}$	$D(\overline{\omega})$
Anchored	15111.68	15055.45
Unanchored	14495.74	14402.80

10.9 (Local Dependence Test). Consider two random variables X and Y, each of which can take on the values 1 or 0. Suppose that a sample of size n is collected from the joint distribution, and let n_{ij} be the number sampled individuals for which $X = i$ and $Y = j$. The *odds ratio* is defined as:

$$\mathrm{odds}(X, Y) = \frac{n_{11} n_{00}}{n_{01} n_{10}} . \tag{10.22}$$

What is the expected odds ratio when X and Y are independent? Explain how the odds ratio could be used as a test for local dependence between X and Y. (Hint: think about the Mantel–Haenszel test.)

10.10 (Odds Ratio). Consider an assessment that contains two observables, X_1 and X_2 both of which can take on values `correct` and `incorrect`. Suppose further that both observables are conditionally independent of the other proficiency variables given the value of *Skill*. Suppose that an assessment containing these observables was given to 100 student and each student was classified on *Skill* based on the observables other than X_1 and X_2. Tables 10.4(a), (b), (c) give the conditional pairwise relationships among the variables in the observed responses, and Table 10.4(d) gives the marginal observation for the two variables.

Calculate the odds ratio for each conditional table and the one for the overall table. Are they consistent? Explain the differences.

10.11 (A Very Simple PPMC Example). Suppose we have ten observations $\mathbf{y} = \{-1.8, 0.0, 6.0, -0.7, 1.4, -1.2, 0.3, -1.1, 0.9, 1.2, 0.5\}$ from a presumed normal distribution with unknown mean μ and *known* variance 1. The sample mean is $\bar{y} = .5$. The sample standard deviation $s = 2.2$. With a noninformative prior, the posterior for μ is $N(.5, .1)$. To generate a replicate data set, first take a draw from for μ from $N(5, .1)$, say μ^{rep}. Then generate a $\mathbf{y}^{\mathrm{rep}}$ by taking ten independent draws from the predictive distribution of y given $\mu = \mu^{\mathrm{rep}}$, which under the stated assumptions is $N(\mu^{\mathrm{rep}}, 1)$. Writing a program or using a spreadsheet, create 20 replicate data sets. For each such replicate set, calculate statistics such as $\bar{y}^{\mathrm{rep}}$ and s^{rep}, and the highest and lowest observations. Look at their distribution over the replication sets. Compare the observed data to the replicates.

Create a "measure of fit" for each observation, both real and replicate, as the squared distance from the corresponding sample mean. Compare the squared deviations of each observed data point with the distribution of the squared deviates of its replicate counterparts. (Note that the generating dis-

Table 10.4 Observed outcome for two items for Exercise 10.10

(a) *Skill*=`high`

	Correct	Incorrect
Correct	19.00	11.00
Incorrect	7.00	5.00

(b) *Skill*=`medium`

	Correct	Incorrect
Correct	4.00	10.00
Incorrect	5.00	12.00

(c) *Skill*=`low`

	Correct	Incorrect
Correct	1.00	5.00
Incorrect	4.00	17.00

(d) All students

	Correct	Incorrect
Correct	24.00	26.00
Incorrect	16.00	34.00

tributions of all ten replicates y_i are the same, but the distributions of the draws will vary due to sampling.)

Repeat the exercise for 1000 replicate data sets, and compare the results.

10.12 (DTF and DIC). Explain how the DIC statistic could be used to test for differential task functioning.

10.13 (OCP for Word Problem). A certain mathematics assessment contains a number of word problems where the student needs to first create an algebraic expression from an English language description of a problem and then solve the algebraic expression. According to the design of the test, each word problem task requires two skills to solve: *BuildAlgebraicExpression*, *SolveAlgebraicExpression*. Both variables are coded `0` (low proficiency) and `1` (high proficiency). However, because the population to which the assessment will be given includes a number of English language learners, to evaluate the assessment is it given along with a general test of English language proficiency. This produces an additional *EnglishLanguage* proficiency variable, also coded `0`–`1`.

Figure 10.14 shows an observable characteristic plot for one of the word problem tasks for this assessment. Is there a problem which should be brought to the attention of the task writers? What is the best form of the conditional probability table for this observable?

10.14 (OCP for DTF). A researcher is interested in testing the fairness of a certain test toward a group with which there has been a history of discrimination. To assess the fairness of the test, the research administers the selects 100 random students from the focal group and 100 random students from the reference group. The proficiency variable *Skill1* has five levels. The research classified the students into the skill levels on *Skill1* and produced the observable characteristic plot shown in Fig. 10.14. Does this plot provide enough evidence of differential task functioning to warrant further investigation? Justify your conclusions.

10.15 (Computer Skills and CAT). A researcher is interested in whether or not prior computer experience effects a student's score on a new computer

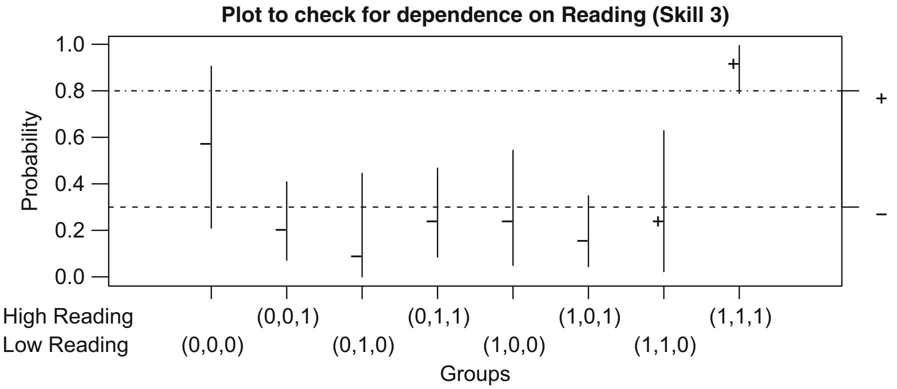


Fig. 10.14 Observable characteristic plot for Exercise 10.13

The skills in this plot, in order of the tuples are *BuildAlgebraicExpression* and *SolveAlgebraicExpression* (both of which are designed to be part of the task) and *EnglishLanguage* (which was not designed to be an explicit part of the task). Reprinted with permission from ETS.

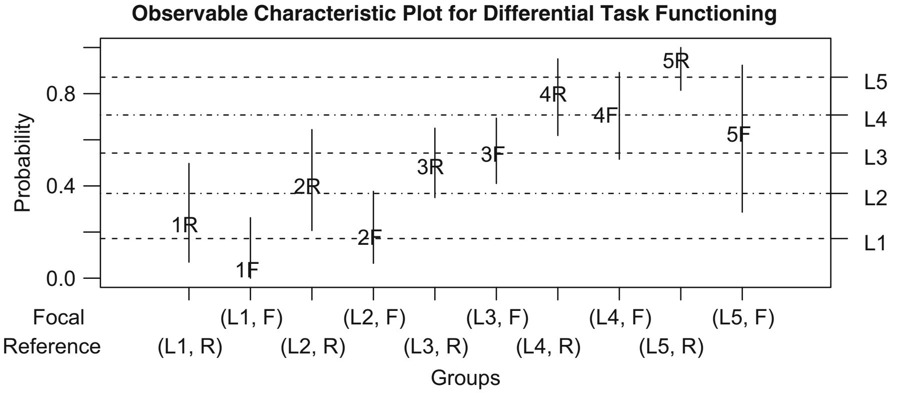


Fig. 10.15 Observable characteristic plot for Exercise 10.14

This plot shows one Skill variable and one group membership variable. The levels of the skill are marked *1*, *2*, *3*, *4*, *5* and the group membership is marked *R* (reference group) and `focal` group. Reprinted with permission from ETS.

delivered assessment. Because the researcher has a limited budget, subject recruitment is done by putting up posters in the computer center and offering free pizza to the first 100 students who agree to take the assessment. Participating students both take the new assessment and answer a questionnaire about prior computer experience. On the basis of these data, the research concludes that there is a small but not significant ($p = .17$) effect of prior computer experience on the assessment. Are these conclusions justified? Explain.

10.16 (Quit Smoking). Freedman et al. (1980) describe the following observational study about smoking: "In 1964, the Public Health Service studied the effects of smoking on health, in a sample of 42,000 households. For men and for women in each age group, they found that those who had never smoked were on average somewhat healthier than the current smokers, but the current smokers were on average much healthier than former smokers."

1. Draw a graph which represents the variables in this study. Include possible hidden variables. Hint: Include variables for currently smoking and who smoked in the past.
2. Upon reading the study somebody suggests, that "The study indicates that smokers should continue to smoke to avoid health problems." Is this conclusion justified by the study?

11

An Illustrative Example

The focus of Part II has been how to build the Bayesian networks. Chapter 8 discussed the issue of how to choose parameterizations for the conditional probability tables that quantify the network. Chapter 9 introduced several techniques for learning the parameters of Bayes nets given a body of assessment data. Chapter 10 suggested several techniques for evaluating how well a proposed network fits the data. This chapter reviews these concepts in terms of an example.

The example we have chosen is the mixed number subtraction example originally collected by Tatsuoka (1983) and introduced in Sect. 6.4. This example is ideal in many respects: It is based on a cognitive analysis of the domain. Both the cognitive analysis and the instructional practices for the domain are closely arranged around a distinguishable set of skills and procedures. Tasks were explicitly built around the features of tasks that do or do not evoke those skills. It has a substantial sample of data collected on a representative sample of the population. It is designed to assess, namely, middle school students currently learning mixed number subtraction (more items, to better determine the unobservable skill variables, would have been nice).

Also, this chapter can lean on our previous work, in particular, the original translation of this model into a Bayesian network by Mislevy (1994); Mislevy (1995b), our use of Markov chain Monte Carlo (MCMC) to estimate the model parameters for this model (Mislevy et al. 1999a), and several experiments with model fit diagnostics using these data (Yan et al. 2004; Sinharay et al. 2004; Sinharay and Almond 2007). (Insightful analyses of Tatsuoka's data from the perspective of cognitive diagnosis include Close et al. (2012); de la Torre and Douglas (2004); Henson et al. (2009); Rupp et al. (2010).)

The structure of this chapter roughly follows the sequence of that prior research. Section 11.1 discusses the issues involved in building the Bayes net for this problem and choosing a parameterization. Section 11.2 discusses using the MCMC algorithm to calibrate the model to the test data, as well as the issues of linking multiple forms. Section 11.3 explores what can be learned from

R. G. Almond et al., *Bayesian Networks in Educational Assessment*,
Statistics for Social and Behavioral Sciences, DOI 10.1007/978-1-4939-2125-6_11

the model checking procedures and proposes an alternative parameterization of the evidence model designed to fix some of the problems.

11.1 Representing the Cognitive Model

Building a Bayesian network to represent a cognitive model proceeds in two steps. The first step (Sect. 11.1.1) is to define the variables and their conditional dependence relationships; that is, the graphical structure of the model. The second step (Sect. 11.1.2) is to choose a parameterization and a prior distribution for the parameters. Section 11.2 then takes up the story of how to adjust the parameters in light of data.

11.1.1 Representing the Cognitive Model as a Bayesian Network

Tatsuoka (1984) developed an assessment of mixed number subtraction skills following a cognitive analysis of the domain (Klein et al. 1981). As in Sect. 6.4, we restrict our attention to students who are using Method B (separate numbers into whole and fractional parts), and the 15 items that did not involve finding a common denominator (allowing us to use a simpler model). Using this method, the 15 problems can be solved using the following five skills:

Skill 1: Basic fraction subtraction.
Skill 2: Simplify/reduce fraction or mixed number.
Skill 3: Separate whole number from fraction.
Skill 4: Borrow one from the whole number in a given mixed number.
Skill 5: Convert a whole number to a fraction.

Students are characterized by a vector $(\theta_1, ..., \theta_5)$ of binary variables, each component indicating whether a student does or does not have *Skill j*. There are relationships among these skills we will want to incorporate in the proficiency model. For example, the prerequisition relationship among skills highlighted in the attribute hierarchy model[1] (Leighton et al. 2004; Gierl et al. 2007) holds implications for task design and inference about examinees. We will see some of them in this example.

Skill 3 is a logical or strong prerequisite for *Skill 4*; it is not possible to borrow one from a whole number to add to a fraction unless one can distinguish and separate these parts of a mixed number. Thus, $P(\textit{Skill 4} = \texttt{Yes}|\textit{Skill 3} = \texttt{No}) = 0$. One way to incorporate this knowledge is to build it into a conditional probability matrix for *Skill 4* given *Skill 3*. Alternatively, we will express this relationship as we did in Sect. 6.4 by introducing a skill *MixedNumber*, denoted θ_{MN}, with three possible states: (0) neither *Skill 3*

[1] The rule space literature calls the aspects of knowledge, skill and ability *attributes*.

nor *Skill 4* present, (1) *Skill 3* present but *Skill 4* absent, and (2) both *Skill 3* and *Skill 4* present. *Skills 3* and *4* are thus logical children of *MixedNumber*.[2]

There are not strong prerequisition relationships between any of the other variables, but there are relationships we should model. They are called weak prerequisites because they are probabilistic rather than deterministic, even though they could be strong empirically. For example, a student can know how to reduce fractions (*Skill 2*) whether or not she can subtract fractions (*Skill 1*), but in this population, subtracting fractions is taught before reducing fractions. Most students who have *Skill 1* are likely to have *Skill 2* than those who do not. In the Bayes net proficiency model fragment, we will model *Skill 2* as a child of *Skill 1* (Fig. 11.1).

Skill 1 is particularly important in this application, because the unit's focus is subtracting mixed numbers in a variety of circumstances. It may be logically possible for a student to be able to reduce fractions without being able to subtract simple fractions, but that's irrelevant to the purpose of this assessment. If *Skill 1* is absent, then the status of the other variables is unimportant; the instructional prescription would be the same in all cases: concentrate on basic fraction subtraction skills. We will see that this purpose holds implications for the mix of tasks as well. Skills 1, 2, 5, and mixed numbers are ordered in the Bayes net roughly according to the order that the skills are introduced, so that we can model weak prerequisite relations we might expect—and information in data can refine them, eliminate them, or counter our expectations.

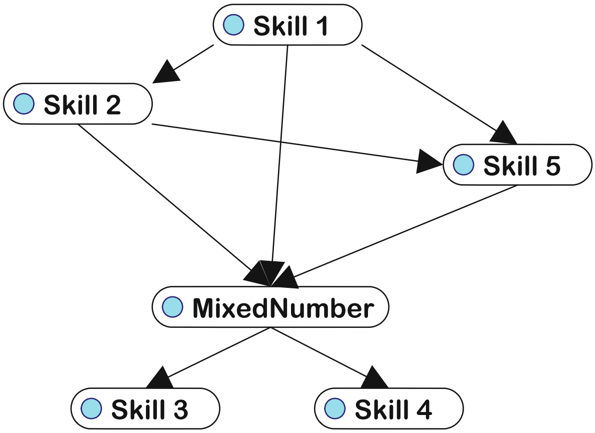


Fig. 11.1 Proficiency model for mixed-number subtraction, method B Reprinted from Sinharay et al. (2004) with permission from ETS.

[2] See von Davier and Haberman (2014) on implications of the alternative representations.

Table 11.1 presents the 15 items used in the example. (To simplify the model, five tasks requiring finding a common denominator, Items 1, 2, 3, 5, and 13 have been omitted.) It includes the Q-Matrix which indicates which skills are used in each task: If Skill j is required for Observable i, then $q_{ij} = 1$, otherwise it is equal to zero. Note that all of the items require *Skill 1*. This assessment cannot provide evidence to support the hypothesis that a student has, for example, *Skill 3* but not *Skill 1*, or any other combination of skills for which *Skill 1* = `No`. But as mentioned above, these distinctions are not germane to the instructional decision at hand.

Table 11.1 Skill requirements for fraction subtraction items

		Skills required					
Item	Text	1	2	3	4	5	EM[a]
6	$\frac{6}{7} - \frac{4}{7} =$	1	0	0	0	0	1
8	$\frac{3}{4} - \frac{3}{4} =$	1	0	0	0	0	1
12	$\frac{11}{8} - \frac{1}{8} =$	1	1	0	0	0	2
9	$3\frac{7}{8} - 2 =$	1	0	1	0	0	3
14	$3\frac{4}{5} - 3\frac{2}{5} =$	1	0	1	0	0	3
16	$4\frac{5}{7} - 1\frac{4}{7} =$	1	0	1	0	0	3
4	$3\frac{1}{2} - 2\frac{3}{2} =$	1	0	1	1	0	4
11	$4\frac{1}{3} - 2\frac{4}{3} =$	1	0	1	1	0	4
17	$7\frac{3}{5} - \frac{4}{5} =$	1	0	1	1	0	4
18	$4\frac{1}{10} - 2\frac{8}{10} =$	1	0	1	1	0	4
20	$4\frac{1}{3} - 1\frac{5}{3} =$	1	0	1	1	0	4
7	$3 - 2\frac{1}{5} =$	1	0	1	1	1	5
15	$2 - \frac{1}{3} =$	1	0	1	1	1	5
19	$7 - 1\frac{4}{3} =$	1	0	1	1	1	5
10	$4\frac{4}{12} - 2\frac{7}{12} =$	1	1	1	1	0	6

[a] This column classifies items with respect to the unique skill patterns they require, which correspond to Evidence models

Recall that a graph can also be represented as a matrix, with a 1 in the cell indicating that there should be an edge between the nodes. Doing this for each unique row in the Q-matrix produces the evidence model fragments, as shown in Fig. 11.1.1. To make the full Bayes net for the assessment, the Evidence model fragments produced from the Q-matrix must be combined with the proficiency model in Fig. 11.1. The Evidence model fragments are replicated (substituting the observable variables) for each item that uses that Evidence model. This produces the network shown in Fig. 11.1.

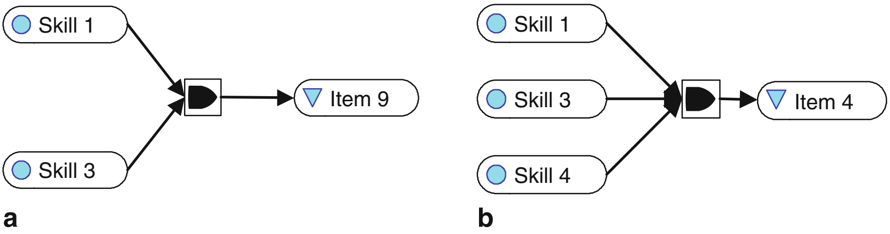


Fig. 11.2 Evidence model fragments for evidence models 3 and 4
a Evidence model 3 (Skills 1 and 3). **b** Evidence model 4 (Skills 1, 3, and 4)

Reprinted from Mislevy et al. (2000) with permission from The National Center for Research on Evaluation, Standards, & Student Testing (CRESST), UCLA.

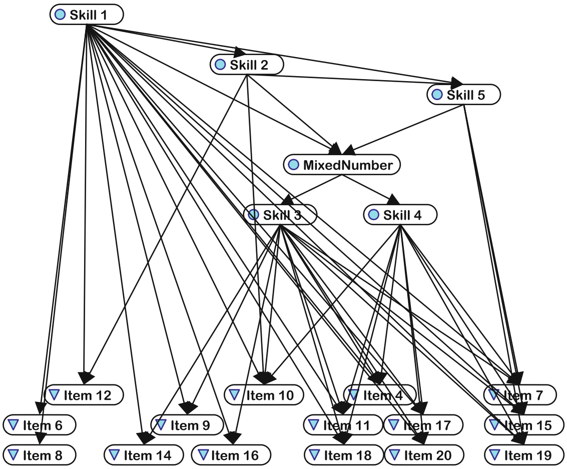


Fig. 11.3 Mixed number subtraction Bayes net
Reprinted from Almond et al. (2006a) with permission from ETS.

The Q-Matrix is a summary representation of the collection of Evidence models for this assessment. As the tasks for this assessment all have a single observable and it depends conjunctively on its set of parent skills, each row of the Q-matrix corresponds to a task-specific Evidence model. Some rows are identical. The tasks in those rows are isomorphs:[3] the mathematical structure, and hence the skills require to solve the problem, are identical, but the individual tasks may use different numbers in the conditional probability distributions. The task models and evidence models for this assessment thus correspond to the unique rows. There are six different unique task types. The

[3] Items that are isomorphic under one method need not be isomorphic under the other. Mislevy (1995b) gives a Bayes net for when it is not known whether a student is using Method A or Method B.

numbers in the last column of Table 11.1 identify the Evidence model for each task.

The Q-matrix therefore shows for each proficiency profile, which evidence/task models should result in correct outcomes, and which should have incorrect outcomes. Table 11.2 shows these ideal response patterns. The conditional probability matrices will characterize, for each task, the tendency of students to make careless errors even if they have all the required skills and to occasionally guess the answer even if they lack one or more of those skills.

Table 11.2 Equivalence classes and evidence models

Equivalence	Evidence model						Class
class	1	2	3	4	5	6	description
1							No Skill 1
2	x						Skill 1 only
3	x		x				Skills 1 & 3
4	x		x	x			Skills 1, 3, & 4
5	x		x	x	x		Skills 1, 3, 4, & 5
6	x	x					Skills 1 & 2
7	x	x	x				Skills 1, 2, & 3
8	x	x	x	x		x	Skills 1, 2, 3, & 4
9	x	x	x	x	x	x	All Skills

An “x” in a given cell indicates that students in the equivalence class corresponding to the row are expected respond correct to tasks from the task/evidence model indicated corresponding to the column. For example, a student in Equivalence Class 8 has Skills 1, 2, 3, and 4 and is expected to make a correct response for the tasks using evidence models 1, 2, 3, 4, and 6, but make an incorrect response for tasks from evidence model 5

Note that there are only nine unique patterns of expected outcomes, even though there are 24 unique proficiency profiles associated with the graph in Fig. 11.1. We call the proficiency profiles that all give rise to the same expected response profile an equivalence class. The first equivalence class contains all twelve proficiency profiles in which *Skill 1* is missing. It would be possible to make distinctions among these profiles, by adding tasks for which students with different profiles within the equivalence had different conditional response probabilities. For example, “Reduce $\frac{6}{8}$ to lowest terms” requires *Skill 2* but not *Skill 1*. Such items distinguish among students who do and do not have *Skill 2* regardless of their standing on *Skill 1*. As Chap. 7 discussed, these test assembly considerations are driven by the hypotheses that are pertinent to test use, and the use of this test places no utility on hypotheses that distinguish among patterns in the “*Skill 1* = `No`” equivalence class.

Most of the other equivalence classes contain only one proficiency profile. The exceptions are Classes 3, 6, and 7, both of which contain two profiles, one with *Skill 5* one without *Skill 5*. The difficulty is that there are no tasks that require *Skill 5* that do not also require *Skill 4*. A closer look at the items (or better yet, reading of Klein et al. (1981)) reveals that in the context of this assessment, *Skill 5* really means being able to move to a mixed number representation in the special case of the minuend being a whole number. Understanding how to do this converts the problem to one more like those in Evidence models 4 and 6. This is why there are no tasks that require *Skill 5* but not *Skill 4*. It is a situationally defined skill, but one that holds important educational value in this unit: The state of having *Skills* 1, 3, and 4 but not *Skill 5* triggers instruction in what to do in this special case.

Over the years, rule space theory (Tatsuoka 1983; Tatsuoka 1984; Tatsuoka 2009; Tatsuoka and Tatsuoka 1989; Tatsuoka 1990; Tatsuoka 1995) and the attribute hierarchy method (Gierl et al. 2007; Leighton and Gierl 2007); Leighton et al. 2004 have developed similar methods for analyzing test forms and their relationships to proficiency profiles. Although the notation differs from the Bayes net notation and the subsequent inferential approaches may differ, there is much to be gained from such analysis. It is relatively simple to translate between the Q-matrix and the graphical structure (Sect. 5.5, Almond 2010a). We can learn quite a lot about the strengths and weakness of an assessment design before we even consider student responses to the question.

11.1.2 Representing the Cognitive Model as a Bayesian Network

Ideally, students with a given proficiency profile should respond according to the pattern implied by the Q-matrix. They should respond correctly to any item for which they possessed all the skills it required, and incorrectly to any item that required one or more skills they did not possess. Real students do not behave like mathematical models. Sometimes students miss items they "should" get right, and get others right when they "should not." False negative responses are errors due to slips and errors of execution, while false positive responses include lucky guesses and answers that happen to be right even though they were obtained through flawed reasoning. The goal of the diagnostic assessment is to classify them into one of the equivalence classes, despite the deviation from "ideal" behavior.

It is here that the Bayesian and rule space approaches part company. The Bayesian approach starts by creating a prior or population distribution over the possible proficiency profiles (a proficiency model) and the likelihood of a positive or negative observed outcome given the proficiency profile. After the outcomes of several tasks are observed, this generative probability model can be inverted using Bayes theorem to draw inferences about the proficiency profile (Chap. 5). Furthermore, the generative model provides a probability for observing the actual data. This can be used to evaluate the fit of the model to the data (Chap. 10).

The rule space method uses a pattern matching approach to classify students. Students are classified into the proficiency profile whose ideal response pattern most closely matches the observed response pattern. (For a more complete explanation, see Tatsuoka and Tatsuoka 1989 or Tatsuoka 1990.) The rule space method does not explicitly define a generative model for the observable outcome variables. This means that most of the methods of Chap. 10 cannot be used to evaluate the fit of the model. Further, coherent probability-based inferences about students through a model are not available. It is not obvious how to estimate parameters, exploit collateral information about tasks and students, deal with missing data, improve the model as data accrues, or extend the analysis to new tasks. For all these reasons, this chapter focuses on the Bayesian network application.

To build the Bayesian model for the mixed number subtraction problem, define the following variables:

> Define the observable outcome variable X_{ij} to be 1 if the response of Examinee i to Item j is correct and 0 if incorrect. Let $\mathbf{X}_i = (X_{i,1}, \ldots, X_{i,J})$, where J is the number of tasks in the assessment, be the outcome vector for Examinee i.
>
> Define the proficiency variable θ_{ik} to be 1 if Examinee i possesses Skill k and 0 if not. Define the proficiency profile for Examinee i as $\boldsymbol{\theta}_i = (\theta_{i,1}, \ldots, \theta_{i,5})$.

The following additional notation will simplify the notation of later expressions:

> Define $\delta_{i,s(j)}$ to be 1 if Examinee i possesses all of the skills required by items in Evidence model s and 0 if not. The notation $s(j)$ is a reminder that which Evidence model is appropriate for Task j is determined by Row j of the Q-matrix. Note that the value of $\delta_{i,s(j)}$ is completely determined by the value of $\boldsymbol{\theta}_i$.
>
> Define $\mathcal{J}_s$ as the set of indices for the tasks that are scored using Evidence model s, and let S be the number of evidence models in the assessment.

At this point we can write the joint distribution of $\boldsymbol{\theta}_i$ and $\mathbf{X}_i$ as

$$\mathrm{P}(\boldsymbol{\theta}_i, \mathbf{X}_i) = \mathrm{P}(\boldsymbol{\theta}_i) \prod_{j=1}^{J} \mathrm{P}(X_{ij}|\boldsymbol{\theta}_i) = \mathrm{P}(\boldsymbol{\theta}_i) \prod_{s=1}^{S} \prod_{j \in \mathcal{J}_s} \mathrm{P}(X_{ij}|\delta_{i,s(j)}) \ . \qquad (11.1)$$

In Eq. 11.1, $\mathrm{P}(\boldsymbol{\theta}_i)$ is the proficiency model and $\mathrm{P}(X_{ij}|\delta_{i,s(j)})$ is (the statistical part of) the evidence model for Item j. So far, this chapter has been a review of material covered in Sect. 6.4. However, fitting the model to the observed data requires first specifying a parameterization for the proficiency and evidence models and a prior distribution for those parameters. These are developed below.

11.1.3 Higher-Level Structure of the Proficiency Model; i.e., $p(\boldsymbol{\theta}|\boldsymbol{\lambda})$ and $p(\boldsymbol{\lambda})$

The proficiency model should be properly thought of as a distribution of proficiency profiles in the population of likely test-takers. Although the domain experts may have some initial guesses as to which skills are common and which are rare, we almost certainly want to refine those guesses with observed data. To refine the proficiency model in a Bayesian context, the experts initial guesses must be expressed as prior distributions over the proficiency model parameters.

Mislevy et al. (1999a) produced a parameterization for the Bayesian network in Mislevy (1995b). They started with a standard assumption that all examinees are exchangeable (before we have observed response data) and therefore a single probability distribution $p(\boldsymbol{\theta}|\boldsymbol{\lambda})$, and a single set of parameters, $\boldsymbol{\lambda}$, is sufficient to describe our beliefs about the proficiency profile $\boldsymbol{\theta}_i$ for any Examinee i.

The independence assumptions implicit in the Bayesian network simplify the task of specifying the joint distribution. With a Bayesian network it is sufficient to specify for each proficiency variable, θ_k, the distribution for that variable conditioned on its parents in the graph (Fig. 11.1).

Mislevy et al. (1999a) used a collection of hyper-Dirichlet distributions (Sect. 8.3) to specify the conditional probability tables. Actually, because all of the variables representing skills are binary, only a single parameter is necessary for each row of the conditional probability table, namely the probability of the skill being present given the state of the parent skills. Modeling the prerequisite relationship between Skills 3 and 4 requires a deviation from this pattern. As mentioned above, Mislevy (1995b) modeled this relationship by introducing a new variable θ_{MN} ("MN" for *Mixed Number*) which takes on three states: `0` for neither Skill 3 nor Skill 4, `1` for Skill 3 only, and `2` for both skills. The conditional probability table for this variable has a conditional multinomial distribution; that is, given the joint state of the parent variables (Skills 1, 2 and 5), the distribution of θ_{MN} is multinomial. Skills 3 and 4 then have logical distributions: conditional probability tables whose entries are all zeros and ones, indicating which of the two skills are mastered.

The parameterized proficiency model, $\mathrm{P}(\boldsymbol{\theta}|\boldsymbol{\lambda})$ can be expressed in the following series of equations:

$$\begin{aligned} \theta_1 &\sim \mathrm{Bern}(\lambda_1) \\ \theta_2\,|\theta_1 = z &\sim \mathrm{Bern}(\lambda_{2z}) \qquad \text{for } z = 0, 1. \end{aligned}$$

That is, there may be different probabilities of having Skill 2 depending on whether a student does or does not have Skill 1; those probabilities are λ_{20} and λ_{21}, respectively.

$$\theta_5 \,|\theta_1 + \theta_2 = z \sim \text{Bern}(\lambda_{5z})\text{for } z = 0, 1, 2.$$

That is, there may be different probabilities of having Skill 5 depending on whether a student has Skills 1 and 2; we allow for different probabilities depending on how many of them the student has: λ_{50} if neither, λ_{51} if just one of them, and λ_{52} if both.

$$\theta_{\text{MN}} \,|(\theta_1 + \theta_2 + \theta_5 = z) \sim \text{Categorical}(\lambda_{\text{MN},z,0}, \lambda_{\text{MN},z,1}, \lambda_{\text{MN},z,2}),$$
$$\text{for } z = 0, 1, 2, 3.$$

As above, the probabilities for θ_{MN} are modeled as depending on Skills 1, 2, and 5, with only the count of those mastered being relevant.

$$\begin{aligned} \theta_3 &= 0 \qquad \text{if } \theta_{\text{MN}} = 0; \\ \theta_3 &= 1 \qquad \text{if } \theta_{\text{MN}} = 1 \text{ or } 2. \\ \theta_4 &= 0 \qquad \text{if } \theta_{\text{MN}} = 0 \text{ or } 1; \\ \theta_4 &= 1 \qquad \text{if } \theta_{\text{MN}} = 2. \end{aligned}$$

Mislevy et al. (1999a) proposed fairly mild but informative Beta priors for the parameters of the Bernoulli distributions, namely, λ_1, λ_{20}, λ_{21}, λ_{50}, λ_{51}, and λ_{52}. The specific prior distribution for λ_1 was Beta$(20, 5)$, indicating that in this population most of the students will possess Skill 1—the prior mean is 80 %, and the effective sample size of the prior is 25 (=20+5) observations.[4] The prior distribution for λ_{21} was also Beta$(20, 5)$, indicating we would expect a student who *does* have Skill 1 to also have Skill 2. However, the prior for λ_{20} was Beta$(5, 20)$, indicating a student who *does not* have Skill 1 also probably does not have Skill 2. Using similar reasoning, we proposed the priors Beta$(5, 20)$, Beta$(12.5, 12.5)$, and Beta$(20, 5)$ for λ_{50}, λ_{51}, and λ_{52}, respectively. For the four three-category distributions, namely, $(\lambda_{\text{MN},z,0}, \lambda_{\text{MN},z,1}, \lambda_{\text{MN},z,2})$, for $z = 0, 1, 2, 3$, Mislevy et al. (1999a) used analogous Dirichlet priors, with parameters that sum to 27 to indicate a weight of 27 observations (this change is relatively small, but makes the numbers divisible by 3), and values that reflect relative frequencies in each category of θ_{MN} depending on whether a student possesses 0, 1, 2, or 3 of the Skills 1, 2, and 5. The actual values of the priors used are as follows:

$$\begin{aligned} (\lambda_{\text{MN},0,0}, \lambda_{\text{MN},0,1}, \lambda_{\text{MN},0,2}) &\sim \text{Dirichlet}(15, 7, 5) \\ (\lambda_{\text{MN},1,0}, \lambda_{\text{MN},1,1}, \lambda_{\text{MN},1,2}) &\sim \text{Dirichlet}(11, 9, 7) \\ (\lambda_{\text{MN},2,0}, \lambda_{\text{MN},2,1}, \lambda_{\text{MN},2,2}) &\sim \text{Dirichlet}(7, 9, 11) \\ (\lambda_{\text{MN},3,0}, \lambda_{\text{MN},3,1}, \lambda_{\text{MN},3,2}) &\sim \text{Dirichlet}(5, 7, 15) \end{aligned}$$

[4] Mislevy et al. (1999a) used Beta$(21, 6)$. The difference in numbers is small.

These numbers show different patterns that reflect a mild expectation that the more "parent" skills a student possessed, the more likely he or she was to have a higher value on the "child" skill.

11.1.4 High Level Structure of the Evidence Models; i.e., $p(\pi)$

All of the tasks in the mixed-number subtraction test yield a single observable outcome variable that can take on the value `right` or `wrong`, coded 1 and 0, respectively. The parents of each observable variable are given by the Q-matrix. All that remains to complete the model is to choose a parametric form for the conditional probability table, and a prior distribution over its parameters.

Mislevy (1995b) proposed an all-or-nothing or DINA model (Sect. 8.4) for the mixed number subtraction example. In this model, there are two cases of interest: either the student has all the skills necessary to solve the problem, $\delta_{i,s(j)} = 1$, or the student lacks one or more of those skills, $\delta_{i,s(j)} = 0$. An all-or-nothing model makes the same prediction for a given item for a student who has none of the skills it requires and a student who has some but not all of the skills.

We expect some deviations from the ideal response patterns. A student who has all of the required skills may still make careless errors or slips, a false-negative response. Define π_{j1} as the probability that a student for whom $\delta_{i,s(j)} = 1$ gets the item correct (the true-positive probability). Similarly, a student who lacks one or more of the required skills may guess the answer or work around the lack of skill. This is a false-positive. Define π_{j0} as the probability that a student for whom $\delta_{i,s(j)} = 0$ gets the item correct (the false-positive probability). We can express the evidence model with the following equation:

$$X_{ij} \left| \delta_{i,s(j)} = z \sim \mathrm{Bern}\left(\pi_{j\delta_{i,s(j)}}\right), \qquad \text{for } z = 0, 1. \right. \tag{11.2}$$

Transforming Eq. 11.2 into a conditional probability table (CPT) for Item j is straightforward. Each row of the CPT corresponds to a configuration of the parent variables, which correspond to either the case $\delta_{i,s(j)} = 0$ or 1. In the former case, the probabilities in that row are $1 - \pi_{j0}$ and π_{j0}. If $\delta_{i,s(j)} = 1$, the conditional probabilities are $1 - \pi_{j1}$ and π_{j1}.

All that remains is to specify for priors for the π's. Again, the Beta distribution is the natural conjugate. As with the proficiency model, we choose beta distributions with effective sample sizes of 25. The actual priors are

$$\pi_{j0} \sim \mathrm{Beta}(5, 20) \ \text{ and } \ \pi_{j1} \sim \mathrm{Beta}(20, 5). \tag{11.3}$$

The mean of the true-positive prior distribution is .8, and the mean of the false-positive prior distribution is .2. These priors are just initial guesses. We expect, and indeed will observe, substantial changes from these prior means to posterior means.

11.1.5 Putting the Pieces Together

The full Bayesian probability model for all the data across all items and examinees, and including all the parameters of higher-level distributions, can now be written as

$$p(\mathbf{X},\boldsymbol{\theta},\boldsymbol{\pi},\boldsymbol{\lambda}) = \prod_i \left(\prod_j p\left(x_{ij} \mid \boldsymbol{\theta}_i, \boldsymbol{\pi}_j\right) p\left(\boldsymbol{\theta}_i \mid \boldsymbol{\lambda}\right) \right) p\left(\boldsymbol{\lambda}\right) \prod_j p\left(\boldsymbol{\pi}_j\right) \ . \quad (11.4)$$

Figure 11.4 is the representation of this model as an acyclic directed graph. The plate notation (see Chap. 8) is used to convey replication over students, evidence models and tasks within evidence models. Note that $\delta_{i,s(j)}$ is a deterministic function (double oval) of the student specific proficiency profile and the evidence model specific Q-matrix row. Note also that the Task plate, j is nested within the evidence model plate, s.

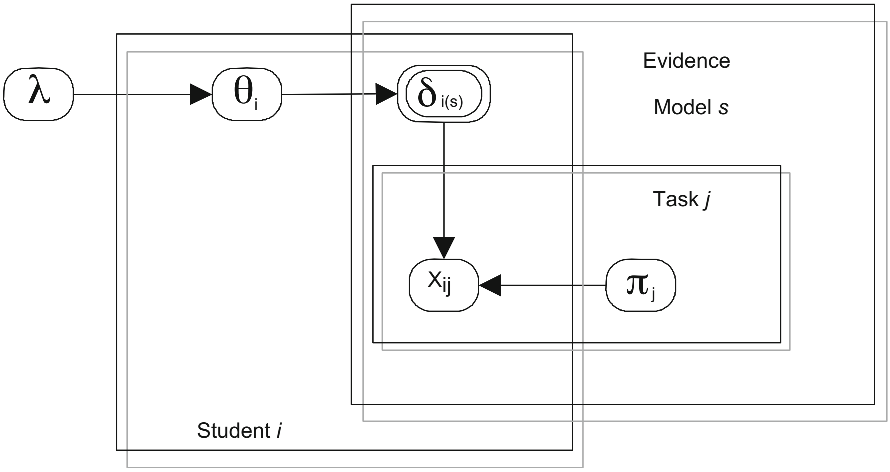


Fig. 11.4 Plate representation of the parameterized mixed-number subtraction model. Reprinted with permission from ETS.

11.2 Calibrating the Model with Field Data

It is straightforward to take the full Bayesian model described in the previous section, and plug it into a MCMC program such as WinBUGS (Lunn et al. 2000) and start turning the Bayesian crank. If we attach it to a sufficiently large power source (say the Three Gorges Dam) we can generate as large a sample of draws as we like from the posterior. What do we gain by doing this Bayesian calibration?

The first thing gained is a new set of adjusted parameters for the model that reflect what is observed in the data. Both the population distribution of skills and the difficulties of the individual tasks may be different from what we expect. Calculating the posterior distribution for the model parameters yields insights into how the observed data differ from our predictions. Section 11.2.1 explores this application. Section 11.2.2 talks about scoring individual students after the calibration.

The same ideas also apply when the same assessment is in use for a long period of time. Often the test developers wish to add new tasks and retire old tasks (usually for security reasons). It is possible that the population characteristics and task-specific evidence model parameters will change over time. Section 11.2.3 shows how calibration can used to link past and present versions of the assessment.

The final reason to sample from the posterior distribution is to evaluate how well the model fits the data, and refine the model on that basis. Section 11.3 explores some possible model checking strategies.

11.2.1 MCMC Estimation

Equation 11.4 gives the joint prior distribution of all data and parameters for the mixed-number subtraction example. Once data $\mathbf{X}$ are observed, applying Bayes theorem yields the joint posterior distribution. The posterior distribution for $\boldsymbol{\theta}$, $\boldsymbol{\pi}$, and $\boldsymbol{\lambda}$ is proportional to Eq. 11.4, but *with the* X_{ij}*'s fixed at their observed values*:

$$p\left(\boldsymbol{\theta}, \boldsymbol{\pi}, \boldsymbol{\lambda} \mid \mathbf{X}\right) \propto \prod_i \left(\prod_j p\left(x_{ij} \mid \boldsymbol{\theta}_i, \boldsymbol{\pi}_j\right) p\left(\theta_i \mid \boldsymbol{\lambda}\right) \right) p\left(\boldsymbol{\lambda}\right) \prod_j p\left(\boldsymbol{\pi}_j\right) \ . \tag{11.5}$$

Using MCMC (Chap. 9), we can draw a sample from this posterior without needing to explicitly calculate the normalization constant. Statistics of the posterior distribution can then be approximated by sample statistics of the posterior sample. Mislevy et al. (1999a) used BUGS (Thomas et al. 1992; Lunn et al. 2000) to draw a sample from the posterior. This chapter presents a reanalysis using StatShop (Almond, Yan, et al. 2006c) and the R package `coda` (Plummer et al. 2006; R Development Core Team 2007). The differences between the two analyses are minor.

MCMC analysis starts one or more Markov chains from arbitrarily chosen starting points. Although the stationary distribution of these chains is the posterior, it may take a while to reach stationarity. Therefore, a number of "burn-in" cycles are produced and discarded at the the beginning of each chain. After that point, the distribution of a large number of draws for a given parameter approximates its marginal distribution, and summaries such as posterior means and variances can be calculated. Gelman and Rubin (1992) (also Brooks and Gelman 1998) recommend running chains from multiple starting points and then looking at the ratio of between chain and within chain variance to verify that stationary has been reached. Graphing of the Gelman–Rubin potential scale reduction factor convergence criterion for each parameter shows as what point the the ratio of variance among chain means matches the pooled variance of draws for that parameter within chains. Values of around 1.1 or 1.2 or less are considered satisfactory (we have seen values as high as 20 in analyses with multiple posterior modes and identification problems) as long as other evidence of nonconvergence, such as visually disparate traces, are not present.

StatShop offers six different strategies for choosing starting values for the MCMC chains based on the prior distribution:

Midpoint: Use the prior mean or median as starting values.

High: Use the mean plus twice the standard deviation of the prior.

Low: Use the mean minus twice the standard deviation from the test run posterior.

HighLow: Partition the parameters into two sets "up" and "down." For the up set, use the mean *plus* twice the standard deviation; for the down set, use the mean *minus* twice the standard deviation.

LowHigh: Same as Step 4, with the "up" and "down" sets reversed.

Random: Use values drawn randomly from the prior.

If we approximate the prior distribution with a multivariate normal distribution, using all of the first five strategies picks as starting points the center of the prior, and 4 points on the 95 % ellipsoid. The analysis shown below used five chains using the first five methods. In practice, three chains are often sufficient, and our standard practice has evolved into using three chains with the first three methods.

We start five chains and run them for 3000 cycle each. We know that the first few cycles are not in the stationary distribution, but we hope that the last cycles are. The next step is to decide if the chains have converged to the stationary distribution, and if they have, how many cycle to discard as burn-in. The Gelman–Rubin R statistic provides one measure of convergence. We can calculate the R statistic for any window for the series (for example each 100 cycles); the usual heuristic is to call the chains converged when the value of R drops below 1.1. One technique useful for determining the point of convergence is to plot the R statistic against the number of cycles (Fig. 11.5 shows this for selected parameters). We see high values at the beginning of the run, reflecting the widely dispersed starting points of the five chains, and values settling down to less than 1.1 by 500 iterations. Typically, we look at the proficiency model parameters first (e.g., Fig. 11.5a and b), as when models have serious problems, they are likely to be apparent in lack of convergence of the proficiency model parameters. If these look okay (e.g., $R < 1.1$ and no clear distinctions in trace lines for different chains), then we go to the evidence model parameters, as occasionally one or two evidence models has difficulty converging even if the rest of the model looks good.

Based on the selected parameters, it looks like any value over 500 would be adequate since the R values are all comfortably below 1.1 by then. The analysis below uses 1000 to be conservative.[5] We confirm this by calculating the R statistic for each parameter in the model for the window of $(1001, 3000)$. The maximum value is 1.01, so convergence to stationarity is very likely. An alternative would be to look at the multivariate R (Brooks and Gelman 1998) for each set of parameters associated with each conditional probability table.

Converging to the stationary distribution is not enough; in addition, the chain must mix well: the sample must be long enough that the chance of visiting any possible value for the joint posterior is roughly equal to its posterior probability. There are two potential problems here. First, the sample could get "stuck" in a local maxima of the posterior and never explore other potential maxima. Starting multiple chains from well dispersed starting values guards against that problem. The second potential problem is that the chain moves slowly through the posterior distribution. This is the *slow mixing* phenomenon discussed in Sect. 9.5.2. In this case, a large MCMC sample is needed to produce an unbiased estimates of descriptive statistics of posterior distributions.

The trace or history plot—a plot of the values in the time series against time—is a robust tool for diagnosing a number of problems, including slow mixing. Figure 11.6 shows the MCMC chain histories for selected parameters. Values from the five chains are plotted on top of one another, so that if there were a convergence problem, the chains would appear separate. In Fig. 11.6, all five chain lie on top of one another; this confirms the earlier findings of convergence. Large gaps in the series, especially behavior that looks like cycles (although they will have irregular periods), is a sign of slow mixing. If the chains are mixing well, the trace plots will look like *white noise* (a series of independent draws from a normal distribution) with no discernible patterns. The series from the mixed-number subtraction example all look like this, so we know the chains are mixing well.

If the chains were mixing slowly, they would need to be run longer to make sure that they cover the parameter space in proportion to the posterior they are trying to approximate. Note that plotting the new longer series in the same size graphic window will compress the x-axis. One sign that the MCMC

[5] Be careful here if using BUGS and the Metropolis sampling method. BUGS uses the first 4000 cycles to adjust the size of the proposal distribution. During the adjustment period, the chain is not guaranteed to have the right stationary distribution, so those cases require the burn-in to be at least 4000 cycles. This is not required in this case because: (a) With all conditional probability tables in the model using the hyper-Dirichlet design pattern, Gibbs sampling works well, and does not require adaptation. (b) StatShop, unlike BUGS, uses a fixed proposal distribution.

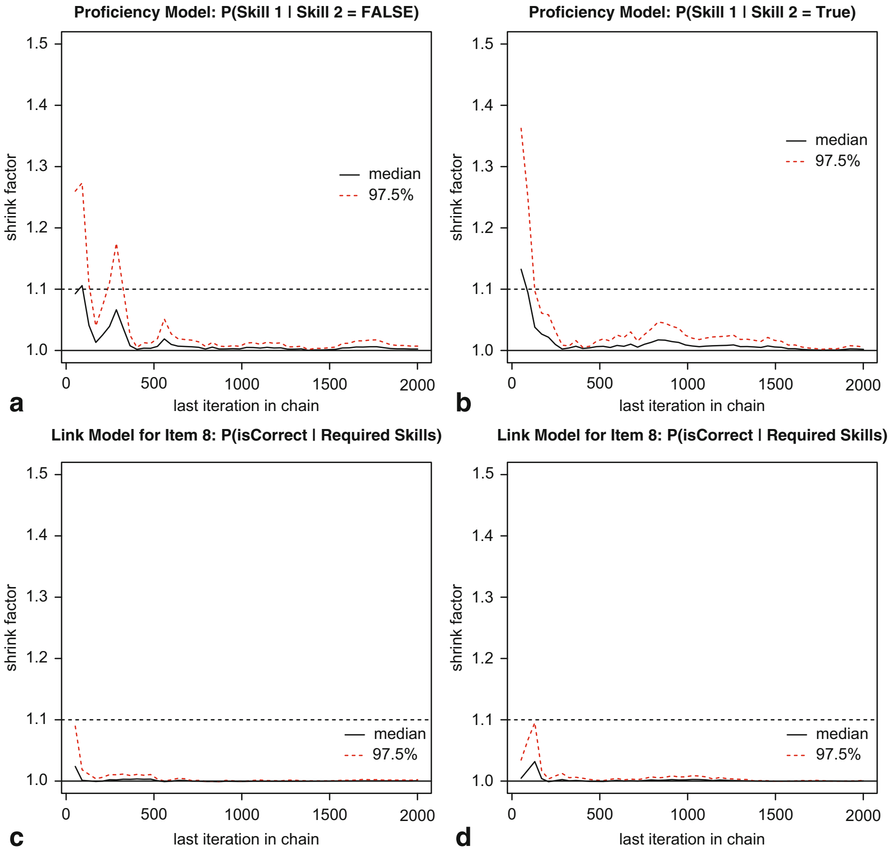


Fig. 11.5 Gelman–Rubin potential scale reduction factors for selected parameters model (five chains; first 2000 updates each). **a** R for $\lambda_{2,0}$. **b** R for $\lambda_{2,1}$. **c** R for $\pi_{8,0}$. **d** R for $\pi_{8,1}$. Reprinted with permission from ETS.

sample is sufficiently large is that the trace plot on the new compressed series now looks like white noise.

Another way to judge how well the chains are mixing is to look at the autocorrelation of the series—the correlation of the series with itself a previous time. The correlation between $x^{(t)}$ and $x^{(t-\ell)}$ is called the *autocorrelation at lag* ℓ. Usually, the autocorrelation is a decreasing function of lag, hopefully it dies out fairly quickly. The autocorrelation at lag 5 is usually a pretty good indicator of how quickly the chain is moving. Autocorrelations and mixing behavior can be different for different parameters in the same problem. High autocorrelations and slow mixing can occur when there is not much information in the data about a parameter, a parameter is unidentified, or when sets of parameters are nearly multicollinear.

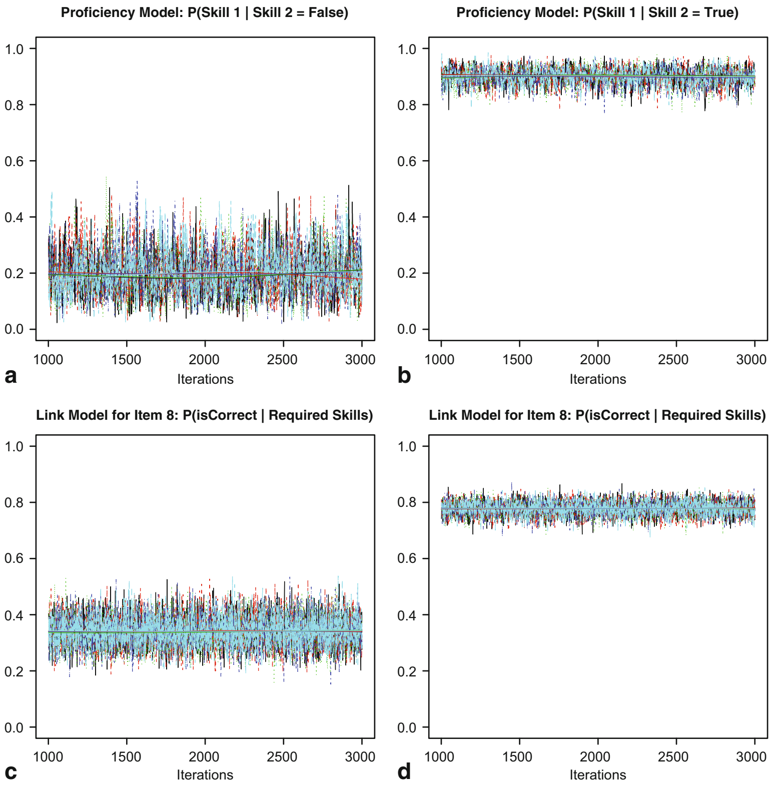


Fig. 11.6 Histories of MCMC chains for selected parameters (five chains; cycles 1001–3000 from each chain). **a** Trace plot for $\lambda_{2,0}$, **b** Trace plot for $\lambda_{2,1}$, **c** Trace plot for $\pi_{8,0}$, **d** [Trace plot for $\pi_{8,1}$. Reprinted with permission from ETS.

A related statistic is the effective MCMC sample size of the chain (Eq. 9.21). If this were a simple random sample from the posterior, there would be 10,000 observations (2000 each from five chains). However, the samples are correlated, so the MCMC estimation error will be larger than that for an uncorrelated sample of the same size. This should be at least several hundred samples. If not, we would want to run the chain longer until the effective sample size is high enough.

The autocorrelation and effective MCMC sample size are closely related. The lowest lag 5 autocorrelation, 0.0038, and the highest MCMC sample size, 8163, appear in the conditional probability tables for Item 18. The highest lag 5 autocorrelation, 0.4626, and the lowest MCMC sample size, 812, appear in the conditional probability tables for *Skill 5* in the proficiency model. These values do not look too bad, so there is no reason to run the chain further.

One trick that is often suggested to reduce autocorrelation is to "thin" the chain, that is to only record every 3rd, every 5th or every 10th cycle.

Although this reduces the autocorrelation in the file that is produced, it also reduces the size of the sample, so the result is often a net loss of effective sample size, without substantially reducing the amount of time required to compute the sample. If the chains are very large, though, thinning will produce smaller series that take up less storage space on the computer and are faster to work with in postprocessing tasks such as producing summary statistics and graphics. The loss of information in the full chains will then be outweighed by the convenience of working with the thinned chains.

The tests we have done above were all for the purpose of convincing ourselves that the sample of 10,000 draws (cycles 1001–3000 from each of five chains) are, in the aggregate, a good approximation of to the posterior distribution. Given that the sample looks good, what does the posterior distribution look like?

The easiest thing to look at is the marginal distribution of each parameter. The trace plot gives us a quick impression of the range and typical values of each series. For example, comparing Fig. 11.6a and b reveals that $\lambda_{2,1} > \lambda_{2,0}$, which corresponds to our expectations. (The variance of $\lambda_{2,0}$ is also larger; this is discussed below). Similarly, comparing Fig. 11.6a and b reveals that $\pi_{8,1} > \pi_{8,0}$, again confirming our modeling assumptions.

Although we can see the range of values each parameter typically takes from the trace plot, it is often easier to estimate the density more directly. At this point, we can pool the data from all five chains to do that estimation. Figure 11.7 shows the posterior distributions of the selected parameters. The probability of having Skill 2 when Skill 1 is present, $\lambda_{2,1}$, has a fairly tight distribution distribution centered nicely at .9. The probability of having Skill 2 when Skill 1 is *not* present, $\lambda_{2,0}$, is centered at .2, with a much larger variance. The true-positive probability of a correct response to Task 8 given $\delta_{(4)} = 1$ is centered at .78, while the false-positive of a correct response when $\delta_{(4)} = 0$ is centered at .34 with about twice the variance. This indicates that there were probably fewer students in the latter condition than in the former.

The marginal posterior distributions are smooth and unimodal, which indicates that the posterior is fairly well behaved. The shapes of the posterior distributions look a lot like beta distributions. Thus, a beta distribution with the same mean and variance is likely to be not too bad an approximation to the posterior. This will be useful for linking this assessment form with another form (Sect. 11.2.3).

When the posterior is generally smooth and especially when it is unimodal, it is often easier to look at the marginal distributions in tabular form than as a series of graphs. Table 11.3 provides the estimated posterior means, standard deviations, and selected quantiles from this calibration run. Looking at the values, we can see that most have moved from their original prior distributions. There are a few exceptions. Note in particular, that both the prior and posterior mean for λ_{20} is .2, and both the prior and posterior standard deviation is 0.078. The calibration has not changed our estimates of this

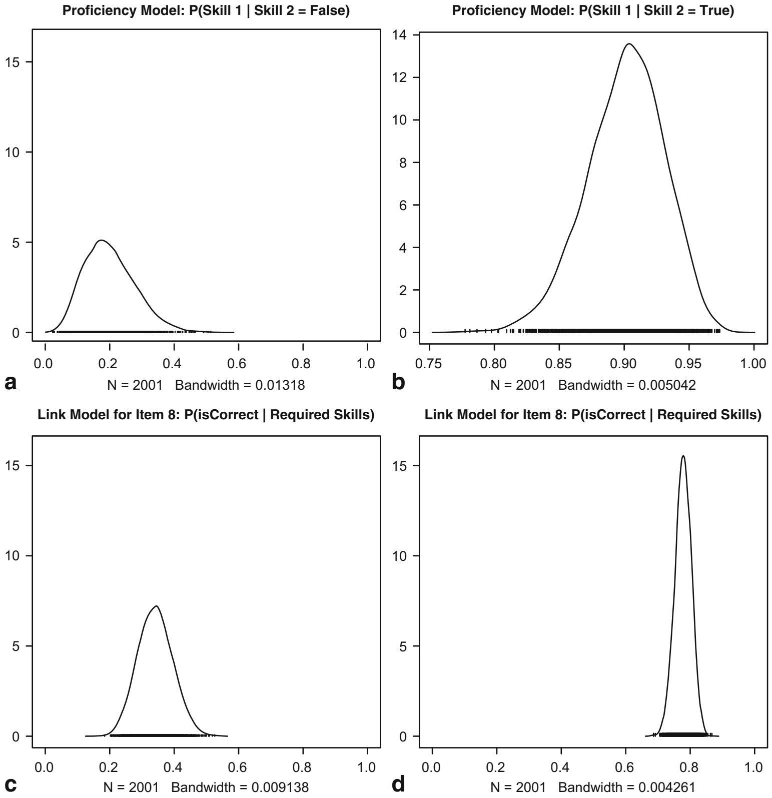


Fig. 11.7 Posterior distributions from MCMC chains for selected parameters (five chains; 2000 samples each chain). **a** Posterior for $\lambda_{2,0}$, **b** Posterior for $\lambda_{2,1}$, **c** Posterior distribution for $\pi_{8,0}$, False Positive, **d** Posterior distribution for $\pi_{8,1}$, True Positive. Reprinted with permission from ETS.

parameter at all. This is because of the structure of the Q-matrix: There are no items that would provide evidence about Skill 2 in the absence of Skill 1. Thus, the prior and posterior are equal. Although this is less than ideal, it does not present a practical issue. A teacher would use the same instructional strategy—teach basic fraction subtraction—for all students who lack Skill 1.

11.2.2 Scoring

Once the model is calibrated, we can use it to draw inferences about individual students—to score them, in common parlance. For the students in the calibration process there are two approaches to scoring.

The first approach is to use the distribution of the proficiency variables for a given student in the MCMC sampler as the posterior distribution for that student. The MCMC sampler can be augmented to include any statistic

Table 11.3 Summary statistics for binary-skills model

Parameter	Mean	SD	2.5 %	50 %	97.5 %	Parameter	Mean	SD	2.5 %	50 %	97.5 %
λ_1	0.810	0.025	0.759	0.811	0.857						
λ_{20}	0.200	0.078	0.071	0.193	0.374	λ_{21}	0.901	0.030	0.838	0.903	0.954
$\lambda_{\mathrm{MN},0,1}$	0.256	0.082	0.111	0.250	0.430	$\lambda_{\mathrm{MN},0,2}$	0.184	0.073	0.065	0.176	0.349
$\lambda_{\mathrm{MN},1,1}$	0.367	0.090	0.197	0.364	0.548	$\lambda_{\mathrm{MN},1,2}$	0.235	0.075	0.107	0.228	0.397
$\lambda_{\mathrm{MN},2,1}$	0.415	0.088	0.245	0.414	0.587	$\lambda_{\mathrm{MN},2,2}$	0.451	0.076	0.308	0.450	0.605
$\lambda_{\mathrm{MN},3,1}$	0.458	0.058	0.331	0.463	0.561	$\lambda_{\mathrm{MN},3,2}$	0.486	0.054	0.391	0.482	0.601
λ_{50}	0.165	0.068	0.057	0.158	0.319	λ_{51}	0.483	0.094	0.303	0.482	0.665
λ_{52}	0.724	0.063	0.584	0.730	0.830						
$\pi_{4,0}$	0.085	0.019	0.051	0.084	0.126	$\pi_{4,1}$	0.870	0.029	0.808	0.872	0.921
$\pi_{6,0}$	0.189	0.055	0.089	0.187	0.303	$\pi_{6,1}$	0.924	0.017	0.888	0.925	0.954
$\pi_{7,0}$	0.151	0.023	0.109	0.151	0.198	$\pi_{7,1}$	0.830	0.041	0.745	0.833	0.903
$\pi_{8,0}$	0.341	0.054	0.238	0.341	0.451	$\pi_{8,1}$	0.778	0.025	0.727	0.779	0.826
$\pi_{9,0}$	0.475	0.052	0.376	0.475	0.576	$\pi_{9,1}$	0.740	0.028	0.684	0.741	0.793
$\pi_{10,0}$	0.042	0.014	0.019	0.040	0.073	$\pi_{10,1}$	0.831	0.036	0.757	0.833	0.896
$\pi_{11,0}$	0.079	0.018	0.047	0.078	0.119	$\pi_{11,1}$	0.872	0.029	0.810	0.874	0.924
$\pi_{12,0}$	0.148	0.044	0.070	0.146	0.241	$\pi_{12,1}$	0.905	0.024	0.855	0.906	0.949
$\pi_{14,0}$	0.183	0.050	0.094	0.181	0.285	$\pi_{14,1}$	0.926	0.018	0.888	0.928	0.959
$\pi_{15,0}$	0.206	0.026	0.157	0.206	0.259	$\pi_{15,1}$	0.862	0.035	0.789	0.864	0.923
$\pi_{16,0}$	0.196	0.047	0.109	0.194	0.294	$\pi_{16,1}$	0.907	0.021	0.864	0.908	0.945
$\pi_{17,0}$	0.076	0.018	0.044	0.075	0.115	$\pi_{17,1}$	0.815	0.033	0.746	0.817	0.876
$\pi_{18,0}$	0.191	0.034	0.129	0.189	0.262	$\pi_{18,1}$	0.809	0.034	0.738	0.811	0.871
$\pi_{19,0}$	0.046	0.014	0.023	0.045	0.077	$\pi_{19,1}$	0.884	0.039	0.801	0.888	0.950
$\pi_{20,0}$	0.035	0.013	0.014	0.033	0.063	$\pi_{20,1}$	0.811	0.034	0.739	0.812	0.873

of the proficiency variables that may be of interest. We obtain the posterior probability that the student has mastered each of the skills. Further, in this example, the sum of the proficiency variables excluding θ_{MN} provides a count of the number of skills acquired by the student, which is a good summary of overall proficiency in the domain of tasks. The advantage of this approach is that it fully accounts for the uncertainty about the higher level parameters for the tasks and the population distribution. The disadvantage is that it only works for students in the calibration sample.

The second approach is to drop the estimated parameters from the calibration into the Bayesian network. To score students with the calibrated model, the easiest approach is drop the posterior means for each conditional probability table into the Bayesian network. Then, the algorithms of Chap. 5 can be used to score the students. Inference about individual students in the mixed-number subtraction example proceeds as in Sect. 6.4. (If we approximate the posterior distribution for each row of a conditional probability with a beta distribution, the resulting model can be used either for scoring students or for additional calibrations.) Although this approach ignores the uncertainty about the parameters, the numerical differences are slight if the higher-level parameters have been estimated with sufficient accuracy. More importantly for practical work, it works with students who were not in the calibration sample.

Table 11.4 shows the responses of nine selected students, grouped by evidence models. Table 11.5 shows how, based on their responses, these examinees' skill-possession probabilities changed from the population rates that serve as priors for students in the population before their responses are observed.

Table 11.4 Selected student responses

		Student number								
EM	Item	26	32	35	36	47	94	127	156	315
1	6	1	1	1	1	1	1	1	1	0
	8	1	1	1	0	1	1	1	1	0
2	12	1	1	1	1	1	1	0	1	1
3	9	1	0	1	0	1	1	0	1	0
	14	1	1	1	1	1	1	0	1	0
	16	1	1	1	1	1	1	0	1	1
4	4	1	1	1	0	1	1	0	1	0
	11	1	1	1	1	0	0	1	1	0
	17	0	1	0	1	0	0	0	1	1
	18	1	0	1	0	0	0	0	0	1
	20	1	0	1	1	0	0	0	1	1
5	7	0	1	0	0	1	0	0	0	0
	15	1	1	0	0	1	0	0	0	0
	19	0	1	0	1	1	0	0	0	0
6	10	1	0	1	1	0	0	0	0	0
Total		12	11	11	9	10	7	3	10	5

1 indicates a correct response to the item;
0 indicates an incorrect response

Except for Student 315, all of the selected examinees correctly answered the questions from evidence model 1, providing evidence that they have Skill 1. Students 26, 32, 35, and 36 also answered most of the items requiring Skills 2, 3, and 4 correctly, so they are very likely to have Skills 2, 3 and 4 as well. Despite missing a few items, Student 32 is likely to have all of the skills. (This is one place where we would like to have had more items, so the data could better distinguish these competing explanations.) As her incorrect responses are not systematic with respect to the skill items required, they are more apt to be caused by slips than missing skills. Student 94 shows evidence of not having Skill 4 and probably not Skill 5 either. The posterior probabilities for Students 35 and 156 indicate that they are very likely to have Skills 1, 2, 3, and 4, but not Skill 5.

Table 11.5 Prior and posterior probabilities for selected examinees

Student	Skill				
	1	2	3	4	5
Prior probabilities					
All students	0.883	0.618	0.921	0.396	0.313
Posterior probabilities					
26	1.000	0.987	1.000	0.999	0.391
32	1.000	0.946	1.000	0.993	0.976
35	1.000	0.980	1.000	0.998	0.038
36	1.000	0.982	1.000	0.989	0.310
47	1.000	0.741	1.000	0.254	0.403
94	1.000	0.671	1.000	0.007	0.204
127	0.920	0.128	0.167	0.011	0.363
156	1.000	0.687	1.000	0.981	0.031
315	0.485	0.478	0.811	0.370	0.103

11.2.3 Online Calibration

Much applied psychometrics is concerned with measurement issues related to testing programs—ongoing series of assessments designed to measure the same set of proficiencies. To support a testing program, the psychometrician must calibrate a series of variant forms of the same essential assessment. These forms may differ in several ways: (1) they may contain new tasks from the existing task models, (2) the testing population may have changed (due to changes in educational policy or cohort effects), or (3) the conceptual assessment framework may have changed, adding new task models or even new proficiencies. We will use the mixed number subtraction data to simulate the first two cases. The third is more challenging, but can be tackled with the same modeling ideas we have been discussing.

Following Mislevy et al. (1999a), we create two overlapping forms of an assessment by dropping three tasks from each one. The Admin 1 form drops Tasks 16, 20, and 19, from evidence models 3, 4, and 5. The Admin 2 form drops Tasks 14, 18, and 7, from the same evidence models. Thus, the two forms are parallel (having the same number of tasks from each Evidence model) and each have 12 tasks. The nine tasks that appear on both forms constitute an anchor set that will help link the two forms during calibration. Note that only tasks from evidence models with three or more instances in the original form were dropped to produce the Admin 1 and Admin 2 forms. Thus, the set of distinct Q-matrix rows is the same for the anchor test and both Admin forms.

In high-stakes assessments, psychometricians often try to *equate* the two forms; that is, try to ensure that scores from both forms have identical meaning (Kolen and Brennan 2004). Strict equating is not necessary in the lower stakes diagnostic setting that the mixed number subtraction test is designed for, but it still is necessary to *link* the two assessments, to ensure that the proficiency variables in both are at least on roughly the same scale.

The easiest way to link the assessments is to simply calibrate them together—all data from all administrations at once, with not-presented items coded as missing at random. This is called concurrent calibration. Both the Evidence model and MCMC algorithms are quite robust to missing data as long as it meets the missing-at-random assumption. Both the Evidence model and MCMC algorithms depend on the number of students in the sample and can be slow, although this restriction is becoming less of a problem as computing power continues to increase.

Mislevy et al. (1999a) proposed a less computationally intensive alternative that does not require item responses to previous administrations, and can be chained across multiple administrations. It uses a three-step procedure:

1. Calibrate the data from the first administration normally (Sect. 11.2.1).
2. Approximate the posterior distribution of the structural parameters with a parametric distribution with a convenient form (e.g., Beta, Dirichlet, or Normal distributions).
3. Calibrate the data from the second administration using the approximate posterior calculated in the previous step as a prior.

To demonstrate this procedure, we randomly divided the original sample of 325 students using Method B, assigning 225 to Admin 1, and the remaining 100 to Admin 2. Dropping Tasks 16, 20, and 19 from Admin 1 student records and Tasks 14, 7, and 18 from Admin 2 student records produced data that is fairly typical for multiple administrations. These data are analyzed below.

The calibration of the first administration data is not qualitatively different from the analysis of the whole data set in the previous section. As there are no problems in the model, it converged easily, and the posterior statistics conditioned on the Admin 1 data are shown in Table 11.6. Comparing this to the full data posterior, Table 11.3, shows that the posterior means differ by less than 0.02 for almost all parameters (except the three dropped items) and the posterior standard deviations are slightly larger.

To link the second administration to the first, we calibrate the second administration's data using the posterior from Admin 1 as a prior for Admin 2. Fixing the values of common parameters at their Admin 1 posterior means (or medians) during the Admin 2 calibration ignores an important contribution to our overall uncertainty: our uncertainty are the values of the parameters. This expedient biases the estimates of the new item parameters.

The easiest way to represent the new prior is to simply include both the old and the new data set in the second calibration. The MCMC technique appropriately handles missing data that are "missing at random" in the sense of Rubin (1977), and both the values of the latent variables and the responses of not-administered items are, so everything should work well. This method produces an accurate representation of the prior (within the accuracy of the MCMC algorithm). With a data set this small, this full calibration technique does not present a problem. However, as the data get larger and larger, the

Table 11.6 Summary statistics for binary-skills model, Admin 1

Parameter	Mean	SD	2.5 %	50 %	97.5 %	Parameter	Mean	SD	2.5 %	50 %	97.5 %
λ_1	0.822	0.027	0.766	0.823	0.873						
λ_{20}	0.198	0.076	0.075	0.191	0.362	λ_{21}	0.890	0.034	0.817	0.893	0.951
$\lambda_{\mathrm{MN},0,1}$	0.260	0.082	0.116	0.256	0.433	$\lambda_{\mathrm{MN},0,2}$	0.187	0.074	0.067	0.179	0.354
$\lambda_{\mathrm{MN},1,1}$	0.370	0.092	0.200	0.367	0.554	$\lambda_{\mathrm{MN},1,2}$	0.219	0.074	0.094	0.212	0.378
$\lambda_{\mathrm{MN},2,1}$	0.401	0.090	0.231	0.399	0.578	$\lambda_{\mathrm{MN},2,2}$	0.415	0.082	0.261	0.413	0.578
$\lambda_{\mathrm{MN},3,1}$	0.389	0.054	0.279	0.391	0.492	$\lambda_{\mathrm{MN},3,2}$	0.541	0.052	0.442	0.540	0.647
λ_{50}	0.166	0.067	0.057	0.158	0.315	λ_{51}	0.506	0.098	0.314	0.506	0.696
λ_{52}	0.797	0.059	0.667	0.801	0.901						
$\pi_{4,0}$	0.097	0.024	0.054	0.095	0.149	$\pi_{4,1}$	0.890	0.032	0.821	0.892	0.944
$\pi_{6,0}$	0.176	0.059	0.075	0.171	0.300	$\pi_{6,1}$	0.933	0.018	0.894	0.935	0.965
$\pi_{7,0}$	0.153	0.030	0.098	0.152	0.215	$\pi_{7,1}$	0.806	0.046	0.710	0.809	0.893
$\pi_{8,0}$	0.327	0.062	0.212	0.324	0.455	$\pi_{8,1}$	0.802	0.029	0.742	0.803	0.855
$\pi_{9,0}$	0.499	0.059	0.384	0.499	0.613	$\pi_{9,1}$	0.770	0.031	0.707	0.771	0.828
$\pi_{10,0}$	0.065	0.021	0.030	0.063	0.111	$\pi_{10,1}$	0.819	0.040	0.735	0.821	0.890
$\pi_{11,0}$	0.070	0.021	0.034	0.068	0.115	$\pi_{11,1}$	0.884	0.033	0.812	0.886	0.942
$\pi_{12,0}$	0.129	0.045	0.054	0.125	0.228	$\pi_{12,1}$	0.907	0.028	0.847	0.908	0.956
$\pi_{14,0}$	0.154	0.054	0.062	0.150	0.272	$\pi_{14,1}$	0.952	0.017	0.912	0.954	0.981
$\pi_{15,0}$	0.180	0.031	0.123	0.179	0.244	$\pi_{15,1}$	0.837	0.044	0.745	0.840	0.916
$\pi_{17,0}$	0.079	0.022	0.041	0.077	0.126	$\pi_{17,1}$	0.810	0.039	0.727	0.812	0.882
$\pi_{18,0}$	0.182	0.031	0.125	0.181	0.246	$\pi_{18,1}$	0.823	0.037	0.746	0.825	0.890

MCMC chain will get slower and slower. Therefore, it is useful to be able to represent the new prior more compactly.

One idea is to represent the posterior with a distribution that has the same functional form as the prior, but different parameters. The idea is similar to the use of conjugate priors, but now as an approximation rather than an exact form of updating. In this model, almost all of the prior are beta distribution (with one Dirichlet distribution). The distributions in Fig. 11.7 are roughly beta-shaped, so this approximation is probably not too bad. One word of caution: Figure 11.7 is only showing the margins of the posterior distribution. The complete distribution spans all of the parameters in the model, and even if they are *a priori* independent, they may be dependent *a posteriori*. York (1992) notes this happens when there are missing data in the model—and in this case, the latent proficiency variables are missing for everybody! For the purposes of computational simplicity, we will ignore the dependence. The effects of the dependency decrease as the sample size increases.

The *method of moments* presented in Sect. 9.6.2 is the easiest way to fit a beta distribution or a Dirichlet distribution to the marginal posterior distribution. These are then plugged into the MCMC setup for Admin 2 as the prior for the respective parameters.

Using this procedure to generate the prior for Admin 2 and running the MCMC sample produces the posterior statistics shown in Table 11.7. Comparing these results to Table 11.7, we see the changes in posterior means of the common parameters are small, and within what is expected from the posterior variance. The posterior standard deviations have dropped slightly, for the most part returning to the levels in the calibration using the full data (Table 11.3). However, compare the posterior standard deviations for the three items new to this administration, Tasks 16, 19, and 20. Here, the posterior standard deviations are substantially larger, reflecting the smaller sample size (about 100, compared to 325).

Table 11.7 Summary statistics for binary-skills model, Admin 2

Parameter	Mean	SD	2.5 %	50 %	97.5 %	Parameter	Mean	SD	2.5 %	50 %	97.5 %
λ_1	0.808	0.024	0.759	0.808	0.854						
λ_{20}	0.200	0.076	0.076	0.192	0.367	λ_{21}	0.898	0.028	0.837	0.900	0.947
$\lambda_{\mathrm{MN},0,1}$	0.260	0.083	0.117	0.254	0.436	$\lambda_{\mathrm{MN},0,2}$	0.187	0.074	0.066	0.177	0.351
$\lambda_{\mathrm{MN},1,1}$	0.369	0.088	0.205	0.366	0.551	$\lambda_{\mathrm{MN},1,2}$	0.208	0.074	0.084	0.201	0.370
$\lambda_{\mathrm{MN},2,1}$	0.425	0.080	0.274	0.424	0.585	$\lambda_{\mathrm{MN},2,2}$	0.393	0.075	0.255	0.391	0.544
$\lambda_{\mathrm{MN},3,1}$	0.444	0.047	0.352	0.443	0.537	$\lambda_{\mathrm{MN},3,2}$	0.488	0.046	0.399	0.487	0.577
λ_{50}	0.166	0.067	0.057	0.159	0.315	λ_{51}	0.512	0.096	0.323	0.512	0.697
λ_{52}	0.785	0.056	0.668	0.788	0.883						
$\pi_{4,0}$	0.077	0.018	0.046	0.076	0.116	$\pi_{4,1}$	0.874	0.029	0.811	0.876	0.926
$\pi_{6,0}$	0.199	0.056	0.099	0.196	0.314	$\pi_{6,1}$	0.921	0.017	0.885	0.922	0.951
$\pi_{8,0}$	0.326	0.055	0.223	0.325	0.437	$\pi_{8,1}$	0.780	0.026	0.728	0.781	0.829
$\pi_{9,0}$	0.464	0.051	0.366	0.463	0.562	$\pi_{9,1}$	0.750	0.028	0.694	0.751	0.803
$\pi_{10,0}$	0.043	0.014	0.020	0.041	0.074	$\pi_{10,1}$	0.822	0.035	0.749	0.823	0.886
$\pi_{11,0}$	0.079	0.018	0.047	0.078	0.119	$\pi_{11,1}$	0.867	0.031	0.800	0.868	0.921
$\pi_{12,0}$	0.150	0.044	0.073	0.147	0.246	$\pi_{12,1}$	0.914	0.023	0.863	0.916	0.954
$\pi_{15,0}$	0.177	0.025	0.130	0.176	0.229	$\pi_{15,1}$	0.850	0.039	0.766	0.853	0.917
$\pi_{16,0}$*	0.197	0.063	0.088	0.192	0.328	$\pi_{16,1}$*	0.882	0.036	0.803	0.885	0.943
$\pi_{17,0}$	0.080	0.019	0.048	0.079	0.120	$\pi_{17,1}$	0.805	0.035	0.733	0.806	0.869
$\pi_{19,0}$*	0.079	0.027	0.034	0.077	0.140	$\pi_{19,1}$*	0.781	0.064	0.649	0.785	0.897
$\pi_{20,0}$*	0.073	0.026	0.030	0.070	0.133	$\pi_{20,1}$*	0.845	0.050	0.735	0.849	0.928

* indicates task new to this administration

Note that the second calibration used the posterior distribution from the first calibration for both the evidence models and proficiency model. This is correct under the assumption that both the first administration and second administration are samples from the same population. Although this is correct in this exercise, it need not hold in real life situations. Often the second administration is taken from different points in time, different geographical regions, or different classrooms. There are many reasons that the proficiency model for different populations might be different: differences in instruction; differences in ancillary, but important skills (e.g., language ability on a math test); variation in who chooses to take the test on a given day (many large testing programs show considerable seasonal variation in their test-taking population).

If the populations may be different, using the posterior from the first calibration as the prior for the second calibration's proficiency model may not be the best strategy. The posterior mean may be reasonable, but the variance overstates our certainty about the value of the second population's proficiency parameters. One strategy is to take the posterior distribution and "soften" the prior by increasing the variance. In the case of Dirichlet and beta distributions, this means multiplying the parameters by a value less than one (1/2 or 1/4). In the case of a normal distribution, the mean remains the same, and the variance is multiplied by a number greater than one (2 or 4). Another strategy would be to create a prior for the second calibration by averaging the prior from the first calibration and the posterior from the first calibration. This would be especially valuable if there had been considerable effort to base the original prior on expert opinion. (In concurrent calibration, the way to accomplish this is to build a hierarchical model for the two populations, with the parameters from the proficiency model drawn from the higher level distribution.)

While it is usual to assume that the proficiency model parameters will be different in the two different administrations, it is not usual to assume the same about the evidence model parameters. In particular, the definition of the proficiency has not changed and neither has the task, so why should the conditional probabilities of task response given proficiency change?

Nevertheless, this is an assumption, and hence should be verified. It is formally equivalent to the problem of differential item function (DIF) discussed in Sect. 10.4, here with population subgroups distinguished by administrations. In the example above, there were no unusual differences between the parameters in Tables 11.6 and 11.7. If there were, that would be cause for concern. There are many reasons that such difference arise, for example, a change in the background knowledge of students typically taking the test. These kinds of problem can arise in international assessment, in particular when tasks need to be translated from one language to another (sometimes distinctions which are subtle in one language are obvious in another). If a large difference is observed in practice, then the usual procedure is to treat the tasks on the two different administrations as two different tasks from the same task models and calibrate a different set of evidence model parameters from each.

The linking methodology described here is a variant on the *Non-Equivalent groups Anchor Test (NEAT)* design (Kolen and Brennan 2004). The quality of the linking depends heavily on the number and type of tasks chosen for the anchor test. The standard error of estimation when estimating the proficiency variables from the anchor test will be reflected in the linking error. When there are multiple proficiency variables, there must be enough evidence (either direct evidence through tasks or indirect evidence though correlations among proficiency variables to provide a reasonable estimate of the distribution of the proficiency variables from the anchor test alone.

The evidence-centered design (ECD; Chap. 12) framework helps guide the construction of a reasonable anchor test. In the example above, the anchor test was chosen to contain all of the different evidence model, and where possible multiple tasks from the same evidence model. The length of the anchor test is a bit short, but that is also true of the entire test. As an initial rule of thumb, the anchor test should contain at least ten binary observables (or more if they are clustered within tasks) and at least four or more observables for each proficiency variable. This advice is based on experience with equating unidimensional tests and identifying factors in factor analysis. More research needs to be done on linking in cognitive diagnostic modeling in general.

Sometimes the changes in the conceptual assessment framework (CAF) are so large that linking is difficult to achieve. Examples of this include adding or removing task models (not just individual tasks, but adding a new kind of task or retiring a particular task type); changing the evidence model structures (i.e., the Bayes net fragments, in particular, changing the parents associated with an observable); and adding or removing variables to the proficiency model. The linking procedure described above will help bridge between the old and new designs, but the effective meanings of the proficiency variables and scores may have changed. Score users need to be appropriately cautioned about the difference between the old and new scores.

11.3 Model Checking

The linking procedures described in the previous section all rely on the model being correct. The advantage of using a fully Bayesian scoring model is that it contains a built-in method of model validation: If the observed data have a very low prior probability then it is likely that the model is inappropriate. However, what "very low probability" means in this context is unclear. Furthermore, this test does not provide information about how to fix the model to make it fit better.

Posterior predictive model checking (Sect. 10.2) answers the question about what is an unusual probability for the data. Comparing the observed data to a number of artificial data sets which are known to fit the model provides a reference distribution for any potential model fit statistic. If those statistics are chosen cleverly enough they should lead to insight into what part of the model is problematic and where it could be improved. The graphical methods developed in Sect. 10.3 provide further guidance into looking at the problem.

This section summaries a series of model checking exercises using the Bayes net model described above with the mixed-number subtraction data (Yan et al. 2003; Sinharay et al. 2004; Sinharay and Almond 2007). Section 11.3.1 looks at the observable characteristic plots and identifies a possible problem. Section 11.3.2 explores using posterior predictive model checks to determine both task (item) fit and person fit.

11.3.1 Observable Characteristic Plots

At its heart the Bayesian network model is a multivariate latent class model (Maris 1999). The proficiency model defines 24 possible proficiency profiles. If we knew the proficiency profile of a given student, we should be able to make accurate predictions about their potential response patterns. Due to the design of the assessment, only nine equivalence classes of proficiency profiles are distinguishable (Table 11.2). Still for the purposes of this assessment, knowing which equivalence class a student belongs to is sufficient to make a prediction about their response to any item.

Table 11.2 represents a fundamental assumption about this test form. Although the proficiency model has nine distinguishable latent classes, for the purposes of any given item students can be grouped into two sets. For the set in which $\delta_{i,s(j)} = 1$, we expect most students will be able to solve the problem, with success rate π_{j1}. For the set in which $\delta_{i,s(j)} = 0$, we expect most students will not be able to solve the problem with a guessing rate of π_{j0}.

What could go wrong with this model? The most obvious possibility is that somehow the latent classes are misclassified, and probability of the students who fell into that latent class getting the item correct would be different than the expected value (either π_{j0} or π_{j1} depending on the value of $\delta_{i,s(j)}$). The observable characteristic plot (Sect. 10.3) is designed to check for this possibility. In this plot, a credibility interval for the success rate of all students falling into a latent class is plotted for each latent class. The estimated values of π_{j0} and π_{j1} are plotted for reference. If the observed credibility region does not overlap the expected success rate, then there is an issue that needs attention.

Yan et al. (2003) developed these plots for the mixed number subtraction problem. However, there was one technical problem; the actual proficiency profile of the students, and hence which equivalence classes they fell into, was unknown. Yan et al. solved this problem by simply looking at a single MCMC iteration. Using the assigned proficiency profile assigned to that student in that iteration, it was simple to calculate the observed proportion correct for students in that equivalence class. A Bayesian credibility interval was calculated by using a Bayesian prior of Beta$(.2, .8)$ for equivalence classes in which the student was expected to get the item wrong according to Table 11.1 and a prior of Beta$(.8, .2)$ for equivalence classes in which the student was expected to get the item wrong.

A known limitation of the Yan et al. (2003) procedure was that it did not take into account the uncertainty about the classification of students into equivalence classes. Sinharay et al. (2004) improved the technique by replacing the use of a single iteration with an average over multiple iterations. Where the Yan et al. procedure used the observed proportion correct for the students falling each equivalence class, the Sinharay et al. procedure used a weighted average with the weights taken from the proportion of MCMC iterations that

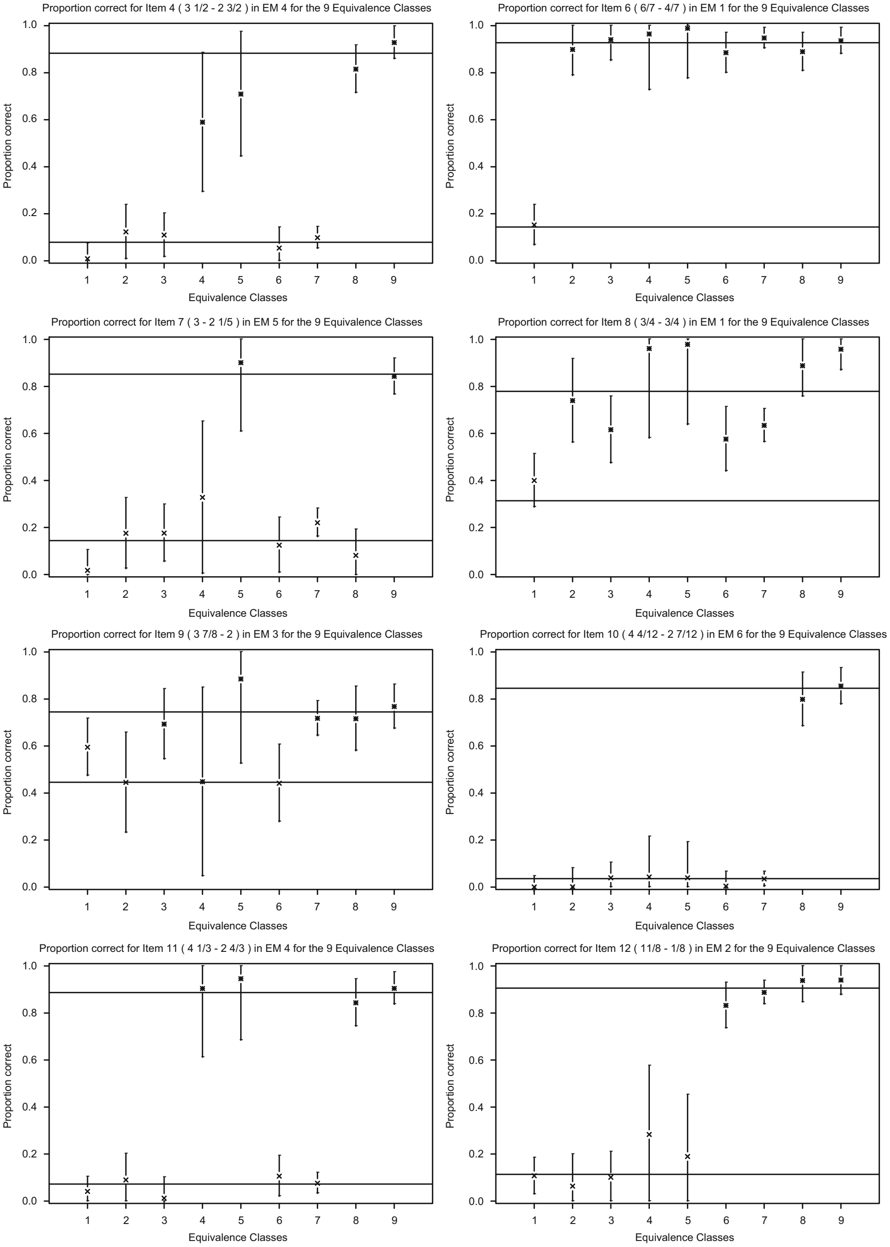


Fig. 11.8 Observable characteristic plots for first eight items. For each equivalence class, the symbol in the center represents the observed success rate for this item and the bar represents a 95 % credibility interval for the success rate. Reprinted from Sinharay et al. (2004) with permission from ETS.

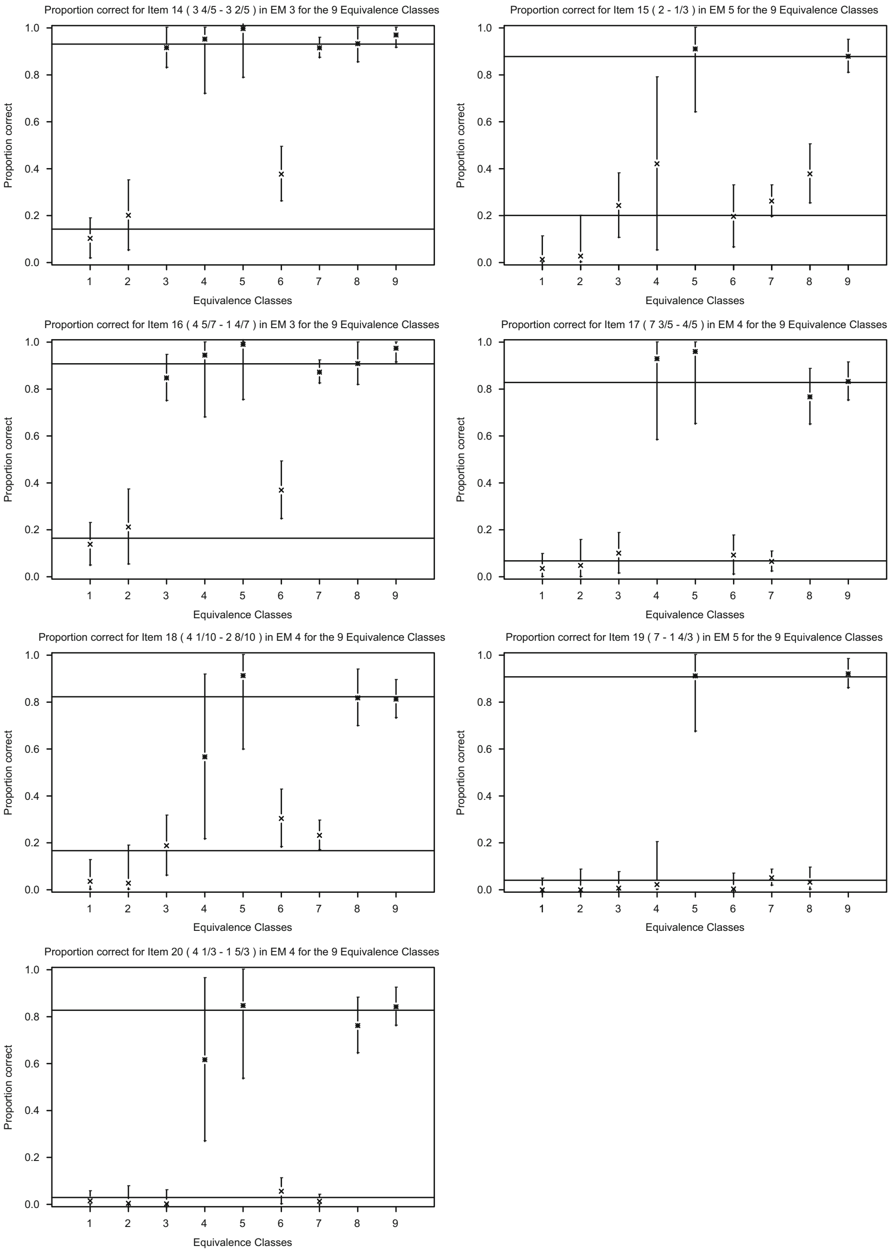


Fig. 11.9 Observable characteristic plots for last seven items. Equivalence classes whose midpoint are plotted with "*" are expected to get the item right, those plotted with "x" are expected to get the item wrong. Reprinted from Sinharay et al. (2004) with permission from ETS.

student fell into that equivalence class. Figures 11.8 and 11.9 show the plots produced by this procedure. The plots produced by the Yan et al. procedure look similar.

Looking at Fig. 11.9, Items 19 and 20 are examples of items that are working well. In each case, the bars for each equivalence class overlap with the upper or lower bar. The patterns are different (representing the different evidence models), but in each case the bars overlap the expected success rate for that evidence model.

Item 18 does not fit as well. Equivalence Classes 1 and 6 do not overlap the lower bar (1 is too low and 6 is too high). Equivalence Class 1 is also problematic in Items 7 and 15, and Item 1 sits very close to the edge. For all of these items, the success rate of students in Equivalence Class 1 is lower than expected. Recall that this is the class for people who have none of the skills. It makes sense that they might be lower than expected. Equivalence Class 6 is also problematic for a number of items; however, a closer examination shows that there are very few people in that equivalence class (5–10 assigned in each MCMC iteration).

These graphs indicate a possible problem with the evidence models taking an all-or-nothing approach to the skill patterns. In particular, they do not distinguish between students with no skills at all and students who lack only one of the required skills. Relaxing the evidence model could produce something that looks more like the observed plots. Define $\delta'_{i,s(j)}$ to be 0 if the student has no skills, 1 if the student has some but not all of the required skills, and 2 if the student has has all of the required skills. In the revised model, there are three probabilities for each evidence model (except for evidence model 1 which requires only the first skill): π'_{j0}, π'_{j1}, and π'_{j2}. Sinharay et al. (2004) and Sinharay and Almond (2007) explore this model among others and find that it does fit better than the original Mislevy (1995b) model.

There is a close correspondence between the items flagged with observable characteristic plot and the observable fit statistic (Eq. 10.10). Sinharay et al. (2004) calculate this statistic for all of the items and note that the two largest values are for Items 8 and 9. Inspecting these items reveals a problem: both admit an alternative solution. Item 8 is $3/4 - 3/4$ which can be solved by noting that anything minus itself is nothing. Item 9 is $3\ 7/8 - 2$, which can be solved by guessing that three and something minus two is one and something. Both items have a very high guessing probability ($\pi_{8,1} = .341$ and $\pi_{9,1} = .475$). As the differences between the high and low values are so small, these items have less evidentiary value than the others. It might be worth checking with the experts to see if these items should be replaced.

11.3.2 Posterior Predictive Checks

If Eq. 11.4 is the correct probability function, then data generated from that same model should be similar to the observed data. The posterior predictive check (Sect. 10.2 formalizes this intuition). A replicate or shadow data set $\mathbf{Y}$ contains draws Y_{ij} for each examinee/item response in the observed data set X_{ij}. The posterior predictive distribution for $\mathbf{Y}$ is $p(\mathbf{Y} \mid \mathbf{X}) = \int p(\mathbf{Y} \mid \boldsymbol{\omega})\, p(\boldsymbol{\omega} \mid \mathbf{X})\, \partial\boldsymbol{\omega}$, where $\boldsymbol{\omega}$ represents all the parameters in the model—in this problem, θ's, π's, and λ's.

The MCMC sampler we have already built produces samples from the posterior distribution for $\boldsymbol{\omega}$. We produced a shadow data set $\mathbf{Y}^{(t)}$ in each cycle, by augmenting the sampler to to draw $Y_{ij}^{(t)}$ from a facsimile of Eq. 11.2:

$$Y_{ij}^{(t)} \left| (\delta_{i,s(j)} = z\,) \sim \text{Bern}\left(\pi_{j\delta_{i,s(j)}}^{(t)}\right), \qquad \text{for } z = 0, 1. \qquad (11.6)\right.$$

Then, for a chosen statistic $D(\cdot,\cdot)$, compute $D(\mathbf{Y}^{(t)}, \boldsymbol{\omega}^{(t)})$ and $D(\mathbf{X}, \boldsymbol{\omega}^{(t)})$ for each cycle of the MCMC sampler. The proportion of cycles in which $D(\mathbf{Y}^{(t)}, \boldsymbol{\omega}^{(t)}) > D(\mathbf{X}, \boldsymbol{\omega}^{(t)})$ gives the posterior predictive p-value for that statistic. When a statistic is constructed so that a higher value means worse fit, a low p-value means the fit statistic for the observed response was usually higher than the shadow response, so the observation was surprising in light of the model.

Yan et al. (2003) chose two statistics based on the Pearson residual (Eq. 10.7) to compare the observed value to its prediction (in this case the probability of success under the model).[6] For each MCMC cycle, let $p_{ij}(\boldsymbol{\omega}^{(t)}) = E\left[x_{ij} \mid \boldsymbol{\omega}^{(t)}\right]$, that is the predicted probability of success (Eq. 11.2) given the sampled parameters and proficiency variables for Iteration t. Then, define person*item squared Pearson residuals as

$$V(u_{ij}, \boldsymbol{\omega}) = \frac{(u_{ij} - p_{ij}(\boldsymbol{\omega}))^2}{p_{ij}(\boldsymbol{\omega})\,(1 - p_{ij}(\boldsymbol{\omega}))}.$$

This can be calculated for both the original data, $V(X_{ij}, \boldsymbol{\omega}^{(t)})$, and the replicated data, $V(Y_{ij}^{(t)}, \boldsymbol{\omega}^{(t)})$. As this statistic depends on the estimated parameters, the value for the original data will be different on every MCMC cycle. Averaging the squared residual across all observables for an individual produces a measure of person fit:

$$PF_i(\mathbf{U}, \boldsymbol{\omega}) = \left(\frac{1}{J}\sum_{j=1}^{J} V(u_{ij}, \boldsymbol{\omega})\right)^{1/2}. \qquad (11.7)$$

[6] Fit indices based on Pearson item-by-person residuals are simple and fairly widely used in IRT, but their statistical properties leave much to be desired; see Meijer and Sijtsma (2001) for a review of person-fit statistics and Glas and Falcón (2003) on item fit, and Haberman (2009) for more recent developments derived from contingency table analysis. Using PPMC somewhat mitigates one serious problem, namely, the lack of theoretical reference distributions, as it effectively creates tailor-made reference distributions for fit statistics. Different fit indices provide better or worse approximations of empirical distributions under the PPMC null model, so research on optimal choices is in order (Levy 2011).

Averaging the squared residual across all individuals responding to an observable produces a measure of observable fit (or item fit):

$$OF_j(\mathbf{U}, \boldsymbol{\omega}) = \left(\frac{1}{N} \sum_{i=1}^{N} V\left(u_{ij}, \boldsymbol{\omega}\right) \right)^{1/2} . \tag{11.8}$$

Averaging the squared residual across all individuals and observables produces a measure of overall fit (or total fit):

$$TF(\mathbf{U}, \boldsymbol{\omega}) = \left(\frac{1}{NJ} \sum_{i=1}^{N} \sum_{j=1}^{J} V\left(u_{ij}, \boldsymbol{\omega}\right) \right)^{1/2} . \tag{11.9}$$

Taking the square roots of the averages puts all three measures on a root mean squared error metric. As each observable is either 0 and 1, and $0 < p_{ij} < 1$, all three measures always range between 0 and 1, with lower values indicating better fit. The posterior mean of all of measure indicates the typical magnitude of discrepancy between observed and predicted observables.

Does a given degree of discrepancy represent good or poor model fit? The posterior predictive distribution for each statistic provides an approximate null distribution against which observed values can be compared. In each MCMC cycle, we compare the fit measure (here high values mean worse fit) of the actual data and the shadow data: Is $PF_i(\mathbf{Y}^{(t),\boldsymbol{\omega}^{(t)})} > PF_i(\mathbf{X}, \boldsymbol{\omega}^{(t)})$ for Person i? Is $OF_j(\mathbf{Y}^{(t)}, \boldsymbol{\omega}^{(t)}) > OF_j(\mathbf{X}, \boldsymbol{\omega}^{(t)})$ for Observable j? For overall model fit, is $TF(\mathbf{Y}^{(t)}, \boldsymbol{\omega}^{(t)}) > TF(\mathbf{X}, \boldsymbol{\omega}^{(t)})$? This can be easily built into the MCMC sampler. We monitor the proportion of cycles in which the inequality holds. If the model fits the data, this should be around .5. If this value is close to 0 (say below .10), that indicates that the original data fits substantially worse than the simulated data, and indication of trouble. Since the shadow data are generated from parameters estimated from the observed data, these probabilities tend to be conservative; that is, they show the observed data fitting better than they would under a true null distribution. They are therefore less useful for absolute indicators of fit than for comparisons of fit among like parameters (e.g., for detecting which items fit relatively worse than others).

Observable-Fit Indices

Most of the observables fit very well with respect to the mean square error indices, although some tasks do fit better than others. The fit indices are pseudo fit probabilities, or frequencies with which mean squares for observed data were higher than for shadow data, or $\mathrm{P}(OF_j(\mathit{Shad}) \geq OF_j(\mathit{Obs}))$. Table 11.8 provides the fit indices. Most of the observables fit well with indices are around 0.5. Items 8 and 9 do not fit as well as all the other items; their item-fit indices are 0.261 and 0.284. These are the same items that the previous analysis flagged as problematic. (See Figs. 11.8 and 11.9).

Table 11.8 Item-fit indices for the mixed-number subtraction test

Item j	4	6	7	8	9	10	11	12	14	15	16	17	18	19	20
$\mathrm{P}(OF_j(\mathit{Shad}) \geq OF_j(\mathit{Obs}))$	0.538	0.569	0.430	0.261	0.284	0.903	0.563	0.529	0.547	0.392	0.472	0.539	0.368	0.712	0.777
$OF_j(\mathit{Obs})$	0.995	0.987	1.020	1.032	1.026	0.925	0.987	0.994	0.991	1.023	1.009	0.996	1.028	0.919	0.874

[a] Sum of squared standardized residuals

Person-Fit Indices

Table 11.9 shows the person fit indexes for selected students. Overall the students fit well with respect to the mean square error indices for all the items. The average value for the pseudo probabilities over the full sample is between .5 and .6. This is not surprising as 15 items is a relatively short test to develop evidence of misfit. Still, small values of this statistic should indicate more unusual response patterns. As examples, Students 26, 94, and 156 all fit very well according to our model with fit indices values around 0.5. Students 36, 47, and 315 did not fit well; their fit indices are all less than 0.1.

Table 11.9 Person-fit p-values for selected students

Student i	26	32	35	36	47	94	127	156	315
$\mathrm{P}(PF_i(\mathit{Shad}) \geq PF_i(\mathit{Obs}))$	0.499	0.266	0.740	0.035	0.057	0.427	0.270	0.493	0.002

Refer back to Table 11.4 for the response patterns for these students. Consider Student 26, who has 12 out of 15 items correct. This student was able to solve all but one of the tasks using evidence models 1, 2, 3, and 4, but missed two out of three tasks using evidence model 5; this looks very much like a person who has Skills 1 through 4, but not Skill 5. Student 156 shows a similar pattern. Student 94 has seven correct observables, from evidence models 1, 2, and 3, and another from evidence model 4; this also corresponds to meeting a requirement of Skills 1 through 3 for these items. Student 127 has only three items correct that are all from evidence model 1, plus Item 11 from evidence model 4; this student also fits the model well, most likely having only Skill 1.

Student 36 does not fit well. The nine items that student got correct are scattered across different evidence models. This examinee missed an item from evidence model 1, requiring only the basic fraction skill (Skill 1), but was able to answer several questions from evidence models 4 and 6 correctly, requiring the more advanced Skill 4. This relatively contradictory pattern of evidence leads to the evidence of misfit.

Student 47 is another case of contradictory evidence. The items this student missed from evidence model 4 provide evidence for a lack of Skill 4, while the tasks this student was able to solve from evidence model 5 indicate that Skill 5 is present. Finally, Student 315 got many relatively difficult items from evidence model 4 correct while missing the basic fraction subtraction items in evidence model 1.

Patterns of misfit may provide interesting diagnostic information about a student. These students may be developing their mixed number skills in a pattern that is different from the usual developmental sequence (although the short length of this assessment makes any such conclusions rather tenuous; again, we would use a longer test in practice, say with five items from each task model). The teacher should look more closely at these individuals and see if there is any different kind of explanation or instruction that is needed.

11.4 Closing Comments

The goal of this chapter is to put the material of Part II in perspective through an example. The first step in building an assessment is to translate the information about the domain gathered from the experts into a mathematical model (Chap. 8; Sect. 11.1). For this example, we were able to build on the work of Klein et al. (1981) and Tatsuoka (1984) in constructing the Bayesian network. Given the Bayesian model, and the data collected by Tatsuoka, the MCMC algorithm (Chap. 9) could be used to calibrate the model to the data, and to link two different forms of the assessment (Mislevy, Almond, et al. 1999a; Sect. 11.2). Finally, because the model was a fully Bayesian model, the methods of Chap. 10 could be used to critique and suggest improvements (Yan et al. 2003; Sinharay et al. 2004; Sinharay and Almond 2007, Sect. 11.3).

Of course, none of this would have been possible without the foundational work of Klein et al. (1981) and Tatsuoka (1984) in building a cognitive model for the domain of mixed number subtraction. The contributions of the papers reviewed in this chapter have been to translate that cognitive model into a mathematical model using the notation of Bayesian networks. This is primarily a problem of knowledge management, and evidence-centered assessment design was developed as a tool to manage the kinds of knowledge that go into developing an assessment. Part III describes the role of ECD in building Bayesian network models for assessment systems.

Exercises

11.1 (Skill 4 as a Prerequisite of Skill 5). How could the graphical structure of Fig. 11.1 be changed to include the relationship that *Skill 4* is a prerequisite *Skill 5*? What additional restrictions would be necessary on the conditional probability tables?

11.2 (Sensitivity to Prior Means). Mislevy et al. (1999a) used a $\text{Beta}(6, 21)$ prior for λ_{20} and $\text{Beta}(21, 6)$ prior for λ_{21}. The posterior mean (and SD) reported in that paper is .22 (.07) for λ_{20} and .90 (.03) for λ_{21}. Compare these to the numbers in the second row of Table 11.3. How sensitive are these results to the choice of prior? Explain the difference in sensitivity.

11.3 (Starting Values). Instead of using the prior distribution to derive starting points, Mislevy et al. (1999a) first did a test run, and then chose starting points using the Midpoint, High, Low, HighLow, and LowHigh strategies using the sample statistics from the test run rather than the prior. How does using the prerun instead of the prior for starting values affect the convergence tests if the posterior is mostly smooth and unimodal? If the posterior has lots of local maxima?

11.4 (Are More Chains Better?). Is it better to run more chains or have fewer chains and run them longer?

11.5 (Missing at Random). Does the non-equivalent groups anchor test (NEAT) design meet the qualifications for missing at random? Missing completely at random?

11.6 (Intelligent Tutoring System). Suppose that the Bayesian network model is embedded in an intelligent tutoring system that controls which tasks (both instructional and assessment) are presented to the students based on the estimated value of the proficiency variables at the time the task is chosen. A researcher proposes to calibrate the assessment model for this system using the MCMC method with a large number of student records taken from using the tutoring system in a number of classrooms. In these records, any given student has recorded answers for only some of the tasks (those selected by the tutoring system). The research plans to rely on the ability of the MCMC method to handle missing data when calibrating the model. Will this work?

11.7 (Effective Sample Sizes). Use the method of moments approximation to calculate the effective sample sizes for the posterior margins of the evidence model parameters, p_{j0}'s and p_{j1}'s, from the second administration, Table 11.7. What is different about the effective samples sizes for the parameters marked with an asterisk in that table? Why are they different?

11.8 (Weight of Evidence). Using the numbers from Table 11.3, calculate the expected weight of evidence for Skill 1 provided by Item 6 and by Item 8. Explain the difference.

11.9. Items 6 and 8 are simple structure tasks in that they tap only a single proficiency variable. All of the other tasks in the mixed number subtraction test are not simple structure. Does the expected weight of evidence for a single skill depend on the previously seen observables for a simple structure task? For a nonsimple structure task? Explain.

11.10 (Many Person Fit Tests). A psychometrician analyzes a data set with 2000 students and finds that the person fit statistic flags 100 of them (using .95 as the threshold). Does this indicate a problem with the model?

11.11 (Length of Bars in Observable Characteristic Plots). Look at the observable characteristic plots, Figs. 11.8 and 11.9. The error bars for Equivalence Classes 4 and 5 are typically longer than for the other equivalence classes. What is the most likely reason?

11.12 (Nonmonotonic Patterns in Observable Characteristic Plots). Look at the observable characteristic plots in Fig. 11.8. Some of them, such as those for Tasks 6, 10 and 12 clearly follow a monotonic pattern: the chance of success is better for those students in higher numbered equivalence classes. Other plots, such as those for Tasks 4, 7, and 11, show a nonmonotonic pattern where the success probability is higher for Equivalence Class 5 than for Equivalence Classes 6 or 7. Is this cause for concern?

11.13 (Model Fit). Agree or disagree: Assessment designers should always chose the model that has the best value of the model fit statistic. Justify your choice.

Part III

Evidence-Centered Assessment Design

12

The Conceptual Assessment Framework

The first two parts of this book have dealt mainly with the mathematics of using Bayesian networks: first for scoring assessments, and then for calibrating scoring models to observed data. Aside from brief digressions in Chaps. 2 and 7, it has largely neglected any formal discussion of assessment design. This follows a common trend because assessment design, as a formal discipline, may be the most neglected area of study within professional measurement. The principles of design typically get short shrift in treatments of psychometrics and educational measurement in favor of statistical and technical topics.

This may be due, in part, to the fact that in most measurement organizations and institutions responsible for testing programs, the responsibilities of test construction and scoring are divided among professionals with very different and specific skill sets and interests. It is uncommon for content experts who write items to interact with the statistical analysts beyond the discussion of items that are inconsistent with the scoring model employed for the test or how they perform in pretesting. Common practice is for test development, administration, and scoring to be "silos" of responsibility that seldom interact except at key hand-off points in the work. Therefore, discussions are targeted toward a particular professional's responsibilities and making sure the "hand-off" to another team goes smoothly.

Getting hand-offs right requires a common understanding of what is expected from each team. As an analogy, the common understanding when constructing a building is provided by the blueprints. Looking at the common blueprints, carpenters, stonemasons, plumbers, electricians, painters, landscapers, project managers, and inspectors can all see what is expected and plan for their tasks, and leave appropriate spaces for other craftsmen to work. In a similar way the *conceptual assessment framework* (CAF) is designed to serve as the blueprint of an assessment. It should provide information to the teams that work on the assessment about what their roles are and what the hand-offs are.

There are situations in which a formal design is not necessary. A single person building a tool shed does not need a full set of blueprints; a rough

R. G. Almond et al., *Bayesian Networks in Educational Assessment*,
Statistics for Social and Behavioral Sciences, DOI 10.1007/978-1-4939-2125-6_12

sketch on the back on an envelope can be sufficient. Likewise, an experienced team of barn raisers do not need formal blueprints; each new barn is similar enough to the last that each of the members of the team can follow their practiced roles.

A formal design is similarly not necessary for a teacher designing a classroom quiz or a professional testing organization producing additional forms for a well-established testing program. When an assessment is meant simply to gauge students' overall proficiency in some domain of tasks, and the tasks are independent performances that can be scored on low-to-high scale (e.g., right or wrong, ratings on a set of essays), the information for the hand-off between test developers and psychometricians is correspondingly simple: "Here are the item data, and for each item, a higher score is better than a lower score." Item writers might employ deep content knowledge and subtle insights into each item and psychometricians might fit wonderfully complex models, but as long as this assessment paradigm holds, the channel of information can be narrow and the professionals at either end do not need to know the details at the other end.

Notice that these situations share several elements that make it possible to get away without an explicit design (at least seemingly). It is presumed that the claims underlying the assessment purpose are sufficiently well understood; that the kinds of observations needed to back them are produced in the tasks; the scoring procedures capture the salient features of tasks performances; the task scores are combined in a way that conveys their evidentiary value; and the results are accurate enough to serve the purpose. This may be the case in some instances, but there are clearly several places where the argument could break down. Even in informal or familiar assessments, it is worth thinking through just what one wants to make inferences about, and how suitable evidence might be obtained. A design framework could improve assessment practices that have grown familiar and comfortable.

More complex assessments are the context for discussing evidence-centered assessment design (ECD) in this book, but ECD is based on a general form of thinking, hypothetico-deductive reasoning (i.e., the scientific method). It is relevant as well to less formal assessment situations, and more broadly to situations almost anywhere that information is being produced and evaluated, such as simple classroom Socratic practice or evaluating homework responses. We activate this kind of thinking, almost automatically, rapidly, iteratively, many times a day. Employing it more consciously and frequently in even routine, wholly familiar, or spontaneous assessment situations, both casual and formal, would be beneficial when a formal design process is not practical.

The need for a formal design becomes stronger when the assessment contains elements that are new, unfamiliar, or complex. There may be a desire to link assessment more tightly to theories of learning in the domain (Pelligrino et al. 2001). Psychometricians may want to provide test developers or teachers with deeper insight into what scores and statistics mean in terms of student learning (Wilson 2004). An organization may want to develop reusable elements for constructing and delivering tasks (Luecht 2012). There may be multiple aspects of proficiency, and different tasks may require different mixes of them, as with the Bayes nets psychometric approach discussed up to now.

One of the primary advantages of Bayesian networks as scoring models for assessment is their flexibility; that flexibility can also present challenges. ECD helps us exploit the flexibility of Bayesian networks. The intent of ECD is that the ECD design process forms the knowledge engineering required to build the appropriate Bayesian network model, while at the same time an understanding of the models and methods required in scoring adds both supporting structure and constraints to task development.

Part III takes up the neglected topic of assessment design. This chapter focuses on the CAF—the ECD blueprint—which is the result of a design process beginning with an examination of the claims an assessment needs to ground and the evidence it needs to back them. Chapter 13 presents an idealized delivery model for an assessment. Chapters 14 and 15 then illustrate the ideas of integrated assessment design and analysis with a prototype biology assessment called Biomass that used complex tasks, a multivariate model of proficiency, and web delivery. Chapter 14 describes the assessment design, and Chap. 15 works through scoring and the calibration of the scoring model with field trial data. Finally, Chap. 16 talks about some of the possibilities for assessments that ECD and Bayesian networks open up.

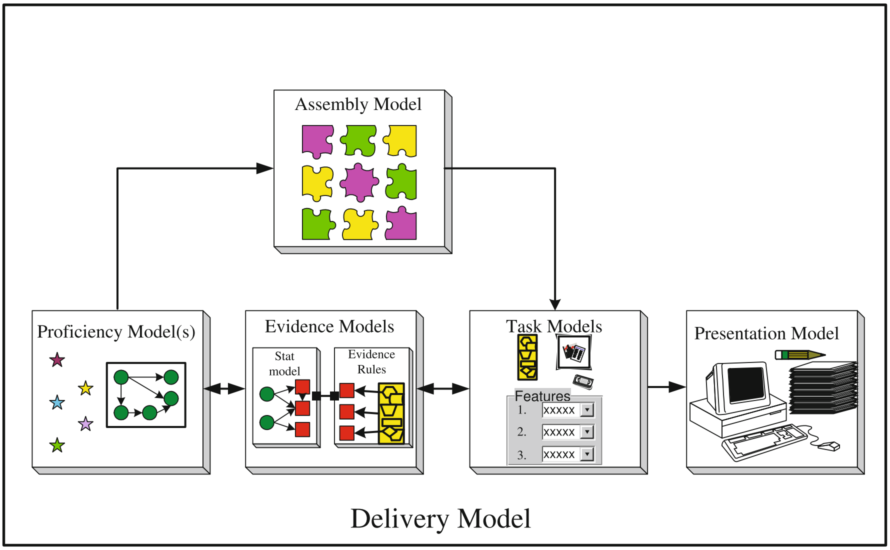


Fig. 12.1 The principal design objects of the CAF
Reprinted from Mislevy et al. (2004) with permission from the Taylor & Francis Group.

Figure 12.1 (repeated here from its original appearance in Chap. 2) summarizes the parts of the CAF. This chapter walks through the CAF framework with simple examples from a hypothetical English language placement assessment. The Biomass example in Chaps. 14 and 15 is real and richer. The interested reader will find further applications in the literature, including the design rationale of the Dental Interactive Simulation Corporation (DISC) scoring engine (Mislevy et al. 2002d), redesign in the College Board Advanced Placement examinations (Huff et al. 2010), unobtrusive assessment in learning games (Shute and Torres 2012), and the Cisco Networking Academy's NetPass simulation-based assessment prototype (Williamson et al. 2004b; Williamson et al. 2006a) and its descendant Packet Tracer simulation-based assessments (Behrens et al. 2012).

Section 12.1 starts the chapter by discussing the design process and the critical ECD concept of *claims*. The next six sections discuss the six models of the CAF: the proficiency models (Sect. 12.2), the task models (Sect. 12.3), the evidence models (Sect. 12.4), the assembly model (Sect. 12.5), the presentation models (Sect. 12.6), and the delivery model (Sect. 12.7). Section 12.8 discusses how these six models work together to form the complete assessment.

12.1 Phases of the Design Process and Evidentiary Arguments

Chapter 2 already introduced the six models that comprise the CAF. However, building those models starting from a description of purpose and a diverse collection of knowledge about a domain of interest can be a complex process (Mislevy, Steinberg, and Almond 2003b). It is a process of building a coherent evidentiary argument, providing the right kinds of evidence about the right kinds of proficiencies to serve the purpose of the assessment, then embodying it in specifications for all of the materials, activities and processes that constitute an operational assessment. Section 12.1.1 gives a brief description of domain analysis and domain modeling processes. Section 12.1.2 describes what is perhaps the most important output of the initial domain modeling: the set of claims that ground the definition of the CAF models, and serve as the first sketch of the evidentiary argument of the assessment.

12.1.1 Domain Analysis and Domain Modeling

Almond et al. (2002a) provides an idealized picture of assessment delivery (Chap. 13 discusses how this process looks when the assessment uses Bayesian networks as the psychometric model). The CAF serves as a blueprint for the pieces of that assessment delivery system and requirements for the processes that make it work. However, the CAF does not spring fully formed from the heads of the assessment designers. As with any design process, the designers must first gather information about the requirements and the constraints of the domain in which they are operating. Then they need to examine a number of trade-offs exploring possible alternatives and weighing competing objectives. Once the overall shape of the assessment is clear, finer-grained design decisions go into completing the CAF.

ECD divides the design process into three stages:

Domain Analysis. The process of gathering and organizing the requirements and information about the domain of the assessment.

Domain Modeling. The development of the central evidentiary argument of the assessment, and sketching the basic models of the CAF.

Conceptual Assessment Framework. Fleshing out the argument from the domain model to make a complete description of the assessment.

As with any design process, it is possible and even desirable to revisit and refine work done at earlier stages as necessary later in the process. Except in well-understood domains and familiar kinds of assessment, iterative design is the norm, rather than a "waterfall" design process. Task prototypes and pilot testing are typical, and can spark not only revisions in task design and scoring procedures, but revisions to the argument and needs to gather additional information about the domain. Nevertheless, the ECD process encourages the designers to think through the issues involved with fitting the pieces of the assessment together at an early conceptual phase, thus avoiding committing resources to something that will require extensive changes later to make it work.

The goal of the domain analysis phase is organizing the information about the domain, which will inform the design of assessment elements. Further, it will provide warrants and backing for the evidentiary arguments for claims of interest in the intended uses of an assessment. In some cases, the challenges will be sorting through a vast literature for a well-studied domain and the need to reconcile conflicting viewpoints. In other cases, the challenge will be finding relevant prior work on the constructs to be measured. In these cases, the design team may need conduct cognitive analyses: observing the behavior of experts and novices performing relevant tasks in natural or laboratory conditions.

In either case, the challenge lies in integrating insights from psychological theories about learning and cognition with a substantive model of performance in the domain of interest (Mislevy 2006; Mislevy 2010). The final model used for the assessment will never capture all of the nuance in the psychological theories. The design team will need to make simplifications to fit the assessment within time and cost constraints. This simplification happens as the design team moves from domain analysis to domain modeling to the final CAF. Some aspects of proficiency can be assumed in the population, for example; others are not relevant to the purpose; target inferences may be needed only at a coarser grain size than research studies have addressed. Good documentation during the domain analysis phase allows those simplifying assumptions to be revisited if they cause problems when the assessment is field tested.

There is often a great deal of information already available that will be useful in designing an assessment, but it has been gathered for purposes other than assessment design. It may be pertinent, but it it is not clear just how. One can begin by organize information along lines that relate to elements of assessment arguments. Mislevy, Steinberg, and Almond (2003b) suggest the following categories:

Valued work. Real world situations in which people do the kinds of things and use the kinds of knowledge we care about.

Task features. Aspects of situations that vary to impact their difficulty or evidentiary focus, or just make similar tasks seem different. Cognitive task analyses are particularly helpful here.

Representational forms. Ways that knowledge is expressed and represented in the domain (e.g., graphs, diagrams, symbols, tables, and vocabulary that must be understood), and the ways people use these representations and situations they use them in.

Performance outcomes. Ways of distinguishing among performances and their outcomes, such as aspects of quality, efficiency, and strategy use, and criteria for recognizing "successful" performances.

Valued knowledge. Knowledge, skills, and abilities that are considered important in the domain.

Knowledge structure and relationships. Information about how knowledge is structured, such as prerequisite relationships, curricula, and knowledge maps.

Knowledge-task relations. Information about which knowledge, skills, and abilities are required for which tasks. Cognitive task analyses are helpful here too.

Some pieces of information fall into more than one category. Lack of information in a category can indicate the need for additional research, perhaps through additional literature searches, discussions with experts, or experiments that fill in gaps.

Together, this domain information and purpose of the assessment will help the designer define the claims the assessment should address and ways to get evidence to support them. The work of domain modeling can begin when sufficient information has been gathered (although one expects to cycle back to get further information when design work shows it is needed). The goal of the domain model is to sketch the evidentiary argument of the assessment in enough detail to identify the critical points of the assessment design and make sure that enough resources are available to complete the design.

A *domain model* consists of a collection of objects called *paradigms*. The paradigms of the domain model are lightweight versions of the more detailed models of the CAF. The reason for building the domain model before the CAF is to avoid the potential rework that might be required to develop the more technical and detailed designs, until the design as a whole is sufficiently worked through that all of the stakeholders involved in the design process are satisfied to move ahead. Domain modeling is meant to be a stage where experts from different areas who might be involved in assessment design—subject matter experts, psychometricians, software designers, teachers, cognitive scientists, graphic designers—can talk with each other about what is needed in the assessment argument and how it might be instantiated, discussing options and trade-offs that cut across their specialties.

Although a paradigm is less detailed than a CAF model, the level of detail required will vary with the application and the similarity of the assessment to other assessments that the design team are familiar with. Suppose initial design discussions suggest that for the targeted claims and necessary evidence, it will suffice to use a familiar form of task, such as an item type used in similar assessments. The task paradigm then only needs to reference the previous work, and indicate how task features and performances will provide the necessary evidence. On the other hand, when it appears that a new kind of simulation task will be needed to obtain evidence for the targeted claims, the task paradigm might need to be quite detailed before the content specialists, the computer implementation team, and the psychometricians are convinced of its feasibility and evoked evidence to try a prototype.

Example 12.1 (Language Placement Test). *University C holds a special 4-week language course before the fall semester for foreign students. The purpose of the course is to ensure that students have sufficient English language competence to take part in academic life. Several sections of the course are offered which differ in their emphasis (for example, reading and writing versus speaking and listening), and many students have sufficient English language skills that they do not need the course all. Therefore, University C offers a placement exam to all incoming foreign student to determine how best to place them into the appropriate section of the English class.* [1]

Note that the two most fundamental parts of the assessment design have already been specified in this brief introduction to the Language Placement Test example: (1) the purpose of the assessment and (2) the targeted population. Both of these are critical first steps in any design process.

The *purpose* is the declared intended use(s) of the assessment, including its primary and secondary uses (e.g., course placement and performance feedback to the student). This defined purpose will drive all subsequent stages of the assessment development to ensure that the assessment is appropriate for its intended use. As the initial step, all ECD claims (Sect. 12.1.2) will be specified in terms of how they support the stated purpose of the assessment.

[1] Some of the issues discussed in this example are similar to those arising in the redesign of Educational Testing Service's (ETS's) Test of English as a Foreign Language (TOEFL$^{\mathrm{TM}}$) (Chapelle et al. 2008). Indeed, this example draws on conversations with members of the TOEFL redesign team. However, the purposes of this hypothetical example and TOEFL are not the same, so the eventual design decisions are not, nor should they expected to be, the same.

In this case, the purpose of the assessment is clearly stated as being for placement into the appropriate level of language course offered by University C. Implicit in this stated purpose is the idea that the content of the assessment must correspond to the intended purpose of the course, that is promoting the English language abilities necessary to participate in academic life. However, even that requires further refinement. Does "academic life" include holding a discussion with the registrar to straighten out a conflict of the student's class assignments? Requesting information from a cafeteria worker about whether or not an entrée meets the student's dietary restrictions? The design team will want to examine the range of activities using English that students will encounter in the university—Bachman and Palmer (1996) call these "target language uses"—and determine the range and the features of those situations as a guide for task models. For this example, we will use a restricted specification that only addresses the classroom context.

The targeted assessment population has also been specified. The *population* is a complete definition of all individuals (or groups) who are eligible to sit for the assessment and to whom the purpose of the assessment will be applied. In this example, the population is defined to be all newly matriculated students who come from a country that does not have English as its primary language. Note that even with this explicit definition of the population there are important subaspects of the population definition that may need to be made explicit, such as by specifying the expectations about prior qualifications or conditions of the population. Some knowledge or skills may be critical to performance, yet need not be included in the psychometric model because it will be known aforehand that all examinees are sufficiently proficient in these respects. For example, it may be worthwhile to specify that the definition of population as incoming freshmen presumes that all such freshmen have been subjected to an admissions process that ensures that they all have some basic proficiency in English.

It is important to make the description of the population and purpose explicit: these critical requirements will drive many of the subsequent design decisions. It is better for an assessment to serve a single purpose well than to do a poor job of supporting many purposes. The domain modeling process helps clarify purposes and highlight trade-offs that are involved. Moreover, a domain model helps a design team think through what needs to be changed about an assessment to meet a new purpose (Fulcher and Davidson 2009).

The population and purpose persist from the domain model to the more detailed CAF. The another aspect of the design that is important to specify in the domain modeling phase is the basic evidentiary argument of the assessment.

12.1.2 Arguments and Claims

The proficiency variables and observable variables are pieces of machinery to aid reasoning in assessment. The Bayes nets help us express the relationships between what students know and what they do, combine evidence across multiple observations, and characterize what we know from evidence and what we do not know. The pieces of machinery—the variables, the conditional probabilities, the independence relationships—take their *meanings* from an underlying *evidentiary argument* (Mislevy 2006).

Toulmin (1958) provides a schema for the structure of arguments (Fig. 12.2). In assessment, its content is developed from the information gathered in domain analysis activities. The focus is a *claim*, or targeted inference, which in assessment is a statement about the participant that we wish to establish. For example, in the Language Placement Exam, a claim might be that a student can write plans for future study. The *data* are observations we can make about the participant that would cause us to believe that the claim does or does not hold. A piece of data might be the rated quality of a written response to the question, "What are your academic goals and how do you plan to achieve them?" on the student's application.

The most obvious data in educational assessments are these aspects of what examinees say or do or make. There are two additional kinds of data in assessment arguments as well, however. First is features of tasks, or the situations that examinees act in that make their performances meaningful as evidence about their capabilities. That is, what is it about this situation that examinees' actions here provide evidence about their knowledge and skills? The features of Tatsuoka's mixed-number subtraction tasks, for example, indicate which skills would be required for a correct response. The Q-matrix and the response together are needed to ascertain the *evidence* that the response *data* from examinees' performances convey.

Second is additional information about the examinees the assessor has, which can also be critical to interpreting response data. Some information of this kind is known generally about the examinee population. It is assumed that the mixed-number subtraction examinees are familiar with fraction representations, so even though this skill is critical to performance, it does not need to be in the psychometric model—it implicitly has a value of "mastered at the required level." In some assessments, what is known specifically about individuals may be known, such as which method of mixed number subtraction they studied, so that the appropriate Q-matrix can be used to interpret the evidence in their responses.

The arrow that goes from specific data up to a specific claim—an inference about a particular examinee based on her responses—is justified by a *warrant*. A warrant is a generalization that underlies the assessment's construction: why a task with such-and-such features is likely to evoke observably different performances from examinees with different proficiencies. In mixed-number subtraction, the warrant is that students with the requisite skills are likely to make correct responses, and those lacking in skills are likely to make incorrect responses. This warrant justifies specific conclusions for each student's particular pattern of responses. Similarly, the warrant in the running application example would be that usually students who are able to write a good essay

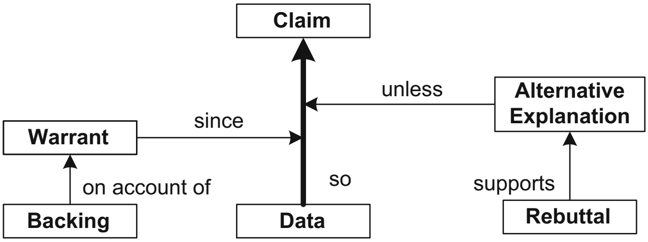


Fig. 12.2 Toulmin's structure for arguments
Reprinted from Mislevy et al, (2003a) with permission from The National Center for Research on Evaluation, Standards, & Student Testing (CRESST), UCLA.

on their application are usually also able to write other material related to plans and goals, for example, papers describing a class project.

A warrant requires *backing*. In mixed-number subtraction, the backing is teacher experience, the instructional design, and the cognitive analysis of Tatsuoka and her colleagues (Klein et al. 1981). In the application example, the backing is the experience at University C with the relationship between student admission essays and later student work. We might also have results from prior studies or theoretical work on writing we can use as part of the backing. The material gathered during the domain analysis is an important source of backing.

In any particular case the relationship between the data and claim may not hold. There can be *alternative explanations*. In the case of the admission essay, somebody else could have written the essay for the student. Some information will support an alternative explanation in a particular case, while other information will weaken it. Data both for and against an alternative are included in the box labeled *Rebuttal data*.[2] For example, we could observe that the student's writing performance is very different in proctored and unproctored writing samples, leading us to suspect that the student is receiving some kind of assistance. Many assessment design decisions are meant to reduce the force of alternative explanations: making sure tasks actually evoke the proficiencies we are interested in, or reducing the demand for extraneous knowledge. From a measurement perspective, this is reducing the validity threats of construct underrepresentation and construct irrelevant variance (Messick 1989).

The next step in the language placement test example is to elaborate a set of claims appropriate to the purpose. At the highest level the claims are generally too broad to be useful. Often the claims are broken down in a hierarchical fashion to get to something specific enough to guide task design. As we see in the following example, there need not be a one-to-one relationship between claims and proficiency variables.

[2] See Schum (1994) for in-depth discussions of the elements of arguments and their broader relation to probabilistic inference.

Example 12.2 (Language Placement Test Claims). *Given the purpose of the test, the primary claim should be about the student being ready to operate in the classroom without need for further instruction. It is immediately apparent that this breaks down further into the ability to read, write, speak, and listen sufficiently well to participate in the classroom; that is, to engage in the kinds of interactions involving language, using the forms and genres, for the kinds of purposes, around which teaching and learning occur in the classroom.*

However, reading, writing, speaking, and listening are themselves complex constructs, and need further specification. The following set of more detailed claims helps a task designer consider what kinds of situations and what kinds of performances are needed to give a concrete meaning to the higher-order claims, and thus what kinds of evidence will effectively give concrete meaning to proficiency variables. How detailed the proficiency model is depends partly on the purpose of the test and partly on how informative finer-grained reports would be. If proficiencies are highly correlated and there are few tasks informing each of the finer-grained claims, then separate measures for them may add nothing but noise to measures of the coarser claims they define (Haberman 2005b). The claims below start to flesh out what an assessment would look like, even though proficiency variables and reports could be at the composite levels of reading, writing, speaking, and listening.

- *Student has sufficient communicative competence in English that he or she can get the full benefit of participation in classroom activities.*
  - *W. Student can* write *English well enough to get the full benefit of participation in classroom activities.*
    - *W.1 Student has sufficient mastery of the mechanics of writing in English to produce texts with an acceptable rate of errors.*
    - *W.2 Student has sufficient mastery of the academic style in English to produce texts appropriate for the classroom.*
    - *W.3 Student has sufficient mastery of the organizational elements of written English to produce texts of the genres found in typical classroom activities.*
      - *W.3.1 Student has command of the paragraph structure, and appropriately breaks documents into paragraphs.*
      - *W.3.2 Student can clearly state the thesis of an argumentative essay.*
      - *W.3.3 Student can clearly state the conclusion of an argumentative essay.*
      - *W.3.4 Student supports arguments with evidence in writing.*
      - *W.3.5 Student appropriately uses function words that indicate the structure of the document.*

        ...
    - *W.4 Student can express ideas about a topic for which they have some knowledge or opinion using written English.*

> *. . . [breaks down to kinds of situation, purposes, and targeted performances that constitute aspects of reading proficiency as needed in the classroom contexts.]*

> R. *Student can* read *English well enough to get the full benefit of participation in classroom activities.*
>
> . . .
>
> L. *Student can* listen *to spoken English well enough to get the full benefit of participation in classroom activities.*
>
> . . .
>
> S. *Student can* speak *English well enough to get the full benefit of participation in classroom activities.*
>
> . . .

Only part of the breakdown is shown here. To complete the design, W.1, W.2, W.3, and W.4 would require further elaboration which will almost certainly produce lower-level claims. (We note in passing that being able to specify claims at this level does not mean tasks will need to be one-to-one to assess them; we will see how richer tasks, with more fidelity to valued real-world situations, can provide evidence to support multiple claims. We may have to do some work to make proper sense of the complex data, some in identifying the evidence and some in properly modeling its relationships in the statistical model, but more importantly in designing tasks that will produce the necessary evidence.)

There are some general terms in these standards that require further specification. For example, what is meant by "full benefit of participation"? This could be difficult to measure as there are large individual differences in the benefits that native English speakers gain from participating in a class. One definition that might work is that the average benefit for people who can meet this claim is similar to the average benefit for native speakers.

Further, these claims will need to be operationally defined by the situations in which they are relevant and what we would want to see people doing in those situations to consider the claim satisfied—that is, we only fully understand the claim when we know the evidence that would ground it. There is some broader range of situations and actions in the real world we care about. There is some narrower range of them that is practicable to build into an assessment. For example, University C probably cares deeply about the student's ability to write term papers, but there is only time for the student to write a short essay on the placement test. Working out the connection between the two is the realm of validity argumentation—far too big to grapple with in its entirety here, but a contemporary take on the key issues can be found in Kane (2013), Messick (1994), Mislevy (2009), and Moss et al. (2006).

These claims are represented stars in the Proficiency Model in Fig. 12.1. While in the figure, the stars are unconnected, claims are more naturally thought of in hierarchical structure in which some claims are actually subclaims of more general claims. In Example 12.2 the highest level communicative competence claim is immediately decomposed into four claims relating to the modal skills of Reading, Writing, Speaking, and Listening. The example goes on to show the next level of breakdown for the Writing branch.

Writing is a high-level skill, and as such it is a composite of many component skills, and the ability to marshal them effectively in particular situations. Indeed, despite their overlap, just what "writing proficiency" means will vary across purposes and contexts; the writing proficiency needed for college classrooms is not the same as what is needed for the factory floor or the law office. That is why just saying we would like to assess writing proficiency is not sufficient to design an assessment. We really need to be more explicit about the claims we want to make and the evidence we need to see. The language proficiency example shows one possible breakdown (based on Deane and Quinlan 2010). However, these next-level skills are themselves compound. The example shows a breakdown for the next level of Claim W.3 (organization in writing). The next level starts to identify some of the elements that go into organization. At this point we can start to see how we might be able to find evidence for the specific claims within a student's writing and how we might design tasks to elicit that evidence. We must get into at least this level of detail to be able to effectively use the claims as a basis for test design.

In the specification of such a hierarchy of claims for the assessment, the assessment designer must determine which claims are intended to be reportable claims and which are used to support the structure and design of the resultant assessment. *Reportable claims* are those that are expected to directly appear on the score report as actionable items. That is, as pieces of information that are intended to be sufficient for decision making. By implication, this means that certain claims have an expectation (preferably explicit) of reliability. In the example provided above, the main claim and the claims based on the four modal skills (Claims R, W, S, and L) are intended to be reported and used for the placement decision. The claims that are lower in the hierarchy provide conceptual support for drawing the conclusions specified in the primary reportable claims, as well as inspiration for task design.

If the purpose of the assessment were different, say providing diagnostic feedback, some of the lower-level claims may also be designated as reportable claims. Because the purpose of diagnostic feedback does not demand as high a reliability, the claims used for only this purpose do not require as much evidence to support them. However, care needs to be taken on score reports that are designed for mixed purposes to ensure that the test user clearly understands the distinction between primary reporting variables that are supported by larger bodies of evidence and secondary diagnostic feedback variables that have weaker evidential support.

Another important issue is to check the alignment of the claims and the decisions that will be made with them. Suppose University C plans to offer three sections: one emphasizing Reading and Writing, one emphasizing Speaking and Listening, and one covering all four topics. There is also an implicit fourth section, which corresponds to placing out of the language course altogether. Can we properly place students on the basis of these claims? Yes. Students for which Claim S and Claim L hold but not Claim R and Claim W are best suited for the first section; students with the opposite pattern are best suited for the second section; students for which three or more of the model claims do not hold are best suited for the third section; and only students for which all four claims hold should be excused from the English course completely.

Recall the influence diagram presented in Sect. 4.5.1. A key lesson from that diagram was that information has value when it can be used to make better decisions than could be made without the information. In terms of the claims it means that the claims for which the assessment provides evidence must provide information that an educator can use to make conditional decisions that are better that what could be done without the evidence.

In summary, claims provide an explicit representation of what assessment results need to address for the intended purpose. They determine the "evidence about what" the assessment will need to elicit. They hold implications for the required test composition and length (to support required levels of reliability for reportable claims). Claims per se can be evaluated with respect to qualitative characteristics: their usefulness, reportability, and fit to the purpose of the assessment.

12.2 The Student Proficiency Model

The claims that are established for a given assessment define, in terms of intentions and semantics, the complex of knowledge, skills, and abilities to be assessed. A proficiency model defines their counterparts in the syntactic space of the psychometric model, including the relationships among them. Specifying their relationships to observable variables in tasks defines them operationally. Section 12.2.1 describes how claims are organized through the use of *proficiency variables*. Bayesian networks are useful for representing proficiency models when the models contains more than one proficiency variable; Sect. 12.2.2 describes how to draw the graphical structure in this case. Section 12.2.3 describes how to define *reporting rules* which go from the proficiency variables (or, more properly, probability distributions over them) to scores that are reported.

12.2.1 Proficiency Variables

Proficiency variables are the machinery through which data from student performance is synthesized in some form as evidence for claims. As mentioned in the previous section, there is not necessarily a one-to-one relationship between claims and proficiency variables. Mislevy, Almond, and Steinberg (2002b) describes a number of approaches an assessment designer can relate claims and proficiency variables. The following two fit particularly well with Bayes nets proficiency models.

- One can encompass multiple claims with a single proficiency variable with a finite number of levels. Each value of the proficiency variable matches up one-to-one with a particular claim or set of claims, as discussed in Example 12.2. The American Council on the Teaching of Foreign Languages's guidelines for reading, for example, have 11 levels (Swender et al. 2012). They range from Low Novice, in which the student can typically only use language in the reading modality in rudimentary ways, up through Superior. Each level is described in terms of several kinds of things a typical student at that level can do, in situations with certain key features, each of which could be formalized as a claim in its own right. An excerpt from the Mid-Intermediate level includes the following:

  > At the Intermediate Mid sublevel, readers are able to understand short, non-complex texts that convey basic information and deal with basic personal and social topics to which the reader brings personal interest or knowledge, although some misunderstandings may occur. Readers at this level may get some meaning from short connected texts featuring description and narration, dealing with familiar topics. (p. 23)

  Exactly what "able to read" means would need to be specified in terms of what kinds of performances are expected in what kinds of situations; this statement does not yet say what the evidence needs to be to support a claim like this. However, the features of tasks that would be required begin to appear in the statement. Note that the phrase "dealing with familiar topics" indicates information is needed about the relationship of a student and a text, since a topic that is familiar to one student may not be familiar to another. The intent is that statements within a level go together well enough to characterize a student in terms of a single level, although there will be some performances above or below that level. This diffuseness is a cost of supporting many distinct claims with a single proficiency variable.

- An alternative approach is useful when claims concern being able to perform at various levels in certain kinds of situations, and the situations require multiple proficiencies in various combinations and at various levels. Distinct proficiency variables are then used to maintain belief about distinct aspects of knowledge and skill, and a claim is associated with particular patterns across them as they are called upon in settings that stress or combine them in different combinations. Students' proficiency in such a domain can be described in terms of which skills they possess at what levels (via proficiency variables), tasks can be described in terms of which skills they require (via task-model variables), and the outcomes expected from any particular matchup can be described in terms of values of observable variables. A claim can be stated about a student's likely performance in tasks with a given configuration of features. The evidence for such a claim is contained in the proficiency as the joint distribution for the particular skills in the particular combinations that are called for by tasks with these features. Mixed-number subtraction, and cognitive diagnosis models in general, are a familiar special case of this approach.

The student proficiency model thus describes the possible states of knowledge, skill, and ability that we expect to see among the members of the target population, as seen through the lens of the model. Different values of the proficiency variables correspond at some level to claims.

Depending on the purpose of the test, not all possible states of a proficiency model are interesting. In the Language Placement Test example, it is not necessary to make distinctions among students who have mastered all the material covered in the course. Similarly, we do not need to cover very low states of English proficiency as such students would not apply or be admitted to the University. Thus, which states of proficiency we consider is colored by the purpose of the assessment.

The usual way to describe the proficiency state of a student is through one or more proficiency variables. This produces a factored representation of the possible proficiency states. For example, in the Language Placement Test example, it is natural to introduce variables to represent *Reading*, *Writing*, *Speaking*, and *Listening*. A *proficiency profile* is a set of values for each of those variables, and each proficiency profile represents a possible state of proficiency for a member of the target population.

In addition to deciding which aspects of proficiency to represent as proficiency variables, the design team must decide whether the variables are discrete (categorical) or continuous. Discrete variables seem more natural when the purpose of the assessment is to classify students into groups: those for whom a set of claims hold, and those for whom the claims do not hold. Continuous variables seem more natural when the purpose requires rank ordering the students as in selection decisions.

Even though the choice between continuous and discrete variables seems important, it is actually fairly easy to derive categorical scores from continuous variables and continuous scores from discrete variables, as we did in Example 10.1. To get a categorical score from a continuous variable, all that is needed is a set of *cut scores* which divide the continuous space into reason. These are often set by standard setting committees, and there is a substantial literature on various methods for setting the cut scores (Hambleton and Pitoniak 2006). Going from discrete to continuous, there are two distinct possible methods. The first is to pick a state of the proficiency variable and report the probability that the student is in that state. The second is to assign a numeric value to each proficiency state, and to take the expected value. Section 8.5 describes one method for translating between continuous and discrete variables in detail.

Another consideration when choosing between continuous and discrete variables is the algorithms used to update the proficiency model when evidence is observed. Chapter 5 develops the model for discrete variables. If continuous variables are used instead, then many of the summations in that algorithm become integrations. In the usual models for educational testing, proficiency variables are parents of the observables. If there are multiple continuous proficiency variables and the observables are discrete, then the required integrals cannot be solved in closed form (Lauritzen 1996). However, this case is essentially Multidimensional Item Response Theory (MIRT), and approximation algorithms have been studied in MIRT (Reckase 2009).

Whether categorical or continuous, the proficiency variables must be defined well enough to pass the clarity test. The claims are useful for providing effective definitions of the variables. Consider the case of an ordered categorical variable. When comparing learners in one state to learners in the next higher state, there should at least one additional claim that holds for learners in the higher state. Thus, the claims provide the target definition of the variables, as well as the task features and the performance expectations around which tasks will be constructed.

In science education, for example, advances have been made in the topic of *learning progressions* (Alonzo and Gotwals 2012). A learning progression is marked by increasingly levels of sophistication in reasoning in a domain, which are codefined in terms of features of task situations and expected performances that are evidence of performance at those levels. Zalles et al. (2010) show how build assessments around learning progressions using ECD, and West et al. (2012) show how to model the resulting data in a Bayes net.

The relationship between claims and continuous variables is a little bit more complex. As a working definition, the claims should map to a specific point on the scale, i.e., the claim should hold for learners above a certain point on the scale. This is difficult to define, of course, and especially thorny if tasks have been created beforehand without regard to claims or cutpoints. It can then be difficult for experts to know exactly where a claim should fall on a scale without an experimental study. For early stages of the model building process it is sufficient to assign claims to falling on high or low parts of the scale. This is similar to the procedure of *item mapping* (Beaton and Allen 1992). Item maps place items along the scale at a point where 50 % (or some other chosen fraction) of the students get the item correct. The software package ConceptMap (Kennedy et al. 2006) does this visually, producing a graph with students on one side and items on the other. Items, however, are not pure representations of claims. They are only one possible realization of a task for which the claimed skill is required and their difficulty may fall higher or lower on the scale than the claim. Wilson (2004) tackles the problem from the opposite direction, more in line with the ECD approach advocated here: having in mind a theory of the construct and constructing tasks that are intentionally instances of targeted performance in targeted situations.

When the proficiency model contains more than one proficiency variable, some claims may require more than one proficiency to be satisfied. For example, consider the claim that a student can solve a mathematical word problem. If Reading and Mathematics are represented by two separate variables, then a certain level of both would be required before the claim is met. Care must be taken in the case where most of the claims defining a particular proficiency level require multiple proficiency variables. The Biomass example in Chaps. 14 and 15 and West et al. (2012) provide discussion and examples on this point. In an assessment with such tasks, it helps to have some tasks with just one or perhaps two parents to help define the scales.

12.2.2 Relationships Among Proficiency Variables

The proficiency model defines the set of possible proficiency profiles that an examinee could have. To make the proficiency model Bayesian, we must define a probability distribution over the set of possible proficiency profiles. This distribution should be based on the target population of test takers; that is, if member of the population is selected at random, the proficiency model should provide the probability that the student has a given proficiency profile. (Chapter 13 discusses the student-specific version of the proficiency model used in scoring.)

When the proficiency model contains multiple variables, an important part of specifying the proficiency model is establishing the dependence structure among the variables; that is, determining the graphical structure. The key is the pattern of conditional independence relationships among the variables. This can be difficult to do when working with domain experts who are unfamiliar with graphical modeling, and may want to produce hierarchical content-based breakdowns of the domain rather than graphical models.

There are a number of reasons why we might draw an edge between two proficiency variables. In some cases they represent a *part-of* relationship, where one variable represents a subskill of another. Another important case is the prerequisite relationship, where a certain level of one skill must be acquired before the second one can be acquired. Or skills can just be correlated for a number of reasons which are not necessarily causal. In particular, if two skills are always taught at the same time in the curriculum, they might be correlated as they are both a function of students' progress through the course.

Example 12.3 (Language Placement Test Proficiency Model). *After some discussion the design committee comes up with the following list of potential proficiency variables: Communication, Reading, Writing, Speaking, Listening, Grammar, Correspondence, Conversation, Sociolinguistic, Purpose (of the communication), Register (linguistic patterns appropriate to the communication). (See Mislevy et al. (2002b) for a more detailed discussion of an ECD model for communicative competence). The committee agrees that the* *first five variables are important for reporting. Others are interested in the remaining variables as a better reflection of their theories of communicative competence.*

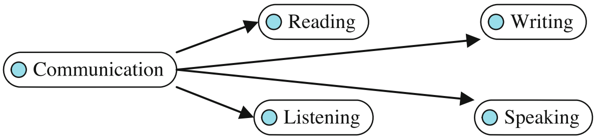


Fig. 12.3 Partial language testing proficiency model showing a part-of relationship Reprinted from Almond et al. (2006a) with permission from ETS.

The first and most obvious relationship is the relationship between the overall Communication variable and the four variables representing the modal skills, Reading, Writing, Speaking, and Listening. This is a part-of relationship, something that is commonly encountered in many knowledge engineering problems. Because of this relationship the variables will be dependent, and there should be an edge between them. Generally with part-of relations, the edge should be oriented from the larger concept to the smaller ones. This produces the graph shown in Fig. 12.3.

Structures like those shown in Fig. 12.3 occur often enough in practice that it is worth examining some of their properties in more detail. Suppose that all of the evidence models in the assessment use one or more of the four modal skills as their parents, and that there are no tasks that provide direct evidence for *Communication*. In this case, the variable *Communication* is identified only indirectly through the prior weights we place on the edges in Fig. 12.3. Depending on the strength of that prior distribution, there can be a ridge or multiple modes of the posterior distribution, which may cause a Markov chain Monte Carlo (MCMC) algorithm to mix poorly (Almond et al. 2008). Generally it is necessary to fix one or more of the distributions, either the marginal distribution of *Communication* (analogous to setting the population distribution in an item response theory (IRT) model to standard normal) or the conditional distribution of one of the modal skills given *Communication* (analogous to setting the loading of one variable to 1 on each factor in a factor analysis).

Example 12.4 (Language Placement Test Proficiency Model, Continued). *The design committee decides that the single communication variable is not enough to explain all of the correlation among the four modal variables. In particular, they note that there is usually a prerequisite relationship between Reading and Writing and between Listening and Speaking, because at least some measure of the receptive skill is usually required for mastering the corresponding productive skill. In the case of prerequisite skills, it* *is again usually best to orient the arrows from the skill that is acquired first to the skill that is acquired last.*

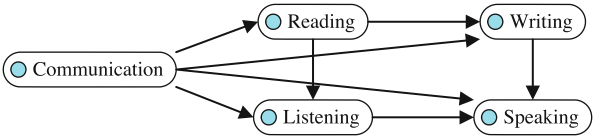


Fig. 12.4 Proficiency model for language testing using only the four modal skills Reprinted from Almond et al. (2006a) with permission from ETS.

The design committee also feels that there may be additional correlation between Reading and Listening (both receptive skills) and Speaking and Writing (both productive skills) beyond those explained by the general Communication variable. Therefore, they decide to place additional edges between Reading and Listening and Speaking and Writing. Care must be taken here, as this edge states that the skills are dependent even after conditioning on Communication (which presumably incorporates common factors such as vocabulary and grammar). Since these edges represent correlations, they can be oriented in either direction. As the intended population mostly contains students for whom English is a second language, and these academic second language learners usually spend more time with the written language, we orient the edges from Reading to Listening and Writing to Speaking. The resulting graph is shown in Fig. 12.4.

There were several places in the construction of Fig. 12.4 that the design committee had to make arbitrary decisions about which direction to orient the edges. Although some authors put great store in using Bayesian networks to represent causal relationships, the functional meaning of the direction of the arrows is mathematical. By orienting the edge from *Reading* to *Listening* the committee is stating that they would rather specify the distribution of *Listening* conditioned on *Reading* than the distribution of *Reading* conditioned on *Listening*. Orienting the edges in the causal direction is usually preferred because (a) it usually yields Bayesian networks with smaller treewidth and (b) it is usually easier for subject matter experts to provide prior probability distributions for edges oriented in direction that corresponds to their opinions on causality. Whether or not the arrows point in a causal direction does not matter mathematically in using a Bayesian network; the concern is whether it expresses an appropriate joint probability distribution over the proficiency variables.

Example 12.5 (Language Placement Test Alternative Proficiency Model). *A group of experts on the design committee feels the proficiency model that only contains the four modal skills does not really match their theory of language use. They propose four new proficiency variables: Grammar,* *Correspondence competence, Conversation competence, and Sociolinguistic competence. Grammar explains much of the observed dependence among the four modal skills, and Correspondence and Conversation competence explain the extra correlation among Reading and Writing and between Speaking and Listening. Sociolinguistic competence represents capabilities that are not well represented in the modal proficiency model. Figure 12.5 shows the graphical representation of their proficiency model.*

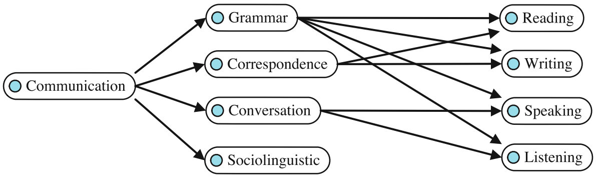


Fig. 12.5 Proficiency model for language with additional communicative competence skills

Reprinted from Almond et al. (2006a) with permission from ETS.

There is seldom a single view of competence in a domain. Research and experience might support alternative views, and it is generally not hard to come up with more proficiencies and alternative structures than can possibly be expressed in one model. The point is not to build a comprehensive psychological model of proficiency in the domain. Rather, it is to build a model that draws on research but is simply sufficient for the purpose of the assessment. Some aspects of competence may not be involved, others may be moot because of the testing population, and for others, fine distinctions and alternative theories are not necessary for the job at hand. If the design committee thought that reporting about *Grammar*, *Correspondence*, *Conversation*, and *Sociolinguistic* competence was important (even if only for diagnostic purposes or as part of aggregate reporting), then the model of Fig. 12.5 has a clear advantage over that of Fig. 12.4. If reporting on those skills is not important, then either model should work adequately and the simpler model should be easier to build and maintain.

Another consideration is how much evidence is required to distinguish between different proficiency profiles. This cannot be calculated exactly until the evidence models and assembly models are built. A heuristic often used at this stage is to consider that between six and ten independent pieces of evidence are required to define a scale with modest reliability. Thus, the model of Fig. 12.4 would require a minimum of a 40-item test to get reliable estimates for all of the proficiency variables, while the model of Fig. 12.5 would need an 80-item test. If testing time is an important consideration, the model with fewer proficiency variables is clearly favored. This heuristic is only a rough guide for the initial stages of discussion. In most testing situations some kind of pretesting will be required to ensure that the test provides adequate evidence for the desired uses.

Still another consideration is whether the proposed kinds of tasks provide the evidence needed to distinguish among the proficiency profiles. This should have been addressed earlier in working through claims and evidence. However, this is another place to recheck this central issue. As an example, consider the *Sociolinguistic* skill in Fig. 12.5. If all of the tasks tap only the four modal skills, *Reading*, *Writing*, *Speaking*, and *Listening*, then the assessment will provide no direct evidence for *Sociolinguistic* competence. The design committee would need to revisit the claims they intended to address concerning sociolinguistic competence, and the evidence they indicated they needed (or failed to indicate they needed). The assessment would require evidence addressing sociolinguistic skills, which might require extending the existing evidence and task models, or constructing new ones; for example, in which sociolinguistic demands were varied in principled ways and the *Sociolinguistic* skill as well as one or more modality skills were parents of observables evidencing sociolinguistic skills.

Example 12.6 (Language Placement Test, Continued: Unused Proficiency Variables). *In the initial draft of this problem, the design team identified two potential proficiency variables, Purpose and Register, that were not included in either Fig. 12.4 or 12.5. Although these concepts appear in the complete list of claims, the team felt that it was not important to report on them. Therefore, they do not appear in the proficiency model.*

It was important, however, that the tasks have a representative sample from the various purposes and registers. Therefore, rather than motivating proficiency variables to report on, these concepts motivate the definition of task model variables that will be used to guide task construction and test assembly, and thus help implicitly define the scope and meaning of the variables that are in the proficiency model. First, each task needs to be tagged with two task model variables that indicate the Purpose of the communication in the task (e.g., `provide information` *or* `make request`*) and the Register of the task (e.g.,* `informal [class discussion]` *or* `formal [lecture]`*). Second, assembly rules need to be added to the assembly model that specify the target distribution of values for these task model variables in the final form (Sect. 7.4).*

Remember that the purpose of drawing the graph is, as a step toward creating a joint probability distribution of the proficiency variables, to serve as an appropriate prior distribution for scoring members of the target population. After the graphical structure of the model is agreed upon, values need to be chosen for parameters of the conditional probability tables. Section 15.1 describes one possible procedure for this later quantification step.

It is also important to keep in mind that the role of edges in the graph is to represent patterns of conditional independence in the joint distribution of the proficiency variables. The substantive considerations discussed above that suggest dependencies due to part-of and prerequisite relationships are important. However, there if there is prior research in the field using these variables, there may also be correlation matrixes resulting from factor analytic studies or structural equation models. A useful trick is to look at the inverse of the correlation matrix. Zeroes represent instances where variables are conditionally independent given the other variables in the model (Dempster 1972; Whittaker 1990). Almond (2010a) shows how this fact can be used to derive the graphical structure of a proficiency model.

12.2.3 Reporting Rules

While the role of the proficiency model is to describe the distribution of proficiency profiles within the target population, the role of a *score* is to provide information about which proficiency profile (or implications of proficiency profiles for performance) a particular student (or group of students) has. In the case of a Bayesian scoring model, this will come from a probability distribution over possible proficiency profiles. The proficiency model serves as a common prior distribution for all students taking the exam. When we observe responses $\mathbf{X}_i$ from a particular student, we can calculate $\mathrm{P}(\boldsymbol{\theta}_i|\mathbf{X}_i)$, a student-specific posterior distribution over the space of profiles, using the methods of Chap. 5 when the proficiency model is a Bayes net. This posterior distribution is called a *scoring model* in Chap. 13. It represents our state of knowledge about that student.

Reporting the entire posterior distribution is usually not practical, either when the users are humans or computer processes. When the score users are humans, it is difficult to represent and many will not know to interpret it. Moreover, it does not directly provide answers to questions users of either kind care about. Instead, we apply *reporting rules* that map states of the scoring model into scores that are shown on a score report when the users are humans, or a value upon which decisions are based when the user is a computer process. In the case of Bayesian scoring models, these are *statistics* of the posterior distribution. Formally, a statistic is a *functional* (an operator that maps a function, in this case the distribution function, into a scalar value) of a probability distribution. Many familiar statistics are useful for developing reporting rules.

Sections 12.1.2 and 12.2.1 discussed ways in which proficiency variables are related to claims. Reporting rules can thus use information in posterior distributions over proficiencies to provide quantitative evidence for given claims. For instance, in the examples below *Most Likely Value* and *Probability of State* are suitable when claims correspond to levels of a proficiency variable. *Most Likely Explanation* is useful when a claim is expressed by a set of profiles over proficiencies. *Expected Score on Market Basket* gives an indication of a claim in terms of expected performance across a representative tasks that implicitly define a claim, whether one or several proficiencies are involved.

Many statistics only involve a single target proficiency variable. In that case, they can be calculated from the marginal distribution of that variable. For Bayesian network models, the most commonly used statistics are:

Most Likely Value. The value of the target variable that has the highest posterior probability. This is also known as the *mode* or *maximum a posteriori* (MAP) score.

Probability of State. It is easy to calculate the probability that the target variable takes on one of its possible states.

Probability of Reaching Cut Score. It is also easy to calculate the probability of any sets of states of the target variable. When the states of the target variable are ordered, one state can be chosen as a cut score, and the statistic used in the probability that the target variable is at or above the cut score is reported.

Marginal Distribution. For a discrete Bayesian network, it is possible to report the posterior distribution for the target variable as a vector of numbers. This is easy to express graphically using a divided bar chart.

Expected Value. If the levels of the target variable are assigned numeric values, then the expected value of the target variable can be calculated. This is also known as the *mean* or *expected a posteriori* (EAP) score. Although the most obvious assignment of consecutive integer values requires the sometimes questionable assumption that the proficiency levels are equally spaced, in some cases the mean provides a single number summary. The discrete IRT model in Sect. 6.1 is such a case. Indeed, computer programs for continuous IRT models often work with just such an approximation "under the hood" (Bock and Aitkin 1981).

Standard Deviation. Again, if the values of the target variable are mapped to numeric values, the standard deviation is easy to compute and provides a simple description of the posterior uncertainty about the correct score level (Bock and Mislevy 1982).

If instead of a single target proficiency variable there are multiple target proficiencies, then there are a number of additional statistics of their joint distribution that are useful. For example, the *Activity selection process* in a coached practice system might check the joint distribution of *Space-Splitting* and *Canopy-System*, so that if the former is `high` and the latter is `low`, it can trigger a review of the canopy system.

Most Likely Explanation. This is the assignment of values to the profile that has the highest posterior probability. This is not always the same as the mode of each variable taken individually. Section 5.6.1 describes the special algorithm used to calculate this.

Probably of Hypothesis. We can also calculate the expected value of any *hypothesis*, that is set of possible proficiency profiles.

Expected Value of Function. The expected value of any function of the target variables is also straightforward to calculate.

For Bayesian networks these values are easy to calculate when all of the target variables fall into a single clique. If that is not the case, often the desired statistics can be calculated by successively conditioning on the target variables in turn.

Defining a canonical collection of tasks called a *market basket* (DeVito et al. 2000) allows a number of statistics that can be used to communicate results as expected performance on tasks chosen to exemplify claims. Let $m \in \mathcal{M}$ be a task from the market basket and let $\mathbf{Y}_m$ be the observables from that task. The evidence model for Task m provides the conditional distribution $\mathrm{P}(\mathbf{Y}_m|\boldsymbol{\theta})$, and the scoring model provides $\mathrm{P}(\boldsymbol{\theta}_i|\mathbf{X}_i)$. Assuming that the operational task observables, $\mathbf{X}$, and the market basket observables, $\mathbf{Y}$, have been calibrated to the same scale using the methods of Part II, we can calculate a predictive distribution for the market basket observables:

$$\mathrm{P}(\mathbf{Y}_m|\mathbf{X}_i) = \int \mathrm{P}(\mathbf{Y}_m|\boldsymbol{\theta})\mathrm{P}(\boldsymbol{\theta}|\mathbf{X}_i)\partial\boldsymbol{\theta}. \tag{12.1}$$

This can be used in a variety of interesting ways:

Expected Score on Task m. We can make a prediction about how likely the student is to obtain various outcomes for Task m. This can be useful if the operational tasks are not published for security reasons. Prediction of performance on the disclosed tasks provides a proxy for actual performance on the operational tasks.

Expected Score on Market Basket. Summing the expected scores across all tasks in the market basket creates an expected score for that form, no matter what operational form an examinee may have taken. As the market basket form is disclosed, this score will have a concrete meaning for the score users.

Expected Weight of Evidence on Task m. The predicted score on the market basket task provides the required information for calculating an expected weight of evidence that the market basket task provides for a hypothesis of interest (Sect. 7.3.2). Tasks that have high expected weight of evidence are interesting because they are on the cusp of what we think that the student should be able to do.

The list of reporting rules presented above is by no means exhaustive, and concentrates mainly on reporting rules that are useful when the proficiency model uses Bayesian networks. So how should the design committee select the right reporting rules to use with a given assessment?

The answer lies in returning to purpose of the assessment, which is to establish whether or not the claims hold for a given student. Therefore, the scores should all be related back to the claims, e.g., such-and-such claim usually holds when the score is above a certain level. The score report should be designed in such a way that it is apparent to the score user what evidence is provided by the assessment results for the claims in the ways discussed previously.

When designing the score report, it is important that it be understandable not only for the design committee but also for the score users who are its intended audience. It is often helpful to build a *prototype score report*, a mockup of the final score report using artificial but realistic numbers, and show it to some potential score users. An important use here is to ensure that the claims the design team chose to make the focus of the assessment are in fact the ones the user community values. Adjusting the focus of the assessment is much less expensive at this stage before production is underway, than later when considerable effort has been expended designing and testing tasks.

Another important use for the prototype score report is as a vehicle for explaining the psychometric consequences of a design decision. The understanding of complex psychometric concepts like reliability is often rudimentary in the score user population. If the design team is trying to decide between two formats for the assessment, a longer one with higher reliability and a shorter one with lower reliability, the best way to get feedback from potential score users is to mock up score reports that illustrate the consequences of the design choice. This frames the question in terms of how they will eventually use the information from the assessment.

Example 12.7 (Language Placement Test, Continued: Prototype Score Report). *To decide between the modal model and the more complicated communicative competence model, the design team builds two prototype score reports. The central display for the modal model is a stacked bar chart, shown in Fig. 12.6.*

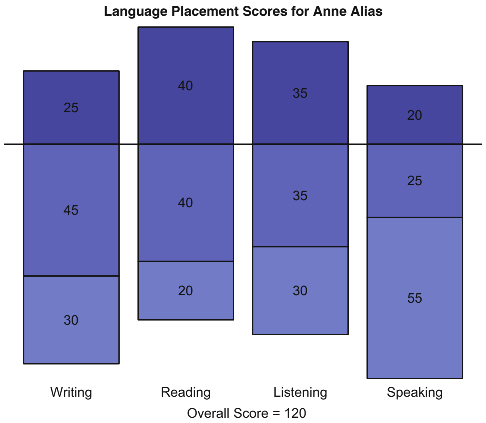


Fig. 12.6 Prototype score report for Language Placement Test. Reprinted with permission from ETS.

A small simulation study convinces them that the reliability of scores based on the Communication node is too low to report as an overall score (See

Exercise 12.10). They decide to use instead the sum of the probabilities that the student is in the highest (passing) state in each of the four modal skills. They multiply this by 100 to get a score in the 0–400 range.

The stacked bar chart in Fig. 12.6 is a useful way to display the results from Bayesian networks. Almond et al. (2009a) describe some considerations in the design. First, note that the probabilities are scaled to percentages, as teachers are more used to working with probabilities expressed as percentages than with raw probabilities or fractions. Note also that the colors for the various levels of the proficiency variables differ only in intensity (value, in the graphic designers' {hue, saturation, value} model). There are two reasons for this: first, it means the report is understandable even for individuals with limited color perception, and second, it means that the report is understandable even if it printed or copied using a printer that can only render grayscale images. Almond et al. (2009a) also describes some variations on this idea that can be used for reporting at the classroom (or other student group) level.

Another important question in score report design is whether or not the way the information is presented is understandable for the target audience. To answer this question, the prototype score report could be used as part of a formal usability study. In this kind of study, representatives of the target score users are asked to perform tasks using the prototype score reports (such as comparing two hypothetical examinees). The design team can then get feedback both about how the target users reacted to the report and how successful they were at performing the requested tasks.

Implicit in the definition of these reporting rules is the need to define a "mastery" level for each of four modal proficiencies, *Reading*, *Writing*, *Speaking*, and *Listening*. The specification of the third level as the origin line in Fig. 12.6 implies that the third level of the variables are associated with mastery. Thus, it is reasonable to ask if the claims associated with the highest level of the proficiency variable correspond to mastery, or in this example, the ability to get value out of college courses without supplemental instruction in English. No matter how much effort goes into this design, the way students actually interact with the assessment is sure to bring surprises. If the assessment has high stakes then it is worth doing a formal validity study using the implicit mastery designations. Even for a more moderate-stakes examination it is worth reviewing the implicit standards after actual student performance data from pilot test are available.

In review, thus far we have discussed the initial specification of the assessment goals and population and their implications for development of the assessment claims and supporting proficiency model for the domain. The overall purpose of assessment and target population are used to derive the identification of claims to be made from the assessment (represented by stars in Fig. 12.1), which in turn helps to specify the number and nature of student model variables (represented by green circles), which themselves have conditional dependencies and interdependencies (represented by arrows) in acquisition and implementation—all of which are used to develop and inform the reporting rules that will link the student model to the characteristics that appear on the score reports.

Of course, to produce score reports for individual students requires evidence from individual students. Section 12.4 looks at how that evidence is represented in the formal assessment design. However, before discussing evidence, it is worth talking about the circumstances under which it will be gathered. This is the province of task models.

12.3 Task Models

A task model is a set of detailed descriptions of task characteristics for a family of similar tasks. The essential form appears in Hively et al. (1968) and Osburn (1968), and Gierl and Haladyna (2012) present more recent work that takes advantage of developments in digital technology. This discussion draws on Almond et al. (2002a) and Mislevy et al. (2003b)

A task model provides a complete framework for the design and development of a family of assessment tasks, as well as information used by the assembly model in constructing test forms and by the evidence model is calculating the evidentiary strength of possible observations. The fundamental elements of a task model include the nature of the material presented to the student (such as directions, initial prompts, response options, etc.), the characteristics of activities that the examinee must undertake to complete the task, and the nature of recorded information as a result of the examinee working on the task (response, time to complete the task, etc.).

At an abstract level, a task presents a collection of material to the examinee—the *presentation material*—and captures some kind of response—the *work product*. For a family of tasks to be similar, there must be some similarity in both the presentation material and the work products. Thus, a primary goal of the task model is to provide a description of the range of and format of the allowable presentation material.

Presentation material is the set of all representations and materials that are displayed, or may be displayed, to an examinee during the delivery of an assessment task. These can include instructions for the task (e.g., "...choose the best answer...") , the initial prompt from which the examinee begins work (e.g., "A train leaves the station heading east at..."), the various ways the examinee can interact with the task (in familiar simple cases the various response options available for the examinee to choose or in other ways indicating areas for them to respond in), as well as any multimedia presentation components such as video or audio clips, graphics, animations, etc. These presentation materials are represented in Fig. 12.1 as the papers, grid, and video clips that appear in the upper right portion of the task model.

The *work product specifications* describe the elements of examinee performance that are recorded and retained as a result of interacting with the assessment task. They define the important aspects of examinee performance, both in process and outcomes, that are used for scoring and for other important data functions in assessment. A computer-based hydraulics system troubleshooting task, for example, can present work products in the form of the final configuration of the system and a file with all the troubleshooting actions and time stamps. Realized work products will be evaluated in terms of the specifications in the evidence model, to identify the relevant evidence from the performance evoked by the task. The work product is represented graphically in Fig. 12.1 as the jumble of shapes in the upper left hand section of the task model.

In a fully detailed task model, the specifications for both presentation material and work products include semantic and functional descriptions of the forms they will take, but do not specify the implementation details. This is so that test developers can focus on the elements of tasks and performance that embody the elements of the argument, but can be rendered in forms that may vary over times and places in different formats or presentation platforms, or adapted to students with disabilities in ways they can better access stimulus materials or produce their responses (Hansen and Mislevy 2005). The Presentation Model discussed shortly contains the information that programmers will working on the project will know how to store the material, and what kind of software and hardware will be required to display it.

One way to develop a task model starts earlier in Domain Modeling: Build some prototype tasks that designers agree generate evidence for the targeted claims, then look at what properties of the examples are generalizable. This produces evidence paradigms and task paradigms, which can then be refined to produce evidence and task models. The key is using the prototypes to elicit discussion about the nature of evidence being sought; that is, how the intuitions behind the prototypes illuminate classes of situations with certain key features, asking examinees to carry out certain kinds of work, and looking for certain key features in their performances. To that end, we consider one possible type of task that might be used in the Language Placement Test example.

Example 12.8 (Language Placement Test, Continued: Lecture Clarification Task). *Looking through the claims associated with the assessment, the design committee decides that one of the more important claims is that a student who places out of the remedial language program is capable of asking a question to clarify a particular ambiguous concept introduced in a lecture. To that end the committee looks at a task in which the student is presented with a short video clip of a lecture and is then asked to produce a question requesting clarification about a point raised in the lecture. This task model will have two pieces of presentation material: the lecture excerpt, stored as a video file, and the written instructions to the student, stored as text with markup.*

Often it is important to know if a task has a particular feature. In the lecture task, the length of the video clip, the topic, and the ambiguous point are salient. *Task model variables* label tasks according to specified features. Various processes that manipulate tasks can use task model variables to select tasks with desired properties, as distinguished by possible values of those variables. For example, task model variables can be used in automated test assembly and adaptive item selection (Sect. 7.4).

Part of the definition of a task model is a list of task model variables that are associated with tasks generated from the task model. That task model defines the range of acceptable values for the task model variables, and it defines *specification rules* which describe how the value of the task model variable relates to the presentation material or the potential work product. In some cases, the task models can be sufficient to allow automated creation of tasks (Gierl and Haladyna 2012).

Specification rules can work either forward or backward. Consider the lecture task and a task model variable that encodes the speech rate of the lecturer. To use the rule in the forwards direction, start with an existing lecture video clip and measure the speech rate. To use the rule in the backward direction, there are two ways. Suppose a task has been requested with a `slow` speech rate. One way to use the rule would be to search through a library of video clips to find one that meets the criteria. Another would be to videotape a new lecture, and to instruct the lecturer to speak slowly. The backward use of specification rules is useful in automatic item generation; the generator can select a random or prescribed value for the variable and then generate presentation material that matches. Specification rules can be quite useful as well when the item generators are test developers. For complex or unique tasks that require human creativity, the rules function as guidelines and constraints, around which unique tasks will be constructed. Mislevy et al. (2002c) discuss roles that task model variables can play. These roles are not mutually exclusive, as variables are often used for more than one purpose. The roles do provide rationale for when and why task model variables should be created.

1. Task Construction. A fundamental tenet of evidence-centered assessment design is that tasks must be built, selected, or recognized to provide the evidence required to support the claims of the assessment. A key role of task model variables is to tell test developers which features can be manipulated to meet those goals. This includes both direct manipulation, where the material is authored by the test developer, and indirect manipulation, where the test developer seeks out material that meets the specifications.

Irvine et al. (1990) introduce the term *radical* for features that change the evidentiary properties of the task, and *incidental* for features that do not. Their work focused on features that drove difficulty in accordance with cognitive theory, and produced a new version of the British Army Recruitment Battery (BARB) with all items generated in real time through task models (Collis et al. 1995; Irvine 2013).

Incidentals also play an important role in the construction of tasks. In high-stakes testing situations, it is often the case that examinees will discuss difficult problems with each other. Being able to easily produce variants on a task by manipulating surface features (e.g., the names of actors in a story) means that examinees cannot select solution strategies based solely on the surface features. Incidentals are also useful in automatic item generation, where they indicate features that can be manipulated to produce variants of the task without affecting psychometric properties.

It can be difficult to tell which variables are radical and which are incidental without pretesting the tasks with members of the target population. Experience has shown that many features that should not matter, in fact do have an impact on the evidentiary properties. There may be no true incidentals; what we call incidentals are simply variables that are not central to the claims and for which the impact is much smaller than the construct-related task model variables in the targeted population.

2. Focusing Evidence. Changing some task model variables will shift the nature of the proficiencies a task evokes, thus altering the nature of the evidence that the task provides. Consider the clarifying question of Example 12.8. Suppose that we change the task so that it now asks for a summary statement rather than a clarifying question. This changes the claims associated with the task, from "can generate questions to clarify uncertainties" or "can generate summaries of information." Thus, the focus of the task has changed. These could be considered two different task models, each of which has a restricted range for the *work product: expected form* variable.

Depending on what variables are in the proficiency model, this may or may not change the graphical structure of the evidence model. In the example above, if the proficiency model is the one shown in Fig. 12.4, then the observables are likely connected to the same proficiency variables because the two task models because both models are built to evoke evidence of the same coarsely-grained proficiency variables, in this case Listening and Speaking. If the proficiency model is finer grained and includes distinct proficiency variables for listening skills concerning details and concerning gist, then changing the value of this task model variable would indicate that a different evidence model fragment needs to be used, so that the appropriate proficiency variables are updated.

3. Assessment Assembly. Often there are many more claims associated with an assessment than there are proficiency variables. In such cases, the tasks selected for a given form of the assessment are a purposive sample from all of the tasks that could be administered. An important goal of the assembly model is to ensure that the sample of tasks is representative of the claims to be made about this assessment. This can be done formally in terms of constraints on an item selection as discussed in Sect. 7.4.

Example 12.9 (Language Placement Test, Continued: Communication Purpose). *Recall that in building the proficiency model, the committee* *considered and then rejected the variable Purpose as they would not use it to organize reporting. Despite this rejection, the purpose of the communication is still important, as the claims for the Speaking variable span a number of different communication purposes.*

To make sure these claims are represented on the test, they add a constraint to the assembly model that there must be at least one task from a number of different purposes. In order to enforce that constraint, each task needs to be tagged with a Purpose variable that provides the purpose of the communication. For example, the task described in Example 12.8 would have Purpose = `clarify`*. The alternative in which the examinee was required to summarize the lecture would have Purpose =* `summarize`*.*

4. Controlling the Psychometric Properties of Tasks and Observables. Ideally, test developers would like to be able to control the psychometric properties of the items at design time. One critical role of task model variables is to identify features that are thought to affect those properties. Evidence models can pick up the values of those task model variables and model the parameters in the statistical part of the model as a function of them (e.g., Geerlings et al. 2011; Fischer 1973; Mislevy et al. 1993; Rijmen et al. 2002). Embretson (1998) illustrates how a cognitive approach can unify assessment design, task construction, and statistical modeling. This leads to the idea of predicting the statistical parameters of computer-generated tasks without any pretesting at all, as in BARB (Collis et al. 1995; Irvine 2013).

When the statistical part of the evidence model uses IRT, then Mislevy et al. (1993) show that the the difficulty parameter is both the easiest to model and the most important in subsequent inferences. This insight applies to Bayesian network models using the IRT-like DiBello–Samejima models. Even if these relationships are too noisy to use without additional pretesting, knowledge of the factors that influence difficulty can help test developers build a balanced pool of potential tasks to pretest, or allow them to reduce the size of the required pretest sample.

5. Characterizing Proficiency. This use is the complement of the previous one. After the evidence model for a task is built, then one can predict how people at various levels of proficiency are likely to perform. For any given proficiency profile, look at the expected performance on a collection of tasks that a person with that profile are likely to produce (right answers in dichotomous tasks, levels of performance in ordered-category tasks). To the extent that this collection of tasks shares common values for task model variables, those values characterize the proficiency profile.

In the previous section, we mentioned that proficiency variables were characterized using claims. In many cases the claims are grounded in abstracted descriptions of tasks, which would have implications for the corresponding task model variables. Consider the three claims: “can read a comic book,” “can read a newspaper,” and “can read a journal article in their field.” All of these have different implications for task model variables describing the text, e.g., the difficult of the vocabulary, the formality of the grammar, and the presence of pictures to supplement the text.

6. Recognizing Task Situations. In interactive tasks, such as the HYDRIVE environment for troubleshooting the hydraulics systems of the F-15 aircraft (Gitomer et al. 1995), "tasks" may be recognized as they arise rather than constructed. While each HYDRIVE task was defined globally by the initial fault and stimulus materials such as the video clip of the pilot's report of a problem, Mislevy and Gitomer (1996) showed how a delivery system can activate the use of evidence model fragments when certain conditions are recognized in a less constrained stream of actions. The current state of a problem is tracked in terms of salient features of the state and past actions—values of *dynamic task model variables*—and certain configurations signal that an instance of a task model has been realized; that is, an instance of an evidence-bearing situation described by a task model has been recognized.

The preceding discussion shows that the considerations of task models from a design perspective for the CAF have much in common with task modeling in automatic task generation, in which computer software automatically generates tasks according to specific needs of assessment (e.g., Gierl and Haladyna 2012; Irvine and Kyllonen 2002). Indeed, the automatic item generation models are subsumed under the more general umbrella of task models in assessment design, with the exception that much of the discussion of automatic item generation efforts expand the concept of task models to include both the conceptual characteristics of these models and the technical requirements for implementation in an automated item generation system.

12.4 Evidence Models

It may seem relatively simple to come up with interesting ideas for tasks and task models. We have argued here, however, that it is not optimal to try to work backward from tasks, to how to score performances, and then ask whether they even provide the evidence needed to support claims. It is better to start from claims and evidence, then begin to craft situations and performances that provide that evidence in some particular form. Even when work does begin with prototype tasks just because designers can most easily leverage their expertise in this way, the prototypes can then effectively play a domain analysis role: As examples of valued work, they can motivate discussion that brings out more explicit statements of claims and classes of evidence. Either way, we begin from a coherent qualitative argument for later interpreting students' performances as evidence. Then we can address the machinery that embodies the argument, which indicates the machinery and the processes by which evidence is identified and accumulated. This is the province of the evidence model.

The evidence model bridges a task model (and hence particular tasks that accord with that task model) to the proficiency model. The connection is easiest to express when, in operation, one evidence model will be hooked up with each task model–proficiency model pairing. In other words, we can use the same task model, but with different proficiency models, and when we do we can use a different evidence model to bridge them.

This arrangement allows for reuse of tasks in different contexts. To use a task to obtain evidence for a fine-grained proficiency model, a fine-grained evidence model is needed. To use the same task to provide evidence for a coarse-grained proficiency model, an evidence model with the correspondingly coarser grain size is needed. The difference in evidence models could be as simple as changing the proficiency variables that are parent of the same observable variables. But different sets of evidence rules can be used in different evidence models for the same task model, to identify and characterize different aspects of performance from the same work products. The observable identified from an essay could be an overall rating of its quality when the task is used for a summative purpose, for example, while several aspects of lexicon, grammar, and structure could be identified when it is used in a diagnostic assessment. Furthermore, finer-grained evidence models can identify more subtle relationships between proficiencies and performances. Different patterns of more-detailed proficiencies can be modeled as parents of different observables, whereas under a coarser-grained composite-proficiency model all the observables have it alone as their parent.

The observable outcome variables are central support pillar of the evidence model bridge. On one side, there are the rules of evidence which link the observables to the work products from which they are derived (Sect. 12.4.1). On the other side, there is a statistical model that links the observables to the proficiency variables (Sect. 12.4.2); earlier chapters describe how to model these with Bayes net fragments. In both cases, the number and nature of the observables drives most of the rest of the design decisions in the evidence model.

12.4.1 Rules of Evidence (for Evidence Identification)

The term "rules of evidence" is adapted from jurisprudence, but the analogy is not bad. In the courtroom, the rules of evidence refer to legal rulings that relate to which observations and testimony will be shown to the jury as possible evidence. In the classroom, the rules of evidence refer to the procedures used to determine which features of the work product will be considered as observables and used to update the proficiency model.

With multiple-choice and other selected-response tasks, there is usually a single-evidence rule, the key-matching rule. If the selected response (as identified in the work product) matches the key, the observable is assigned the value `correct` or `1`. If the selected response does not match, then the observable is assigned the value `incorrect` or `0`. Even this simple scoring rule hides some complex design decisions. Consider what happens when the work product is null (i.e., the student made no selection). In many assessments, the null work product is assigned the value `incorrect`; however, sometimes different scores are assigned for incorrect selections and so another value, say `omitted` is needed for the observable. Null work products from tasks that an examinee chooses to skip may be treated differently from ones that are not reached and from those that are not presented (Mislevy 2015).

Under the simple key-matching evidence rule, we cannot assign a value to the observable unless we know the value of the key. The key is the simplest example of *evidence rule data*, or task-specific parameters of the evidence rules. Other more complex examples might include task-specific details in a scoring rubric, facts to look for in a short response answer, or a composite expert concept map against which to compare each examinee's. In all cases, this is additional information that must be authored with the task if the task is to be used in this particular evidence model.

A second commonly seen type of evidence rule is the scoring rubric used by human raters when scoring a constructed response. The observable variable is the rating. In the case of holistic scoring, there is a single observable for each work product; in the case of analytic or trait scoring, there may be more than one observable per work product. In either case, the rubric should provide specific descriptions of the characteristics of the responses at each response type, and sample work products with scores and rationales for those scores.

Bejar et al. (2006) describe some considerations for human scoring. Even if the goal is eventually to use a computer program to assign values to observables, a good clean rubric for human scoring is often a good starting point. Computer scoring generally uses one of two methods: procedural methods or machine learning methods. In procedural scoring, the rubric becomes the specification for computer software (Braun et al. 2006). In the case of machine learning algorithms, the parameters of the algorithm are learned from the a sample of human-scored data (Deane 2006). Here the human scoring rubric is necessary to build the corpus of human-scored examples used to train the algorithm. Automated scoring of essays (e.g., Attali and Burstein 2006) is probably the most familiar application of this approach. Gobert et al. (2012) is an example tuned to discovering meaningful patterns in students' science investigations—i.e., automated "feature detectors" as evidence identification processes.

In any case, a corpus of examples annotated with the "correct" value[3] for the observable variables is a valuable resource. In the case of human scoring, this corpus can be used both for training raters and for checking that raters are performing consistently (Baldwin et al. 2008). In the case of procedural algorithms, the corpus becomes a valuable set of test cases for ensuring the algorithm works. In the case of machine learning, the corpus is central to both training and testing. It is worth the effort to capture a sample of authentic

[3] Rather than "correct," we should say "targeted value for training." Statistical models bootstrapped from experts' judgments can be more accurate than the experts themselves (Bowman 1963).

examinee responses for this corpus, as this will inform the design team about the unexpected ways real examinees respond to the tasks.

The downside of human scoring is the expense and the time. Even discounting the cost of performing the ratings, the logistics of getting the work product to the raters often means a delay between the time the examinee produces the work product and the time the examinee receives the scores (although in many sporting competitions, such as gymnastics and diving, the judges watch the performance and can provide a score immediately). Feedback from the task is usually given days to weeks after the assessment and the examinee has forgotten much of the thinking that went into the work product. Further, the information from the human scored responses cannot be used in adaptive task selection.

The alternative is some kind of automated scoring (Williamson et al. 2006b, provides a survey). As mentioned above, automated scoring algorithms can be divided into two categories: procedural algorithms, whose parameters are determined when the task is created, and machine learning algorithms, whose parameters are learned from a sample of scored student responses.

The key-matching algorithm used for selected response items provides a simple example of a procedural scoring algorithm. Often, a cleverly chosen work product can enable an apparently complex task to be scored procedurally. For example, selecting a word from a paragraph is a task that seems open-ended to the examinee, but that can be easily scored by the computer. Chapter 14 contains additional examples. As a general approach, working with the knowledge representations in a learning area is fruitful because (a) learning to work with the representations is essential to developing competence in the domain, (b) it is natural to present information to examinees and to require creating or completing a domain representation as a work product, and (c) in computer-based assessments the work product can be structured so it seems very open-ended to the examinee yet it is straightforward to characterize its key features (Scalise and Gifford 2006).

A wide variety of machine learning algorithms have been applied to the task of assigning scores to complex work products. These range in complexity from linear regression (Attali and Burstein 2006) to neural networks (Stevens and Thadani 2007). Despite the differences in the structural form, the basic principle is similar: the parameters of the algorithm are learned from a sample of student responses.

ETS's e-rater® (Attali and Burstein 2006; Burstein et al. 2013) system for scoring essays provides a good example. First, several natural language processing tools are run on the student essay to determine values for multiple features (11 in version 13.1, which was used operationally in 2013). Typical features are counts of errors of various types (grammar, usage, mechanics, and style), measures of the richness of the vocabulary, measures of the organizational structure, and measures of how similar the vocabulary is to high-scoring essays. These features may be transformed before putting them into a regression equation; for example, the error counts are normalized by dividing by the document length, and then the square root is taken to reduce the skewness of the measures.

The features are then put into a regression equation to determine the score, with weights that best predict human scores from the feature values. E-rater supports two kinds of regression models: prompt-specific and generic. To build a prompt-specific model (here a prompt corresponds to a single task in the ECD framework), the regression weights are determined from several hundred human-scored essays. In the generic model, the regression weights are determined from a sample of several hundred, sometimes thousands, of human-scored essays taken from several prompts (different tasks from the same task model in ECD terms). The advantage of the generic model is that new weights are not needed for new tasks from the same task model; the disadvantage is that it cannot take advantage of several features, such as the task-specific vocabulary, and does not account for prompt effects in the ratings. Hybrid approaches include intercepts that account for average differences in prompt difficulty within a generic model.

This approach extends easily to the case where there are multiple observables. In this case, different features of the work product, with different weights, produce different observables. Deane and Quinlan (2010) shows some preliminary factor analysis results in which different e-rater features map onto different strands of writing skill as measured by human raters.

Note that the calculation of observables can occur in several steps. These steps can involve the creation of intermediate observable variables; for example, the feature variables in the e-rater. Some of these intermediate observables can be used for feedback. ETS's Criterion$^{\mathrm{SM}}$ system uses the output from the grammar checking system in e-rater to provide feedback to essay writers.

The evidence rules can also involve producing intermediate processed work products necessary for later calculations. Consider a short segment of audio captured as part of a speaking task. It might be helpful to put this through a filter designed to eliminate background noise before further processing, either human or computer. It also might be helpful to note the position and duration of pauses in the recording. Similarly, it may be useful to correct spelling in an essay before putting it through a vocabulary matching program.

Evidence rules can get quite complicated. Regardless of their form, it is important that the evidence rules provide clear instructions about how the values of the observable outcome variables depend on the observed work products. That means that the observables as fed into the statistical part of the evidence model are well defined.

"Well defined" does not mean there must be a single value. Neural networks, for example, can provide weights for each possible score value. Typically the highest is used as "the" score, but the entire vector of relative strengths could be entered into a Bayes net as virtual evidence (Sect. 5.2.3). While some performances will point strongly to one of the values, others with uneven mixes of features are harder to rate, and the more equivocal information they bear is properly reflected by a more spread out virtual evidence vector.

12.4.2 Statistical Models of Evidence (for Evidence Accumulation)

The rules of evidence span the distance between the work product and the observable variables. The statistical part of the evidence model spans the remaining distance between the observables and the proficiency model. This part of the evidence model presents the rules for how to update the proficiency variables given a particular pattern of observed outcomes from a task.

Evidence-centered assessment design is intended to be neutral to the mathematical form chosen for the proficiency and evidence models. However, the language was chosen to be natural when these are represented in a Bayesian framework. The proficiency model is then a set of proficiency profiles and a probability distribution over possible values that represents our current state of knowledge about the examinee's proficiency. The evidence model provides the conditional distribution for the observables given the proficiency variables. A given pattern of evidence then induces a likelihood over the proficiency variables, through the conditional probability distributions. The two are combined through Bayes' Theorem to give a posterior distribution over the proficiency variables.

Two important special cases are Bayesian networks and IRT, either unidimensional or multidimensional. In all cases the proficiency variables (sometimes called proficiency parameters) can initially be given a population distribution that serves as a prior, or if desired a diffuse prior. With Bayes nets, the proficiency variables are discrete, and the statistical part of the evidence model is represented with a Bayesian network fragment. With IRT, the proficiency variables are continuous and the evidence model is represented by item response functions. Once the posterior is obtained, it can be used directly or various reporting rules can be applied. Most of the statistics described in Sect. 12.2.3 for Bayes nets have versions that are appropriate to IRT and MIRT models.

The task of building the statistical part of the evidence model can be divided into two steps: specifying which variables are involved and how they are related to each other, and specifying the parameters of the relationships. When we are representing the statistical part with a Bayesian network, these steps become specifying the graphical structure and specifying the conditional probability tables.

Note that the statistical part of the evidence model involves three kinds of variables: proficiency variables, observable outcome variables, and other variables local to the evidence model (in particular, unobserved variables introduced to model local dependence among the observables, like the context variables introduced in Sect. 6.2). The proficiency variables appearing in the evidence model are references to the corresponding variable proficiency model;

their definitions are identical to those in the proficiency model. In the case of a Bayesian model, their marginal distribution is provided by the proficiency model.

An evidence model references a subset of the proficiency variables (possibly all of them, as necessarily happens when there is only one proficiency variable in the proficiency model). The referenced proficiency variables constitute the *boundary* of the proficiency model/evidence model relationship. In the case of Bayesian network model, the updating algorithms require that the boundary variables appear within a clique in the proficiency model (Sect. 5.4.1). Therefore, there are significant computational implications for the number of proficiency variables that appear as boundary variables. The necessary moralization of boundary variables imposes additional computational burden as the number of these boundary variables increases.

There are two key local independence assumptions associated with the boundary variables. The first is that the observable outcome variables are independent of the other proficiency variables given the boundary variables. In other words, evidence from a given task is always propagated through the boundary variables of its evidence models; the evidence it provides for any other proficiency variable is always indirect. The second is that the observables from two different evidence models are independent given the boundary variables from either of the evidence models. In other words, once the evidence from the observables has been absorbed into the proficiency model, the observables can be discarded and not consulted again. Although these assumptions are stated in terms of Bayesian network models, most other representations use some variation of these assumptions.

The local independence assumptions for Bayesian network models are weaker than those used for typical IRT models. In particular, when a task has multiple observables, the Bayesian network can constructed so as to model the dependence among the observables Almond et al. (2006b), while the most widely used IRT models assume that the observables are locally independent. An additional context or testlet variable can be introduced in either Bayes nets (Sect. 6.2) or IRT (Wainer et al. 2007) to model local dependence between items from a set (i.e., observables from the same task), although other models are also possible (Almond et al. 2006b; Wilson and Adams 1995).

There are two useful notational forms to describe the relationship among the variables in the statistical part of the evidence model. There is the graphical notation that has been used extensively throughout this book (in particular Chap. 4). A widely used alternative is the the Q-matrix, a matrix in which the columns correspond to proficiency variables, the rows correspond to observables, and the entries are positive when the given proficiency variable has a direct influence on the corresponding observable (Sect. 5.6). The graphical notation has more flexibility when there is dependency among the observable variables that must be modeled. However, the Q-matrix notation is more compact in the common case where each task yields a single observable (requiring only one row per evidence model).

Example 12.10 (Evidence Model for Lecture Clarification Task). *The design team wants to build an evidence model for the Lecture Clarification Task (Example 12.8) for use with the four modal skills proficiency model (Fig. 12.4). After some discussion, the design team identifies three observable outcomes from the task:*

- *Pronunciation: Was adequate pronunciation used in the work product?*
- *CorrectForm: Was the work product in the form of a question or statement as called for in the task directive?*
- *OnTopic: Was the work product relevant to the point at issue?*

Assume for the moment that the design team is able to build appropriate rules of evidence for either human or automatic scoring of these three variables.

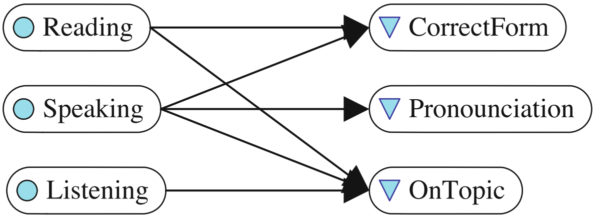


Fig. 12.7 Evidence model for lecture clarification task for use with modal proficiency model

Reprinted from Almond et al. (2006a) with permission from ETS.

Figure 12.7 gives a graphical structure for this evidence model. OnTopic depends on both Listening (to identify and understand the ambiguity in the lecture), Speaking (to make the clarification request understood to the listener), and Reading (to understand the instructions). CorrectForm depends only on Speaking (to be able to utter statements and questions) and Reading (to understand the written directive for the task). Pronunciation depends only on Speaking; the evidence rules call for off-topic but intelligible speech to be given a high value.

Note that the proficiency variables (marked with circles) do not have parents in this graph. Their distribution is given in the proficiency model, and must not be repeated here (or it would be counted twice). Consequently, Fig. 12.7 is a Bayes net fragment and not a complete network. The observable variables (and any other local evidence model variables) marked with triangles are unique to this model and must be defined here.

The graphical structure chosen here is specific to both the task model in one direction and the proficiency model in the other direction. If we were using the alternative "communicative competence" proficiency model (Fig. 12.5), then different proficiency variables would be available, and the design team should consider a different graphical structure (and maybe even different observables). This an important reason for separating the evidence models and task models in ECD. The evidence model adapts the evidence from the task for use with a given proficiency model, adjusting the grain size and focus to be appropriate for the purposes supported by the proficiency model.

After the design team builds the structural part of the evidence model, they choose a parameterization for the conditional probability relationships that must be defined. At this stage in the design, they usually stop short of assigning values to the parameters. This is because those values may vary from task to task. Just as tasks are instances of task models, there are instances of the evidence model call *links*. A link is an evidence model with its parameter values adjusted to a particular task. Chapter 13 describes links and their relationships to tasks in more detail. Because examinees can respond to tasks in unexpected ways, it is difficult to assign final values to the link parameters without data from a pretest of the tasks.

Although exact values of the *weights of evidence*—the values of the link parameters—are not assigned at the CAF development stage, a method must be chosen to assign the parameter values. One simple method, appropriate for low-stakes testing situations, is to simply assign the same values to the link parameters for all of the links for tasks coming from the same task model. This is similar in spirit to a teacher assigning point values to the items in a quiz, and its psychometric properties are no worse. In this case, the link parameters are assigned at the evidence-model level, and this is done at this stage of the assessment design process (although it could be reviewed and revised after seeing pretest data).

Part II discussed another method for assigning the link parameters: Bayesian inference. Bayesian inference requires both pretest data and prior distributions for all of the link parameters. These priors are usually defined at the level of the evidence model, and hence become part of the evidence model construction process.

As noted, there can be patterns of local dependence among the observables. If there are multiple observables they may be dependent even given the proficiency variable because they come from the same task, due for example to familiarity with the topic or a misunderstanding of the task. Another common pattern is sequential dependence, where the observables represent steps in a multistep task. Finally, there may be functional dependence among the observables because the come from the same work product. For example, a null or unintelligible response would produce low values for all three observables on the Lecture Clarification Task (Example 12.10).

There are a number of possible approaches to modeling the local dependence of the observables in the model:

- Ignore it and hope the approximation error from ignoring it not strong. Often the task model and rules of evidence can be designed to minimize the dependence among observables.

- Combine the observables into a superobservable by adding another rule of evidence to combine the dependent observable (Wainer and Kiely 1987). For example, if the task consisted of a reading passage followed by a number of questions asking about the passage, the superobservable might be the sum of the individual question scores.
- Add a local *TopicFamiliarity* or *Context* variable to soak up the dependence (Wainer et al. 2007, Almond et al. 2006b).
- Order the observables, and then make each observable a parent of the next one in the series (Almond, Mulder, et al. 2006b call this the *cascading* pattern).
- Build a custom evidence model that captures the detail.

Although the different models suggest different mechanisms producing the local dependence, the goal of the modeling is the same in all cases: to produce a likelihood for the proficiency variable(s) given a pattern of evidence indicated by the observed variables. If the model parameters are learned from pretest data, it is likely that all of the models will produce roughly the same likelihood for a given pattern of evidence (Almond et al. 2006b). However, adding and removing observables from the evidence model can cause the learned parameter values to be no longer appropriate, as dependence may increase or decrease. However, as long as the evidence model is treated as a unit, the differences between calibrated evidence models of various types is minimal.

When the evidence model only taps a single proficiency variable, the model that combines the all of the observables into a single superobservable has a distinct advantage: it has fewer parameters than some of the other models. That makes it easier to estimate with smaller data sets and gets away from possible issues with identifiability or collinearity in the more complex models. This can be seen in the preference for holistic scoring over analytic scoring when human raters score essays (Breland et al. 1987). The analytic scores add little additional information when the goal is to assess overall writing performance.

Multiple observables become more interesting when there are also multiple proficiency variables. Different observables from the same task may tap different combinations of proficiencies. When a subset of observables address the same combination of proficiency variables, the designer might consider using an additional rule of evidence to combine the similar observables rather than modeling the local dependence.

Often feedback observables are chosen from intermediate observables that a calculated as part of the rules for calculating the overall observables. The student can be given values task by task with these feedback observables, not as estimates of proficiency but as descriptions and evaluations of particular performances. For example ETS's Criterion$^{\text{SM}}$ can provide both feedback on grammar, usage and style issues in the essay and an overall score using the e-rater engine (Attali and Burstein 2006). The first step in the processing is to put the essay through a grammar checker that identifies issues of grammar, usage, mechanics and style. These are sent to Criterion's feedback mechanism to provide low-level feedback. They are also used as part of the overall task scoring, which is then accumulated over tasks as evidence for a proficiency that spans tasks. Error rates for the four kinds of errors are among the features used in computing the final e-rater score.

Observables can also be used for research purposes, to inform the test designers about some aspect of how students are approaching the task. As an example of these research observables, consider a computerized test that records both the answer provided by the examinee and the response time the examinee required to provide the response. While the response itself is used as evidence of the ability of interest, the response time is typically not used as evidence (there are many alternative explanations for why an examinee may have a longer or shorter response time than expected). The response time is is used for other decision making about the test. For example, tracking response times for test tasks provides an empirical basis for predicting the required testing time for a new assessment, and allows the test designers to adjust the assembly model so that the assessment will fit within an allotted time slot without adding a speededness component to the assessment.

12.5 The Assembly Model

Most of this book talks about assessments when in many cases what is important is not an individual assessment, but an assessment program: a series of assessments that are all meant to be comparable in some sense. An assessment program defines a series of forms, each a collection of tasks that are all administered on the same occasion. There are many reasons why an assessment program may require multiple forms. Two of the most common are that the assessment will be given to the same students on multiple occasions (as part of a longitudinal study) and test security (i.e., so that examinees who discuss the contents of the test with previous examinees will not have an unfair advantage).

In designing an assessment program, we would like the forms to each provide similar evidence for the proficiencies. The question of how many tasks of what type are required to make a valid form of the assessment immediately arises. It is the role of the assembly model to answer this question.

Consider one of the most complex cases that the assembly model must cover, the case of a *computer adaptive test* (CAT; Wainer et al. 2000). CAT does not use fixed forms; rather, the computer assembles the form as the examinee interacts with the assessment. This means each form is potentially unique. The CAT algorithm tries optimize information about a student, subject to various constraints about what kinds of tasks must be in the form, and what kinds of tasks cannot appear together.

Typical CAT construction proceeds in two stages. The process starts with the universe of all tasks which have been authored, reviewed, pilot-tested, and judged suitable for the current assessment. This is sometimes called the *vat*. The first stage selects a *pool* of tasks from the vat for deployment in an operational version of the test. Typically a pool will be in the field for a period of time (from a week to a year) and then will be replaced with a new pool. This allows new tasks to enter the field and old ones to be retired. The second stage happens after the pool is in the field and an examinee sits for the assessment. At this point the algorithm selects tasks for that examinee to attempt. The selection happens after each previous task has been scored (this assumes automatic scoring) so the CAT algorithm can take current estimates of the examinee's proficiency into account while selecting tasks.

In a CAT, the activity selection process chooses tasks to optimize some kind of information criteria, called the *target rule*. In IRT-CAT, this could be, for example, the Fisher information the assessment provides at the current best estimate of the examinee's proficiency, or expected minimum variance for the posterior. Almost all of the quasiutilities discussed in Chap. 7 are fodder for designing target rules. Adaptive Content with Evidence-based Diagnosis (ACED) (Shute et al. 2008) used expected weight of evidence.

The activity selection process is not free to pick any task in the pool to meet the target; the process is subject to a number of *constraints*. There are generally two kinds of constraints that are put on the form: minimums (and maximums) for certain task types, and overlap constraints about two tasks that are too similar appearing on the same form. Sometimes the constraints are written in terms of task models, but usually the constraints can be implemented by operating on task model variables (Sect. 7.4).

For high-stakes assessments, where test security is a concern, there are usually constraints on how often items appear on tests across examinees. Test security is an important issue in many testing programs; Veldkamp et al. (2010) provide a concise overview of the item exposure challenge and main approaches for tackling it. Wang et al. (2011) apply the ideas to cognitive diagnosis modeling, which transfers readily to Bayes nets proficiency models.

Minimum (and maximum) constraints are usually concerned with the breadth and depth of the evidence provided for each proficiency variable. This also influences the meaning of the proficiency variables. Consider the variable *Reading* in the language assessment. Claims about *Reading* are usually defined in terms of the genre and complexity of the text to be read. In order to provide evidence of for those claims, the tasks must be chosen to span the genres and difficulties that are targeted by the assessment. If a test form randomly selects only a single genre of content, then the effective meaning of *Reading* would be different from the intended meaning.

Example 12.11 (Language Placement Test, Continued: Register Constraint).

In building the proficiency model for the Language Placement Test, the design team identified two variables, Register and Purpose, for which they decided not to build proficiency model variables. One of the claims associated *with the Speaking variable is that students can speak in a register appropriate to the situation; for example, they can distinguish between language that is appropriate to use with their instructor and language that is appropriate to use with their peers. In order to validate that claim, a constraint is added to the assembly model a Register variable is added to speaking tasks with two possible values:* `inferior-to-superior` *and* `peer-to-peer`*. A constraint is added to the assemble model that at least one speaking task of each type must be included on every form of the assessment. The Purpose variable similarly requires a constraint to enforce the distribution over several values.*

An overlap constraint becomes necessary when two tasks are too close in content to appear in the same form. If solving one task would provide hints for another, then local independence assumption would be violated; that is, the observables from the two tasks would be dependent even conditioned on the proficiency variables. In this case, some kind of exclusion rule is necessary. For example, the reading comprehension tasks could include a task model variable describing the topic of the reading material. An exclusion rule would guarantee that not too many reading texts about the same topic appear on the same form.

When the CAT is implemented, an activity selection process will use the information targets and constraints to build an assessment according to the following algorithm. At each stage of the testing (when it is necessary to select a new task) the activity selection process consults the student-specific copy proficiency model called the student record) to get the current state of knowledge about the student, then selects a task that maximizes the current information target subject to the constraints. The last step that is required is a *stopping rule*, a criterion for when to stop. It may be based on testing time, number of tasks administered, obtaining sufficient information about the proficiency variables, or some combination of these.

An alternative to CAT is a *linear form* in which all examinees see the same items in the same sequence (often several linear forms are randomly assigned or spiraled to the examinees). The linear form can either be computer administered or presented in a paper-and-pencil format. In a linear assessment the pool stage of assessment construction is skipped and the activity selection process (which runs in advance of the assessment now) goes straight from the vat to the form. The same kinds of constraints are relevant to both the linear and CAT assessments; however, the information targets are now slightly different. The goal in a linear form is to optimize the information for a population of test takers. Linear forms usually provide about as much information as a CAT for examinees in the middle of the proficiency distribution, but the CAT provides more information for examinees in the tails of the distribution. A big advantage of linear tests over CAT is that the forms are created in advance. Problems with automatic test construction algorithms can be detected and fixed ahead of time.

Multistage testing offers a possibility midway between linear testing and task-by-task adaptivity (Lord 1980; Yan et al. 2014). In a multistage test, a number of short linear forms, called stages, are constructed. At the end of each stage, the form to use in the next stage is selected based on the current proficiency estimate for the student. Because there is usually a small number of forms for the stages, they can be hand-checked, so any strange interaction of the constraints and information criteria can be fixed in advance. However, there is usually sufficient information gain from just a few stages that the amount of testing can be reduced.

When the assessment will report about more than one variable, the activity selection process needs to balance the information about all of the reporting variables. The algorithms described in Sect. 7.4 are helpful in this case.

One strategy is to pick a primary target (for example, *Communicative Competence* in Fig. 12.4). As the other proficiencies are linked to this proficiency, the selection algorithm should choose among tasks designed to address the this node.

Madigan and Almond (1995) predict that the primary proficiency approach will have an unusual and unwanted behavior. In the Language Proficiency example, an assessment built in this way is apt to switch rapidly among the four modes (e.g., first a Reading, then a Listening, then a Speaking, then a Writing task, and then back to Reading again; see Sect. 7.4.2 for more discussion). Madigan and Almond (1995) suggest a strategy called critiquing (Barr and Feigenbaum 1982). In terms of the assembly model, this means that there are multiple targets and some intermediate stopping rules for switching between them. Another way to think about this that the multiple targets represent different sections of the assessment, each of which has its own proficiency target and stopping rule.

Example 12.12 (Language Placement Test, Continued: Target and Stopping Rules). *For the language placement assessment (Example 12.3) the design committee decides that they need good information about all four modal variables: Reading, Writing, Speaking, and Listening. They also decide to use expected weight of evidence as their information metric. In this case, targets consist of a binary hypothesis, that is, both a target variable and a target level. The goal is to first identify students who are weak in one or more of the proficiencies to place them into remedial classes, and then to identify students who are strong in certain proficiencies so they can be placed outside the language support program. Consequently, they decide to try to gather evidence of low values before evidence of high values. The assembly model has the following target and stopping rules:*

1. *Choose a task, t, to maximize $EWOE(\mathit{Reading} > \texttt{Low} : E_t)$. Continue doing this until either $\mathrm{P}(\mathit{Reading} = \texttt{Low}) > 0.9$ or $\mathrm{P}(\mathit{Reading} > \texttt{Low}) > 0.66$.*

2. *Choose a task, t, to maximize $EWOE(Listening > \texttt{Low} : E_t)$. Continue doing this until either* $\mathrm{P}(Listening = \texttt{Low}) > 0.9$ *or* $\mathrm{P}(Listening > \texttt{Low}) > 0.66$.
3. *Do the same thing with Speaking.*
4. *Do the same thing with Writing.*
5. *If* $\mathrm{P}(Reading > \texttt{Low}) > 0.66$, *then choose a task, t, to maximize $EWOE(Reading \geq \texttt{High} : E_t)$. Continue doing this until either* $\mathrm{P}(Reading \geq \texttt{High}) > 0.9$ *or* $\mathrm{P}(Reading < \texttt{High}) > 0.9$.
6. *If* $\mathrm{P}(Listening > \texttt{Low}) > 0.66$, *do the same thing for Listening.*
7. *If* $\mathrm{P}(Speaking > \texttt{Low}) > 0.66$, *do the same thing for Speaking.*
8. *If* $\mathrm{P}(Writing > \texttt{Low}) > 0.66$, *do the same thing for Writing.*

To complete the assembly model, the design team needs to add task-variable minimum constraints to ensure a selection of genres, purposes, and registers. The implementation team might also need to add overlap constraints to avoid multiple tasks on the same topic within a given form.

The kind of stopping rule used in this example, looking for thresholds for proficiency variables above and below certain cut scores, is appropriate when the assessment will be used for classification decisions (e.g., the placement decision). An alternative stopping rule is to try to bring the standard error of measurement below a threshold. This is appropriate when the purpose of the assessment is to order the students or when the cut score is not known in advance.

The stopping rule is just one way of expressing the fundamental question of the assembly model (or for that matter the entire CAF): Does the assessment provide enough evidence to support the claims we want to make based on the information it provides? Whether the assessment is adaptive or linear, the assessment design needs to be checked to ensure that it provide adequate evidence. This includes both a theoretical check towards the end of the deign phase (a construct validity argument, see Kane 2006) and an empirical check when the completed assessment is fielded (criterion, concurrent and predictive validity checks). Naturally, the purposes for which the assessment will be used and the stakes (the consequences to participants for incorrect decisions) will influence the amount of evidence that is needed.

Once the evidence models and assembly model are specified, there is enough information to construct a Q-matrix for the assessment. In particular, the evidence models correspond to rows of the Q-matrix and the assembly model describes how many times each row type is repeated. A simple check at this stage is to simply scan down the rows of the evidence model to see that each proficiency is represented a reasonable number of times (Almond 2010a). More sophisticated analyses can look for other patterns of evidence that may be problematic (such as blocking, see Gierl et al. 2007). These checks can help the design team uncover the need for new kinds of tasks to gather evidence not provided by the current assessment.

12.6 The Presentation Model

Although the number of assessments delivered by computer increased rapidly during the end of the twentieth century, paper-and-pencil administration is still frequently used in the beginning of the twenty-first century. Computer delivery offers advantages (e.g., interactivity and automatic capture of responses) but can be challenging in remote locations, and it can be difficult to obtain enough computer workstations to accommodate a high-volume test. Many assessments must be designed for both computer and paper-and-pencil delivery. New delivery methods are constantly being explored. Small devices such as smartphones and tablet computers are being explored as modes of delivery. More assistive technologies are becoming available for persons with disabilities.

One of the reasons for formal assessment design methodologies like ECD is to promote reuse of assessment elements. In particular, we would like to use the same task models with multiple presentation platforms. The role of the *presentation model* is to adapt tasks for particular platforms. The task model defines the elements, the presentation material that makes up the task, the interactions the platform must support, and the work products that must be collected. The presentation model describes how they are rendered in the delivery platform.

The presentation model is a style sheet for the task model, describing how the elements of the task are arranged. For example, consider the piece of presentation material providing the examinee with instructions for how to complete the task.[4] The presentation model for computer delivery might call for this to always be displayed in a constant location, for example, a sidebar on the screen. The presentation model for paper delivery might call for tasks with common instructions to be presented together and the instructions to be printed once at the top of the page on which the tasks appear. If the assessment is for examinees with limited reading ability (say, young children), the presentation model may also call for text-to-speech capability.

A question that can be important is whether alternate modes of presentation affect the evidentiary properties of a task. Bridgeman et al. (2001), in one of the few formal studies to investigate this question, looked at the issue of screen size, resolution, and display rate on test performance. The study found little difference for a math test, where most items fit on a single screen even at smaller screen sizes. It did find a difference for a verbal test that included passage-based reading comprehension tasks. In particular, tasks were more difficult if an examinee had to scroll to see the entire text.

Whether or not such differences matter depends on the purpose of the assessment. In a low-stakes testing situation, the convenience of being able to use whatever equipment is at hand may outweigh the construct-irrelevant variance caused by students using different presentation platforms. Accurate

[4] In British usage, this would be called the *rubric* for the task.

comparisons among students at different sites are not required. In high-stakes testing situations, however, it may be important to control the presentation platform tightly so as to not give examinees with access to better equipment an unfair advantage. In either case, the presentation model must be clear about what range of presentation platforms are appropriate.

Although assistive technologies have expanded the population of examinees to provide access to more individuals, they also bring with them questions about when a given accommodation is appropriate, or even possible. Consider creating a presentation model for a read-aloud protocol that reads the item text using a synthesized human voice. This works fairly well for a purely text-based item, although the time limits may need to be adjusted as listening often takes longer than reading silently. Additional thought is needed at the level of the presentation model if the physical layout of the text is important to the meaning (for example, in a table or an equation). The presentation model needs to understand how to convey the physical layout through text. A task that depends substantially on a visual stimulus (say a task that involves reading a map or describing a piece of artwork) may not be compatible at all with the read-aloud presentation model. In this case, the accommodation needs to be made at the level of the assembly model. The task types that rely on visual stimulus need to be replaced with tasks that do not.

Even that might not be sufficient if the task type was critical to obtaining evidence to support one or more claims. Consider the use of the read-aloud accommodation on a reading test. Part of the reading construct is *decoding*, mapping the letters on the printed page to sounds. The read-aloud presentation has the computer do the decoding for the examinee. For young elementary students, decoding is an important component of the reading construct. Thus, the read-aloud accommodation would remove evidence about an important component of the construct and the test with read-aloud would be substantially different. Most college students have no difficulty with the decoding skills. Consequently, the read-aloud accommodation presents little problems when assessing this population.

Accommodations for special needs is a complex issue, and this discussion has only scratched the surface. Hansen and Mislevy (2004) and Mislevy et al. (2013) use Toulmin diagrams to map out how the accommodation changes the evidentiary argument, and how properly chosen accommodations can provide more valid arguments for more diverse populations of students. The ECD framework allows us to frame a complex question about accommodations as a question about evidence. This framing allows the test designers to weigh the consequences of adapting tasks for new kinds of presentation. It also enables them to design systems that can present comparable tasks in different ways to different students based on their profiles of knowledge and skill as are needed for access but irrelevant to the targeted proficiencies.

When the assessment includes simulations or when tasks are embedded in games, then the capabilities of the simulator or game engine are part of the presentation model. Consider the design of the physics game *Newton's Playground* (Shute et al. 2013). Both *Newton's Playground* and the commercial game *Crayon Physics Deluxe* that inspired it are built off the same Box2D physics engine; however, they differ in the way they use it. In both games, players make a ball by drawing a circle on the screen, but the two games differ in how they calculate the mass of the ball. In *Crayon Physics* the mass is proportional to the area of the ball on the screen, while in *Newton's Playground*, the mass of the ball is proportional to the length of the line used to draw it. Thus players can draw heavy objects by scribbling inside of them. Thus, the *Newton's Playground* presentation model has a capability not present in the *Crayon Physics* model: it can create tasks where students need to distinguish between mass and volume, a facet of physics understanding that Kennedy and Wilson (2006) identified as a key state in a physics-learning progression.

12.7 The Delivery Model

Most of the important decisions made in the course of designing an assessment are recorded in the proficiency, evidence, task, assembly, and presentation models. However, there are a few important decisions that do not seem to fit anywhere else. For example, "Who can see the score reports and for how long are they are available?" "What kind of identification must candidates show before sitting for the assessment?" "How long will the assessment take?" The role of the *delivery model*[5] is to capture these extra questions that do not seem to fit anywhere else.

Although these questions may seem unrelated to the evidentiary argument, they actually set the stage for a number of hidden assumptions within the evidentiary argument. For example, we assume that the person whose name appears on the score report was also the person who sat for the exam. If this is not true, the observations we make in the course of the assessment provide little evidence on way or another about the true proficiencies of the candidate. Similarly, the rules for how score reports delivered are important because because the score report could be altered or changed (akin to "chain of custody" issues in jurisprudence). Thus, these additional constraints help eliminate potential threats to the evidentiary argument.

The question of assessment length is one that arises frequently. Ideally, this would be the province of the assembly model; that is, the assessment would be over when enough evidence was gathered for the purposes of the assessment. In practice, the assessment always takes time away from other activities (unless the assessment can be embedded within those activities). Thus, in almost all assessment designs the length is set by external considerations. For example, the assessment will be 45 min long because that is the length of one class period, or the assessment will be no longer than 2 hours long because too few schools would adopt it if it were longer.

[5] The delivery model appears as the surrounding box in Fig. 12.1.

This has implications that bear on the other models. The time limit immediately impacts the assembly model, and how much evidence can be gathered. This also impacts the proficiency model. The limited amount of evidence almost certainly means that only a few proficiencies can be reliability measured. This, in turn, argues for a smaller proficiency model. This is a tension that is seen in many assessment designs: While the subject matter experts would like a large proficiency model that spans the entire domain, the time constraint means that only a small portion of that domain can be measured in a typical large-scale assessment. Evidence-centered assessment design does not resolve any of these dilemmas, but it does help the design team recognize them early so that they can work out resolutions appropriate to the purpose of the assessment being designed.

12.8 Putting It All Together

When writing about ECD, describing the models in a linear fashion is almost unavoidable. There is a temptation to then conclude that building a CAF is a linear process, in which the design team first develops the proficiency model, then the evidence models and task models, then the assembly models, and finally the presentation and delivery models. In practice the assessment design process is seldom so linear. The design team will need to work back and forth between all parts of the model, frequently going back an revisiting previous work to make sure it works well with the new pieces, and to take into consideration newly discovered constraints.

As part of this process, there is a need to evaluate the success of the design at each stage of development and work with the team to revise the design as needed. ECD presents a clear criterion: the assessment must provide sufficient evidence to support its *purpose*, as expressed in its claims. The ECD process lays out an explicit *chain of reasoning* connecting task to observable evidence to proficiency variables to claims. It is easy to get caught up in technical details of the assessment design process such as creating tasks or fitting measurement models, and lose sight of the purpose. It is worth frequently returning to initial statements claims and evidence to ask whether the current design is appropriate to the purpose.

Another question that often arises is how much detail is needed in a completed CAF. The answer is enough so that all of the development work needed to implement and operate the assessment (see the next chapter) is clear to the teams that need to do that work. If the assessment is similar to other previous assessments, many details can be left vaguely specified as "same as last time." The ECD process is most valuable when item development is costly and/or complex, pretesting samples are problematic or expensive to obtain, novel uses for an assessment are being considered, a new construct is being addressed, or a strong and explicit argument for the construct and content validity of the assessment is needed at the outset.

ECD is a means of formalizing and monitoring the process of good design for educational assessments. It shares with any formal design process the goal of coordinating the work of the various teams building the assessment. When any part of the design changes, that change will have impact on many different teams. The formal design allows the team to assess the impact of the change before making it, allowing the design team to weigh the costs and benefits of any changes and notify the people whose work will be impacted. This ultimately makes the assessment stronger and easier to produce.

ECD differs from other formal design and development processes in its emphasis on the chain of reasoning between claims and evidence, both as a tool for effective design and as an argument for the construct validity of an assessment (Kane 2006). The design and development process forges this initial chain of reasoning, incorporating the evidential arguments between design components and conclusions, and the scoring process takes advantage of this structure to ground appropriate inferences about participants based on their interaction with the assessment. The next chapter explores the processes of implementing and operating the assessment.

Exercises

12.1 (Competing Tasks). Consider the claim, "The student has sufficient writing skills to complete assigned term papers." Now consider two possible observations related to that claim. (1) The student performs well on an admission essay, and (2) The student performs poorly on a timed essay in a placement test. Draw a Toulmin diagram for these two arguments. How would you reconcile the contradictory evidence?

12.2 (Memorization of Tasks). A certain high-stakes testing program uses a computer adaptive test in which the same pool is in the field for about a month. The design team for this program discovers a web site on which students discuss tasks from the test and their solutions. Build a Toulmin diagram that contains the possibility that students have memorized the question and the answer.

12.3 (Score Ordering). Consider the score report in Fig 12.6. Does the order (left-to-right) of the bars in that figure matter? What is the best order and why? If an additional bar is added for overall communicative competence, where should it be placed?

12.4 (Statistic Choices). Consider the score report design from Example 12.7. The design committee is considering three different statistics for *Reading*:

- *Proficiency Level.* One of `low`, `medium` or `high` depending on whether $\mathrm{P}(\mathit{Reading} = \texttt{low})$, $\mathrm{P}(\mathit{Reading} = \texttt{medium})$ or $\mathrm{P}(\mathit{Reading} = \texttt{high})$ is larger.

- *Probability of Mastery.* $100\mathrm{P}(\mathit{Reading} = \texttt{high})$ (recall that the `high` state is considered strong enough ability to place out of the remedial course).
- *EAP Score.* $100\mathrm{P}(\mathit{Reading} = \texttt{medium}) + 200\mathrm{P}(\mathit{Reading} = \texttt{high})$

Assume that University C has a limited number of spaces in the remedial classes and wants to give preference to the student who need help the most. What is the best statistic choice for the placement decision? What is the best statistic choice for providing information to the instructor about the students enrolled in the class? Explain your answers.

12.5 (Reading Comprehension Task Model). One of the task types for the language placement test is based on the classic reading comprehension task. In this task, the student is presented with a piece of text to read and then asked a series of questions that are meant to provide evidence that the student understood what was read.

Build a task paradigm (a sketch of a task model) for this task type. Your task paradigm should include the following features:

- A description of the presentation material.
- A description of the expected work products.
- A list of the most important task model variables and the roles that they play.

12.6 (Reading Comprehension Evidence Model). Build an evidence paradigm (a sketch of an evidence model) for the reading comprehension task described in Exercise 12.5. This evidence paradigm should answer the following questions:

- What are the observables?
- How do the observables relate to the work product (evidence rules)?
- What kind of data in addition to the task is required for the evidence rules?
- How do the observables relate to the proficiency variables (in either of the two proficiency models under consideration)?
- How should local dependence among the observables be modeled?
- What kind of task-level feedback should be available, and are additional observables necessary to support it?

12.7 (Read-Script Task Model). Another task under consideration for the language placement assessment is a script reading task, in which the student is expected to read text that appears on the computer screen into a microphone.

Build a task paradigm (a sketch of a task model) for this task type. Your task paradigm should include the following features:

- A description of the presentation material.
- A description of the expected work products.
- A list of the most important task model variables and the roles that they play.

12.8 (Read-Script Evidence Model). Build an evidence paradigm (a sketch of an evidence model) for the script reading task described in Exercise 12.7. This evidence paradigm should answer the following questions:

- What are the observables?
- How do the observables relate to the work product (evidence rules)?
- What kind of data in addition to the task is required for the evidence rules?
- How do the observables relate to the proficiency variables (in either of the two proficiency models under consideration)?
- How should local dependence among the observables be modeled?
- What kind of task-level feedback should be available, and are additional observables necessary to support it?

12.9 (Read Script Accommodation). One of the foreign students admitted to University C has severely limited vision, and normally accesses online text through the use of a read-aloud program. This student requests a read-aloud accommodation on the Language Placement assessment. Will this accommodation affect the evidence from the script reading task (Exercises 12.7 and 12.8)? Should the student be provided with adapted version of this task, a substitute task, or an assessment that does not include this task (and an advisory note that this part of the construct was not tested)?

12.10 (Reliability of Overall Score). Use the proficiency model in Fig. 12.4, and an assembly model that consists of four sections:

1. A *Reading* section that consists of 30 discrete items each of which has a single binary observable. Each one taps only the *Reading* variable.
2. A *Listening* section that consists of 30 discrete items each of which has a single binary observable. Each one taps only the *Listening* variable.
3. A *Speaking* section that consists of six partial credit items, each of which has a single observable that ranges from 0–4. Each one taps only the *Speaking* proficiency. (There is almost certainly some dependence on *Reading* or *Listening* as well. Assume for the moment that almost all examinees have the necessary prerequisite skills, so we do not need to model this dependence. See the next exercise for more details.)
4. A *Writing* section that consists of four partial credit items, two of which have a single observable that ranges from 0–6 and two of which have a single observable that ranges from 0–4. Again, assume that each one taps only the *Writing* proficiency.

Perform a simulation study to look at the reliability of this assessment design. Use hyper-Dirichlet distributions for the and pick reasonable values for the parameters of the proficiency model (they should be correlated and the marginal distributions should put be close to $\mathrm{P}(X = \mathtt{low}) = .25$, $\mathrm{P}(X = \mathtt{medium}) = .5$, and $\mathrm{P}(X = \mathtt{high}) = .25$). Use the DiBello–Samejima model for the evidence models and pick reasonable values for the parameters (the average difficulty should be zero and the average discrimination should be one).

Now, perform a simulation using the following steps:

1. Randomly select “true” values for the proficiency variables for 1000 simulees.
2. Randomly generate observables for Form A of the assessment for each of the simulees (there should be a total of 70 observables, 60 binary and 10 partial credit).
3. Calculate the MAP and EAP scores for Form A (assigning a value of 1 to `medium` and 2 to `high`) for each of the four modal proficiency variables and the overall *Communication* variable.
4. Random generate another 70 observable for Form B of the assessment for each of the simulees (We are pretending that we can generate an exactly parallel form).
5. Calculate the MAP and EAP scores for Form B using only the second set of data.
6. Look at the reliability by comparing the similarity of the scores for Form A and Form B. Calculate the correlation coefficient for the EAP scores, and Cohen’s Kappa for the MAP scores.

Is the reliability acceptable for the overall *Communication* variable? For the four modal variables? How does the reliability of *Reading* and *Listening* compare to the reliability of *Writing* and *Speaking*? What might this imply about the validity of the assessment with this design?

12.11 (Reliability and Integrated Tasks). Start with the setup for the previous problem and change the evidence models for the *Speaking* and *Writing* tasks. Assume that they are now integrated tasks that require either *Reading* or *Listening* in addition the the *Speaking* or *Writing* variable which is the primary target. Look at two variants of the evidence model. In Variant 1, use the compensatory DiBello–Samejima model, and chose sensible values for the parameters; assume that the discrimination for the primary skill, *Speaking* or *Writing* is higher than for the secondary skill, *Reading* or *Listening.* In Variant 2, use the inhibitor model and assume that a `medium` level of the secondary skill is necessary to solve the task.

Repeat the simulation study of the previous exercise. What effects do the integrated tasks have on the reliability?

13

The Evidence Accumulation Process

This chapter talks about how the conceptual assessment framework (CAF) described in the previous chapter is used to score an assessment.[1] Basically, the models of the CAF lay out the structures for data, materials, and messages needed for the activities that constitute an operating assessment system. A four-process architecture describes agents (people, computers, or some mix) that actually carry out the operations—from determining what to do, to interacting with examinees, to capturing and evaluating their work, to creating and reporting results.

Although we are interested in the case where the measurement model (that is, the proficiency model and all of the evidence models) are Bayes nets, much of the discussion in this chapter will apply to any type of measurement model. This is a universal protocol for scoring: a list of things that processes using the proficiency model and evidence models must do to score an assessment.

Before we can describe how to score an assessment, we must describe what an assessment looks like as it operates. Section 13.1 defines four processes that can be used to describe any assessment implementation. However, there are still a large number of steps between an assessment design (Chap. 12) and an assessment implementation. Section 13.2 describes some of the most important steps in building an assessment. Finally, Sect. 13.3 describes the *evidence accumulation process* or EAP[2], the process that is responsible for generating the statistics that will be displayed on the score report. An example from the ACED (Adaptive Content with Evidence-based Diagnosis; Example 7.5;

[1] With suitably broad definitions for both "score" and "assessment." By score we mean synthesizing evidence in students' performances for inferences about what they can know or can do more broadly. By assessment we mean a systematic process for doing this, which could be a familiar test, but could also be embedded in an interactive simulation or a game, and the user could be not only a teacher or employer but an instructional system or the students themselves.

[2] The abbreviation EAP is also commonly used for *expectation a posteriori*, or posterior mean, Bayesian point estimates. It should be clear from the context which one we are referring to.

R. G. Almond et al., *Bayesian Networks in Educational Assessment*,
Statistics for Social and Behavioral Sciences, DOI 10.1007/978-1-4939-2125-6_13

Shute et al. 2005; Shute et al. 2007; Shute et al. 2008) illustrates the interplay between a Bayes-net EAP and the activity selection process in linear and adaptive assessment strategies.

13.1 The Four-Process Architecture

The four-process architecture (Almond et al. 2002a; Almond et al. 2002b) provides a fundamental paradigm for conceptualizing, implementing, and communicating about any assessment. Regardless of the intent of assessment (diagnosis, certification, placement, etc.), the administration methodology (paper and pencil, computerized adaptive testing, complex simulations, etc.), or the scoring method (number-right, item response theory (IRT), neural networks, etc.), all assessments require the same four fundamental processes (at least in some trivial form) outlined in the four-process architecture. Establishing a common conceptualization and language for these assessment processes is an important step in the development of a more thorough, efficient, and complete means for researchers and test development professionals to understand, implement, and communicate about assessment.

Figure 13.1 depicts the basic four-process architecture, including the interactions between the processes and key elements of operational administration. We walk through the processes below, but Table 13.1 summarized the essential features. This table also indicates the CAF models that the processes rely upon, as specifying input, output, or information each needs to do its job. More detailed discussion of the interrelationships among the CAF models and the delivery processes appears in Almond et al. (2002a).

The four processes fundamental to every assessment are:

1. *Activity selection process.* The activity selection process selects assessment tasks for an examinee from the library of available tasks (the *task/evidence composite library*). It can operate either piecemeal or *en masse* as required by the assessment design (e.g., linear or adaptive testing). "Available tasks" include all aspects of the examination, including of course tasks that elicit information about measurement variables from which claims are made about participants, but also procedures to collect demographic data about them. In a more general system designed for learning, such as HYDRIVE (Sects. 1.2, 12.3, and 12.4), the collection of tasks could include some whose primary focus was instruction in addition to the assessment tasks. The activity selection process then relays information about the selected task(s) to the presentation process.
2. *Presentation process.* The presentation process is responsible for the presentation of the selected task(s) to the examinee. It obtains the information about the task from the database (the task/evidence composite library) and renders the selected task(s) in an appropriate form for the mode of administration. In an interactive task, it manages the interchanges

Table 13.1 Summary of the four processes

Process	Input	Output
Activity selection (ASP). Selects tasks, ends testing, initiates subtests; in learning systems, can select instructional or practice tasks	• Instructions from test administrator • Information from TECL about available tasks	• Requests to TECL for information about available tasks • Instructions to PP
Presentation (PP) Interacts with examinee: Presents stimuli, manages tool use; captures work products; in simulations, updates situation according to examinee actions	• Instructions from PP • Presentation material [TM] from TECL	• Requests to TECL for presentation material • Work products [TM] to EIP
Evidence identification (EIP; aka response processing, task-level scoring). Given work products and evidence rules, determines values of observable variables for both task-level feedback and for accumulating evidence over tasks	• Work products [TM] from PP • Evidence-rule data [EM-RE] from TECL	• Requests to TECL for evidence-rule data • Task-level feedback to examinee [EM-RE] • Values of observable variables [EM-RE] to EAP
Evidence accumulation (EAP; aka summary scoring, test-level scoring). Given values of observable variables, integrates evidence in the form of belief about student proficiency variables, via measurement model such as IRT or Bayes net	• Measurement-model fragments [EM-SM] from TECL • Parameters/weights [EM-SM] from TECL • Values of observable variables [EM-RE] from EIP	• Requests to TECL for measurement-model fragments [EM-SM]. • Requests to TECL for parameters/weights [EM-SM] • Examinee scoring record [PM] to ASP

Bracketed abbreviations in second and third columns indicate models of the conceptual assessment framework (CAF) where models, messages, or data are specified: *PM* proficiency model, *TM* task model, *EM-RE* evidence model, rules of evidence, *EM-SM* evidence model, statistical model, *TECL* task/evidence composite library

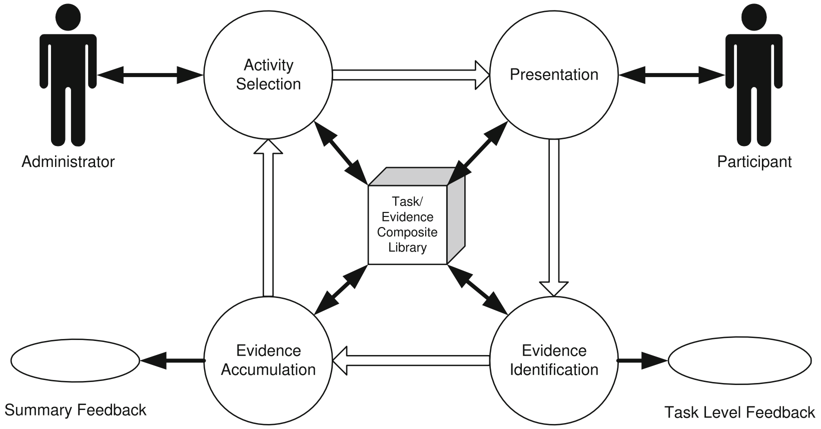


Fig. 13.1 The four-process architecture for assessment delivery
Reprinted from Mislevy et al. (2004) with permission from the Taylor & Francis Group.

between the system and the examinee. During administration of the task, the presentation process records the work product (the raw response) of the examinee and delivers it to the evidence identification process (EIP).

3. *Evidence identification process* (also called *response processing*). Evidence identification is the first stage of scoring: identifying the key features of the examinee performance that bear evidence about the examinee's knowledge, skills, and abilities. Evidence identification receives the work product(s) from the presentation process and assigns values for one or more *observable outcome variables*. It records the observables in the examinee record and sends them to the EAP for aggregation across several tasks, as well as to the activity selection process for generating task-based feedback or other immediate activity selection decisions.
4. *Evidence accumulation process* (also called *summary scoring*). The EAP acts on the evidence provided in the observed outcomes from particular tasks, as received from the evidence identification process, to update the current belief about the examinee's knowledge, skills, and abilities. This updated belief is then available for the next iteration of the activity selection process and/or for summary feedback.

Central to each of these processes is the *task/evidence composite library.* The task/evidence composite library is a database of assessment task objects, descriptions of these objects, the information necessary to select and present them, indication of the examinee work product to be retained, the information necessary to score the work product, and the process for integrating task evidence into updated beliefs about examinee knowledge, skill, and ability.

The IMS Global Consortium adopted the four-process architecture as part of their information model for question and test interoperability (IMS 2000). In doing so, they renamed the "evidence identification" as "response processing" and "evidence accumulation" as "summary scoring." (They also renamed "work products" as "responses" and "observables" as "outcomes.") The names using "evidence" fit better with the evidence-centered design philosophy that is the focus of this book, and they apply just as well to assessments that do not look like familiar tests.

13.1.1 A Simple Example of the Four-Process Framework

The roles of these four processes, and the function of the physical elements of the four-process architecture, can be illustrated by following a dichotomously scored IRT CAT (item response theory computer adaptive test[3]; see Sect. 7.4.1) through a single cycle of the administration process.

We begin in the top left-hand corner with the role of the administrator. The *administrator* is the person(s) responsible for initializing and maintaining the assessment delivery. In initializing the assessment, the administrator must make various choices, such as initializing the examination in the proper mode when alternative modes are available. For example, in the Biomass assessment (Steinberg et al. 2003; this volume, Chap. 14) the administrator could choose between a diagnostic assessment to support learning and an end-of-unit summary assessment. The administrator might also make examinee-specific configuration decisions; for example, whether the examinee will receive special accommodation for disability, such as increased time or larger screen size. In this role the administrator will access a preexisting examinee profile or create one for the administration.

The *examinee record* is the collection of data about the examinee for the assessment administration. Initially, the record may possess little more than basic demographic information and data about the registration of the examinee for the assessment. As the assessment progresses, the examinee record will accumulate data related to that examinee's performance on the assessment such as those describing the tasks presented to the examinee, the examinee response latencies, and the navigation of the examinee through the assessment interface.

A key part of the examinee record is the *scoring model.* It contains variables that can be used in reporting scores; particularly, variables describing the current belief about examinee knowledge, skills, and abilities. The scoring model variables will start at initial values based on the population of learners who typically use the assessment as described in the proficiency model. As the assessment progresses, the values of the scoring model variables gradually change to reflect the evidence provided by the participants' performances.

[3] See Wainer et al. (2000) for a good introduction to IRT CAT, and van der Linden and Glas (2010) for a more technical treatment.

For the IRT CAT example, the assessment begins with the creation of a new examinee record for the examinee based on the examinee's profile and the rules given by the assessment design. The scoring model is a distribution over the single proficiency variable, θ. Typically this is either a flat distribution or a normal distribution whose mean and variance are given by the population.

Once initialization is complete, the activity selection process—in this case an automated routine—accesses the task/evidence composite library to select an initial task (in this case, all of the tasks are simple items) for the examinee. In so doing, the activity selection process considers a variety of features of potential tasks, all of which are stored in the task/evidence composite library. These factors may include item content, item exposure rates/control variables, information function of the item, item weights of evidence (in this case, item parameters), and dependence with other items in the task/evidence composite library. Because the test is adaptive, the activity selection process also considers the current state of knowledge about the examinee's proficiency (initially vague) to aid in selection. Once it has selected a task, the activity selection process sends a message to the presentation process informing it about which task to schedule next and delivering the parameters necessary to adapt the item for the individual.

The presentation process receives from the activity selection process the messages describing the task and the manner in which it is to be administered. Acting on this information, the presentation process searches the task/evidence composite library and retrieves the task and associated presentation material to present to the examinee. The examinee interacts with the presentation process to complete the selected task. During this interaction the presentation process records the *work product* (captured part of the response) of the examinee in accordance with the requirements of the task. In the case of single multiple-choice items, the work product is simply an identifier for the selection made by the examinee. In other situations, the work product may contain a constructed response or a trace of examinee actions, results, and navigation through a simulation.

The presentation process sends the resulting work product to the evidence identification process, which accesses the task/evidence composite library to obtain the rules for evaluating the work product. The evidence identification process applies the *evidence rules* to the work product to determine from the raw work product a collection of *observed outcome variables* that can be used to update beliefs about the examinee. In the case of multiple-choice items, this can be a simple process of fetching the key, determining whether the examinee response matches the key, and setting the appropriate observation to a value of "correct" or "incorrect." For diagnostic feedback based upon multiple-choice items, we may need to produce additional observables based on the selected distractor. Constructed response items use more complex evidence rules which are often expressed as rubrics for human raters or as algorithms for automated scoring. Most rules of evidence are parameterized and along with the task itself, the task/evidence composite library must store this *evidence-rule data.*

For example, for multiple-choice items the key must be indicated, and for essay tasks, task-specific scoring rubrics or scoring notes are required.

Designated results of the evidence identification process can be output as task level feedback, either immediately during the assessment or stored in a database for future use. The task level feedback can be as simple as indicating whether the examinee got the item correct or incorrect or as complex as providing explicit diagnostic feedback regarding the strategies and cognitive processes used to solve a complex task, depending on the nature and purpose of the assessment.

The evidence identification process sends the observed outcomes from the evaluation of the work product to the EAP. The EAP accesses the task/evidence composite library to obtain the links.[4] In the case of CAT examinations using dichotomously scored multiple-choice items, this process entails accessing the task/evidence composite library to acquire the item parameters and to process the item response data to update the current belief about the examinee on the basis of the observables obtained from the item response. The updated beliefs about the examinee and the associated data, such as the identification of the item that was administered, the response, actions, work product, etc., are then placed into the examinee record, where they remain available for the next iteration of the four-process cycle.

The examinee record thus always contains what is currently known about the examinee. Any time a score report is needed, the examinee record can be queried to produce summary feedback for the examinee. This can be as basic as a number-right score for a linear multiple-choice test or as complex as a set of diagnostic and instructionally relevant statements describing examinee knowledge and strategies, depending on the nature and purpose of the assessment (and how the four processes are defined to meet that purpose).

The four-process architecture is a general enough framework that most existing assessment delivery implementations can be mapped into it. Often the evidence identification process is lumped in with either the presentation process or EAP, but separating it out, at least conceptually, gives us a great deal more flexibility. A simple example of this is a test with written essay tasks. We could use either human scorers or sophisticated natural language processing algorithms to score an assessment. Maintaining evidence identification as a separate process allows us to swap one for the other without changing the other processes or the messages passed among all the processes.

The goal of this chapter is to show how to implement the EAP for an assessment that uses Bayesian networks as its primary scoring model. Section 13.3

[4] As we will see below, links are task-specific versions of the evidence models defined in the previous chapter. They contain explicit information and procedures describing how the new evidence will be used to update the current belief about examinee knowledge, skills, and abilities. In particular, they contain *weights of evidence*, the values of the task specific parameters in the link, such as conditional probability tables in Bayes nets models and item parameters in IRT models

describes the primary operations needed to support the other three processes and how they play out with Bayesian networks. Section 13.2 describes how we fill the task/evidence composite library with the data needed to support all four processes.

13.2 Producing an Assessment

This section describes how we can go from the CAF described in Chap. 12 to the four-process architecture. Section 13.2.1 describes the authoring of tasks. Section 13.2.2 describes the authoring of evidence rules and the evidence-rule data they require. Finally, Sect. 13.2.3 describes the process of calibrating the measurement model from pretest data, a process that produces *links*, or task-specific versions of the evidence model.

13.2.1 Tasks and Task Model Variables

One frequently asked question about the CAF is "Where are the items (tasks)?" The task models that are defined in the CAF are not objects, but metaobjects: objects that describe other objects. A task model specifies what kind of information must be presented to the examinee, but not the exact information. That is left to the task.

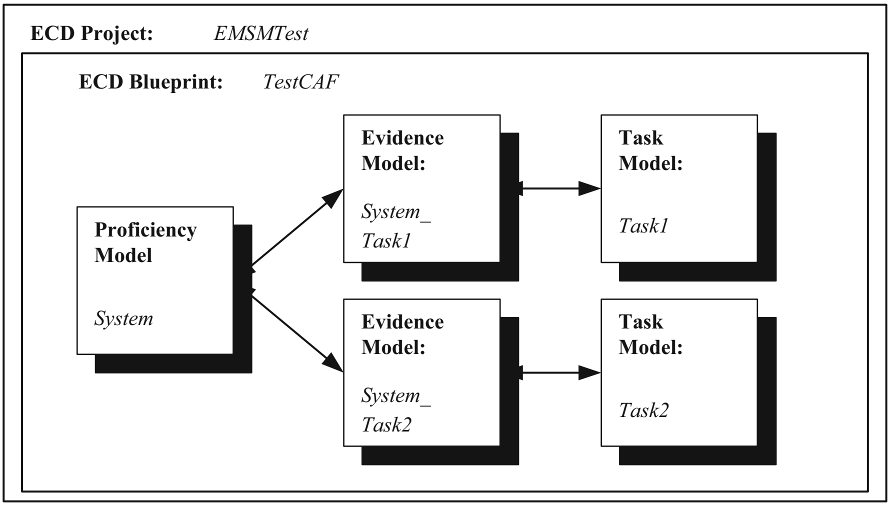


Fig. 13.2 Assessment blueprint for a small test. Reprinted with permission from ETS.

Each task model in the CAF defines a family of potential tasks. The task author creates instances of the task model as realized tasks. Consider the CAF

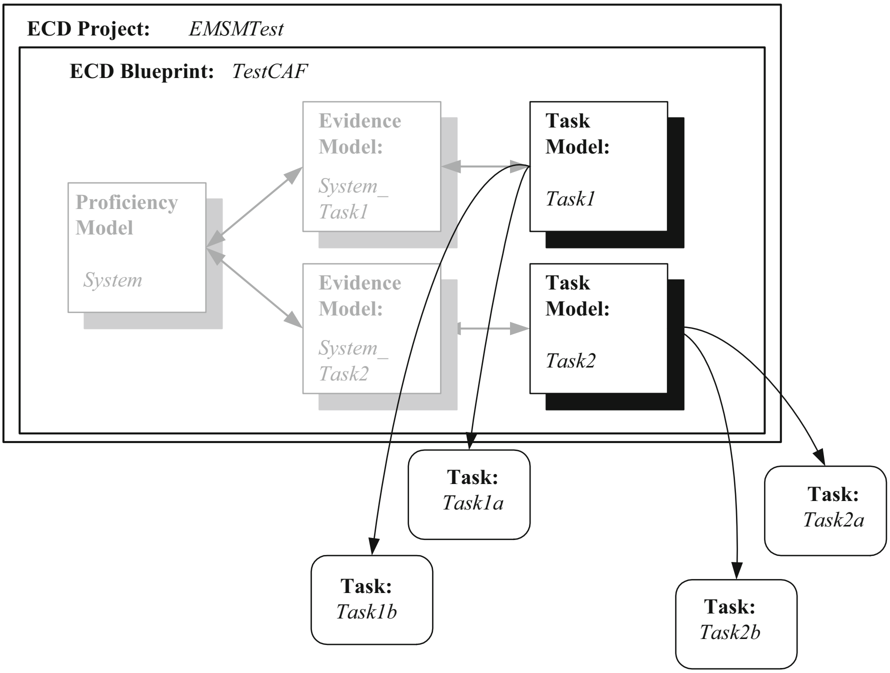


Fig. 13.3 Tasks generated from CAF in Fig. 13.2
The two tasks are generated from each of two task models. Reprinted with permission from ETS.

given in Fig. 13.2. Assume that two tasks are authored for each of the two task models; Fig. 13.3 shows the result.

The task author has two important activities in authoring a task. The first is to find or produce material which will be presented to the examinee as a stimulus for the problem. This includes text, pictures, and sound stimulus material, as well as prompts and in the case of multiple choice the key and any distractors. The second activity is to assign values for all of the task model variables. The task model defines the name and possible values for the required task model variables. (The evidence-rule data also depend on the task model variables, so it is often authored at the same time.) In the final task, the values for all of those variables must be determined.

Task model variables' values are related to the presentation material. For example in a subtraction problem, the *Minuend* and *Subtrahend* are presentation material; specifically, numbers. The *Number of Digits in the Subtrahend* is a task model variable whose value is clearly related to the *Subtrahend*. The task model variables may be determined before or after the work product. For example, suppose that the *Semantic Density* of a reading passage is a task model variable. The task author might first find a passage and then evaluate its semantic density, or the author might try to find a passage that meets a target density.

Mislevy, Steinberg, and Almond (2002c) define a number of roles for task model variables (Sect. 12.3). Any task model variable may be used for one or more of these roles. In brief they are: facilitating task construction, controlling evidential focus, constraining assessment assembly, mediating the relationship between performance and proficiency (in particular, determining difficulty), and characterizing proficiency.

There is an advantage when, for some task model variables, there is a correspondence between the levels of the task model variables and the levels of the proficiency variables. For example, suppose the proficiency *Skill 1* has three levels: `high`, `medium`, and `low`. Suppose we can further categorize the tasks using *Skill 1* into two categories: `simple` applications, and `complex` applications. The natural claims for the Skill variable are as follows:

`high` A person who operates at the `high` level of *Skill 1* can usually solve problems which require a `complex` application of the skill.

`medium` A person who operates at the `medium` level of *Skill 1* can usually solve problems which require a `simple` application of the skill, but has difficulty with problems which require a `complex` application of the skill.

`low` A person who operates at the `low` level of *Skill 1* has difficulty with problems which require a `simple` application of the skill.

In this scheme, tasks that require a `simple` application of the skill will have a good weight of evidence for distinguishing between the `low` and `medium` levels of *Skill 1*. Tasks that require a `complex` application of the skill will have good weight of evidence for distinguishing between `medium` and `high` levels of the skill. (See Exercise 7.7.)

Some task models variables are used to generate details that lend verisimilitude to the task but do not otherwise contribute to the purpose of the task. For example, the names of the actors in a word problem can usually be freely modified without substantially changing the difficulty of the tasks. Collis et al. (1995) call such task model variables *incidentals* and contrast them to *radicals* which are used to manipulate difficulty or evidential focus, or otherwise affect the the form or the parameters of the evidence model.

Identifying and manipulating incidentals allows for the automatic generation of tasks (Gierl and Haladyna 2012). Because the new tasks differ on surface features, it is harder to memorize a task and answer. Security increases without having to calibrate additional tasks. It is hard to find pure incidental variables, though. Returning to the example of the names of the actors in the word problem, using unusual names will add difficulty because the examinees must decode the unfamiliar words rather than just recognize them. Ideally, the changes to the conditional probabilities caused by variables considered incidental will be smaller than other sources of error in the scoring process.

Variables that are radicals (that is, they are believed to affect the relationship between performance and proficiency) can be passed on to the calibration process. There they can be used to model the parameters of the evidence models, either directly or through some sort of hierarchical modeling that reflects clustering of the tasks according to their task model variables (Geerlings et al. 2011; Mislevy et al. 1993). Section 13.2.3 takes up the issue of calibration.

A step beyond automated task *generation* is automated task *recognition*. It is useful in open-ended performances such as problem-solving simulations and language proficiency interviews. The idea is that in more open-ended problem-solving spaces, recurring evidence-bearing situations arise as examinees work their way through a complex task. A task model can now be used to describe a class of such situations: *emergent tasks*. An instance of an emergent task is recognized when a prespecified set of values for the defining task model variables occurs.

Human raters use intuitive "automated task recognition" informally when they give broad ratings to complex performances. An assessor conducting an oral proficiency interview, for example, recognizes places where an examinee should use past tense and notes whether or not she does. This information is incorporated into the holistic rating she assigns the examinee. An example of a testing program that applied the approach more formally was the Praxis III: Classroom Performance Assessments^{TM} assessment for beginning teachers (Dwyer 1998). Trained assessors make observations with respect to 19 categories of more generally described dimensions of teaching practice that are meant to apply across a range of teaching styles, curricula, and student compositions.

An example of a computer-based simulation assessment that used automated task recognition is HYDRIVE (Mislevy and Gitomer 1996). If an examinee worked himself into a situation where the problem-solving strategy called space-splitting was possible, then an instance of a space-splitting task was recognized and the next sequence of actions that had an effect on the active path in the problem was interpreted as a work product for this kind of task. Recognizing an instance of a space-splitting task required posting an examinee's moves to the student record and running a program called an agent to evaluate the state of the problem as it evolved, summarized in a set of dynamic task model variables—that is, task model variables whose values are computed by the presentation process. The agent that recognized instances of space-splitting tasks (and other agents that recognized instances of other classes of tasks) continually monitored the task model variables that defined classes of emergent tasks. When an instance was recognized, the agent signaled the presentation process to capture the associated work product and send it to the appropriate evidence identification process.

Task authors must perform one final step before completing their work, and that is to identify any data required by the evidence rules. For example, a multiple-choice item is useless unless the key is known. A scoring rubric for an essay task is based on the markers of evidence that are needed to support the claims that performance on the task is meant to support. If the same task is to be used in multiple assessments with multiple purposes and hence multiple evidence models, then evidence-rule data must be developed for each evidence model that might be used to score the task. The next section takes this up in more detail.

13.2.2 Evidence Rules

The reason for the central database in the four-process architecture being called the task/evidence composite library is that it contains both information required by the task models and information required by the evidence models. The "task" part of the task/evidence composite consists of the presentation material for that task and any task model variables that are needed by the activity selection process (i.e., have the role of assessment assembly), or any of the other processes. The "evidence" part consists of both parameters for the conditional probability tables in the *links* (the specialized version of the Bayes net fragment in the evidence model for this task) and any data required for the evidence rules. Section 13.2.3 looks at the issue of creating and calibrating links. This section looks at the evidence rules.

Evidence rules are a mechanism for the reasoning between the raw response that is captured as the work product of the examinee interacting with the task, and belief about the examinee's proficiencies. In some cases these rules may be very simple; for example, matching the key in a multiple-choice item. In other cases they could be quite complex, like the grading of an essay or the application of a neural network to evaluate a hundred automatically recognized linguistic features of the same kind of essay. Evidence rules produce values for one or more observable outcome variables.

In order to reuse evidence rules across multiple tasks (usually from the same task model), they must be parameterized. For a key-matching rule, the parameter is the key. For scoring an essay, in addition to the scoring rubric (the evidence rules) there are often prompt-specific scoring notes instructing the scorers about what specific points to look for in the student essays. These serve as the "parameters" for the essay-scoring rules. In complex situations the parameter of an evidence rule could be quite complicated. Often these are expressed with scoring tables or some other similar data structures. These parameters are called evidence-rule data.

ROC Analysis

An interesting case is one in which the observable outcome is determined by making a cut on an underlying count or continuous property of the work product. The count or continuous property could be the output of one or more *parsing rules* which extract relevant features from a work product.

For example, consider a task in which an examinee is asked to write down as many names for articles of clothing as possible within a specified time period. We can define an intermediate variable by counting the number of correct and incorrect answers provided. Suppose that one of the final observables is *Correct Set Size* which can take on values `small`, `medium`, and `large`. In this case the evidence-rule data will be a threshold for cutting the count variable produced in the parsing step. This kind of cut point rule is interesting because we can tune the thresholds to give the task better evidential properties.

Suppose that both the proficiency variable and and final observable outcome variable are binary. When both variables are binary, it is usually the case that one state of the proficiency variable is better (more positive) than the other; similarly one state of the observable is usually considered better (e.g., correct) than the other. It should also be the case that when the proficiency variable is in the positive state, the correct value of the observable is more likely than when the proficiency variable is negative (if this does not hold, then the observable generally is not providing good evidence). In this case, we can "classify" the student on the basis of the observable. If the proficiency variable is in the positive state and the observable is in the correct state, then this is a true positive; however, when the proficiency variable is in the negative state and the observable is in the correct state, we call that a false positive. True and false negatives are similarly defined. Table 13.2 shows graphically the joint probabilities for the observable and the proficiency.

Table 13.2 Confusion matrix for binary proficiency and observable

		Proficiency	
		High	Low
Observable	Correct	True positive, p_{tp}	False positive, p_{fp}
	Incorrect	False negative, p_{fn}	True negative, p_{tn}

The diagonal cells of Table 13.2, often called a *confusion matrix*, contain the probabilities of the observable that point toward the actual proficiency state, p_{tp} and p_{tn}. The off-diagonal cells contain the probabilities of the observables that point towards the other proficiency state, p_{fp} and p_{fn}. False positives are also called *Type I Errors* and false negatives are called *Type II Errors*. A number of statistics can be defined based on these numbers:

- The *sensitivity* is defined as $p_{tp}/(p_{tp} + p_{fn})$; this is also called the *recall*. It is the proportion of positive cases that the observable actually indicates as positive.

- The *specificity* is defined as $p_{tn}/(p_{fp}+p_{tn})$. This is the proportion of negative cases the observable indicates as negative.
- The *positive predictive value* is defined as $p_{tp}/(p_{tp}+p_{fp})$; this is also called the *precision*. The *false discovery rate* is the complement of the positive predictive value, $p_{fp}/(p_{tp}+p_{fp})$.
- The *negative predictive value* is defined as $p_{tn}/(p_{fn}+p_{tn})$.
- The *accuracy* is defined as $p_{tp}+p_{tn}$. Note that if one of the states is rare, then it is possible to have fairly high accuracy just by chance. Cohen's kappa is the accuracy corrected for chance agreement (see Sect. 7.5.1).

When the observable variable is created from a feature of the work product via a cut point, then the sensitivity and specificity will depend on the chosen cut point. If having more of the feature is better, then moving the cut point higher should decrease the Type I (false positive) error rate and the expense of the Type II (false negative) error rates. Moving the cut point lower has the opposite effect. The receiver operator characteristic (ROC) curve (Fig. 13.4) is a useful tool for visualizing the trade-off between the two types of errors.

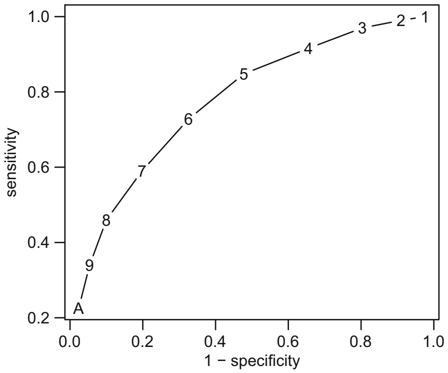


Fig. 13.4 A sample receiver operating characteristic (ROC) curve
This curve plots the sensitivity (true positive rate) vs. 1− specificity (false positive rate) for various choices of the cut point for an evidence rule. It looks like 6 or 7 would make good choices. Reprinted with permission from ETS.

The ROC curve comes out of signal detection theory, but it is extensively used in the field of medical testing. The basic form is to plot the true positive rate (sensitivity) versus the false positive rate (1− specificity). The diagonal line at 45° represents the worst possible test, one that simple guesses randomly. The more area under the curve above the diagonal line, the better the test. Different tests can be compared in this way.

The ROC is most often used to explore the trade-off between false positive and false negative errors. We can pick a cut point based on which type of error we would rather avoid. Consider the example in Fig. 13.4.[5] If we choose 5 as the cut-off, we get a sensitivity of 88 %, but a specificity of only 50 %. If we choose 8 as the cut-off, we get a specificity of 90 % but only get a sensitivity of 50 %. Depending on what type of error we would rather make we can adjust the test to either end or in between. This is actually fairly typical for a situation in which the underlying feature is a count. In this case, the ideal cut point is often between the two possible values, and one of them is used.

Evidence Rule Analysis

However much research goes into the CAF, it is still a theory of how the assessment should work. As such it predicts how the students should behave when they address the tasks. When the tasks are placed in front of actual students, the results can be surprising. These surprises happen frequently enough that the best practice for assessment design is to perform some kind of pilot test. Usually a small scale pilot is first, often with students videoed or interviewed after completing the assessment to learn how students actually think about and interact with the item. Later a larger scale pilot is used to get statistical information about the students' performance.

Suppose we have pretest data for a large number of examinees that are a representative sample of the potential assessment population. We can now do some model checking on the evidence rules. This evidence rule analysis is very similar to conventional item analysis. However, given that the target proficiency model is a Bayesian network, there are some techniques that are particularly attractive.

The first place to start is with the marginal distribution of the observed outcome variables. For dichotomous observables, this is sometimes called the P^+ in item analysis. Obviously, if all or nearly all of the pretest population obtain the same value for the observed outcome, then the task will have little weight of evidence for almost any hypothesis. This is grounds for revisiting the design and implementation of both the evidence rules and the task.

The strategies for fixing such a task depend on which observable values were not observed in the pretest data. If the best possible outcomes were not observed, the task may be too difficult for the target population. The designers should consider simplifying the problem, or even if that kind of task is appropriate for this population. If the worst possible outcomes were not observed, then the task is too easy. The designers should again consider modifying or dropping the problem. There are though a number of situations in which tasks that are too easy are still useful. In an end-of-course or end-of-unit assessment, easy tasks serve as confidence builders for the students, letting them know that they have mastered as least some of the material. Also, getting a very easy task wrong has a high diagnostic value in identifying students who have very low levels of ability and need remedial work.

[5] This example was generated artificially using two Poisson distributions, one with mean 8.5 and one with mean 5.5.

If there are more than two observable outcomes, often the best solution is to combine a state with very few students with one of the other states. When the states are ordered, this is often a problem with one of the middle states. The question then arises of whether to merge the unneeded state with its higher or lower neighbor. Reviewing the claims associated with the observable states helps determine which combination is appropriate in a given situation.

Every evidence model assumes a relationship between one or more proficiency variables and an observable outcome variable. This assumption can be checked through the two-way table formed by one of the observable outcome variable and one of the proficiency variables it provides evidence for. The problem is that the proficiency variables are latent, and we cannot observe them directly. However, we can approximate their values using the other tasks in the pilot test.

If we have built the links for all of the tasks, then estimating the value of the proficiency variable is simply a matter of applying the usual scoring algorithm to the rest of the tasks. The catch is that we may not yet have performed the calibration step (Sect. 13.2.3), so we do not have final values for the link parameters. As this is just an exploratory procedure, setting the link parameters to the mean of their prior distributions (from the evidence models) should produce a close enough approximation to get a rough estimate of proficiency—not good enough for important decisions about students, but useful for exploration and troubleshooting.

With a Bayesian scoring model, there are two frequently used scores: MAP and EAP scores. They lead to different ways of constructing the two-way score-by-observable table. The MAP produces as a "score" the most likely value of the proficiency variable. In this case, the table of interest is the crosstab of the estimated proficiency variable and the observed outcome variable, a matrix A where a_{ij} is the number of examinees who were classified as having proficiency state i and obtained observable value j.

The EAP leads to the *expected accuracy matrix* (Sect. 7.5.3). Here we use the marginal distribution of the proficiency variable for Examinee e, $\mathrm{P}(S_e)$. Suppose that Examinee e obtains the Observed Outcome j. Then $\mathrm{P}(S_e)$ becomes a contribution to the jth row of the matrix A. That is

$$a_{ij} = \sum_{\text{Examinee } e \text{ gets Outcome } j} \mathrm{P}(S_e = s_i) \ .$$

The matrix constructed from the MAP estimates may suffer from a high degree of variability when the sample size is small. The matrix constructed from the marginal distribution should have smaller estimation error.

Table 13.3 shows the expected accuracy matrix for an observable called *PC3* in an experimental task, Task Exp4.1. In this case there was a pretest with 500 examinees. The abilities are estimated from another set of 12 tasks with known link parameters. The table on the left shows the raw numbers. The entry in each cell is the sum of the probability of being in the state

Table 13.3 Expected accuracy matrix for observable *PC3* in Task Exp4.1

		Skill 1		
		Low	Med	High
Exp4.1	2	8.86	83.38	67.76
	1	66.47	107.22	20.30
	0	44.60	73.86	27.54
	Total	119.93	264.45	115.60

Unnormalized

	Skill 1		
	Low	Med	High
2	0.07	0.32	0.59
1	0.55	0.41	0.18
0	0.37	0.28	0.24

Normalized

indicated by the column for all students who got the observable value on the left. For example, the cell in the upper left is the sum of the probability of being in the low state for all of the students for the 160 students who got a 2 on *PC3* in Task Exp4.1. The differing column totals for the left-hand table make the columns difficult to compare. To make the table easier to interpret, we divide by the column totals, producing the table on the right. Now we can see the pattern we hope: as the proficiency gets higher, the probabilities shift towards the better values of the observable.

Table 13.4 MAP accuracy matrix for Task Exp4.1

		Skill 1		
		Low	Med	High
Exp4.1	2	7	85	68
	1	63	111	20
	0	44	74	28
	Total	114	270	116

Unnormalized

	Skill 1		
	Low	Med	High
2	0.06	0.31	0.59
1	0.55	0.41	0.17
0	0.39	0.27	0.24

Normalized

Table 13.4 shows the same analysis, but using the MAP estimates of proficiency rather than the marginal distributions. Again, we normalize the table by dividing by the column sums to make the results easier to interpret. With a sample of 500, the differences between the two analyses are negligible.

Table 13.5 MAP accuracy matrix for Task Exp6.1

		Skill 1		
		Low	Med	High
Exp6.1	2	9	68	52
	1	1	0	1
	0	104	202	63
	Total	114	270	116

Unnormalized

	Skill 1		
	Low	Med	High
2	0.08	0.25	0.45
1	0.02	0.00	0.01
0	0.91	0.75	0.54

Normalized

Table 13.5 shows a task that has a problem: At all three proficiency levels, almost nobody picked the middle category. One possibility is a problem with the instructions for the task, causing students to not approach the task correctly. A second possibility is a problem with the scoring rules, that somehow the distinction between 1 and 2 score (or 0 and 1 scores) is not properly being made. One simple possibility is to eliminate the middle scoring category and just assign scores of 0 or 2 for this task. (The conditional probability tables in the evidence model need to be collapsed if the middle category is eliminated.)

Ultimately, we are interested in tasks that have a high expected weight of evidence for cut points on the proficiency variable. One way of determining the strength of the relationship is to look at the the mutual information between the proficiency variable and the observable. Observables for which the mutual information with the target proficiency variable is small (in particular, where it small with respect to other similar tasks) are not contributing much to the overall estimation. Designers might consider dropping or reworking them.

A central assumption of the partitioning of our model into proficiency model and evidence model is that the observable outcome variables in our model are independent of all of the other proficiency variables given the boundary variables. This is a testable hypothesis. For simplicity, consider the case where the footprint consists of a single proficiency variable. Using the scores and observed values from the pretest data, we can construct a three-way table with the observed outcome, the boundary variable and a different proficiency variable. Table 13.6 shows an example.

Table 13.6 Three-way table of two observables given proficiency variable

$\theta = \mathtt{high}$		
	$Y = 1$	$Y = 0$
$X = 1$	35	7
$X = 0$	35	11

$\theta = \mathtt{med}$		
	$Y = 1$	$Y = 0$
$X = 1$	31	99
$X = 0$	33	50

$\theta = \mathtt{low}$		
	$Y = 1$	$Y = 0$
$X = 1$	43	102
$X = 0$	18	36

MAP estimates for θ were used in calculating this table.

In this three-way table, we can test this conditional independence assumption using a chi-square or Mantel–Haenszel test (see Bishop et al. 1975; Fleiss et al. 2003). Essentially, what the Mantel–Haenszel test does is to fit the model assuming that the observable and boundary variables are independent given the boundary variable, and looks at the residuals from that fit. If the residuals are high, then we reject the independence hypothesis and question whether the evidence model is truly appropriate for the task. Bishop et al. (1975) suggest, as an alternative, fitting both models with and without independence and looking at the change of deviance in the models.

The χ^2 test for this difference is straightforward to calculate. Consider the data in Table 13.6. The three states for the boundary variable *Skill 1* (denoted θ) define three strata of ability. We essentially perform a χ^2 test for independence at each strata. Consider the subtable on the left. If the conditional independence property holds, the the expected values for the cells in that table will be $\mathrm{P}(X|\textit{Skill 1} = \texttt{high})\mathrm{P}(Y|\textit{Skill 1} = \texttt{high})$. The expected values for the other two strata are calculated similarly. However, we do not know the corresponding probabilities, so we need to estimate them from data.

Consider a configuration of possible states, $x \in \mathrm{states}(X)$, $y \in \mathrm{states}(Y)$, $z \in \mathrm{states}(\theta)$, and let n_{xyz} be the observed number of student who had MAP a proficiency level of z and observed values x and y for X and Y. To estimate the various probabilities, we need to look at the sums across the various dimensions. Call the sum over all of the states of X, $n_{+yz} = \sum_{x \in \mathrm{states}(X)} n_{xyz}$, the sum over all the states of Y, $n_{x+z} = \sum_{y \in \mathrm{states}(Y)} n_{xyz}$, and the sub over the states of both X and Y, $n_{++z} = \sum_{x \in \mathrm{states}(X)} \sum_{y \in \mathrm{states}(Y)} n_{xyz}$. Under the hypothesized conditional independence assumption, the expected value for each cell is then, $\hat{n}_{xyz} = n_{x+z} n_{+yz} / n_{++z}$, and we can calculate a χ^2 goodness of fit statistic in the usual way:

$$\chi^2 = \sum_{z \in \mathrm{states}(\theta)} \sum_{x \in \mathrm{states}(X)} \sum_{y \in \mathrm{states}(Y)} \frac{\left(\hat{n}_{xyz} - n_{xyz}\right)^2}{\hat{n}_{xyz}} . \tag{13.1}$$

The obvious reference distribution is a χ^2 distribution with $(|X|-1)(|Y|-1)|\theta|$ degrees of freedom (here $|X|$ refers to the number of possible states for variable X). The problem is that the values of θ are not known, but rather are estimated from data. Therefore, we are not sure that the χ^2 distribution is the proper reference distribution. However, the quantile of the χ^2 distribution still provides a rough heuristic for cases that require further observation. Using the MAP for θ, the χ^2 value for Table 13.6 is 7.1, which is close to the critical value of 7.8, so it may be worthwhile to look for causes of dependence between the two tasks generating X and Y.[6]

Table 13.7 Three-way table of two observables given marginal proficiency

$\theta = \texttt{high}$		
	$Y = \texttt{1}$	$Y = \texttt{0}$
$X = \texttt{1}$	34.65	22.78
$X = \texttt{0}$	36.04	17.90

$\theta = \texttt{med}$		
	$Y = \texttt{1}$	$Y = \texttt{0}$
$X = \texttt{1}$	36.26	89.13
$X = \texttt{0}$	32.46	42.94

$\theta = \texttt{low}$		
	$Y = \texttt{1}$	$Y = \texttt{0}$
$X = \texttt{1}$	38.08	96.04
$X = \texttt{0}$	17.48	36.14

Marginal estimates for θ contributed to fractional values for each case.

We could perform a similar test using the expected accuracy matrix (i.e., using the marginal distributions in place of the MAP entry). The values of

[6] The table was actually produced from simulated data in which there was residual conditional dependence. This may be a problem with the χ^2 reference difference, or it could be the dependence was too small to detect with this sample.

n_{xyz} would no longer be integers, but fractions as each observation would contribute only a fractional value to each cell (Table 13.7). This is essentially the method proposed in Sinharay et al. (2004), and is similar to the model checking techniques covered in Chap. 7. The χ^2 value in this example is 5.0. This suggests that the χ^2 reference distribution is not adequate (this was noted by Sinharay et al. 2004). Comparing Tables 13.6 and 13.7 shows that using the marginal values in place of the MAP estimates smoothes the table quite a bit; hence the lower χ^2 values. We could instead use posterior predictive model checking as discussed in Sect. 10.2 to find a reference distribution for this statistic.

Differential item functioning (DIF) (or in our case differential task functioning) can also be expressed as a conditional independence assumption (Sect. 10.4). A test is considered fair if given the proficiency variables in the boundary for the evidence model, the task performance is independence of group membership (e.g., gender, race). We can similarly construct a three-way table using the boundary variables, the observed outcome and a demographic variable representing group membership and explicitly test for this independence assumption using the Mantel–Haenszel test (Holland and Thayer 1988).

One frequent response to problems of this type is to change the evidence rules associated with observables flagged as problematic. However, changing the evidence rules could change the classification of the pretest examinees on one or more skills. The best thing to do would be to rerun these tests after the adjustments to the evidence rules. Calibration (next section) also will change the classifications, so these tests are best run before calibration and again after calibration.

13.2.3 Evidence Models, Links, and Calibration

Although evidence-centered design encourages test developers to make the factors which they are manipulating in tasks (the task model variables) explicit, this alone is not sufficient to completely control variability in task performance. In actual practice, examinees can surprise the test designers, and react to a task in completely unexpected ways. This difference between theory and the real world happens often enough that it is necessary to verify the performance of any task before using it. The evidence rule analysis techniques described in the previous section will catch gross departures from the expected results, but smaller changes are also possible; for example, a task could be easier or harder than expected, or the dependency on one of the proficiency variables could be stronger or weaker than predicted by the model. In low-stakes situations, this variability is commonly ignored and the end users live with a certain amount of approximation error in their estimates. In situations where that is not acceptable, the evidence models can be calibrated to produce more accurate results.

Just as the task is a realization of the task model, the *link* is the realization of the evidence model for a specific task. (If the task model supports more

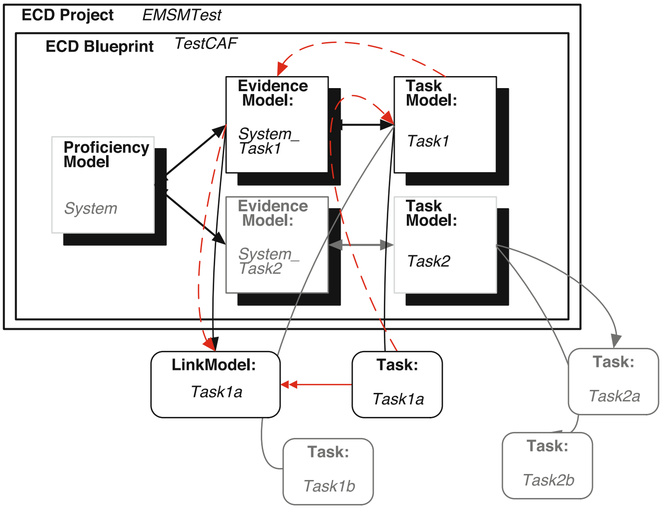


Fig. 13.5 Link model generated from Task 1a in Fig. 13.3
The link corresponding to Task 1a is created through calibration. Reprinted with permission from ETS.

than one evidence model, the task will have a link corresponding to each one). The link has exactly the same graphical structure as its parent evidence model. It only differs in the values of the parameters (weights of evidence), which have been tuned for the specific task.

Just as the task model defines a set of possible tasks, the evidence model defines a prior distribution over the set of possible links. The methods of Chap. 9 (Markov chain Monte Carlo, MCMC or expectation–maximization, EM algorithms) can be used to calculate the posterior (given pretest data) distribution for a link for a given task. In later scoring, the posterior means of the parameters for the links are used to score the assessment (although the full posterior distributions are maintained for future calibrations, as discussed in Sect. 9.6.2 with regard to their use in online calibration).

For lower-stakes situations, we can skip the calibration step, and simply use the prior (evidence model) mean as the parameters for scoring. (The previous section used this trick to get a preliminary scoring in order to test the evidence rules.) In this case, the link is essentially a copy of the evidence model. If controlling the task model variables sufficiently constrains the operating characteristics of tasks arising from task models, setting the link parameters equal to the evidence model parameters might not be a bad approximation. For example, an automatic task generation procedure might be allowed to randomly select incidentals, but hold all radicals fixed. In this case, the error by assuming that all links have the same parameter might be ignorable (although in moderate-to-high-stakes assessments, this assumption is worth testing).

Even if the radicals are not fixed, their values can be used during calibration. Fischer (1973), Embretson (1998), and Mislevy et al. (1993) demonstrate this idea using IRT models. These ideas have seldom been put into practice, in large part because of the expense of coding task model features retrospectively. Evidence-centered design solves that problem by making the capture of the values for the task model variables, in particular, the radical variables, part of the design process. A notable illustration is the British Army Recruitment Battery (BARB) (Irvine 2013). Every item in every test for every examinee is generated from a model built around a cognitive theory, radicals, and incidentals. Both the presentation materials and link models are generated automatically.[7]

Usually links are not calibrated one at a time, but instead all of the links for an entire form or pool are calibrated simultaneously with the proficiency model. The calibrated links and proficiency models go into the task/evidence composite library to form the data to drive the EAP. The assessment description is an index to all of the material stored in the task/evidence composite library that should be used for administering and scoring a given assessment.

13.3 Scoring

In the four-process architecture, the responsibility for scoring is divided between the EIP and the evidence accumulation process (EAP). The EIP is responsible for scoring that is local to a particular task; that is, processing the work product to produce the task level observed outcomes. These outcomes can be used to drive task level feedback or sent to the EAP so the evidence they contain about proficiency variables can be integrated with evidence from the results from other tasks, or both. The EAP is responsible for scoring that occurs across tasks in an assessment or a section. It produces the scores that are used on summary score reports.

We can think about the four processes as four agents, each of which responds to messages from the others.[8] Each of the agents follows a certain protocol: a set of messages that the agent will accept, and a description of how they respond to those messages. That protocol defines how each process behaves for the other processes, so that each process can regard the others as

[7] Prof. Sidney Irvine, the creator of BARB, relates that the inspiration for BARB came from a challenge from Dr. John Anderson: “What would tests be like with no item-banks, no IRT--and no money!”

[8] This is the fundamental idea of object-oriented programming: that parts of the program can be thought of as “agents” who communicate through messages. Software engineers have found that by dividing up a complicated program in this way, they can assign responsibility for each of the pieces to a different programmer, and have reasonable assurance that when all of the pieces are assembled the system will work properly.

a black box that does the things defined in its protocol. This is very useful for the assessment designer, as the process of implementation can now be split into several pieces, which can be located on different machines (say a client workstation in the test center and a large server in the computer center) and which can be developed by different software vendors.

The goal of this section is to describe the protocol for the EAP, which will give a good idea for how scoring works in four-process architecture. Section 13.3.1 describes the basic protocols necessary to support scoring a linear test, or any test in which the adaptivity is based on task level outcomes and not accumulated scores. Section 13.3.2 describes the minor changes that need to be made to these protocols to support adaptive testing. Section 13.3.3 describes some technical issues like the handling of omitted and repeated tasks. Finally, Sect. 13.3.4 describes how score reports are generated.

13.3.1 Basic Scoring Protocols

The heart of the EAP is what it does when presented with new evidence in the form of observed outcomes sent from the evidence identification process. For a Bayes net-based scoring engine, the basic idea is that it finds the link particular to a given task, and "docks" it with the scoring model—the proficiency model for a particular examinee. It then instantiates the evidence in the link and propagates it to the scoring model. The docked link can now be discarded. Scores are reported by querying the scoring model. Figure 13.6 gives a general picture. Almond and Mislevy (1999) gives a detailed description.

We can describe this protocol more formally by considering the EAP as a server that reacts to a number of different incoming messages. The three most important messages are:

`Candidate Begin` A new candidate has arrived to start the assessment.
`Absorb Evidence` New evidence has arrived for a task from the EAP.
`Report Score` A score report is needed for this candidate.

We assume that each message contains header information to provide context. In particular, each message is associated with a particular candidate and a particular assessment (this defines which task/evidence library the EAP should search for data) and a task.[9]

To better understand how the EAP must work, we explore what happens in response to these three messages. For this example, we assume that the assessment description calls for two proficiency models, a Bayesian network model for the primary scoring and a number right model which is used primarily to count tasks. This is actually a trick used in building the Biomass assessment (see Chap. 14). The Bayes net was used for the primary inference, but the number right model was used produce descriptive information about

[9] Although there are special IDs for the beginning and end of an assessment, when the task is not needed.

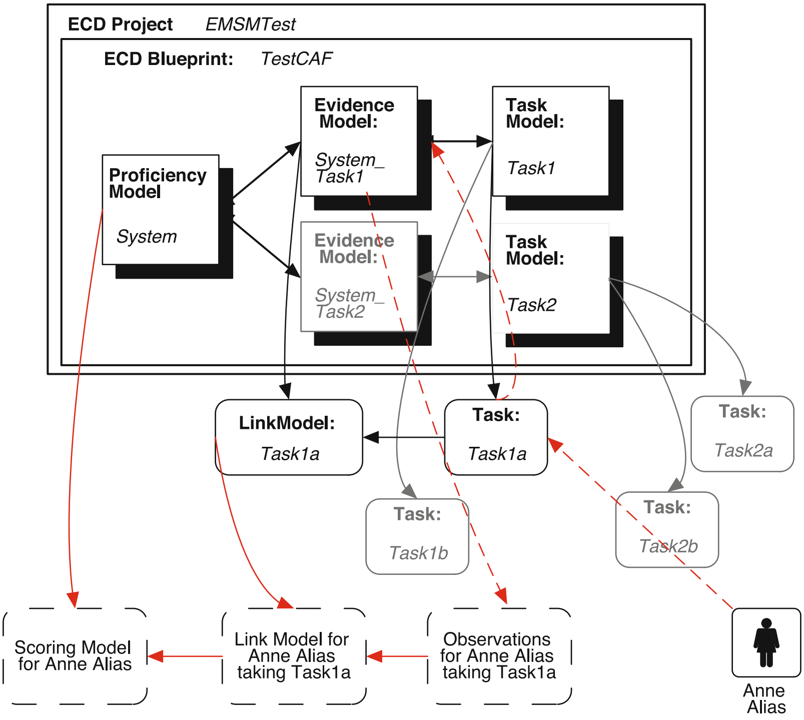


Fig. 13.6 Absorbing evidence from Task 1a for a single candidate
The scoring model is created from the proficiency model when the candidate begins the assessment. As evidence arrives from Task 1a, a copy of the link is obtained and the observable values are instantiated in that copy. The evidence is then propagated into the scoring model and the link copy is discarded. Reprinted with permission from ETS.

the number of tasks the student had attempted for the score report. This design required two sets of links for each task, one for the Bayes net proficiency model and one for the number right proficiency model. The idea can be further extended to additional EAPs, such as ones that simply accumulate whether a given misconception is evidenced in a task where it is apt to surface.

`Candidate Begin`. This message is received at the start of an assessment, and tells the EAP to start an examinee record for a given candidate. The message header should contain both the identifier for the candidate and the identifier for the assessment. The EAP then looks up in the *assessment description* (essentially the assembly model for the assessment) what the appropriate proficiency model is for this assessment. It then asks the proficiency model to create a scoring model for this candidate. If the assembly model contains more than one proficiency model, it creates multiple scoring models; the examinee record is then a container for those scoring models.

In the Bayesian network framework, the scoring model is essentially an examinee-specific copy of the proficiency model.[10] The proficiency model is thus the prior value of the scoring model, containing our prior beliefs about the examinee before seeing any responses. As we see more responses, the scoring model will move away from the proficiency model to become a posterior specific for that student. The proficiency model will remain at the initial state ready to produce a new scoring model for the next examinee.

In the number right case, the scoring model is a collection of counters; counters for the number of tasks seen, the number of tasks regarded as "correct," the number of score points earned and the number of score points that could have been earned. If there are subscores, each one of them requires a set of counters as well. The counters are set to their initial values (usually zero, but may be something else according to the scoring rule defined in the assessment description).

At this point, we have a new examinee record for the first examinee. In our running example it contains two scoring models: a Bayes net initialized to the prior distribution values defined in the Bayes net proficiency model, and set of counters for the number right model initialized to the values defined in the number right proficiency model.

`Absorb Evidence`. When the evidence identification process finishes processing the work product from a task response, it sends an `Absorb Evidence` message to the EAP. The header for the `Absorb Evidence` message must convey which examinee, which assessment, and which task the evidence comes from. The body of the message contains a number of names of observable outcome variables and their values. The EAP then must fetch the appropriate scoring model from the examinee record collection and the appropriate links for the task named in the message from the task/evidence library. It can then use these links to update the scoring model.

In the case of the example, one link corresponds to the number right scoring model and one to the Bayes net scoring model.[11] Consider the link for the Bayesian network model. It contains a Bayesian network fragment linking some of the proficiency variables to one or more observable variables. The EAP finds the observable variable in the message and instantiates the variables in the Bayes net fragment at the values in the message.

It is possible that one or more of the observable variables in the fragment is missing from the message, or a given observable variable is irrelevant to a particular performance. In these cases, such observables are left uninstan-

[10] This is the reason we depreciated the use of the term "student model." The proficiency model refers to the properties of a population of students, the scoring model refers to a single student.

[11] It is possible to have multiple links update the same scoring model (as when a task comprises multiple subtasks with distinct links), but this adds complexity without adding insight. For the moment we will assume that there is exactly one link for each task and scoring model pair.

tiated. It is also possible that there are observable variables in the message that do not appear in the Bayes net fragment. This could happen if the EIP generates additional observables for task level feedback, or if they are needed for the number right score. The Bayes net EAP ignores the extra variables.

Once the observable variables are instantiated in the Bayes net fragment, the marginal distribution for the boundary variables is calculated. Recall that the boundary variables are the proficiency variables found in both the proficiency model and evidence model (and by extension in both the scoring model and the link). Thus, the marginal distribution over the boundary nodes can be entered into the scoring model as virtual evidence. This procedure is described in more detail in Sect. 5.4 (also Almond et al. 1999).

The link for the number right model does a simpler version of the same thing. It looks for the observable that corresponds to "correct" for this task, and if it has the "correct" value, it increments the "correct" counter. It also increments the number of tasks counter, no matter what the value of the parameter. The number right link has an additional parameter, a scoring weight. It multiplies the "correct" value by the weight and adds that to the score counter. It also increments the possible score counter by the value of the weight.

After this update, the examinee record contains two updated scoring models. The Bayesian network model contains the posterior distribution given the observations from this task (and all previous tasks from which evidence was absorbed). The number right model model contains the counts of the number of tasks seen and the number for which the "correct" value was observed. The EAP stores the examinee record away for the next time a message comes in for this examinee; either another `Absorb Evidence` message or a `Report Score` message.

`Report Score`. Any time another process needs information about the state of the scoring model it can send a message requesting the value of certain scores. The message header contains an identifier for both the examinee and the assessment. The body of the message contains a list of the desired scores. The EAP finds the scoring model for the examinee and the Reporting rule (Sect. 12.2.3) for each requested score. It returns a message giving a value for each one.

Note that the scoring models always contain all of the accumulated evidence about the examinee. In the number right model, this is a count of the number of tasks and the scores. The scores are simple functions of the counter variables in the scoring model. For example, the *percent correct* is the number of tasks that were "correct" divided by the number of tasks seen then multiplied by 100 (to make a percentage). The percentage score is calculated similarly, except it uses the number of weighted points earned divided by the total number of possible points (based on the tasks the student has seen).

In the Bayes net model, the scoring model contains our posterior belief about the examinee given the observations made so far, in the form of a posterior distribution over the proficiency variables involved in that EAP. All of the scores we report are statistics of that posterior distribution. Common examples are the marginal distribution or mode (MAP estimate) of one of the proficiency variables. Section 12.2.3 provides a large number of examples. These can be calculated using the algorithms described in Chap. 5.

A fully realized EAP will require other messages as well, such as `Candidate End` (close scoring model and clean up), `Save Candidate` (save scoring model to file or database for use in later session), and `Restore Candidate` (recreate scoring model from previous save). These are mostly details for the programmers to worry about, and do not affect the logic of how the tests are scored.

In summary, the EAP for any particular examinee taking a linear test performs the following sequence of actions:

1. When the examinee sits for the examination, the proficiency model creates an initial scoring model for that examinee. In the case of a Bayesian scoring model, this is a prior distribution over the proficiency variables.
2. As the EAP receives observed outcomes for this examinee from the evidence identification process, it updates the scoring models for the examinee. In the case of a Bayesian scoring model, this prior estimate of ability in the scoring model is updated using the likelihood of the observed outcomes and Bayes' rule. This likelihood is specified by the link for each task. Afterward, the scoring model contains the posterior estimate of ability. (For a linear test, the distribution over the proficiency variables in the scoring model can be updated all at once, or task by task. The answer is the same, since the likelihood for the score vector is just the product of the task likelihoods.)
3. When we wish to draw inferences from our scoring model (in a linear test, usually only at the end of the test), the scores are defined as functions of the scoring model variables. In the case of a Bayesian model, scores are usually statistics of the current posterior distribution of the proficiency variables, such as the posterior mean and standard deviation, or posterior mean and standard deviation for some transformation of the proficiency variables such as a scale score or a market-basket score (Sect. 12.2.3).

13.3.2 Adaptive Testing

The three basic messages described in the previous section are sufficient for a linear test—a test in which the sequence of tasks is fixed in advance—or for a test in which the examinee can pick the sequence of tasks. An adaptive test requires closer communication between the EAP and the activity selection process. Before selecting the next item in an adaptive test, the activity selection process first consults the EAP about our current beliefs about examinee proficiency. This can happen in one of two ways: (1) either the activity selection process can query the EAP about the value of the statistics it wants, or (2) the EAP can automatically generate certain statistics (including the ones needed by activity selection) in response to every `Absorb Evidence` message. On the basis of these statistics, activity selection selects an item which maximizes the value of evidence, subject to content constraints and exposure controls.

Automatically reporting the value of key statistics (for example, the marginal distribution of reporting variables, or the percentage of tasks solved correctly) at every iteration of the update has other advantages. Often looking for a trace of the probability values for certain nodes over time (as new observed outcomes arrive) can help explain how scores came about. Recall that the evidence balance sheets (Chap. 7) are calculated by looking at the difference in probability at adjacent time points.

If we plan to use expected weight of evidence to do task selection as described in Sect. 7.3, the EAP must support one additional message type, a `Calculate EWOE` message. This message's header must specify the examinee, so that the EAP can find the correct scoring model. The body of the message should specify both what hypothesis is under consideration and a list of candidate tasks. In response to this message, the EAP fetches the scoring model for the examinee and the link for each task in the list of possibilities.

Recall that a hypothesis corresponds to a set of possible proficiency profiles. To calculate the expected weight of evidence, the EAP starts by instantiating the hypothesis in the scoring model; that is, it restricts the variables to values that appear in the hypothesis. It then performs the update algorithm in reverse, that is, it calculates the marginal distribution over the boundary variables for each link, and propagates it to the link. Given the marginal distribution for the boundary variables and the link, the EAP calculates the joint distribution of the state of the observable variables given that the hypothesis holds. It calculates this conditional distribution for each possible link.

Next, the EAP temporarily instantiates the negation of the hypothesis in the scoring model and repeats the calculation. From this data we can calculate the expected weight of evidence (Eq. 7.5) for each task. This can be returned as an ordered list of tasks or a value for each task.

This calculation is typically too expensive to do for all of the tasks in the task/evidence composite library. Typically some sort of heuristics are necessary to prune the list of possibilities before calculating the expected weight of evidence. One simple heuristic is to look for tasks which will provide direct or almost direct (one or two nodes removed) evidence about the hypothesis node. Another possibility is to divide the tasks up into sections covering the content domain, and then search for the best tasks within each section. This approach has the advantage of not jumping around to items of very different types.

Example 13.1 (ACED). *Adaptive Content with Evidence-based Diagnosis (ACED) is a computer-based assessment of sequences appropriate for a course in middle school mathematics (Shute et al. 2005; Shute et al. 2007; Shute et al. 2008). It is a prototype designed to explore: (a) the use of the Madigan and*

Almond (1995) expected weight of evidence algorithm (Sect. 7.3) to select the next task in a assessment, (b) the use of targeted diagnostic feedback, and (c) the use of technological solutions to make the assessment accessible to students with visual disabilities.

Graf (2003) describes the construction of the proficiency model. ACED spanned three sequence types—arithmetic, geometric, and other recursive sequences—commonly taught in 8th grade, but tasks were only developed for the geometric sequences. The model is expressed as a tree-shaped Bayesian network with the following proficiency with an overall sequences proficiency node at the top, with nodes for arithmetic, geometric, and other recursive sequences as its immediate children. Figure 13.7 shows the details for the geometric sequences branch of the model.

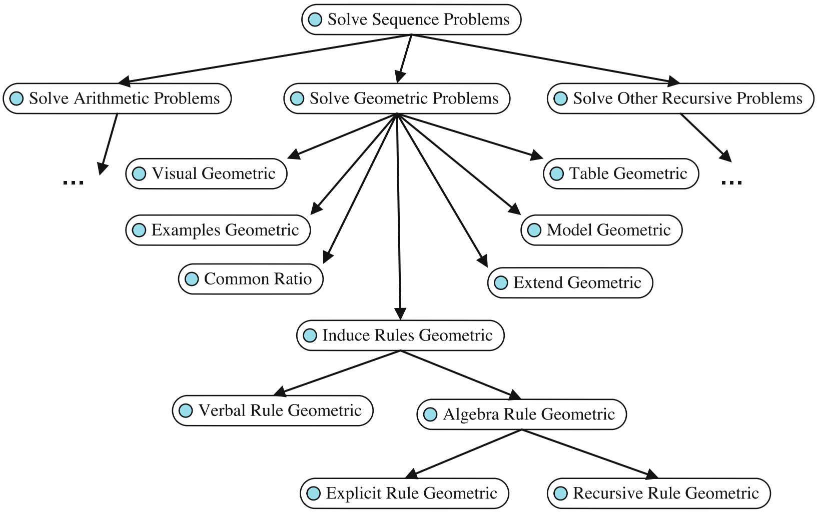


Fig. 13.7 ACED proficiency model
Only the central branch of the model for geometric sequences is elaborated. The other two branches were symmetric to it Reprinted with permission from ETS.

ACED used the expected weight of evidence algorithm almost exactly as described in Sect. 7.3. If the SolveGeometricProblems node was chosen as the hypothesis, the activity selection process went through the 63 tasks developed for ACED and picked the most informative task. As this was always the same task at the beginning of the test, ACED selected a random task for the first one to force different sequences. The student answered the first task, and ACED updated the proficiency model. ACED then checked the remaining 62 tasks to find the one with the highest expected weight of evidence.

Table 13.8 shows the expected weight of evidence calculation using selected tasks (all medium difficulty). The second column of numbers is produced *by temporarily instantiating the target variable, Solve Geometric Problems to `low`. The values in the column are the probability that the each of the observables will be correct after the instantiation. To get the numbers in the first column, we retract the previous instantiation and now set the hypothesis to be true. In this case Solve Geometric Problems $\geq$ `medium` is a compound hypothesis. To set this to be true, we use virtual evidence: We set the likelihood of any state for which it holds (i.e., `high` and `medium`) to be one and the likelihood for any state for which it does not hold (i.e., `low`) to be zero. After propagating these values, the probabilities that the observables are correct give the numbers in the first column. With these sets of numbers, the expected weight of evidence can be easily calculated using Eqs. 7.1 and 7.5.*

Table 13.8 Calculation of expected weight of evidence

	$H =$ *Solve Geometric Problems* $\geq$ `medium`		
Task	$\mathrm{P}(X = 1 \mid H)$	$\mathrm{P}(X = 1 \mid \overline{H})$	EWOE
Common Ratio Task	0.72	0.62	0.0222
Explicit Rule Task	0.29	0.28	0.0002
Recursive Rule Task	0.49	0.46	0.0018
Verbal Rule Task	0.44	0.31	0.0372
Table Task	0.55	0.36	0.0746
Visual Task	0.56	0.34	0.1001

Numbers based on a simplified version of the Bayesian network model for ACED (Shute et al. 2008; Example 7.5); in particular, this model only uses medium difficulty tasks.

ACED continued this process until the student had seen all 63 tasks (these were short math items taking a minute to solve, not the complex performance tasks used in Biomass). The ACED design team talked over rules that would be used to switch hypotheses nodes from the arithmetic, to the geometric, to the other-recursive sequence nodes following the critiquing strategy discussed in Madigan and Almond (1995), should the other two branches be implemented.

ACED was built so that the activity selection process could be switched between the adaptive expected weight of evidence and a linear algorithm that returns the tasks in a fixed sequence. It also had two feedback modes, one that provided accuracy–only feedback, and one that provided an elaborated feedback based on misconceptions middle school student are likely to have about sequences. This allowed an evaluation of ACED (Shute et al. 2007; Shute et al. 2008) where about 300 students were randomized into four different treatments: (1) adaptive sequencing and elaborated feedback, (2) adaptive sequencing and accuracy only feedback, (3) linear sequencing and elaborated feedback, and (4) a control group which did not use ACED at all. The partic-

ipants in the study were also given a short pretest and posttest on geometric sequences.

Even though the task sequence was adaptive, ACED was configured to present all 63 tasks to each student. Shute et al. (2008) looked at the correlations between the posttest score and the expected a posteriori (EAP) scores for various subsequences of the items. For the adaptive sequence conditions, the correlation between the ACED EAP scores based on the first 20 items and the posttest score was as high as the correlation between the EAP score from all 63 items and the posttest. Furthermore, that correlation was about as large as the reliability of the posttest. Therefore, the adaptive version of ACED was doing about as well as possible after 20 items. The linear version, in contrast, needed nearly the full 63 items to reach the same level of correlation with the posttest. To be fair, the linear sequence was chosen in such a way that the full 63 items were needed to span the space of geometric sequence problems. A 20- or 30-task linear sequence that was selected with that test length in mind should do better.

However, this was not the most interesting affect of the adaptive selection reported in Shute et al. (2008). Looking at the difference between the pretest and the posttest scores, the control and accuracy–only feedback showed virtually no change. The elaborated feedback paired with the linear sequence showed a small but not significant gain. The only significant gain between pretest and posttest was shown by the group that got both elaborated feedback and adaptive sequencing.

This result was somewhat surprising. After all, the claim made about the EWOE algorithm is that is provides optimal information about a student, and not that it optimal for learning. Note that the highest EWOE will be for an observable that the student has a 50 % chance of getting correct based on the current state of the proficiency model. It is possible that this kind of task is one that is in the right place in that student's zone of proximal development (Vygotsky 1978) to optimally promote learning. However, this speculation needs to be confirmed with a more systematic study.

13.3.3 Technical Considerations

The preceding sections show a relatively simple procedure for scoring assessments based on just a small number of messages. Adding a few additional messages enables the system to support adaptive testing as well. However, real-world testing situations are seldom so simple. There are two issues we need to work through: how to handle missing responses and repeated responses.

Handling omitted responses is never a simple problem. It is an issue that cuts across all of the models in the conceptual assessment framework and all of the processes in the four-process architecture. Handling missingness as a student interacts with an assessment belongs as part of the delivery model but we will see it holds implications for the task model, evidence model, and assembly model.

Part of the problem is that there are many reasons that a response may be omitted, and they can hold different implications for how it should change our beliefs (Mislevy 2015). The National Assessment for Educational Progress (NAEP) uses three codes for missing values: `omitted` for when a subject has left an item blank but answered later items, `not reached` for when a subject left an item and all subsequent items in the booklet blank, and `error` when the subject made multiple selections or some other problem occurred which caused it to be impossible to record a response. The situation is even more complex in the case of extended constructed response tasks or simulations with large complex work products. Here the system must be robust enough to deal with complex patterns of omitted responses.

The EAP described in the previous sections handles missing observables by not instantiating the corresponding node in the graph. This is equivalent to treating the omitted response as if it was neither positive nor negative evidence. This is in fact the statistically appropriate way to handle missing values when those missing values satisfy Rubin's conditions of missing at random (MAR) or missing completely at random (MCAR) (Sect. 9.3). NAEP's `not reached` and `error` missing values are both MCAR. So are hypothetical responses to items on a NAEP form that a student was not administered. Further, responses to tasks that are not presented to an examinee in adaptive testing are MAR.

One particular result is worth underscoring here, because it applies to two of the most assessment situations where one might most want to use Bayesian networks as a measurement model, due to their modularity and recombinability. The first is in adaptive testing: Students are presented tasks based on the values of their previous responses, in order to maximize information or to obtain evidence about a particular claim (Sect. 7.4.1). The second is in simulation- and game-based assessment: an examinee's actions influence the task situation as it evolves. Since in both cases the task depends on observable behavior and beyond that not on the values of either the latent variables or the unobserved responses, the missingness is MAR. Hence, the correct likelihood is obtained through the conditional probability matrices of the observations given the proficiencies (Rubin 1976).

Sometimes, however, the fact that the examinee deliberately omitted the response in fact provides negative evidence for the skills required. In particular, examinees who have self-confidence in the required skills are more likely to attempt the task, and at least sometimes self-confidence in having skills is correlated with actually having them. Simply omitting the response does not seem appropriate. Rubin calls this "nonignorable" or "informative" missingness, because observing it should cause us to revise our beliefs about proficiency. A simple solution for omitted multiple-choice responses is to treat them as partially correct, at the guessing level, or the reciprocal of the number of alternatives (Lord 1980). This is what NAEP does with `omitted` responses. Another approach is to formally introduce an additional response category for tasks, `omitted`, and estimate conditional probabilities for `omitted` given proficiency like the other responses. This approach can be implemented using a DiBello–Samejima categorical IRT strategy, but with the underlying links now obtained through a nominal response IRT model (such as Bock 1972). More sophisticated alternatives we will not explore here further introduce models for examinees' propensities to omit, and model omission and response jointly (see Mislevy 2015). Ultimately how to handle different kinds of missingness is a policy decision to be made by the administrative authority for the test, ideally informed jointly by theory about missingness and an understanding of the missingness mechanisms that are in play.

The key to omitted response handling in the four-process architecture is that the evidence rules must specify exactly what happens for omitted, null, invalid, or other categories of missing work products, according to the policy decision of the test. There are two approaches the EIP can be instructed to take that require no additional machinery: (1) set the observed outcome variables to a predefined default value (e.g., to have them counted a "wrong" answers), and (2) omit the observables from the message (to have them omitted from scoring). In either case, the EIP could create an additional observable that would report whether the work product was present or not (which could then be modeled explicitly in the statistical part of the model). The options for response processing for not-reached values would be similar.

The approach adding a new response value for `omitted` requires comparably extended evidence models and links. Lord's approach of treating omits as fractionally correct requires an additional `Absorb Virtual Evidence` message (Sect. 5.2.3): Rather than indicating a single value for an observed outcome variable, this message provides a predetermined vector across its possible values to propagate to the proficiency model. A virtual-evidence vector of (.75, .25) over the values 0 and 1 would be used for an omitted four-alternative multiple-choice item.

If the rules for navigation in the test allow the examinees to return and resubmit results, the EAP may need to deal with repeated collections of observed outcomes from the same task (this will not be true if the results are not sent to the EAP until the end of the assessment, but will be true if they are sent as they arrive). If the EAP applies the normal processing rules to the repeated tasks, it will record the evidence as if it were two independent administrations of the task. Although this may not be a bad approximation in some tutoring systems, it is problematic in a higher-stakes assessment.

One possible solution is to add a `Retract Evidence` message to the EAP's interface. (This would also have uses in calculating the influence of a particular task, and in diagnostics which call for calculating the scoring leaving out a particular task.) In this case the protocol for the EAP could be to replace the evidence from the previous attempt at the task with the evidence from the latest attempt. Even this is likely to be unsatisfactory in the case where task level feedback, such as a hint, has been presented to the examinee between the first and second attempts. One way to deal with multiple attempts with feedback is to define an observed outcome variable that combines information about correctness and number of attempts, then use a graded response model. The values of an observable outcome variable for a dichotomous item with feedback or would not simply be `Right` and `Wrong`, but `Right-on-the-first-attempt`, `Right-on-the-second-attempt`, etc.

13.3.4 Score Reports

The last role of the EAP is always to calculate the scores. As was mentioned previously, it can be called upon to do so any time it receives a `Calculate Scores` message; however, such a message will almost certainly be sent at the end of the assessment. The body of the `Calculate Scores` message contains a list of which scores are required, along with any parameters that might be necessary (for example, a request for a percentile may have a parameter describing which percentile is required, e.g., 50, 75, 90, 95). Sometimes the score report is generated immediately by the presentation process; sometimes the values of the statistics are stored in a database with the examinee record so the score reports can be generated later on demand.

As mentioned earlier the scores are always some kind of statistic of the scoring model. In the cases of a Bayesian network scoring model, this might be the marginal distribution of one of the nodes. In the case of the number right scoring model, it might be the percentage of tasks answered correctly.

More complex kinds of reporting might require additional messages. Consider the case of market basket reporting where the goal is to predict how the examinee would perform on a standard set of tasks (the market basket). This is fairly straightforward with a Bayesian scoring model. The scoring model contains the posterior information about the student's proficiencies. A link is required for each task in the market basket; that link gives the likelihood of any pattern of observables given a proficiency profile. This can be used to provide a probability distributions over possible observable patterns for each of the market basket tasks.

The desired information to be placed in the score report should be decided early in the design process. It is good practice to reason backwards from that information to the statistics required to generate it, and then to ensure that proficiency models (and hence the scoring models) for the assessment support it. In the Biomass score report (Fig. 14.4, Chap. 14) a count of the number of tasks is required in addition to the information from the Bayes margins. To obtain the task counts, an additional number right proficiency model was added (actually this was just a task counter model, as it did not use a notion of "right"). Together the two scoring models provided the information required for each assessment.

It is worth spending a great deal of time thinking about the score report. The score report and the actual tasks are the two parts of the assessment that are visible to the end users. Ultimately, the assessment will be judged on whether or not the score report provides the information needed by the end users for the purpose to which they want to put the assessment. It is incumbent on the score designers to transform the information that resides in the scoring models at the end of an assessment into a form in which the end users can understand both what is known and what remains unknown about the student.

Exercises

13.1 (Four-Process Chart Fill-in). For each of the following scenarios, describe what the four processes are and whether they are human or computer processes, and whether they happen at the time the examinee sits for the assessment or afterward.

1. A student taking a college entrance examination administered with paper booklets and scanned answer sheets.
2. A student using a practice examination from a study guide for that college entrance examination.
3. A student reviewing the practice book with a tutor.
4. A student using an online study system for this assessment that simulates the real assessment.
5. A student using an online study system for this assessment that drills the student on certain item types.
6. A student using an adaptive online study system for this assessment.

13.2 (IRT CAT). Section 6.1 introduced a discrete Bayes net approximation of an IRT model. Design a CAT item selection procedure, beginning with Item 3: Use Expected Weight of Evidence to determine the next most informative item to administer if a correct response to Item 3 is observed, and if an incorrect response is observed. Determine the next item to administer among the remaining ones if the second response is correct, and if it is incorrect, and so on. Repeat the exercise using mutual information to select items. Compare the results.

13.3 (Raw Mouse Clicks vs. Higher Level Events). A certain testing program is using computer-administered multiple choice tasks where the computer places circles next to the options. For each of the following parts of the software, say whether it logically belongs in the presentation or evidence identification process. Justify your answers.

1. A subroutine which maps the location of the mouse click (e.g., screen coordinate $(104, 121)$) to which option was selected (e.g., `A`, `B`, `C`, `D`, or `E`)
2. A subroutine which compares the selection made by the examinee to the key and returns a value of `correct` or `incorrect` according to whether or not it matches.

13.4 (ROC Calculation). A certain vocabulary test designed for 8th grade students has a task that presents a certain word and asks the examinee to write down as many related words as possible within 3 min. To try to figure out how many words to expect, the design team chose a group of experts (college educated adults) and a group of novices (6th grade students) and gave each of them the task. They tested 15 of each. The numbers are given in Table 13.9. Calculate an ROC curve for distinguishing between experts and novices on this task.

Table 13.9 Data for Exercise 13.4

Experts	24, 27, 17, 31, 22, 16, 19, 17, 28, 28, 21, 14, 31, 26, 33
Novices	8, 9, 7, 8, 7, 10, 7, 6, 7, 7, 10, 7, 14, 7, 17

13.5 (Constructing an Expected Accuracy Matrix). Table 13.10 shows the estimated scores and observed outcomes for an item called Exp1.1, for 10 randomly selected students in a (simulated) pretest. Construct the expected accuracy and MAP accuracy matrixes for these data. Is the difference in the normalized values bigger or smaller than that between the normalized version of Tables 13.3 and 13.4?

Table 13.10 Ten randomly selected entries from a set of pretest data

ID	EAP	MAP	P($\theta = \mathtt{high}$)	P($\theta = \mathtt{med}$)	P($\theta = \mathtt{low}$)	Exp1.1
Simulee7	0.46	Low	0.01	0.43	0.56	1
Simulee378	0.46	Low	0.01	0.43	0.56	0
Simulee1	1.70	High	0.71	0.27	0.01	1
Simulee441	0.61	Med	0.07	0.47	0.46	0
Simulee64	0.45	Low	0.07	0.32	0.62	1
Simulee183	0.61	Med	0.07	0.47	0.46	1
Simulee40	0.66	Med	0.04	0.58	0.38	1
Simulee333	0.58	Med	0.03	0.52	0.45	1
Simulee422	0.29	Low	0.01	0.28	0.72	0
Simulee487	1.78	High	0.79	0.20	0.01	0

13.6 (Mutual Information). Calculate the mutual information between the target skill and the observable for the two tasks in Table 13.11. The pretest form had six reference tasks, which had mutual information of 0.0048, 0.0266, 0.0121, 0.0492, 0.0317, and 0.0697. Is one of them abnormally low? If so, what is the problem? [Hint: Look at normalized tables.]

Table 13.11 Expected accuracy matrix for two experimental tasks

		Skill 1		
		Low	Med	High
Exp1.1	1	69.78	107.57	74.61
	0	117.96	93.23	36.76

		Skill 1		
		Low	Med	High
Exp2.1	1	60.04	72.07	41.86
	0	127.71	128.73	69.50

13.7 (Distractor Analysis). A multiple-choice task consists of three kinds of presentation material: *the stem*, or the initial question, *the key* or the correct answer, and the *distractors* or incorrect answers. A problem that is frequently observed with multiple-choice tasks (especially vocabulary tests) is that one of the distractors will not be understood except by people with high ability, and hence the task will provide weak information. Look at the expected accuracy matrices for the two experimental tasks in Table 13.12. Does one of them exhibit this problem? If so, which one?

Table 13.12 Expected accuracy matrix (normalized) for two multiple-choice tasks

		Skill 1		
		Low	Med	High
Exp7.1	d	0.19	0.17	0.07
	c	0.25	0.27	0.33
	b[a]	0.30	0.39	0.44
	a	0.26	0.18	0.17

		Skill 1		
		Low	Med	High
Exp8.1	d	0.06	0.10	0.26
	c[a]	0.36	0.44	0.43
	b	0.44	0.38	0.28
	a	0.14	0.09	0.03

The key is marked with an "a."

13.8 (Differential Task Functioning Detection). To look for possible differences in the way male and female students approached a given item, the design team used a pretest sample of 50 male and 50 female students to produce the table shown in Table 13.13. Do these data indicate that there is cause for concern?

Table 13.13 Data for differential task functioning detection problem (Exercise 13.8)

	Male		
	Low	Med	High
`Incorrect`	7.15	6.21	3.64
`Correct`	2.72	12.58	17.69

	Female		
	Low	Med	High
`Incorrect`	7.04	8.22	4.74
`Correct`	3.11	11.22	16.66

13.9 (Conditional Independence Test). The design team has some concern that the two observables X and Y are dependent even though they are from different tasks. To test this hypothesis they produce the three-way table shown in Table 13.14 using the MAP value for the proficiency. On the basis of these data, is there cause for concern?

Table 13.14 Data for conditional independence test problem (Exercise 13.9).

	Skill=`low`			*Skill*=`med`			*Skill*=`high`		
	0	1	2	0	1	2	0	1	2
0	99	24	31	35	24	9	0	0	2
1	34	1	7	51	58	25	8	14	33
2	2	1	0	3	6	2	3	12	16

13.10 (Expected Weight of Evidence). Continuing from Example 13.1, assume that we present the student with *Visual Task* and the student gets a correct result. Table 13.15 gives the conditional probabilities after this new evidence is absorbed. Calculate the expected weight of evidence for the remaining tasks. Which is the best task to present next?

Table 13.15 Calculation of expected weight of evidence after one observation

	H = *Solve Geometric Problems* $\geq$ `medium`		
Task	$\mathrm{P}(X=1\vert H)$	$\mathrm{P}(X=1\vert \overline{H})$	EWOE
Common Ratio Task	0.72	0.62	
Explicit Rule Task	0.29	0.28	
Recursive Rule Task	0.49	0.46	
Verbal Rule Task	0.45	0.31	
Table Task	0.56	0.36	
Visual Task	1.00	0.00	– –

Numbers based on the Bayesian network model for ACED (Shute et al. 2008; Example 7.5), conditioned on a correct result from *Visual Task*

13.11 (Repeated Tasks with Context Effect). A certain low-stakes assessment consists of a number of extended tasks which the students can attempt over the course of several weeks. The students are allowed to make multiple attempts at the tasks on different days, and the design team decides to treat these as independent pieces of evidence. Several of the tasks require a fair amount of background reading, and it is thought that students who have previously studied the topic will have some advantage. To model this, the evidence models for those tasks has a local variable called *Context*.

The question is, how should this variable be treated on repeated attempts at the task? Should there be different instances of *Context* local to each link? Or should *Context* be shared across separate instances? In either case, how does the model need to be adjusted to take this into account?

14

Biomass: An Assessment of Science Standards

This chapter and the following one illustrate the use of evidence-centered design (ECD) and Bayes nets to build the assessment *Biomass*, a prototype of an interactive, inquiry-based assessment of secondary biology (Steinberg et al. 2003). Section 14.1 provides a background for the project. Section 14.2 summarizes domain analysis and domain modeling activities that supported the design of assessment objects and delivery processes, and Sect. 14.3 describes the resulting Conceptual Assessment Framework. Section 14.4 describes the four-process delivery system used in the prototype. The following chapter provides numerical details about the models and our efforts to refine the models from data.

14.1 Design Goals

A primary purpose of the Biomass project was to provide a test-bed for the ECD methodology then under development. As such, we were trying to design an assessment which would meet several goals:

- Provide meaningful feedback (both task-level and summary feedback) based on standards for the domain.
- Demonstrate ECD modularity and support for repurposing the assessment by building two variants that support different purposes: (a) formative assessment for classroom use, and (b) a culminating assessment that provides evidence of whether or not standards have been met at the end of the unit.
- Take advantage of web-based infrastructure to support complex, interactive, automatically-scored tasks.
- Demonstrate the ability of ECD to disentangle evidence from complex, integrated tasks that tap more than one proficiency.

The result was Biomass, a web-delivered, interactive assessment that can be used in two ways: as a formative assessment that supports learning in a

R. G. Almond et al., *Bayesian Networks in Educational Assessment*,
Statistics for Social and Behavioral Sciences, DOI 10.1007/978-1-4939-2125-6_14

standards-based curriculum, and as a culminating assessment that provides evidence of whether standards have been met, for purposes such as college admissions or course placement.

When used as a culminating assessment, Biomass reports on students' learning in terms of standards in the domain, and also provides supplemental information for further study. To familiarize students with the forms and conventions of the culminating test as well as the content and expectations, there is a formative version that can be used for coached practice. When used as a formative assessment, Biomass supports practice and self-evaluation. In this use it informs students and teachers about progress toward mastery. It addresses the same knowledge base and skills as the culminating assessment, but it would be used by students working individually or together, often as part of a course, to practice and to prepare for the culminating assessment. The feedback in the self-evaluation use is more detailed than feedback from the culminating assessment.

The Biomass assessment is intended to be "standards-based." Despite all the activity surrounding standards, a gap remains between published lists of standards and sound systems for assessing students in terms of those standards. Standards descriptions often reflect inconsistent mixtures of what knowledge is valued, how it can be recognized, and activities for eliciting evidence. They are typically represented as discrete pieces of hierarchically organized text that do not reflect the integrated nature of the knowledge and skill they are meant to foster. A recent exception is the Next Generation Science Standards (NGSS; NGSS Lead States 2013). Its "performance expectations" are activities for learning or assessment that integrate content knowledge, scientific processes, and crosscutting themes such as cause-and-effect. Biomass lays out ECD models for generating such tasks, and making sense of the performances they evoke. The particular standards Biomass used are from a precursor of the NGSS, namely the National Science Education Standards (NRC 1996). Table 14.1 contains excerpts that illustrate crosscutting themes, scientific processes (with an emphasis on inquiry), and some of the life sciences topics addressed there.

The evidence-centered approach to standards-based assessment illustrated in Biomass moves from statements of standards in a content area, through claims about students' capabilities that the standards imply, to the kinds of evidence one would need to justify those claims, to the development of assessment activities that elicit such evidence, and finally to measurement models for synthesizing evidence from students' work in terms of standards-based claims. Rather than thinking at the level of individual tasks, we see tasks as instances of prototypical ways of getting evidence about aspects of knowledge that the standards bring to light. The ECD approach helps us recognize aspects of knowledge that are similar across content areas and skill levels, and craft schemas for obtaining evidence about such knowledge as it specializes to different particulars.

Table 14.1 A hierarchical textual representation of science standards

Unifying concepts and processes of science (All grades)
Systems, order, and organization
Evidence, models, and explanation
Constancy, change, and measurement
Evolution and equilibrium
Form and function
Science as inquiry (Grades 9–12)
Abilities necessary to do scientific inquiry
Identify questions and concepts that guide scientific investigation
Design and conduct scientific investigation
Use technology and mathematics to improve investigation and communication
Form and revise scientific explanations and models using logic and evidence
Recognize and analyze alternative explanations and models
Communicate and defend scientific argument
Understandings about scientific inquiry
Life Science (Grades 9–12)
Molecular basis of heredity
Theories/models
Chromosome theory of inheritance
Chromosome mapping
Base-pair complementarity
Chi-square model
Punnett squares

Excerpts from the National Science Education Standards (NRC 1996)

Science is a good domain to illustrate this approach. Science standards (e.g., Table 14.1) typically reflect both domain knowledge about the facts and theories of science and process knowledge about the scientific method. It is generally easier to test science facts (e.g., How many planets are there in the solar system?[1]) than the process knowledge (e.g., Design an experiment to provide a good test of this hypothesis). Building tasks that tackle the higher-order knowledge presents two challenges. First, ideally the students will be using this knowledge constructively; this means that the assessment system will need to extract evidence about the targeted knowledge from complex work products. Second, it is difficult to design a problem in which the process knowledge is used in a meaningful way without reference to domain knowledge; the assessment system must be able to untangle the various kinds

[1] Science facts also have a tendency to go out of date as our understanding of the world changes, e.g., the recent reclassification of Pluto from planet to dwarf planet. However, the process knowledge changes more slowly. Model-based reasoning and investigation will be important for a long time.

of evidence about various mixes of content knowledge, processes capabilities, and understanding of crosscutting or unifying themes.

14.2 Designing Biomass

To promote reuse and encourage freer thinking about the domain, ECD divides the assessment design process into three stages. The first stage, *Domain Analysis*, consists mainly of gathering and organizing extant knowledge about the domain (e.g., standards, cognitive models, previous assessments, and other key literature) and requirements for the assessment. The second stage, *Domain Modeling*, involves building a preliminary, less-detailed version of the conceptual assessment framework (CAF) that embodies the assessment argument but is not yet limited by the practical constraints of administering the test. It is often possible to work out important design issues with the lighter-weight models before the expense of building and calibrating tasks is incurred.

In typical projects, there is usually some working back and forth across these stages, such as what we learn in piloting—a light application of assessment delivery—can tell us where we need to revise models in the CAF, sharpen arguments in domain modeling, or gather more information, say in think-alouds, to flesh out areas in domain analysis that take on new importance. The ECD layers are not rigid steps in some waterfall development process, but rather a guide to distinct kinds of thinking that needs to be carried out to create an assessment, representations that are useful at different points, and connections across the design stages and different experts' roles. This section, then, describes the key results of the domain analysis and modeling for Biomass, and the following section describes the CAF.

The first steps in designing Biomass were to choose a subject matter domain and convene a panel of domain experts. Data gathering then proceeded with the substantive foundation: selecting pertinent educational standards, choosing illustrative topics within the subject matter domain, and defining relationships between standards and subject matter topic content. The focus then shifted to the claims we would want to make about students as a result of their performance on assessment tasks, what we would need to observe as evidence to support those claims, and the nature of assessment activities that would provide students the opportunity to produce that evidence—all in light of the affordances and constraints of an appropriate mode of assessment delivery.

14.2.1 Reconceiving Standards

Biology was chosen as the subject matter domain because several comprehensive sets of science standards had been developed (e.g., NRC 1996; AAAS 1994, which are consistent with, but not as fully integrated as, NGSS). Because these standards also reflect common themes that cut across all science subjects, biology provided a good context to demonstrate reusability and scalability of assessment design elements. The expert panel next identified a set of crosscutting themes to focus on, and then picked biology topics that would best illustrate them. The chosen themes were *unifying concepts* and *scientific inquiry* (Table 14.1). According to the NSES standards, these themes apply broadly across domains of science. They would provide a way for us to illustrate a task design approach that could be applied in many domains, using argument structures that would be instantiated with the models and problems of particular content areas. In Biomass, we focused on the way biological phenomena are studied across different *levels of organization* (e.g., the molecular, cellular, organism, and population levels) and *the use of models and evidence* to reason about and explain biological phenomena. The particular content topics chosen in which to illustrate these themes in assessment design were *transmission genetics* and *microevolution*. The experts developed detailed concept maps for these subdomains (see Fig. 14.1 for an example).

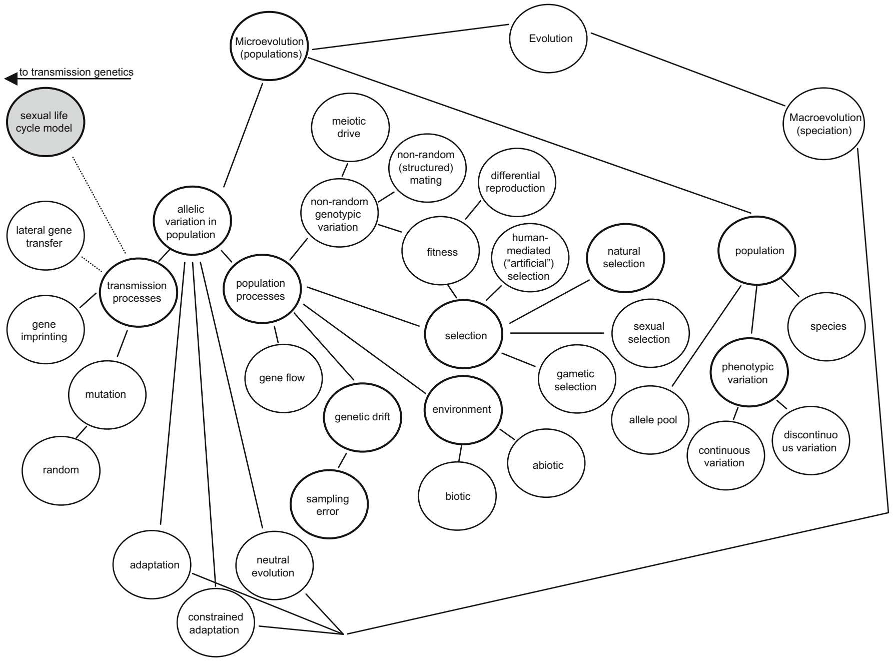


Fig. 14.1 A concept map for mechanisms of evolution
Reprinted with permission from ETS.

The expert panel suggested Fig. 14.2 as a way to show the relationships among unifying themes and science content in terms of the capabilities we want students to acquire.

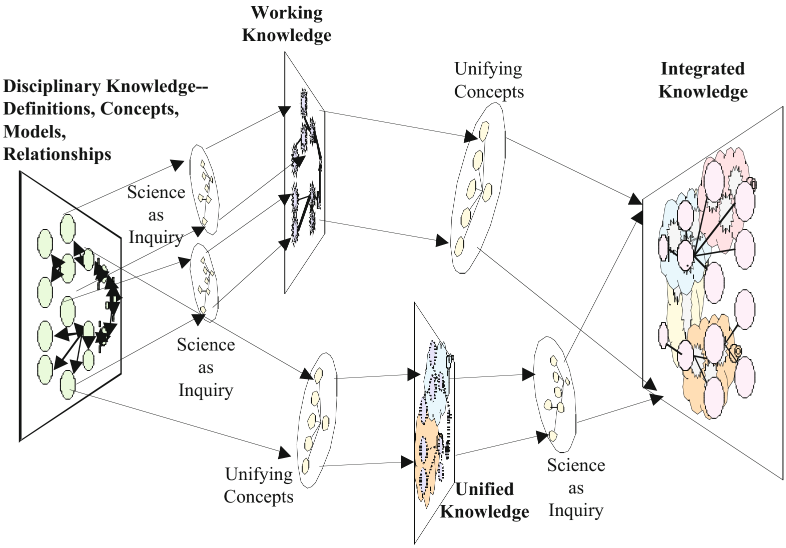


Fig. 14.2 Representation of a standards-based domain for assessment design

Planes represent classes of claims, while *circles*, *gears*, and *clouds* represent individual claims about Domain, Working, and Integrated Knowledge. Reprinted from Steinberg et al. (2003) with permission from The National Center for Research on Evaluation, Standards, and Student Testing (CRESST), UCLA.

At the left of Fig. 14.2 is Disciplinary, or Declarative, Knowledge: the definitions, models, and relationships in transmission genetics and microevolution. Two alternative elaborations of Disciplinary Knowledge appear to its right. Along the top, Disciplinary Knowledge is extended by understanding how to use it as the substantive grounding of inquiry, to produce what we will call Working Knowledge: the capability to use definitions, concepts, models, and relationships in inquiry, as when explaining a particular phenomenon in terms of underlying models or investigating the plausibility of an explanatory model. The bottom of the figure shows Disciplinary Knowledge extended by understanding how it relates to a particular unifying concept or process. For example, seeing cells through the unifying concept of form and function helps one understand how cellular structures facilitate cellular processes. The unifying concepts add structure and explanatory power to the myriad elements of disciplinary knowledge. The right side of Fig 14.2 portrays Integrated Knowledge, through which one can use models, evidence, and explanations from different topic areas and different levels of organization to address increasingly broader applied problems.

14.2.2 Defining Claims

Claims specify what one would want to be able to say about a student as the consequence of assessment. In Biomass, a claim addresses Disciplinary Knowledge, Working Knowledge, or Integrated Knowledge.

A large number of claims about Disciplinary Knowledge can be developed from instructional materials and concept maps in transmission genetics and microevolution. They address terms, concepts, and knowledge representations. Claims concerning Unified Knowledge address relationships among terms, concepts, and knowledge representations across levels or content areas. Claims can be cast at different levels of specificity. Specific claims are useful for guiding learning, while more encompassing claims that subsume more detailed ones can be sampled broadly for summative evaluation. As examples, the following are a Disciplinary Knowledge claim and a more specific claim that it encompasses:

C1: *The student understands the entities, events, and outcomes constituting the Mendelian model.*

C2: *The student can reexpress a verbal description of dominance relationship in terms of allele*[2] *notation.*

Claims about Working Knowledge involve Disciplinary Knowledge put to work in explaining situations, making predictions, or solving problems. An example of a Working Knowledge claim is

C3: *Given complete data, the student can use the data to evaluate a hypothesis about a situation involving a population-level model across time.*

Claims about Integrated Knowledge also involve Disciplinary Knowledge as it is put to work in explaining situations or solving problems, but additionally involve connections across different levels or content areas; for example,

C4: *The student can reason through the Mendelian, sexual life cycle, natural selection, and genetic drift models for prediction in a situation involving the cellular, organism, and/or population level(s) across transmission genetics and mechanisms of evolution and across time.*

[2] An allele is a short stretch of DNA on a particular location on a chromosome that (along with its pair on the matched chromosome) controls the expression of a particular genetic trait. Allele notation uses letters to represent the various alleles and a pair of letters to represent a genotype. For Mendel's pea experiment, the letter "T" could be used to represent the allele for tall plants, and the letter "t" could be used for the allele for short plants. In this configuration, the possible genotypes would be "TT", "Tt," and "tt." Because the tall allele is dominant, the first two result in tall phenotypes and the third results in shorter plants.

As the expert panel saw it, Working Knowledge and Integrated Knowledge always entail some Disciplinary Knowledge. Inquiry is always inquiry about something, and model-based reasoning is always carried out with some particular model(s).

14.2.3 Defining Evidence

Once we had established what we wanted to assess in the form of claims, we set about defining evidence that would be needed to support them. Our attention focused on Working and Integrated Knowledge, because methods for assessing Disciplinary Knowledge are familiar in the content areas being addressed. Methods for assessing inquiry are less familiar, and the computer-based platform offered interesting possibilities for obtaining direct evidence about inquiry processes. Evidence in the form of potential observations related to each claim was first considered independent of specific tasks. As an example, Table 14.2 lists some of the observations the experts thought would support the claim that a student could design and conduct a scientific investigation in a given area, based on research by Stewart and Hafner (1994), for example, on model-based reasoning and White and Frederiksen (1998) on inquiry.[3]

Table 14.2 Potential observations related to scientific investigation

Recognition of need to obtain additional data
Efficacious specification of appropriate methodology(s) for gathering data
Adequacy of model testing
Efficacious specification of methodology(s) for testing model
Identification of outcomes of model testing that bear on current hypothesis (i.e., confirming/disconfirming evidence)
Association of anomalous data with relevant aspect(s) of relevant model(s)
Impasse specified in terms of data/model mismatch
Accuracy of model revisions

Three features of the observations listed in Table 14.2 are worth mentioning. First, they are cast broadly enough to apply to inquiry processes carried out in many scientific domains and across educational levels. They can therefore be used to guide task design beyond the content areas Biomass addresses.

[3] Follow-on work from Biomass led to a National Science Foundation-supported project called Principled Assessment Design for Inquiry (PADI) (Mislevy and Haertel 2006). The PADI project created “design patterns” that organize kinds of claims, observations, work products, and task features for assessing students’ capabilities with science processes such as model-based reasoning and systems thinking (Cheng et al. 2010; Mislevy et al. 2009).

Second, because the observations can be applied across content areas, any specific instantiation will need to further specify the Disciplinary Knowledge and inquiry techniques that are involved. A task providing evidence about inquiry will necessarily depend on the required disciplinary knowledge. In Sect. 14.3 we will see implications for building the proficiency and evidence models and interpreting the proficiency variables.

Third, these observations focus on the nature of the thinking they reveal, not on the specific form of data. One can go a long way in defining evidence before specifying exactly how to get it. Thinking about the relationship between what we want to observe and the way knowledge is manifested before specifying a particular type of task opens thinking about what can and should be considered as evidence, and ways in which one might acquire it. Evidence of a particular aspect of knowledge or kind of thinking can usually be obtained in many ways, such as mutliple-choice questions, constructive exercises, open-ended verbal explanations, hands-on laboratory work, or any number of other methods. Each approach has its own costs and benefits, its own advantages and disadvantages. Determining what to use in a given assessment context depends on the particular constraints, resources, and purposes for that assessment. The form of the data is secondary to its evidentiary import.

14.3 The Biomass Conceptual Assessment Framework

The Biomass Conceptual Assessment Framework builds on the design rationale described above to provide blueprints for the operational elements of the assessment, namely the proficiency model, task models, and evidence models (Chaps. 2 and 12). We focus now on the way Bayes nets were constructed, to model students in terms of variables that reflect standards-based claims.

14.3.1 The Proficiency Model

Figure 14.3 shows the Biomass Proficiency Model. The full model contains 15 variables. Each node represents an aspect of knowledge or skill about which we want to accumulate evidence. These variables are derived from the conceptual representation of knowledge in the domain represented in Fig. 14.2. There are nodes that concern disciplinary knowledge, working knowledge, and integrated knowledge. Each of the claims the expert panel articulated is associated with one or more of these variables.

When the numbers are added to this graph (next chapter) the proficiency model defines a probability distribution over the possible *proficiency profiles*—assignments of values to each proficiency variable. This probability distribution represents the population distribution of proficiency profiles in the targeted population. Using the algorithms discussed in Chaps. 5 and 13, this probability distribution will be updated for specific students using the evidence accumulated in the assessment, yielding a distribution over proficiency profiles specific to that student.

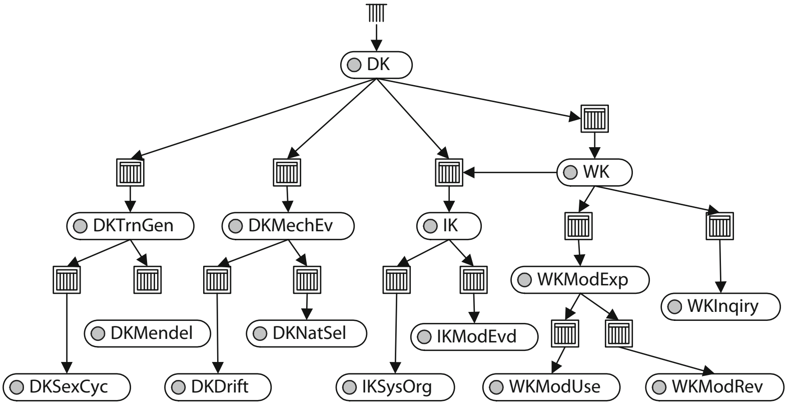


Fig. 14.3 The biomass proficiency model
Reprinted with permission from ETS.

Each possible proficiency profile has a number of claims which it does and does not support. In a standards-based assessment many claims correspond to one or more standards being met, thus the evidence gathered to infer the proficiency profile of a student can be used to infer the claims supported and hence the standards that are met.

Whether or not a claim is supported may depend on the value of several proficiency variables; however, many claims are supported by just one of the proficiency variables. For example, Claim C1 states that "The student understands the entities, events, and outcomes constituting the Mendelian model." Claim C1 is associated with the proficiency variable *DKMendel*, or Disciplinary Knowledge about the Mendelian Model. Like all the proficiency variables in Biomass, *DKMendel* can take three ordered values, `high`, `medium`, and `low`. They are interpreted as claims that a student has High, Medium, or Low proficiency respectively for the competence represented at the highest level by the associated standard.

Note that the effective meaning of `high`, `medium`, and `low` proficiency for *DKMendel* accrues from the kinds of evidence that has been defined for that purpose; that is, the space of tasks and observations that can be generated from the analysis of potential evidence as described in Sect. 14.2.3.

(A development that has occurred since the Biomass project is the use of learning progressions (Alonzo and Gotwals 2012; Corcoran et al. 2009), as discussed in Sect. 12.2.1. A learning progression is a sequence of increasingly sophisticated understandings or capabilities that learners typically move through in some domain area. They are often marked by the kinds of things learners can do in situations with various features. This conception is neatly matched to ECD assessment design. In simple cases, levels of a learning progression correspond directly to values of an ordered-state variable in a Bayes net proficiency model with theory-defined meanings, and the features of performances and tasks correspond to evidence-model and task-model variables (West et al. 2012; Zalles et al. 2010). In more complicated cases, the levels of a more coarsely defined learning progression are configurations of values of finer-grained proficiency variables (West et al. 2012; Wilson 2009).)

When multiple aspects of skill or knowledge are required in combination, a claim can be modeled as depending on the values of more than one proficiency variable. The compensatory, conjunctive, and disjunctive relationships among proficiency variables described in Chaps. 6 and 8 are prototypical patterns for the relationships between claims and proficiency variables. Section 14.3.4 illustrates how these design patterns were used in Biomass to model performance in tasks that depend jointly on disciplinary and inquiry aspects of knowledge. An example of a claim that requires a conjunction of proficiencies is this:

C5: *Given complete data, the student can use the data to evaluate a hypothesis about a situation involving a population-level model across time.*

Claim C5 is associated with both Disciplinary Knowledge about Mechanisms of Evolution (*DKMechEv*) and Integrated Knowledge about Models and Evidence (*IKModEvd*). Support for C5 is represented as the conjunctive combination of these two proficiency variables, that is, high values on both.

The Biomass proficiency model is rather sparse, in terms of the number of variables it contains. For example, Claim C2 concerning allele representation is defined at a finer grain size than the variables in this proficiency model. It was used to guide the definition of evidence and construction of tasks; the evidence afforded by such tasks was one portion of the body of evidence bearing on *DKMendel*, the proficiency variable that corresponds to the broader Claim C1 that encompasses C2. The basic structure of the proficiency model could be elaborated with additional proficiency variables at finer grain sizes or for additional topics in a manner discussed below. However, those additional variables will require evidence (observables from multiple tasks) to support them. Even for an assessment intended to be embedded in normal classroom activity, the amount of time that can be spent gathering evidence is limited. Thus, assessment design always requires trading-off grain size in the proficiency model and time spent gathering evidence.

The Biomass proficiency model shows a tripartite hierarchical structure of Disciplinary Knowledge, Working Knowledge, and Integrated Knowledge. The three corresponding variables at the highest level of the hierarchy are abbreviated *DK*, *WK*, and *IK* respectively. The probability distribution for a variable at this highest level summarizes evidence across all the aspects of the given kind of knowledge addressed in the Biomass, corresponding to broadly cast claims. This tripartite structure is common to many branches of science (either different subjects within biology, such as botany or cell biology, or different sciences such as physics or psychology). The model structure supports ready reuse at this general level.

Finer-grained proficiency variables can be added to the basic model structure as children of any higher level proficiency variables. Here *DK* has two children, for the subareas addressed in the prototype: transmission genetics (*DKTrnGen*) and mechanisms of microevolution (*DKMechEv*). If Biomass were extended to additional areas of a biology course, additional disciplinary knowledge proficiency variables would be added at this level as children of *DK*. *DKTrnGen* itself has two children, concerning the sexual cycle (*DKSexCyc*) and the Mendelian model (*DKMendel*). *DKMechEv* similarly has two children, namely genetic drift (*DKDrift*) and natural selection (*DKNatSel*).

Working Knowledge (*WK*) also has children that represent the subareas of inquiry (*WKInqry*) and models and explanation (*WKModExp*), and models and explanation itself has two children concerning model use (*WKModUse*) and model revision (*WKModRev*). Integrated knowledge (*IK*) has two children, concerning systems and organization (*IKSysOrg*) and models and evidence (*IKModEv*).

The choice of grain size depends on the intended use of an assessment, since finer-grained proficiency variables are needed to support finer-grained claims. In the use of Biomass as a culminating assessment, for example, reports are provided at the level exemplified by *DKTrnGen*. In its use as a learning assessment, feedback is provided at the level exemplified by *DKSexCyc* and *DKNatSel*. This is because grain size trades off with accuracy. A student's proficiencies at higher levels in the hierarchy are based on more evidence—i.e., more task performances and resulting observable variables—than are proficiencies in subdomains. The same phenomenon applies more generally to subscores in assessments, whenever they are based on only portions of the evidence that go into overall scores.

Figure 14.3 shows children at each level of the tree as conditionally independent given their parents, the higher-level proficiencies. With this structure, additional subtopics can be added within any level of the tree without affecting the structure of the network elsewhere. The form of the conditional distributions among proficiency variables and parameters for these distributions will be detailed in Sect. 14.4.

As noted above, both integrated knowledge and working knowledge are conceived as something a student knows or can do *with particular disciplinary knowledge*. Section 14.3.4 will detail how this relationship is effected in evidence models to model performance. To anticipate, relevant aspects of both disciplinary knowledge and, say, working knowledge are required conjunctively for good performance. This modeling choice gives rise to a *conditional interpretation* of working knowledge: To be high on *WK* means that a student can carry out inquiry if she also possesses requisite levels of disciplinary knowledge for the situation at hand.

The interpretation of *IK* is also conditional. To be high on *IK* means a student is likely to do well reasoning through models at different scales or from different areas *if* she *also* has the requisite levels of disciplinary knowledge in those models as well.

In the Bayesian framework, the proficiency model (and student-specific scoring model) will contain a probability distribution over all of the possible proficiency profiles. The final step in defining the proficiency model is deciding on the summary statistics of that posterior distribution that will be reported on the final score report. Biomass used a fairly conventional choice of reporting the marginal distributions of all of the proficiency variables, providing the probabilities that a given subject is at or above the `Advanced` (`high`) and `Basic`(`medium`) levels. Figure 14.4 shows a sample score report for the classroom learning assessment. The proficiency variables are briefly defined on the form in terms of the claims, while more complete definitions are available in the interpretation guide. The score report for the culminating assessment is similar, but would not have the finer detailed variables. (The two assessments actually used the same proficiency model, only the reporting rules were different. In the case of the culminating assessment only a selected subset of variables are reported.)

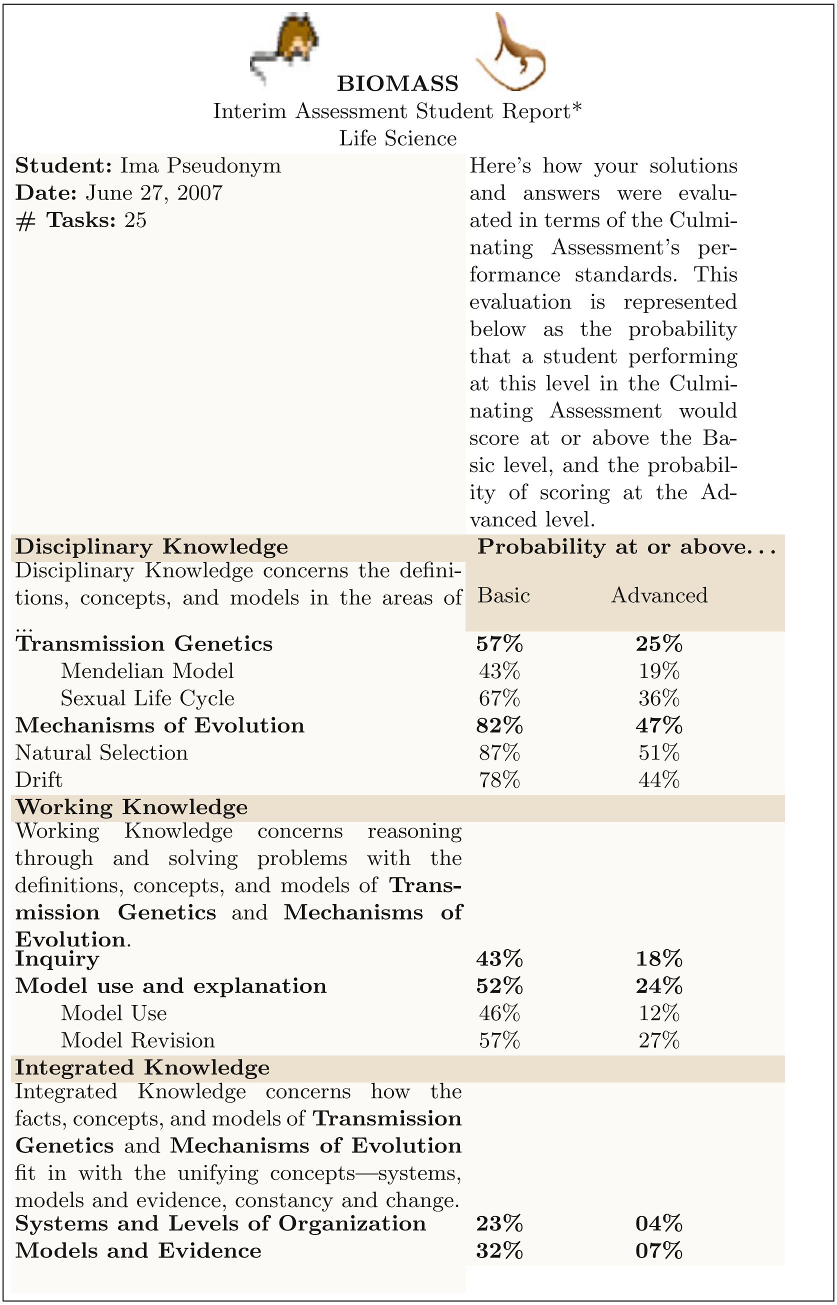
The score report is typically all a test user sees of the proficiency model. When seeking input from potential users about design options, having score reports that illustrate the implications of the design choices will enable the users to provide meaningful feedback. We have found it to be good practice to show a prospective score report to potential users early in the development process. This will provide valuable information about whether or not the proficiency variables and scores derived from them provide useful and actionable information to the end users. (Recall the value of information calculations in Example 4.1.)

In Biomass, the primary test user is likely to be the classroom teacher. The teacher is not just concerned with the proficiency profile of a single student, but of all the students in the classroom. Fortunately, averaging the marginal distributions across students provides a good description of the class average. Almond et al. (2009a) explores a number of ways of presenting information from Bayesian networks for a class full of students.

14.3.2 The Assembly Model

An Assembly Model specifies the rationale by which tasks are combined into an assessment in a fixed test or the algorithm by which they are sequentially selected in an adaptive test. The considerations that enter the specification include content coverage, time constraints, and amount of information about particular proficiency variables (Sect. 7.4). There are actually two assembly models for the Biomass prototype, one for the learning mode and the other for culminating assessment. In both cases, the desire to assess Working Knowledge in the form of investigations imposed the need to present segments of investigations consisting of multiple steps and providing multiple observable variables.

For the learning mode, the goal of presenting feedback in terms of proficiency estimates at a more detailed grain size than the culminating assessment,



BIOMASS

Interim Assessment Student Report*

Life Science

Student: Ima Pseudonym
Date: June 27, 2007
# Tasks: 25

Here's how your solutions and answers were evaluated in terms of the Culminating Assessment's performance standards. This evaluation is represented below as the probability that a student performing at this level in the Culminating Assessment would score at or above the Basic level, and the probability of scoring at the Advanced level.

Disciplinary Knowledge	**Probability at or above...**	
Disciplinary Knowledge concerns the definitions, concepts, and models in the areas of ...	Basic	Advanced
Transmission Genetics	**57%**	**25%**
Mendelian Model	43%	19%
Sexual Life Cycle	67%	36%
Mechanisms of Evolution	**82%**	**47%**
Natural Selection	87%	51%
Drift	78%	44%
Working Knowledge		
Working Knowledge concerns reasoning through and solving problems with the definitions, concepts, and models of **Transmission Genetics** and **Mechanisms of Evolution**.		
Inquiry	**43%**	**18%**
Model use and explanation	**52%**	**24%**
Model Use	46%	12%
Model Revision	57%	27%
Integrated Knowledge		
Integrated Knowledge concerns how the facts, concepts, and models of **Transmission Genetics** and **Mechanisms of Evolution** fit in with the unifying concepts—systems, models and evidence, constancy and change.		
Systems and Levels of Organization	**23%**	**04%**
Models and Evidence	**32%**	**07%**

Fig. 14.4 Biomass: a sample score report from the Biomass classroom assessment Reprinted with permission from ETS.

coupled with the requirement to be able to use the investigations freely over several class periods if desired, led to the full sequence of 17 segments of an investigation. This meant that multiple observable variables were provided not only for higher-level nodes such as *DK*, *WK*, and *IK*, but also for the most detailed nodes such as *WKModUse* and model revision *WKModRev*. Information is obtained from at least five observable variables for each reported proficiency. In conjunction with task-level textual feedback, this fine level of detail helps the student go back over, and repeat if desired, parts of tasks, or study those aspects of knowledge outside the task.

For the culminating mode, testing time is a more pressing constraint. Two class periods, the time typical for a final examination or an Advanced Placement test, does not permit the full experience of extended investigation. The culminating tasks thus focus on only segments of an investigation, as noted above. Model revision may be assessed by providing a model that does not accord with data in some way, rather than having the student work through repeated cycles of proposing, testing, and revising models. Further, because less information can be gathered, the selection of tasks for the culminating assessment spans the domain areas more broadly but less deeply. Samples of aspects of knowledge involved in *DK*, *WK*, and *IK* are obtained in a balance across topics and aspects of integrated and working knowledge. Higher-level variables in the proficiency model are used as parents in the Bayes net for the culminating test, so reports are at a coarser grain size. Again at least five observables support every variable reported, but they are now sampled across a broader spread of the content area.

A significant challenge in Biomass is to represent all the steps in the scientific method in a coherent fashion. This was not possible in the limited time allowed to the culminating assessment, but the learning assessment mode could spend some additional time to tell a more complete story. In Biomass, the tasks were grouped into two extended scenarios: an investigation of transmission genetics using a population of field mice (code named "mice"), and an investigation of microevolution and genetics using a population of lizards (code named "lizard").

Each scenario was broken up into a number of *segments* based on steps of the investigation. For example, the mice scenario had the segments shown in Table 14.3. The lizard scenario had 11 segments. Both scenarios had a special "Segment 0" that provided background about the scenario, but did not have any scored activities. As the classroom learning use of biomass was envisaged to run over many days, the students were free to tackle the segments in any order and to go back and repeat previous segments.

A key challenge in designing simulation-based assessments (and Biomass is essentially a worked-out simulation) is what to do if a student gets badly off track. Making a key mistake early in the simulation means that they are unlikely to be able to perform well in later stages of the task. Once the student has gotten off track, further interaction with the simulator is unlikely

Table 14.3 Segments in the Biomass "mice" scenario

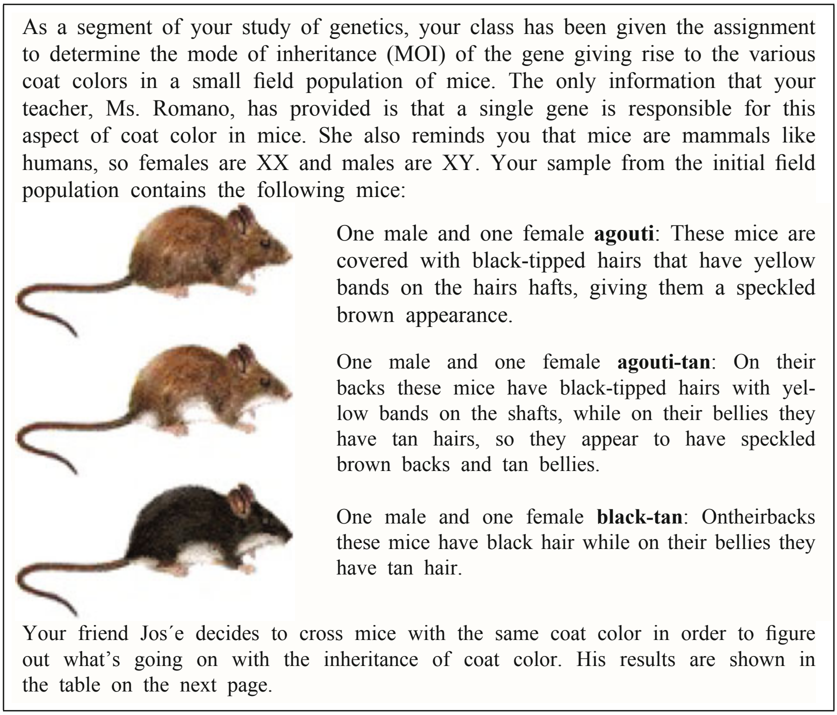
0	Scenario
1	Formalize H_0
2	Select verification method
3	Cross expectations/Punnett square
4	Do data support H_0?
5	Recognize disconfirming data
6	Explain disconfirming data
7	Hint for new H_0
8	Build new H_0
9	Given new H_0, what next?
10	Select crosses for testing new H_0
11	Cross expectations/Punnett square
12	Given cross results, what next?
13	Interpret chi-squared results
14	Select final confirming crosses
15	Explain final confirming crosses
16	Interpret final confirming crosses
17	Connect genetics and cell/life cycles

to provide additional information of substantial value. In this respect, a large simulation is likely to provide less information than a series of smaller minisimulations.

To get around this problem, Biomass introduced a surrogate investigator, José. Rather than performing the experiments directly, the student's role is to advise José. As José always makes the right choice (sometimes after discussion, advice, or false starts), the student is always on track at the start of the segment. This eliminates one source of potential problem. As the student has the option of changing their responses after seeing initial feedback from each task, being able to jump ahead and see what José did actually provides little scoring advantage. Besides, the scores from the learning assessment are not appropriate for high-stakes purposes, so there is little incentive for cheating.

Figure 14.5 shows the initial scenario from the "mice" scenario. All of the subsequent tasks depend on this initial data as well as data gathered in later experiments. The student is free to return to earlier segments to help recall details about earlier parts of the scenario.

During the development of Biomass, there was a vigorous discussion about what was a "task." The original conception used with the experts was that each of the extended scenarios was a "task." The implementation team found this extended notion of task too clumsy to work with. The definition of task they decided on was the unit of information passed around the four-process architecture; this corresponded to the segments in Table 14.3. Thus, all four activities from the first segment (described in Sect. 14.3.3) were called a single "task" in the original Biomass design. Under this choice, several evidence models were required to score independent parts of the first task. An alter-



As a segment of your study of genetics, your class has been given the assignment to determine the mode of inheritance (MOI) of the gene giving rise to the various coat colors in a small field population of mice. The only information that your teacher, Ms. Romano, has provided is that a single gene is responsible for this aspect of coat color in mice. She also reminds you that mice are mammals like humans, so females are XX and males are XY. Your sample from the initial field population contains the following mice:

One male and one female **agouti**: These mice are covered with black-tipped hairs that have yellow bands on the hairs hafts, giving them a speckled brown appearance.

One male and one female **agouti-tan**: On their backs these mice have black-tipped hairs with yellow bands on the shafts, while on their bellies they have tan hairs, so they appear to have speckled brown backs and tan bellies.

One male and one female **black-tan**: Ontheirbacks these mice have black hair while on their bellies they have tan hair.

Your friend Jos´e decides to cross mice with the same coat color in order to figure out what's going on with the inheritance of coat color. His results are shown in the table on the next page.

Fig. 14.5 Biomass: the introductory screen
Reprinted with permission from ETS.

native we could have chosen would be to designate as a "task" an activity that produces a group of work products that are scored together. As with all design decisions, there are advantages and disadvantages to drawing the line in different places, and such design decisions can be revisited for each project.

14.3.3 Task Models

Although each task model defines a collection of possible tasks, in Biomass there was only one realization of each task model. Even so, thinking about all of the possible tasks helps the design team identify those elements of the task that are important for the evidence that will be collected. Furthermore, generalizing from a specific instance to the general class is generally easier for the design team than trying to build the task model without reference to any specific instance. Thus, it is a common practice for the task model and first task from that model to be developed simultaneously.

Task models describe situations in which students will perform work that produces evidence about their proficiencies. Section 14.2.2 described the kinds of claims that Biomass targets. Section 14.2.3 described the kinds of observations we need to make in order to ground them. Once we had defined a collection of observations at a higher level of abstraction, we had to start developing situations that provided students the opportunity to provide such evidence, i.e., tasks. Biomass task models lay out the stimulus materials, tools, directives, and work products that comprise tasks, and the task model variables used to express the key features of a given task.

In assessment design, identifying the salient knowledge representations for a given domain helps us think about how information is conveyed both to and from the student (Gitomer and Steinberg 1999; Mislevy et al. 2010). When information is being conveyed to a student, we think about the representations we need to use in presenting material to the student. When information is coming from the student, we think about representations the student can use to create a work product. The experts thus turned their attention to the representational forms by which information about the targeted topics would typically be communicated within a learning environment.

In biology, specifically within transmission genetics and microevolution, there are conventional forms for conveying information: Punnett Squares, phenotypic distributions, allele symbols, pedigree and chromosome diagrams, and population tables, to name a few. These play central roles in Biomass tasks; in particular, a fair number of tasks (including Task 1 below) called for the student to reexpress information presented in one form in a different form, or to distill and interpret what she learned in more open-ended exploration in the form of a representation actually used in the domain.

The data-driven nature of working in these areas of biology leads to tasks that emphasize the manipulation and interpretation of data, as Working Knowledge. The number of different knowledge representations necessary for conveying information and solving problems calls for working across multiple knowledge representations, providing evidence for Integrated Knowledge claims. Having students work with or create representations will allow us to produce evaluations for observations that bear on DK through the Mendelian Model aspect of proficiency, i.e., *DKMendel*, and instances of *WK* and *IK* that require knowledge of the Mendelian model.

The experts also identified a number of ways to communicate about investigative methodology in the contexts of transmission genetics and microevolution. At a high level, there are the steps in the hypothetico-deductive framework. At a lower level, there are the rules governing the selection of test populations and individuals within them. Having students work with these rules within this framework, using whichever knowledge representations are appropriate for presenting information and capturing work products (Table 14.4), provides performances we can evaluate for observables such as Efficacious Methodology.

To illustrate these ideas, we show the initial screens of Biomass's experimental investigation in transmission genetics, which was written from the point of view of the hypothetical student José. The emphasis was on use

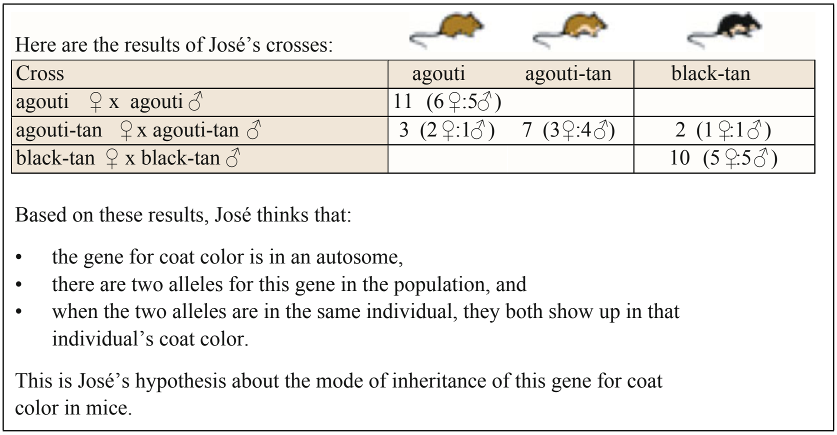
Table 14.4 Connecting knowledge representations with investigation steps

Methodology	Associated knowledge representation
FORMULATE H_0	Hypothesis expressed in standard form or alternative form
GENERATE DATA	Population summary cross table; cross choice table
ANALYZE DATA	Hypothesis expressed in standard form or alternative form; Population summary cross table; chi-sq table
ACCEPT H_0*?*	Hypothesis expressed in standard form or alternative form; Population summary cross/H_0 connections table

and revision of the Mendelian Model to determine the mode of inheritance for coat color in agouti mice. The problem was constrained to a single trait controlled by three different forms (alleles) of a single gene. Figure 14.5 sets the stage, by introducing the mice, the problem, and José's initial crosses. Figure 14.6 shows the results of the crosses and José's hypothesis about the mode of inheritance. Figure 14.7 asks the student to represent the hypothesis in an alternative representational form. Responses to tasks that require a student to translate a textual representation such as that of Fig. 14.6 to an allele representation such as Fig. 14.7 are just the kind of evidence needed to support Claim C2, hence the more encompassing Claim C1. The evidence will eventually be reflected in the probability distribution for *DKMendel*, the proficiency variable associated with C1.

The set of screens illustrated by these figures is a specific task generated from a task model. The task model includes variables specifying the number and nature of the representational forms to be made available to the student for expressing whatever the hypothesis happens to be, as well as variables specifying the elements of the hypothesis itself. For example, which representational forms will be used? Which of the four kinds of dominance relationships will underlie the problem? A task model also describes operational task presentation requirements—in this case a Web-based drag-and-drop capability for filling in a mode of inheritance table like the one shown in Fig. 14.7.

The highest-level attributes of a task model delineate its purpose, domain, audience, platform, and feedback options. At the Claim level, features specifying the type of knowledge, domain topics, nature, and number of models were described. The next lower level starts to shape specific tasks more directly by specifying the general form (e.g., scenario) in which individual activities appear, the nature of help and guidance, the type of activity to be carried out (e.g., field investigation), constraints on that activity (e.g., population sizes, nature of the "field") and additional content specification (e.g., organism). At this level, a task developer could create a task concerning the mode of inheritance of peas or imaginary dragons, thus using a different organism and



Here are the results of José's crosses:

Cross	agouti	agouti-tan	black-tan
agouti ♀ x agouti ♂	11 (6♀:5♂)		
agouti-tan ♀x agouti-tan ♂	3 (2♀:1♂)	7 (3♀:4♂)	2 (1♀:1♂)
black-tan ♀ x black-tan ♂			10 (5♀:5♂)

Based on these results, José thinks that:

- the gene for coat color is in an autosome,
- there are two alleles for this gene in the population, and
- when the two alleles are in the same individual, they both show up in that individual's coat color.

This is José's hypothesis about the mode of inheritance of this gene for coat color in mice.

Fig. 14.6 Biomass: background for first task
Reprinted from Mislevy et al. (2002a) with permission from The National Center for Research on Evaluation, Standards, and Student Testing (CRESST), UCLA.

mode of inheritance but the same representational forms and directives. Task model variables at the individual task level address specific features of knowledge representations that are employed for problem, reference, or response data. At this lowest (i.e., most specific) level, a task developer could create an alternative task using the same context of mice, but with different features or underlying modes of inheritance.

In addition to filling out the mode of inheritance table, the first segment of the Mice investigation presented three additional tasks[4] to a student. In both cases, information was presented using certain knowledge representations, directives were specified, and the student would respond by adding information to the knowledge representation to produce a work product.

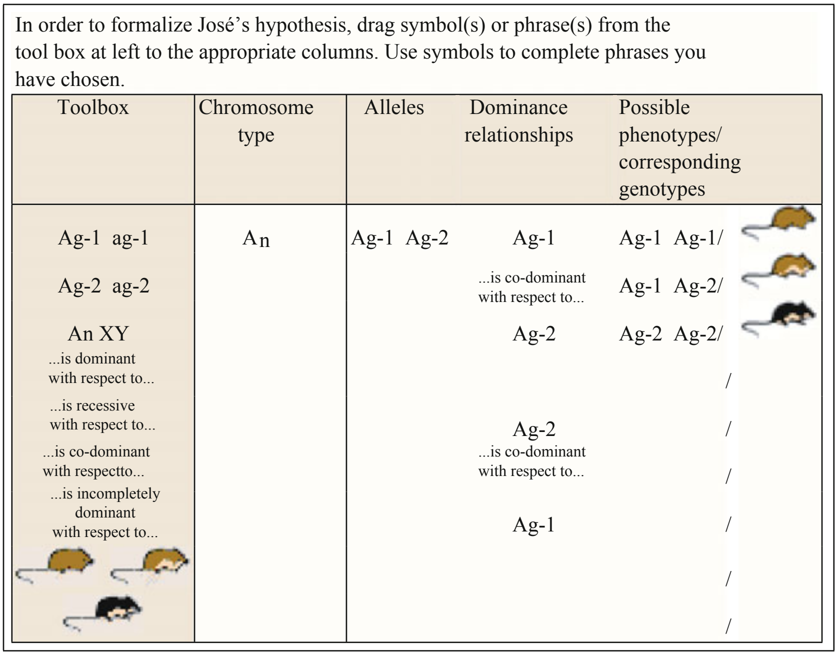
Figure 14.8 shows the first of these, the Population Attribute table. Its purpose is to obtain evidence bearing on the following Working Knowledge claim:

C10: Given incomplete data and data collection resulting in anomalous data and/or one or more deficient models can generate data to explore natural phenomena at the population level(s) across mechanisms of evolution and across time,

where "to explore natural phenomena" with regard to model revision is to

(a) recognize need for revision of models
(b) reason through and revise models

[4] For the purposes of the four-process model, these four tasks were grouped into a single "task set." The various evidence identification and evidence accumulation processes then pulled out the work products and observables from the various tasks for their separate analyses.



In order to formalize José's hypothesis, drag symbol(s) or phrase(s) from the tool box at left to the appropriate columns. Use symbols to complete phrases you have chosen.

Toolbox	Chromosome type	Alleles	Dominance relationships	Possible phenotypes/ corresponding genotypes
Ag-1 ag-1	An	Ag-1 Ag-2	Ag-1	Ag-1 Ag-1/
Ag-2 ag-2			...is co-dominant with respect to...	Ag-1 Ag-2/
An XY			Ag-2	Ag-2 Ag-2/
...is dominant with respect to...				/
...is recessive with respect to...			Ag-2	/
...is co-dominant with respectto...			...is co-dominant with respect to...	/
...is incompletely dominant with respect to...			Ag-1	/
				/
				/

Fig. 14.7 Biomass: first task is to complete a table for allele representation of mode of inheritance
When the task is presented, only the icons in the toolbox are present. The student drags *selected icons* to appropriate places in the *four columns to the right* in such a way as to describe the mode of inheritance of hair color. Reprinted from Mislevy et al. (2002a) with permission from The National Center for Research on Evaluation, Standards, & Student Testing (CRESST), UCLA.

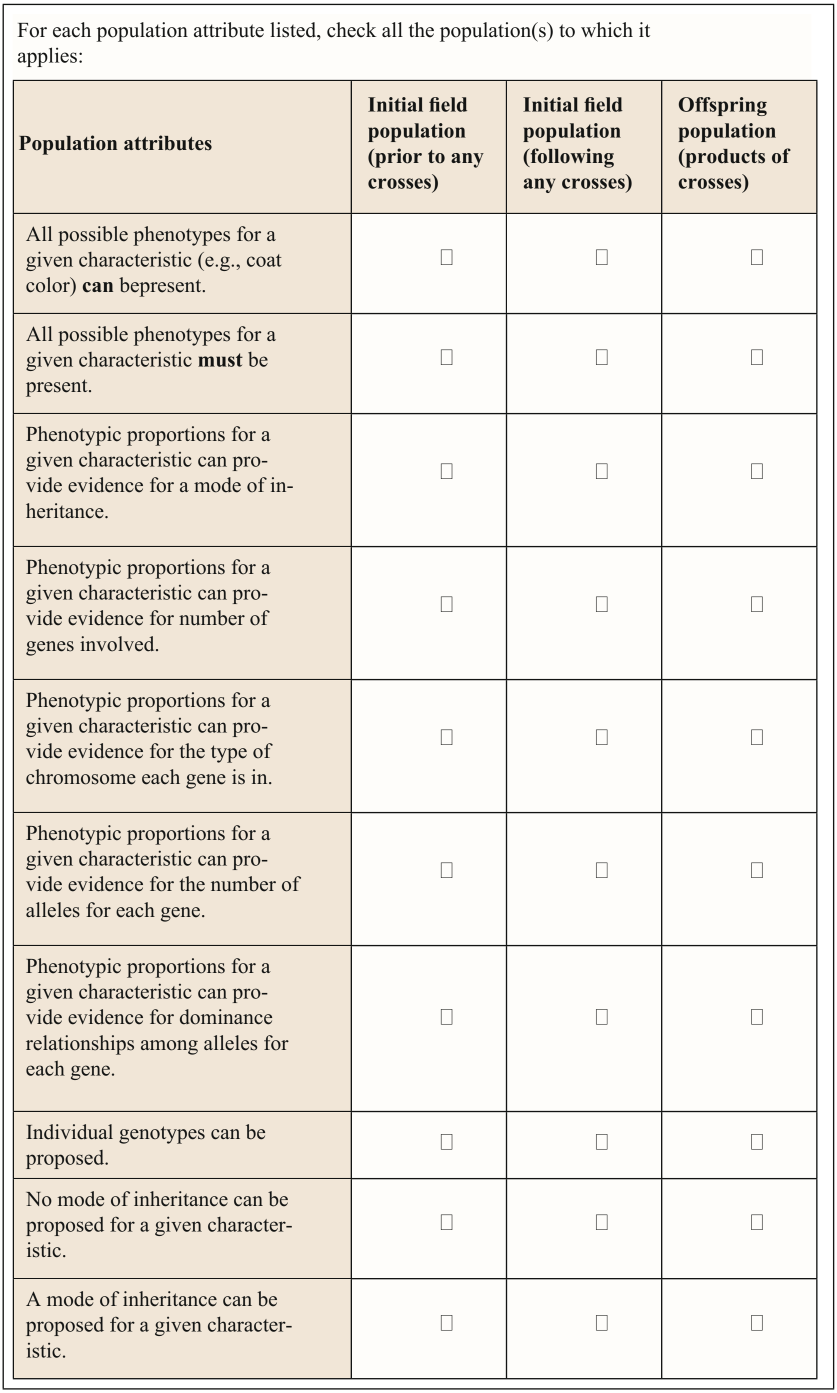
(c) recognize need for revision of and revise models

for the purpose of explanation, prediction, and/or model evaluation and comparison.

Two additional tasks were presented with the initial segment. The third task was a series of three multiple-choice questions designed to check whether the student understood the basic parts of the Mendelian model used in the first section. The last task was a special two-part question (Fig. 14.9) asking the student about the next logical steps in the hypothetico-deductive framework.

The development of the task models and tasks in both the transmission genetics (mice) and microevolution (lizard) scenarios was similar. In addition to the stimulus material for the task, task specific feedback was developed for each task. This feedback material was customized based on information from the evidence models.

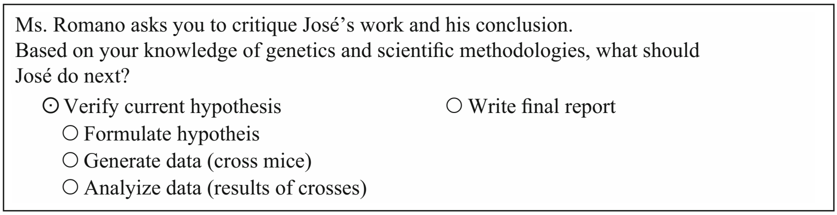
The task models for the culminating assessment are more focused and less extended investigations, as would suit an assessment setting with time



For each population attribute listed, check all the population(s) to which it applies:

Population attributes	**Initial field population (prior to any crosses)**	**Initial field population (following any crosses)**	**Offspring population (products of crosses)**
All possible phenotypes for a given characteristic (e.g., coat color) **can** bepresent.	□	□	□
All possible phenotypes for a given characteristic **must** be present.	□	□	□
Phenotypic proportions for a given characteristic can provide evidence for a mode of inheritance.	□	□	□
Phenotypic proportions for a given characteristic can provide evidence for number of genes involved.	□	□	□
Phenotypic proportions for a given characteristic can provide evidence for the type of chromosome each gene is in.	□	□	□
Phenotypic proportions for a given characteristic can provide evidence for the number of alleles for each gene.	□	□	□
Phenotypic proportions for a given characteristic can provide evidence for dominance relationships among alleles for each gene.	□	□	□
Individual genotypes can be proposed.	□	□	□
No mode of inheritance can be proposed for a given characteristic.	□	□	□
A mode of inheritance can be proposed for a given characteristic.	□	□	□

Fig. 14.8 Biomass: second task, population attribute table
Reprinted with permission from ETS.



Ms. Romano asks you to critique José's work and his conclusion.
Based on your knowledge of genetics and scientific methodologies, what should José do next?

⊙ Verify current hypothesis
○ Formulate hypotheis
○ Generate data (cross mice)
○ Analyze data (results of crosses)

○ Write final report

Fig. 14.9 Biomass: fourth task, what to do next?

The second set of choices appears only after the student selects the "verify current hypothesis" option. Reprinted with permission from ETS.

constraints and individual work. But the same family of observable variables, bearing on the same aspects of proficiency, appear in both the learning and culminating task models. This relationship between learning and culminating tasks allows students to become familiar with interfaces, knowledge representations, and expectations for evaluation during the course of study, so these crucial components of complex tasks will not "drop in from the sky" in the culminating assessment.

Altogether, four multistage investigative scenarios (two for classroom use, two for the culminating assessment) were developed, each consisting of a sequence of segments that a student would work through in the course of the larger task. Each segment presented information about results from any previous segments that were needed in the current segment, in order to reduce dependencies across segments. As described in the following section, however, dependencies did occur within segments.

14.3.4 Evidence Models

We complete our tour of the Biomass CAF with a look at the evidence models. Recall that the evidence models serves as a bridge between the work product defined in the task model and the proficiency variables defined in the proficiency model. The center support for that bridge is a collection of observable outcome variables that are the center point of the evidence model.

The evidence model bridge has two spans:

- The *rules of evidence* that describe how to set the observable outcome variables based on the work product produced by the student.
- The *Bayes net fragments* (i.e., evidence model fragments, or EMFs) that describes how the observables relate to the proficiency variables.[5]

[5] More generally, the ECD framework specifies a statistical relationship between the proficiency and observable variables. Some informal assessments do not implement this model. When it is implemented, Bayes nets are one way to do it, and we like their flexibility, but true-score and latent-variable psychometric models such as

In the classroom assessment, the evidence models need to support additional observables whose role is to provide feedback to the students about their performance. Biomass distinguished *feedback observables* that were passed to the feedback system and *final observables* that were passed to the evidence accumulation process. Feedback observables require rules of evidence to describe how they are set, but do not necessarily appear in an EMF. Final observables must appear in an EMF, as well as requiring rules of evidence to evaluate them (from work products, other observables, or some combination). Some observables play both roles, being involved in both task-based feedback and the summative feedback that comes from the evidence accumulation process.

There is also another class of observables called *auxiliary observables* that are not directly used for either feedback or involved in an EMF. Sometimes these are intermediate steps in the calculations, sometimes these are of interest for research purposes (for example, timing information that is collected but not scored). If any of these auxiliary variables are important for research purposes, they must be defined in the evidence model as well.

Rules of Evidence

Although in Biomass the rules of evidence were written in a "IF . . . THEN . . . " format, test designers can use any mechanism for specifying the rules that is clear and unambiguous. Eventually, the evidence rules will wind up as instructions for human raters or requirements for programmers who will write computer code to execute the rules. In the case of more complex rules, a set of predefined work products and the corresponding observable values that can be used for testing will eventually be necessary.

Start with the simplest task in the first scenario, the three multiple-choice questions. The work product in this case is the option selected by the student for each task. The evidence rule is simple:

> If the *selection* made by the student matches the *key*, set the *outcome* variable to `correct`.

Although this rule is fairly simple, it can still teach us some lessons. First, there is a need for a *key* or in more complex cases *evidence rule data.* If we expect the evidence model to be used by different tasks from the same task model, we will need to build in some mechanism to accommodate the different expectations from task variants. In this simple example, the key is just an indicator for the preferred selection. More complex evidence rules might have more complicated descriptions of what the expected results might be. For example, in the automated scoring rule for an essay, the evidence rule data might be a complex computational–linguistic model needed for a text scoring algorithm (Deane 2006).

item response theory (IRT), latent class models, and cognitive diagnosis models can all play this role.

Second, the nature of the evidence rule must be consistent with the purpose of the assessment. In this case, the observable outcome variable only records whether the response is correct or incorrect. In a more complex case the key might be matched to the misconceptions (Bart et al. 1994; Graf 2008). In this case the matching might set several observable variables (either final or feedback) based on the option selected. In this case, the evidence rule data will be a table showing for each possible selection what variables should be set to what values.

The evidence rules for the second task in the segment (the population attribute table, Fig. 14.8) and the last task (what to do next, Fig. 14.9) are only slightly more complex. For the second task, the final (and feedback) observables were based on whether the student selected the boxes that correctly indicated what could be inferred from three populations of mice with different evidentiary properties. There was a column for population, and rows for different potential inferences. The three observables were, for each population (column), whether all inferences were correct (`high`), mostly correct except for a few less-critical instances (`medium`), or missing key inferences (`low`). It is frequently the case in developing the evidence rules for complex tasks that the final observables (here "degree of correct inferences in a column) are summaries of more primitive observables (right/wrong in each cell of the table). Often if the fine grained observables are all providing evidence about the same proficiency variable, combining many of them into a single coarser grained observable will considerably simplify the task of building the EMF, with little loss of information. In this case we just determined whether all the entries in a column were correct. Alternatively, we could have used the count of correct values in a column to make a graded response observable, and ignored the difference among patterns with the same number correct.

The first task, filling out the diagram shown in Fig. 14.7, illustrates the importance of correctly defining the work product in a complex task. Even though the diagram looks quite complex to the student, it has a very simple representation inside the computer. (And it can be reused for other mode-of-inheritance tasks.) The table itself is a collection of cells organized into rows and columns. The work product is a list of what objects the student dropped into each cell. Given that representation it is easy to write computer code to answer questions about what the student placed in each row and/or column. Table 14.5 shows some of the evidence rules for this first task at the highest level. At a lower level, expressions like "MOI(Chromosome Type)" would need to be translated into more specific instructions about what to look for in which column of the table.

Note that the highest score for *MendModGen(1)* is obtained by filling in the table correctly, but partial credit is given for a set of entries that is incorrect but internally consistent. Giving a response like this is more evidence for a claim of understanding the concepts of the Mendelian model, even though it is wrong, than completing the table with internal inconsistencies.

Table 14.5 Rules of evidence for table task

MendModRep(1) [Chromosome Type]
IF Response = MOI(Chromosome Type) is correct
THEN *MendModRep(1)* = 2
ELSE *MendModRep(1)* = 1

MendModRep(2) [Number of Alleles]
IF Response = MOI(Number of Alleles) is correct
THEN *MendModRep(2)* = 3
ELSE IF Response = MOI(Number of Alleles) is partially correct
THEN *MendModRep(2)* = 2
ELSE *MendModRep(2)* = 1

MendModGen(1) [Dominance Relationships]
IF [All aspects of Mendel's Model represented using symbolic forms] are correct
AND [Phenotypic patterns related to all elements of MOI]
THEN *MendModGen(1)* = 3
ELSE IF [Coherent phenotypic patterns related to all elements of MOI]
THEN *MendModGen(1)* = 2
ELSE *MendModGen(1)* = 1

Evidence Model Bayes Net Fragments

Completing the evidence models for the four tasks in the first segment of the Biomass classroom assessment requires specifying evidence model Bayes net fragments, or EMFs (Sect. 5.4) to link the (final) observables to the proficiency variables. Each EMF contained between one and ten observable variables, and had from one to four proficiency variables in its footprint. Figures 14.10, 14.11, 14.12, and 14.13 depict the EMFs of the four tasks from the first segment.

Note that the EMFs in Figs. 14.10 and 14.11 include a *Context* variable to account for possible conditional dependence among observable variables due to shared stimulus materials, work products, or investigation activities (see Sect. 6.2). The *Context* variables for the different EMFs are different variables. In both cases, in addition to understanding the general science content, the students must understand what is required in this particular activity. The *Context* variables provide an alternative explanation for poor (or good) performance on all of the work products in a given task with multiple observables.

The EMF shown in Fig. 14.10, for example, is appropriate for modeling the information about the proficiency variable Disciplinary Knowledge about the Mendelian model (*DKMendel*) based on the observations made from filling out the models of inheritance table (Fig. 14.7). The variable specifies two kinds of observables: *MendModRep(x)*—observable variables bearing on representations,—and *MendModGen(x)*—variables bearing on general terminology and concepts. Given *DKMendel* and *Context*, all seven observables are independent and all are to be modeled using the compensatory design pattern (Sect. 8.5). Again, the (unobservable) variable *Context* is local to the evidence model. Thus, it must be given a distribution to complete the evidence model. This is in contrast to *DKMendel*, a proficiency variable that is "borrowed" from the proficiency model. Its distribution is specified in the proficiency model and not the evidence model.

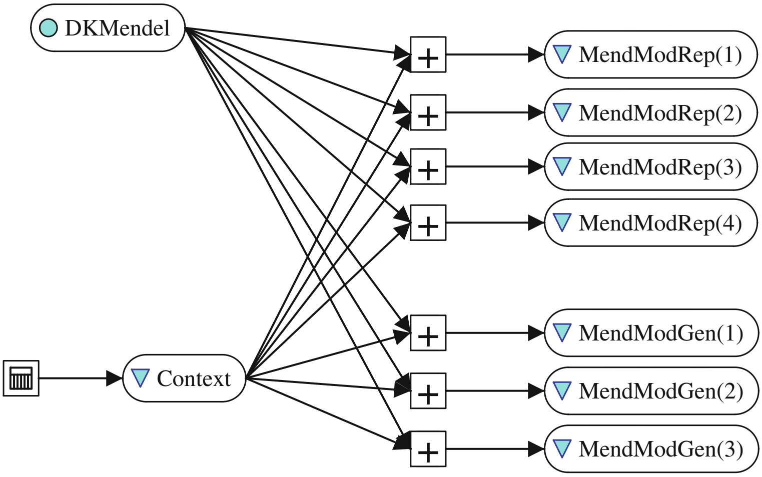


Fig. 14.10 An evidence model Bayes net fragment with seven observables

This is a fragment of the evidence model Bayes net fragment for the first task (Fig. 14.7) for the first task segment with seven observables bearing on knowledge about the Mendelian model (*DKMendel*). The observations concern representational forms (*MendModRep(x)*) or general terminology and concepts (*MendModGen(x)*) A compensatory relationship with a *Context* variable specific to a single task setting allows for conditional dependence among the observable variables. Reprinted from Mislevy et al. (2002a) with permission from The National Center for Research on Evaluation, Standards, & Student Testing (CRESST), UCLA.

The second task in the first segment, filling out the population attribute table shown in Fig. 14.8, requires both basic knowledge about the Mendelian model (*DKMendel*) and Working Knowledge about Inquiry (*WKInqry*) as well as an understanding of the task situation and directives (*Context*). Figure 14.11 shows the EMF for this segment. Note that the footprint of this evidence model is (*DkMendel*, *WKInquiry*), and that this will induce an edge between those two variables in the proficiency model.

The evidence model for the three multiple-choice tasks (Fig. 14.12) is very simple. A single proficiency variable *DKMendel* renders them conditionally independent. This evidence model could be equivalently rendered as three independent evidence models each with one observable. The icon for

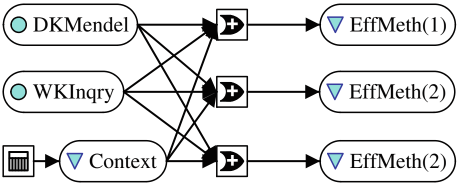


Fig. 14.11 An evidence model using three observables and a context effect

This is an evidence model Bayes net fragment for the Population Attribute Table task in the first Biomass segment shown as Fig. 14.8. This evidence model assesses the conjunction of Disciplinary Knowledge about the Mendelian model (*DKMendel*) and Working Knowledge about Inquiry (*WKInqry*), followed by a compensatory relationship with a Context variable that introduces conditional dependence among three observables that all concern Efficacious Methodology as applied with the Mendelian Model (*EffMeth(1)–EffMeth(3)*). This new distribution type is described in Sect. 15.1.2, Eq. 15.6. Reprinted from Mislevy et al. (2002a) with permission from The National Center for Research on Evaluation, Standards, and Student Testing (CRESST), UCLA.

the compensatory distribution does not indicate a compensatory relationship (meaningless with a single proficiency variable) but rather that the DiBello–Samejima style pseudo-item response theory (IRT) models (Sect. 8.5) will be used to specify the probabilities.

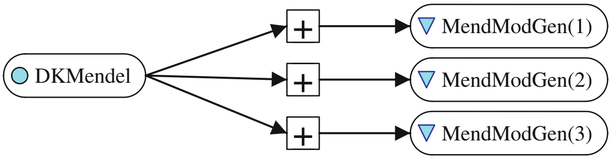


Fig. 14.12 An evidence model fragment with three conditionally-independent observables

This is an evidence model Bayes net fragment for Evidence Model 3 showing three conditionally independent observables concerning general terminology and concepts in the Mendelian model (*MendModGen(x)*) that depend on Disciplinary Knowledge about the Mendelian model (*DKMendel*). Reprinted from Mislevy et al. (2002a) with permission from The National Center for Research on Evaluation, Standards, and Student Testing (CRESST), UCLA.

The last task in the first segment, "what to do next," (Fig. 14.9) is interesting because although Domain Knowledge of the Mendelian model (*DKMendel*) is needed, the task is mostly about inquiry skills (*WKInqry*). This is a situation for which an inhibitor distribution (Sect. 8.5) is ideally suited. Once the student reaches the basic level of *DKMendel*, additional levels do not help. Figure 14.13 shows the graphical structure of this model.

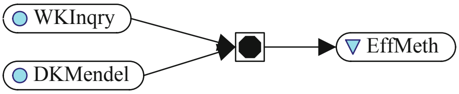


Fig. 14.13 An evidence model fragment using the inhibitor relationship

This is an evidence model Bayes net fragment for Evidence Model 4, showing one observable about the efficaciousness of solution methodology (*EffMeth*), which depends on Working Knowledge about Inquiry (*WKInqry)* in an "inhibitor" or "hurdle" relationship with respect to Disciplinary Knowledge about the Mendelian model (*DKMendel*). That is, a medium level of proficiency in *DKMendel* is required, but above this minimum, response probabilities depend only on *WKInqry.* Reprinted from Mislevy et al. (2002a) with permission from The National Center for Research on Evaluation, Standards, and Student Testing (CRESST), UCLA.

A total of 48 evidence models were needed to manage incoming information from the various tasks in Biomass. The description of the models given here is not complete, as an operational measurement model requires numbers for all of the conditional probability tables in both the proficiency model and the EMFs. Getting the numbers into the Biomass measurement model used many of the steps and procedures specific to the Bayes net models described in Part II. Chapter 15 describes both how the initial expert parameters were specified and how they were later refined with data.

14.4 The Assessment Delivery Processes

After the numbers were added to the models described above (using the methods described in Chap. 15), Biomass was ready for implementation. The Biomass project uses the four-process assessment delivery system (Sects. 2.4.2 and 13.1). This section considers implications of the Biomass design rationale for each of the processes in turn.

Biomass is a web-delivered assessment.[6] That means it must fit within the established protocols that govern the Internet, in particular, transmission control protocol/Internet protocol (TCP/IP) and hypertext transfer protocol (HTTP). Section 14.4.1 briefly describes those protocols and how they fit with the four-process architecture. In particular, the Internet provides a mechanism for distributing the workload of any program over several computers. The four-process architecture provides a mechanism for thinking through those issues, as discussed in Sects. 14.4.2 through 14.4.5.

[6] Here "web-delivered" means using internet protocols and client–server architecture. In our early tests the client and server programs actually ran on the same computer. But if an assessment were delivered across locations, especially if it had high stakes, internet security would become important. We do not address this issue, as material in this rapidly evolving field has a half-life measured in months at best.

14.4.1 Biomass Architecture

Biomass is a web-delivered assessment. This means it can be run in a typical web-browser by anybody with a computer that is connected to the same network on which the Biomass server is running. To understand the implications of this for the design of Biomass requires a minimal knowledge of the protocols that support the communication between the web server and the browser. The description below should be enough to follow the details of the Biomass design that follow. More complete descriptions are readily available on the Internet.

The basic communication protocol used both on the Internet and local area networks (LANs) is known as TCP/IP. Under TCP/IP every computer is assigned an 6-byte IP address, which is like a phone number. You can contact any computer on the Internet if you know its IP address. (If you know its name but not its IP address, you use a domain name service, or DNS, to look up the IP address). In addition to the IP address you need to know the port number of the program you want to talk to on the other computer. Different programs typically have different port numbers. For example, email is typically sent on port 25, and web page requests typically come in on port 80.

If you have the IP address and the port number of a program on another (or the same) computer, you can open up a communication channel, called a *socket* between the two programs. Typically, one of the two programs listens on the socket and then responds to messages it receives. The other program sends a message then listens for a reply. Often the listening program sets up a miniprogram called a *thread* to calculate the proper reply to the message it just received. The operating system handles sharing the computer run time among many threads, so it appears to users as if they are all running at the same time.

Combining TCP/IP with the four-process architecture makes it possible to run any of the four processes on any computer. For example, you could run the activity selection, presentation, and evidence accumulation process on one machine, but use another machine to do evidence identification (for example, a special server that provided natural language processing to score essays). In Biomass, all four processes[7] ran on the same machine, with some processes implemented in Visual Basic and some in Java (a decision based mainly on the resources available to do the programming). Using TCP/IP to communicate between the programs avoided many of the hassles normally associated with integrating programs written in two different languages.

Rather than have all four processes communicate directly with each other (which would require each of them to know the other IP address and port), Biomass used a central message center process to handle communications among the processes. Having this one process responsible for routine communications among all the others made it simple to change the flow among

[7] Actually, the presentation was split between the client and the server, as we will see below.

the four-process model to support the two different modes of the Biomass assessment (Sect. 14.4.7). The message center served another purpose: it was possible to structure the message center so that it responded to each message with the next response from a script. This facilitated testing each of the four processes before the final integration.

HTTP is a layer that exists on top of TCP/IP for sending web pages across the Internet. To start, a program called a *browser* sends a "get" message to another machine called a *server*. The server responds by sending a file back to the browser in reply. That response is often in a special format called hypertext markup language (HTML). The HTML page contains data and instructions for displaying it in the browser. It may also contain references to other files on the server (or on a different server). Those files could be images, audio, video, or special lightweight programs called *applets* (Flash and Java are common languages for applets). Different browsers have different capabilities as far as what kinds of media they can display, what kinds of applets they can play, and how much computation can be done within the web page. When the user is finished with the content on this page, the user clicks on a *link* in the browser, which causes the browser to send a new "get" message to the indicated server.

One useful kind of object that can be described in HTML is a *form*, in which the user makes selections and fills in fields. After filling out a form, the browser sends what is known as a "post" message. A post message is a little bit more complex than a "get" message, in that in addition to the address where it is to be sent, it contains data expressed as key–value pairs, where the key and the value could be any string. Using extensible markup language (XML) large, complex objects can be described as strings. Biomass did not take advantage of this, but NetPASS (Williamson et al. 2004a, Behrens et al. 2004) did, using XML to pass around completed network diagrams and troubleshooting protocols. Usually when receiving post messages, the server does not just find the requested file and return it, but rather does some calculations involving the posted data to figure out what the next page to display should be.

There are three big challenges in building a web-based assessment. The first is test security. If the examinee is running the assessment from her own computer in her own home, there is little to stop her from opening an new browser window and searching the Internet for information related to the task on hand. If that behavior is not a desired part of the assessment, then some form of proctoring will be required.

The second is that there are a large number of different types of computers and browsers out there on the Internet. All of the different operating systems and browsers support different sets of capabilities. Even though there is supposed to be a common core of services supported by all browsers, some implementations are incomplete and some provide extensions to the core feature set. Moreover, they differ as to how they handle mistakes in the HTML coding. One browser may "fix" the mistake so as to be close to the desired rendering of the page, while another browser may produce something that is completely illegible. This difficulty requires that developers either do extensive crossplatform testing or restrict the assessment to one browser/operating system platform (irritating any user for whom that is not the preferred platform).

The third is that the web server expects results to be calculated synchronously, but the four-process architecture sends messages asynchronously, returning a new message when it it ready. Generally, sending a "get" or "post" message to the web server causes it to launch a thread that is supposed to calculate the reply. To make this work with the four-process message center, the Biomass presentation process "parked" the thread in a special waiting list and waited for a reply from the message center. When a message came from the message center for that user, it woke the appropriate thread which then turned the message into a web page to be delivered back to the user.

Based on this architecture, we can now look at each of the four processes in turn.

14.4.2 The Presentation Process

The primary responsibility of the presentation process is to present the task to the examinee and return the work product to the evidence identification process. However, the Biomass presentation process played a number of other roles as well. In particular, it presented both task-based feedback (in the classroom assessment) and the summary feedback (the score report). Also, it played a special role in starting the assessment and provided some special screens for teachers to configure Biomass for a particular classroom setting.

As the presentation process was built on a web server, most of the presentation was in the form of web pages written in HTML. As HTML allows the capability to embed images, tables, and forms, it could easily handle most of the demands of the Biomass tasks. In fact, only 10 of the 29 segments of the Biomass scenarios required something more. The something more was the drag-and-drop capability for table completion tasks like the one shown in Fig. 14.7. A single drag-and-drop applet configured in different ways for each task provided enough additional capability beyond that which was provided by HTML forms to support all of the tasks in Biomass.

The lesson is that what determines the quality of the assessment is not the sophistication of the technology, but how well the technology supports gathering the evidence we need to make the inferences to ground the targeted claims of the assessment. In a previous assessment project, creating a design rationale for assessing dental hygienists' problem-solving skills (Breyer et al. 1999), a sophisticated simulator was available for scenes and actions in a dental practice. However, a discussion with the experts revealed that a crucial part of the hygienist's job was reasoning from history and examination results to possible underlying conditions, and what the possibilities were for what to do next—all tremendously important, and all happening unobserved in the head of the candidate. We suggested a work product that was a simplified insurance form to capture the results of this thinking, as evidence that was at once closer to the proficiencies of interest and authentic to actual clinical practice. In Biomass, the key was understanding that experiments are an important part of assessing Integrated Knowledge. The examinee can act in ways that reveal their understanding of how to set up an experiment or analyze data from an experiment using quite simple interactions, as long as it provides the right kind of evidence.

One question that almost always comes up in designing a web-based application is how much work should be done on the client machine (within the web browser) and how much on the server. In particular, when the evidence rules involve simple key matching it is tempting to build the logic for the evidence identification process into the web page. However, this effectively changes the work product from the selection made by the examinee to a variable indicating whether the selection was correct or not. There are a number of reasons for keeping the evidence identification on the server side: (1) Flexibility—if the purpose of the assessment changes, the evidence rules might change as well. In particular, there might be different task-level feedback depending on the selected options. (2) Scalability—embedding the evidence identification rules in the web page might work for simple multiple-choice rules but not for a complex essay grading system. (3) Security—If the evidence identification rules are embedded in the web page, then the key must be as well. This offers a hacker an opportunity to reverse engineer the web page to find the key.

On the other hand, the presentation process needs to do at least some interpretation on the client side. If the work product is defined to be the raw mouse clicks, information such as "Mouse Down at (125,234)" is useless unless the layout of the elements on the screen is known. Usually, the best level for the work product is one level up, in this case which selections were made. A rule of thumb is that usually the client should provide semantically meaningful results. This maximizes flexibility, in that if hardware or task details change on the client side and require revising this step of identification, it is more likely that changes in code will not be needed on the server side. Note that it is possible for the work product to contain additional information that is not directly used by the Evidence Identification process. For example, the presentation process may want to capture timing information that is not used for scoring, but is used for verifying that tasks meet their design targets for how quickly the student can complete the tasks.

The feedback functions are usually fairly simple to implement. A number of extensions to HTML (e.g., active server pages, ASP; Java server pages, JSP; and personal home page, PHP) allow the developer to switch content on and off depending on the values of feedback observables. This approach was used to customize the task-based feedback in Biomass based on the observed values of the feedback observables. For score reports, values from the evidence accumulation process (stored in the administrative process described below) can be inserted into the appropriate places in score-report web forms.

Finally, the special startup and configuration screens again are simple to implement using basic HTML codes. The key difference is that in many cases, the other three processes are not yet started. In Biomass, a special administrative mode took care of these special startup and configuration tasks. In Biomass, the message center had a special administrative mode that routes traffic between the administrative and presentation processes. If the administrative process and message center were not started, then the presentation process would take charge of starting them as well.

From the message center's perspective, every time the user pressed a submit button on the screen, the browser turned this into an HTTP post request. If the message included work products, these were coded as strings.[8] The web server then formatted this post request as a message center message and sent it to the message center for appropriate routine. If it was a request by the user to jump to a new task, this was routed to the activity selection process for further processing. If the student was moving using the normal work flow, then the message would contain either feedback or final work products that would be sent to evidence identification for the next stage of processing.

14.4.3 Evidence Identification

The role of the evidence identification process is to identify the essential features of the work product that provide evidence about the examinee's current knowledge, skills, and abilities. They could be as simple as the correctness of a multiple-choice response or as complicated as subtle patterns across hundreds of actions and final results in a minimally constrained hour-long computerized patient management problem. These features are recorded as values of observable variables, possibly after multiple stages of processing. As discussed above, feedback observables are sent back to the presentation process to use in customizing task-level feedback; final observables are sent on to the evidence accumulation process for use in summary level scoring.

The challenge of building the evidence identification process lies in translating evidence rules, such as those found in Table 14.5, into computer code. Note that whatever additional meaning is placed on evidence rules, they will ultimately become specifications for the evidence identification process code. In Biomass, as each task was a one-off; the code for the evidence identification process was unique for each task. The evidence rules were quick enough to process that both the feedback and final observables were calculated at each time step, even if only one was requested.

The heart of the evidence identification process was a dispatch system that determined, based on the ID of the task, which evidence identification code needed to be run. In Biomass, the key was programmed into that code. In a system with more different variants of tasks and sharing of scoring code,

[8] XML was just coming in as a new technology when Biomass was built. If we had done it a year later we would have used XML to encode the strings.

the evidence identification process would also need to fetch evidence rule data from the task/evidence composite library, again keyed by task ID. For example, a key-matching rule for a multiple choice task, would need to fetch the key from the database. More sophisticated tasks, such as a variant of the one shown in Fig. 14.7, would require more sophisticated evidence rule data; for example, a list of the expected dominance relationship and the expected genotype-phenotype pairs. This evidence rule data must be authored (or automatically generated: "automated automated scoring" (DiCerbo and Behrens 2012)) along with the task.

Testing has proved to be a significant challenge in implementing many evidence identification processes. Thorough testing requires sample work products that will generate each level of the observable, and the more complex the work product, the more difficult they are to produce. Often the presentation process must be at least be partially built to be able to generate the required work products. If the work product represents a complex construction, the space of possible work products can be quite large, and generating enough work products to provide sufficient testing can be difficult. Pilot test data can be helpful because real students always come up with surprising ways to approach a problem that designers had not anticipated.

In the case of Biomass, the four-process architecture made it fairly easy to write special rules for the message center so that instead of using all four processes, it just fed the evidence identification process the sample work products for testing. The generated observables could then be compared to what was expected. This produced an automatic testing harness for the evidence identification (and all of the other) processes.

In addition to the testing of the evidence identification process, these test work products play a role in the final integration testing of the whole system. Now, instead of actual work products, the system testers need scripts for how to generate work products, and then descriptions of what the observables and expected feedback will be.

14.4.4 Evidence Accumulation

Section 13.3 already provides many details on how the the evidence accumulation process (abbreviated EAP here) works, and Sect. 5.4 provides the mathematical algorithms. The Biomass evidence accumulation process is simply an implementation of the algorithms described there. When a new student began the Biomass Assessment, the EAP would create a scoring model for that student by copying the proficiency model Bayesian network. These scoring models would be indexed by the student ID (assigned by the Biomass administrative process).

When the observables for a task came from the evidence identification process, the header of that message would contain two critical pieces of information, the student ID and the task ID. The EAP would use the student ID to find the appropriate scoring model, and the task ID to find the evidence model fragment (or fragments) associated with the task. (Chap. 15 describes the evidence model fragments for the first task.) It would then instantiate the observables in the evidence model fragment to their observed values and calculate the distribution over the footprint variables. This distribution would then be inserted as virtual evidence into an appropriate clique of the scoring model.

Upon request, the EAP could calculate statistics of the scoring model to use for score reporting and other purposes. In the Biomass implementation, the statistics were divided into two categories. Statistics that were easy to calculate, like the marginal distributions of the reporting variables, were marked "report on update" and reported after absorbing the evidence from each task. As Biomass was not adaptive, these were not used by the activity selection process, but were stored by the administrative process and could be used for later analyses, such as constructing a evidence balance sheet (Sect. 7.2.1). Statistics that were time-consuming to calculate would not be reported after each task, but only when requested (because generating an extended score report was required).

The classroom version of Biomass was intended to be used in multiple sessions across many days. This required the EAP to additionally have the capability to save the scoring model as a file and restore the saved model. Early testing revealed a bug in the software that was causing the restored scoring model to lose the evidence. This proved to be an easy bug to work around as all of the observable variables for the student were stored in the administrative process database. To restore the model, the administrative process and EAP could work together to replay the series of tasks taken by the student—skipping the presentation and evidence identification processes, since the saved observables from the early work were already on hand.

The classroom version of Biomass raised another issue because the student could attempt the same task multiple times. The Biomass design team talked over the issues discussed in Sect. 13.3.3 and eventually decided to simply treat the evidence from each attempt as independent evidence. This solution was the easiest to implement, and should be a close enough approximation for the relatively low-stakes purposes of the classroom assessment. However, being able to intelligently account in the scoring for feedback and multiple attempts that (in some systems, intentionally) produce learning remains an active area of research (Sect. 16.2.2; Ritter et al. 2007).

Biomass actually ran two evidence accumulation processes in parallel. The first, the Bayes net scoring process described here, was used for the primary scoring. The second, a simple number-right scoring process, was used for the purpose of counting tasks to provide a tasks completed field on the score report. The idea of running multiple scoring engines in parallel is a very powerful feature of the four-process architecture. For example, an IRT-based scoring engine could be used to provide a scale score on overall proficiency while a Bayes net scoring engine provided proficiency-based diagnosis. Small special-purpose evidence accumulation processes can also be be included, for example, to simply count instances of particular misconceptions or kinds of errors, for feedback at the end of a segment or the end of the assessment (of course evidence identification processes would need to be able to identify them first).

14.4.5 Activity Selection

The classroom version of Biomass supported two mechanisms for activity selection, but both were very simple compared to the expected-weight-of-evidence adaptive task selection of Adaptive Content with Evidence-based Diagnosis (ACED) (Sect. 13.1). The first activity selection scheme in Biomass was a simple linear mode in which the student moved from segment to segment in the scenario (following the next step in the sequence given in Table 14.3, or the similar table for the "lizard" scenario). This was implemented through table look up: for each task the activity selection process looked up the task ID of the next task and sent a message to the presentation process to present that task. The second activity selection mechanism was user selection. Again, this was implemented through a simple table lookup: given the name of the segment, the activity selection returned the appropriate task ID.

The culminating assessment was even simpler. The option of user control was no longer available, so only the "next task" mechanism was implemented. One of the segments in the culminating assessment had a quite complex task in which the student had to navigate through several screens, formulating hypotheses and performing experiments to test those hypotheses. It was open ended in the sense that the student could formulate and test a large number of possible hypotheses. However, the culminating assessment had no interim feedback, and hence this entire set of activities could be considered one "task" even though it spanned several web pages. Thus, all the complexity could be buried in the presentation process and not affect the activity selection process.

14.4.6 The Task/Evidence Composite Library

In the center of the four-process delivery system is the task/evidence composite library. It makes available to the four processes the information they need to carry out their functions. The task/evidence composite library for Biomass was not stored in a single location, but rather each of the four processes maintained the part of the library that it used. In particular, the tasks (and their feedback) were stored as a series of web pages in a location known to the presentation process. Similarly the Bayes net fragments were stored as a series of files in a location known to the evidence accumulation process.

What was important about the library is that the task ID served as the primary key to each of the separate databases. In particular, the presentation and evidence accumulation process could locate the right data files given the task ID. Similarly, the evidence identification process could select the right evidence rules based on the task ID. Finally, the activity selection process knew how to go from one task ID to the next.

14.4.7 Controlling the Flow of Information Among the Processes

One of the critical differences between four-process delivery architecture and traditional delivery systems is the way in which the flow of information through the system is managed. In traditional delivery, this flow is usually "hard-wired"; that is, delivery processes are strictly defined to receive fixed information from, and send fixed information to, certain other processes in a fixed sequence. When a designer changes the purpose of an assessment—its context or conditions for use—all this must change as well. The two modes of Biomass prototype illustrate the differing logic requirements for the sequence of interactions among the delivery processes for the learning and culminating modes.

The purpose of the Culminating Assessment is to determine a student's level of proficiency at the end of a course, providing overall results and summary feedback at the end of the testing session. To this end, the activity selection process tells the presentation process to start each successive task after capturing the work products of the previous one. All of the work products can be sent to evidence identification at once for task-level scoring. Evidence identification uses the resulting observable variables in two ways. Some observable variables can be used to generate task-level feedback, which is presented to the examinee at the end of the assessment. Some observable variables are passed on to evidence accumulation to update beliefs about proficiency variables, and subsequently ground higher-level feedback, instructional decisions, or score reports. Note that all of the task- and summary-scoring can be accomplished by evidence identification and evidence accumulation at a distant time or place from the actual testing session. Therefore, in the Biomass Culminating assessment, the message center was configured to send messages containing work products from the presentation process to both the activity selection process and the evidence identification process. That way the activity selection could jump directly to the next task, while the evidence identification and accumulation processes worked in the background to generate the final score report.

The purposes of the Learning Assessment, on the other hand, are to provide practice and support learning in preparation for the Culminating Assessment. These purposes are served by immediate feedback, cumulative scoring, and opportunity to repeat a task. However, because the task sequence was not adaptive the messages still did not need to flow around all four processes before a new task message could be sent.

The need for immediate task-based feedback did however cause a more complex message flow. As before, the activity section process tells the presentation process to start a new task. The student produces work products which are sent to evidence identification to calculate the feedback observables. The feedback observables are sent back to the presentation process which then generates appropriate feedback screens. After viewing the feedback, the student decides to either repeat the task, in which case the feedback cycle will repeat, or to submit the work product (perhaps after minor changes) for final scoring. In the latter case, the work product is submitted again to the evidence identification process, this time as a final work product. Now the evidence identification process sends the final observables to the evidence identification process. As in the Culminating Assessment, the task selection does not depend on either the output of evidence identification or evidence accumulation. Therefore, as soon as the final work product is received, the activity selection process can send a message to the presentation process about what task to present next. Thus, the student can be working on the next task while the scoring happens in the background.

When the evidence accumulation process is finished scoring, it sends a message containing the critical scores for the score report back to the message center. In Biomass, the scores were only sent to the administrative process to be recorded in the database. When the students finished the assessment (either by choice in the Learning Assessment or by reaching the end in the Culminating Assessment), they were given the option of viewing a summary score report. If a student had not completed the assessment, the report would be based only on those tasks the student had completed.

As this report was generated from data in the administrative process, the student did not even need to be logged into the system to generate a score report. As long as the scores were in the database, Biomass could generate the report. It would be straightforward to implement a teacher's view in which the teacher could monitor a classroom of students (or teams) working through Biomass, and visit the workstations of the students who needed the most help. Almond et al. (2009a) describe some possible classroom views, using data from the ACED assessment.

14.5 Conclusion

This chapter might seem a bit of an odd man out—way too much on assessment design for a book on Bayes nets. This would be true if the book were just about Bayes nets as an analytic method, to be applied with a some unspecified assessment. Even with this limited goal, there is a lot to learn about Bayes nets technically, and it has indeed been our purpose to provide a solid foundation for this aspect of the assessment enterprise.

The larger point, though, is that using Bayes nets in assessment effectively is not just about applying an analytic method to data already gathered from an assessment that already exists. Rather, it can, and ideally should, arise from designing an assessment system (all the elements, processes, and activities) that embodies an assessment argument for some purpose. For some purposes and in some contexts, the kinds of inferences that are desired, the kinds of evidence that supports them, and the kinds of tasks that evoke the evidence will produce data that Bayes nets are well suited for reasoning about.

This will be the case especially when the evidence involves elements such as multiple aspects of proficiency, in different combinations in different tasks; conditional dependencies arise among observable variables, whether testlets of structural relationships among task conditions or examinee actions; multiple complex tasks need to be authored around the same evidentiary structure; or multiple complex situations need to be assembled flexibly, as in adaptive testing and tutoring systems, yet provide information in the same proficiency metrics (Almond and Mislevy 1999).

Biomass was a demonstration assessment meant to show these principles in action. It illustrates several innovative and ambitious features, including the following:

- Designing assessments in terms of re-usable schemas, objects, and processes.
- Developing assessments to assess standards, in such a way as to both give them concrete meaning and address higher-level forms of knowledge–in this case, inquiry in science, with content from transmission genetics and microevolution.
- Using dynamically assembled Bayesian inference networks to manage the accumulation of evidence in a multivariate model, from multivariate and sometimes conditionally dependent observations.

This chapter has shown how the models and approaches of evidence-centered design can be used to organize the design and implementation of such an assessment, and do so in a way that lends itself to the reuse of the materials and processes. The success with Biomass led to other projects, such as NetPASS, ACED, and SimCityEDU (Mislevy et al. 2014), using the ECD methodology as part of their design philosophy.

Although this chapter tells a major portion of the Biomass story, one significant piece is missing: how the Bayesian networks for the proficiency and evidence models were built and how the parameters for those models were set. The next chapter provides some of the more interesting details of that process.

Exercises

The problems in this chapter build on the concepts of previous chapters. Refer back to those chapters as necessary in addressing these problems.

14.1 (Task Model for Unified Knowledge). What might a task model for Unified Knowledge look like? What elements would it need to include?

14.2 (All Integrated Knowledge Tasks Assembly Model). Suppose that in the Biomass assembly model, the designers included only tasks that tap integrated knowledge. Would this present a problem for measurement?

14.3 (All Proficiency Pairs). Suppose that the designers proposed an assembly model in which (1) each of the domain knowledge proficiencies was paired with each of the working knowledge proficiencies in at least one task, (2) each of the domain knowledge proficiencies was paired with each of the integrated knowledge proficiencies in at least one task, and (3) each of the working knowledge proficiencies was paired with each of the integrated knowledge proficiencies in at least one task. Would this present a problem for measurement?

14.4 (Compensatory vs Inhibitor models for *Context*). Consider Task 1 (filling out the table in Fig. 14.7), whose evidence model structure is given in Fig. 14.10. Suppose that an inhibitor or a conjunctive design pattern was used for the conditional probability tables instead of the compensatory pattern that was chosen. How would that change the interpretation of *Context*?

14.5 (Mice Make Me Go EEK!). Suppose that in the field trial, a portion of the subjects had difficulty answering the questions because every time they saw the pictures of the mice, they just wanted to jump up on their chairs and scream "EEK!"[9] If this portion is sufficiently large, then it should probably be taken into account in the measurement model for Biomass. Explain how the measurement model for Biomass could be modified to include this reaction.

14.6 (Radical or Incidental). For each of the following task model variables, explain whether it is radical or incidental. If it is radical explain whether it effects difficulty, evidentiary focus, or both. If it is incidental, explain whether there is a range beyond which which it becomes radical.

1. The species of animal/plant being crossed.
2. Whether the species reproduces sexually or asexually.
3. The name of the fictitious student used in the examples.
4. The phenotypic trait expressed by the gene (e.g., hair color).
5. How many alleles there are.
6. The type of dominance relationship (e.g., dominance, codominance, in complete dominance).
7. Which chromosome the gene is found on.
8. Whether or not the gene is sex-linked.
9. How the initial field population was gathered.

[9] This is known as the Slartibartfast effect (Adams 1978).

15

The Biomass Measurement Model

The previous chapter described the basic design and construction of the Biomass assessment. It showed how the graphical structures for the proficiency and evidence model Bayes net fragments arise jointly from the theory of learning in the domain and the construction of tasks, to embody an assessment argument. This chapter examines how to populate the conditional probability tables with numbers.

Quantification of a Bayesian network comes about in two phases. The first phase is specifying a prior distribution for the unknown quantities. The second is updating that prior with data. As described in Fig. 8.3, this distribution, whether prior or posterior, "floats above" the basic Bayesian network. If we want to score a student, we simply have the current mean values "drop down" into the network for that purpose.

Section 15.1 describes how the prior for the Biomass Bayesian model was constructed, both in terms of choosing parameterizations for the conditional probability tables, and choosing initial parameters. Section 15.2 describes a brief expedition to gather data, which yielded 28 observations of questionable representativeness. Section 15.3 describes how those data can be used to update the values for the conditional probability tables.

Taken together, this chapter and the previous one illustrate an interplay between substantive knowledge and empirical evidence in constructing and refining the statistical model in a Bayes net assessment with complex tasks. The previous chapter discussed in detail the design process which began with an analysis of the domain and the targeted proficiencies. The range of situations, representations, and forms of activities that could adduce to obtain evidence about students' proficiencies was explored. The task models were constructed by test developers and substantive experts, expressly to provide this evidence; the tasks were structured such that the salient features of performance could be identified, and their relationships to proficiencies specified (at least provisionally). Thus, the Bayes net structures arose in conjunction

R. G. Almond et al., *Bayesian Networks in Educational Assessment*,
Statistics for Social and Behavioral Sciences, DOI 10.1007/978-1-4939-2125-6_15

with the building of the evidentiary argument from the start, rather than being built only further down the road when data arrive.

This, we argue, is a good assessment design practice. The general approach is indeed reflected in high-quality assessments, with Bayes nets or other psychometric models (see for example, Leighton and Gierl 2007; Wiggins 1998; Wilson 2004). It is sometimes employed in some formative assessments and learning systems, but more often decisions are made in such assessments in a more ad hoc manner. They may have used expert judgment (which might be good, but might not be) to design tasks and evaluation schemes, and usually with little data to start. The problem is that if they have wired-in rules to synthesize evidence and make decisions, they cannot exploit the time-tested machinery of statistical probability to check models, improve estimates, or incorporate other sources of information as data arrive. There is no well-understood pathway for improvement.

The Bayes net approach can model different kinds of relationships and deal with different kinds of data in these applications. The 1980s saw much debate about ways of dealing with uncertain evidence in the context of expert systems: probability theory, of course, but also fuzzy logic (Zadeh 1965), belief theory (Dempster 1990), and credibility factors (Shortliffe and Buchanan 1975). Researchers such as Pearl (1988) and Spiegelhalter et al. (1993) argued that probability had substantial advantages for dealing with uncertainty in the context of noisy information and complicated relationships. Their work on Bayes nets extended probability-based reasoning in practical ways to expert systems, which tackle the same challenges we face in complex assessments; they start with lots of expert beliefs but not much data. With the Bayes nets framework, expert beliefs are a starting point—for design as well as for inference—rather than an ending point.

In this spirit, the initial Biomass network specifications could be used as an expert-data-only system to ground inferences about students' strengths and weaknesses with regard to science standards. We see in this chapter the way that initial pilot data are used to refine estimates and provide some initial model checks. Presumably better inferences about students could be based on the improved model. The process can continue as more data accumulated, using the model estimation and model checking methods discussed in earlier chapters, to improve the model (and also feed back to improve task design).

15.1 Specifying Prior Distributions

The complete Biomass conceptual assessment framework contained 15 proficiency variables and 28 segments, where each segment consists of multiple tasks with multiple supporting evidence models. This chapter focuses on the models that support the first segment of an investigation in transmission genetics, called "Mice1." In the scenario, a student José discovers a population of mice, notes how many mice have each of four coat colorings, and decides to investigate the mode of inheritance of coat color in mice. This segment takes from 10–20 min to complete and yields 14 observable variables, each providing a single categorical response on a 3-point scale. These 14 observables come from the four tasks described in Sect. 14.3.3 and were scored using the four evidence models described in Sect. 14.3.4. The first segment only provides evidence about the proficiency variables *DKMendel* and/or *WKInqry*, so this chapter will use a simplified proficiency model containing only those two variables (Table 15.1).

Table 15.1 Task and evidence models from the first Biomass segment

Task model		**Evidence model**		**Observables**
Name	**Figure**	**Name**	**Figure**	**count**
TM 1	14.7	EM 1	14.10	7
TM 2	14.8	EM 2	14.11	3
TM 3	Not shown	EM 3	14.12	3
TM 4	14.9	EM 4	14.13	1

Task/Evidence Model 1 (TM1/EM1) concerns re-expressing José's verbal hypothesis about the mode of inheritance with a tabular diagram. Figure 14.7 shows the tabular diagram and Fig. 14.10 shows the evidence model graph fragment.

Task/Evidence Model 2 (TM2/EM2) concerns a table (Fig. 14.8) that a student was asked to fill out, containing several statements about implications of the mode of inheritance. The form of the EMF for this task is shown in Fig. 14.11.

Task/Evidence Model 3 (TM3/EM3) consists of three multiple-choice questions about implications of forms of dominance. *DKMendel* is the only proficiency variable, and the responses are posited to be conditionally independent. The form of this EMF is shown in Fig. 14.12.

Task/Evidence Model 4 (EM4) asks what José should do next, after having formalized his hypothesis about the mode of inheritance of hair color based on the field population (Fig. 14.9). The form of this EMF is shown in Fig. 14.13.

The first two evidence models include "context" variables (Sect. 6.2), $Context_{TM1}$ and $Context_{TM2}$. These are variables local to the respective evidence models meant to handle conditional dependence among the observed outcomes that is not explained by the proficiency variables alone. (It is thus similar in spirit to the testlet parameter of Wainer et al. 2007.) These variables are local to the scoring model for a specific task, and hence we add a subscript indicating the task (or task model) they are related to. Although they are placed in the evidence model to preserve this task dependency, they are more like a proficiency variable in that they designate a latent property of the student, rather than an observable property of the student's work. In each case, we assume that *Context* can take on two values that we will label `familiar` and `unfamiliar`; familiarity is a common source of conditional dependence, but there are plenty of others, and this mathematical structure accounts for any of them that have a compensatory effect with the proficiencies of interest (Yen 1993).[1]

Altogether, to specify the Bayes net model for Segment 1 of the Biomass interim assessment we need to specify conditional probability tables for 18 variables: 2 proficiency variables, 2 context variables, and 14 observables. The 2 proficiency variables were modeled using the hyper-Dirichlet distribution (Sect. 15.1.1). The 14 observable variables were each modeled using a variant of the DiBello–Samejima design patterns (Sect. 8.5); Sect. 15.1.2 describes this process. The two context variables are given fixed distributions, since their role is simply to model conditional dependence among the observables in a task. Section 15.1.3 provides a tabular summary of the prior and compares it with the posterior after updating using the pilot data from the experiment described in Sect. 15.2.

15.1.1 Specification of Proficiency Variable Priors

In the first segment, only two proficiency variables, *DKMendel* and *WKInqry*, appear in the footprint of any of the four evidence model for that section. For the calculations in this chapter, we have simplified the proficiency model from 15 variables to only those two variables. Both *DKMendel* and *WKInqry* are categorical and can take on one of the three values: `high`, `medium`, and `low`.

The experts expect that *DKMendel* and *WKInqry* are correlated; this demands an edge between them. The question remains how to orient the edge. If the student learns the inquiry skills in the context of genetics, then they would need to learn the domain knowledge first. This would suggest orienting the edge from *DKMendel* to *WKInqry*. However, they may have learned the inquiry skills in the context another discipline (e.g., ecology or chemistry). However, in Biomass we are always *assessing* working knowledge skills in the context of genetics, so it makes sense to orient the edge from *DKMendel* to *WKInqry*.

Because in the reduced graph *DKMendel* has no parents, it has a conditional probability table (CPT) with a single row (actually in this case an

[1] There is neither interest in, nor data sufficient for, estimating the extent to which a student's performance on the observables in the task are meaningfully different than that which would be expected under conditional independence. The strength that is estimated for the contribution of a task's *Context* variable reflects the frequency with which students' performances in the task are a bit better or worse as a set than would be expected. The practical impact is to appropriately reduce the strength of updating for proficiency variable.

unconditional probability table). Positing prior exchangeability for students[2], we have the following distribution for *DKMendel*:

$$\textit{DKMendel}_i \sim \mathrm{Cat}(\lambda_1, \lambda_2, \lambda_3) \ ,$$

where λ_m is the probability that Student i is in State m of *DKMendel*. The natural conjugate prior is the Dirichlet law. We posit the relatively uninformative prior

$$(\lambda_1, \lambda_2, \lambda_3) \sim \mathrm{Dir}\,(3, 4, 3) \ . \tag{15.1}$$

The variable *WKInqry* also takes one of three possible values: `high`, `medium`, and `low`. However, its distribution is now conditional on the state of *DKMendel*. Therefore, it has a CPT with three rows, each of which is an independent categorical distribution.

$$\textit{WKInqry}_i | \textit{DKMendel}_i = p \sim \mathrm{Cat}\,(\lambda_{p1}, \lambda_{p2}, \lambda_{p3})\,,$$

where λ_{pm} is the probability that Student i is in State m of *WKInqry* given that she is in State p of *DKMendel*. Assigning an independent Dirichlet law to each row of this table produces a hyper-Dirichlet law for this CPT. The experts anticipate that *DKMendel* and *WKInqry* will be positively associated among students; students with more knowledge about the concepts and representational forms of the Mendelian model will probably have more skill in applying their knowledge. We posit a set of parameters for the hyper-Dirichlet law reflect that positive association:

$$\begin{aligned}(\lambda_{11}, \lambda_{12}, \lambda_{13}) &\sim \mathrm{Dir}(5, 3, 2)\\ (\lambda_{21}, \lambda_{22}, \lambda_{23}) &\sim \mathrm{Dir}(3, 4, 3)\\ (\lambda_{31}, \lambda_{32}, \lambda_{33}) &\sim \mathrm{Dir}(2, 3, 5).\end{aligned} \tag{15.2}$$

The strength of a Dirichlet prior can be gauged by summing its parameters (Sect. 3.5.3). In the simple Dirichlet-multinomial case, the posterior parameters will be the sum of the prior parameters and the observed cell counts. Thus, the sum of the prior parameters gives an effective sample size for the prior. In this case, the prior for *DKMendel* has effective sample size of ten, thus it would be weighted equally with ten observations (this would be quite mild in comparison with a substantial sample of several thousand, but is fairly large in comparison with the field trial of size 28).

Judging the strength of the hyper-Dirichlet prior is a bit trickier. In principle, it works the same way, summing the parameters in each row gives the effective strength of the Dirichlet distribution for each row. However, when calculating the posterior only those members of the sample for which *DKMendel* = `low` will contribute to the posterior. In this example, we sample

[2] One could posit different priors for different students in the field trial, based on, say, how many courses they had taken in genetics and how many in science in general.

to be divided evenly among the three categories of *DKMendel*. This means that even though each row of Eqs. 15.1 and 15.2 sum to the same value, the prior for *WKInqry* will have about three times the strength of that for *DKMendel*.

Some care must be taken if one of the states of the parent variable is rare. For example, if the expected probability for *DKMendel* = `high` is 0.05, then only 1 out of 20 observations is used update that row of the conditional probability table. In these cases, a much larger sample size is needed to learn much about that value from data. It helps to use structured, parameterized conditional probability matrices like those discussed in Sect. 8.5, because they use both data from other rows and the beliefs about the inter-relationships among probabilities throughout the table to produce smooth and plausible estimates throughout the table.

15.1.2 Specification of Evidence Model Priors

Strictly speaking, the four activities involved in Segment 1 of Biomass are tasks and not task models. The graph fragments used to score them are link models and not evidence models. The common practice is to specify priors at the level of the evidence model parameters, and after observing data, calibrate the posterior parameters for the link model. However, in Biomass there is only one task per task model and hence, only one link model per evidence model and the distinction is purely semantic. The details for the four evidence/link models for the four tasks from the first segment are given below.

Evidence Model 1

Task 1 (EM1) concerned re-expressing José's verbally stated hypothesis about the mode of inheritance in tabular form (Fig. 14.7). A student would use the applet shown in Fig. 14.7 to express José's hypothesis using the table by dragging elements from palette of symbols and terms and dropping them into the table. The work product consisted of a set of lists for each cell of the table, describing which of the elements where dropped into that cell. The observables were all based on applying rules of evidence to the work product. The three *MendModGen* observable variables concerned the degree of correctness of the elements in given cells; for example, on a 1–3 scale, how accurately the dominance relation the student constructed matched José's working hypothesis.

The four *MendModRep* observables concerned the consistency among different portions of the work product. For example, José posited a dominant relationship, but if a student indicates codominance and genotype/phenotype combinations that were consistent with codominance, then the *MendModRep* observable (consistency) gets a high value and the *MendModGen* observable (accuracy) gets a low value. Seven distinct aspects of this solution are captured as values of observable variables, all providing evidence about *DKMendel*.

Note that all seven observables are based on the same work product. Although each of the observables is posited to depend on only one proficiency variable, *DKMendel*, they are probably dependent beyond their relationship through *DKMendel*. We model this situation as described in Sect. 6.2, by introducing the independent context proficiency variable *Context for TM1*, or $Context_{TM1}$. It has two values, *familiar* and *unfamiliar*. $Context_{TM1}$ is an additional parent of all the observables within this task. The form of this EMF is shown in Fig. 14.10.

Quantifying the EMF shown in Fig. 14.10 requires specifying eight conditional probability tables: one for each of the seven observables, and one for the *Context* variable (which is local to the evidence model). The proficiency variable, *DKMendel,* is borrowed from the proficiency model. Its distribution was already specified in the proficiency model, so, we do not need to specify it again here.

Next, we need to choose a parameterization for the conditional probability tables for the observable variables. The experts decided that the influence of the *DKMendel* and *Context* variables should be combined using the compensatory design pattern. Given this, the DiBello-Samejima formula (Almond et al. 2001; this volume, Sect. 8.5) can be used to establish the values in the tables, given the values of certain parameters: an intercept or difficulty parameter, and a slope or discrimination parameter for each parent variable. Thus, to specify this evidence model, we must specify prior distributions for 21 parameters (one intercept and two slope parameters for each of seven observables).

Consider any one of these seven observables, Observable $1j$. To start the construction, we introduce effective theta[3] values for *DKMendel* and $Context_{TM1}$: θ^*_{DKM} and θ^*_{C1}. These are related to the corresponding variables as follows:

$$\theta^*_{DKM} = \begin{cases} 1 & DKMendel = \texttt{high} \\ 0 & DKMendel = \texttt{medium} \\ -1 & DKMendel = \texttt{low} \end{cases} \tag{15.3}$$

$$\theta^*_{C1} = \begin{cases} 1 & Context_{TM1} = \texttt{familiar} \\ -1 & Context_{TM1} = \texttt{unfamiliar} \end{cases} \tag{15.4}$$

The effective theta for a particular combination of parent variable states is given by:

$$\tilde{\theta}_{1j} = \alpha_{1jDKM}\theta^*_{DKM} + \alpha_{1jContext}\theta^*_{C1} + \beta_{1j} \tag{15.5}$$

[3] The calculations in this section use an older algorithm for assigning the effective theta values to parent states than the one given in Sect. 8.5.1. In particular, it uses points that are equally spaced in the interval $[-1, 1]$ rather than equally spaced according to a normal distribution. This is historically accurate as the Biomass work was done before we realized that there was an advantage of the alternative spacing when chaining DiBello–Samejima distributions together.

Recall (Eq. 8.12) that the Samejima graded response model is derived from curves that look like

$$\mathrm{P}(X \geq x_m|\theta) = P^*_m(\theta) = \mathrm{logit}^{-1}(\theta - d_m).$$

Thus, as there are three possible levels for each observable variable, we identify the latent scale by specifying $d_1 = 0$ and $d_2 = 1$. These are treated as fixed in the analysis below.

The experts thought that Observable 1 was particularly easy, and so recommended initial parameter values included $\beta_{11} = +1$ for the intercept[4], along with $\alpha_{11DKM} = 1$ and $\alpha_{11Context} = .5$ for slopes. They did not have information to expect Observables 2 through 7 would be especially hard or easy, so they used an initial value of 0 for $\beta_{12} - \beta_{17}$, and again values of 1 and .5 for the slopes for *DKMendel* and *Context*. Table 15.2 shows the conditional probability table with those parameter values. Later we will be able to compare these initial tables with probabilities that are revised in accordance with information about them in the pilot data.

Table 15.2 Initial conditional distributions for observables 2–7 of Task 1

DKMendel	*Context*		$\mathrm{P}(X = k)$		
θ^*_{DKM}	θ^*_{C1}	$\tilde{\theta}_{11}$	`Low`	`Medium`	`High`
−1	−1	−0.50	0.82	0.11	0.08
−1	1	0.50	0.62	0.20	0.18
0	−1	0.50	0.62	0.20	0.18
0	1	1.50	0.38	0.24	0.38
1	−1	1.50	0.38	0.24	0.38
1	1	2.50	0.18	0.20	0.62

The value θ^*_{DKM} is coded `low`=−1, `medium`=0, `high`=1; θ^*_C is coded `unfamiliar`=−1, `familiar`=1; probabilities calculated via Eq. 15.5 using $\alpha_{1jDKM} = 1$, $\alpha_{1jContext}$ =.5, $\beta_{1j} = 0$, and fixed values $d_1 = 0$ and $d_2 = 1$

A full specification of the model requires a law for the parameters α_{1jDKM}, $\alpha_{1jContext}$, and β_{1j}, where j is an index for the seven observable variables. The first intercept parameter was given the distribution $\beta_{11} \sim \mathrm{N}(1, 1)$, and the rest were given the distribution $\beta_{1j} \sim \mathrm{N}(0, 1)$. The slope parameters $\alpha_{1jDKM} \sim \mathrm{N}^+(1, 1)$ and $\alpha_{1jContext} \sim \mathrm{N}^+(0.5, 1)$ for $j = 1, \ldots, 7$, and where $\mathrm{N}^+(\cdot)$ represents a normal distribution truncated at zero. (These truncated-normal priors for slopes have their maxima at the initial values 1 and .5 respectively, but their means are higher.)

Finally, we need a distribution for $Context_{TM1}$. The context variable is independent of all of the other (unobserved) variables, and is given the dis-

[4] Recall that β_{11} is an intercept, not a difficulty, parameter. Hence, larger values result in higher probabilities of success, i.e., easier tasks.

tribution $\mathit{Context}_{TM1} \sim$ Bernoulli(.5)—specifically, probabilities of .5 for the values of -1 and $+1$. The context variable in this task and all the others in the example are given this fixed distribution. (Specifying any two values is actually sufficient to effect conditional dependence in the CPTs, but centering them around zero and having the values be ± 1 simplifies the interpretation of the parameters in the DiBello–Samejima model.)

Evidence Model 2

Task 2 (EM2) concerned a table (Fig. 14.8) that a student was asked to fill out with several statements about implications of the mode of inheritance. In each case, the student was to indicate if this statement could be confirmed or rejected on the basis of data from the field population alone, from the offspring of matings of known members of the field population, and from the offspring of matings of the next generation after that. For example, it is a common misconception that if there were more tan mice than black mice in the field population, then tan is the expression of a dominant allele. Maybe, but maybe not! The recessive allele could be much more common than the dominant allele. There are three variables in this cluster, posited by our experts to depend conjunctively on *DKMendel* and *WKInqry*, and conditionally dependent beyond these joint influences. The form of this EMF was shown as Fig. 14.11.

The DiBello–Samejima equations for the compensatory conjunctive distribution is more complex than any of those explored in Sect. 8.5, but illustrates the flexibility of that procedure. As before, we need effective thetas for each of the parent variables. We again define θ^*_{DKM} using Eq. 15.3, and we define θ^*_{WKI} by substituting *WKInqry* for *DKMendel* in that equation. For the context variable, θ^*_{C2} is defined by substituting $\mathit{Context}_{TM2}$ for $\mathit{Context}_{TM1}$ in Eq. 15.4.

According to the experts, both domain knowledge and working knowledge about inquiry are required to solve this problem, therefore, the contribution to the effective theta from the proficiency variables should be $\min(\theta^*_{DKM}, \theta^*_{WKI})$. However, there is an additional term for the context effect, the extra dependency introduced because all three observables come from the same work product. Combining these two terms gives the following effective theta for Observable j:

$$\tilde{\theta}_{2j} = \alpha_{2jDKMWKI} \min(\theta^*_{DKM}, \theta^*_{WKI}) + \alpha_{2jContext}\theta^*_{C2} + \beta_{2j} \qquad (15.6)$$

The experts thought that in this task, Observables 1 and 2 were on the hard side, and posited initial values of $\alpha_{21DKMWKI} = 1$, $\alpha_{21Context} = 0.5$ and $\beta_{21} = -0.5$. Plugging these values into Eq. 15.6 yields the conditional probability table shown in Table 15.3.

Again, a complete specification of the evidence model requires a prior for the parameters for all three observable variables. Since the experts thought the first two observables were harder than the third, we set intercept parameters

Table 15.3 Initial conditional distributions for observable 1 of Task 2

DKMendel	*WKInqry*	*Context*		P($X = k$)		
θ^*_{DKM}	θ^*_{WKI}	θ^*_{C2}	$\tilde{\theta}_{11}$	`Low`	`Medium`	`High`
−1	−1	−1	−2.00	0.88	0.07	0.05
−1	0	−1	−2.00	0.88	0.07	0.05
−1	1	−1	−2.00	0.88	0.07	0.05
0	−1	−1	−2.00	0.88	0.07	0.05
1	−1	−1	−2.00	0.88	0.07	0.05
−1	−1	1	−1.00	0.73	0.15	0.12
−1	0	1	−1.00	0.73	0.15	0.12
−1	1	1	−1.00	0.73	0.15	0.12
0	−1	1	−1.00	0.73	0.15	0.12
1	−1	1	−1.00	0.73	0.15	0.12
0	0	−1	−1.00	0.73	0.15	0.12
0	1	−1	−1.00	0.73	0.15	0.12
0	1	−1	−1.00	0.73	0.15	0.12
0	0	1	0.00	0.50	0.23	0.27
0	1	1	0.00	0.50	0.23	0.27
0	1	1	0.00	0.50	0.23	0.27
1	1	−1	0.00	0.50	0.23	0.27
1	1	1	1.00	0.27	0.23	0.50

Both θ^*_{DKM} and θ^*_{WKI} are coded `low` $= -1$, `medium` $= 0$, and `high` $= 1$; θ^*_C is coded `unfamiliar`$=-1$, `familiar`$=1$. Probabilities calculated via Eq. 15.6 using $\alpha_{21DKMWKI} = 1$, $\alpha_{21Context} = .5$, $\beta_{21} = -.5$, and fixed values $d_1 = 0$ and $d_2 = 1$

$\beta_{21}, \beta_{22} \sim \mathrm{N}(-0.5, 1)$, and $\beta_{23} \sim \mathrm{N}(0, 1)$. The slope parameters $\alpha_{2jDKMWKI} \sim \mathrm{N}^+(1, 1)$ and $\alpha_{2jContext} \sim \mathrm{N}^+(0.5, 1)$ for $j = 1, 2, 3$.

Finally, we again need a distribution for $Context_{TM2}$. The context variable is independent of all of the other (unobserved) variables, and as with Evidence Model 1, we give it a fixed Bernoulli(.5) distribution.

Evidence Model 3

Task 3 (EM3) consisted of three multiple-choice questions about implications of forms of dominance. *DKMendel* is the only proficiency variable, and the responses are posited to be conditionally independent. The form of this EMF was shown in Fig. 14.12. Note that because these are multiple choice items without a partial credit option, the middle category is not used for the observables, which are just either `low` or `high`. The Samejima-type graded response model simplifies to a one-parameter logistic model, with only one item location, $d = 0$.

The effective theta for *DKMendel*, θ^*_{DKM} is again coded using Eq. 15.3. Adding a slope and intercept produces the effective theta for each item. The experts said all three are items of typical difficulty, so the initial intercept parameters are 0, and the initial slope parameters are 1. (Thus the effective theta for the observables is identical to the effective theta for *DKMendel a priori*, but the model can allow the difficulty and discrimination of the item to be adjusted to better fit the observed data.) Equation 15.7 shows expression for the effective theta. Table 15.4 gives conditional response probabilities that correspond to our initial values for the intercept and slope.

$$\tilde{\theta}_{3j} = \alpha_{3jDKM}\theta^{*}_{DKM} + \beta_{3j} \tag{15.7}$$

Table 15.4 Initial conditional probability distributions for all three observables of Task 3

DKMendel		P($X = k$)	
θ^{*}_{DKM}	$\tilde{\theta}_{11}$	`Low`	`High`
-1	-1.00	0.73	0.27
0	0.00	0.50	0.50
1	1.00	0.27	0.73

The value θ^{*}_{DKM} is coded `low`$=-1$, `high`$=1$

Again, priors are required for the parameters. The chosen priors are $\beta_{3j} \sim \mathrm{N}(0,1)$ for the intercept and $\alpha_{3jDKM} \sim \mathrm{N}^{+}(1,1)$ for the slope. The three observables are conditionally independent given the proficiency variable, so there is no context variable for this evidence model.

Evidence Model 4

Task 4 (EM4) asked what José should do next, after having formalized his hypothesis about the mode of inheritance of hair color based on the field population (14.9). There is just one observable variable. The key to its solution is a central tenet of inquiry in transmission genetics: Simply generating a hypothesis that is consistent with a field population is not sufficient to conclude a mode of inheritance; one must carry out crosses, test the hypothesis, and revise if necessary. Our experts indicated that a student must know at least a bit about the Mendelian model to respond to this question, but the quality of the response would depend mainly on the ability to apply inquiry skills in this domain. The evidence model fragment therefore must reflect an inhibition relationship, in which a student must be above the `low` level of *DKMendel* to have chances at making a high-quality response that increase with increasing levels of *WKInqry*.

Figure 14.13 shows the structure of the EM fragment for Evidence Model 4, where *DKMendel* is an inhibitor of *WKInqry*—note the stop sign as a symbol for the structure of the distribution. Again, we assign effect theta values for

WKInqry by substituting it for *DKMendel* in Eq. 15.3. The effective theta for the observable then becomes:

$$\tilde{\theta} = \begin{cases} -1\alpha_{4,1} + \beta_{4,1} & \mathit{DKMendel} = \texttt{low} \\ \theta^*_{WKI}\alpha_{4,1} + \beta_{4,1} & \mathit{DKMendel} > \texttt{low} \end{cases} \tag{15.8}$$

Again the effective theta is entered into the DiBello–Samejima model with d_1 and d_2 fixed at 0 and 1. Table 15.5 gives a set of conditional probabilities that are obtained when $\alpha_{4,1} = 1$ and $\beta_{4,1} = 0$.

Table 15.5 Initial conditional distribution for observable 1 of Task 4

DKMendel	*WKInqry*		$\mathrm{P}(X = k)$		
θ^*_{DKM}	θ^*_{WKI}	$\tilde{\theta}_{11}$	`Low`	`Medium`	`High`
−1	−1	−1.00	0.62	0.20	0.18
−1	0	−1.00	0.62	0.20	0.18
−1	1	−1.00	0.62	0.20	0.18
0	−1	−1.00	0.62	0.20	0.18
1	−1	−1.00	0.62	0.20	0.18
0	0	0.00	0.38	0.24	0.38
1	0	0.00	0.38	0.24	0.38
0	1	1.00	0.18	0.20	0.62
1	1	1.00	0.18	0.20	0.62

Both θ^*_{DKM} and θ^*_{WKI} are coded `low` $= -1$, `medium` $= 0$, and `high` $= 1$

Finishing the model requires prior distributions for the two parameters. Following the patterns used for the other distributions, we use $\beta_{4,1} \sim \mathrm{N}(0,1)$ and $\alpha_{4,1} \sim \mathrm{N}^+(1,1)$.

15.1.3 Summary Statistics

Table 15.6 gives summary statistics for the prior distributions described above. These statistics are based on 50,000 draws from the prior using the Gibbs sampler, without the response data. Section 15.3.1 will compare these with posteriors obtained after incorporating information from pilot testing. Note that the prior distributions for all the item slopes of a given type are identical, while the item difficulties vary in selected cases in accordance with the experts' judgments of their difficulties.

We will also calculate an expected proficiency level or EAP (expected a posteriori) score for each student. This is obtained by assigning a numeric value to each of the proficiency levels (`low` $= 1$, `medium` $= 2$, `high` $= 3$)[5] and then taking the expectation over the probability that the student is at each

[5] This particular score makes the implicit assumption that the difference between `low` and `medium` is the same size as the difference between `medium` and `high`.

proficiency state. Before looking at a student's response data and using the prior means for the model parameters, the prior EAP score for each student is 2.00; that is, before seeing anyone's performance, our best guess is that they are at the `medium` level. The wide dispersion of the Dirichlet priors on proficiency variables, though, says we are quite willing to believe they might be `low` or `high`, after seeing some responses that suggest this.

15.2 Pilot Testing

15.2.1 A Convenience Sample

Part of building a calibration sample is crafting an argument for why the sampled individuals are representative of the population of interest. Random sampling from that population builds the strongest possible argument. Often, however, random sampling is prohibitively expensive or impractical, or permissions are difficult to obtain. In those cases, researchers make do with less optimal samples. However, they are still obliged to describe the methods for gather the sample, so that future readers can judge the suitability of the sample for the designated purpose.

The population of interest for Biomass was high school students taking an AP Biology class; that is, primarily high school seniors with a strong science background. Furthermore, as the interim assessment was meant to take place within the context of the class unit on genetics, we needed students who had at least a little bit of familiarity with the concepts and terminology of genetics.

When collecting a sample for calibrating Biomass, we were faced with several problems. First, as we had almost no budget, we knew that we would need to make a number of compromises. Second, as it was summer, we could not simply walk into an AP biology class and take the sample. Third, the sampled individuals would not necessarily have the correct background in genetics. This last one we overcame by creating a mini tutorial providing enough background in genetics to try be able to tackle the first segment of the Mice tasks. However, this in itself created another problem: tutorial and tasks together took about 20 min to complete. As we only had Biomass running on one computer[6], this severely limited our throughput.

Our first attempt at building a sample was to visit a summer program in science enrichment for high school students at a nearby college. However,

[6] Nominally, as Biomass is a web application, one computer, the server, can supply the assessment to many other computers, the clients, which only need a web browser. However, in practice, firewall and security restrictions made it difficult to run Biomass in a computer lab (without a lot of additional administrative effort). Also, Biomass required a fairly large monitor, or else the student had to do a lot of scrolling. In practice, these restrictions forced us to mostly use one computer for testing.

Table 15.6 Summary statistics of parameter prior distributions

Parameter		Prior mean (SD)
Evidence model 1		
Slopes for *DKMendel*	$\alpha_{11DKM}-\alpha_{17DKM}$	1.29 (0.79)
Slopes for $Context_{TM1}$	$\alpha_{11Context}-\alpha_{17Context}$	1.00 (0.70)
Intercepts	β_{11}	1.00 (1.00)
	$\beta_{12}-\beta_{17}$	0.00 (1.00)
Evidence model 2		
Slopes for conjunction	$\alpha_{21}-\alpha_{23}$	1.29 (0.79)
Slopes for $Context_{TM2}$	$\alpha_{21Context}-\alpha_{23Context}$	1.00 (0.70)
Intercepts	β_{21},β_{22}	−0.50 (1.00)
	β_{23}	0.00 (1.00)
Evidence model 3		
Slopes for *DKMendel*	$\alpha_{31}-\alpha_{33}$	1.29 (0.79)
Intercepts	$\beta_{31}-\beta_{33}$	0.00 (1.00)
Evidence model 4		
Slope	$\alpha_{4,1}$	1.29 (0.79)
Intercept	$\beta_{4,1}$	0.00 (1.00)
Proficiency model		
Distribution of *DKMendel*	λ_1	0.30 (0.14)
	λ_2	0.40 (0.15)
	λ_3	0.30 (0.14)
Conditional distribution of *WKInqry* given *DKMendel*	λ_{11}	0.50 (0.15)
	λ_{12}	0.30 (0.14)
	λ_{13}	0.20 (0.12)
	λ_{21}	0.30 (0.14)
	λ_{22}	0.40 (0.15)
	λ_{23}	0.30 (0.14)
	λ_{31}	0.20 (0.12)
	λ_{32}	0.30 (0.14)
	λ_{33}	0.50 (0.15)
Individual student, before responses		
DKMendel	λ_1	2.00 (0.77)
WKInqry	λ_2	2.00 (0.82)

we only were able to get 1 hour of testing time, and hence we only got three students from this sampling effort.

Our second attempt involved going to a local comic book shop that had an open game night on Wednesday evenings. People would randomly drop into that location all afternoon and evening, and the shop tended to be frequented by students from the local high school and the local college; thus, it was at least likely to attract students close to the correct age range.

The individuals who participated in this second data collection effort included some people who were at the extreme ends of the data collection range, and who provided some interesting anecdotal evidence about how Biomass was performing. The first was a college graduate who was at the time working as a pharmacy technician. She drew a Punnett square on a sheet of scrap paper while attempting to solve the first task. This was an important knowledge representation in genetics, a class of observable we used in later segments, but not the first one. It was interesting that she retained these knowledge representations from her earlier training. The second was a middle school student who begged to be allowed to try even though he was below the age cutoff. He had recently completed a unit in genetics at middle school, and thought that he really knew the material. And he did—at least in terms of vocabulary and representations. But the tasks that required a great deal of working knowledge, particularly knowledge of the scientific inquiry process, really stumped him. This provides anecdotal support for splitting domain knowledge and working knowledge in the proficiency model.

The final source of data was a number of summer interns at ETS. These were students in graduate school, who varied greatly as to the number of science courses they had taken in high school and as undergraduates. On top of that, many had been high school students or undergraduates in a different country and had taken science course in languages other than English, thus, they may not have been familiar with the English names for various terms from genetics.

Altogether, these three data collection efforts yielded a field trial sample of 28 individuals (Table 15.7). Calling this sample representative of the desired population is an extreme stretch. We would be reluctant to make any strong conclusions about Biomass, or actually adjust the parameters of the models, on the basis of this rather haphazard field trial. On the other hand, this sample is useful to test the machinery for performing the parameter update. Unlike simulated data, it is difficult to predict how the data from this field trial will behave. Adjusting on the basis of these numbers should make the Biomass more suitable for the population of which this field trial is representative, whatever that may be. The next sections overlook the obvious limitations in the data collection and proceed to analyze the data.

Table 15.7 Observed responses

Student	x_{11}	x_{12}	x_{13}	x_{14}	x_{15}	x_{16}	x_{17}	x_{21}	x_{22}	x_{23}	x_{31}	x_{32}	x_{33}	x_4	Mean
1	3	1	1	1	1	1	1	1	1	1	3	3	1	2	1.50
2	1	1	1	1	1	1	1	1	1	1	1	1	1	3	1.14
3	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1.00
4	3	1	1	1	1	1	1	1	1	1	1	1	3	2	1.36
5	3	3	3	1	3	1	1	1	1	1	1	3	3	3	2.00
6	3	1	1	1	1	1	1	1	1	1	3	3	1	3	1.57
7	3	1	1	1	1	1	1	1	1	2	1	3	1	3	1.50
8	3	1	3	1	3	1	1	1	1	2	1	3	3	2	1.86
9	1	1	1	1	1	1	1	1	1	1	1	1	1	2	1.07
10	3	1	1	1	1	1	1	1	1	1	1	3	3	2	1.50
11	3	3	3	1	3	1	1	2	1	1	1	3	3	2	2.00
12	3	3	3	1	3	1	1	1	1	2	1	3	1	2	1.86
13	3	3	3	1	3	1	1	1	1	2	3	3	3	3	2.21
14	3	3	3	1	3	1	1	1	1	1	1	3	1	3	1.86
15	3	1	1	1	1	1	1	2	1	1	3	3	3	2	1.71
16	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1.00
17	3	3	1	1	1	1	1	3	1	3	3	3	3	3	2.14
18	3	1	3	1	3	1	1	1	1	3	3	1	3	3	2.00
19	3	3	1	1	1	1	1	1	1	1	3	1	1	2	1.50
20	1	1	1	1	1	1	1	1	1	1	1	3	1	2	1.21
21	3	3	3	1	3	1	1	1	1	1	1	1	1	1	1.57
22	3	3	3	1	3	1	1	3	1	3	3	3	3	3	2.43
23	3	1	1	1	1	1	1	3	1	2	3	3	3	3	1.93
24	1	1	1	1	1	1	1	1	1	2	3	3	3	3	1.64
25	3	3	3	1	3	1	1	1	1	2	3	3	3	2	2.14
26	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1.00
27	3	1	1	1	1	1	1	1	1	2	3	3	1	3	1.64
28	3	1	1	1	1	1	1	1	1	1	3	3	3	2	1.64
Mean	2.50	1.71	1.71	1.00	1.71	1.00	1.00	1.29	1.00	1.50	1.93	2.36	2.00	2.29	1.64

Responses are coded: `low` = 1, `medium` = 2, `high` = 3

15.2.2 Item and other Exploratory Analyses

Before spending a lot of time with fancy analysis, it is worth looking at some simple descriptive statistics that tell us in a basic way how the assessment is working. Whether the eventual method for scoring is based on classical test theory, IRT, or Bayes nets, the same type of descriptive statistics can be used in item analysis: number right (or in this case, sum of the partial credit scores) statistics for each examinee, and marginal distributions, or $P+$, for each observable outcome variable.

Table 15.7 codes the responses `low` = 1, `medium` = 2, and `high` = 3. Note the dearth of '2' responses, except for the last observable. In most cases, the students did well or poorly on most aspects of the tasks, with few performances of intermediate quality—even though the average of all the responses was 1.62, just about in the middle. For the observables from Task 3, $x_{3,1}$, $x_{3,2}$, and $x_{3,3}$, there are no 2 responses because these were multiple-choice items scored dichotomously; incorrect responses were scored as `low` and correct responses as `high`. For the other observables, it is worth going back and checking to make sure that the evidence rules were implemented correctly. If, in fact, all of the work products produced correctly reflect `low` and `high` performance, it may be necessary to reconsider what partial credit is for this task (see Exercise 15.12). This is one place where having a sample of students who are studying the material in Biomass would be particularly important: There is apt to be more instability and partial understanding among students just learning and applying concepts than ones who have either completed their study or not yet started it. It is possible that the '2's missing from this sample are particularly useful in formative assessment, and that is precisely why our experts who taught this material crafted them for the assessment. We note too that the students also showed a great range in performance: Nothing but the lowest responses from Students 3 and 26, to a majority of 3's for Student 22.

It is often worth checking data records on outlying observations like these. A quick check of the logs reveals that Student 26 logged in with the user name "try." Thus, rather than a real subject attempting the assessment, this observation may represent an administrator testing or demonstrating the system. As there are only 28 students, this record was retained for the analysis. However, such spurious records are common in field trials and real data must be carefully cleaned.

The items range from very difficult (nobody did better than the lowest response on $x_{1,4}$, $x_{1,6}$, $x_{1,7}$, and $x_{2,2}$) to very easy (most students answered $x_{1,1}$ correctly). Did these results accord with the experts' prior expectations? Yes, for the most part. There were three items for which they had opinions other than "typical." They expected Observable 1 of Task 1 to be easier, and it turned out to be the easiest one in the study. They expected Observables 1 and 2 of Task 2 to be harder than typical, and they were. But the four observables noted above on which every student was rated `low` were not expected to be different from typical. This may be due again to the fact that the students in the field trial are not exactly the same as the ones the experts had in mind as a target population. They thought about how hard a task would be for a student who had been working through a unit on this material, and would be familiar with the notation and expectations used in the prototype. Our field-trial students did not have this advantage, which could differ from one task to the next. A second plausible explanation is a coding error in the implementation of the evidence rules, and it was worth going back to check the work products to see if they all truly reflect low performance (they did). Another possible explanation is that there is some problem with the task design, perhaps unclear wording of the directions, that caused all of the field trial subjects to miss this particular aspect of the task. Small-sample trials, where each subject is watched or video recorded, and offers thoughts during or after the experience, are particularly useful in early stages of developing interactive tasks like these before collecting large calibration samples.

These kinds of statistics are useful even with a small pilot test sample that is chosen for convenience rather than carefully chosen to be representative of the population. As seen above, relatively simple statistics can be used to spot problems with the design and/or implementation of the tasks and evidence rules, and thus provide an important step in the quality control of the assessment.

15.3 Updating Based on Pilot Test Data

We have spent some effort to build a Bayesian probability framework that expresses the experts' beliefs about the key relationships between knowledge and performance in the Biomass tasks. The probability distributions express the qualitative structure of the relationships, and task and examinee parameters express the quantitative relationships within that structure. Note that the experts' opinion about the parameters is expressed through a collection of prior laws which describe not only the experts' best guess as to the parameter values, but also their degree of certainty about those estimates. Overlooking its limits, we can use the field trial responses to show how the Bayesian machinery uses data to update the model.

This section describes three applications of the field trial data to the Biomass model. Section 15.3.1 describes using the Bayesian machinery to update the parameter values. Section 15.3.2 looks at some statistics for model fit. Finally, Sect. 15.3.3 shows a simple (hypothetical) validity check.

15.3.1 Posterior Distributions

WinBUGS (Thomas et al. 1992; Lunn et al. 2000) can calculate the posterior distribution, using the MCMC approach of Sect. 9.6. To do this, the prior distribution given in Sect. 15.1 must be expressed as BUGS code.[7] After reading the outcome data shown in Table 15.7, BUGS generates a specified number draws from the posterior distribution. As there are only 28 response vectors, the MCMC algorithm runs quickly and it is easy to generate many samples from the posterior. Tables 15.9 and 15.10 give summaries of posterior distributions for parameters conditional on this data, along with the priors to facilitate comparison. We will look at population parameters, evidence-model parameters, and student proficiency parameters in turn.

The prior distributions we posited for the parameters were fairly mild, but the sample size was also fairly small. The prior distributions—that is, the substantive theory, the task design, the structures relating performance to proficiency, and initial beliefs about direction and strength of evidence—are

[7] The analysis described in this section was originally done in WinBUGS (Mislevy et al. 2002a); however, during the editing of this chapter we reran the MCMC simulations using JAGS (Plummer 2012), which is essentially similar.

thus essential to reasoning about students from their performance. We will see that within this structure, estimates for some parameters were substantially revised on the basis of even the small calibration sample, while others changed very little. In addition to the means of the parameters' posterior laws, we will look at the variances to quantify the relative impact of the data on various parameters. As the precision of the law (the reciprocal of the variance) quantifies the amount of information in the distribution, the increase in precision provides an indication of how much information was gained from the sample. The percentage increase in precision is calculated as follows:

$$\%\ \text{Increase in precision} = 100 \times \frac{(\text{posterior SD})^{-2} - (\text{prior SD})^{-2}}{(\text{prior SD})^{-2}}.$$

A value of zero indicates no new information, while a value of 100 means there was twice as much information about a parameter after seeing the data than before seeing it.

Table 15.8 Summary statistics of prior and posterior population parameter distributions

Parameter		Prior		Posterior		% Increase
		Mean	(SD)	Mean	(SD)	in precision
Distribution of	λ_1	0.30	(0.14)	0.30	(0.10)	97
DKMendel	λ_2	0.40	(0.15)	0.47	(0.12)	48
	λ_3	0.30	(0.14)	0.22	(0.11)	65
Conditional	λ_{11}	0.50	(0.15)	0.50	(0.15)	0
distribution of	λ_{12}	0.30	(0.14)	0.30	(0.14)	0
WKInqry	λ_{13}	0.20	(0.12)	0.20	(0.12)	2
given	λ_{21}	0.30	(0.14)	0.32	(0.15)	−11
DKMendel	λ_{22}	0.40	(0.15)	0.41	(0.15)	−4
	λ_{23}	0.30	(0.14)	0.27	(0.13)	8
	λ_{31}	0.20	(0.12)	0.19	(0.11)	13
	λ_{32}	0.30	(0.14)	0.31	(0.14)	−5
	λ_{33}	0.50	(0.15)	0.50	(0.15)	0

Table 15.8 shows the posterior means and variances and the percentage increase in precision in the laws for the proficiency model parameters. There are moderate increases in precision for the distribution of *DKMendel*, since every student contributes something, with information from all of their responses (sometimes confounded with information about *WKInqry*). The direction of the change is to move belief about the distribution of *WKInqry* in the sample from `high` to `medium`; that is, the mean of the law for λ_3 decreases from .30 to .22, while the mean for λ_2 increases from .40 to .47. The assessment was more difficult for the sample than the priors anticipated.

Table 15.9 Summary statistics of item parameter distributions

Parameter		Prior		Posterior		% Increase
		Mean	(SD)	Mean	(SD)	in precision
Evidence model 1						
Slopes for *DKMendel*	α_{11DKM}	1.29	(0.79)	2.08	(0.75)	12
	α_{12DKM}	1.29	(0.79)	1.22	(0.71)	26
	α_{13DKM}	1.29	(0.79)	1.02	(0.64)	53
	α_{14DKM}	1.29	(0.79)	0.90	(0.61)	66
	α_{15DKM}	1.29	(0.79)	1.02	(0.64)	52
	α_{16DKM}	1.29	(0.79)	0.90	(0.61)	65
	α_{17DKM}	1.29	(0.79)	0.90	(0.62)	65
Slopes for $Context_{TM1}$	$\alpha_{11Context}$	1.00	(0.70)	1.14	(0.53)	75
	$\alpha_{12Context}$	1.00	(0.70)	1.93	(0.52)	77
	$\alpha_{13Context}$	1.00	(0.70)	2.96	(0.61)	32
	$\alpha_{14Context}$	1.00	(0.70)	0.62	(0.43)	172
	$\alpha_{15Context}$	1.00	(0.70)	2.95	(0.61)	30
	$\alpha_{16Context}$	1.00	(0.70)	0.61	(0.42)	172
	$\alpha_{17Context}$	1.00	(0.70)	0.62	(0.43)	168
Intercepts	β_{11}	1.00	(1.00)	2.60	(0.58)	196
	β_{12}	0.00	(1.00)	0.04	(0.53)	255
	β_{13}	0.00	(1.00)	0.05	(0.62)	165
	β_{14}	0.00	(1.00)	−2.54	(0.60)	175
	β_{15}	0.00	(1.00)	0.06	(0.61)	166
	β_{16}	0.00	(1.00)	−2.54	(0.61)	170
	β_{17}	0.00	(1.00)	−2.54	(0.61)	173
Evidence model 2						
Slopes for conjunction	α_{21}	1.29	(0.79)	2.07	(0.79)	1
	α_{22}	1.29	(0.79)	1.19	(0.70)	30
	α_{23}	1.29	(0.79)	1.46	(0.68)	34
Slopes for $Context_{TM2}$	$\alpha_{21Context}$	1.00	(0.70)	0.81	(0.56)	56
	$\alpha_{22Context}$	1.00	(0.70)	0.61	(0.44)	143
	$\alpha_{23Context}$	1.00	(0.70)	0.62	(0.45)	144
Intercepts	β_{21}	−0.50	(1.00)	−1.12	(0.61)	166
	β_{22}	−0.50	(1.00)	−2.59	(0.65)	136
	β_{23}	0.00	(1.00)	−0.19	(0.49)	314
Evidence model 3						
Slopes for *DKMendel*	α_{31}	1.29	(0.79)	1.91	(0.74)	14
	α_{32}	1.29	(0.79)	1.85	(0.72)	23
	α_{33}	1.29	(0.79)	1.89	(0.74)	16
Intercepts	β_{31}	0.00	(1.00)	0.01	(0.50)	306
	β_{32}	0.00	(1.00)	1.00	(0.50)	298
	β_{33}	0.00	(1.00)	0.17	(0.50)	308
Evidence model 4						
Slope	$\alpha_{4,1}$	1.29	(0.79)	0.64	(0.42)	252
Intercept	$\beta_{4,1}$	0.00	(1.00)	1.14	(0.39)	565

Table 15.10 Prior and posterior expected proficiency levels

	DKMendel			*WKInqry*		
Student	Prior Mean (SD)	Posterior Mean (SD)	% Increase in precision	Prior Mean (SD)	Posterior Mean (SD)	% Increase in precision
1	2.00 (0.78)	2.01 (0.50)	141	2.00 (0.81)	1.82 (0.78)	8
2	2.00 (0.78)	1.06 (0.24)	948	2.00 (0.81)	1.69 (0.78)	10
3	2.00 (0.78)	1.03 (0.17)	2020	2.00 (0.81)	1.70 (0.78)	9
4	2.00 (0.78)	1.66 (0.53)	113	2.00 (0.81)	1.79 (0.79)	7
5	2.00 (0.78)	2.10 (0.48)	165	2.00 (0.81)	1.61 (0.73)	25
6	2.00 (0.78)	2.06 (0.46)	187	2.00 (0.81)	1.61 (0.73)	26
7	2.00 (0.78)	1.85 (0.46)	187	2.00 (0.81)	1.86 (0.75)	19
8	2.00 (0.78)	1.91 (0.51)	129	2.00 (0.81)	2.03 (0.76)	16
9	2.00 (0.78)	1.04 (0.19)	1610	2.00 (0.81)	1.71 (0.78)	9
10	2.00 (0.78)	2.00 (0.50)	140	2.00 (0.81)	1.83 (0.79)	7
11	2.00 (0.78)	2.17 (0.48)	155	2.00 (0.81)	2.22 (0.69)	38
12	2.00 (0.78)	1.82 (0.49)	146	2.00 (0.81)	2.01 (0.76)	14
13	2.00 (0.78)	2.47 (0.51)	127	2.00 (0.81)	1.91 (0.73)	23
14	2.00 (0.78)	1.79 (0.49)	147	2.00 (0.81)	1.63 (0.74)	20
15	2.00 (0.78)	2.46 (0.52)	125	2.00 (0.81)	2.28 (0.68)	43
16	2.00 (0.78)	1.03 (0.16)	2183	2.00 (0.81)	1.71 (0.78)	8
17	2.00 (0.78)	2.86 (0.35)	390	2.00 (0.81)	2.71 (0.51)	159
18	2.00 (0.78)	2.00 (0.46)	185	2.00 (0.81)	1.98 (0.73)	25
19	2.00 (0.78)	1.78 (0.58)	77	2.00 (0.81)	1.80 (0.79)	7
20	2.00 (0.78)	1.19 (0.39)	289	2.00 (0.81)	1.73 (0.78)	7
21	2.00 (0.78)	1.29 (0.46)	179	2.00 (0.81)	1.78 (0.80)	3
22	2.00 (0.78)	2.73 (0.44)	204	2.00 (0.81)	2.64 (0.54)	127
23	2.00 (0.78)	2.59 (0.50)	143	2.00 (0.81)	2.46 (0.62)	75
24	2.00 (0.78)	2.03 (0.43)	219	2.00 (0.81)	1.89 (0.74)	21
25	2.00 (0.78)	2.47 (0.52)	121	2.00 (0.81)	2.13 (0.72)	26
26	2.00 (0.78)	1.03 (0.16)	2150	2.00 (0.81)	1.71 (0.79)	7
27	2.00 (0.78)	2.10 (0.44)	212	2.00 (0.81)	1.88 (0.74)	22
28	2.00 (0.78)	2.37 (0.53)	113	2.00 (0.81)	1.83 (0.76)	14

The values for the examinee priors are means and standard deviations of the student proficiency variables coded `high` = 3, `medium` = 2, and `low` = 1

There is virtually no change in the conditional distributions of *WKInqry* given *DKMendel*. One reason is related to the effective size of the sample for the conditional distribution. As the subjects are split over three different possible values of *DKMendel*, the effective sample size is more like 9 subjects than 28. Furthermore, the value of *DKMendel* is not known with certainty, which further reduces the effective sample size (Mislevy 1984). A second reason is related to the test length. While almost all observables provide information about *DKMendel*, only observables from Tasks 2 and 4 provide information about *WKInqry* (an effective test length of four observables compared to 14 for *DKMendel*). Another problem comes with the way that *DKMendel* and *WKInqry* interact to influence the observables in the first four tasks. In all cases where both variables are present, the relationship type is conjunctive (the inhibitor relationship is a special type of conjunctive relationship). Unless *DKMendel* is at least at the `medium` level, then such tasks provide little information about *WKInqry*.

Task parameter posteriors showed means that departed significantly from the priors. The prior means for the slopes for Context variables, for example, were initially all at 1.00; their posterior means ranged from 0.61 up to 2.96, which indicate from small conditional dependence for some observables to quite substantial. The prior means for the slopes for proficiency variables were initially 1.29, while their posterior means ranged from 0.64 to 2.08. Intercept parameter means, which were initially at 0 for typical items, ranged from -2.5 for items on which no one succeeded, up to 2.60, for the first observable on Task 1, where most of the students did well.

Before looking more closely at each evidence model, we note from the increase in precision values that the sample data provided notably more information about intercept parameters than it did about slope parameters. Precision for intercepts increased between 165 and 565 %, or one-and-a-half to five-and-a-half times as much information came from the data as from the priors. This is consistent with item parameter estimation in IRT, where intercepts and threshold parameters are easier to estimate than item slopes.

Task 1 was representing the mode of inheritance for mice coat color. The seven observables in Evidence Model 1 concerned aspects of correctness and consistency in a student's re-expression of the mode of inheritance for coat color.The experts thought the first item would be easier than average, but did not express expectations other than "typical" for the other items. In the pilot data, Observable 1 was in fact quite easy, and the posterior mean for its intercept moved substantially, from 1.00 to 2.60.

Observables 4, 6, and 7 were quite difficult for these students: *nobody* got a score above 1. The means for their intercepts went from 0.00 to −2.54. This is a problem that we already observed during the item analysis (Sect. 15.2.2): nobody in the field trial got an observed outcome other than `low`. Perhaps there is a problem with the task design (e.g., instructions are unclear) or the evidence rules, to be investigated with user studies.

The remaining three observables, 2, 3, and 5, were about half and half 1s and 3s. The posterior means of their intercepts were all around 0, nearly the same as their prior means, but with about 170 % more precision. Even a small data set was enough to move the posteriors for the intercepts notably to reflect their apparent difference in difficulty.

The distributions for the slopes for *DKMendel* didn't change much at all, except for Observation 1; its mean moved from 1.29 to 2.08. Not many students missed this item, but the ones who did didn't do well on other tasks involving *DKMendel* either.

The slopes for $Context_{TM1}$ showed an interesting pattern. They were very high for the three middle-difficulty items, as students tended to do quite well or quite poorly on all three—much less variation than would have been expected had the responses been conditionally independent given *DKMendel*. This is

Table 15.11 Revised conditional distributions for observable 3 of Task 1

DKMendel	*Context*		P($X = k$)		
θ^*_{DKM}	θ^*_{C1}	$\tilde{\theta}_{11}$	`Low`	`Medium`	`High`
−1	−1	−3.10	0.96	0.03	0.02
−1	1	−0.67	0.66	0.18	0.16
0	−1	−1.17	0.76	0.13	0.10
0	1	1.26	0.22	0.21	0.57
1	−1	0.75	0.32	0.24	0.44
1	1	3.19	0.04	0.06	0.90

Equation 15.5 with $\alpha_{13DKM} = 1.93$, $\alpha_{13Context} = 1.22$, and $\beta_{13} = 0.04$

probably due to understanding or not understanding the particulars of this task, above and beyond Mendelian Knowledge. (The $Context_{TM1}$ slopes were much lower for the three observables that nearly everyone missed and for the one that most students got right, in a manner less strongly related to the middling three conditionally dependent items.) This is potentially a problem, as it may greatly decrease the amount of information that comes from this task. However, in this case it may be a problem with the sample. Recall that Task 1 involves filling out a chart to correspond to the mode of inheritance for the mice. Examinees who understood what was required in this task, could likely do all parts of the task. Examinees who didn't understand the representation may have just had difficulty figuring out what was required. Biomass was intended to be used in the context of the a Biology class where such representations would appear in the course text and in classroom work before the students were expected to use them. Thus, if the field data were more representative, this problem might not appear. If it did, it might be better to drop or merge the observables.

The slope and intercept parameter values determine the entries in the conditional probability matrices for Evidence Model 1. Observables 3 and 4 of Task 1 started with the same initial conditional probability tables (Table 15.2). Tables 15.11 and 15.12 contain the revised conditional probabilities for these observables. They are calculated through the DiBello–Samejima structure with the posterior means of their respective task parameters. The revised conditional distributions for Observable 3 show it is much easier than Observable 4. Even a student with the high Mendelian Knowledge and familiarity with this task context is only modeled as having 12 % chances of getting a 3 on Observable 4. Also, in Table 15.11 the differences in the rows with the same *DKMendel* value but different *Context* values show how it is expressed that Observable 3 is conditionally associated strongly with other observables within Task 1.

Task 2 is the population attribute table (Fig. 14.8), and Evidence Model 2 contains the three observables that indicate whether, in each of three columns (kinds of populations), a student correctly identified all inferences that could

Table 15.12 Revised conditional probability table for observable 4 of Task 1

DKMendel	*Context*		P($X = k$)		
θ^*_{DKM}	θ^*_{C1}	$\tilde{\theta}_{1,3}$	`Low`	`Medium`	`High`
-1	-1	-4.06	0.98	0.01	0.01
-1	1	-2.82	0.94	0.03	0.02
0	-1	-3.16	0.96	0.03	0.02
0	1	-1.93	0.87	0.08	0.05
1	-1	-2.27	0.91	0.06	0.04
1	1	-1.03	0.74	0.15	0.12

Equation 15.5 with $\alpha_{14DKM} = 0.90$, $\alpha_{14Context} = 0.62$, and $\beta_{14} = -2.544$

be drawn from a given kind of population, not all but the critical ones, or missing the critical ones (`high`, `medium`, and `low` levels of performance respectively).

The intercept posteriors reflected increased precision, with means moving to indicate the success of the students in identifying the inferential properties of the mice populations. The three observables of Task 2 showed slightly increased slopes for the conjunction of *DMendel* and *WKInqry* but with little increase in precision, and lower slopes for *Context*$_{TM2}$. It is intriguing to see that evidence is particularly weak for the slope parameters of the conjunction of *DKMendel* and *WKInqry* in Evidence Model 2. Note that the combination of `high` on both *DKMendel* and *WKInqry* is rare in the field test population. Further investigation (in particular, a better sample) is needed to determine whether this is a pervasive characteristic of combinations such as conjunctions and disjunctions.

The revised conditional probability matrix for Observable 1 of Task 2 is shown in Table 15.13, to be compared with the initial probabilities shown in Table 15.3. This is another difficult item, although not as difficult as Observable 4 of Task 1. The highest performing students succeeded on this task. The final row of the conditional probability table reflects that high performance here is most likely to arise from high levels of understanding the Mendelian model, inferential reasoning from the different kinds of populations, *and* familiarity with the context—that is, the situations and representations reflected in the population attribute table.

Task 3 was three multiple-choice items concerning aspects of the Mendelian model. The three observables are right/wrong indicators of correctness, and are modeled as conditionally independent given *DKMendel.* The evidence model fragments are based on the discrete two-parameter logistic model. This task is thus most similar to familiar IRT modeling of unidimensional multiple-choice items. The posterior means of the slope parameters are higher than the prior means (nearly 2 in each case), although not estimated very precisely. The posterior means for their intercepts reflect medium difficulty for these students, and posterior precision increased by 300 % over the prior.

Table 15.13 Revised conditional distributions for observable 1 of Task 2

DKMendel	*WKInqry*	*Context*		P($X = k$)		
θ^*_{DKM}	θ^*_{WKI}	θ^*_{C2}	$\tilde{\theta}_{11}$	`Low`	`Medium`	`Migh`
−1	−1	−1	−4.00	0.98	0.01	0.01
−1	0	−1	−4.00	0.98	0.01	0.01
−1	1	−1	−4.00	0.98	0.01	0.01
0	−1	−1	−4.00	0.98	0.01	0.01
1	−1	−1	−4.00	0.98	0.01	0.01
−1	−1	1	−2.38	0.92	0.05	0.03
−1	0	1	−2.38	0.92	0.05	0.03
−1	1	1	−2.38	0.92	0.05	0.03
0	−1	1	−2.38	0.92	0.05	0.03
1	−1	1	−2.38	0.92	0.05	0.03
0	0	−1	−1.93	0.92	0.05	0.03
0	1	−1	−1.93	0.92	0.05	0.03
0	1	−1	−1.93	0.92	0.05	0.03
0	0	1	−0.31	0.58	0.21	0.21
0	1	1	−0.31	0.58	0.21	0.21
0	1	1	−0.31	0.58	0.21	0.21
1	1	−1	0.13	0.47	0.24	0.30
1	1	1	1.75	0.15	0.17	0.68

Calculated via Eq. 15.6 using $\alpha_{21DKMWKI} = 2.07$, $\alpha_{21Context} = 0.81$, $\beta_{21} = -1.12$, and fixed values $d_1 = 0$ and $d_2 = 1$

These items provide a good deal of evidence about Mendelian model knowledge (relatively speaking; they are just three observables. As such are particularly useful in disambiguating the evidence about *WKInqry* in the tasks that provide evidence about inquiry skills in the context of the Mendelian model, such as Tasks 1, 2, and 4.

Task 4 concerns the next step in the investigation. It targets *WKInqry*, but requires at least a `medium` level on *DKMendel*. As the structure of the conditional probability matrix shows (Table 15.5), this inhibitor relationship means that even students who understand inquiry fairly well are unlikely to succeed if they don't know enough about the Mendelian model to apply their knowledge to the situation at hand. Compared to all the other observables in the analysis, the data provided most information about the slope and intercept parameters in this evidence model: 252 % for the slope, 565 % for the intercept. Its intercept indicates it was relatively easy for this sample. The slope is not high, having posterior mean of 0.64. As a short series of choices to answer questions, there is less information than an open-ended explanation might have provided, or an analysis of subsequent steps a student would actually take in an investigation.

Looking at posterior EAP scores for individual students (Table 15.10), we see that the information in 14 responses each from 28 students was sufficient to impact distributions for individual student posteriors for *DKMendel* substantially, but it had a more modest effect on *WKInqry*. We noted earlier the impact on the distribution of the proficiency variables (i.e., the λ's, Table 15.8): there is a strong effect on the population distribution for *DKMendel* but the posterior for *WKInqry* has almost the same variance as the prior.

Table 15.8 also shows the increase in precision for the posterior distributions for the proficiency variable distributions for each student in the field trial. The means are expected values over proficiency states coded as integers, so high precision corresponds to probability concentrated on one particular value. Thus, posterior precision is very high for students who performed at high levels on all tasks or at low levels on all tasks. Almost all of their posterior probability is on the highest or the lowest value of an proficiency variable.

In general, we learn more about *DKMendel* than about *WKInqry*, mainly because there are more observables that provide information about *DKMendel*. The increases in precision are quite substantial for *DKMendel*. However, for only those students who did quite well overall is the posterior for *WKInqry* more precise. It also has higher posterior means for those students. For students who did poorly overall, the fact that all of the observables provided information about *DKMendel* meant that we could be more confident that their proficiency was low, but the conjunctive relationships with *WKInqry* meant we obtained little information about their inquiry proficiency. This assessment can't really tell us if they would have been able to perform well in inquiry tasks that involved some other topic with which they were more proficient. Their *WKInqry* posteriors are not much different than the priors. Note that this interpretation depends not on just the data, but the theory and expert opinion about the nature and relationships of domain knowledge and inquiry capabilities in science.

The full Biomass assessment, spanning 28 segments each with multiple tasks, would provide greater increases in precision. However, even with this short test we are pretty sure that a student who does poorly on most of the observables is `low` and a student who does well is `high` with regard to Mendelian Knowledge. We are far less sure about inquiry skills; since our information about them is confounded with Mendelian knowledge and there are far fewer observables, we have higher confidence only for those students who performed well on those tasks that required both proficiencies. For these students, we can infer that their understanding of inquiry was high *as reflected in the context of of the Mendelian model*. Whether they would show good inquiry skill in the context of other models and other domains would require evidence obtained in such situations.

We can quantify how much we learned about students in a Bayesian approximation of reliability. Consider a latent proficiency variable θ on a measured scale, and data $\mathbf{x}$. An analogue of reliability is the proportion of the variance of θ accounted for by $\mathbf{x}$. The numerator is the variance of students' EAP scores, and the denominator is the sum of this and the average of their posterior variances:

$$\rho \approx \frac{Var[E(\theta|x)]}{Var[E(\theta|x)] + E[Var(\theta|x)]}. \tag{15.9}$$

In the pilot data, the reliabilities were .62 for *DKMendel* and .14 for *WKInqry*. Not bad at all for *DKMendel*, given there were only 14 observations; useful enough to distinguish students who are clearly having difficulty and those who are clearly doing well, but not accurate enough for high-stakes inferences. Not good at all for *WKInqry*. Clearly more evidence would be needed even for low-stakes purposes.

15.3.2 Some Observations on Model Fit

Model criticism is an essential facet of Bayesian (or any other) statistical inference. When the data do not accord well with the model, then the inferences that probability-based reasoning allows us to draw through models are suspect. This brief section illustrates some techniques used to examining fit in the kinds of models we have discussed in this volume.

The particular technique we are exploring is the use of preposterior predictive data sets, created in the course of MCMC iterations (Sect. 10.2). For each observed response x_{imj} in the realized data, we can define another variable y_{imj} that follows exactly the same distribution we have proposed and fit for x_{imj} but is never observed. If our model is correct, the actual data are a plausible draw from the predicted distribution of the shadow data. Thus, the distribution of the shadow data or any summary statistic of it that is accumulated over the MCMC iterations constitutes a tailor-made null distribution against which to evaluate how surprising the data are in light of the model we have proposed.

Table 15.14 presents one draw of simulated preposterior predictive responses. The last rows and columns give provide the average observable sum scores and item scores for both the field trial data (x mean) and the shadow data (y mean). Note that the averages for both observables and students approximate those of the observed data closely, including the observables on which no actual students did better than `low`. There are a few more ‘2’ responses than in the actual data, but still most observed values are either ‘1’ or ‘3.’

The marginal means we have shown are just one possibility for a test statistic. Any statistic of actual responses, such as correlations and joint distributions, could be calculated on the shadow data set as well. Because the distribution of the test statistic can be accumulated over iterations, the posterior predictive distribution is an empirical null distribution for the test statistics, which can be used to evaluate how typical or how surprising the corresponding feature of the real data was.

We extended the BUGS code to add a preposterior predictive data distribution, and then used an index of examinee fit as the test statistic. Define the

Table 15.14 A set of simulated preposterior predictive responses

Student	y_{11}	y_{12}	y_{13}	y_{14}	y_{15}	y_{16}	y_{17}	y_{21}	y_{22}	y_{23}	y_{31}	y_{32}	y_{33}	y_4	y mean	x mean
1	3	1	1	1	1	1	1	1	1	3	3	3	1	3	1.71	1.50
2	1	1	1	1	1	1	1	1	1	1	1	1	1	3	1.14	1.14
3	3	1	1	1	1	1	1	1	1	1	1	1	1	3	1.29	1.00
4	1	1	1	2	3	1	1	2	1	1	1	3	1	1	1.43	1.36
5	3	2	3	1	3	1	1	1	3	2	3	3	3	3	2.29	2.00
6	1	1	1	1	1	1	1	1	1	1	3	1	1	3	1.29	1.57
7	2	1	1	1	1	1	1	1	1	1	1	1	1	3	1.21	1.50
8	3	3	3	1	3	1	1	1	1	3	1	3	3	1	2.00	1.86
9	1	1	1	1	1	1	1	1	1	1	1	1	1	2	1.07	1.07
10	3	1	1	1	1	1	1	1	1	2	1	1	3	2	1.43	1.50
11	3	3	3	1	3	1	1	1	3	1	1	3	3	3	2.14	2.00
12	3	3	3	1	3	1	1	1	1	2	1	3	3	3	2.07	1.86
13	3	3	3	1	3	1	1	1	1	2	1	3	3	3	2.07	2.21
14	3	2	3	1	3	1	1	1	1	2	3	3	3	3	2.14	1.86
15	3	1	1	1	3	1	1	1	1	2	3	3	3	1	1.79	1.71
16	1	1	1	1	1	1	1	1	1	1	1	1	1	2	1.07	1.00
17	3	1	1	1	1	1	1	1	1	2	3	3	3	1	1.64	2.14
18	3	3	1	1	3	1	1	2	1	2	3	3	3	2	2.07	2.00
19	1	1	1	1	3	1	1	1	1	1	3	3	1	1	1.43	1.50
20	1	1	1	1	1	1	1	1	1	1	1	1	1	3	1.14	1.21
21	3	3	3	1	3	1	1	1	1	1	3	3	3	1	2.00	1.57
22	3	3	3	1	3	1	1	3	1	3	3	3	3	2	2.36	2.43
23	3	1	1	1	2	1	1	3	1	1	3	1	3	1	1.64	1.93
24	3	1	1	1	1	1	1	1	1	3	1	1	1	3	1.43	1.64
25	3	2	3	1	3	1	1	1	1	3	1	3	3	3	2.07	2.14
26	1	1	1	1	1	1	1	1	1	1	1	1	1	3	1.14	1.00
27	3	3	1	1	2	1	1	1	1	1	1	3	1	2	1.57	1.64
28	2	1	1	1	1	1	1	1	1	1	1	3	1	2	1.29	1.64
y Mean	2.36	1.68	1.64	1.04	2.00	1.00	1.00	1.21	1.14	1.64	1.79	2.21	2.00	2.25	1.64	
x Mean	2.50	1.71	1.71	1.00	1.71	1.00	1.00	1.29	1.00	1.50	1.93	2.36	2.00	2.29		1.64

The x mean refers to the original data, the y mean refers to the shadow data

fit mean square for Examinee i as follows:

$$Z_i = \frac{1}{14} \sum_m \sum_j \left(x_{imj} - E\left[x_{imj}\right] \right)^2 \ , \tag{15.10}$$

where responses are coded `high` $= 3$, `medium` $= 2$, and `low` $= 1$, and

$$E\left[x_{imj}\right] = \sum_{k=1}^{3} k\mathrm{P}(x_{imj} = k | \mathit{DKMendel}_i, \mathit{WKInqry}_i, \boldsymbol{\alpha}_m, \beta_m) \ . \tag{15.11}$$

In each iteration of the MCMC sampler, there are sampled values for both the proficiency variables and the evidence model parameters, so the Eq. 15.11 can be easily calculated. Then Eq. 15.10 is calculated with both the observed and shadow data, producing two realizations of the examinee fit index. The relevant index is the proportion of iterations in which the fit mean square for the y's is greater than the one for the x's. This is known as the posterior predictive p-value. Note that different values are used for the evidence model parameters (and the proficiency model parameters as well) for each iteration, so over the course of MCMC cycles, the estimates take into account uncertainty about the values of the model parameters, as well as the examinee proficiency.

One run with 30,000 iterations produced p-values across the 28 examinees between .88 for Examinee 9 (the best fit) and .09 for Examinee 22 (the worst fit). Examinee 22's pattern is somewhat uncommon because of `high` values for all observables except for the four observables (Task 1: Observables 4, 6 and 7, and Task 2, Observable 2) on which nobody scored well. We would have expected some correct responses from someone who performed so well on the rest of the assessment. This may be a problem as the posterior conditional probability tables for those observables (e.g., Table 15.12) still have a probability of over .5 for a `low` outcome when both the *DKMendel* and *Context* variable are in their highest states.

The fact that the lowest empirical p-value was only .09 caused us some concern about the power of the test. We did a second run with an additional fictitious response vector, one with high values for the harder observables and low values for the easier ones:

$$x_{\text{badfit}} = (1,1,1,3,1,3,3,3,3,3,3,1,1,1) .$$

We were comforted to see that of 20,000 draws of a shadow response pattern to this maximally bad fitting pattern, only 4 had a higher mean square–an empirical p-value of .0002. When a response vector is seriously out of sorts, this index will flag it.

However, this examinee fit test does suffer from low power. First, there is the generally conservative nature of the posterior predictive data procedure: it generally takes a large deviation from the expected to overcome the uncertainty about the parameters (Sinharay 2005). The second is the short length of the test (just the first segment of Biomass). Getting both information about an examinee's ability on two proficiency variables and information about model fit is asking a lot of 14 observables. A longer test would have more power to detect potentially interesting response patterns.

15.3.3 A Quick Validity Check

Messick (1989) describes validity as "an integrated evaluative judgment of the degree to which empirical evidence and theoretical rationales support the *adequacy* and *appropriateness* of *inferences* and *actions* based on test scores or other modes of assessment" (p. 13; emphasis original). We will touch here on aspects of both empirical evidence and theoretical rationales.

Embretson (1983) distinguished lines of validity argumentation that concerned assessment design, that is, why tasks created and scored in such-and-such way ought to provide evidence about the construct of interest, and the subsequent relationships of scores from an assessment with past, concurrent, or future test scores and events. She called these "construct representation" and "nomothetic span" sources of validity evidence.

For Biomass, the ECD process described in Chap. 14 inherently provides construct representation evidence. The entire process of Domain Analysis in Biomass was meant expressly to gather information from research and experience in the domain. Domain Modeling was meant expressly to lay out the theoretical rationale for why what features of performance in what kinds of situations ought to provide evidence about the targeted proficiencies, for the targeted population, for the the targeted purposes. The translation of this argument into ECD student, evidence, and task models links the argument to the structure of the pieces of machinery that instantiate the argument. The model building, model fitting, and model criticism support the link of reasoning from performances to scores. The ECD framework makes explicit and shareable (and re-usable) coordinated lines of validity argumentation that are often reside only in test developers' heads and work processes.

Looking at the relationships between Biomass EAP scores and previous Biology experience can provide one line of nomothetic-span validity evidence. Because the field test data contains people who are at a variety of levels with respect to biology and genetics, it should provide an opportunity to quickly check an aspect of validity. If Biomass is a valid assessment of the domain knowledge about genetics as well as the more general working knowledge about scientific inquiry, we would expect that examinees who have had previous training in Biology will do better than the others. (Stronger evidence that bears directly on the formative use of Biomass would be to see if feedback on the tasks would in fact lead to improved understanding in the domain.)

And unfortunately, we don't have either of these kinds of actual data for the pilot sample. We can however illustrate the thinking with a hypothetical extension of the example. Let us suppose that in addition to the scores from Table 15.10, Students 17–28 had previous experience, and Students 1–16 did not. One simple way to check the relationship is to perform a Student's t test on the EAP scores for the two groups.

Looking first at the expected proficiency level for *DKMendel*, we that the group with previous training has an average EAP score of 2.04, compared to an an average EAP score of 1.78 for the group without previous training ($t(26) = 1.24$, p-value = .23). For *WKInqry* the story is similar, the group with previous training again has an average EAP score of 2.04, compared to an an average EAP score of 1.84 for the group without previous training ($t(26) = 1.92$, p-value = .07). Thus the validity test points in the correct direction, although the validity evidence from this study would be considered marginally significant just for *WKInqry*. There are several factors to consider here. First, the size of the sample is quite small (only 28 students altogether); a larger sample size would make the test much more sensitive to small differences between the groups. Second, the length of the test is short (only 4 tasks, containing 14 observables). A longer test would provide more opportunities for examinees at high and low proficiency levels to distinguish themselves. Finally, the non-representativeness of the sample further weakens inferences.

Even though the empirical validity evidence from this study is weak, we still have one line of strong evidence for construct validity developed through the ECD design methodology. If Biomass were to be used in an application for which its consequential validity was important—impact on learning as a formative assessment, for example—more careful study would be needed. In this respect, a test built with Bayesian networks is no different than one built with a more familiar methodology (e.g., classical test theory or IRT).

15.4 Conclusion

A big advantage of the Bayesian approach to psychometrics is how smoothly it scales from the case of no data to large amounts of data. Section 15.1 builds a perfectly functional scoring model for Biomass. It leans heavily on the ECD design to accomplish this. In fact, the ECD design structures most of the work. In many cases the only "psychometric" work the designers had to do was to pick a particular parametric form, and prior laws that match the experts' description of the observable difficulty (i.e., "typical," "harder," or "easier"), and how the proficiencies interact to when solving the problem (e.g., "conjunctive," "compensatory").

Section 15.3 goes on to describe how, with an appropriate sample, we could update the parameters of the scoring model. Given the limitations with the field data described in Sect. 15.2 and the fact that Biomass was never in use in actual classroom practice, the fielded version of Biomass only ever used the prior parameters.

We would not claim generalizability for the *results* of the pilot study. We would anticipate the item and population parameters to continue to change if more data were obtained. Knowing that the estimates of some of the parameters depending materially on priors, we would want to investigate the sensitivity of inferences to priors; ideally we would obtain enough data that inferences would be robust with respect to a range of reasonable priors. More importantly, though, with more data we would be able to fit and compare alternative scoring schemes for performances and explore alternative forms for evidence models using methods described in the preceding chapters. We would hope to feed back what we learned to improved task construction as well. The value of the example is its detailed walk through of *processes*, and of *ways of thinking* about assessment, design, and Bayes net modeling.

Although it can be difficult to obtain a high quality sample for an assessment like Biomass in classrooms, an alternative now exists. If Biomass were made available over the Internet, chances are a number of classes would use it, and a student sample could be gathered quite naturally over time. However, few teachers would be willing to adopt Biomass unless it is able to provide scores. The Bayesian approach to psychometrics offers a solution to this chicken-and-egg dilemma. Release Biomass 1.0 using a scoring model based entirely on the prior. This is certainly no worse than a number right score, and, because of the grounding in the ECD design, quite possibly better. When sufficient data come in, the parameters can be updated and Biomass 1.1 can be released. This process can continue indefinitely into the future, and creates a viable model for releasing low-to-moderate-stakes assessments quickly, and improving them over time once they are fielded.

This chapter also illustrates the other important use of data in Bayesian psychometrics, the ability to provide critiquing information about the model. Bayesian psychometrics supports sophisticated model checking procedures, like posterior predictive data sets. We have see, however, that there is still a great deal of power in the simple statistics used for item analysis. Even with a small pilot sample, these provide vital checks on both the design and implementation of a new assessment. This practice of piloting and improvement in both task design and evidence rules is a large part of the effort in practical assessment design.

Exercises

15.1 (Dependence of *DKMendel* and *WKInqry*). Section 15.1.1 claimed that in a submodel of the Biomass proficiency model built in the previous chapter (Fig. 14.3), *DKMendel* and *WKInqry* are dependent. Verify that claim.

15.2 (Alternative Parameterizations). Suppose that one of the states of *DKMendel* was anticipated to be rare in the sample. Would reparameterizing the CPT for *WKInqry* help? Which parameterization would you choose and why?

15.3 (Edge Orientation). How would the analysis in this chapter differ if the edge in the proficiency model was oriented from *WKInqry* to *DKMendel* instead of from *DKMendel* to *WKInqry*? How would the ultimate estimates for the evidence model parameters differed? The proficiency model parameters?

15.4 (Data Collection Effort). Obviously, the data collection effort described in Sect. 15.2 is biased. Describe some of the sources of bias and how the actual data from the correct population might differ from the real data.

15.5 (Multiple Populations). The field data described in Sect. 15.2 included both high school and college students. Assume for the moment that the subject's education level does not influence the evidence model parameters, just the proficiency model. How can the model be changed to account for difference between these two populations? Hint: add a demographic variable to the proficiency model.

15.6 (Prior Strength). Consider the prior for *DKMendel* given in Equation 15.1. How could it be made stronger? Weaker?

15.7 (Prior Association). Consider the prior for *WKInqry* given *DKMendel* (Eq. 15.2). How could the association be made stronger? Weaker? Does this affect the "strength" of the prior?

15.8 (Truncated Normal). In constructing the prior distributions for the evidence models, why is the truncated normal distribution $\mathrm{N}^{+}(1,1)$ used for α_{1jDKM} instead of the untruncated normal distribution $\mathrm{N}(1,1)$? If a lognormal distribution were used instead of a normal distribution, would truncation still be necessary?

15.9 (Context Variable). What would happen if the context variables were given a marginal probability $\mathit{Context} \sim \mathrm{Bernoulli}(.95)$ instead of $\mathit{Context} \sim \mathrm{Bernoulli}(.5)$? How would this affect the estimated parameters? How would this affect the interpretation of *Context*?

15.10 (No Context Variable). Why is no context variable needed for Evidence Model 4?

15.11 (Truncated Normal Mean). The parameter $\alpha_{1,1,DKM}$ has the distribution $\mathrm{N}^{+}(1,1)$ but when BUGS is run with no data, it has a prior mean of 1.29 (Table 15.6). Why is this mean not 1.00? Why is the observed standard deviation less than 1?

15.12 (Partial Credit Evidence Rules). Consider any of the *MendModGen* observables from Task 1. These observables all have three levels and are all based on the *DKMendel* proficiency variable, which also has three levels. If the observation is to provide the maximum possible information about *DKMendel* what is the relationship between the states of the proficiency variable and observable that must be expressed in the evidence rules?

15.13 (Negative Precision Increase). Some of the entries in Table 15.8 show a negative increase in precision from prior to posterior. What does this mean?

15.14 (Percent Increase in *WKInqry*). Using the data from Table 15.10, plot the posterior mean of *DKMendel* against the percent increase in posterior precision for *WKInqry*. Explain the relationship.

15.15 (Project). This chapter contain all of the information necessary to recreate the analysis in Sect. 15.3. Implement this analysis in BUGS and recreate the analysis. Try several variant models and create several replicate data sets. How well does the model fit? Are there better fitting alternatives? How do alternative prior distributions for parameters (e.g., as in Exercise 15.11) affect parameter estimates?

15.16 (Project2). Four observables, $x_{1,4}$, $x_{1,6}$, $x_{1,7}$, and $x_{2,2}$ have no persons in the sample with a response other than `low`. Drop those items from the sample and rerun the calibration. Does this improve the fit statistic for Person 22?

16

The Future of Bayesian Networks in Educational Assessment

From one perspective, evidence-centered assessment design and Bayesian networks are just notations. It is easy to express familiar assessment design patterns using these notations. Bayesian networks are just a way to parameterize multidimensional latent class models. What have we gained?

What we have gained is a notation that scales up from familiar assessment designs and purposes, to the wide array of new kinds of assessments made possible by advances in technology, cognitive psychology, and learning sciences. This notation can help test designers know when they can reuse familiar models and procedures, and do so efficiently, and when they need to adapt or extend the models for a new purpose. Biomass (the previous two chapters) is an example of the new kinds of things that can be done with Bayesian networks.

This final chapter cannot hope to tie up all of the loose ends left in this book or explore more advanced applications. We rather hope that our readers will take up the challenge. This final chapter provides some pointers into the literature to start you on that path. Section 16.1 provides an incomplete review of various applications of Bayesian networks. Section 16.2 looks at recent developments in Bayesian networks that should find uses in educational applications. Section 16.3 looks at the important issue of integrating assessment with instruction, and Sect. 16.4 looks at issues of validity. Finally, Sect. 16.5 describes some problems that are still open.

16.1 Applications of Bayesian Networks

By far the most common application of Bayesian networks in education is in intelligent tutoring systems (ITSs; some representative examples include Aleven and Koedinger 2002; Baker et al. 2010; Bunt and Conati 2002, 2003; Corbett and Anderson 1994; Crowley and Medvedeva 2006; El Saadawi et al. 2008; Gamboa and Fred 2001; Gertner et al. 1998; Graesser et al. 2001; Henze and Nejdl 1999; Koedinger and Aleven 2007; Ley et al. 2010; Madigan et al.

R. G. Almond et al., *Bayesian Networks in Educational Assessment*,
Statistics for Social and Behavioral Sciences, DOI 10.1007/978-1-4939-2125-6_16

1995b; Martin and VanLehn 1995; Millán and Pérez-de-la-Cruz 2002; Ritter et al. 2007; Sao Pedro et al. 2013; VanLehn and Martin 1997; VanLehn 2008; Vomlel 2003; Vomlel 2004; Zapata-Rivera et al. 1999; Zapata-Rivera and Greer 2004a; Zapata-Rivera and Greer 2004b). Bayesian networks have been popular for intelligent tutoring applications because they offer a principled mechanism for reasoning about uncertainty and because of the ready availability of software to perform that reasoning. Often, the Bayesian network is a part of the student model of the tutoring system, although the term as used in the intelligent tutoring literature differs slightly from the term *proficiency model* used in this book. In many applications noted above, the "student model" spans the pieces this book would assign to the proficiency model and the evidence model.

The student model in an ITS is used to make a number of different kinds of decisions. One important decision is the problem of task selection, which is discussed in detail in Chap. 7. However, other decisions may happen within the context of a task, such as how much help or scaffolding to give while a student is attempting a problem. Often, the student model contains information about aspects of the student not related to knowledge, skills, and abilities, for example, learning preferences or effect (Conati and Maclaren 2009).

A standard distinction between ITSs and educational assessments is that the former are deployed as learning environments, while the latter are distinguished in learn–assess cycles. These cycles might be infrequent and hold high stakes, such as end of course tests, or shorter and used for guidance, such as formative tests. The shorter the cycles, though, the more ITSs and assessments resemble one another. In general, the closer the ongoing instruction and the lower the stakes, the lower is the need for high reliability; each decision is less critical and there is more opportunity to correct errors.

A high-stakes assessment needs high reliability for the reporting variables. This means having a large enough number of tasks to provide evidence about each reporting variable. As usual, there is a limited amount of time to administer assessment tasks, the number of proficiency variables is limited by practical considerations. In this situation, Bayesian networks have little computational advantage over other similar methods, like multidimensional item response theory (MIRT), unless the combinations of proficiencies or interdependencies among observables are outside those that MIRT models can handle.

An ITS, in contrast, can be deployed over a long period of time. Thus, its proficiency model can grow to tens or hundreds of variables. Here, the ability of the Bayesian networks to break large problems into small pieces has distinct computational advantages. This is true whether it is a discrete Bayesian network (as is developed in major part of this book) or a model with continuous variables. Rijmen (2008) uses Bayesian network ideas to develop efficient algorithms for MIRT models taking advantage of conditional independence assumptions inherent in special proficiency models (e.g., the bifactor model, in which each task has exactly two parents: a general proficiency and one specialized proficiency, as in a mathematics test with a general math proficiency and unique factors for algebra, geometry, and statistics, with each item associated with just one of the latter).

As mentioned, Bayes nets provide advantage over the more familiar alternatives in handling complex simulation tasks. Such tasks have multiple observable outcomes and potentially complex patterns of dependence among them. Here, Bayes nets provide a flexible language for describing that local dependence. Although Wainer et al. (2007) develop an elaborated testlet IRT model, adding a latent variable to explain dependence among the related observed outcome variables, this is only one possible pattern for dependence (Almond et al. 2006b). It is advantageous for an assessment program to provide test developers with a library of design patterns with preconstructed Bayes net fragments to handle complex but recurring evidentiary situations: reusable, ready-made argument structures around which they can write unique tasks. Chapter 15 develops several different design patterns for use in the Biomass assessment (also see Mislevy et al. 2002d).

Another advantage of using Bayes nets in complicated but lower-stakes applications is that calibrating the model (e.g., all of the machinery developed in Part II) is not strictly necessary. An ITS using a Bayes net scoring engine could be fielded using just the expert's best guess as to the parameter values. This would be no worse than a test scored using a weighted number right scheme. The fielded system would produce scores, which would encourage educators to use it. These early users could provide the data needed to perform later calibration and model checking analyses.

Consider for instance adaptive content with evidence-based diagnosis (ACED) (Example 13.1; Shute et al. 2008), an assessment *for* learning system designed to assess/teach about algebraic sequences that had a Bayesian network scoring engine. The initial version of ACED had 63 tasks (all with a single binary observable), which the experts had classified as `easy`, `medium`, or hard. The designers also provided a Q-matrix linking each task to one or more proficiency variables. For the initial field trial, parameter values were set based on the prior distribution, that is, all `easy` tasks were assigned a difficulty value of -1, `medium` tasks were assigned a difficulty of 0, and `hard` tasks were assigned a difficulty value of 1. Similarly, discrimination parameters were assigned based on the perceived importance of the proficiency variables for each task type. Using these expert guesses for parameter values, the reliability of the overall early assessment program (EAP) score from the Bayesian network was 0.88, which was also the reliability of the number right score using the tasks. The lesson here is that it is not the scoring model that makes the reliability, but the quality and consistency of the tasks.

The ACED tasks were all provided with informative feedback, which showed students a worked solution if they got a problem wrong, and an adaptive task selection algorithm based on the expected weight of evidence (Sect. 7.3). In a randomized trial of ACED (Shute et al. 2008), students who were given both informative feedback and adaptive task selection showed significant improvement in their ability to solve sequence problems from pretest to posttest. Even though the students were obviously learning from the informative feedback, the EAP score from the Bayes net was still a good prediction of the posttest score. The correlation was about 0.68, which is near the upper bound determined by the reliability of the 20-item posttest. Thus, providing feedback and learning during an assessment do not seem to hurt measurement properties. In ACED, the feedback improved the measurement quality,[1] with the correlation between EAP score and posttest slightly higher in the groups which received informative feedback.

16.2 Extensions to the Basic Bayesian Network Model

Another advantage of Bayes nets is that there are a large number of applications in addition to those in educational assessment. That means other communities of practice are developing new techniques, which may be useful for educational testing. In particular, there are many conferences in the artificial intelligence community where the latest developments are often first published. Many of those conference proceedings are indexed in the CiteSeer database (`http://citeseerx.ist.psu.edu/`). Two developments that hold implications for educational testing are object-oriented Bayes nets and dynamic Bayes nets. They are summarized in Sects. 16.2.1 and 16.2.2. Section 16.2.3 notes a third development, namely tools to help assessment designers particularly take advantage of these and other advances in the more general Bayes nets world.

16.2.1 Object-Oriented Bayes Nets

Breese et al. (1994) introduced the concept of knowledge-based model construction. Bayesian networks and influence diagrams had proved to be good at managing uncertainty in decision-making in ways that did not get tripped up by dependence issues, but a Bayes net that covers all possible contingencies that might be needed for a range of applications would be entirely too large. To make the influence diagram more manageable, the knowledge-based model construction splits the network into pieces and then uses logical rules to assemble a model from just the pieces needed to solve a specific problem.

Mahoney and Laskey (1996) (see also Laskey and Mahoney 2000) present a nice example of knowledge-based model construction. Consider a traffic investigator who needs to recreate the scene of an accident to determine who is at fault. Certain conditions or contexts require different parts of the model.

[1] The sample size was not large enough for the differences in correlations to be significant; however, that does not really matter here. The point is that the fact that students are learning during the assessment did not decrease its predictive validity.

For example, if it was raining at the time of the accident, then there is a set of nodes relating to wet road conditions, which is not necessary when the road conditions are dry. Similarly, there are whole sets of nodes that are only necessary if it was nighttime, the accident occurred at an intersection, there was limited visibility in one direction, there were witnesses, and so forth.

Knowledge-based model construction inspired the idea of splitting the total graphical model for an assessment into proficiency and evidence model fragments (Sect. 5.4.1; Almond and Mislevy 1999; Almond et al. 1999). Moreover, using evidence models takes advantage of the special local independence assumptions constructed into assessments. Each task provides a context for the evidence model, and the local independence assumption means that the nodes that are common across tasks only appear in the proficiency model. The "knowledge" for knowledge-based assessment-model construction is the selection of a task to present.

There are more advanced situations in educational assessment that can benefit from a more complete version of knowledge-based model construction, such as simulation-based assessment and direct observation. Consider an observer watching a teacher in front of a classroom either live or on video. There can be a whole collection of observable outcomes related to how the teacher handles disruptive students that may not be relevant if no student behaves disruptively during the observation period. A simulation-based assessment can have certain evidence model fragments that only become active if the student takes a certain path through the simulation. Agents recognize an instance of a particular task configuration and alert the presentation process to capture the work product that will yield the values of the observables in the evidence model (Sect. 13.2.1).

An important development in the field of knowledge-based model construction is the idea of object-oriented Bayesian networks (Koller and Pfeffer 1997; Laskey and Mahoney 2000). Think again of the accident investigation example, and consider Bayes net fragments associated with multiple vehicles at the scene of the accident. Vehicles form a general *class* and we create an *instance* of that class for each vehicle at the scene. Each object would have a Bayesian network fragment that has nodes relating to its properties (e.g., location, velocity, acceleration, and accident status) that are copies of the nodes in the Bayes net for the class with some labeling convention used to make them unique.

However, the object-oriented nature allows for more sophisticated reuse. A key idea is *inheritance*, in which the default values for properties of an object are taken from its *parent*. For example, the *PassengerCar* and *Truck* objects may both inherit from the *Vehicle* class. The more general class would contain common nodes for common properties (e.g., location), and the more specific classes would have additional nodes appropriate for the subclass (e.g., number of passengers or cargo weight). The advantage is that the subclasses can inherit default properties from the more general parent classes, only overriding those properties that are different. Note that often the inheritance hierarchy is a matter of convenience and not taxonomy. For example, the *Bus* class might inherit from the *Truck* class not because a bus is a kind of truck, but because there are many similar properties (e.g., both are heavy and require special licenses to drive).

In assessment, task models and task shells form a natural inheritance hierarchy. Not only can inheritance properties be exploited to make task construction more efficient (Vendlinski et al. 2008), analogous structures can be exploited for corresponding evidentiary relationships to make psychometric model building more efficient. The idea can also be exploited to learn link model parameters for tasks in the same family. Geerlings et al. (2011) and Johnson and Sinharay (2003) show how such task hierarchies can be used to more efficiently calibrate assessments, effectively reducing the size of the pretest sample needed for each new assessment.

16.2.2 Dynamic Bayesian Networks

Throughout this book, we have mostly relied on the simplifying assumption that a student's proficiency does not change during the course of measurement. However, we know that students learn, and that is why we teach them. This is especially true in Assessment *for* Learning systems like intelligent tutors and ACED (Shute et al. 2008, see also Examples 7.5 and 13.1 and Sect. 14.4.5), which are designed to change student knowledge states. Consider the following example.

Example 16.1 (The Aha Moment). *Suppose that Aisha is struggling with the geometric table concept. She gets a medium and two easy table items wrong. Suddenly, something in the feedback kicks in and she has an "Aha" moment. She subsequently gets two easy items right and a medium item right.*

ACED treats this as a set of 6 items: 3 right and 3 wrong. It ignores the temporal sequence and assumes that Aisha's proficiency is the same throughout the testing period. It would probably give her a pretty flat distribution for the table proficiency variable. The EAP score would dip down below zero and then come back up to zero, but it would do so too slowly. If we took the temporal ordering of the data into account, we can see that the past observations should be discounted in some way, and that her EAP score should now be positive.

One simple technique for dealing with changing proficiency states is called fading. The idea is that raising the likelihood to a power less than one will make it flatter. If we raise the likelihood to, say, the power $1/t$, where t being the number of time periods that has elapsed since the observation was made, then more recent observations will be weighted more heavily than past observations. Eventually, the influence of past observations will decrease to zero. The ability to do fading is built into several Bayesian network software packages. Compared to modeling change, fading has the advantages of being simpler and robust to patterns of change. On the other hand, it does not use the data efficiently, and knowledge about patterns of change is not utilized (Mislevy 1995a).

More complex models for change can be described through *dynamic Bayesian networks* (DBNs) (Dean and Kanazawa 1989). The fundamental idea is that a model that changes over time can be described by two pieces; a single time-slice model that describes the initial state of the system, and a two-time-slice model that describes how the system changes from one-time slice to the next. The two-time-slice models can be chained for as many time points as needed. Figure 16.1 shows the basic framework, as it would appear in an assessment setting; each vertical panel represents a time slice at which students are assessed. The upper variables are the proficiency variables ($\mathbf{S}_t$) and the lower ones are the observable outcome variables ($\mathbf{O}_t$). The vertical links between the proficiency variables are the familiar evidence models and much of this book describes possibilities for building these models. The horizontal links between the proficiency variables at two-time slices ($\mathbf{S}_t$ and $\mathbf{S}_{t+1}$) represent the two-time-slice model. Underlying this model is a Markov property that what happens in any time slice is independent of the past history given the previous time slice. The simple form of this model has made it attractive and it has been studied by a number of authors (e.g., Boyen and Koller 1998; Murphy and Russell 2001; Koller and Learner 2001; Takikawa et al. 2002).

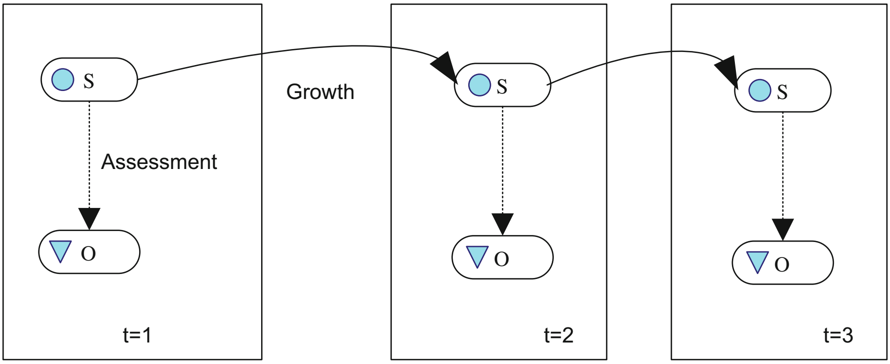


Fig. 16.1 A basic dynamic Bayesian network
Reprinted with permission from ETS.

What's new here is modeling the transition from one-time point to the next. We will mention two ways of approaching this, which can be used separately or together depending on how the learning system is constructed. They are mathematical learning models and Markov decision process (MDP) models.

Mathematical learning models build from the work of mathematical psychologists going back to L. L. Thurstone and Clark L. Hull in the first half of the twentieth century, and formalization by Estes, Bush, Mosteller, and others in the 1950s and 1960s (for an overview see Restle and Greeno 1970). The basic idea used in practice systems and many tutoring systems is that a student has or has not mastered some skill at Time t, makes a response, and the probability of mastery at time $t+1$ depends on the previous mastery state and the value of the response (Corbett and Anderson, 1994, Cen et al. 2006). Many extensions are possible, but Fig. 16.2 shows the basic form.

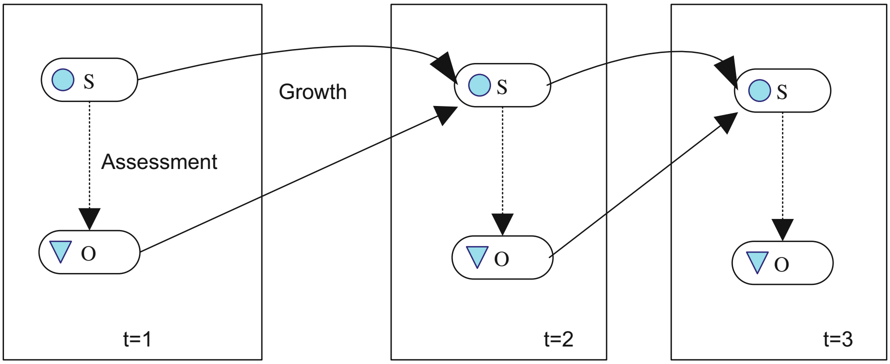


Fig. 16.2 A learning-model dynamic Bayesian network
Reprinted with permission from ETS.

Suppose, for example, a student who is in a mastery state at time t (that is, $S_t = 1$) remains a master. A nonmaster may become a master at time $t+1$ after an incorrect response with some probability δ^-, or after a correct response with some presumably higher probability δ^+:

$$P\left(S_{t+1} = 1 \,|S_t, O_t\,\right) = \begin{cases} 1 & \text{if} \quad\;\; S_t = 1 \\ \delta^- & \text{if } S_t = 0, O_t = 0 \\ \delta^+ & \text{if } S_t = 0, O_t = 1 \end{cases} .$$

Levy (2014) shows how to estimate the learning parameters along with the conditional response probabilities in the Markov chain Monte Carlo (MCMC) framework of Chap. 9.

The *MDP* (Boutilier et al. 1999) is an elaboration on DBN that includes nodes for decisions and utilities (similar to the way an influence diagram is a generalization of a Bayesian network). In particular, between each time slice, an *action* is chosen, which influences how the variables are likely to change between the time slices. Often, many of the key variables in the model cannot be observed resulting in a partially observed MDP (POMDP). The reason for building POMDPs is *planning*: to choose a sequence of actions or a *policy* for choosing actions that will optimize the probability of reaching a given goal. Hoey et al. (2001) describe the software package SPUDD (stochastic planning using decision diagrams) that helps to select optimal policies for MDPs. Applications of MDP have been used, for example, to control elevators (Nikovski and Brand 2003) and model learning in ITSs (Reye 2004).

Almond (2007a), (2007b) maps the fundamental problem of integrating educational information to the MDP framework. Figure 16.3 shows the general framework. Again the horizontal links between the proficiency variables at two-time slices represent the two-time-slice model. In general, this transition can depend on what kind of instruction the student receives between measurement opportunities. The choice of instruction is the action in the POMDP model.

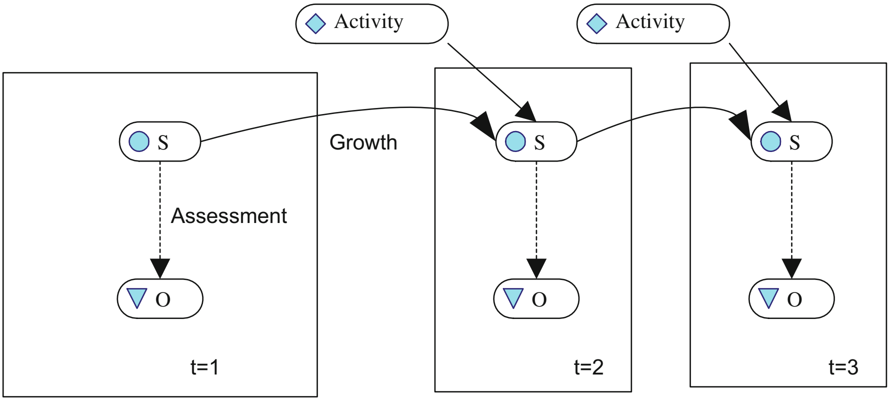


Fig. 16.3 Instruction as a partially observed Markov decision process Reprinted with permission from ETS.

DBNs and MDPs have both been used in ITSs ; Mayo and Mitrovic (2001) review several systems. For example, Matsuda and VanLehn (2000) describe the use of an MDP to make decisions about selecting hints or new problems. Murray et al. (2004) use a DBN to calculate expected utility for a limited set of actions and select the action that will maximize utility at the next time step. While this may lead to action choices that are suboptimal when considered as part of a sequence of actions, they are probably close enough to optimal for practical purposes. Sabourin et al. (2013) use a dynamic Bayes net to model students' improving self-regulation skills in an exploratory game environment. Reye (2004) describes the relationship between MDP approaches and other approaches used for updating student models in the intelligent tutoring literature. Conati and Maclaren (2009) use a DBN to track user affect in an educational computer game.

16.2.3 Assessment-Design Support

As we have noted previously, this book is mainly about using Bayes nets in educational assessment, but as a part of a coherent system of argumentation, design, and inference. The idea is not to take some assessment and ask "might we now use Bayes nets to make sense of the data that it will produce?" Rather, ask what evidence do we need to ground the inferences we need to make? What situations and actions will produce the evidence? How do we need to evaluate, synthesize, and characterize the evidence to support the targeted inferences? Bayes nets have features that lend themselves well to complexities of evidentiary reasoning in complex assessments, and we may find we can capitalize on them, from the beginning, in producing assessments that suit our purposes.

The Bayes net structures and calculations have great potential to support a richer space of assessment, but they cannot fulfill their promise unless they are used effectively in conjunction with chunks of evidentiary arguments—evidentiary arguments that are grounded solidly in an understanding of the nature of proficiency in a domain and how we know it when we see it. The kinds of Bayes net structures that we have discussed are conjunctive and compensatory combinations of proficiency, conditional dependence among observable variables, and Markov processes across time points. What task designers need is building blocks of Bayes nets fragments *connected with the kinds of evidence and arguments they need to make.*

In other words, we would like to provide tools and libraries of "argument chunks" that capture recurring relationships among proficiencies and observations in Bayes nets, so that assessment designers do not need to build Bayes nets from scratch every time. We want to provide them these evidentiary argument skeletons around which they can construct tasks that will evoke evidence about targeted aspects of proficiency. We want to provide design patterns, based on research and experience, which address aspects of proficiency that arise in many domains, such as model-based reasoning, inquiry, and systems' thinking—kinds of situations, work products, and observable features that can then be tailored to the domain and purpose, with links to Bayes nets fragments which too can then be tailored to the particulars of the application (Mislevy et al. 2002d).

Assessment-design support tools have been explored by a number of researchers (e.g., Conejo et al. 2004; Luecht 2012). Our own experience with such systems has been the ECD-based design systems PORTAL (Almond et al. 2002b) and PADI (Mislevy and Riconscente 2006). Such supports will be necessary to scale-up the use of Bayes nets psychometric models we have discussed, to support the flexible and recombinable assessments for which they are a natural way to make sense of students' performances.

16.3 Connections with Instruction

ITSs naturally tie assessment and instruction together, but that relationship is more universally important. Example 16.2 is an old joke that illustrates the point.

Example 16.2 (Lost Car Keys). *One night, a psychometrician was kneeling down underneath a street lamp peering intently at the ground. A friend walked up behind him.*

"What are you doing?" asks the friend.
"Looking for my car keys," replies the psychometrician.
"Where did you loose them?"
"Out there," says the psychometrician gesturing out towards the darkness.
"Then why are you searching here?" asks the friend.
"Because the light is better!"

The metaphor is obvious, but the point is not always taken. It is far too easy to find IQ tests that define IQ as whatever it is the test measures, or science tests that test science facts rather than scientific reasoning because testing fact is easy and testing reasoning is hard.

Pelligrino et al. (2001) put the need for alignment more strongly: "*Educational assessment does not exist in isolation, but must be aligned with curriculum and instruction if it is to support learning* [italic theirs]." Wilson (2004) makes it a central theme of his textbook on assessment design.

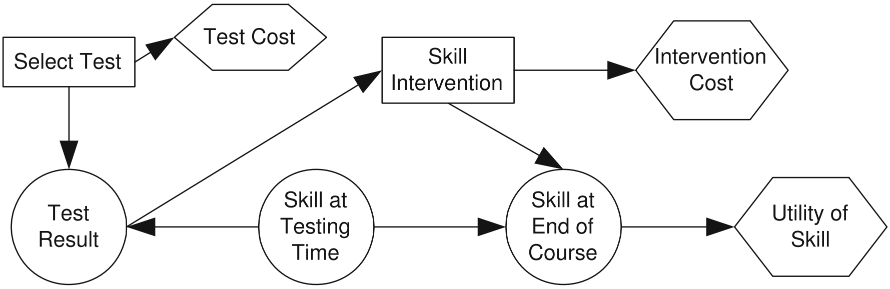


Fig. 16.4 Influence diagram for skill training decision
Reprinted from Almond (2007) with permission from ETS.

The need for alignment is especially critical if the rationale for the assessment is cognitive diagnosis. It is worth revisiting the influence diagram in Fig. 4.14 (repeated here as Fig. 16.4). The educator finds the assessment valuable precisely when she can make a better decision about instruction using the information than she can make without the information. An immediate corollary is that if the educators set of options is impoverished (say limited to continuing mainstream instruction or expensive one-on-one tutoring), the educator may not need a sophisticated assessment; simple classroom observation may be sufficient.

Even if the educator has a rich set of options, there needs to be an alignment between the options and the assessment. Making distinctions among students that do not correspond to instructional options has limited value. For example, the mixed-number subtraction example (Sect. 11.1), attempting to diagnose the presence or absence of other skills for students who lacked basic fraction subtraction ability, has limited pedagogical utility. A cognitively diagnostic assessment that makes distinctions that do not correspond to instructional options may however have utility as a research tool. In particular, identifying important and common cognitive states could improve curriculum design and create new instructional options. On the other hand, until those new options are in place, the distinctions will have limited utility to a classroom teacher or administrator.

The approach taken in this book has been to treat assessment design as an engineering discipline. While there are important scientific and mathematical principles that underlie good assessment design, it is still largely a creative endeavor with plenty of acceptable solutions. As with other branches of engineering, the results are usually the most satisfactory when the design process is customer focused, or at least involves numerous chances for feedback from potential customers.

One tool that we have found useful in soliciting input from potential score users is the *prospective score report*. This is simply a mock-up of what a score report might look like under the current score design. Bayesian networks are helpful here because it is easy to simulate plausible response patterns and proficiency profiles. Prospective score reports are very useful for verifying that the claims targeted by the current assessment design are in fact the ones that the prospective customers want. They are also useful for conveying the consequences of psychometric issues to score users with limited training in psychometrics. If an issue will cause them a problem, it most likely will show up on the score report in some way. Almond et al. (2009b) and Zapata-Rivera et al. (2012) illustrate these ideas in the context of graphical representations in score reports for teachers.

16.3.1 Ubiquitous Assessment

Assessments used to drive instructional decision making are often called *formative assessment*. Black and Wiliam (1998a, 1998b) note that teachers who employ formative assessment are often better at promoting learning than those who do not. However, formative assessment is usually associated with activities done by a classroom teacher in the course of instruction, not with special testing situations in which a formal score report is generated.

Although the two assessment purposes are often thought to be incompatible, this is not necessarily the case. Consider once again the assessment *for* learning system, ACED (Shute et al. 2008). In this system, students are asked to solve mathematics problems and provided feedback when they get incorrect answers. In the field trial of ACED, the group that showed significant learning gains had the highest correlation between the Bayesian network score from the activities and the posttest score. This suggests that the same assessment can be used for both formative and summative purposes. (One important caveat, in order to be used for formative purposes, is that the report/feedback must be available immediately, or at least very quickly. Often the lag between testing and score report generation prevents an assessment from being used for formative purposes.)

Taking the idea further, it may be the case that we only rarely need to take time out of the regular learning activities for assessment. *Any* activity performed by students potentially provides evidence about their knowledge, skills, and abilities: assessment need not be confined to special testing activities. Tanimoto (2001) calls this idea *unobtrusive assessment*, emphasizing the fact that assessment does not need to be segregated from normal classroom activities. Shute et al. (2009) call it *stealth assessment*, suggesting that assessment can be embedded in activities that students enjoy, like simulations and games. Almond (2010b) calls the idea *ubiquitous assessment*, emphasizing the fact that all activities performed by a student are potential evidence.[2]

For an activity to provide evidence about claims of interest, it really only requires an evidence model that links the natural work product to the proficiencies of interest. For example, if a student reads a web page with instructional material and then presses an "OK" button to go onto the next page, that button press is a potential work product. This is particularly weak evidence (a student could have pressed "OK" after only skimming the page). Usually, much stronger evidence is available, such as results from practice exercises or a more extended project or investigation.

The evidence from such found assessments may not be as strong as the evidence from a formally engineered assessment, but it might make up for what it lacks in quality in two ways: quantity and timeliness. If we are assessing all the time, we are not longer constrained by the time limits of formal assessments. Evidence is gathered as students work, so feedback can be made when it is most useful—in real time if desired, or in debriefing sessions the student, the teacher, or the system might propose. Teachers can gather evidence from extended tasks that require multiple sessions to complete.

They can also gather evidence from group work. This is another potential application of Bayesian networks. A group task is just like a task with multiple proficiency inputs. Some of the same design considerations that are required

[2] They may indeed be evidence, but whether they should be captured and how and when they should be used raises issues of privacy and consent. Bennett (2013) suggests that situations under which performance may be monitored and used should be clearly delineated from those that are not, stakeholders or their guardians must be informed of the situations and uses and provide consent, and data must be scrupulously protected.

to untangle evidence in diagnostic assessment are required to untangle evidence from multiple students. This particular application has not yet been extensively explored; some applications are reported in Chung et al. (2002) and Singley et al. (1999).

Finally, another challenge arises when setting up a ubiquitous assessment system: the evidence may be gathered over a long stretch of time. This means that students will grow over the time scale of assessment. The methods of Sect. 16.2.2 need to be pressed into service to take this critical feature of the inferential situation into account. Note that these models address the instructional part of the activity and not just the assessment part.

16.4 Evidence-Centered Assessment Design and Validity

There is nothing special about Bayesian networks that forces assessments that use them to be valid. The evidence-centered assessment design process, however, encourages test designers to think about the relationship between tasks used to gather evidence of student proficiency and the claims that the designers wish to make about students. Evidence-centered assessment design makes the "construct representation" line of a validity argument (Embretson 1983) central to the design process, and encourages test designers to document the rationale for key design decisions (Kane 2006). A particular task is included in an assessment precisely because it provides evidence for a claim of interest. To the extent that test designers document the sources that went into establishing the evidential relationships, this provides a public strand of the validity argument. Similarly, tasks that do not provide evidence, or only provide weak evidence for the claims of interest, are not included in the assessment.

Ultimately, the measurement model of the assessment is only a model. This book has encouraged readers to think about assessment design as a process of model building. The specific details about the computational techniques, whether the proficiency variables are discrete or continuous, and whether the model is based on classical test theory, IRT, or Bayesian networks are all secondary concerns. The real questions are these: (1) Is the model an adequate representation of the theory of cognition that underlies the assessment for the purpose at hand? (2) Can the model adequately account for student's performance? (3) How well do the scores in fact serve the purpose of the assessment?

The ECD process focuses on the first question, the constructive part. It helps the designer devise task situations, scoring rubrics, and psychometric models to synthesize evidence that will all be pertinent to the purpose of the assessment. The Bayesian paradigm offers some assistance for the second question too, the model-fit part. Any assessment instrument needs to be subjected to a program of field testing designed to address the model fit. Any pattern of responses will have a prior probability. If too many surprising events occur, the model is brought into question (Chap. 10). Such results can be used to refine the model, and to the extent to which the model reflects the experts' theory of cognition, used to identify weaknesses in that theory.

As these pilot studies are often small scale, they can take advantage of that size to look at kinds of tasks that would not be feasible in a larger study. In particular, in early stages of the assessment design, it is often useful to look at natural activities that represent valued work in the domain of interest. The ideal task might be to simply videotape and rate the student's performance in that authentic activity. While this is not feasible as part of a large-scale testing program, it is feasible as part of a small-scale validity study.

The third question, how well the assessment serves the purpose, goes beyond how it is constructed and whether the data fit the model. Additional studies are needed to see, for example, if better decisions are made using the assessment rather than an alternative, or if its instructional recommendations do in fact lead to better learning.

In these investigations, data can be used to produce evidence models that link authentic criterion activities to the proficiency model of the assessment. The assessment could then predict student's performance on these tasks based on the proficiency estimates. This is related to the idea of market basket reporting (DeVito and Koenig 2000), but has the advantage of focusing on tasks that represent valued work in the domain of interest— tasks that the score users ultimately care about.

Another important aspect of validity is the relationship between the states of the proficiency variables and educational standards set by various government bodies and consortia. The traditional approach to standards and assessments is retrospective: The standards are defined, the assessment is defined and pretested, and then a panel of experts is convened to determine the passing scores. Bayesian networks and ECD encourage an approach that is prospective (Bejar et al. 2007): The standards are defined and then tasks are selected for the assessment that provide evidence about whether or not the candidate meets the standards. Intuitively, it would seem that the prospective approach should yield instruments that are better focused on making judgments about the standards. This has yet to be verified empirically. Furthermore, it is quite possible that there will be difficulties in the design or implementation of the assessment, which makes it measure something other than what was intended. Thus, the prospective approach to standard setting still requires validation. Procedures for that validation are still largely unexplored.

16.5 What We Still Do Not Know

Ultimately, the way we will learn more about Bayesian networks in educational assessment is from people trying to use Bayesian networks in real applications. Much of the content of this book is driven by what we learned from HYDRIVE

(Gitomer et al. 1995; Mislevy and Gitomer 1996), DISC (Mislevy et al. 1999b; Mislevy et al. 2002d), Biomass (Steinberg et al. 2003, Chap. 14), NetPASS (Behrens et al. 2004), ACED (Shute 2004; Shute et al. 2005, 2008), and the alternative scoring research for Educational Testing Service's (ETS) internet and computing technology (ICT) Literacy Assessment (Katz et al. 2004). We have learned also from applications of others, such as those of Shute et al. (2009), Martin and VanLehn (1994), and Iseli et al. (2010). Applications like these help sort out the problems that are important because they block progress toward practical applications from problems that are of a more theoretical nature.

As Bayesian networks provide an alternative notation for familiar psychometric models, a large number of theoretical issues arise in studying how familiar psychometric principles and issues play out in Bayesian networks. In some cases, the Bayesian network view may help bring clarity (for example, casting differential item functioning as a question of conditional independence; Sect. 10.4). In other cases, our lack of experience with Bayesian networks makes them harder to use at first. In particular, using Bayesian networks, it is easy to specify models that are only weakly identifiable from data, or parts of models that can only be identified from the prior distribution. A lot is known about model identification in factor analysis and structural equation modeling, but relatively little in Bayesian networks.

One question that has arisen in many of our conversions is how does one equate a Bayesian network assessment. Standard equating is critical when students are compared for high stakes purposes, and the evidence coming from different test forms needs to be equivalent with regard to construct representation and measurement error. For these situations, we note that it is possible to use multiple evidence accumulation processes, and a program can still use standard test design and scoring procedures that ensure equivalency of this kind when they need to, with a parallel Bayes net evidence accumulation engine to produce more detailed feedback.

But high-stake purposes and strictly equated tests are not really playing to the strengths of Bayes nets. For the purposes of modeling complex evidentiary relationships in assessments closer to learning, the key issue is calibrating the assessment so that the proficiency variables actually represent the proficiencies they are named after. Two alternative forms of the same assessment share a common proficiency model but have different evidence models for the different tasks. If both forms were calibrated in a common framework (Sect. 9.6.2), they are linked in a sense that they both provide estimates of the same proficiencies.

The greatest value of Bayesian networks lies in their application to problems of cognitive diagnosis and inference from complex assessments, such as simulations and games meant to support learning, where true equating is not necessary and linking is sufficient. (Beware of mission creep, though. It is not uncommon for an assessment created for one purpose, say low-stake instructional guidance, to be adopted for a different purpose, say a selection decision or or teacher evaluation. In these cases, it is necessary to reexamine the entire assessment argument, not just the measurement model.)

One practical problem that arises in almost all applications is the grain-size problem. How detailed should the proficiency model be? How many variables? How many levels for each variable? Cognitive scientists like to make highly detailed models that match their detailed knowledge of the domain. Psychometricians prefer simpler models that require less evidence to estimate proficiencies, especially when the time available for testing is limited. There is no "right" answer to this problem: every application must find a solution that is appropriate to the constrains and purposes of the assessment to be designed. One good rule is that the grain size needs to be fine enough to support whatever decisions will be made. A corollary follows: In early design phases, modeling at one level (more detailed) can be useful to think about finer-grained aspects of proficiency and features of situations to evoke them. Design at this level will effectively define the proficiencies that do appear in the model.

Ultimately, good assessments are made from good tasks. Even complex psychometric models can only go so far in extracting evidence from tasks that do not provide adequate evidence. The potential of Bayes nets for educational assessment comes from their properties of modularity, computational qualities, and the flexibility to handle coherently different configurations of evidence. The potential will only be realized through thoughtful design and modeling of educationally meaningful observation situations, which can evoke good evidence about the right capabilities. We have tried to give our readers a foundation to take advantage of these features in a range of new forms of assessment in this book. We cannot wait to see what they accomplish.

Exercises

16.1. Design an assessment that uses Bayesian networks as a scoring model. Use the principles of evidence-centered assessment design. Publish your assessment model.

16.2. Join the ECD wiki (`http://ecd.ralmond.net/ecdwiki/`) and contribute to the discussion about evidence-centered assessment design. (Contact the authors for an editing password.)

16.3. The authors of this book have made several mistakes and the book contains several important omissions (not that we meant to!). Find them and bring them to the attention of the authors.

A

Bayesian Network Resources

This appendix provides pointers to online versions of various resources, which may be useful when studying Bayesian networks. This includes two kinds of resources: software packages we have found useful and sample Bayesian networks and data for study.

The risk with including internet links in a printed book is that they can become dated long before the text, and that there is no easy way to update them. Therefore, we have mirrored most of the content in this appendix on the evidence-centered design (ECD) Wiki (`http://ecd.ralmond.net/ecdwiki/`). The ECD Wiki can be read by anyone, but is only open for editing by members of the Bayes net community. (Congratulations! By reading this book you have become a member of that community. Send email to the authors mailto:`almond@acm.org` to get the editing password.[1]) The resources and errata for this book (including most of the contents of this appendix) are available on this web site, `http://ecd.ralmond.net/ecdwiki/BN/BN`.

We, the authors, do not believe that we have created a complete or perfect description of ECD or the use of Bayesian networks in assessment. In our experience, each time we undertake a new project, the project takes us in new directions. We have found others' questions about and perspectives on ECD to be valuable in refining our thinking, so we hope you will become part of the conversation.

A.1 Software

In preparing this book, we have used three different types of software: Basic Bayesian network manipulation software (Sect. A.1.1), software for construct-

[1] If you come across a page reference that seems to require a password to access, that means that the page has not been written yet, and the computer is asking if you want to write the page. We are looking for enterprising individuals to help us fill in the content. Please volunteer.

R. G. Almond et al., *Bayesian Networks in Educational Assessment*,
Statistics for Social and Behavioral Sciences, DOI 10.1007/978-1-4939-2125-6

ing Bayesian networks by hand (Sect. A.1.2), and software for estimating parameters using Markov chain Monte Carlo (MCMC) (Sect. A.1.3).

A.1.1 Bayesian Network Manipulation

Shortly after the publication of Pearl (1988), a large number of software packages started appearing, which could carry out the basic manipulation of Bayesian networks described in Chap. 5. Some of these survived and others did not. Almond (1995) had a list of packages available at the time, and I [Russell] tried to keep up with the changes on the internet, but soon gave up. Not only did packages appear and disappear, but vendors also rearranged their web sites, so links quickly broke. The solution turned out to be a wiki, where many hands could help fix broken links. Currently, the best list of available software is the Wikipedia article on Bayesian networks (`https://en.wikipedia.org/wiki/Bayesian_network`). The list of applications at the end includes both free and commercial software.

We describe only four software packages below because they have played a special role in the creation of this book.

`Netica`: Netica® is a commercial Bayesian network software environment offered by Norsys, LLC (`http://www.norsys.com/`). Netica comes in two formats: a graphical interface version and an API for embedding in other applications. Netica is available in both the full commercial version and a free student version, which limits the size of a network that can be saved. The student version is adequate for studying the examples in this course, but you will want the full commercial version for a serious project.
The Netica graphical interface works only in Microsoft Windows®. Norsys supports Netica when used with various tools for running Windows programs on other platforms. We have used Netica successfully with both WINE (`http://www.winehq.org/`) and CrossOver (`http://www.codeweavers.com/products/`).

`GeNie` and `Smile`: GeNie (graphical interface version) and Smile (API version) are Bayesian network software packages offered by the Decision Systems Laboratory, University of Pittsburgh (`http://genie.sis.pitt.edu/`). A notable feature of these packages is that they will translate Bayesian networks files from one format to another, so this is a useful tool when transporting networks. We have used this tool to make alternate versions of various Bayesian network packages available.

`StatShop`: StatShop was an internal Bayesian network package developed at ETS (Almond, Yan, et al. 2006c). It is still a research prototype and was never formally released, but interested people can write to Duanli Yan (`mailto:dyan@ets.org`) to request a license. Once you have a license, you can contact Russell Almond (`mailto:almond@acm.org`) for further instructions and help. A word of warning! Although StatShop runs all of the examples described here, it has never really undergone the refinement it needs to be a standalone tool. Considerable programming expertise is probably necessary to get it running properly.

`HUGIN`: HUGIN Expert A/S (`http://www.hugin.com/`) is mentioned several times in this book as it plays an important role as one of the first commercial Bayes net packages. It was created by taking the Bayes net software for the the MUNIN network (Andreassen et al. 1987) and modifying it to support arbirary networks (Andersen et al. 1989). Twenty-five years later it is still being actively developed, supported and improved.

A.1.2 Manual Construction of Bayesian Networks

Constructing a Bayesian network for an assessment starting from the cognitive analysis of the domain through the definition of the network structure and conditional probabilities is a complex process. This book attempts to discuss many of the issues that come up, but each project offers unique challenges. The Bayesian network software described in the previous section only supports the last part of the process. Typically, Bayesian network software only offers support for creating the graphical structure and adding the numbers. Other tools are necessary to manage the knowledge that goes into the model construction.

While we were constructing Biomass (Chaps. 14 and 15), we used a custom tool called Portal. When working on other projects, we found that Portal was both too complicated and not flexible enough to support our needs. Instead, what we found was that it was better for each design team to develop their own set of custom forms and spreadsheets in which to capture the knowledge important to the particular assessment purpose. The programmers could then extract information from these tables to build the Bayesian networks (Almond 2010a).

The three R (R Development Core Team 2007) packages described here are useful for writing programs that parse other data files and construct Bayesian networks. All three are available as free downloads from `http://pluto.coe.fsu.edu/RNetica/`.

`CPTtools`: This package has R code available to build conditional probability tables using the DiBello–Samejima distribution and some of the other methods described in Chap. 8. It also has R functions for the evidence balance sheet (Chap. 7) and the observable characteristic plot (Chap. 10). It does not require any other packages to run.

`RNetica`: This is basically an R binding for the Netica API, and requires a Netica API license (see Sect. A.1.1). Together with CPTtools, most of the packages in this book can be constructed in Netica.

`SSX`: The StatShop XML (SSX) package provides tools for reading/writing StatShop (see Sect. A.1.1) network files.

A.1.3 Markov Chain Monte Carlo

Chapter 9 describes two different algorithms for estimating Bayesian network parameters from data: the expectation–maximization algorithm (EM algorithm) and MCMC. Many Bayesian network packages have some kind of parameter learning built in, which is usually a variant of the EM algorithm. (StatShop had a built-in MCMC engine.) Once again, the Wikipedia page on Bayesian networks has a good list of available software.

Often, however, we have found that the best approach is to write custom model estimation code in BUGS (Bayesian inference Using Gibbs Sampling) (Spiegelhalter et al. 1995) or a similar general purpose MCMC package. The Windows version WinBUGS (Lunn et al. 2000) has a convenient graphical interface that makes it easy for beginners to get started; however, WinBUGS is no longer maintained, and hence better alternatives are available for serious work. We recommend either OpenBUGS (`http://www.openbugs.info/w/FrontPage`; Lunn et al. 2009) or JAGS (Just Another Gibbs Compiler) (`http://www-fis.iarc.fr/~martyn/software/jags/`; Plummer 2012). The package Stan (`http://mc-stan.org/`) is a promising alternative, but currently, it lacks support for the discrete nodes needed for discrete Bayesian networks (this may be fixed by the time you read this).

A.2 Sample Bayesian Networks

As we have used the book in instruction, we have found it helpful for students to work through some larger scale examples of Bayesian networks. To that end we provide a number of networks for use by readers of the book. Instructionally, these provide excellent tools for homework and projects.

Each network is provided in a number of different formats (see the listing of tools in Sect. A.1.1). First, they are provided in StatShop XML and HTML formats. The HTML version of the network provides a complete human-readable description of every conditional probability table. Second, they are provided as a collection of Netica networks. Note that GeNie is able to convert the Netica file to formats that are compatible with a number of other Bayesian network programs.

The following examples are provided at `http://ecd.ralmond.net/ecdwiki/BN/BN`:

Evidence Model Student Model: This is a very small example with a three proficiency variables and four evidence model Bayes net fragments used for testing transferring information between evidence model and proficiency model fragments (Sect. 5.4; Almond and Mislevy 1999; Almond et al. 1999).

IRT5: This is the simple five-item IRT models with and without context effects explored in Sects. 6.1 and 6.2. This network is used Examples 6.1, 6.2, 6.3, 6.4, and 10.1.

Design: This is a small example contrasting the Compensatory, Conjunctive, and Disjunctive design patterns discussed in Chap. 8.

Latent Class: This is the small latent class model used in Chap. 9. It is used in Examples 9.1, 9.2, 9.3, and 9.4.

`Language Testing:` This is a language exam containing both single modality and integrated tasks first described in Mislevy (1995c). It is used in Examples 1.1, 7.1, 7.6, 7.7, and 7.8. It has several instances of each task, so that it can be used for simulation experiments, and has several simulated data sets.

Mixed Number Subtraction: This is the classic mixed-number subtraction test of Tatsuoka (1983) adapted into a Bayesian network by Mislevy (1994); Mislevy (1995b). As this is one of the first published examples of a cognitively diagnostic assessment, this example has seen a lot of use, including use in Sect. 6.4 and Chap. 11. Only the Bayesian network is provided, not the data gathered by Tatsuoka et al.

ACED: Adaptive content with evidence-based diagnosis (ACED) is one of the first fielded diagnostic assessments using Bayesian networks as a scoring model (Shute et al. 2005; Shute et al. 2007; Shute et al. 2008). The web site contains the proficiency model, the evidence models for all 63 tasks and data from over 200 students involved in the field trial. ACED is explored in Examples 7.5 and 13.1.

References

Adams, D. (1978). *Hitchhiker's guide to the galaxy: The primary phase* (Vinyl LP ed.). London: BBC. LP record.

Adams, R. J., Wilson, M. R., Wang, W.-C. (1997). The multidimensional random coefficients multinomial logit model. *Applied Psychological Measurement*, *21*, 1–23.

Akaike, H. (1973). Information theory and an extension of the maximum likelihood principle. In B. N. Petrov F. Cáki (Eds.), *Proceedings of the 2nd international symposium on information theory* (pp. 267–281). Budapest: Akademiai Kiado.

Aleven, V., Koedinger, K. R. (2002, Mar-Apr). An effective metacognitive strategy: Learning by doing and explaining with a computer-based cognitive tutor. *Cognitive Science*, *26*(2), 147–179.

Almond, R. G. (1995). *Graphical belief modeling.* London: Chapman and Hall. Retrieved from `http://www.crcpress.com/product/isbn/9780412066610`

Almond, R. G. (2007a). Cognitive modeling to represent growth (learning) using Markov decision processes. *Technology, Instruction, Cognition and Learning (TICL)*, *5*, 313–324. Retrieved from `http://www.oldcitypublishing.com/TICL/TICL.html`.

Almond, R. G. (2007b). *An illustration of the use of Markov decision processes to represent student growth (learning)* (ETS Research Report No. RR-07-40). Princeton: Educational Testing Service. Retrieved from `http://www.ets.org/research/researcher/RR-07-40.html`.

Almond, R. G. (2010a). "I can name that Bayesian network in two matrixes". *International Journal of Approximate Reasoning*, *51*, 167–178. Retrieved from `http://dx.doi.org/10.1016/j.ijar.2009.04.005`. doi: 10.1016/j.ijar.2009.04.005.

Almond, R. G. (2010b). Using evidence centered design to think about assessments. In V. J. Shute B. J. Becker (Eds.), *Innovative assessment for the 21st century: Supporting educational needs* (pp. 75–100). New York: Springer. doi: 10.1007/978-1-4419-6530-1_6.

R. G. Almond et al., *Bayesian Networks in Educational Assessment*,
Statistics for Social and Behavioral Sciences, DOI 10.1007/978-1-4939-2125-6

Almond, R. G., DiBello, L. V., Jenkins, F., Mislevy, R. J., Senturk, D., Steinberg, L. S., Yan, D. (2001). Models for conditional probability tables in educational assessment. In T. Jaakkola T. Richardson (Eds.), *Artificial intelligence and statistics 2001* (pp. 137–143). San Francisco: Morgan Kaufmann.

Almond, R. G., Herskovits, E., Mislevy, R. J., Steinberg, L. S. (1999). Transfer of information between system and evidence models. In D. Heckerman J. Whittaker (Eds.), *Artificial intelligence and statistics 99* (pp. 181–186). San Francisco: Morgan Kaufmann.

Almond, R. G., Kim, Y. J., Shute, V. J., Ventura, M. (2013). Debugging the evidence chain. In R. G. Almond O. Mengshoel (Eds.), *Proceedings of the 2013 uai application workshops: Big data meet complex models and models for spatial, temporal and network data (uai2013aw)* (pp. 1–10). Aachen. Retrieved from `http://ceur-ws.org/Vol-XXX/paper-01.pdf`.

Almond, R. G., Mislevy, R. J. (1999). Graphical models and computerized adaptive testing. *Applied Psychological Measurement*, *23*, 223–238.

Almond, R. G., Mislevy, R. J., Williamson, D. M., Yan, D. (2006a, April). *Bayesian networks in educational assessment.* Paper presented at Annual meeting of the National Council on Measurement in Education (NCME). San Francisco, CA.

Almond, R. G., Mislevy, R. J., Williamson, D. M., Yan, D. (2007, April). *Bayesian networks in educational assessment.* Paper presented at Annual meeting of the National Council on Measurement in Education (NCME) Chicago, IL.

Almond, R. G., Mislevy, R. J., Williamson, D. M., Yan, D. (2010, April). *Bayesian networks in educational assessment.* Paper presented at annual meeting of the National Council on Measurement in Education (NCME). Denver, CO.

Almond, R. G., Mulder, J., Hemat, L. A., Yan, D. (2006b). *Models for local dependence among observable outcome variables* (ETS Research Report No. RR-06-36). Princeton: Educational Testing Service. Retrieved from `http://www.ets.org/research/researcher/RR-06-36.html`.

Almond, R. G., Shute, V. J., Underwood, J. S., Zapata-Rivera, J.-D. (2009a). Bayesian networks: A teacher's view. *International Journal of Approximate Reasoning*, *50*, 450–460. doi: 10.1016/j.ijar.2008.04.011.

Almond, R. G., Shute, V. J., Underwood, J. S., Zapata-Rivera, J.-D. (2009b). Bayesian networks: A teacher's view. *International Journal of Approximate Reasoning*, *50*, 450–460.

Almond, R. G., Steinberg, L. S., Mislevy, R. J. (2002a). Enhancing the design and delivery of assessment systems: A four-process architecture. *Journal of Technology , Learning, and Assessment*, *1*(5), 1–63. Retrieved from `http://ejournals.bc.edu/ojs/index.php/jtla/article/view/1671`.

Almond, R. G., Steinberg, L. S., Mislevy, R. J. (2002b). A framework for reusing assessment components. In H. Yanai, O. A., K. Shigemasu, Y. Kano, J. J. Meulman (Eds.), *New developments in psychometrics* (pp. 281–288). Tokyo: Springer.

Almond, R. G., Yan, D., Hemat, L. A. (2008). Parameter recovery studies with a diagnostic Bayesian network model. *Behaviormetrika*, *35*(2), 159–185.

Almond, R. G., Yan, D., Matukhin, A., Chang, D. (2006c). *StatShop testing* (Research Memorandum No. RM-06-04). Princeton: Educational Testing Service.

Alonzo, A. C., Gotwals, A. W. (Eds.). (2012). *Learning progressions in science: Current challenges and future directions.* Rotterdam: Sense.

American Association for the Advancement of Science. (1994). *Benchmarks for scientific literacy.* New York: Oxford University Press.

Andersen, S. A., Madigan, D., Perlman, M. D. (1996). A characterization of Markov equivalence classes for acyclic digraphs. *Annals of Statistics*, *25*, 505–541.

Anderson, R. D., Vastag, G. (2004). Causal modeling alternative in operations research: Overview and application. *European Journal of Operational Research*, *156*(1), 92–109. doi: 10.1016/s0377-2217(02)00904-9.

Andreassen, S., Woldbye, M., Falck, B., Andersen, S. K. (1987). Munin—a causal probabilistic network for interpretation of elecromyographic findings. In J. P. McDermott (Ed.), *Proceedings of the 10th international joint conference on artificial intelligence* (Vol. 1, pp. 366–372). San Francisco, CA.

Andrich, D. (1985). A latent trait model for items with response dependencies: Implications for test construction and analysis. In S. E. Embretson (Ed.), *Test design: Developments in psychology and psychometrics* (pp. 245–275). New York: Academic.

Attali, Y., Burstein, J. (2006). Automated essay scoring with e-rater® v. 2.0. *The Journal of Technology, Learning, and Assessment*, *4*(3), 13–18. Retrieved from `http://escholorship.bc.edu/jtla/vol4/3/`.

Bacchetti, P., Segal, M. R., Jewell, N. P. (1993). Backcalculation of HIV infection rates (with discussion). *Statistical Sciences*, *8*, 82–119.

Bachman, L. F., Palmer, A. S. (1996). *Language testing in practice: Designing and developing useful language tests* (Vol. 1). Oxford: Oxford University Press.

Baker, R. S. J. d., Corbett, A. T., Gowda, S. M., Wagner, A. Z., MacLaren, B. A., Kauffman, L. R., et al. (2010). Contextual slip and prediction of student performance after use of an intelligent tutor. In P. De Bra, A. Kobsa, D. Chin (Eds.), *User modeling, adaptation, and personalization* (pp. 52–63). New York: Springer.

Baldwin, D., Fowles, M., Livingston, S. (2008). *Guidelines for constructed-responses and other performance assessments* [Research Report]. Princeton: Educational Testing Service. Retrieved from `http://`

www.ets.org/Media/About_ETS/pdf/8561_ConstructedResponse_guidelines.pdf.

Barr, A., Feigenbaum, E. (1982). *Handbook of artificial intelligence* (Vol. 2). Los Altos: HeurisTech.

Bart, W. M., Post, T., Behr, M. J., Lesh, R. (1994). A diagnostic analysis of a proportional reasoning test item: An introduction to the properties of a semi-dense item. *Focus on Learning Problems in Mathematics*, *16*(3), 1–11.

Barton, P. E. (2003). *Parsing the achievement gap: Baselines for tracking progress* (Policy Information Center Report). Educational Testing Service. Retrieved from http://www.ets.org.

Beaton, A. E., Allen, N. L. (1992). Interpreting scales through scale anchoring. *Journal of Educational Statistics*, *17*(2), 192–204.

Behrens, J. T., Mislevy, R. J., Bauer, M. I., Williamson, D. M., Levy, R. (2004). Introduction to evidence centered design and lessons learned from its application in a global e-learning program. *International Journal of Measurement*, *4*, 295–301.

Behrens, J. T., Mislevy, R. J., DiCerbo, K. E., Levy, R. (2012). An evidence centered design for learning and assessment in the digital world. In M. C. Mayrath, J. Clarke-Midura, D. Robinson (Eds.), *Technology-based assessments for 21st century skills: Theoretical and practical implications from modern research* (pp. 13–54). Charlotte: Information Age.

Bejar, I. I., Braun, H. I., Tannenbaum, R. (2007). A prospective, predictive and progressive approach to standard setting. In R. L. Lissitz (Ed.), *Assessing and modeling cognitive development in school: Intellectual growth and standard setting* (pp. 1–30). Maple Grove: JAM.

Bejar, I. I., Williamson, D. M., Mislevy, R. J. (2006). Human scoring. In D. M. Williamson, R. J. Mislevy, I. I. Bejar (Eds.), *Automated scoring of complex tasks in computer-based testing* (pp. 49–82). Mahwah: Lawrence Erlbaum.

Bennett, R. E. (2013). *Preparing for the future: What educational assessment must do.* Princeton: The Gordon Commission. Retrieved from http://www.gordoncommission.org/rsc/pdf/bennett_preparing_future_assessment.pdf.

Berger, J. O. (1985). *Statistical decision theory and Bayesian analysis.* New York: Springer.

Berliner, M. (2005, May). Physical-statistical modeling and prediction. Paper presented at "Some Challenging Applications of Statistical Modeling and Analysis," a special seminar series presented at Harvard University on the occasion of the retirement of Arthur P. Dempster. Retrieved from http://www.stat.harvard.edu/Dempster_Symposium/Berliner.pdf.

Bertelè, U., Brioschi, F. (1972). *Nonserial dynamic programming.* New York: Academic.

Best, N. G., Cowles, M. K., Vines, K. (1996). Coda: Convergence diagnosis and output analysis software for Gibbs sampling output version 0.30 [Computer software manual]. Cambridge, UK.

Bishop, Y. M., Fienberg, S. E., Holland, P. W. (1975). *Discrete multivariate analysis.* Cambridge: MIT Press.

Black, P., Wiliam, D. (1998a). Assessment and classroom learning. *Assessment in Education: Principles, Policy, and Practice*, *5*(1), 7–74.

Black, P., Wiliam, D. (1998b). Inside the black box: Raising standards through classroom assessment. *Phi Delta Kappan*, *80*(2), 139–147.

Bock, R. D. (1972). Estimating item parameters and latent ability when responses are scored in two or more nominal categories. *Psychometrika*, *37*(1), 29–51.

Bock, R. D., Aitkin, M. (1981). Marginal maximum likelihood estimation of item parameters: An application of an EM-algorithm. *Psychometrika*, *46*, 443–459.

Bock, R. D., Lieberman, M. (1970). Fitting a response model for n dichotomously scored items. *Psychometrika*, *35*, 179–197.

Bock, R. D., Mislevy, R. J. (1982). Adaptive EAP estimation of ability in a microcomputer environment. *Applied Psychological Measurement*, *6*(4), 431–444.

Bollen, K. A. (1989). *Structural equations with latent variables.* New York: Wiley.

Boutilier, C., Dean, T., Hanks, S. (1999). Decision-theoretic planning: Structural assumptions and computational leverage. *Journal of Artificial Intelligence Research*, *11*, 1–94. Retrieved from `citeseer.ist.psu.edu/viewdoc/download?doi=10.1.1.65.9397&rep=rep1&type=pdf`.

Bowman, E. H. (1963). Consistency and optimality in managerial decision making. *Management Science*, *9*(2), 310–321.

Box, G. E. P. (1976). Science and statistics. *Journal of the American Statistical Association*, *71*, 791–799.

Box, G. E. P., Tiao, G. C. (1973). *Bayesian inference in statistical analysis.* New York: Wiley.

Boyen, X., Koller, D. (1998). Tractable inference for complex stochastic process. In G. Cooper S. Moral (Eds.), *Uncertainty in Artificial Intelligence: Proceedings of the 14th Annual Conference* (pp. 33–42). San Francisco: Morgan Kaufmann.

Bradlow, E. T., Wainer, H., Wang, X. (1999). A Bayesian random effects model for testlets. *Psychometrika*, *64*, 153–168.

Braun, H. I., Bejar, I. I., Williamson, D. M. (2006). Rule-based methods for automated scoring. In D. M. Williamson, R. J. Mislevy, I. I. Bejar (Eds.), *Automated scoring of complex tasks in computer-based testing* (pp. 83–122). Hillsdale: Lawrence Erlbaum.

Breese, J. S., Goldman, R. P., Wellman, M. P. (1994). Introduction to the special section on knowledge-based construction of probabilistic and decision models. *IEEE Transactions on System, Man and Cybernetics*, *24*, 1577–1579.

Breiman, L., Friedman, J. H., Olshen, R., Stone, C. J. (1984). *Classification and regression trees.* Belmont: Wadsworth.

Breland, H. M., Camp, R., Jones, R. J., Morris, M. M., Rock, D. A. (1987). *Assessing writing skills.* New York, NY: College Entrance Examination Board.

Brennan, R. L. (2001). An essay on the history and future of reliability from the perspective of replications. *Journal of Educational Measurement*, *38*(4), 295–317.

Brennan, R. L., Prediger, D. J. (1977). *Coefficient kappa: Some uses, misuses, and alternatives* (Technical Bulletin No. 29). ACT.

Breyer, F. J., Mislevy, R. J., Steinberg, L. S., Almond, R. G. (1999, April). Designing technology-based assessments: It's the evidence for the inferences that are important. Paper presented at the Annual Convention of the Society for Industrial Organizational Psychology, Atlanta, GA.

Bridgeman, B., Lennon, M. L., Jackenthal, A. (2001). *Effects of screen size, screen resolution, and display rate on computer-based test performance* (Research Report No. RR-01-23). Princeton: Educational Testing Service.

Brooks, S., Gelman, A. (1998). General methods for monitoring convergence of iterative simulations. *Journal of Computational and Graphical Statistics*, *7*, 434–55.

Bunt, A., Conati, C. (2002). Assessing effective exploration in open learning environments using Bayesian networks. In S. A. Cerri, G. Gouarderes, & F. Paraguacu (Eds.), *Proceedings of ITS 2002, 6th international conference on intelligent tutoring systems, Biarritz, France, 4–7 June 2002.*

Bunt, A., Conati, C. (2003). Probabilistic student modelling to improve exploratory behaviour. *User Modeling and User-Adapted Interaction*, *13*(3), 269–309.

Buntine, W. L. (1994). Operations for learning with graphical models. *Journal of Artificial Intelligence Research*, *2*, 159–225.

Buntine, W. L. (1996). A guide to the literature on learning probabilistic networks from data. *IEEE Transactions on Knowledge and Data Engineering*, *8*, 195–210.

Burstein, J., Tetreault, J., Madnani, N. (2013). The E-rater® automated essay scoring system. In M. D. Shermis J. J. Burstein (Eds.), *Handbook of automated essay evaluation: Current applications and new directions* (pp. 55–67). New York: Routledge.

Cai, L. (2008). SEM of another flavour: Two new applications of the supplemented em algorithm. *British Journal of Mathematical and Statistical Psychology*, *61*, 309–329.

Cannings, C., Thompson, E. A., Skolnick, M. H. (1978). Probability functions on complex pedigrees. *Advances in Applied Probability*, *10*, 26–61.

Cen, H., Koedinger, K. R., Junker, B. W. (2006). Learning factors analysis: A general method for cognitive model evaluation and improvement. In M. Ikeda, K. Ashlay, T.-W. Chan (Eds.), *Intelligent tutoring systems, 8th international conference. Lecture notes in computer science: Vol. 4053.* (pp. 164–175). Berlin: Springer.

Chaloner, K. M., Duncan, G. T. (1983). Assessment of a beta prior distribution: PM elicitation. *The Statistician*, *32*, 174–180.

Chambers, J. L. (2004). *Programming with data: A guide to the S language.* New York: Springer.

Chapelle, C., Enright, M., Jamieson, J. (2008). *Building a validity argument for the Test of English as a Foreign Language.* New York: Routledge.

Cheng, B. H., Ructtinger, L., Fujii, R., Mislevy, R. J. (2010). *Assessing systems thinking and complexity in science* (Large-Scale Assessment Technical Report No. 7). Menlo Park: SRI International. Retrieved from `http://ecd.sri.com/downloads/ECD_TR7_Systems_Thinking_FL.pdf`.

Cheng, Y. (2009). When cognitive diagnosis meets computerized adaptive testing: CD-CAT. *Psychometrika*, *74*(4), 619–632.

Chickering, D. (1996). Learning equivalence classes of Bayesian-network structures. In P. Besnard S. Hanks (Eds.), *Uncertainty in artificial intelligence: Proceedings of the 11th conference* (pp. 87–98). San Mateo: Morgan Kaufmann.

Chung, G. K. W. K., Delacruz, G. C., de Vries, L. F., Phan, C. H., Srivastava, M. B., Alarcon, R. (2002, September). *Fusing wireless sensor data to measure small-group collaborative processes in real-time.* Paper presented at the annual conference of the the National Center for Research on Evaluation, Standards, and Student Testing (CRESST), Los Angeles, CA.

Close, C. N., Davison, M. L., Davenport, E. (2012, April). *An exploratory technique for finding the Q-matrix in cognitive diagnostic assessment: Combining theory with data.* Paper presented at the annual meeting of the National Council on Measurement in Education, Vancouver, British Columbia, Canada.

Cobb, B. R., Shenoy, P. P. (2005). Hybrid Bayesian networks with linear deterministic variables. In F. Bacchus T. Jaakkola (Eds.), *Uncertainty in artificial intelligence: Proceedings of the 21st conference* (pp. 136–144). Corvallis: AUAI Press.

Collis, J. M., Tapsfield, P. G. C., Irvine, S. H., Dann, P. L., Wright, D. (1995). The British Army Recruit Battery goes operational: From theory to practice in computer-based testing using item generation techniques. *International Journal of Selection and Assessment*, *3*, 96–104.

Conati, C., Maclaren, H. (2009). Empirically building and evaluating a probabilistic model of user affect. *User Modeling and User-Adapted Interaction*, *19*, 267–303.

Conejo, R., Guzmán, E., Millán, E., Trella, M., Pérez-de-la-Cruz, J. L., Ríos, A. (2004). Siette: A web-based tool for adaptive testing. *International Journal of Artificial Intelligence in Education*, *14*(1), 29–61.

Cooper, G. F., Herskovits, E. (1992). A Bayesian method for the induction of probabilistic networks from data. *Machine Learning*, *9*, 309–347.

Corbett, A. T., Anderson, J. R. (1994). Knowledge tracing: Modeling the acquisition of procedural knowledge. *User Modeling and User-Adapted Interaction*, *4*, 253–278.

Corcoran, T., Mosher, F. A., Rogat, A. (2009). *Learning progressions in science: An evidence-based approach to reform* (Vol. 13; CPRE Research Report No. RR-63). Philadelphia: Consortium for Policy Research in Education.

Cowell, R. G., Dawid, A. P. (1992). Fast retraction of evidence in a probabilistic expert system. *Statistics and Computing*, *2*, 36–41.

Cowell, R. G., Dawid, A. P., Lauritzen, S. L., Spiegelhalter, D. J. (1999). *Probabilistic networks and expert systems.* New York: Springer.

Cowell, R. G., Dawid, A. P., Spiegelhalter, D. J. (1993). Sequential model criticism in probabilistic expert systems. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *15*, 209–129.

Cox, D. R., Wermuth, N. (1996). *Multivariate dependencies: Models, analysis and interpretation.* London: Chapman and Hall.

Cronbach, L. J. (1989). Intelligence: Measurement, theory, and public policy. In R. L. Linn (Ed.), *Construct validation after thirty years* (pp. 147–171). Champaign: University of Illinois Press.

Crowley, R., Medvedeva, O. (2006). An intelligent tutoring system for visual classification problem solving. *Artificial Intelligence in Medicine*, *36*(1), 85–117. doi: 10.1016/j.artmed.2005.01.005.

Daniel, B., Zapata-Rivera, J.-D., McCalla, G. I. (2003). A Bayesian computational model of social capital in virtual communities. In M. Huysman, E. Wenger, W. Volker (Eds.), *Proceedings of the first international conference on communities and technologies: C&T 2003.* Deventer: Kluwer Academic.

Darroch, J. N., Lauritzen, S. L., Speed, T. P. (1980). Markov fields and log-linear interaction models for contingency tables. *The Annals of Statistics*, *8*, 522–539.

Dawid, A. P. (1979). Conditional independence in statistical theory. *Journal of the Royal Statistical Society, Series B*, *41*, 1–31.

Dayton, C. M. (1999). *Latent class scaling analysis.* Thousand Oaks: Sage.

de Finetti, B. (1990). *Theory of probability, Volume I.* New York: Wiley.

de la Torre, J. (2008). An empirically based method of Q-matrix validation for the DINA model: Development and applications. *Journal of Educational Measurement*, *45*, 343–362.

de la Torre, J., Douglas, J. A. (2004). Higher-order latent trait models for cognitive diagnosis. *Psychometrika*, *69*, 333–353.

Dean, T., Kanazawa, K. (1989). A model for reasoning about persistence and causation. *Computer Intelligence*, *5*, 142–150.

Deane, P. (2006). Strategies for evidence identification through linguistic assessment of textual responses. In D. M. Williamson, R. J. Mislevy, I. I. Bejar (Eds.), *Automated scoring of complex tasks in computer-based testing* (pp. 313–371). Hillsdale: Lawrence Erlbaum.

Deane, P., Quinlan, T. (2010). What automated analyses of corpora can tell us about students' writing skills. *Journal of Writing Research*, *2*(2), 151–177.

DeGroot, M. H. (1970). *Optimal statistical decisions.* New York: McGraw-Hill.

Deming, W. E., Stephan, F. F. (1940). On a least squares adjustment of a sampled frequency table when the expected marginal totals are known. *The Annals of Mathematical Statistics*, *11*(4), 427–444.

Dempster, A. P. (1968). A generalization of Bayesian inference (with discussion). *Journal of the Royal Statistical Society, Series B*, *30*, 205–247.

Dempster, A. P. (1972). Covariance selection. *Biometrics*, *28*, 157–175.

Dempster, A. P. (1990). Bayes, Fisher and belief functions. In S. Geisser, J. S. Hodges, S. J. Press, A. Zellner (Eds.), *Bayesian likelihood methods in statistics and econometrics* (pp. 35–47). Amsterdam: Elsevier Science.

Dempster, A. P., Laird, N., Rubin, D. B. (1977). Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society (Series B)*, *39*, 1–38.

DeVito, P. J., Koenig, J. A. (Eds.). (2000). *Designing a market basket for NAEP.* Washington, DC: National. Retrieved from `http://www.nap.edu/catalog/9891.html`.

DiCerbo, K. E., Behrens, J. T. (2012). Implications of the digital ocean on current and future assessment. In R. L. Lissitz H. Jiao (Eds.), *Computers and their impact on state assessment* (pp. 273–306). Charlotte: Information Age.

Díez, F. J. (1993). Parameter adjustment in Bayes networks. the generalized noisy or-gate. In D. Heckerman A. Mamdani (Eds.), *Uncertainty in artificial intelligence: Proceedings of the 9th conference* (pp. 99–105). San Francisco: Morgan Kaufmann.

Díez, F. J., Druzdzel, M. J. (2006). *Canonical probabilistic models for knowledge engineering* (Technical Report No. CISIAD-06-01). Madrid: UNED.

Doucet, A., de Freitas, N., Gordon, N. (2001). *Sequential Monte Carlo methods in practice.* New York: Springer.

Draper, D. (1995). Assessment and propagation of model uncertainty. *Journal of the Royal Statistical Society (Series B)*, *57*, 45–98.

Draper, D., Hodges, J. S., Leamer, E. E., Morris, C. N., Rubin, D. B. (1987). *A research agenda for assessment and propagation of model uncertainty* (Rand Note No. N-2683-RC). Santa Monica: RAND.

Dwyer, C. A. (1998). Psychometrics of Praxis III: Classroom performance assessments. *Journal of Personnel Evaluation in Education*, *12*(2), 163–187.

Edwards, D. (1990). Hierarchical interaction models. *Journal of the Royal Statistical Society (Series B)*, *52*, 3–20.

Edwards, D. (1995). *Introduction to graphical modelling.* New York: Springer.

Efron, B. (1979). Bootstrap methods: Another look at the jackknife. *The Annals of Statistics*, *7*, 1–26.

El Saadawi, G. M., Tseytlin, E., Legowski, E., Jukic, D., Castine, M., Fine, J., et al. (2008). A natural language intelligent tutoring system for training pathologists: Implementation and evaluation. *Advances In Health Sciences Education*, *13*(5), 709–722. doi: 10.1007/s10459-007-9081-3.

Embretson, S. E. (1983). Construct validity: Construct representation versus nomothetic span. *Psychological Bulletin*, *93*, 179–197.

Embretson, S. E. (1998). A cognitive design system approach to generating valid tests: Application to abstract reasoning. *Psychological Methods*, *3*, 380–396.

Epstein, E. S. (1969). A scoring system for probability forecasts of ranked categories. *Journal of Applied Meteorology*, *8*, 985–987.

Feller, W. (1968). *An introduction to probability theory and its applications* (3rd ed.). New York: Wiley.

Fienberg, S. E. (1970). An iterative procedure for estimation in contingency tables. *The Annals of Mathematical Statistics*, *41*, 907–917.

Fischer, G. H. (1973). The linear logistic test model as an instrument in educational research. *Acta Psychologica*, *37*, 359–374.

Fleiss, J. L., Levin, B., Paik, M. C. (2003). *Statistical methods for rates and proportions.* New York: Wiley.

Formann, A. K. (1985). Constrained latent class models: Theory and applications. *British Journal of Mathematical and Statistical Psychology*, *38*, 87–111.

Freedman, D., Pisani, R., Purves, R. (1980). *Statistics.* New York: W. W. Norton.

Fulcher, G., Davidson, F. (2009). Test architecture, test retrofit. *Language Testing*, *26*(1), 123–144.

Gamboa, H., Fred, A. L. N. (2001). Designing intelligent tutoring systems: A Bayesian approach. In *Proceedings of the 3rd international conference on enterprise information systems (ICEIS 2002)* (Vol. 3, pp. 452–458). Setubal: ICEIS Press.

Geerlings, H., Glas, C. A. W., van der Linden, W. J. (2011). Modeling rule-based item generation. *Psychometrika*, *76*(2), 337–359.

Gelman, A., Carlin, J. B., Stern, H. S., Dunson, D. B., Vehtari, A., Rubin, D. B. (2013b). *Bayesian data analysis* (3rd ed.). London: CRC.

Gelman, A., Carlin, J. B., Stern, H. S., Rubin, D. B. (2013a). *Bayesian data analysis* (2nd ed.). London: Chapman and Hall.

Gelman, A., Carlin, J. B., Stern, H. S., Dunson, D. B., Vehtari, A., Rubin, D. B. (2013b). *Bayesian data analysis* (3rd ed.). London: CRC.

Gelman, A., Meng, X. L., Stern, H. S. (1996). Posterior predictive assessment of model fitness via realized discrepancies (with discussion). *Statistica Sinica*, *6*, 733–807.

Gelman, A., Rubin, D. B. (1992). Inference from iterative simulation using multiple sequences. *Statistical Science*, *7*, 457–511.

Geman, S., Geman, D. (1984). Stochastic relaxation, Gibbs distributions, and the Bayesian restoration of images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *6*, 721–741.

Gertner, A., Conati, C., VanLehn, K. (1998). Procedural help in Andes: Generating hints using a Bayesian network student model. In *Proceedings of the fifteenth national conference on artificial intelligence AAAI-98* (pp. 106–111). Cambridge: MIT Press.

Gierl, M. J., Haladyna, T. M. (2012). *Automatic item generation: Theory and practice.* New York: Routledge.

Gierl, M. J., Leighton, J. P., Hunka, S. M. (2007). Using the attribute hierarchy method to make diagnostic inferences about examinees' cognitive skills. In J. P. Leighton M. J. Gierl (Eds.), *Cognitive diagnostic assessment: Theories and applications* (pp. 242–274). Cambridge: Cambridge University Press.

Gilks, W. R., Richardson, S., Spiegelhalter, D. J. (Eds.). (1996). *Markov chain Monte Carlo in practice.* London: Chapman and Hall.

Gilula, Z., Haberman, S. J. (1995). Prediction functions for categorical panel data. *The Annals of Statistics*, *23*(4), 1130–1142.

Gilula, Z., Haberman, S. J. (2001). Analysis of categorical response profiles by informative summaries. *Sociological Methodology*, *31*(1), 129–187.

Gitomer, D. H., Steinberg, L. S. (1999). Representational issues in assessment design. In I. E. Sigel (Ed.), *Development of mental representation* (pp. 351–370). Mahwah: Lawrence Erlbaum.

Gitomer, D. H., Steinberg, L. S., Mislevy, R. J. (1995). Diagnostic assessment of trouble-shooting skill in an intelligent tutoring system. In P. D. Nichols, S. F. Chipman, R. L. Brennan (Eds.), *Cognitively diagnostic assessment* (pp. 73–101). Mahwah: Lawrence Erlbaum.

Glas, C. A. W., Falcón, J. C. S. (2003). A comparison of item-fit statistics for the three-parameter logistic model. *Applied Psychological Measurement*, *27*(2), 87–106.

Glas, C. A. W., van der Linden, W. J. (2001, July). *Modeling variability in item parameters in CAT.* Paper presented at the International Meeting of the Psychometric Society, Osaka, Japan.

Glas, C. A. W., van der Linden, W. J. (2003). Computerized adaptive testing with item cloning. *Applied Psychological Measurement*, *27*, 247–261.

Glasziou, P., Hilden, J. (1989). Test selection measures. *Medical Decision Making*, *9*, 133–141.

Gobert, J. D., Sao Pedro, M. A., Baker, R. S. J. d., Toto, E., Montalvo, O. (2012). Leveraging educational data mining for real time performance assessment of scientific inquiry skills within microworlds. *Journal of Educational Data Mining*, *4*, 153–185.

Good, I. (1952). Rational decisions. *Journal of the Royal Statistical Society (Series B)*, *14*, 104-114.

Good, I. (1971). 46656 varieties of Bayesian. *American Statistician*, *25*, 62–63.

Good, I. (1976). The Bayesian influence, or how to sweep subjectivism under the carpet. In C. A. Hooker W. Harper (Eds.), *Foundations of probability theory, statistical inference, and statistical theories of science* (Vol. 2, pp. 125–174). Dordrecht: D. Reidel Publishing.

Good, I. (1983). *Good thinking.* Minneapolis: University of Minnesota Press.

Good, I. (1985). Weight of evidence: A brief survey. In J. Bernardo, M. DeGroot, D. Lindley, A. Smith (Eds.), *Bayesian statistics 2* (pp. 249–269). Amsterdam: North-Holland.

Good, I., Card, W. (1971). The diagnostic process with special reference to errors. *Methods of Information in Medicine*, *10*, 176–188.

Goodman, L. A., Kruskal, W. H. (1954). Measures of association for cross classifications. *Journal of the American Statistical Association*, *49*(268), 732–764. Retrieved from http://www.jstor.org/stable/2281536.

Graesser, A. C., VanLehn, K., Rose, C. P., Jordan, P. W., Harter, D. (2001). Intelligent tutoring systems with conversational dialogue. *AI Magazine*, *22*(4), 39–52.

Graf, E. A. (2003, September). *Designing a proficiency model and associated item models for a mathematics unit on sequences.* Paper presented at the Cross Division Math Forum, Princeton, NJ.

Graf, E. A. (2008). *Approaches to the design of diagnostic item models* (Research Report No. RR-08-07). Educational Testing Service. Retrieved from http://www.ets.org/research/researcher/RR-08-07.html.

Guttman, I. (1967). The use of the concept of a future observation in goodness-of-fit problems. *Journal of the Royal Statistical Society, Series B*, *29*, 83–100.

Haberman, S. J. (1972). Log-linear fit for contingency tables—Algorithm AS51. *Applied Statistics*, *21*, 218–225.

Haberman, S. J. (2005a). *Latent-class item response models* (Research Report No. RR-05-28). Princeton: Educational Testing Service.

Haberman, S. J. (2005b). *When can subscores have value?* (Research Report No. RR-05-08). ETS. Retrieved from http://www.ets.org/research/researcher/RR-05-08.html.

Haberman, S. J. (2009). *Use of generalized residuals to examine goodness of fit of item response models* (Research Report No. RR-09-15). Princeton: Educational Testing Service.

Haberman, S. J., Sinharay, S., Chon, K. H. (2013). Assessing item fit for unidimensional item response theory models using residuals from estimated item response functions. *Psychometrika*, *78*(3), 417–440.

Haertel, E. H. (1984). An application of latent class models to assessment data. *Applied Psychological Measurement*, *8*, 333–346.

Haertel, E. H. (1989). Using restricted latent class models to map the skill structure of achievement test items. *Journal of Educational Measurement*, *26*, 301–321.

Hambleton, R. K., Pitoniak, M. J. (2006). Educational measurement. In R. L. Brennan (Ed.), *Educational measurement* (4th ed., pp. 433–470). Westport: American Council on Education/Praeger.

Hambleton, R. K., Swaminathan, H., Rogers, H. J. (1991). *Fundamentals of item response theory.* Newbury Park: Sage.

Hansen, E. G., Mislevy, R. J. (2004, April). *Toward a unified validity framework for ensuring access to assessments by individuals with disabilities and English language learners.* Paper presented at the annual meeting of the National Council on Measurement in Education. Retrieved from http://www.ets.org/research/dload/NCME2004-Hansen.pdf.

Hansen, E. G., Mislevy, R. J. (2005). Accessibility of computer-based testing for individuals with disabilities and English language learners within a validity framework. In M. Hricko (Ed.), *Online assessment and measurement: Foundation, challenges, and issues* (pp. 212–259). Hershey: Idea Group.

Hansen, E. G., Mislevy, R. J., Steinberg, L. S. (2003). Evidence-centered assessment design and individuals with disabilities. Paper presented at annual meeting of the National Council on Measurement in Education. Retrieved from http://www.ets.org/research/dload/ncme03-hansen.pdf.

Hartz, S. M. (2002). *A Bayesian framework for the unified model for assessing cognitive abilities: Blending theory with practice.* Unpublished doctoral dissertation, University of Illinois at Urbana-Champaign.

Hastings, W. (1970). Monte Carlo sampling methods using Markov chains and their applications. *Biometrika*, *57*, 97–109.

Heckerman, D. (1991). *Probabilistic similarity networks.* New York: ACM Press.

Heckerman, D. (1998). A tutorial on learning with Bayesian networks. In M. I. Jordan (Ed.), *Learning in graphical models* (pp. 301–354). Amsterdam: Kluwer Academic.

Heckerman, D., Gieger, D., Chickering, D. (1995). Learning Bayesian networks: The combination of knowledge and statistical data. *Machine Learning*, *20*, 197–243.

Heckerman, D., Horvitz, E., Middleton, B. (1993). An approximate nonmyopic computation for value of information. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *15*, 292–298.

Heidelberger, P., Welch, P. D. (1981). A spectral method for confidence interval generation and run length control in simulations. *Communications of the ACM*, *24*, 233–245.

Henrion, M., Druzdzel, M. J. (1990). Qualitative propagation and scenario-based approaches to explanation of probabilistic reasoning. In *Uncertainty in artificial intelligence: Proceedings of the 6th conference* (pp. 10–20). Mountain View: Association for Uncertainty in AI.

Henson, R. A., Douglas, J. A. (2005). Test construction for cognitive diagnosis. *Applied Psychological Measurement*, *29*(4), 262–277.

Henson, R. A., Templin, J. L., Willse, J. T. (2009). Defining a family of cognitive diagnosis models using log-linear models with latent variables. *Psychometrika*, *74*, 191–210.

Henze, N., Nejdl, W. (1999). Student modeling in an active learning environment using Bayesian networks. In *Proceedings of the seventh international conference on user modeling, UM99.*

Hilden, J. (1970). GENEXX—An algebraic approach to pedigree probability calculus. *Clinical Genetics*, *1*, 319–348.

Hively, W., Patterson, H. L., Page, S. H. (1968). A 'universe-defined' system of arithmetic achievement tests. *Journal of Educational Measurement*, *5*(4), 275–290.

Hoey, J., St-Aubin, R., Hu, A., Boutilier, C. (2001). SPUDD: Stochastic planning using decision diagrams. In K. Laskey H. Prad (Eds.), *Uncertainty in artificial intelligence: Proceedings of the 15th conference* (pp. 279–288). San Francisco: Morgan Kaufmann.

Holland, P. W. (1986). Statistics and causal inference. *Journal of the American Statistical Association*, *81*, 945–960.

Holland, P. W., Thayer, D. T. (1988). Differential item performance and the Mantel-Haenszel procedure. In H. Wainer H. I. Braun (Eds.), *Test validity* (pp. 129–145). Mahwah: Lawrence Erlbaum.

Holland, P. W., Wainer, H. (1993). *Differential item functioning.* Mahwah: Lawrence Erlbaum.

Howard, R. A., Matheson, J. E. (1984). Influence diagrams. In A. Howard J. E. Matheson (Eds.), *Readings on the principles and applications of decision analysis* (Vol. 2, pp. 717–762). Menlo Park: Strategic Decisions Group.

Hrycej, T. (1990). Gibbs sampling in Bayesian networks. *Artificial Intelligence*, *46*, 351–363.

Huff, K., Steinberg, L. S., Matts, T. (2010). The promises and challenges of implementing evidence-centered design in large-scale assessment. *Applied Measurement in Education*, *23*(4), 310–324.

IMS Global Learning Consortium. (2000). IMS question & test interoperability information model specification (Version 1.0 ed.) [Computer software manual]. Retrieved from `http://www.imsproject.org.`

Irvine, S. H. (2013). *Tests for recruitment across cultures: a tactical psychometric handbook.* Amsterdam: IOS.

Irvine, S. H., Dann, P. L., Anderson, J. D. (1990). Towards a theory of algorithm-determined cognitive test construction. *British Journal of Psychology*, *81*, 173–195.

Irvine, S. H., Kyllonen, P. (Eds.). (2002). *Generating items for cognitive tests: Theory and practice.* Mahwah: Erlbaum.

Iseli, M. R., Koenig, A. D., Lee, J. J., Wainess, R. (2010). *Automatic assessment of complex task performance in games and simulations* (CSE Technical Report No. 775). Los Angeles: The National Center for Research on Evaluation, Standards, Student Testing (CRESST). Retrieved from `http://www.cse.ucla.edu/products/reports/R775.pdf`.

Jaakkola, T. S. (2001). Tutorial on variational approximation methods. In M. Opper D. Saad (Eds.), *Advanced mean field methods: Theory and practice* (pp. 129–159). Cambridge: MIT Press.

Jaynes, E. T. (1968). Prior probabilities. *IEEE Transactions on Systems Science and Cybernetics*, *SSC-4*, 227–241.

Jeffreys, H. (1961). *Theory of probability* (3rd ed.). New York: Oxford University Press.

Jensen, F. V. (1988). *Junction trees and decomposable hypergraphs* (Judex Research Report). Aalborg: Judex.

Jensen, F. V. (1996). *An introduction to Bayesian networks.* New York: Springer.

Johnson, M. S., Sinharay, S. (2003). *Calibration of polytomous item families using Bayesian hierarchical modeling* (Research Report No. RR-03-23). Princeton: Educational Testing Service.

Jordan, M. I. (Ed.). (1998). *Learning in graphical models.* Amsterdam: Kluwer Academic.

Joreskog, K. G., Sorbom, D. (1979). *Advances in factor analysis and structural equation models.* Cambridge: Abt.

Junker, B. W. (1999). *Some statistical models and computational methods that may be useful for cognitively-relevant assessment.*

Junker, B. W., Sijtsma, K. (2001). Cognitive assessment models with few assumptions, and connections with nonparametric item response theory. *Applied Psychological Measurement*, *25*, 258–272.

Kadane, J. B. (1980). Predictive and structural methods for eliciting prior distributions. In A. Zellner (Ed.), *Bayesian analysis and statistics* (pp. 89–93). Amsterdam: North-Holland.

Kadane, J. B., Dickey, J. M., Winkler, R. L., Smith, W. S., Peters, S. C. (1980). Interactive elicitation of opinion for a normal linear model. *Journal of the American Statistical Association*, *75*, 845–854.

Kahneman, D., Slovic, P., Tversky, A. (1982). *Judgment under uncertainty: Heuristics and biases.* Cambridge: Cambridge University Press.

Kane, M. T. (1992). An argument-based approach to validity. *Psychological Bulletin*, *112*, 527–535.

Kane, M. T. (2006). Validation. In R. L. Brennan (Ed.), *Educational measurement* (4th ed., pp. 17–64). Westport: American Council on Education/Praeger.

Kane, M. T. (2013). Validating the interpretations and uses of test scores. *Journal of Educational Measurement*, *50*(1), 1–73.

Kaplan, D. (2000). *Structural equation modeling: Foundations and extensions.* Thousand Oaks: Sage.

Katz, I. R., Williamson, D. M., Nadelman, H. L., Kirsch, I., Almond, R. G., Cooper, P. L., et al. (2004, June). *Assessing information and communications technology literacy for higher education.* Paper presented at the 30th annual conference of the International Association for Educational Assessment, Philadelphia, PA.

Kennedy, C. A., Wilson, M. R. (2006, October). *Using progress variables to map intellectual development.* Paper presented at MSDE/MARCES conference, College Park, MD.

Kennedy, C. A., Wilson, M. R., Draney, K., Tutunciyan, S., Vorp, R. (2006). *ConceptMap.* [Computer software]. Berkeley: Bear Center. Retrieved from `http://bearcenter.berkeley.edu/GradeMap`.

Kim, J. H., Pearl, J. (1983). A computational model for causal and diagnostic reasoning in inference systems. In *Proceedings of the 8th international joint conference on artificial intelligence* (pp. 190–193). Karlsruhe: William Kaufmann.

Klein, M. F., Birenbaum, M., Standiford, S. N., Tatsuoka, K. K. (1981). *Logical error analysis and construction of tests to diagnose student "bugs" in addition and subtraction of fractions* (Research Report No. 81-6). Computer-based Education Research Laboratory, University of Illinois.

Koedinger, K. R., Aleven, V. (2007). Exploring the assistance dilemma in experiments with cognitive tutors. *Educational Psychology Review*, *19*(3), 239–264. doi: 10.1007/s10648-007-9049-0.

Kohavi, R. (1995). A study of cross-validation and bootstrap for accuracy estimation and model selection. In C. S. Mellish (Ed.), *Proceedings of IJCAI-95*, Montreal, Canada (pp. 1137–1143). Los Altos, CA: Morgan Kaufmann.

Kolen, M. J., Brennan, R. L. (2004). *Test equating, scaling, and linking: Methods and practices.* New York: Springer.

Koller, D., Learner, U. (2001). Sampling in factored dynamic systems. In A. Doucet, N. de Freitas, N. Gordon (Eds.), *Sequential Monte Carlo methods in practice* (pp. 445–464). New York: Springer.

Koller, D., Pfeffer, A. (1997). Object-oriented Bayesian networks. *Uncertainty in Artificial Intelligence: Proceedings of the 13th Conference* (pp. 302–313). Retrieved from `http://citeseer.nj.nec.com/koller97objectoriented.html`.

Lange, K. (1995). A quasi-Newton acceleration of the EM algorithm. *Statistica Sinica*, *5*(1), 1–18.

Laskey, K. B., Mahoney, S. M. (2000). Network engineering for agile belief network models. *IEEE Transactions on Knowledge and Data Engineering*, *12*, 481–486.

Lauritzen, S. L. (1992). Propagation of probabilities, means, and variances in mixed graphical association models. *Journal of the American Statistical Association*, *87*, 1098–1108.

Lauritzen, S. L. (1996). *Graphical models.* Oxford: Oxford University Press.

Lauritzen, S. L., Spiegelhalter, D. J. (1988). Local computation with probabilities on graphical structures and their application to expert systems (with discussion). *Journal of the Royal Statistical Society, Series B*, *50*, 205–247. (Reprinted in Shafer and Pearl (1990)).

Lee, P. M. (1989). *Bayesian statistics: An introduction.* Oxford: Oxford University Press.

Leighton, J. P., Gierl, M. J. (Eds.). (2007). *Cognitive diagnostic assessment: Theories and applications.* Cambridge: Cambridge University Press.

Leighton, J. P., Gierl, M. J., Hunka, S. M. (2004). The attribute hierarchy model: An approach for integrating cognitive theory with assessment practice. *Journal of Educational Measurement*, *41*, 205–236.

Levy, R. (2011). Posterior predictive model checking for conjunctive multidimensionality in item response theory. *Journal of Educational and Behavioral Statistics*, *36*, 672–694.

Levy, R. (2014). *Dynamic Bayesian network modeling of game based diagnostic assessments* (CSE Technical Report 837). Los Angeles: The National Center for Research on Evaluation, Standards, Student Testing (CRESST).

Levy, R., Mislevy, R. J., Sinharay, S. (2009). Posterior predictive model checking for multidimensionality in item response theory. *Applied Psychological Measurement*, *33*, 519–537.

Ley, T., Kump, B., Albert, D. (2010). A methodology for eliciting, modelling, and evaluating expert knowledge for an adaptive work-integrated learning system. *International Journal of Human-Computer Studies*, *68*(4), 185–208. doi: 10.1016/j.ijhcs.2009.12.001.

Li, Z., D'Ambrosio, B. (1994). Efficient inference in Bayes nets as a combinatorial optimization problem. *Intl Journal of Approximate Reasoning*, *11*, 55–81.

Little, R., Rubin, D. B. (1987). *Statistical analysis with missing data.* New York: Wiley.

Liu, J., Xu, G., Ying, Z. (2012). Data-driven learning of Q-matrix. *Applied Psychological Measurement*, *36*, 548–564.

Liu, J. S. (2001). *Monte Carlo strategies in scientific computing.* New York: Springer.

Lord, F. M. (1980). *Applications of item response theory to practical testing problems.* Mahwah: Lawrence Erlbaum.

Louis, T. A. (1982). Finding the observed information matrix when using the EM algorithm. *Journal of the Royal Statistical Society, Series B*, *44*, 226–233.

Luecht, R. M. (2012). An introduction to assessment engineering for automatic item generation. In M. J. Gierl T. M. Haladyna (Eds.), *Automatic item generation: Theory and practice* (pp. 59–76). New York: Routledge.

Lunn, D. J., Spiegelhalter, D. J., Thomas, A., Best, N. G. (2009). The BUGS project: Evolution, critique and future directions (with discussion). *Statistics in Medicine*, *28*, 3049–3082.

Lunn, D. J., Thomas, A., Best, N. G., Spiegelhalter, D. J. (2000). WinBUGS – a Bayesian modeling framework: Concepts, structure, and extensibility. *Statistics and Computing*, *10*, 325–337.

Lynch, S. M. (2007). *Introduction to applied Bayesian statistics and estimation for social sciences.* New York: Springer.

Madigan, D., Almond, R. G. (1995). Test selection strategies for belief networks. In D. Fisher H. J. Lenz (Eds.), *Learning from data: AI and Statistics V* (pp. 89–98). New York: Springer.

Madigan, D., Gavrin, J., Raftery, A. E. (1995). Enhancing the predictive performance of Bayesian graphical models. *Communications in Statistics: Theory and Methods*, *24*, 2271–2292.

Madigan, D., Hunt, E., Levidow, B. (1995). *Bayesian graphical modeling for intelligent tutoring systems* (Tech. Rep.). Seattle: University of Washington, Department of Statistics.

Madigan, D., Mosurski, K., Almond, R. G. (1997). Graphical explanation in belief networks. *Journal of Computational Graphics and Statistics*, *6*(2), 160–181. Retrieved from `http://www.amstat.org/publications/jcgs/index.cfm?fuseaction=madiganjun`.

Madigan, D., Raftery, A. E., Volinsky, C., Hoeting, J. (1996). Bayesian model averaging. In *Proceedings of the AAAI Workshop on Integrating Multiple Learned Models.*

Mahoney, S. M., Laskey, K. B. (1996). Network engineering for complex belief networks. In E. Horvitz F. Jensen (Eds.), *Uncertainty in artificial intelligence: Proceedings of the 12th conference* (pp. 389–396). San Francisco: Morgan Kaufmann.

Maris, E. (1999). Estimating multiple classification latent class models. *Psychometrika*, *64*(2), 187–212.

Martin, J., VanLehn, K. (1994). *Discrete factor analysis: Learning hidden variables in Bayesian networks* (Technical Report No. LRDC-ONR-94-1). Pittsburgh: LRDC, University of Pittsburgh.

Martin, J., VanLehn, K. (1995). A Bayesian approach to cognitive assessment. In P. D. Nichols, S. F. Chipman, R. L. Brennan (Eds.), *Cognitively diagnostic assessment* (pp. 141–165). Mahwah: Lawrence Erlbaum.

Matheson, J. E. (1990). Using influence diagrams to value information and control. In R. M. Oliver, J. Q. Smith (Eds.), *Influence diagrams, belief nets and decision analysis* (pp. 25–48). New York: Wiley.

Matsuda, N., VanLehn, K. (2000). Decision theoretic instructional planner for intelligent tutoring systems. In B. du Boulay (Ed.), *Proceedings for workshop on modeling human teaching tactics and strategies (ITS 2000)* (pp. 72–83).

Mayo, M., Mitrovic, A. (2001). Optimising ITS behaviour with Bayesian networks and decision theory. *International Journal of Artificial Intelligence in Education*, *12*(2), 124–153.

McLachlan, G., Krishnan, T. (2008). *The EM algorithm and extensions* (2nd ed.). New York: Wiley.

Meijer, R. R., Sijtsma, K. (2001). Methodology review: Evaluating person fit. *Applied Psychological Measurement*, *25*(2), 107–135.

Melnick, D. (1996). The experience of the National Board of Medical Examiners. In E. Mancall, P. Vashook, J. Dockery (Eds.), *Computer-based examinations for board certification* (pp. 111–120). Chicago: American Board of Medical Specialties.

Messick, S. (1989). Validity. In R. L. Linn (Ed.), *Educational measurement* (3rd ed., pp. 13–103). New York: American Council on Education/Macmillan.

Messick, S. (1994). The interplay of evidence and consequences in the validation of performance assessments. *Educational Researcher*, *23(2)*, 13–23.

Metropolis, N., Ulam, S. (1949). The Monte Carlo method. *Journal of the American Statistical Association*, *44*, 335–341.

Millán, E., Pérez-de-la-Cruz, J. L. (2002). A Bayesian diagnostic algorithm for student modeling and its evaluation. *User Modeling and User-Adapted Interaction*, *12*(2–3), 281–330.

Miller, P. (1983). Attending: Critiquing a physician's management plan. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *5*, 449–461.

Mislevy, J., Rupp, A. A., Harring, J. R. (2012). Detecting local item dependence in polytomous adaptive data. *Journal of Educational Measurement*, *49*, 127–147.

Mislevy, R. J. (1984). Estimating latent distributions. *Psychometrika*, *49*, 359–381.

Mislevy, R. J. (1986). Bayes modal estimation in item response models. *Psychometrika*, *51*, 177–195.

Mislevy, R. J. (1994). Evidence and inference in educational assessment. *Psychometrika*, *59*, 439–483.

Mislevy, R. J. (1995a). *Information-decay pursuit of dynamic parameters in student models* (Research Report No. RM-94-14-onr). Princeton: Educational Testing Service. Retrieved from `http://www.ets.org/research/policy_research_reports/rm-94-14-onr`.

Mislevy, R. J. (1995b). Probability-based inference in cognitive diagnosis. In P. D. Nichols, S. F. Chipman, R. L. Brennan (Eds.), *Cognitively diagnostic assessment* (p. 43-71). Mahwah: Lawrence Erlbaum.

Mislevy, R. J. (1995c). Test theory and language learning in assessment. *Language Testing*, *12*, 341–369.

Mislevy, R. J. (2006). Cognitive psychology and educational assessment. In R. L. Brennan (Ed.), *Educational measurement* (4th ed., pp. 257–305). American Council on Education/Praeger.

Mislevy, R. J. (2009). Validity from the perspective of model-based reasoning. In R. W. Lissitz (Ed.), *The concept of validity: Revisions, new directions, and applications* (pp. 83–108). Charlotte: Information Age.

Mislevy, R. J. (2010). Some implications of expertise research for educational assessment. *Research Papers in Education*, *25*(3), 253–270.

Mislevy, R. J. (2015). Missing responses in item response theory. In W. J. van der Linden R. K. Hambleton (Eds.), *Handbook of item response theory* (2nd ed.). New York: Chapman & Hall.

Mislevy, R. J., Almond, R. G., DiBello, L. V., Jenkins, F., Steinberg, L. S., Yan, D., Senturk, D. (2002). *Modeling conditional probabilities in complex educational assessments* (CSE Technical Report No. 580). Los Angeles: The National Center for Research on Evaluation, Standards, Student Testing (CRESST). Retrieved from `http://www.cresst.org/reports/TR580.pdf`

Mislevy, R. J., Almond, R. G., Lukas, J. F. (2004). *A brief introduction to evidence-centered design* (CSE Technical Report No. 632). Los Angeles: The National Center for Research on Evaluation, Standards, Student Testing (CRESST). Retrieved from `http://www.cresst.org/reports/r632.pdf` (Also ETS Research Report RR-03-32.)

Mislevy, R. J., Almond, R. G., Steinberg, L. S. (1998). *A note on knowledge-based model construction in educational assessment* (CSE Technical Report No. 480). Los Angeles: The National Center for Research on Evaluation, Standards, Student Testing (CRESST). Retrieved from `http://www.cresst.org/reports/TECH480.pdf`

Mislevy, R. J., Almond, R. G., Steinberg, L. S. (2002). Design and analysis in a task-based language assessment. *Language Testing*, *19*(4), 477–496.

Mislevy, R. J., Almond, R. G., Yan, D., Steinberg, L. S. (1999). Bayes nets in educational assessment: Where the numbers come from. In K. B. Laskey H. Prade (Eds.), *Uncertainty in artificial intelligence: Proceedings of the 15th conference* (pp. 437–446). San Francisco: Morgan Kaufmann.

Mislevy, R. J., Almond, R. G., Yan, D., Steinberg, L. S. (2000). *Bayes nets in educational assessment: Where do the numbers come from?* (CSE Technical Report No. 518). Los Angeles: The National Center for Research on Evaluation, Standards, Student Testing (CRESST). Retrieved from `http://www.cse.ucla.edu/products/reports/TECH518.pdf`.

Mislevy, R. J., Behrens, J. T., Bennett, R. E., DeMark, S. F., Frezzo, D. C., Levy, R., et al. (2010). On the roles of external knowledge representations in assessment design. *The Journal of Technology, Learning and Assessment*, *8*(2). Retrieved from `http://napoleon.bc.edu/ojs/index.php/jtla/article/viewFile/1621/1465`.

Mislevy, R. J., Behrens, J. T., DiCerbo, K. E., Levy, R. (2012). Design and discovery in educational assessment: Evidence-centered design, psychometrics, and educational data mining. *JEDM-Journal of Educational Data Mining*, *4*(1), 11–48.

Mislevy, R. J., Gitomer, D. H. (1996). The role of probability based inference in an intelligent tutoring system. *User-Modeling and User-Adapted Interaction*, *5*, 253–282.

Mislevy, R. J., Haertel, G. D. (2006). Implications of evidence-centered design for educational testing. *Educational Measurement: Issues and Practice*, *25*(4), 6–20.

Mislevy, R. J., Haertel, G. D., Cheng, B. H., Ructtinger, L., DeBarger, A., Murray, E., et al. (2013). A 'conditional' sense of fairness in assessment. *Educational Research and Evaluation*, *19*(2–3), 121–140.

Mislevy, R. J., Oranje, A., Bauer, M. I., von Davier, A., Hao, J., Corrigan, S., et al. (2014). *Psychometric considerations in game-based assessment.* New York: Institute of Play.

Mislevy, R. J., Riconscente, M. M. (2006). Evidence-centered assessment design. In S. Downing & T. Haladyna (Eds.), *Handbook of Test Development* (pp. 61–90). Mahwah, NJ: Erlbaum.

Mislevy, R. J., Riconscente, M. M., Rutstein, D. W. (2009). *Design patterns for assessing model-based reasoning* (Large-Scale Assessment Technical Report No. 6). Menlo Park: SRI International. Retrieved from `http://ecd.sri.com/downloads/ECD_TR6_Model-Based_Reasoning.pdf`.

Mislevy, R. J., Sheehan, K. M., Wingersky, M. S. (1993). How to equate tests with little or no data. *Journal of Educational Measurement*, *30*, 55–78.

Mislevy, R. J., Steinberg, L. S., Almond, R. G. (2002). On the roles of task model variables in assessment design. In S. H. Irvine P. Kyllonen (Eds.), *Generating items for cognitive tests: Theory and practice* (p. 97–128). Mahwah: Lawrence Erlbaum.

Mislevy, R. J., Steinberg, L. S., Almond, R. G. (2003a). *On the structure of educational assessments* (CSE Technical Report No. 597). Los Angeles: The National Center for Research on Evaluation, Standards, Student Testing (CRESST). Retrieved from `http://www.cse.ucla.edu/products/reports/TR597.pdf`.

Mislevy, R. J., Steinberg, L. S., Almond, R. G. (2003b). On the structure of educational assessment (with discussion). *Measurement: Interdisciplinary Research and Perspective*, *1*(1), 3–62.

Mislevy, R. J., Steinberg, L. S., Almond, R. G., Haertel, G. D., Penuel, W. (2003). Leverage points for improving educational assessment. In B. Means G. D. Haertel (Eds.), *Evaluating the effects of technology in education* (pp. 149–180). Mahwah: Lawrence Erlbaum.

Mislevy, R. J., Steinberg, L. S., Almond, R. G., Lukas, J. F. (2006). Concepts, terminology and basic models of evidence-centered design. In D. M. Williamson, R. J. Mislevy, I. I. Bejar (Eds.), *Automated scoring of complex tasks in computer-based testing* (pp. 15–47). Hillsdale: Lawrence Erlbaum.

Mislevy, R. J., Steinberg, L. S., Breyer, F. J., Almond, R. G., Johnson, L. (1999). A cognitive task analysis, with implications for designing a simulation-based assessment system. *Computers and Human Behavior*, *15*, 29–42.

Mislevy, R. J., Steinberg, L. S., Breyer, F. J., Almond, R. G., Johnson, L. (2002). Making sense of data from complex assessments. *Applied Measurement in Education*, *15*(4), 363–389.

Mislevy, R. J., Wingersky, M. S., Sheehan, K. M. (1994). *Dealing with uncertainty about item parameters: Expected response functions* (Research Report No. RR-94-28-ONR). Princeton: Educational Testing Service. Retrieved from `http://www.ets.org/research/researcher/RR-94-28-ONR.html`.

Mislevy, R. J., Wu, P.-K. (1996). *Missing responses and Bayesian IRT ability estimation: Omits, choice, time limits, and adaptive testing* (Research Report No. RR-96-30-ONR). Princeton: Educational Testing Service.

Morgan, M. G., Henrion, M. (1990). *Uncertainty: A guide to dealing with uncertainty in quantitative risk and policy analysis.* Cambridge: Cambridge University Press.

Moss, P. A., Girard, B. J., Haniford, L. C. (2006). Validity in educational assessment. *Review of Research in Education*, *30*, 109–162.

Moss, P. A., Pullin, D. C., Gee, J. P., Haertel, E. H., Young, L. J. (2008). *Assessment, equity, and opportunity to learn.* Cambridge: Cambridge University Press.

Moussouris, J. (1974). Gibbs and Markov random systems with constraints. *Journal of Statistical Physics*, *10*, 11–33.

Mulder, J., van der Linden, W. J. (2010). Multidimensional adaptive testing with Kullback–Leibler information item selection. In *Elements of adaptive testing* (pp. 77–101). New York: Springer.

Muraki, E. (1992). A generalized partial credit model: Application of an EM algorithm. *Applied Psychological Measurement*, *16*(2), 159–176. doi: 10.1177/014662169201600206.

Murphy, K. P., Russell, S. (2001). Rao-Blackwellised particle filtering for dynamic Bayesian networks. In A. Doucet, N. de Freitas, N. Gordon (Eds.), *Sequential Monte Carlo methods in practice* (pp. 499–515). New York: Springer.

Murphy, K. P., Weiss, Y., Jordan, M. I. (1999). Loopy belief propagation for approximate inference: An empirical study. In K. B. Laskey H. Prade (Eds.), *Uncertainty in Artificial Intelligence: Proceedings of the 15th conference* (pp. 467–475). San Mateo: Morgan Kaufmann.

Murray, R., VanLehn, K., Mostow, J. (2004). Looking ahead to select tutorial actions: A decision-theoretic approach. *International Journal of Artificial Intelligence in Education*, *14*(3, 4), 235–278.

National Research Council (Ed.). (1996). *National science education standards.* Washington, DC: National Academies Press.

Neal, R. M. (2003). Slice sampling (with discussion). *Annals of Statistics*, *31*, 705–767.

Neapolitan, R. E. (1990). *Probabilistic reasoning in expert systems: Theory and algorithms.* New York: Wiley.

Neapolitan, R. E. (2004). *Learning Bayesian networks.* Englewood Cliffs: Prentice Hall.

Newell, A., Simon, H. A. (1972). *Human problem solving.* Englewood Cliffs: Prentice Hall.

Neyman, J., Scott, E. L. (1948). Consistent estimators based on partially consistent observations. *Econometrika*, *16*, 1–32.

NGSS Lead States. (2013). *Next generation science standards: For states, by states.* Washington, DC: National Academies Press.

Nichols, P. D., Chipman, S. F., Brennan, R. L. (Eds.). (1995). *Cognitively diagnostic assessment.* Mahwah: Lawrence Erlbaum.

Nicholson, A. E., Jitnah, N. (1998). Using mutual information to determine relevance in Bayesian networks. In H-Y Lee, H. Motoda (Eds.), *Pacific Rim International Conference on Artificial Intelligence* (pp. 399–410). Berlin: Springer.

Nikovski, D., Brand, M. (2003). Model minimization of dynamic belief networks for group elevator control. In *Uncertainty in artificial intelligence: Proceedings of the 19th conference, 1st Bayesian modeling application workshop.* (Vol. 3, pp. 9–13).

Nocedal, J., Wright, S. J. (2006). *Numerical optimization* (2nd ed.). New York: Springer.

Norsys, Inc. (2004). Netica [Computer software manual]. Retrieved from `http://www.norsys.com`.

O'Hagan, A. (1976). On posterior joint and marginal modes. *Biometrika*, *63*, 329–333.

O'Hagan, A., Buck, C. E., Daneshkhah, A., Eiser, J. R., Garthwaite, P. H., Jenkinson, D. J., et al. (2006). *Uncertain judgements: Eliciting experts' probabilities.* New York: Wiley.

Oliver, R. M., Smith, J. Q. (1990). *Influence diagrams, belief nets and decision analysis.* New York: Wiley.

Osburn, H. (1968). Item sampling for achievement testing. *Educational and Psychological Measurement*, *28*, 95–104.

Patz, R. J., Junker, B. W. (1999a). Applications and extensions of MCMC in IRT: Multiple item types, missing data, and rated responses. *Journal of Educational and Behavioral Statistics*, *24*(4), 342–366.

Patz, R. J., Junker, B. W. (1999b). A straight forward approach to Markov chain Monte Carlo methods for item response models. *Journal of Educational and Behavioral Statistics*, *24*, 146–178.

Pearl, J. (1988). *Probabilistic reasoning in intelligent systems: Networks of plausible inference.* San Mateo: Morgan Kaufmann.

Pearl, J. (1998). Graphs, causality, and structural equation models. *Sociological Methods and Research*, *27*(2), 226–284.

Pearl, J. (2009). *Causality: Models, reasoning and inference* (2nd ed.). Cambridge: Cambridge University Press.

Pelligrino, J., Glaser, R., Chudowsky, N. (Eds.). (2001). *Knowing what students know: The science and design of educational assessment.* Washington, DC: National Research Council.

Plummer, M. (2008). Penalized loss functions for Bayesian model comparison. *Biostatistics*, *9*(3), 523–539. doi: 10.1093/biostatistics/kxm049.

Plummer, M. (2012, May). JAGS version 3.2.0 user manual (3.2.0 ed.) [Computer software manual]. Retrieved from `http://mcmc-jags.sourceforge.net/`.

Plummer, M., Best, N. G., Cowles, M. K., Vines, K. (2006). coda: Output analysis and diagnostics for MCMC [Computer software manual]. Retrieved from `http://cran.r-project.org/web/packages/coda/`.

R Development Core Team. (2007). R: A language and environment for statistical computing [Computer software manual]. Vienna, Austria. Retrieved from `http://www.R-project.org`.

Rasch, G. (1960). *Probabilistic models for some intelligence and attainment tests.* Chicago: The University of Chicago Press.

Reckase, M. D. (2009). *Multidimensional item response theory.* New York: Springer.

Restle, F., Greeno, J. G. (1970). *Introduction to mathematical psychology.* Reading: Addison-Wesley.

Reye, J. (2004). Student modelling based on belief networks. *International Journal of Artificial Intelligence in Education*, *14*, 63–96.

Rijmen, F. (2008). Bayesian networks with a logistic regression model for the conditional probabilities. *International Journal of Approximate Reasoning*, *48*, 659–666.

Rijmen, F., De Boeck, P., Leuven, K. (2002). The random weights linear logistic test model. *Applied Psychological Measurement*, *26*(3), 271–285.

Ritter, S., Anderson, J. R., Koedinger, K. R., Corbett, A. T. (2007). Cognitive tutor: Applied research in mathematics education. *Psychonomic Bulletin and Review*, *14*(2), 249–255.

Robins, J. M., van der Vaart, A., Ventura, V. (2000). The asymptotic distribution of p-values in composite null models (with discussion). *Journal of the American Statistical Association*, *95*(422), 1143–1172.

Ross, S. M. (1988). *A first course in probability.* New York: Macmillan.

Rost, J. (1990). Rasch models in latent classes: An integration of two approaches to item analysis. *Applied Psychological Measurement*, *14*, 271–282.

Roussos, L. A., DiBello, L. V., Stout, W. F., Hartz, S. M., Henson, R. A., Templin, J. L. (2007a). The fusion model skills diagnosis system. In J. P. Leighton M. J. Gierl (Eds.), *Cognitive diagnostic assessment: Theories and applications* (pp. 281–292). Cambridge: Cambridge University Press.

Roussos, L. A., Templin, J. L., Henson, R. A. (2007b). Skills diagnosis using IRT-based latent class models. *Journal of Educational Measurement*, *44*(4), 293–311.

Rubin, D. B. (1976). Inference and missing data. *Biometrika*, *63*(3), 581–592.

Rubin, D. B. (1977). Formalizing subjective notions about the effect of nonrespondents in sample surveys. *Journal of the American Statistical Association*, *72*, 538–543.

Rubin, D. B. (1984). Bayesian justifiable and relevant frequency calculations for the applied statistician. *Annals of Statistics*, *12*, 1151–1172.

Rubin, D. B. (1987). *Multiple imputation for nonresponse in surveys.* New York: Wiley.

Rupp, A. A., Templin, J. L., Henson, R. A. (2010). *Diagnostic measurement: Theory, methods, and applications.* New York: Guilford.

Russell, M. K. (2011). Accessible test design. In M. K. Russell M. Kavanaugh (Eds.), *Assessing students in the margin: Challenges, strategies, and techniques* (pp. 407–423). Charlotte: Information Age.

Sabourin, J., Mott, B., Lester, J. (2013). Utilizing dynamic Bayes nets to improve early prediction models of self-regulated learning. In S. Carberry, S. Weibelzahl, A. Micarelli, G. Semeraro (Eds.), *User modeling, adaptation, and personalization* (pp. 228–241). New York: Springer.

Samejima, F. (1969). Estimation of latent ability using a response pattern of graded scores. *Psychometrika Monograph No. 17*, *34*(4), (Part 2).

Sao Pedro, M. A., Baker, R. S. J. d., Gobert, J. D., Montalvo, O., Nakama, A. (2013). Leveraging machine-learned detectors of systematic inquiry behavior to estimate and predict transfer of inquiry skill. *User Modeling and User-Adapted Interaction*, *23*(1), 1–39.

Savage, L. J. (1971). Elicitation of personal probabilities and expectations. *Journal of the American Statistical Association*, *66*(336), 783–801.

Savage, L. J. (1972). *The foundations of statistics* (2nd ed.). New York: Dover.

Scalise, K., Gifford, B. (2006). Computer-based assessment in e-learning: A framework for constructing “intermediate constraint” questions and tasks for technology platforms. *The Journal of Technology, Learning and Assessment*, *4*(6).

Schum, D. A. (1994). *The evidential foundations of probabilistic reasoning.* New York: Wiley.

Schwarz, G. (1978). Estimating the dimension of a model. *Annals of Statistics*, *6*, 461–464.

Shachter, R. D. (1986). Evaluating influence diagrams. *Operations Research*, *34*, 871–82.

Shafer, G. (1976). *A mathematical theory of evidence.* Princeton: Princeton University Press.

Shafer, G. (1996). *The art of causal conjecture.* Cambridge: MIT Press.

Shaftel, J., Yang, X., Glasnapp, D., Poggio, J. (2005). Improving assessment validity for students with disabilities in large-scale assessment programs. *Educational Assessment*, *10*(4), 357–375.

Shenoy, P. P. (1991). A fusion algorithm for solving Bayesian decision problems. In B. D'Ambrosio, P. Smets, P. Bonissone (Eds.), *Uncertainty in artificial intelligence: Proceedings of the 7th conference* (pp. 361–369). San Mateo, CA.

Shenoy, P. P., Shafer, G. (1990). Axioms for probability and belief-function propagation. In R. D. Shachter, T. Levitt, L. N. Lemmer J. F.and Kanal (Eds.), *Uncertainty in Artificial Intelligence: Proceedings of the 4th Conference* (pp. 169–198). Amsterdam: North-Holland.

Shortliffe, E., Buchanan, B. (1975). A model of inexact reasoning in medicine. *Mathematical Biosciences*, *23*, 351–379.

Shute, V. J. (2003, April). *Under the hood of adaptive e-learning: Diagnostic assessment, student modeling, and selection rules.* Paper presented at annual meeting of the American Educational Research Association, Chicago, IL.

Shute, V. J. (2004). Towards automating ECD-based diagnostic assessments. *Technology, Instruction, Cognition, and Learning*, *2*(1–2), 1–18.

Shute, V. J. (2006, April). *Assessments for learning: Great idea, but do they work?* Paper presented at the annual meeting of the American Educational Research Association, San Francisco, CA .

Shute, V. J., Graf, E. A., Hansen, E. G. (2005). Designing adaptive, diagnostic math assessments for individuals with and without visual disabilities. In L. M. Pytlikzillig, R. H. Bruning, M. Bodvarsson (Eds.), *Technology-based education: Bringing researchers and practitioners together* (pp. 169–202). Charlotte: Information Age.

Shute, V. J., Hansen, E. G., Almond, R. G. (2007). *An assessment for learning system called ACED: The impact of feedback and adaptivity on learning.* (Research Report No. RR-07-26). Princeton: Educational Testing Service. Retrieved from `http://www.ets.org/research/researcher/RR-07-26.html`.

Shute, V. J., Hansen, E. G., Almond, R. G. (2008). You can't fatten a hog by weighing it - or can you? Evaluating an assessment for learning system called ACED. *International Journal of Artificial Intelligence in Education*, *18*(4), 289–316. Retrieved from `http://www.ijaied.org/iaied/ijaied/abstract/Vol_18/Shute08.html`.

Shute, V. J., Torres, R. (2012). Where streams converge: Using evidence-centered design to assess quest to learn. In M. C. Mayrath, J. Clarke-Midura, D. Robinson (Eds.), *Technology-based assessments for 21st century skills: Theoretical and practical implications from modern research* (pp. 91–124). Charlotte: Information Age.

Shute, V. J., Ventura, M., Bauer, M. I., Zapata-Rivera, J.-D. (2009). Melding the power of serious games and embedded assessment to monitor and foster learning: Flow and grow. In U. Ritterfeld, M. J. Cody, P. Vorderer (Eds.), *Serious games: Mechanisms and effects* (pp. 295–321). New York: Routledge.

Shute, V. J., Ventura, M., Kim, Y. J. (2013). Assessment and learning of informal physics in Newton's playground. *Journal of Educational Research*, *106*(6), 423–430. doi: 10.1080/00220671.2013.832970.

Singley, M. K., Fairweather, P. G., Swerling, S. (1999). Team tutoring systems: Reifying roles in problem solving. In C. M. Hoadley J. Roschelle (Eds.), *Proceedings of the 1999 conference on computer support for collaborative learning* (pp. 538–548). Mahwah: Lawrence Erlbaum.

Sinharay, S. (2003). *Assessing convergence of the Markov chain Monte Carlo algorithms: A review* (Research Report No. RR-03-07). Princeton: Educational Testing Service.

Sinharay, S. (2005). Assessing fit of unidimensional item response theory models using a Bayesian approach. *Journal of Educational Measurement*, *42*(4), 375–394.

Sinharay, S. (2006). Model diagnostics for Bayesian networks. *Journal of Educational and Behavioral Statistics*, *31*(1), 1–33.

Sinharay, S., Almond, R. G. (2007). Assessing fit of cognitively diagnostic models—A case study. *Educational and Psychological Measurement*, *67*(2), 239–257.

Sinharay, S., Almond, R. G., Yan, D. (2004). *Assessing fit of models with discrete proficiency variables in educational assessment* (Research Report No. RR-04-07). Princeton: Educational Testing Service. Retrieved from `http://www.ets.org/research/researcher/RR-04-07.html`.

Smith, J. K. (2003). Reconsidering reliability in classroom assessment and grading. *Educational Measurement: Issues and Practice*, *22*(4), 26–33.

Spiegelhalter, D. J., Best, N. G., Carlin, B. P., van der Linde, A. (2002). Bayesian measures of model complexity and fit (with discussion). *Journal of the Royal Statistical Society (Series B)*, *64*, 583–639.

Spiegelhalter, D. J., Dawid, A. P., Lauritzen, S. L., Cowell, R. G. (1993). Bayesian analysis in expert systems. *Statistical Science*, *8*, 219–283.

Spiegelhalter, D. J., Knill-Jones, R. (1984). Statistical and knowledge-based approaches to clinical decision support systems, with an application in gastroenterology. *Journal of the Royal Statistical Society (Series A)*, *147*, 35–77.

Spiegelhalter, D. J., Lauritzen, S. L. (1990). Sequential updating of conditional probabilities on directed graphical structures. *Networks*, *20*, 579–605.

Spiegelhalter, D. J., Thomas, A., Best, N. G., Gilks, W. R. (1995). BUGS: Bayesian inference using Gibbs sampling, version 0.50 [Computer software manual]. Cambridge: MRC Biostatistics Unit. Retrieved from `http://www.mrc-bsu.cam.ac.uk/bugs/`.

Spiegelhalter, D. J., Thomas, A., Best, N. G., Gilks, W. R. (n.d.). Bugs 0.5 examples volume 1 (version i) [Computer software manual]. Retrieved from `http://www.mrc-bsu.cam.ac.uk/bugs/documentation/contents.shtml`.

Spirtes, P., Meek, C., Richardson, T. S. (1997). A polynomial-time algorithm for determining DAG equivalence in the presence of latent variables and selection bias. In D. Madigan P. Smythe (Eds.), *Preliminary papers of the sixth international workshop on AI and statistics* (pp. 489–501).

Srinivas, S. (1993). A generalization of the noisy-or model, the generalized noisy or-gate. In D. Heckerman A. Mamdani (Eds.), *Uncertainty in artificial intelligence: Proceedings of the 9th conference* (pp. 208–215). San Mateo: Morgan Kaufmann.

Steinberg, L. S., Almond, R. G., Baird, A. B., Cahallan, C., Chernick, H., DiBello, L. V., et al. (2003). *Introduction to the Biomass project: An illustration of evidence-centered assessment design and delivery capability* (CSE Report No. 609). Los Angeles: National Center for Research on Evaluation, Standards, and Student Testing (CRESST). Retrieved from `http://www.cse.ucla.edu/reports/R609.pdf`.

Steinberg, L. S., Gitomer, D. H. (1996). Intelligent tutoring and assessment built on an understanding of a technical problem-solving task. *Instructional Science*, *24*, 223–258.

Stevens, R. H., Thadani, V. (2007). Quantifying student's scientific problem solving efficiency and effectiveness. *Technology, Instruction, Cognition, and Learning*, *5*(4), 325–338.

Stewart, J., Hafner, R. (1994). Research on problem solving: Genetics. In D. Gabel (Ed.), *Handbook of research on science teaching and learning* (pp. 284–300). New York: Macmillan.

Suermondt, H. (1992). *Explanation in Bayesian belief networks.* Unpublished doctoral dissertation, Departments of Computer Science and Medicine, Stanford University.

Suppes, P. (1969). Stimulus response theory of finite automata. *Journal of Mathematical Psychology*, *6*, 327–355.

Swender, E., Conrad, D., Vicars, R. (2012). *ACTFL Proficiency Guidelines 2012.* Alexandria: American Council on the Teaching of Foreign Languages.

Takikawa, M., D'Ambrosio, B., Wright, E. (2002). Real-time inference with large-scale temporal Bayes nets. In J. Breese D. Koller (Eds.), *Uncer-*

tainty in artificial intelligence: Proceedings of the 18th conference. San Mateo: Morgan Kaufmann.

Tanimoto, S. (2001). Distributed transcripts for online learning: Design issues. *Journal of Interactive Media in Education*, *2001*(2). Retrieved from http://www-jime.open.ac.uk/2001/2/.

Tatsuoka, K. K. (1983). Rule space: An approach for dealing with misconceptions based on item response theory. *Journal of Educational Measurement*, *20*, 345–354.

Tatsuoka, K. K. (1984). *Analysis of errors in fraction addition and subtraction problems* (Vol. 20; NIE Final report No. NIE-G-81-002). Champaign: University of Illinois at Urbana-Champaign, Computer-Based Education Research.

Tatsuoka, K. K. (1990). Toward an integration of item response theory and cognitive error diagnosis. In N. Frederiksen, R. Glaser, A. Lesgold, M. G. Shafto (Eds.), *Diagnostic monitoring of skill and knowledge acquisition* (pp. 453–488). Mahwah: Lawrence Erlbaum.

Tatsuoka, K. K. (1995). Architecture of knowledge structures and cognitive diagnosis: A statistical pattern recognition approach. In P. D. Nichols, S. F. Chipman, R. L. Brennan (Eds.), *Cognitively diagnostic assessment* (pp. 327–359). Mahwah: Lawrence Erlbaum.

Tatsuoka, K. K. (2009). *Cognitive assessment: An introduction to the rule space method.* New York: CRC.

Tatsuoka, K. K., Linn, R. L., Tatsuoka, M. M., Yamamoto, K. (1988). Differential item functioning resulting from the use of different solution strategies. *Journal of Educational Measurement*, *25*(4), 301–319.

Tatsuoka, M. M., Tatsuoka, K. K. (1989). Rule space. In S. Kotz N. L. Johnson (Eds.), *Encyclopedia of statistical sciences* (pp. 217–220). New York: Wiley.

Thissen, D., Wainer, H. (2001). *Test scoring.* Mahwah: Lawrence Erlbaum.

Thomas, A., Spiegelhalter, D. J., Gilks, W. R. (1992). BUGS: A program to perform Bayesian inference using Gibbs sampling. In J. M. Bernardo, J. O. Berger, A. P. Dawid, A. F. M. Smith (Eds.), *Bayesian statistics 4* (pp. 837–842). Gloucestershire: Clarendon.

Toulmin, S. E. (1958). *The uses of argument.* Cambridge: Cambridge University Press.

van der Gaag, L. C., Bodlaender, H. L., Feelders, A. (2004). Monotonicity in Bayesian networks. In M. Chickering J. Halpern (Eds.), *Uncertainty in artificial intelligence: Proceedings of the 20th conference* (pp. 569–576). Arlington: AUAI Press.

van der Linden, W. J. (2005). *Linear models for optimal test design.* New York: Springer.

van der Linden, W. J., Glas, C. A. W. (2010). *Elements of adaptive testing.* New York: Springer.

VanLehn, K. (2008). Intelligent tutoring systems for continuous, embedded assessment. In C. Dwyer (Ed.), *The future of assessment: Shaping teaching and learning* (pp. 113–138). Mahwah: Lawrence Erlbaum.

VanLehn, K., Martin, J. (1997). Evaluation of an assessment system based on Bayesian student modeling. *International Journal of Artificial Intelligence in Education*, 8, 179–221.

Veldkamp, B. P., Verschoor, A. J., Eggen, T. J. (2010). A multiple objective test assembly approach for exposure control problems in computerized adaptive testing. *Psicológica*, *31*(2), 335–355.

Vendlinski, T. P., Baker, E. L., Niemi, D. (2008). Templates and objects in authoring problem-solving assessments. In E. L. Baker, J. Dickieson, W. Wulfeck, H. F. O'Neil (Eds.), *User modeling, adaptation, and personalization* (pp. 309–333). Mahwah: Lawrence Erlbaum.

Vomlel, J. (2003). Two applications of Bayesian networks. In *Proceedings of the conference Znalosti 2003, Ostrava, Czech Republic* (pp. 73–82).

Vomlel, J. (2004). Bayesian networks in educational testing. *International Journal of Uncertainty Fuzziness and Knowledge Based Systems*, *12*, 83–100.

von Davier, M. (2008). A general diagnostic model applied to language testing data. *British Journal of Mathematical and Statistical Psychology*, *61*, 287–307.

von Davier, M., Haberman, S. J. (2014). Hierarchical diagnostic classification models morphing into unidimensional 'diagnostic' classification models—A commentary. *Psychometrika*, *79*(2), 340–346. doi: 10.1007/s11336-013-9363-z.

Vygotsky, L. (1978). *Mind in society: The development of higher mental processes.* Cambridge: Harvard University Press.

Wainer, H., Bradlow, E. T., Wang, X. (2007). *Testlet response theory and its applications.* Cambridge: Cambridge University Press.

Wainer, H., Dorans, N. J., Flaugher, R., Green, B. F., Mislevy, R. J., Steinberg, L., Thissen, D. (2000). *Computerized adaptive testing: A primer* (2nd ed.). Mahwah: Lawrence Erlbaum.

Wainer, H., Kiely, G. L. (1987). Item clusters and computerized adaptive testing: A case for testlets. *Journal of Educational Measurement*, *24*, 185–201.

Walley, P. (1991). *Statistical reasoning with imprecise probabilities.* London: Chapman and Hall.

Wang, C., Chang, H.-H., Huebner, A. (2011). Restrictive stochastic item selection methods in cognitive diagnostic computerized adaptive testing. *Journal of Educational Measurement*, *48*(3), 255–273.

Weaver, W. (1948). Probability, rarity, interest, and surprise. *Scientific Monthly*, *67*, 390–392.

Weiss, Y. (2000). Correctness of local probability propagation in graphical models with loops. *Neural Computation*, *12*, 1–41.

West, P., Rutstein, D. W., Mislevy, R. J., Liu, J., Levy, R. (2012). A Bayes net approach to modeling learning progressions. In A. C. Alonzo A. W. Gotwals (Eds.), *Learning progressions in science: Current challenges and future directions* (pp. 255–291). Rotterdam: Sense.

White, B. Y., Frederiksen, J. R. (1998). Inquiry, modeling, and metacognition: Making science accessible to all students. *Cognition and Instruction*, *16*, 3–118.

Whittaker, J. (1990). *Graphical models in applied multivariate statistics.* New York: Wiley.

Wiggins, G. P. (1998). *Educative assessment: Designing assessments to inform and improve student performance.* San Francisco: Jossey-Bass.

Williamson, D. M. (2000). *Utility of model criticism indices for Bayesian inference networks in cognitive assessment.* Unpublished doctoral dissertation, Fordham University.

Williamson, D. M., Almond, R. G., Mislevy, R. J., Levy, R. (2006). An application of Bayesian networks in automated scoring of computerized simulation tasks. In D. M. Williamson, R. J. Mislevy, I. I. Bejar (Eds.), *Automated scoring of complex tasks in computer-based testing* (pp. 201–257). Hillsdale: Lawrence Erlbaum.

Williamson, D. M., Bauer, M. I., Steinberg, L. S., Mislevy, R. J., DeMark, S. F. (2004). Design rationale for a complex performance assessment. *International Journal of Testing*, *4*, 303–332.

Williamson, D. M., Mislevy, R. J., Almond, R. G. (2000). Model criticism of Bayesian networks with latent variables. In C. Boutilier M. Goldszmidt (Eds.), *Uncertainty in artificial intelligence: Proceedings of the 16th conference* (pp. 634–643). San Mateo: Morgan Kaufmann.

Williamson, D. M., Mislevy, R. J., Almond, R. G. (2004). Evidence-centered design for certification and licensure. *CLEAR Exam Review*, *14*, 14–18.

Williamson, D. M., Mislevy, R. J., Bejar, I. I. (Eds.). (2006). *Automated scoring of complex tasks in computer-based testing.* Hillsdale: Lawrence Erlbaum.

Wilson, M. R. (2004). *Constructing measures: An item response modeling approach.* New York: Routledge.

Wilson, M. R. (2009). Measuring progressions: Assessment structures underlying a learning progression. *Journal of Research in Science Teaching*, *46*(6), 716–730.

Wilson, M. R., Adams, R. J. (1995). Rasch models for item bundles. *Psychometrika*, *60*(2), 181–198.

Wright, S. (1921). Correlation and causation. *Journal of Agricultural Research*, *20*, 557–585.

Wright, S. (1934). The method of path coefficients. *Annals of Mathematical Statistics*, *5*, 161–215.

Yamamoto, K. (1987). *A model that combines IRT and latent class models.* Unpublished doctoral dissertation, University of Illinois, Champaign-Urbana.

Yan, D., Almond, R. G., Mislevy, R. J. (2004). *Comparison of two models for cognitive diagnosis* (Research Report No. RR-04-02). Princeton: Educational Testing Service. Retrieved from http://www.ets.org/research/researcher/RR-04-02.html.

Yan, D., von Davier, A. A., Lewis, C. (Eds.). (2014). *Computerized multistage testing: Theory and applications.* New York: CRC.

Yan, D., Mislevy, R. J., Almond, R. G. (2003). *Design and analysis in a cognitive assessment* (Research Report No. RR-03-32). Princeton: Educational Testing Service. Retrieved from http://www.ets.org/research/researcher/RR-03-32.html.

Yen, W. M. (1993). Scaling performance assessments: Strategies for managing local item dependence. *Journal of Educational Measurement*, *30*, 187–213.

York, J. (1992). Use of the Gibbs sampler in expert systems. *Artificial Intelligence*, *56*, 115–130.

Zadeh, L. (1965). Fuzzy sets. *Information and Control*, *8*, 338–353.

Zalles, D., Haertel, G. D., Mislevy, R. J. (2010). *Using evidence-centered design to support assessment, design and validation of learning progressions* (Large-Scale Assessment Technical Report No. 10). Menlo Park: SRI International. Retrieved from http://ecd.sri.com/downloads/ECD_TR10_Learning_Progressions.pdf.

Zapata-Rivera, J.-D. (2002). cbCPT: Knowledge engineering support for CPTs in Bayesian networks. In *Proceedings of the 15th Canadian conference on artificial intelligence AI 2002* (pp. 368–370).

Zapata-Rivera, J.-D., Greer, J. E. (2004a). Inspectable Bayesian student modelling servers in multi-agent tutoring systems. *International Journal of Human-Computer Studies*, *61*(4), 535–563. doi: 10.1016/j.ijhcs.2003.12.017.

Zapata-Rivera, J.-D., Greer, J. E. (2004b). Interacting with inspectable Bayesian student models. *International Journal of Artificial Intelligence in Education*, *14*(2), 127–163.

Zapata-Rivera, J.-D., Neufeld, E., Greer, J. E. (1999). Visualization of Bayesian belief networks. In *IEEE Visualization 1999: Late Breaking Hot Topics Proceedings* (pp. 85–88). New Brunswick: IEEE Press.

Zapata-Rivera, J.-D., VanWinkle, W. H., Zwick, R. J. (2012). *Applying score design principles in the design of score reports for CBAL teachers* (Research Memorandum No. RM-12-20). Princeton: Educational Testing Service.

Zimowski, M. F., Muraki, E., Mislevy, R. J., Bock, R. D. (2003). BILOG-MG: Multiple-group IRT analysis and test maintenance for binary items [Computer software manual]. Chicago: Scientific Software International.

Author Index

R. G. Almond et al., *Bayesian Networks in Educational Assessment*, Statistics for Social and Behavioral Sciences, DOI 10.1007/978-1-4939-2125-6

Subject Index

R. G. Almond et al., *Bayesian Networks in Educational Assessment*,
Statistics for Social and Behavioral Sciences, DOI 10.1007/978-1-4939-2125-6