CHAPTER THREE

DATA PREPROCESSING

OUTLINES

- Why preprocess the data?
- Descriptive data summarization
  - Graphic display of basic descriptive summaries
- Major Tasks in Data Preprocessing
  - Data cleaning
  - Data integration
  - Data transformation
  - Data reduction
  - Discretization and concept hierarchy generation

Why Data Preprocessing?

- Data which do not have the required quality has the effect of **bad quality mining results!**

- Quality decisions must be based on quality data

- Quality data is key for success in data warehousing and data mining

Why Data Preprocessing?

- Data in the real world is full of dirty
  - **incomplete**:
    - lacking attribute values that is vital for decision making so they have to be added,
    - lacking certain attributes of interest in certain dimension and should be again added with the required value,
    - containing only aggregate data so that the primary source of the aggregation should be included
  - **noisy**: containing errors or outliers that deviate from the expected
  - **inconsistent**: containing discrepancies in codes or names of the organization or domain
  - etc

Why Data Preprocessing?

- Incomplete, noisy and inconsistent data are commonplace properties of large real world databases and data sources
- Data cleaning routine work to clean such problems so that results can be accepted
- Before starting data preprocessing, it will be adviceable to have overall picture of the data we have so that it tell as high level summary such as
  - General property of the data
  - Which data values should be considered as noise or outliers
- This can be done with the help of descriptive data summarization

Descriptive data summarization

- Descriptive summary about data can be generated with the help of measure of central tendency of the data and dispersion of the data
- Measure of ***central tendency*** includes
  - Mean(algebraic function)
  - Median(ordinal type of data),holistic function
  - Mode
  - Mid-Range
- Measure of ***dispersion*** includes
  - range
  - The five number summary (based on Quartiles)
  - Interquartile range (IQR)
  - Standard deviation

Major Tasks in Data Preprocessing

- Data pre-processing in data mining activity refers to the processing of data attributes and values to prepare for the mining operation.

- Any activity performed prior to mining the data is called **pre-processing**

- This involves:
  - Data cleaning
  - Data integration
  - Data transformation
  - Data reduction
  - Data Discretization and concept hierarchy generation

Data Cleaning

- Refers to the process of
  - filling in missing values,
  - smooth noisy data,
  - identify or remove outliers, and
  - resolve inconsistencies

Missing Data

- Data attribute value is not always available (missing data)
  - E.g.,
    - Some patient doesn’t have know their address (patient DB)
    - Some drivers education level is not recorded (traffic penalty data)
    - Encoders doesn’t understand the patient disease code and left un encoded (patient DB)
    - Encoder jump patient telephone unrecorded as he/she feel unimportant (patient DB)
    - Encoder leave the value of an attribute as the attribute is not among the valid value list (Example: age is valid in some system if it is less than 120 hence 130 can not be recorded)

Missing Data

- Causes for missing data
  - equipment malfunction
  - inconsistent with other recorded data and thus deleted
  - data not entered due to lack of understanding
  - certain data may not be considered important at the time of entry and hence left blank
  - not register history or changes of the data
- Missing data may need to be inferred.

How to Handle Missing Data?

- **Ignore the tuple:** usually done when class label is missing (assuming the tasks is classification—not effective when the percentage of missing values per attribute varies considerably.
- **Fill in the missing value manually:** tedious and infeasible
- **Use a global constant to fill in the missing value:** E.g., “unknown”, a new class?! Simple but not recommended as this constant may form some interesting pattern for the data mining task which mislead decision process

How to Handle Missing Data?

- **Use the attribute mean:** for all samples belonging to the same class to fill in the missing value with the class mean

- **Use the most probable value:** fill in the missing values by predicting its value from correlation of the available values and values of other attributes through regression analysis, inference-based tools such as Bayesian formula or decision tree

- Except the first two approach, the rest filled values are incorrect

- The last two approaches are the most commonly used technique to fill missing data

Noisy Data

- Noise is a characteristics of an attribute value when it has incorrect attribute value
- Noise is defined as a random error or variance in a measured variable
- Incorrect attribute values may be due to
  - faulty data collection instruments
  - data entry problems
  - data transmission problems
  - technology limitation
  - inconsistency in naming convention

How to Handle Noisy Data?

- Noisy data can be handled by the techniques such as
  - Simple Discretization Methods (Binning method)
  - Clustering
  - Regression
  - Combined computer and human inspection
    - detect suspicious values and check by human

Handling Noisy Data by Simple Discretization Methods (Binning)

- **Algorithm**
  1. Sort the data and partition into bins
  2. Choose the number of bins (N) and do binning
     - The bins can be ***equal-depth*** or ***equal-width***
  3. Do Smoothing
     - The algorithm can be
       1. smooth by bin means,
       2. smooth by bin median,
       3. smooth by bin boundaries, etc.

Handling Noisy Data by Simple Discretization Methods (Binning)

- Equal-width (distance) partitioning:
  - It divides the range into *N* intervals of equal size: uniform grid
  - if *A* and *B* are the lowest and highest values of the attribute, the width of intervals will be: $W = (B-A)/N$.

Handling Noisy Data by Simple Discretization Methods (Binning)

- Equal-width: Example:
  - Given the data set (say 24, 21, 28, 8, 4, 26, 34, 21, 29, 15, 9, 25)
    - Determine the number of bins N (say 3)
    - Sort the data as 4, 8, 9, 15, 21, 21, 24, 25, 26, 28, 29, 34
    - Determine the range R = Max – Min = 31
    - Divide the range into N equal width where the ith bin is $[X_{i-1}, X_i)$ where X_0=Min and X_N=Max and $X_i = X_{i-1} + R/N$ (R/N=10)
    - Hence X0= 4, X1 = 14, X2 = 24, and X3 = 34
    - Therefore:
      - Bin 1 = 4,8,9
      - Bin 2 = 15, 21, 21
      - Bin3 = 24, 25, 26, 28, 29, 34

Handling Noisy Data by Simple Discretization Methods (Binning)

- Equal-width (distance) partitioning:
  - The most straightforward approach
  - But outliers may dominate presentation
  - Skewed data is not handled well.

Data Cleaning: Handling Noisy Data by Simple Discretization Methods (Binning)

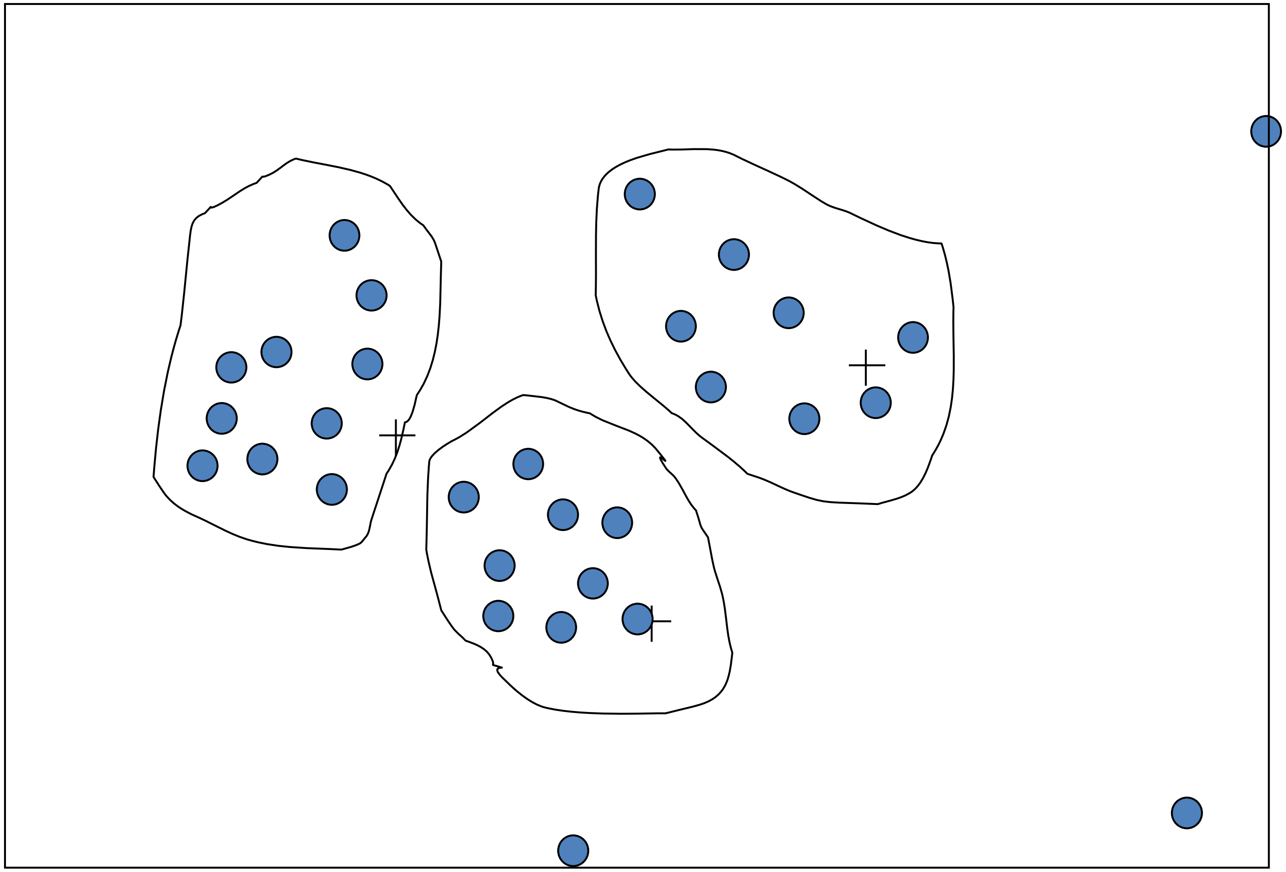
- Equal-depth (frequency) partitioning:
  - It divides the range into *N* intervals, each containing approximately same number of samples
  - Given the data set (say 24, 21, 28, 8, 4, 26, 34, 21, 29, 15, 9, 25)
    - Determine the number of bins : N (say 3)
    - Determine the number of data elements F(F=12)
    - Sort the data as 4, 8, 9, 15, 21, 21, 24, 25, 26, 28, 29, 34
    - Place F/N (12/3 = 4) element in order into the different bins
    - Therefore:
      - Bin 1 = 4,8,9 ,15
      - Bin 2 = 21, 21,24, 25
      - Bin3 = 26, 28, 29, 34

Data Cleaning: Handling Noisy Data by Simple Discretization Methods (Binning)

- Equal-depth (frequency) partitioning:
  - Good data scaling
  - Managing categorical attributes can be tricky.

Handling Noisy Data by Cluster Analysis

- Detect and remove outliers



Data Integration

- **Data integration:**
  - Combines data from multiple sources (databases, data cubes, or files) into a coherent store
- **There are a number of issues to consider during data integration**
- **Some of these are**
  - Schema integration issue
  - Entity identification issue
  - Data value conflict issue
  - Avoiding redundancy issue

Data Integration

- Schema integration
  - Schema refers to the design of an entity and its relation in the data source
  - Integrate metadata from different sources

- Entity identification problem:
  - identify real world entities from multiple data sources which are identical so that they can be integrated properly
  - As data source for data mining differ, the same entity will have different representation in the different sources
  - Identical entities may have different representation of attribute naming in different sources

Data Integration

- **Data value conflict issue**
  - Involves detecting and resolving data value conflicts
  - for the same real world entity, attribute values from different sources may be different
  - possible reasons: different representations, different scales, measurement unit used

Data Integration

- **Avoiding redundancy issue**
  - Redundant data occur often during integration of multiple databases
    - The same attribute may have different names in different databases
    - One attribute may be a “derived” attribute in another table, e.g., annual revenue from monthly revenue
  - Redundant data may be able to be detected by ***correlation analysis for numeric data***

$$r_{A,B} = \frac{\sum_{i=1}^{N}(a_i - \bar{A})(b_i - \bar{B})}{N\sigma_A\sigma_B} = \frac{\sum_{i=1}^{N}(a_i b_i)N\bar{A}\bar{B}}{N\sigma_A\sigma_B}$$

Data Integration

- **Avoiding redundancy issue**
  - The correlation between two attribute A and B ($r_{A,B}$) is always in the range from -1 to +1.
  - $r_{A,B}$ = -1 is to mean negatively correlated, $r_{A,B}$ = 0 to mean uncorrelated and $r_{A,B}$ = +1 is perfectly correlated
  - Careful integration of the data from multiple sources may help to reduce/avoid redundancies and inconsistencies and improve mining speed and quality

Data Transformation

- Data transformation is the process of transforming or consolidating data into a form appropriate for mining which is more appropriate for measurement of similarity and distance

- This involves
  - Smoothing
  - Aggregation
  - Generalization
  - Normalization
  - Attribute/feature construction

Data Transformation

- **Smoothing:** concerned mainly to remove noise from data using techniques such as binning, clustering, and regression

- **Aggregation:** summarization or aggregation operations are performed

- **Generalization:** concept hierarchy climbing (from low level into higher level)

Data Transformation

- **Normalization:** scaled to fall within a small, specified range
  - Used mainly
    - for classification algorithms such as neural network,
    - distance measurements such as clustering, nearest neighbor approach
  - Normalization exist in various forms
    1. min-max normalization
    2. z-score normalization
    3. normalization by decimal scaling
    4. Attribute/feature construction

Data Transformation: Normalization

- Normalization by decimal scaling

$$v' = \frac{v}{10^j}$$ Where *j* is the smallest integer such that Max(|v’|)<1

- For example:
  - given the data set V= 132, -89, 756, -1560, 234, -345 and 1234
  - The value of v’ becomes in the range from -1 to +1 if j=10,000
  - In that case V’ = 0.0132, -0.0089, 0.0756, -0.156, 0.0234, -0.0345, and 0.1234

Data Transformation: Attribute construction

- Attribute construction is the process of driving new attributes from the existing attributes.

- Attribute construction is important to improve performance of data mining as the derived attribute will have more discriminative power than the base attributes

- Enable to discover missing information or information hidden within the data set

- For example area can be derived from width and height which may be more informative than any of the two or their combinations

Data Reduction

- Warehouse may store terabytes of data
- Complex data analysis/mining may take a very long time to run on the complete data set
- Data reduction tries to obtain a reduced representation of the data set that is much smaller in volume but yet produces the same (or almost the same or better) analytical results

Data Reduction

- Data reduction strategies includes
  - Data cube aggregation
  - Attribute subset selection
  - Dimensionality reduction
    - Huffman coding
    - Wavelet transforms
    - Principal component analysis
  - Numerosity reduction
    - Regression and log-linear models
    - Histograms
    - Clustering
    - Sampling

Data reduction strategies: by Data Cube Aggregation

- Data cube aggregation and using it for data mining task reduces the data set size significantly
- For example, one can aggregate sales amount specified at each year and quarter into the sum of the sales amount per year
- Multiple levels of aggregation in data cubes further reduce the size of data to deal with
- One should select appropriate levels of aggregation
- Use the most reduced representation which is sufficient to solve the task

Data reduction strategies: by Attribute subset selection

- Removes irrelevant attribute by attribute relevance analysis
- Let us assume we have **d** set of attributes in the data set.
- This set has 2^d sub sets of attributes and dimensionality reduction refers to selection of the subset which has the minimum number of elements in it and represent the pattern as close as possible with the original attributes
- Hence attribute subset selection may refer one approach of dimensionality reduction

Data reduction strategies: by Attribute subset selection

- Several heuristic for attribute subset (feature) selection exists
- Four of them are:
  - step-wise forward selection
  - step-wise backward elimination
  - combining forward selection and backward elimination
  - decision-tree induction algorithm

Data reduction strategies: by Attribute subset selection

- step-wise forward selection
  - Start with empty set
  - The best single-feature is picked first
  - Then next best feature will be selected conditioned by the first, ...
  - Stop when the selected feature set closely represent the entire features

Data reduction strategies: by Attribute subset selection

– step-wise backward elimination

- Start with all the feature set elements
- The feature which is most irrelevant will be discarded first
- Then next most irrelevant feature will be discarded and repeated, ...
- Stop when removing the next candidate attribute for removal affects the pattern significantly

Data reduction strategies: By Attribute subset selection

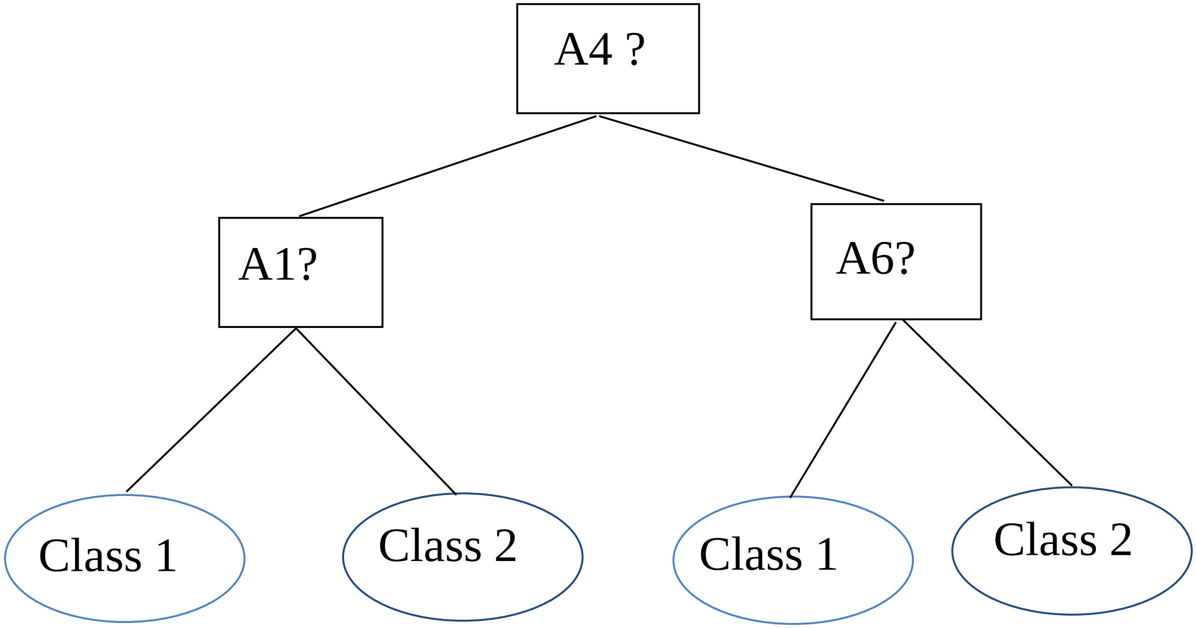
- combining forward selection and backward elimination
  - At each step, the procedure selects the best feature and remove the most irrelevant

- decision-tree induction algorithm
  - This algorithm generate a decision tree using some of the attributes
  - The attributes used in building the decision tree will be taken as attributes that represents closely the entire attributes

Data reduction strategies: By Attribute subset selection

Example of Decision Tree Induction

Initial attribute set:
{A1, A2, A3, A4, A5, A6}

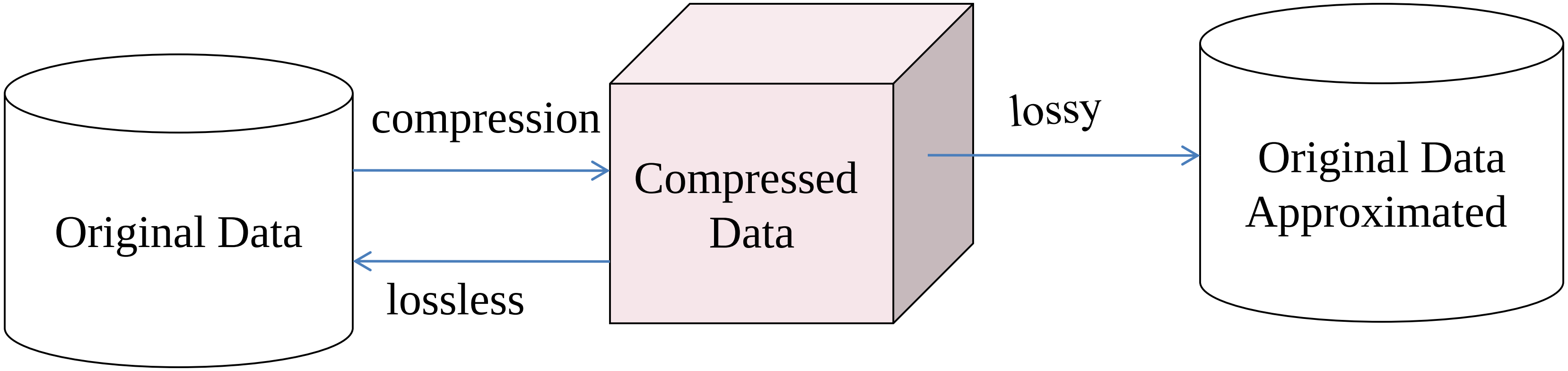


-------> Reduced attribute set: {A1, A4, A6}

Data reduction strategies: Dimensionality reduction

- Tries to compress the data using encoding scheme such as
  - minimum length encoding,
    - Huffman Encoding
  - wavelet encoding,
  - principal component analysis, etc

Data reduction strategies: Dimensionality reduction



Data reduction strategies: Dimensionality reduction

- Compression can be made on data such as string, audio, and video
- String compression
  - There are extensive theories and well-tuned algorithms
  - Typically lossless
  - But only limited manipulation is possible without expansion

- Wavelet transformation and principal component analysis are two of the most common dimension reduction approaches which are lossy

Data reduction : Huffman coding for dimension reduction

- **Huffman coding** is an entropy encoding algorithm used for lossless data compression.
- The term refers to the use of a variable-length code table for encoding a source symbol (such as a character in a file) where the variable-length code table has been derived in a particular way based on the estimated probability of occurrence for each possible value of the source symbol.

Data reduction : Huffman coding for dimension reduction

- Given the data as:

  “***aaab babdbab abcb dbaeb ababd cbaab dbcaebf***”

- This data will be analyzed and may be the followed character set and their frequency will be generated

character	**frquency**
a	**12**
b	**15**
c	**3**
d	**4**
e	**2**
f	**1**
space	**6**
Total	**43**

This will require 43 byte in the normal circumstance

Data reduction : Huffman coding for dimension reduction

- One possible compression is as shown in the table called Huffman coding

character	frquency	possible code	Total bit required
b	15	0	15
1	12	10	24
space	6	110	18
d	4	1110	16
c	3	11110	15
e	2	111110	12
f	1	111111	6
Total	43		106/8=13.25

•The above can be represented by replacing each character with the specified code which require a total of 106 bits (13.25 bytes)

•This reduce the total memory requirement to 30.8%

Data Reduction: Numerosity Reduction using Clustering

- Partition data set into clusters, and one can store cluster representation only (cluster index)
- Can be very effective if data can be organized into distinct clusters but not effective if data is "smeared"
- Can have hierarchical clustering and be stored in multi-dimensional index tree structures like B+ tree
- There are many choices of clustering definitions and clustering algorithms, further detailed will be given later

Data Reduction: Numerosity Reduction using sampling

- Sampling can be used as a data reduction technique because it allows a large data set to be represented by a much smaller sample (or subset) of the data.

- Different types of sampling exists
  - Simple random sampling without replacement
  - Simple random sampling with replacement
  - Cluster sampling (do sampling on cluster, choose m clusters out of the N)
  - Stratified sampling (do simple random sampling on each cluster)

Data Reduction: Numerosity Reduction using sampling

- Simple random sampling may have very poor performance for skewed dataset
- Stratified sampling need to approximate the percentage of each class (or subpopulation of interest) in the overall database
    - Appropriate for data which are skewed

Data Discretization and concept hierarchy generation

- ***Data discritization*** refers to transforming the data set which is usually continous into discrete interval values

- ***Concept hierarchy*** refers to generating the concept levels so that data mining function can be applied at specific concept level

- Data discretization techniques can be used to reduce the number of values for a given continuous attribute by dividing the range of attribute into intervals

- Interval labels can be used to replace actual data values

- This leads to concise, easy to use, knowledge level representation of mining result

Data Discretization and concept hierarchy generation

- ***Discretization:***
  - Divide the range of a continuous attribute into intervals
  - Reduce data size by discretization
  - Prepare data for further analysis
  - Some classification algorithms only accept categorical attributes and hence numeric attribute require discretization.

- ***Concept hierarchies***
  - Reduce the data by collecting and replacing low level concepts (such as measure of all the days for the attribute time) by higher level concepts (such as measure of all the weeks that reduce the data from N tuples to N/7 tuple).
  - Concept hierarchy generation refers to finding hierarchical relation in the data set and build the concept hierarchy automatically

Data Discretization and concept hierarchy generation

- Types of attributes:
  - ***Nominal/Categorical***
    - finite number of possible values, no ordering among values
    - values from an unordered set like location, address, color, sex, marital status
  - ***Ordinal***
    - There are two types of ordinal type attribute
    - ***Discrete ordinal attribute:***
      - it is an attribute whose possible values are ordered and preceding and succeeding elements are well defined
      - Example: age, academic rank, letter grade,
    - ***Continuous ordinal attribute***
      - it is an attribute whose values are ordered but preceding and succeeding elements are not well defined
      - Example: height, area, voltage, etc

Discretization and concept hierarchy generation for numeric data

- It is difficult and laborious to specify concept hierarchies for numerical attributes because of the wide diversity of possible data ranges and frequent update of data values

- Concept hierarchies for numerical data can be constructed automatically based on the data discretization process

Discretization and concept hierarchy generation for numeric data

- Methods for discretization and concept hierarchy generation
  - Binning
  - Histogram analysis
  - Clustering analysis
  - Entropy-based discretization
  - Segmentation by natural partitioning
  - Interval merging by χ^2 Analysis