UNIT – I Introduction

- Data Mining,
- Data Mining Task Primitives,
- Data: Data, Information and Knowledge;
- Attribute Types: Nominal, Binary, Ordinal and Numeric attributes, Discrete versus Continuous Attributes;
- Introduction to Data Preprocessing,
- Data Cleaning: Missing values, Noisy data;
- Data integration: Correlation analysis;
- Transformation: Min-max normalization, z-score normalization and decimal scaling;
- Data reduction: Data Cube Aggregation, Attribute Subset Selection, sampling;
- Data Discretization: Binning, Histogram Analysis

Data Mining

Data Mining is defined as the method to extract information from huge sets of data. Data mining is mining of knowledge from information and getting information by analyzing huge data. Data mining is the process of discovering meaningful new correlations, patterns and trends by sifting through large amounts of data stored in repositories, using pattern recognition technologies as well as statistical and mathematical techniques.

Data Mining Task Primitives

Data Mining Tasks are as follows

1. Description: Most of the times, researchers and analysts need ways to describe patterns and trends lying within the data. Descriptions of patterns and trends are required. Data mining models should be transparent, that means, whatever the data mining model result should describe clear and unambiguous patterns that can interpretable.
2. Estimation: It helps to approximate the value of a numeric target variable using a set of numeric and/or categorical predictor variables. Models are built using entire records and this provides the predictors. Using this predictor, target variable's value can be estimated for new observations.
3. Prediction: It is used to predict missing or unavailable numerical data values rather than class labels. Regression Analysis is generally used for prediction. Prediction can also be used for identification of distribution trends based on available data.
4. Classification: Classification predicts the class of objects whose class label is unknown. By analyzing training data, a model is built which will be further used to identify class of objects.
5. Clustering: A cluster is a collection of records that are similar to one another, and dissimilar to records in other clusters.
6. Association: Associations are used to identify patterns that find always together. This process refers to the process of finding the relationship between attributes and determining association rules.

Data: Data, Information and Knowledge

Data: is any raw record.
Information: analyzed or processed data.
Knowledge: acquired from information.
The steps involved in the process of knowledge discovery (Figure 1.1) are as follows:

1. Data cleaning: to remove noise and inconsistent data
2. Data integration: multiple data sources may be combined.
3. Data selection: data relevant to the analysis task are retrieved from the database.
4. Data transformation: data are transformed and consolidated into forms appropriate for mining by performing summary or aggregation operations.
5. Data mining: an essential process where intelligent methods are applied to extract data patterns.
6. Pattern evaluation: to identify the truly interesting patterns representing knowledge.

7. Knowledge presentation: visualization and knowledge representation techniques are used to present mined knowledge to users.

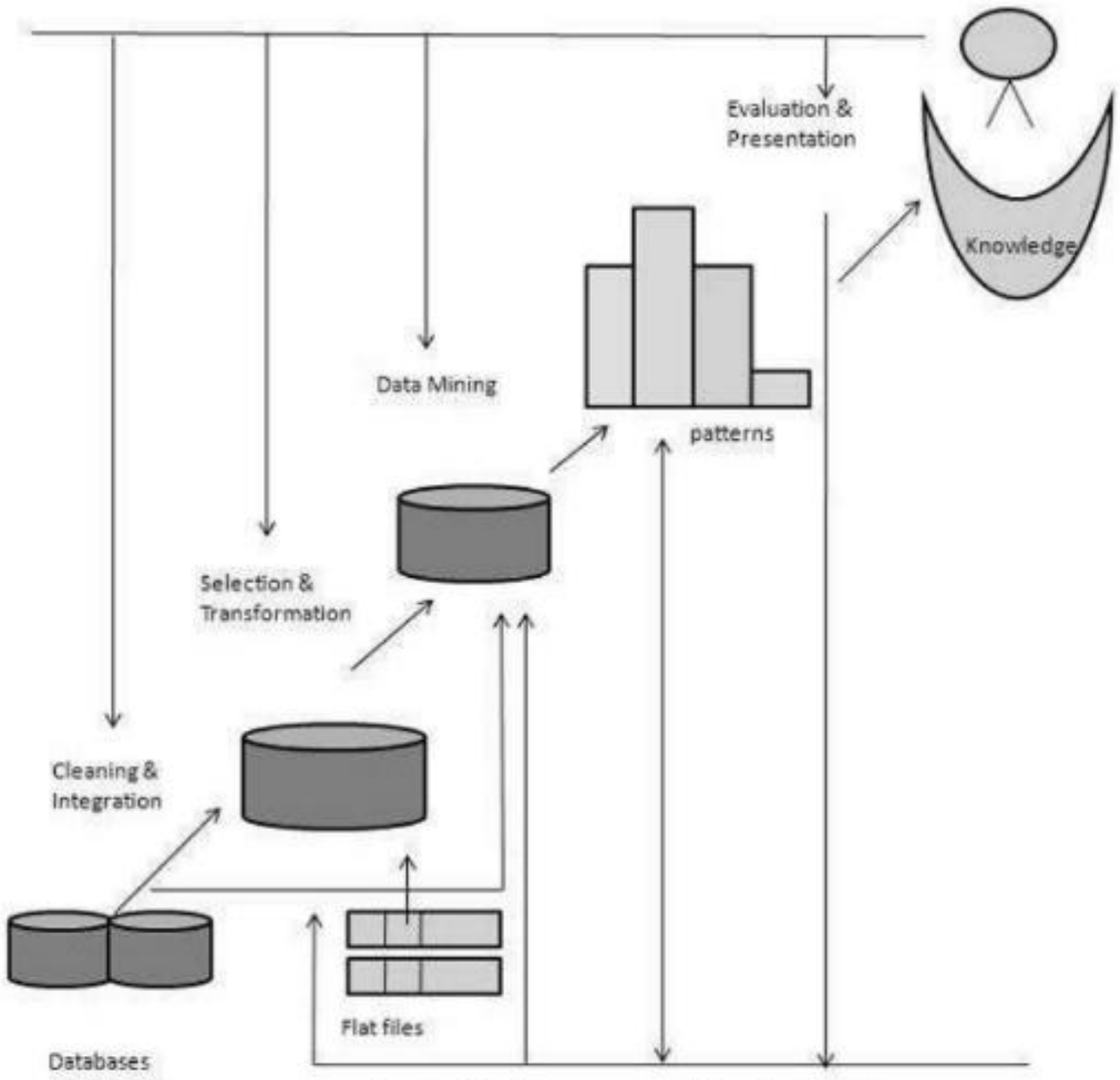


Figure 1.1: Steps involved in the process of knowledge discovery.

Attribute Types:

There are different types of attributes. They are:

1. Nominal
2. Binary
3. Ordinal and Numeric attributes
4. Discrete versus Continuous Attributes

1. Nominal: Nominal means "relating to names." The values of a nominal attribute are symbols or names of things. Each value will be categorical like code, or state. The values do not have any meaningful order. In computer science, the values are also known as enumerations.
   Example – Marital Status of person can be single, married, divorced.
2. Binary: A binary attribute is a nominal attribute, difference is in binary there are only two categories or states: 0 or 1, where 0 typically means that the attribute is absent and 1 means that it is present. Binary attributes can also have values in true and false.
   Example – Medical test of patient may be binary, 1 represent the patient having disease and 0 represent he is not having disease.
3. Ordinal and Numeric attributes: An ordinal attribute is an attribute with possible values that have a meaningful order or ranking among them, but the magnitude between successive values is not known.
   Example – grades of student can be A+, A, B+, B, C
4. Discrete versus Continuous Attributes:
   A discrete attribute has a finite or countably infinite set of values, which may or may not be represented as integers.
   Example - attributes hair color, smoker, medical test, and drink size each have a finite number of values, and so are discrete.
   Continuous attributes are not discrete. Has real numbers as attribute values.
   Example – temperature, weight.

Introduction to Data Preprocessing

- Process that involves transformation of data into information through classifying, sorting, merging, recording, retrieving, transmitting, or reporting is called data processing. Data processing can be manual or computer based.
- In Business related world, data processing refers to data processing so as to enable effective functioning of the organizations and businesses.
- Computer data processing refers to a process that takes the data input via a program and summarizes, analyze the same or convert it to useful information.
- The processing of data may also be automated.
- Data processing systems are also known as information systems.
- When data processing does not involve any data manipulation and only converts the data type it may be called as data conversion.

The Quality or properties or characteristics of data are as follows:

1. Accuracy
2. Completeness

3. Consistency
4. Timeliness
5. Believability
6. Interpretability

Preprocessing of data is required to fulfill above qualities of data. Incomplete, inaccurate, inconsistent data cannot be used in data mining techniques.

Major tasks in data pre-processing:

Data Cleaning: This process consists of filling of missing values, smoothening noisy data, identifying and removing any outliers present and resolving inconsistencies.

Data integration: This refers to integrating data from multiple sources like databases, data cubes, or files.

Data transformation: Normalization and aggregation.

Data reduction: In data reduction the amount of data is reduced but same analytical results are produced.

Data discretization: Part of data reduction, replacing numerical attributes with nominal ones.

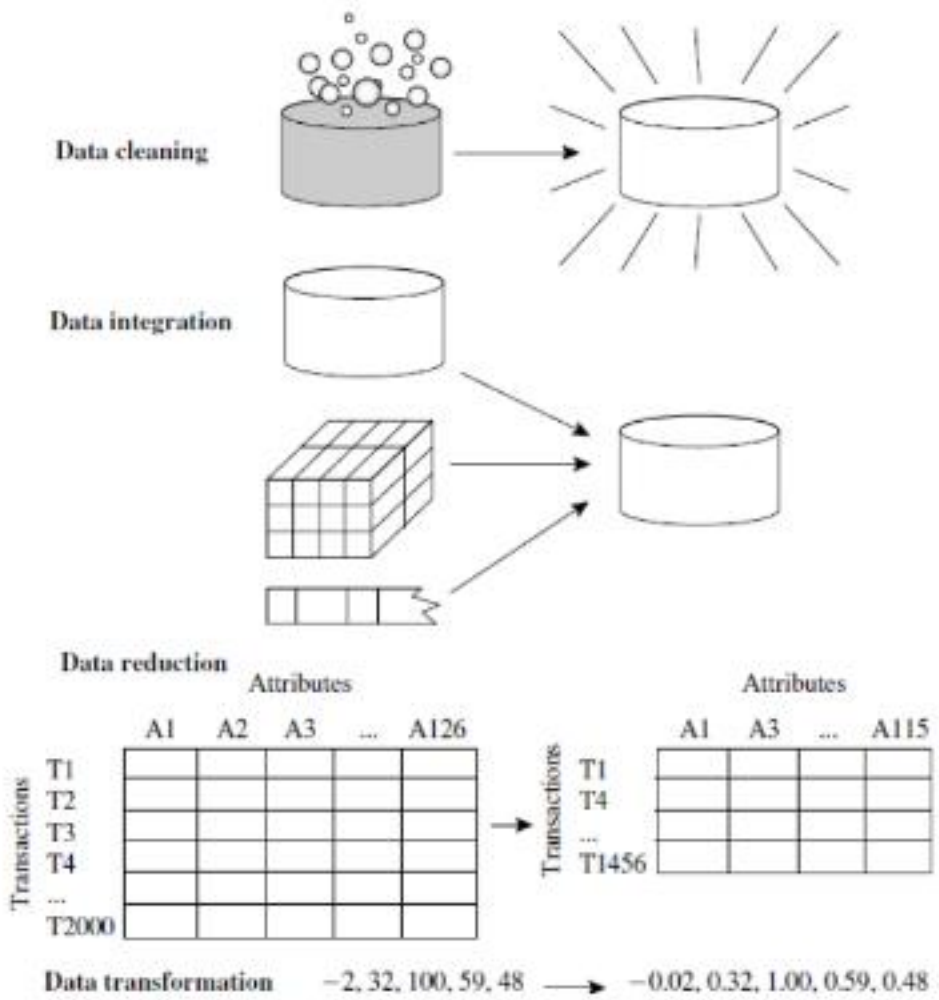


Figure 1.2: Forms or steps or tasks of data preprocessing

Data Cleaning: Missing values, Noisy data

Data Cleaning: Data cleaning routines attempt to fill in missing values, smooth out noise while identifying outliers, and correct inconsistencies in the data. The data cleaning process detects and removes the errors and inconsistencies and improve the quality of the data. Data quality problem arise due to misspellings during data entry, missing values or any other invalid data.

Missing Values:

This involves searching for empty fields where value should occur.

Data preprocessing is one of the most important stages in data mining. Real world data is incomplete, noisy or inconsistent, this data is corrected in data preprocessing out the noise and correcting inconsistencies.

There are several techniques for dealing with missing data, choosing one of them would be dependent on problems domain and the goal for data mining process.

Following are the different ways for handle missing values in databases:

1. **Ignore the data row:** In case of classification suppose a class label is missing for a row, such a data row, such a data row could be ignored, or many attributes within a row are missing even in this case data row could be ignored. If the percentage of such rows is high it will result in poor performance. For example, suppose we have to build a model for predicting student success in college. For this purpose a student's database having information about age, score, address, etc. and column classifying their success in college to "LOW", "MEDIUM" and "HIGH". In this the data rows in which the success column is missing. These types of rows are of no use in the model therefore they can be ignored.
2. **Fill the missing values manually:** This is not feasible for large data set and also time consuming.
3. **Use a global constant to fill in for missing values:** When missing values are different to be predicted, a global constant value like "unknown", "N/A" or "minus infinity" can be used to fill all the missing values. For example, consider the students database, if the address attribute is missing for some students it does not makes sense in filling up these values rather a global constant can be used.
4. **Use attribute mean:** For missing values, mean or median of its discrete values may be used as a replacement. For example, in a database of family incomes, missing values may be replaced with the average income.
5. **Use attribute mean for all samples belonging to the same class:** Instead of replacing the missing values by mean or median of all the rows in the database, rather we could

consider class wise data for missing values to be replaced by its mean or median to make it more relevant. For example, consider a car pricing database with classes like "luxury" and "low budget" and missing values need to filled in, replacing missing cost of a luxury car with average cost of all luxury car makes the data more accurate.

6. **Use a data-mining algorithm to predict the most probable value:** Missing values may also be filled up by using techniques like regression, inference based tools using Bayesian formalism, decision trees, clustering algorithms. For example, clustering method may be used to form clusters and then the mean or median of that cluster may be used for missing value. Decision tree may be used to predict the most probable value based on the other attributes.

Noisy data: A random error or variance in a measure variable is known as noise.

Noise in the data may be introduced due to:

- Fault in data collection instruments.
- Error introduced as data entry by a human or a computer.
- Data transmission errors.

Different types of noise in data:

- Unknown encoding: Gender: E
- Out of range values: Temperature: 1004, Age: 125
- Inconsistent entries : DOB : 10-Feb-2003; Age : 30
- Inconsistent formats : DOB : 11-Feb-1984; DOJ : 2/11/2007

Data smoothing techniques are:

1. Binning
2. Regression
3. Outlier analysis

1. **Binning:** Considering the neighborhood of the sorted data smoothening can be applied.
   - The sorted data is placed into bins or buckets.
   - Smoothing can be done by bins means, bins medians, bins boundaries.
   - Example: Sorted data for price (in dollars): 4, 8, 15, 21, 21, 24, 25, 28, 34
     - **Partition into (equal-frequency) bins**:
       Bin 1: 4, 8, 15
       Bin 2: 21, 21, 24
       Bin 3: 25, 28, 34
     - **Smoothing by bin means**:
       Bin 1: 9, 9, 9
       Bin 2: 22, 22, 22
       Bin 3: 29, 29, 29

- **Smoothing by bin medians**:
  Bin 1: 8, 8, 8
  Bin 2: 21, 21, 21
  Bin 3: 28, 28, 28
- **Smoothing by bin boundaries**:
  Bin 1: 4, 4, 15
  Bin 2: 21, 21, 24
  Bin 3: 25, 25, 34

2. **Outlier analysis by clustering:**
   - Partition data set into clusters and one can store cluster representation only, i.e. replace all values of the cluster by that one value representing the cluster.
   - Outliers can be detected by using clustering techniques, where related values are organized into groups or clusters.
   - Perform clustering on attributes values and replace all values in the cluster by a cluster representative.

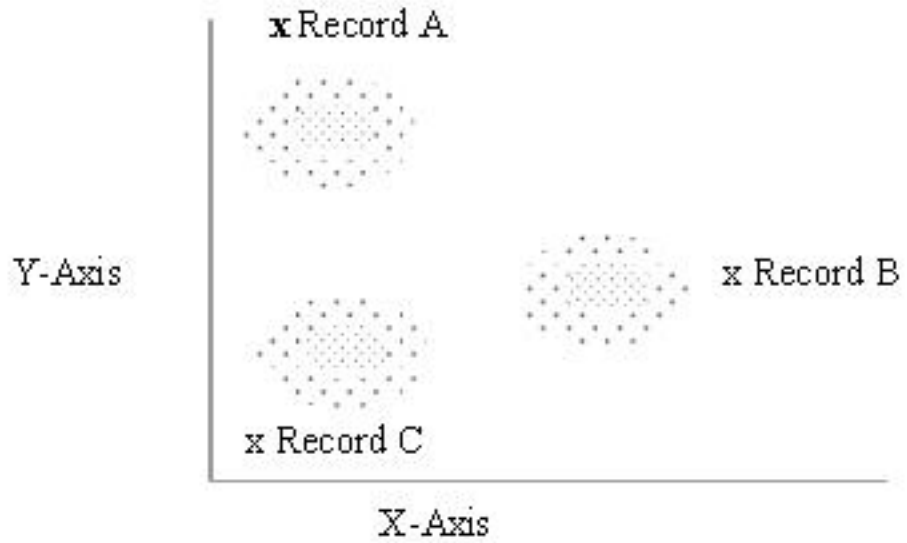


Fig. 1.3 Graphical Example of Clustering

3. **Regression :**
   - Regression is a statistical measure used to determine the strength of the relationship between one dependent variable denoted by Y and a series of independent changing variables.
   - Smooth by fitting the data into regression functions.
   - Use regression analysis on values of attributes to fill missing values.
   - The two basic types of regression are linear regression and multiple regressions.
   - The difference between Linear and multiple regressions is that former uses one independent variable to predict the outcome, while the later uses two or more independent variable to predict the outcome.
   - The general form of each type of regression is :
     Linear Regression: $Y = a+bX+u$

Multiple Regression: $Y = a + b_1X_1 + b_2X_2 + b_3X_3 + \ldots\ldots\ldots\ldots + b_tX_t + u$

Where, Y = the variable that we are trying to predict

X = the variable that we are using to predict Y

a = the intercept

b = the slope

u = the regression residual.

- In multiple regressions each variable is differentiated with subscripted numbers.
- Regressions use a group of random variables for prediction and find a mathematical relationship between them. This relationship is depicted in the form of a straight line (Line regression) that approximates all the points in the best way.
- Regression may be used to determine for e.g. price of commodity, interest rates, the price movement of an asset influenced by industries or sectors.

Data integration: Correlation analysis

Data Integration:

Introduction to Data Integration: A coherent data store (e.g. a Data Warehouse) is prepared by collecting data from multiple sources like multiple databases, data cubes or flat files.

Entity Identification Problem:

- Schema integration is an issue as to integrate metadata from different sources is a difficult task.
- Identity real world entities from multiple data sources and their matching are the entity identification problem. For example, Roll number in one database and enrolment number in another database refers to the same attribute.
- Such conflicts may create problem for schema integration.

Redundancy and Correlation Analysis:

- Data redundancy occurs when data from multiple sources is considered for integration.
- Attribute naming may be a problem as same attributes may have different names in multiple databases.
- An attribute may be derived attribute in another table e.g. "yearly income".
- Redundancy can be detected using correlation analysis.
- To reduce or avoid redundancies and inconsistencies data integration must be carried out carefully. This will also improve mining algorithm speed and quality.

The X^2 (Chi-square):

- It is used to test hypotheses about the shape or proportions of a population distribution by means of sample data.
- For nominal data, a correlation relationship between two attributes, P and Q, can be obtained by an X^2 (Chi-square) test.
- These nominal variables, also called "attribute variables" or "categorical variables", classify observations into a small number of categories, which are not numbers. It doesn't work for numeric data.
- Examples of nominal variables include Gender (the possible values are male or female), Marital Status (Married, unmarried or divorced), etc.
- The Chi-square test is used to test the probability of independence of a distribution of data but does not gives you any details about the relationship between them.
    - Chi-square test is defined by,

$$X^2 = \sum \left[\frac{(O - E)^2}{E} \right]$$

Where, X^2 = Chi-square

E = Frequently expected which is the amount of subjects that you would expect to find in each category based on known information.

O = Frequently observed which is the amount of subjects you actually found to be in each category in the present data.

- **Degrees of freedom :** The degree of freedom (DF) is equal to :

$$\mathbf{DF = (r-1)*(c-1)}$$

Where, r is the number of levels for one categorical variable and c is the number of levels for the other categorical variable.

- **Expected frequencies :** It is the count which is computed for each level of categorical attribute. The formula for expected frequency is

$$E_{r,c} = (n_r * n_c)/n$$

    - Where $E_{r,c}$ is the expected frequency count for level r of attitude X and level c of attribute Y,
    - n_r is the sum of sample observations at level r of attribute X,
    - n_c is the sum of sample observations at level c of attribute Y,
    - n is the total size of sample data.

Transformation: Min-max normalization, z-score normalization and decimal scaling;

- Data Transformation by Normalization or standardization is the process of making an entire set of values has a particular property.
- Following methods may be used for normalization:
  1. Min-Max
  2. Z-score
  3. Decimal scaling

1. **Min-Max Normalization:** Min-max normalization performs a linear transformation on the original data. Following formula may be used to perform mapping a v_i value, of an attribute A from range $[min_A, max_A]$ to a new value v_i' in range $[new\_min_A, new\_max_A]$,

   $$v_i' = \frac{v_i - min_A}{max_A - min_A}(new\_min_A - new\_max_A) + new\_min_A$$

   Example: Suppose that the minimum and maximum values for the attribute income are \$12,000 and \$98,000, respectively. We would like to map income to the range [0.0, 1.0]. By min-max normalization, a value of \$73,600 for income is transformed to $\frac{73,600-12,000}{98,000-12,000}(1.0 - 0.0) + 0 = 0.716$.

2. **Z-score:** In Z-score normalization, data is normalized based on the mean and standard deviation. Z-score is also known as Zero mean normalization.

   $$v_i' = \frac{v_i - \bar{A}}{\sigma_A}$$

   where, $\bar{A}$ = mean of A

   σ_A = Standard deviation of all values of A

   Example: Suppose that the mean and standard deviation of the values for the attribute *income* are \$54,000 and \$16,000, respectively. With z-score normalization, a value of \$73,600 for *income* is transformed to $\frac{73,600-54,000}{16,000} = 1.225$

3. **Decimal Scaling:** Based on the maximum absolute value of the attributes the decimal point is moved. This process is called as Decimal Scale Normalization

   $v_i' = \frac{v_i}{10^k}$ for the smallest k such that max $(|v_i'|)<1$.

   Example: Suppose that the recorded values of A range from -813 to 119. The maximum absolute value of A is 119. To normalize by decimal scaling, we therefore divide each value by 1000 (i.e., j = 3) so that -813 normalizes to -0.813 and 119 normalizes to 0.119.

Data reduction: Data Cube Aggregation, Attribute Subset Selection, sampling

Data reduction techniques are applied to get a reduced representation of the data set that is comparatively very smaller in volume, yet maintains the integrity of the original data. That is, mining on the reduced data set should be more efficient yet produce the approximately same analytical results.

Need for Data Reduction:

1. **Reducing the number of attributes:**
    - **Data cube aggregation:** This process is involves applying OLAP operation like roll-up, slice or dice operations.
    - **Removing irrelevant attributes:** In this attribute selection methods like filtering and wrapper methods may be used, it also involves searching the attribute space.
    - **Principle component analysis (numeric attributes only):** This involves representing the data in a compact form by using a lower dimensional space.
2. **Reducing the number of attribute values :**
    - **Binning(histograms):** This involves representing the attributes into groups called as bins, this will result into lesser number of attributes.
    - **Clustering:** Grouping the data based on their similarity into groups called as clusters.
    - Aggregation or generalization.
3. **Reducing the number of tuples:**

    To reduce the number of tuples, sampling may be used.

Data reduction technique.

i. Dimensionality reduction: attribute subset selection
ii. Data compression
iii. Numerosity reduction: data cube aggregation, sampling

i. Data cube aggregation :-

- It reduces the data to the concept level needed in the analysis and uses the smallest (most detailed) level necessary to solve the problem
- Queries regarding aggregated information should be answered using data cube when possible.

Example:

Total annual sales of TV in USA is aggregated quarterly as shown in Fig. 1.3

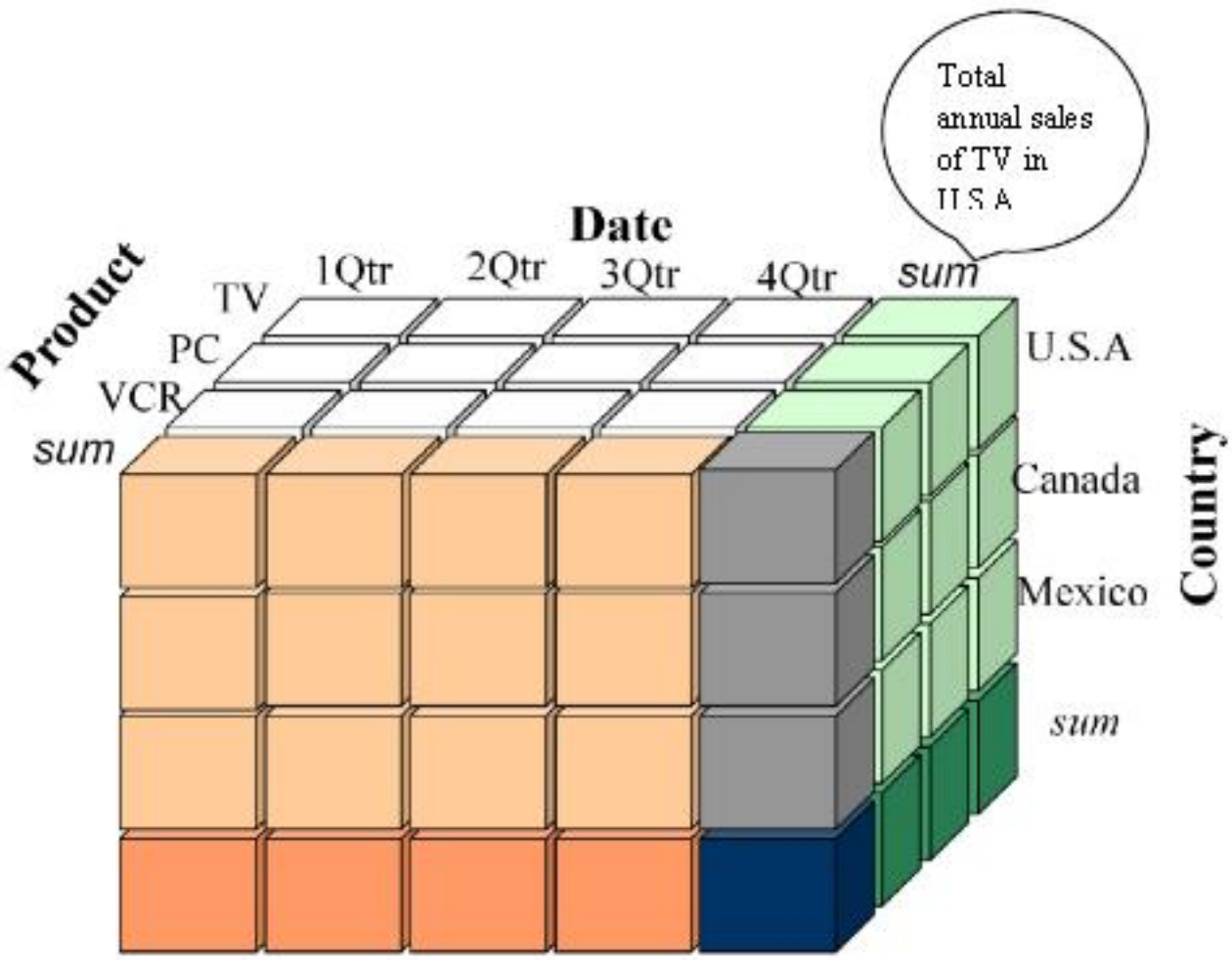


Figure 1.3: Example of data cube

Attribute subset selection:

How to find a good subset of the original attributes?

Attribute subset selection refers to a process in which minimum set of attributes are selected in such a way that their distribution represents the same as the original data set distribution considering all the attributes.

Different attribute subset selection techniques:

1. **Forward selection :**
   - Start with empty set of attributes.
   - Determine the best of the original attributes and add it to the set.
   - At each step, find the best of the remaining original attributes and add it to the set.
2. **Stepwise backward elimination :**
   - Starts with the full set of attributes.
   - At each step, it removes the worst attribute remaining in the set.
3. **Combination of forward selection and backward elimination :**

- The process combines and selects the best attribute and removes the worst among the remaining attributes.
- For all above method stopping criteria is different and it requires a threshold on the measure used to stop the attribute selection process.

4. **Decision tree induction :**
   - ID3, C4.5 intended for classification.
   - Construct a flow chart like structure.
   - A decision tree is a tree in which :
     - Each internal node tests an attribute.
     - Each branch corresponds to attribute value.
     - Each leaf node assigns a classification.

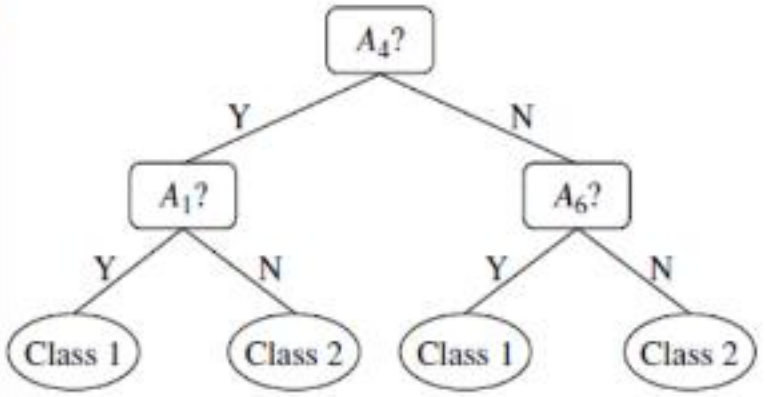
Forward selection	Backward elimination	Decision tree induction
Initial attribute set: $\{A_1, A_2, A_3, A_4, A_5, A_6\}$ Initial reduced set: $\{\}$ $\Rightarrow \{A_1\}$ $\Rightarrow \{A_1, A_4\}$ => Reduced attribute set: $\{A_1, A_4, A_6\}$	Initial attribute set: $\{A_1, A_2, A_3, A_4, A_5, A_6\}$ $\Rightarrow \{A_1, A_3, A_4, A_5, A_6\}$ $\Rightarrow \{A_1, A_4, A_5, A_6\}$ => Reduced attribute set: $\{A_1, A_4, A_6\}$	Initial attribute set: $\{A_1, A_2, A_3, A_4, A_5, A_6\}$  => Reduced attribute set: $\{A_1, A_4, A_6\}$

Figure 1.4: Attribute subset selection.

Sampling:

- Sampling is used in preliminary investigation as well as final analysis of data.
- Sampling is important in data mining as processing the entire data set is expensive and time consuming.

Types of sampling:

1. **Simple random sampling :**
   There is an equal probability of selecting any particular item.
2. **Sampling without replacement :**
   As each item is selected, it is removed from the population.

3. **Sampling with replacement :**
   The objects selected for the sample is not removed from the population. In this technique the same object may be selected multiple times.
4. **Stratified sampling :**
   The data is split into partitions and samples are drawn from each partition randomly.

Data Discretization: Binning, Histogram Analysis

- The range of a continuous attribute is divided into intervals.
- Categorical attributes are accepted by only a few classification algorithms.
- By Discretization the size of the data is reduced and prepared for further analysis.
- Dividing the range of attributes into intervals would reduce the number of values for a given continuous attribute.
- Actual data values may be replaced by interval labels.
- Discretization process may be applied recursively on an attribute.

Binning: Considering the neighborhood of the sorted data smoothening can be applied.

- The sorted data is placed into bins or buckets.
- Smoothing can be done by bins means, bins medians, bins boundaries.
- Example: Sorted data for price (in dollars): 4, 8, 15, 21, 21, 24, 25, 28, 34
  - **Partition into (equal-frequency) bins**:
    Bin 1: 4, 8, 15
    Bin 2: 21, 21, 24
    Bin 3: 25, 28, 34
  - **Smoothing by bin means**:
    Bin 1: 9, 9, 9
    Bin 2: 22, 22, 22
    Bin 3: 29, 29, 29
  - **Smoothing by bin medians**:
    Bin 1: 8, 8, 8
    Bin 2: 21, 21, 21
    Bin 3: 28, 28, 28
  - **Smoothing by bin boundaries**:
    Bin 1: 4, 4, 15
    Bin 2: 21, 21, 24
    Bin 3: 25, 25, 34

Histogram Analysis:

Histogram analysis is an unsupervised discretization technique because it does not use class information. Histograms use binning to approximate data distributions and are a popular form of data reduction. The histogram analysis algorithm can be applied recursively to each partition in order to automatically generate a multilevel concept hierarchy, with the procedure terminating

once a prespecified number of concept levels has been reached. A minimum interval size can also be used per level to control the recursive procedure. This specifies the minimum width of a partition, or the minimum number of values for each partition at each level.

Example: The following data are a list of All Electronics prices for commonly sold items (rounded to the nearest dollar). The numbers have been sorted: 1, 1, 5, 5, 5, 5, 5, 8, 8, 10, 10, 10, 10, 12, 14, 14, 14, 15, 15, 15, 15, 15, 15, 18, 18, 18, 18, 18, 18, 18, 18, 20, 20, 20, 20, 20, 20, 20, 21, 21, 21, 21, 25, 25, 25, 25, 25, 28, 28, 30, 30, 30.

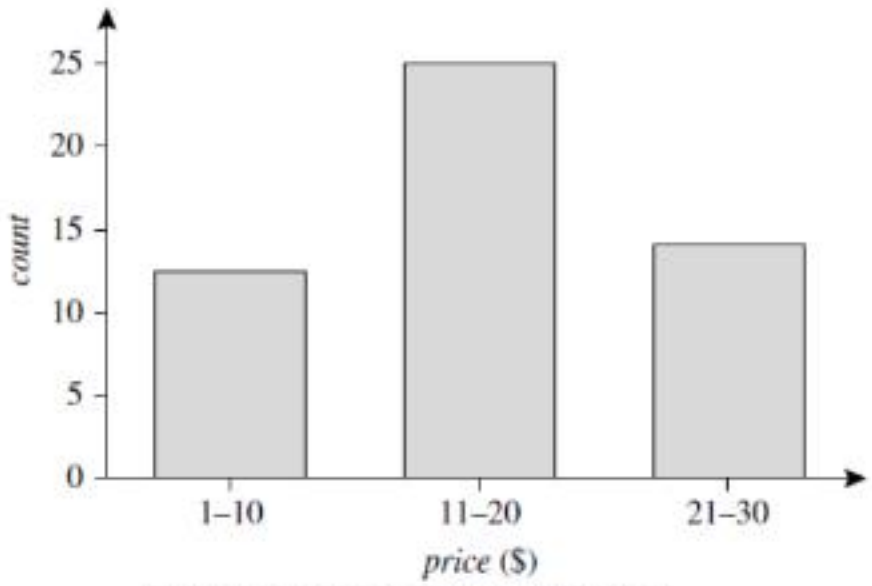


Figure 1.5: Equal-width histogram

Equal-width histogram: In an equal-width histogram, the width of each bucket range is uniform (Figure 1.5).

UNIT – II Data Warehouse

- Data Warehouse,
- Operational Database Systems and Data Warehouses(OLTP Vs OLAP),
- A Multidimensional Data Model: Data Cubes, Stars, Snowflakes, and Fact Constellations Schemas;
- OLAP Operations in the Multidimensional Data Model, Concept Hierarchies,
- Data Warehouse Architecture,
- The Process of Data Warehouse Design,
- A three-tier data warehousing architecture,
- Types of OLAP Servers: ROLAP versus MOLAP versus HOLAP.

Data Warehouse:

- Data warehouses generalize and consolidate data in multidimensional space. Data Warehouse is used to store historical data which helps to take strategic decision for business. It is used for Online Analytical Processing (OLAP) which helps to analyze the data. Data warehousing gives designs and tools to business administrators to efficiently sort out, comprehend, and utilize their information to settle on key choices.
- Data is de-normalized so tables are not complex and reduces the response time for analytical queries.
- Data-modeling techniques like start schema are used for the Data Warehouse design.
- Read operation on data warehouse are optimized as it has been used frequently for analysis purpose so performance is high for queries.

OLAP:

OLAP defined:

- OLAP or the Online Analytical supports the multidimensional view of data.
- OLAP provide fast, steady and proficient access to the various views of information.
- The complex queries can be processed.
- It's easy to analyze information by processing complex queries on multidimensional views of data.
- Data warehouse is generally used to analyze the information where huge amount of historical data is stored.
- Information in data warehouse is related to more than one dimension like sales, market trends, buying patterns, supplier, etc.

Differences between OLTP and OLAP:

Feature	*OLTP*	*OLAP*
Characteristic	operational processing	informational processing
Orientation	transaction	analysis
User	clerk, DBA, database professional	knowledge worker (e.g., manager, executive, analyst)
Function	day-to-day operations	long-term informational requirements decision support
DB design	ER-based, application-oriented	star/snowflake, subject-oriented
Data	current, guaranteed up-to-date	historic, accuracy maintained over time
Summarization	primitive, highly detailed	summarized, consolidated
View	detailed, flat relational	summarized, multidimensional
Unit of work	short, simple transaction	complex query
Access	read/write	mostly read
Focus	data in	information out
Operations	index/hash on primary key	lots of scans
Number of records accessed	tens	millions
Number of users	thousands	hundreds
DB size	GB to high-order GB	$\geq$ TB
Priority	high performance, high availability	high flexibility, end-user autonomy
Metric	transaction throughput	query throughput, response time

OLAP Operations or OLAP Techniques:

- OLAP techniques are implemented to retrieve the information from data warehouse into OLAP multi-dimensional databases. So information can be retrieved using front end system.
- Multidimensional models are used to inhabit data in multi-dimensional matrices like Data Cubes or Hypercubes. A standard spreadsheet, signifying a conventional database, is a two-dimensional matrix.

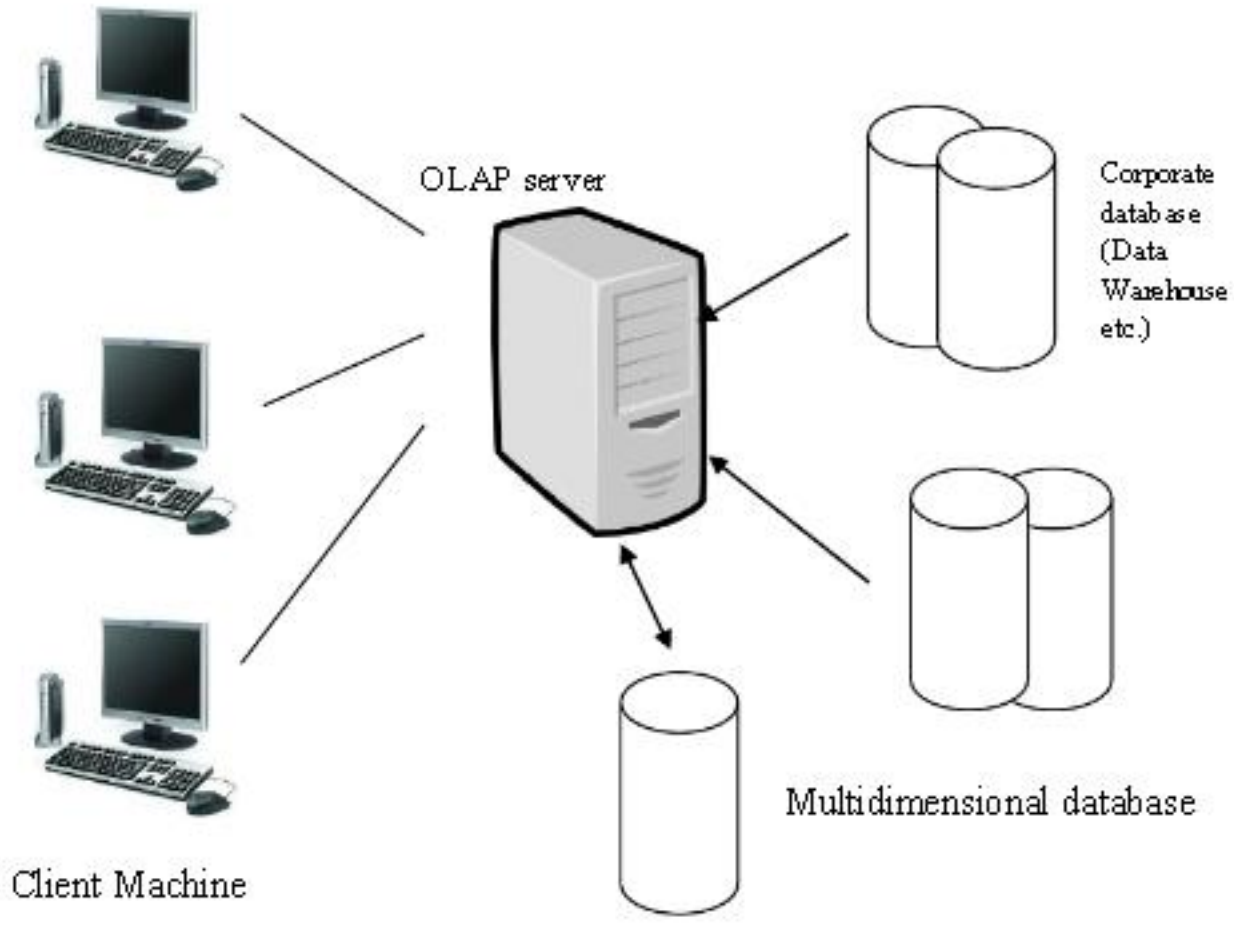


Figure 1.1: OLAP Implementation

A Multidimensional Data Model: Stars, Snowflakes, and Fact Constellations Schemas

A multidimensional model has two types of tables:

1. **Dimension tables:** set of smaller attendant tables, one for each dimension.
2. **Fact tables:** contains facts of measures. It contains bulk of data with no redundancy.

Star Schema: It contains

1. One fact table.
2. Any number of dimension tables.

Example: All Electronics sales (Figure 1.3) contains

1. One Fact table (sales).
2. 4 dimension tables (time, item, branch and location).

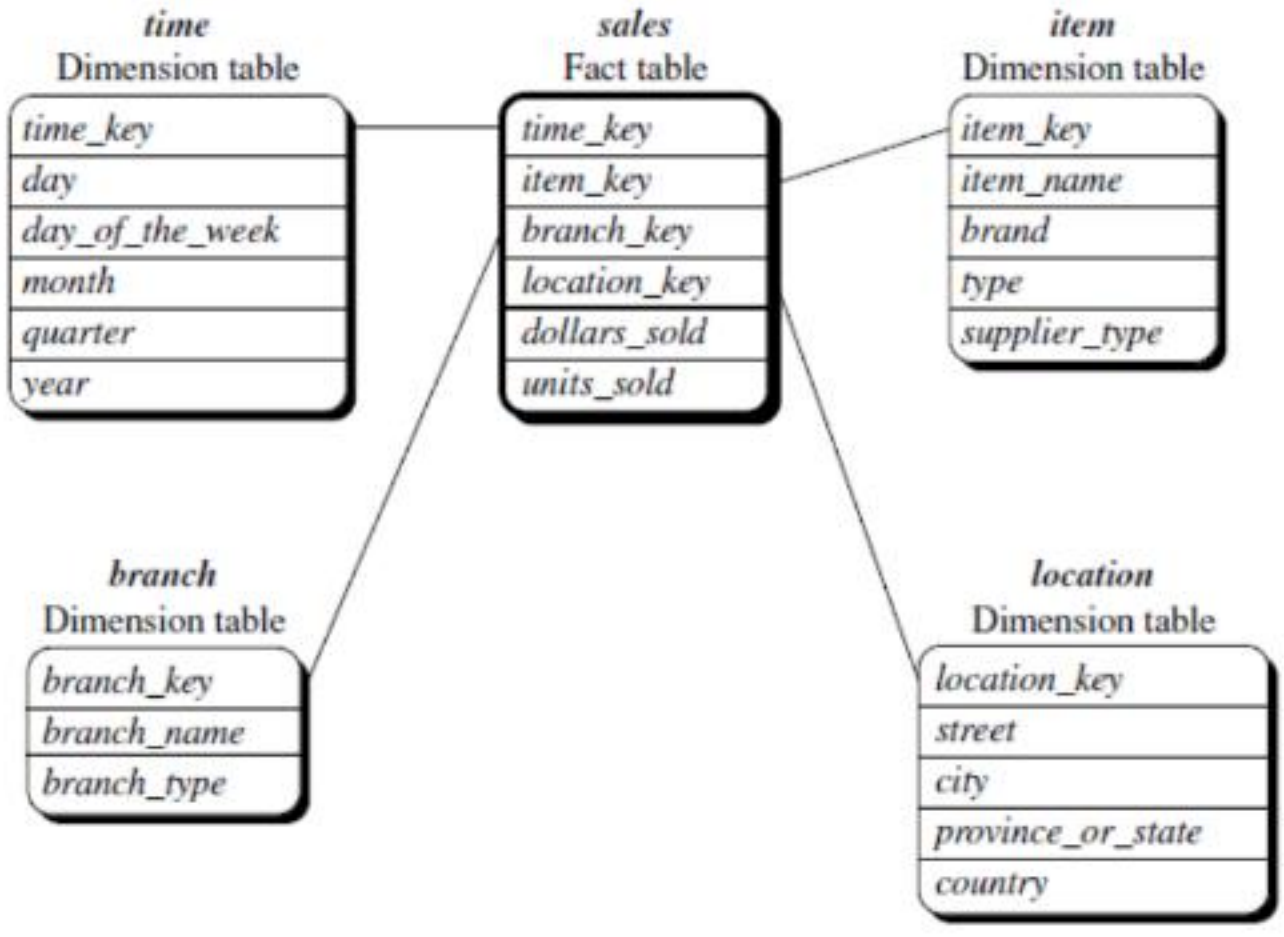


Figure 1.3: Star scheme

Snowflakes Schema: It contains

1. One fact table.
2. Any number of normalized dimension tables.

Example: All Electronics sales (Figure 1.4) contains

1. One Fact table (sales).
2. 6 normalized dimension tables (time, item, branch and location).

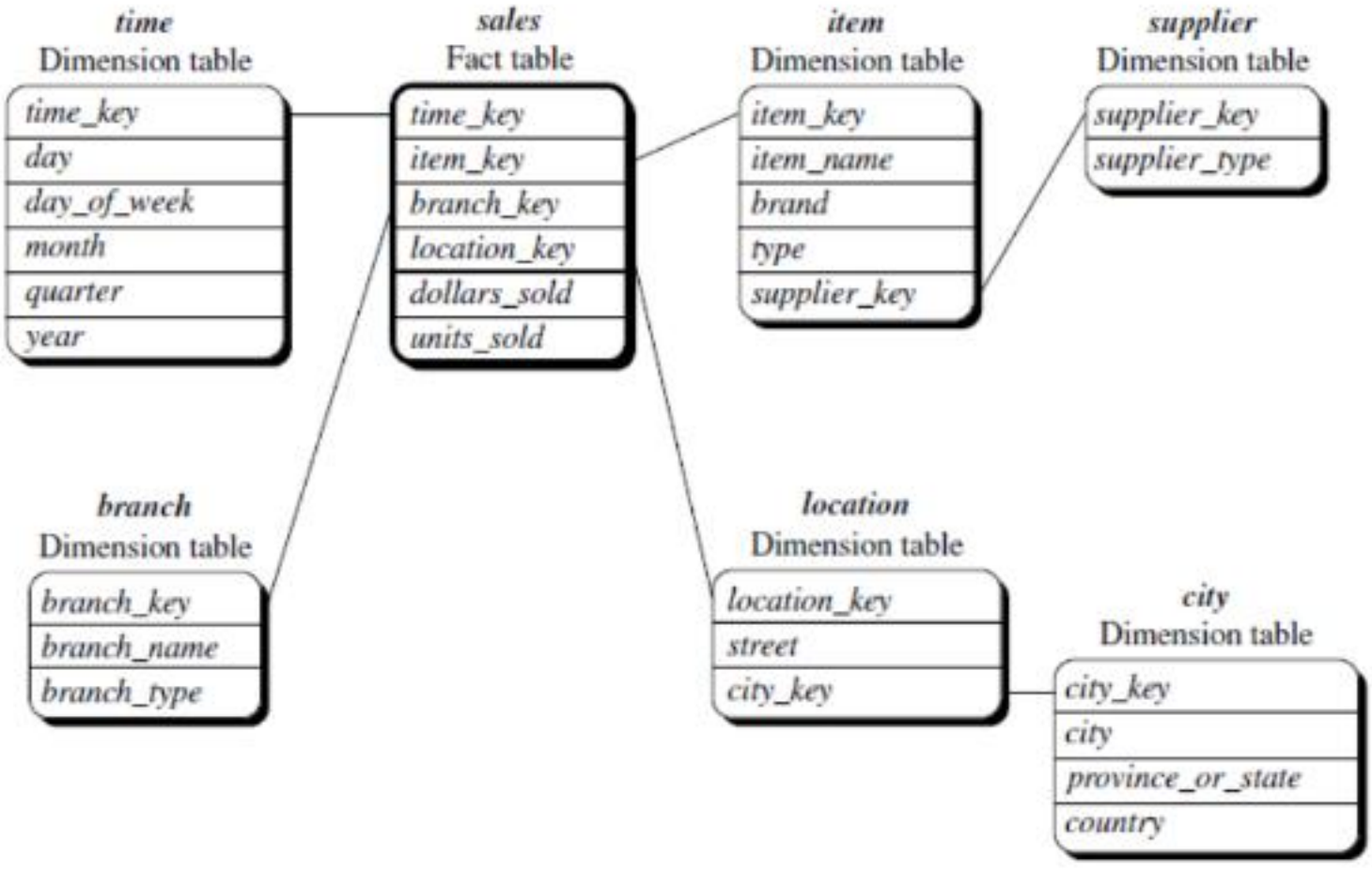


Figure 1.4: Snowflakes scheme

Fact Constellations Schemas: It contains

1. One fact table.
2. Any number of normalized dimension tables.

Example: All Electronics sales (Figure 1.5) contains

1. One Fact table (sales).
2. 6 normalized dimension tables (time, item, branch and location).

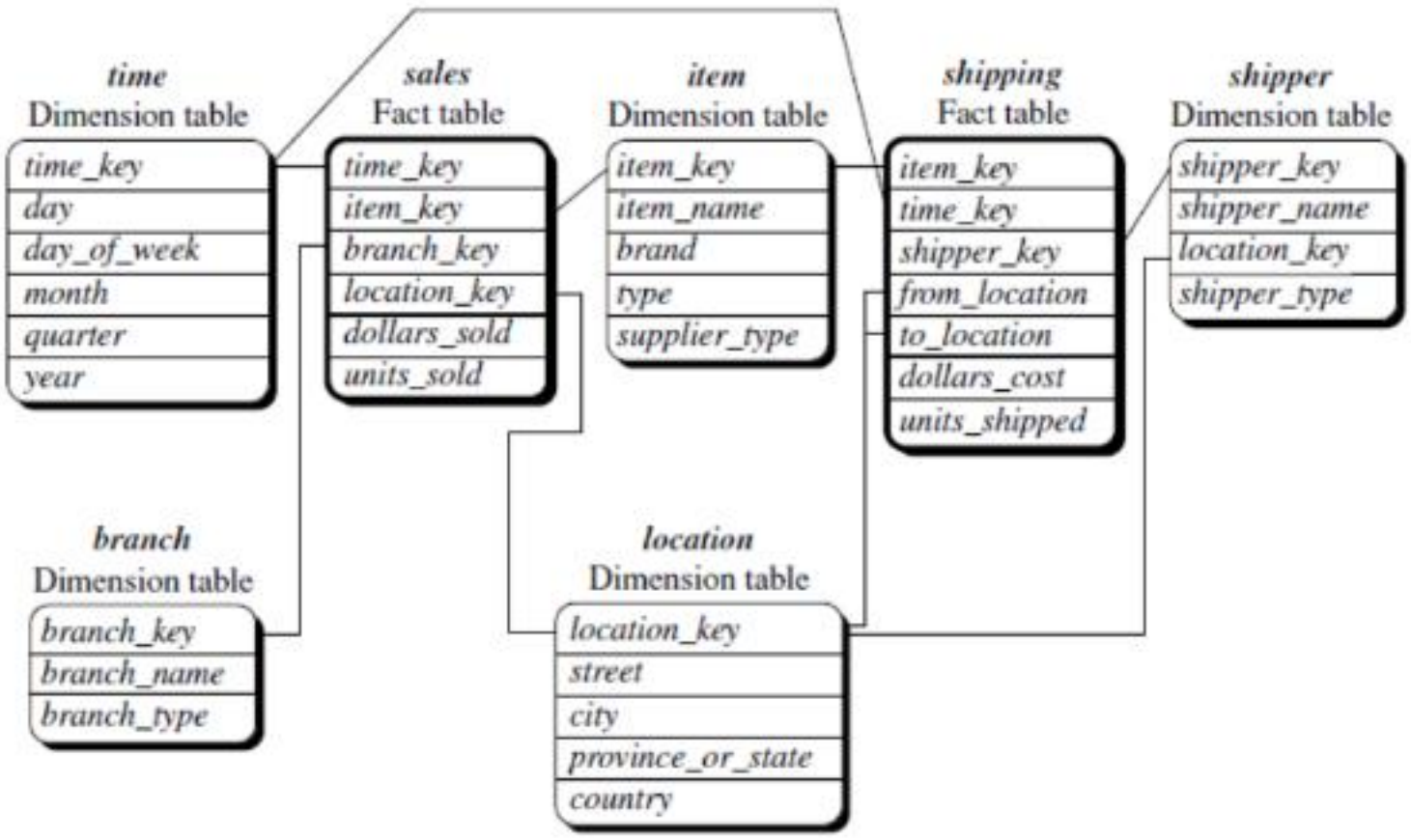


Figure 1.5: Fact Constellations scheme

OLAP Operations in the Multidimensional Data Model

OLAP operations (Figure 1.6) are:

- Roll-up
- Drill-down
- Slice
- Dice
- Pivot

Let us consider example, a company of Electronics Products. Data cube of company consists of 3 dimensions Location (aggregated with respect to city), Time (is aggregated with respect to quarters) and item (aggregated with respect to item types).

Roll Up: Consolidation is rolling up or adding data relationship with respect to one or more dimensions. For example, adding up all product sales to get total City data. The result of roll up operation performed on the central cube by climbing up the concept hierarchy for location. This hierarchy was defined as *the* total order street <city<province_or_state<country.

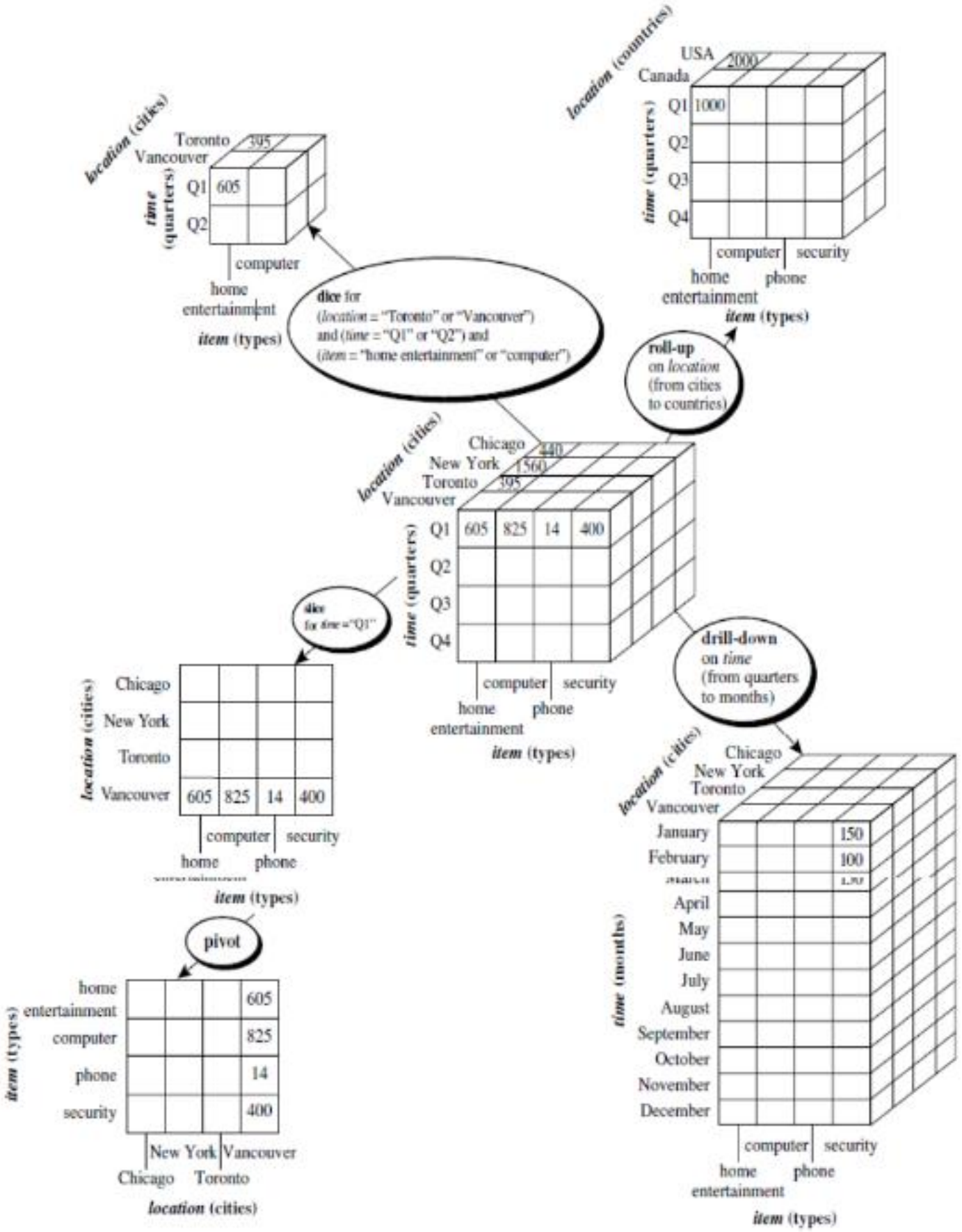


Figure 1.6: OLAP Operations.

Drill-down: Drill down is defined as changing the view of data to a greater level of detail. The result of drill down operations performed on the upper cube by stepping down a concept hierarchy for time defined as day<month<quarter<year.

Slicing: Slicing and dicing refers to the ability to look at a database from various viewpoints. Slice operation carry out selection with respect to one dimension of the given cube and produces a sub cube. The slice operation where the sales data are selected from the left cube for the dimension time using the criterion time = "Q1".

Dice: Dice operation carry out selection with respect to two or more dimensions of the given cube and produces a sub cube. The dice operation is performed on the left cube based on three dimension as Location, Time and Item, where the criteria is (location = "Torrento" or "" Vancouver") and (time = "Q1" or "Q2") and (time = "home entertainment" or "computer").

Pivot/Rotate: Pivot technique is used for visualization of data. This operation rotates the data axis to give another presentation of the data.

Other OLAP operations:

- **Drill across:** This technique is used when there is need to executes query which involves more than one fact table.
- **Drill through:** This technique uses relational SQL

The Process of Data Warehouse Design

The process of data warehouse design consists of the following steps:

1. First, select a business process to model. Depending on business model decide whether it should follow data warehouse or data mart model.
2. Select the business process grain
3. Select the dimensions that will apply to each fact table record.
4. Select the measures that will populate each fact table record.

A three-tier data warehousing architecture

The three tiers of three-tier data warehousing architecture are:

1. The last or bottom tier is a warehouse database server which mostly contains a relational database system. These tools and utilities like data extraction,

cleaning, and transformation, load and refresh functions update the data warehouse.

2. The middle tier is an OLAP server.
3. The top tier is a front-end client layer. It may consist reports, query, data mining and analysis tools.

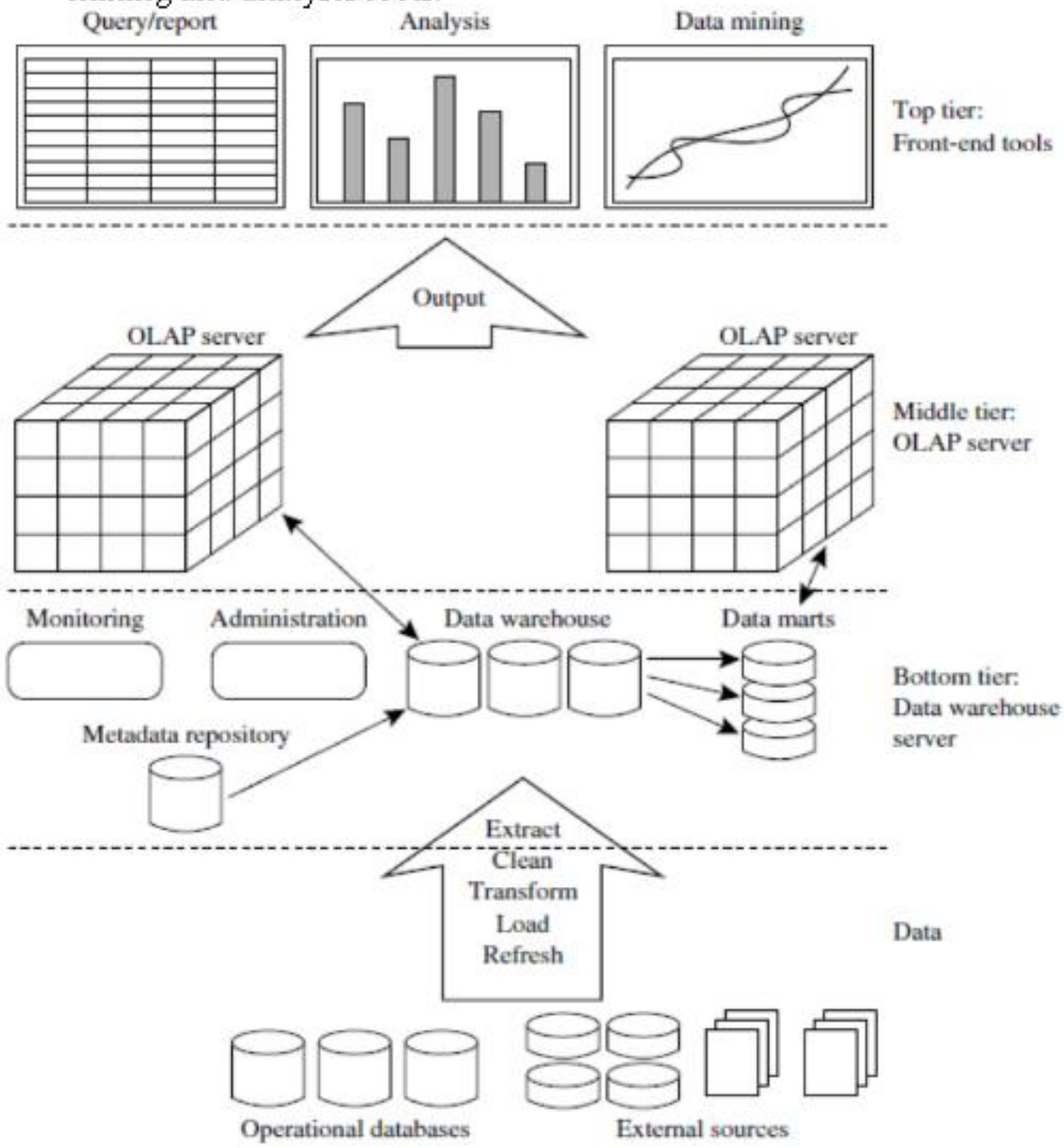


Figure 1.7: Three tier architecture.

Types of OLAP Servers: ROLAP versus MOLAP versus HOLAP.

Type	Strength	Risk
MOLAP	- Fast - Flexible Querying (within confine of Cube) - Write=Back capabilitie	- Size limit (Overall & within Dimension) - "Proprietary" Database - Cube Management - Drill-thru/around limitations
ROLAP	- Scalable Data Volumes - Scalable number of User - Query Managemen Layer provide OLAP "Open" Database	- Admin overhead for Query performance (Aggregate Management)
HOLAP	- Theoretically best of both worlds	- Expense and Admin overhead for multiple products - plus Risks listed above

Unit III
Measuring Data Similarity and Dissimilarity

Measuring Data Similarity and Dissimilarity,
Proximity Measures for Nominal Attributes and Binary Attributes, interval scaled;
Dissimilarity of Numeric Data: Minskowski Distance, Euclidean distance and Manhattan distance;
Proximity Measures for Categorical, Ordinal Attributes, Ratio scaled variables;
Dissimilarity for Attributes of Mixed Types
Cosine Similarity.

Data sets are made up of data objects. A data object is an entity. Data objects can be described by attributes. An attribute is a data field, represents feature of a data object.

Similarity:

- Numerical measure of how similar two data objects are.
- Value is higher when objects are more alike.
- Often falls in the range [0,1].

Dissimilarity:

- Numerical measure of how different two data objects are.
- Lower when objects are more alike.
- Minimum dissimilarity is often 0.
- Upper limit varies.

Proximity refers to a similarity or dissimilarity

Nominal Attributes

Nominal means something related to names. The values of a nominal attribute are symbols or names of things. Each value represents some kind of category, code, or state. The values do not have any meaningful order.
Example: marital status can take on the values single, married, divorced, and widowed.

Binary Attributes

A binary attribute is a nominal attribute with only two categories or states: 0 or 1, where 0 typically means that the attribute is absent and 1 means that it is present.

Ordinal Attributes

An ordinal attribute is an attribute with possible values that have a meaningful but the magnitude between successive values is not known.
Example: Grade can take values A+, A, B+, B, C.

Ratio scaled variables are A ratio-scaled attribute is a numeric attribute with an inherent zero-point. The value will be ratio form. In addition, the values are ordered, and we can also compute the difference between values, as well as the mean, median, and mode.
Example: weight height ratio.

Measuring Data Similarity and Dissimilarity
Many data mining application need to find the similarity between objects like in clustering and outlier analysis. Proximity is similarity and dissimilarity measures.

Suppose that we have n objects that are described by p attributes. The objects are $x_1 = (x_{11}, x_{12}, \ldots, x_{1p})$, $x_2 = (x_{21}, x_{22}, \ldots, x_{2p})$, and so on, where x_{ij} is the value for object x_i of the jth attribute.

Data matrix: This structure stores the n data objects in the form of a relational table, or n-by-p matrix.

$$\begin{bmatrix} x_{11} & \cdots & x_{1f} & \cdots & x_{1p} \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ x_{i1} & \cdots & x_{if} & \cdots & x_{ip} \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ x_{n1} & \cdots & x_{nf} & \cdots & x_{np} \end{bmatrix}$$

A data matrix is made up of two entities, they rows (for objects) and columns (for attributes). Therefore, the data matrix is often called a **two-mode** matrix.

Dissimilarity matrix: This structure stores a collection of proximities that are available for all pairs of n objects.

$$\begin{bmatrix} 0 & & & & \\ d(2,1) & 0 & & & \\ d(3,1) & d(3,2) & 0 & & \\ \vdots & \vdots & \vdots & & \\ d(n,1) & d(n,2) & \cdots & \cdots & 0 \end{bmatrix}$$

where $d(i, j)$ is the measured dissimilarity between objects i and j. $d(i, j)$ is a non-negative number that is close to 0 when objects i and j are highly similar to each other, and becomes larger the more they differ. $d(i, i) = 0$ means the difference between an object and itself is always 0.

Measures of similarity can often be expressed as a function of measures of dissimilarity.

$$\text{sim}(i, j) = 1 - d(i, j).$$

The dissimilarity matrix contains one kind of entity (dissimilarities) and so is called a one-mode matrix. Many clustering and nearest-neighbor algorithms operate on a dissimilarity matrix.

Proximity Measures for Nominal Attributes and Binary Attributes, interval scaled
Proximity Measures for Nominal Attributes:
A nominal attribute can take on two or more states. Example, image is a nominal attribute that may have three states: red, green and blue.
The dissimilarity between two objects i and j can be computed based on the ratio of mismatches:

$$d(i, j) = \frac{p - m}{p}$$

Where m is number of similar attributes in objects i and j. p is total number of attributes.
Example 1:

Roll no	Marks	Grade
111	98	O
112	83	A
113	75	B

Find d(Rollno111, Rollno111), d(Rollno112, Rollno111) .

d(Rollno111, Rollno111) = $\frac{p-m}{p}$= (2-2)/2 = 0

d(Rollno112, Rollno111) = $\frac{p-m}{p}$= (2-0)/2=1

d(i, j) = 0 if both are similar.

d(i, j) = 0 if both are not similar.

Example 2: Suppose that we have the sample data the *object-identifier* and the attribute *test-1* are available, where *test-1* is nominal. Compute dissimilarity matrix.

Object Identifier	***test-1 (nominal)***
1	code A
2	code B
3	code C
4	code A

Dissimilarity matrix is :

$$\begin{bmatrix} 0 & & & \\ d(2,1) & 0 & & \\ d(3,1) & d(3,2) & 0 & \\ d(4,1) & d(4,2) & d(4,3) & 0 \end{bmatrix}$$

d(i, j) = 0 if objects i and j match, otherwise 1 if the objects differ. Hence we get,

$$\begin{bmatrix} 0 & & & \\ 1 & 0 & & \\ 1 & 1 & 0 & \\ 0 & 1 & 1 & 0 \end{bmatrix}$$

only d(4,1) objects are same, that is, both are code A so d(4,1) = 0 and all others are 1.

Proximity Measures for Binary Attributes

For binary attributes, dissimilarity and similarity measures for objects is described as symmetric or asymmetric. A binary attribute has only two states 0 and 1, 0 represents absent

and 1 represents present. For example, if a person comes for diabetes test, then in binary attributes result will be either 0 or 1. 0 represents no diabetes and 1 represents diabetes.

Table: Binary attributes

Object *i* \ Object *j*	1	0	sum
1	q	r	$q+r$
0	s	t	$s+t$
sum	$q+s$	$r+t$	p

For symmetric binary attributes 2 states are equally important. Dissimilarity based on symmetric binary attributes is called **symmetric binary dissimilarity**. If objects i and j are described by symmetric binary attributes, then the dissimilarity between i and j is

$$d(i, j) = \frac{r + s}{q + r + s + t}$$

For asymmetric binary attributes, the two states are not equally important. The dissimilarity based on single important attribute is called **asymmetric binary dissimilarity**.

$$d(i, j) = \frac{r + s}{q + r + s}$$

The asymmetric binary similarity between the objects i and j can be computed as (also called as Jaccard coefficient)

$$sim_{Jaccard}(i, j) = \frac{q}{q + r + s}$$

Summary of proximity measure of binary attributes:

A contingency table for binary data

Object *i* \ Object *j*	1	0	sum
1	q	r	$q+r$
0	s	t	$s+t$
sum	$q+s$	$r+t$	p

Distance measure for symmetric binary variables:

$$d(i,j) = \frac{r+s}{q+r+s+t}$$

Distance measure for asymmetric binary variables:

$$d(i,j) = \frac{r+s}{q+r+s}$$

Jaccard coefficient (*similarity* measure for *asymmetric* binary variables):

$$sim_{Jaccard}(i,j) = \frac{q}{q+r+s}$$

Example 1:

Fruit	Shape sphere	Sweet	Sour	Crunchy
object i = apple	Yes	Yes	Yes	Yes
object j = banana	No	Yes	No	No

Apple = (1, 1, 1, 1)
Banana = (0, 1, 0, 0)
Let q = number of variables that are positive for both objects.
r = number of variables that are positive for object i and negative for object j.
s = number of variables that are negative for object i and positive for object j.
t = number of variables that are negative for both objects.
p = total number of objects = q+r+s+t

Object i	Object j		
		Yes	No
	Yes	q	r
	No	s	t

Now,
q = 1, r = 3, s = 0, t =0
Therefore, p = 1+3+0+0 = 4
Distance measure for symmetric binary variable:
$d(i, j) = \frac{r+s}{q+r+s+t}$
= ¾ = 0.75
Distance measure for asymmetric binary variable:
$d(i,j) = \frac{r+s}{q+r+s}$
= ¾ = 0.75
Jaccard Coefficient (similarity measure for asymmetric binary variable):
$d(i, j) = \frac{q}{q+r+s}$
= ¼ =0.25

Proximity Measures for interval scaled

Interval-Scaled

- Measured on a scale of equal-sized units.
- Values have order.

$$d=\|p-q\|$$

where d is dissimilarity, p and q are attributes.

Dissimilarity of Numeric Data: Minskowski Distance, Euclidean distance and Manhattan distance

Euclidean distance: Let $i = (xi1, xi2, : : : , xip)$ and $j = (xj1, xj2, : : : , xjp)$ be two objects described by p numeric attributes. The Euclidean distance between objects i and j is defined as

$$d(i, j) = \sqrt{(x_{i1} - x_{j1})^2 + (x_{i2} - x_{j2})^2 + \cdots + (x_{ip} - x_{jp})^2}$$

Manhattan (or city block) distance:

$$d(i, j) = |x_{i1} - x_{j1}| + |x_{i2} - x_{j2}| + \cdots + |x_{ip} - x_{jp}|$$

Minkowski distance: is a generalization of the Euclidean and Manhattan distances. It is defined as

$$d(i, j) = \sqrt[h]{|x_{i1} - x_{j1}|^h + |x_{i2} - x_{j2}|^h + \cdots + |x_{ip} - x_{jp}|^h}$$

where h is a real number such that $h \geq 1$.

If $h = 1$ then it is Manhattan distance and $h = 2$ then it is Euclidean distance.

Example 1:
Let $x1 = (1, 2)$ and $x2 = (3, 5)$ represent two objects. Find Euclidean distance, Manhattan distance and Minkowski distance with h=2.

Euclidean distance = $\sqrt{(3-1)^2 + (5-2)^2}$

= $\sqrt{4+9}$ = 3.61.

Manhattan distance = |3-1|+|5-2| = 5.

Minkowski distance = $\sqrt{(3-1)^2 + (5-2)^2}$

= $\sqrt{4+9}$ = 3.61.

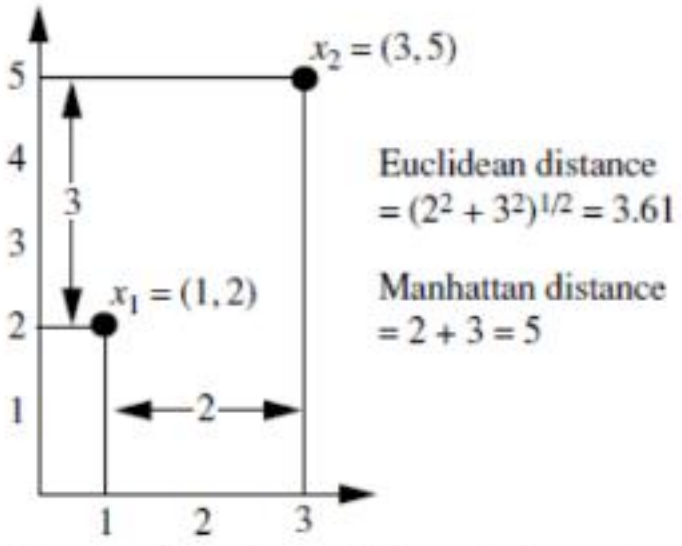


figure: Euclidean and Manhattan distance between two objects.

Example 2: Calculate Minkowski distance with h=3 for following data.

	Cost	Time	Weight	Incentive
Object x1	0	3	4	5
Object x2	7	6	3	-1

Minkowski distance = $\sqrt[3]{|0-7|^3 + |3-6|^3 + |4-3|^3 + |5-(-1)|^3}$

= $\sqrt[3]{343 + 27 + 1 + 216}$

Proximity Measures for Ordinal Attributes

Ordinal attributes act similar to numerical attributed when computing dissimilarity between objects. Suppose f is an attribute from a set of ordinal attributes describing n objects. The dissimilarity computation with respect to f involves the following steps:

1. The value of f for the ith object is x_{if}, and f has M_f ordered states, representing the ranking 1, ..., M_f. Replace each x_{if} by its corresponding rank, $r_{if} \in \{1, \ldots, M_f\}$.
2. Since each ordinal attribute can have a different number of states, it is often necessary to map the range of each attribute onto [0.0, 1.0] so that each attribute has equal weight. We perform such data normalization by replacing the rank r_{if} of the ith object in the fth attribute by
$$z_{if} = \frac{r_{if} - 1}{M_f - 1}$$
3. Dissimilarity can then be computed using any of the distance measures for numeric (Minskowski Distance, Euclidean distance and Manhattan distance) attributes, using z_{if} to represent the f value for the ith object.

Example: Sample data with object identifier and test – 1 ordinal attribute is given below.

Object Identifier	test- I (ordinal)
1	excellent
2	fair
3	good
4	excellent

There are three states for test-1: fair, good, and excellent, that is, Mf = 3.
Step 1: Replace each value for *test-1* by its rank, the four objects are assigned the ranks 3, 1, 2, and 3, respectively. Hence ranka are 1 - fair, 2 – good and 3 – excellent.
Step 2: Normalize the ranking by mapping rank 1 to 0.0, rank 2 to 0.5, and rank 3 to 1.0.
Step 3: Use the Euclidean distance, which results in the following dissimilarity matrix:

$$\begin{bmatrix} 0 & & & \\ 1.0 & 0 & & \\ 0.5 & 0.5 & 0 & \\ 0 & 1.0 & 0.5 & 0 \end{bmatrix}$$

$d(2,1) = \sqrt{(1.0-0.0)^2} = 1$ this is dissimilarity measure between object 1 whose value is excellent (rank is 3 and normalized value in step 2 for rank 3 is 1.0) and object 2 whose value is fair(rank is 1 and normalized value in step 2 for rank 1is 0.0).
$d(4,2) = \sqrt{(1.0-1.0)^2} = 0$ this is dissimilarity measure between object 2whose value is excellent (rank is 3 and normalized value in step 2 for rank 3 is 1.0) and object 4 whose value is excellent (rank is 3 and normalized value in step 2 for rank 3 is 1.0).
This shows that if d(i, j) =0 for similar objects.

Dissimilarity for Attributes of Mixed Types
A more preferable approach is to process all attribute types together, performing a single analysis. One such technique combines the different attributes into a single dissimilarity matrix, bringing all of the meaningful attributes onto a common scale of the interval [0.0,1.0]. Suppose that the data set contains *p* attributes of mixed type. The dissimilarity d(i, j) between objects *i* and *j* is defined as

$$d(i,j) = \frac{\sum_{f=1}^{p} \delta_{ij}^{(f)} d_{ij}^{(f)}}{\sum_{f=1}^{p} \delta_{ij}^{(f)}},$$

If x_{if} or x_{jf} is missing or if $x_{if} = x_{if} = 0$ then $\delta_{ij}^{(f)} = 0$, otherwise, $\delta_{ij}^{(f)} = 1$
$d_{ij}^{(f)}$ is calculated by its type:

- If f is numeric: $d_{ij}^{(f)} = \frac{|x_{if} - x_{jf}|}{max_h x_{hf} - min_h x_{hf}}$, where h runs over all nonmissing objects for attribute f.
- If f is nominal or binary: $d_{ij}^{(f)} = 0$ if $x_{if} = x_{jf}$; otherwise, $d_{ij}^{(f)} = 1$.
- If f is ordinal: compute the ranks r_{if} and $z_{if} = \frac{r_{if} - 1}{M_f - 1}$, and treat z_{if} as numeric.

Cosine Similarity

Cosine similarity is a measure of similarity that is used to compare documents or files.

Cosine similarity is given by

$$\text{sim}(x, y) = \frac{x.y}{\|x\|\|y\|}$$

where $\|x\| = \sqrt{x_1^2 + x_2^2 + \ldots + x_p^2}$

and $\|y\| = \sqrt{y_1^2 + y_2^2 + \ldots + y_p^2}$

Example – Find cosine similarity between document 1 and document 2

Table : Term-Frequency Vector

Document	team	coach	hockey	baseball	soccer	penalty	score	win	loss	season
Document1	5	0	3	0	2	0	0	2	0	0
Document2	3	0	2	0	1	1	0	1	0	1

How similar are Document1 and Document2?

Let us say, Document1 as x and Document2 as y.

$$\text{sim}(x, y) = \frac{x.y}{\|x\|\|y\|}$$

$x.y = 5x3+0x0+3x2+0x0+2x1+0x1+0x0+2x1+0x0+0x1 = 25$

$\|x\| = \sqrt{5^2 + 0^2 + 3^2 + 0^2 + 2^2 + 0^2 + 0^2 + 2^2 + 0^2 + 0^2} = 6.48$

$\|y\| = \sqrt{3^2 + 0^2 + 2^2 + 0^2 + 1^2 + 1^2 + 0^2 + 1^2 + 0^2 + 1^2} = 4.12$

$\text{sim}(x, y) = 0.94$

UNIT – IV

Association Rules Mining

Market basket Analysis
Frequent item set
Closed item set
Association Rules
Apriori Algorithm
Generating Association Rules from Frequent Item sets
Improving the Efficiency of a-priori
Mining Frequent Item sets without Candidate Generation: FP Growth Algorithm
Mining Various Kinds of Association Rules: Mining multilevel association rules, constraint based association rule mining, Meta rule-Guided Mining of Association Rules.

Market Basket Analysis :

What is Market Basket Analysis?

- Market basket analysis is a modeling technique which is also called as affinity analysis, it helps identifying which items are likely to be purchased together.
- Let market-basket problem have some large number of items, e.g. "bread", "milk", etc. Customer buy the subset of items as per his need and marketer gets the information that which things customer has taken together. So the marketers use this information to put the items on different position.
- **For example:** If someone buys a packet of milk also tends to buy a loaf of bread at the same time.

  Milk => Bread
- Market basket analysis algorithm are straightforward; difficulties arise mainly in dealing with large amounts of transactional data, where after applying algorithm it may give rise to large number of rules which may be trivial in nature.

How is it used?

- Market basket analysis is used in deciding the location of items inside a store, for e.g. if a customer buys a packet of bread he is more likely to buy a packet of butter too, keeping the bread and butter next to each other in a store would result in customers getting tempted to buy one item with the other.
- The problem of large volume of trivial results can be overcome with the help of differential market basket analysis which enables in finding interesting results and eliminates the large volume.
- Using differential analysis it is possible to compare results between various stores, between customers in various demographic groups.
- Some special observations among the rules for e.g. if there is a rule which holds in one store but not in any other (or vice versa) then it may be really interesting to note that there is something special about that store in the way it has organized its item inside the store may be in a more lucrative way. These types of insights will improve company sales.

- Identification of sets of items purchases or events occurring in a sequence, something that may be of interest to direct marketers, criminologists and many others, this approach may be termed as Predictive market basket analysis.

Applications of Market Basket Analysis :

- Credit card transactions done by a customer may be analysed.
- Phone calling patterns may be analysed.
- Fraudulent Medical insurance claims can be identified.
- For a financial services company :
    - Analysis of credit and debit card purchases.
    - Analysis of cheque payments made.
    - Analysis of service/products taken e.g. a customer who has taken executive credit card is also likely to take personal loan of $5,000 or less.
- For a telecom operator :
    - Analysis of telephone calling patterns.
    - Analysis of value-added services taken together. Rather than considering services taken together at a point in time, it could be services taken over a period of, let's say, six months.
- Various ways can be used to apply market basket analysis :
    - Special combo offers may be offered to the customers on the products sold together.
    - Placement of items nearby inside a store which may results in customers getting tempted to buy one product with the other.
    - The layout of catalog of an e-commerce site may be defined.
    - Inventory may be managed based on product demands.

Support, Confidence, Frequent Itemsets, Closed Itemsets and Association Rules:

Assume $I = \{I1, I2, \ldots, In\}$ is an itemset and D is a set of database transactions. Each transaction T with its identifier TID is a nonempty itemset and T is subset of D. Let A be a set of items. A transaction T is said to contain A if A is subset of T. An association rule is written as

$$A \rightarrow B$$

where $A \subset I$, $B \subset I$, $A \neq \emptyset$, $B \neq \emptyset$ and $A \cap B = \emptyset$.

$$\text{Support } (A \rightarrow B) = P(A \cap B)$$
$$\text{Confidence } (A \rightarrow B) = P(B|A)$$

Rules that satisfy both a minimum support threshold (min_sup) and a minimum confidence threshold (min_conf) are called strong. By convention, support and confidence values are written in percentage.

Frequent Item sets :

- A set of items is itemset. The itemset which appear frequently in transaction dataset are frequent itemsets. Frequency or support count or count of itemset is number of transactions that contain itemset.
- An itemset X is frequent if X's support is no less than a minimum support threshold.
- An itemset that contains *k* items is a ***k*-itemset**. The set {*computer, antivirus software*} is a 2-itemset.
- If support of itemset is ≥ min_sup then it is a frequent itemset.

Closed Itemsets:

- An itemset is closed if none of its immediate supersets has the same support as the itemset.
- An itemset *X* is **closed** in a data set *D* if there exists no proper super-itemset *Y* such that *Y* has the same support count as *X* in *D*. An itemset *X* is a **closed frequent itemset** in set *D* if *X* is both closed and frequent in *D*. An itemset *X* is a **maximal frequent itemset** (or **max-itemset**) in a data set *D* if *X* is frequent, and there exists no super-itemset *Y* such that $X \subset Y$ and *Y* is frequent in *D*.

Association Rules:

- The items databases other repositories are considered for finding frequent patterns, associations, correlation, or casual structures.
- It searches for interesting relationships among items in a given data set by examining transactions, or shop carts, we can find which items are

Association Rule	Support	Confidence	Confidence %
2^3 => 5	2	2/2 = 1	100%
3^5 => 2	2	2/2 = 1	100%
2^5 => 3	2	2/3 = 0.66	66%
2 => 3^5	2	2/3 = 0.66	66%
3 => 2^5	2	2/3 = 0.66	66%
5 => 2^3	2	2/3 = 0.66	66%

If the minimum confidence threshold is 70% (Given), then only the first and second rules above are output, since these are the only ones generated that are strong.

Final rules are :

Rule1: 2^3 => 5 Rule 2: 3^5 => 2

Example 4.3.2 Find the frequent item sets in the following database of nine transactions, with a minimum support 50% and confidence 50%.

Transaction ID	Items Bought
2000	A,B,C
1000	A,C
4000	A,D
5000	B,E,F

Solution :

Step 1: Scan D for count of each candidate. The candidate list is {A,B,C,D,E,F} and find the support

C_1 =

Itemset	Support
{A}	3
{B}	2
{C}	2
{D}	1
{E}	1
{F}	1

Step 2 : Compare candidate support count with minimum support count (50%)

L_1=

Itemset	Support.
{A}	3
{B}	2
{C}	2

Step 3 : Generate candidate C_2 from L_1

C_2=

Itemset
{A,B}
{A,C}
{B,C}

Step 4 : Scan D for count of each candidate in C_2 and find the support.

C_2=

Itemset	Support
{A,B}	1
{A,C}	2
{B,C}	1

Step 5 : Compare candidate (C_2) support count with the minimum support count

L_2=

Itemset	Support
{A,C}	2

Step 6 : So data contain the frequent item 1 (AC).

Therefore the association rule that can be generated from L are as shown below with the support and confidence.

Association Rule	Support	Confidence	Confidence %
A -> C	2	2/3 = 0.66	66%
C -> A	2	2/2 = 1	100%

Minimum confidence threshold is 50% (Given), then both the rules are output, as the confidence is above 50%.

So final rules are :

Rule 1: A -> C

Rule 2: C -> A

Example 3: Consider the transaction database given below. Use Apriori algorithm with minimum support count 2. Generate the association rules along with its confidence.

TID	List of item_IDs
T100	I1,I2,I5
T200	I2,I4
T300	I2,I3
T400	I1,I2,I4
T500	I1,I3
T600	I2,I3
T700	I1,I3
T800	I1,I2,I3,I5
T900	I1,I2,I3

Soultion :

Step 1 : Scan transaction database D and find count for item-1 set which is the candidate. The candidate list is {I1,I2,I3,I4,I5} and find the support.

C_1 =

Itemset	Support
{I1}	6
{I2}	7
{I3}	6
{I4}	2
{I5}	2

Step 2 : Find out whether each candidate item is present in at least two transactions(As support count is given is 2).

L_1=

Itemset	Support.
{1}	6
{2}	7
{3}	6
{4}	2
{5}	2

Step 3 : Generate candidate C_2 from L_1 and find the support of 2-itemset..

C_2=

Itemset	Support
{1 2}	4
{1 3}	4
{1 4}	1
{1 5}	2
{2 3}	4
{2 4}	2
{2 5}	2
{3 4}	0
{3 5}	1
{4 5}	0

Step 4 : Compare candidate (C_2) generated in step 3 with the support count, and prune those itemsets which do not satisfy the minimum support count.

L_2=

Itemset	Support
{1 2}	4
{1 3}	4
{1 5}	2
{2 3}	4
{2 4}	2
{2 5}	2

Step 5 : Generate candidate C_3 from L_2

C_3=

Itemset
{1,2,3}
{1,2,5}
{1,2,4}

Step 6 : Scan D for count of each candidate in C_3 and find their support count.

C_3=

Itemset	Support
{1 2 3}	2
{1 2 5}	2
{1 2 4}	1

Step 7 : Compare candidate (C_3) support count with the minimum support count and prune those itemsets which do not satisfy the minimum support count.

L_3=

Itemset	Support
{1 2 3}	2
{1 2 5}	2

Step 8 : Frequent itemsets are {I1,I2,I3} and {I1,I2,I5}

Let us consider the frequent item set (I1,I2,I5).

Following are the association rule that can be generated shown below with the support and confidence.

Association Rule	Support	Confidence	Confidence %
I1^I2 => I5	2	2/4 = 1	50%
I1^I5 => I2	2	2/2 = 1	100%
I2^5 => I1	2	2/2 = 1	100%
I1=> I2^I5	2	2/6 = 0.66	33%
I2 => I1^I5	2	2/7 = 0.66	29%
I5=> I1^I2	2	2/2 = 0.66	100%

Suppose the minimum confidence threshold is 75% , then only the following rules will be considered as output, they are strong rules.

Rules	Confidence
I1^I5 => I2	100%
I2^5 => I1	100%
I5 => I1 ^ I2	100%

Example 4 Consider the following transactions :

TID	Items
01	1,3,4,6
02	2,3,5,7
03	1,2,3,5,8
04	2,5,9,10
05	1,4

Apply the Apriori with minimum support of 30% and minimum confidence of 75% and find large item set L.

Solution :

Step 1 : Scan the transaction Database D and find the count for item-1set which is the candidate. The candidate list is {1,2,3,4,5,6,7,8,9,10} and find the support.

$C_1 =$

Itemset	Support
{1}	3
{2}	3
{3}	3
{4}	2
{5}	3
{6}	1
{7}	1
{8}	1
{9}	1
{10}	1

Step 2 : Find out whether each candidate item is present in at least 30% of transaction As support count given is (i.e. 30%)

$L_1 =$

Itemset	Support.
{1}	3
{2}	3
{3}	3
{4}	2
{5}	3

Step 3 : Generate candidate C_2 from L_1 and find the support of 2-itemsets..

$C_2 =$

Itemset	Support
{1 2}	1
{1 3}	2
{1 4}	2
{1 5}	1
{2 3}	2
{2 4}	0
{2 5}	3

{3 4}	1
{3 5}	2

Step 4 : Compare candidate (C_2) generated in step 3 with the support count and prune those itemsets which do not satisfy the minimum support count.

L_2=

Itemset	Support
{1 3}	2
{1 4}	2
{2 3}	2
{2 5}	3

Step 5 : Generate candidate C_3 from L_2 and find the support.

C_3=

Itemset	Support
{1,2,3}	1
{2,3,5}	2
{1,3,4}	1

Step 6 : Compare candidate (C_3) support count with the minimum support.

L_3=

Itemset	Support
{2 3 5}	2

Therefore the database contains the frequent item set (2,3,5).

Following are the association rules that can be generated from L_3 are as shown below with the support and confidence.

Association Rule	Support	Confidence	Confidence %
2^3 => 5	2	2/2 = 1	100%
3^5 => 2	2	2/2 = 1	100%
2^5 => 3	2	2/3 = 0.66	66%
2 => 3^5	2	2/3 = 0.66	66%
3 => 2^5	2	2/3 = 0.66	66%
5 => 2^3	2	2/3 = 0.66	66%

Give minimum confidence threshold is 75%, so only the first and second rules above are output, since these are the only ones generated that are strong.

Final rules are :

Rule1: 2^3 => 5 Rule 2: 3^5 => 2

Example 5 A database has four transactions. Let min sup = 60% and min conf = 80%

TID	DATE	Items-bought
T100	10/15/99	{K,A,D,B}
T200	10/15/99	{D,A,C,E,B}
T300	10/19/99	{C,A,B,E}
T400	10/22/99	{B,A,D}

Find all frequent itemsets using apriori algorithm. List strong association rules (with supports S and confidence C).

Solution :

Step 1 : Scan D for count of each candidate. The candidate list is {A,B,C,D,E,K} and find the support

C_1 =

Itemset	Support
{A}	4
{B}	4
{C}	2
{D}	3
{E}	2
{K}	1

Step 2 : Compare candidate support count with minimum support count (60%)

L_1=

Itemset	Support.
{A}	4
{B}	4
{D}	3

Step 3 : Generate candidate C_2 from L_1

C_2=

Itemset
{A,B}
{A,D}
{B,D}

Step 4 : Scan D for count of each candidate in C_2 and find the support.

C_2=

Itemset	Support
{A,B}	4
{A,D}	3
{B,D}	3

Step 5 : Compare candidate (C_2) support count with the minimum support count

L_2=

Itemset	Support
{A,B}	4
{A,D}	3
{B,D}	3

Step 6: Generate candidate C_3 from L_2.

C_3=

Itemset
{A,B,D}

Step 7 : Scan D for count of each candidate in C_3.

C_3=

Itemset	Support
{A,B,D}	3

Step 8 : Compare candidate (C_3) support count with the minimum support count.

L_3=

Itemset	Support
{A,B,D}	3

Step 9 : So data contain the frequent itemset (A,B,D).

Therefore the association rule that can be generated from L are as shown below with the support and confidence.

Association Rule	Support	Confidence	Confidence %
A^B -> D	3	¾ = 0.75	75%
A^D - > B	3	3/3 = 1	100%
B^D -> A	3	3/3 = 1	100%
A => B^D	3	¾ = 0.75	75%
B =>A^D	3	¾ = 0.75	75%
D =>A^B	3	3/3 = 1	100%

If the Minimum confidence threshold is 80% (Given), then only the SECOND, THIRD and LAST rules above are output, since the only ones generated that are strong.

Example 6 Apply the apriori algorithm on the following data with Minimum support = 2

TID	List of item_IDs
T100	I1,I2,I4
T200	I1,I2,I5
T300	I1,I3,I5
T400	I2,I4
T500	I2,I3
T600	I1,I2,I3,I5
T700	I1,I3
T800	I1,I2,I3
T900	I2,I3
T1000	I3,I5

Soultion :

Step 1 : Scan D for count of each candidate. The candidate list is {I1,I2,I3,I4,I5} and find the support.

C_1 =

Itemset	Support
{I1}	6
{I2}	7
{I3}	7
{I4}	2
{I5}	4

Step 2 : Compare candidate support count with minimum support count (i.e. 2).

L_1=

Itemset	Support.
{1}	6
{2}	7
{3}	6
{4}	2

{5}	2

Step 3 : Generate candidate C_2 from L_1 and find the support.

C_2=

Itemset	Support
{1 2}	4
{1 3}	4
{1 4}	1
{1 5}	3
{2 3}	4
{2 4}	2
{2 5}	2
{3 4}	0
{3 5}	3
{4 5}	0

Step 4 : Compare candidate (C_2) support count with the minimum support count.

L_2=

Itemset	Support
{1 2}	4
{1 3}	4
{1 5}	3
{2 3}	4
{2 4}	2
{2 5}	2
{3 5}	3

Step 5 : Generate candidate C_3 from L_2

C_3=

Itemset
{1,2,3}
{1,2,5}
{1,2,4}
{1,3,5}
{2,3,5}

Step 6 : Scan D for count of each candidate in C_3.

C_3=

Itemset	Support
{1 2 3}	2

{1 2 5}	2
{1 2 4}	0
{1,3,5}	2
{2,3,5}	0

Step 7 : Compare candidate (C_3) support count with the minimum support count

L_3=

Itemset	Support
{1 2 3}	2
{1 2 5}	2
{1,3,5}	2

Step 8 : So the data contains the Frequent itemsets are {I1,I2,I3}, {I1,I2,I5} and {I1,I3,I5}.

Let us assume the data contains the frequent item set (I1,I2,I5) then are the association rule that can be generated shown below with the support and confidence.

Association Rule	Support	Confidence	Confidence %
I1^I2 => I5	2	2/4 = 1	50%
I1^I5 => I2	2	2/2 = 1	100%
I2^5 => I1	2	2/2 = 1	100%
I1=> I2^I5	2	2/6 = 0.66	33%
I2 => I1^I5	2	2/7 = 0.66	29%
I5=> I1^I2	2	2/2 = 0.66	100%

If the minimum confidence threshold is 70% , then only the SECOND, THIRD, LAST rules above are output, since these are the only ones generated that are strong. Similarly do for frequent itemset {I1,I2,I3} and {I1,I3,I5}.

Example 7 : A database has four transactions. Let minimum support and confidence be 50%.

TID	Items
100	1,3, 4
200	2,3,5
300	1,2,3,5
400	2,5
500	1,2,3
600	3,5

700	1,2,3,5
800	1,5
900	1,3

Solution :

Step 1 : Scan D for count of each candidate. The candidate list is {1,2,3,4,5} and find the support.

C_1 =

Itemset	Support
{1}	6
{2}	5
{3}	7
{4}	1
{5}	6

Step 2 : Compare candidate support count with minimum support count (i.e. 50%)

L_1=

Itemset	Support.
{1}	6
{2}	5
{3}	7
{5}	6

Step 3 : Generate candidate C_2 from L_1 and find the support.

C_2=

Itemset	Support
{1 2}	3
{1 3}	5
{1 5}	3
{2 3}	4
{2 5}	4
{3 5}	4

Step 4 : Compare candidate (C_2) support count with the minimum support count.

L_2=

Itemset	Support
{1 3}	5

So data contain the frequent item set (1,3)

Therefore the association rule that can be generated from L_2 are as shown below with the support and confidence.

Association Rule	Support	Confidence	Confidence %
1 => 3	5	5/6 = 0.83	83%
3 => 1	5	5/7 = 0.71	71%

Given the minimum confidence threshold is 50%, so both the rules are strong.

Final rules are :

Rule1: 1=> 3 Rule 2: 3 => 1

Example 4.3.8 : Consider the following transaction database :

TID	Items
01	A,B,C,D
02	A,B,C,D,E,G
03	A,C,G,H,K
04	B,C,D,E,K
05	D,E,F,H,L
06	A,B,C,D,L
07	B,I,E,K,L
08	A,B,D,E,K
09	A,E,F,H,L
10	B,C,D,F

Apply the Apriori algorithm with minimum support of 30% and minimum confidence of 70%, and find all the association rules in the data set.

Solution :

Step 1: Generate the single item set:

Itemset	Support
{A}	6
{B}	7
{C}	6
{D}	7
{E}	6
{F}	3
{G}	2
{H}	3

{I}	1
{K}	4
{L}	4

Itemset above 30% support	
{A}	6
{B}	7
{C}	6
{D}	7
{E}	6
{F}	3
{G}	3
{H}	3
{K}	4
{L}	4

Step 2 : Generate 2 item set :

Itemset	Support.
AB	4
AC	4
AD	4
AE	3
AF	1
AH	2
AK	2
AL	2
BC	5
BD	6
BE	4
BF	1
BH	0
BK	3
BL	2
CD	5
CE	2
CF	1
CH	1
CK	2
CL	1
DE	4
DF	2

Itemset above 30 %Support.	
AB	4
AC	4
AD	4
AE	3
BC	5
BD	6
BE	4
BK	3
CD	5
DE	4
EK	3
EL	3

DH	1
DK	2
DL	2
EF	2
EH	2
EK	3
EL	3
FH	2
FK	0
FL	2
HK	1
HL	2
KL	1

Step 3 : Generate 3 item set :

Item sets of 3 items

Itemset	Support
ABC	3
ABD	4
ABE	2
ABK	1
ACD	3
ACE	1
ADE	2
AEK	1
AEL	1
BCD	5
BCE	2
BCK	1
BDE	3
BDK	2
BEK	2
BEL	1
CDE	2
DEK	2
DEL	1

Itemset above 30% support

Itemset	Support
ABC	3
ABD	4
ACD	3

BCD	5
BDE	3

Step 4 : Generate 4 item set

Itemset	**Support**
ABCD	3
ABDE	2
BCDE	2

Therefore ABCD is the large item set with minimum support 30%.
Following rules generated.

Association Rule	Confidence	Confidence %
A -> BCD	3/6 = 0.5	50%
B -> ACD	3/7 = 0.43	43%
C -> ABD	3/6 = 0.5	50%
D -> ABC	3/7 = 0.43	43%
AB -> CD	¾ = 0.75	75%
BC -> AD	3/5 = 0.6	60%
CD -> AB	3/5 = 0.6	60%
AC -> BD	¾ = 0.75	75%
AD -> BC	¾ = 0.75	75%
BCD -> A	3/5 = 0.6	60%
ACD -> B	3/3 = 1	100%
ABD -> C	¾ = 0.75	75%
ABC -> D	3/3 = 1	100%

From the above rules generated, only the rules having greater than 70% are considered as final rules. So final rules are,

AB -> CD
AC -> BD
AD -> BC
ACD -> B
ABD -> C
ABC -> D

Improving the Efficiency of Apriori :

There are many variations of Apriori algorithm that have been proposed to improve the efficiency, few of them are given as :

- **Hash-based item set counting :** The item sets can be hashed into corresponding buckets, For a particular iteration a k-item set can be generated and hashed into their respective bucket and increase the bucket count, the bucket with a count lesser than the support should not be considered as a candidate set.
- **Transaction reduction :** A transaction that does not contain k- frequent item set will never have k+1 frequent itemset, such a transaction should be reduced from future scans.
- **Partitioning :** In this technique only two database scans are needed to mine the frequent item sets. The algorithm has two phases, in the first phase, the transaction database is divided into non overlapping partitions. The minimum support count of a partition is min support X number of transactions in that partition. Local frequent items set are found out in each partition. The local frequent items sets may or may not be frequent with respect to the entire database however a frequent item set from database has to be frequent in at least one of the partitions. All the frequent item sets with respect to each partition forms the global candidate item sets. In the second phase of the algorithm, a second scan of database for actual support of each item is found, these are global frequent item sets.
- **Sampling :** Rather than finding the frequent item sets in the entire database D, a subset of transactions are picked up and searched for frequent item sets. A lower threshold of minimum support is considered as this reduces the possibility of missing the actual frequent item set due to a higher support count.
- **Dynamic item set counting :** In this the database is partitioned into blocks and is marked by start points. It maintains a count-so-far, if this count-so-far crosses minimum support, the item set is added to the frequent item set collection which can be further used to generate longer candidate item set.

FP-Growth Algorithm (A Pattern Growth Approach for Mining Frequent Itemsets) :

Definition of FP-tree :

An FP-tree structure which consists of :

- One root labeled as "null".
- A set of item prefix sub-trees with each node formed by three fields : item-name, count, node-link.
- A frequent-item header table with two fields for each entry : item name, head of node-link.
- It contains the complete information for frequent pattern mining.
- The size of the FP-tree is bounded by size of the database, but due to frequent items sharing, the size of the tree is usually much smaller than its original database.
- High compaction is achieved by placing more frequently items closer to the root (being thus more likely to be shared).
- The FP-Tree contains everything from the database we need to know for mining frequent.

Patterns :

- The size of the FP-tree is ≤ Occurrence of frequent patterns in database.
- This approach is very efficient due to :
    - Compression of a large database into a smaller data structure.
    - It is a fragment pattern growth mining method or simply FP-growth.
    - It adopts a divide-and-conquer strategy.
- The database of frequent items is compressed into a FP-Tree, and the association information of items is preserved.
- Then mine each such database separately.

FP-Tree Algorithm :

FP-Tree construction algorithm given by Jiawei Han et al. :

Algorithm : Fp. Growth, Mine frequent item sets using an FP-tree by pattern fragment growth.

Input :

- D, a transaction database.
- min_sup, the minimum support count threshold.

Output : The complete set of frequent patterns.

Method :

1. A FP tree is constructed in the following steps :
    a. Scan the transaction database D once, Collect F, the set of frequent items, and their support count. Sort F by support count is descending order as L, the list of frequent items.
    b. Create the root of an FP tree, and label it as "null". For each transaction Trans D do the following.
    Select and sort the frequent items in Trans according to the order of L. Let the sorted frequent item list in Trans be [p|P], where p is the first element and P is the remaining list. Call insert_tree([p|P].T),which is performed as follows. If T has a child N such that N_item.name = p.item.name, then increment N's count by 1; else create a new node N, and let its count be 1, its parent link be linked to T, and its node-link to the nodes with the same item.name via the node-link structure. If P is nonempty, call insert_tree(P, N) recursively.

Analysis :

- Two scans of the DB are necessary. The first collects the set of frequent items and the second constructs the FP-tree.
- The cost of inserting a transaction Trans into the FP-tree is O(|Trans|), where |Trans| is the number of frequent items in Trans.

FP-Growth Algorithm given by Jiawei Han et al :

- FP-Growth allows frequent itemset discovery without candidate itemset generation.
- Once the FP tree is generated, it is mined by calling FP-growth(FP_tree, null).

Procedure FP_growth (Tree, α) :

1. if Tree contains a single path P then
2. for each combinaon (denoted as β) of the nodes in the path P.
3. generate paern β ∪ α with support_count = minimum support count of nodes in β;
4. else for each α_i in the header of Tree {
5. generate paern $\beta = \alpha_i \cup \alpha$ with support_count = α_i. Support_count;
6. construct β's condional pattern base and then β's condional FP_tree Treeβ;
7. if Tree β≠ϕ then
8. call FP_growth (Treeβ, β);}

4.4.3 FP Tree Size :

- Many transactions share items due to which the size of the FP-Tree can have a smaller size compared to uncompressed data.
- **Best case scenario :** All transactions have the same set of items which results in a single path in the FP tree.
- **Worst case scenario :** Every transaction has a distinct set of items, i.e. no common items.
  - FP-tree size is as large as the original data.
  - FP-tree storage is also higher, it needs to store the pointers between the nodes and the counter.
- FP-Tree is dependent on the other of the items. Ordering of items by decreasing support will not always result in a smaller FP-Tree size (it's heuristic).

4.4.4. Examples on FP Tree :

4.4.1 : Transactions consist of a set of items I={a,b,c,.....}, min support = 3

TID	Items Bought
1	f,a,c,d,g,i,m,p
2	a,b,c,f,l,m,o
3	b,f,h,j,o
4	b,c,k,s,p
5	a,f,c,e,l,p,m,n

Solution :

Step 1 : Find the minimum support of each item.

Item	Support
a	3
b	3
c	4
d	1
e	1
f	4
g	1
h	1
i	1
j	1
k	1
l	2
m	3
n	1
o	2
P	3

Consider items with min support = 3 (given)

Item	Support
a	3
b	3
c	4
f	4
m	3
p	3

Step 2 : Order all items in itemset in frequency descending order (min support = 3)

(Note : Consider only items with min support = 3)

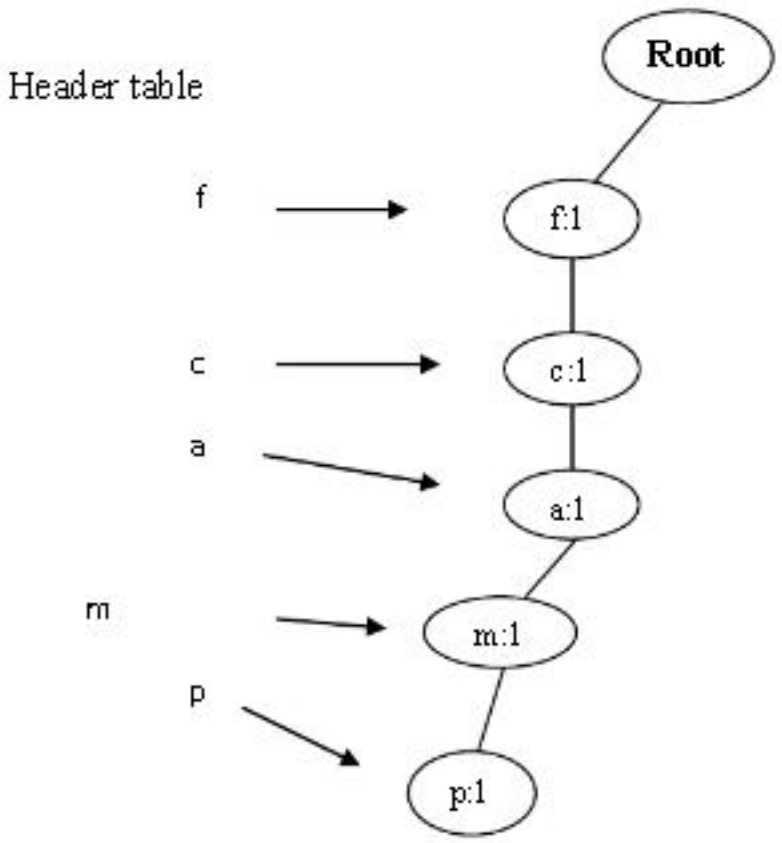
TID	Items Bought	(Ordered frequent items)
1	f,a,c,d,g,I,m,p	f,c,a,m,p
2	a,b,c,f,l,m,o	f,c,a,b,m
3	b,f,h,j,o	f,b
4	b,c,k,s,p	c,b,p
5	a,f,c,e,l,p,m,n	f,c,a,m,p

(f:4,c:4,a:3,b:3,m:3,p:3)

Step 3 : FP Tree construction

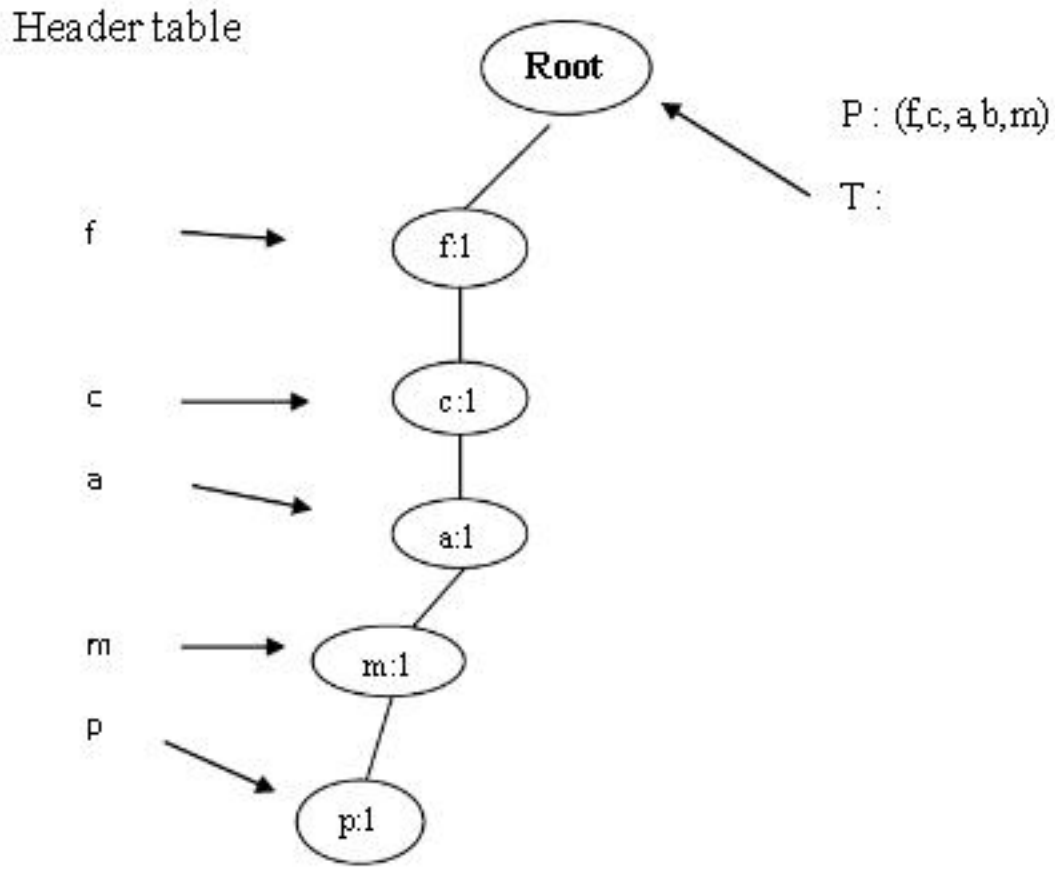
Originally Empty

Step 4 : Insert the first transaction (f,c,a,m,p)

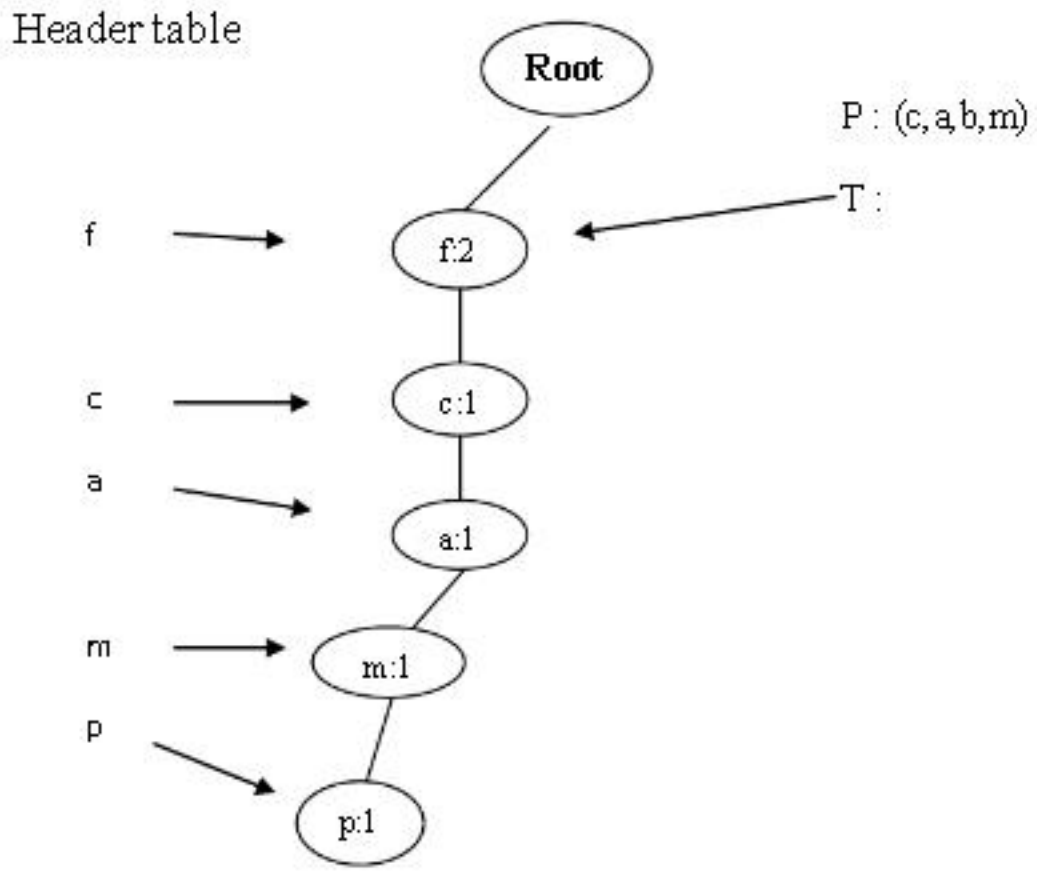


Step 5 : Start the insertion of Second transaction (f,c,a,b,m).

i. The transaction T is pointing to the real node.

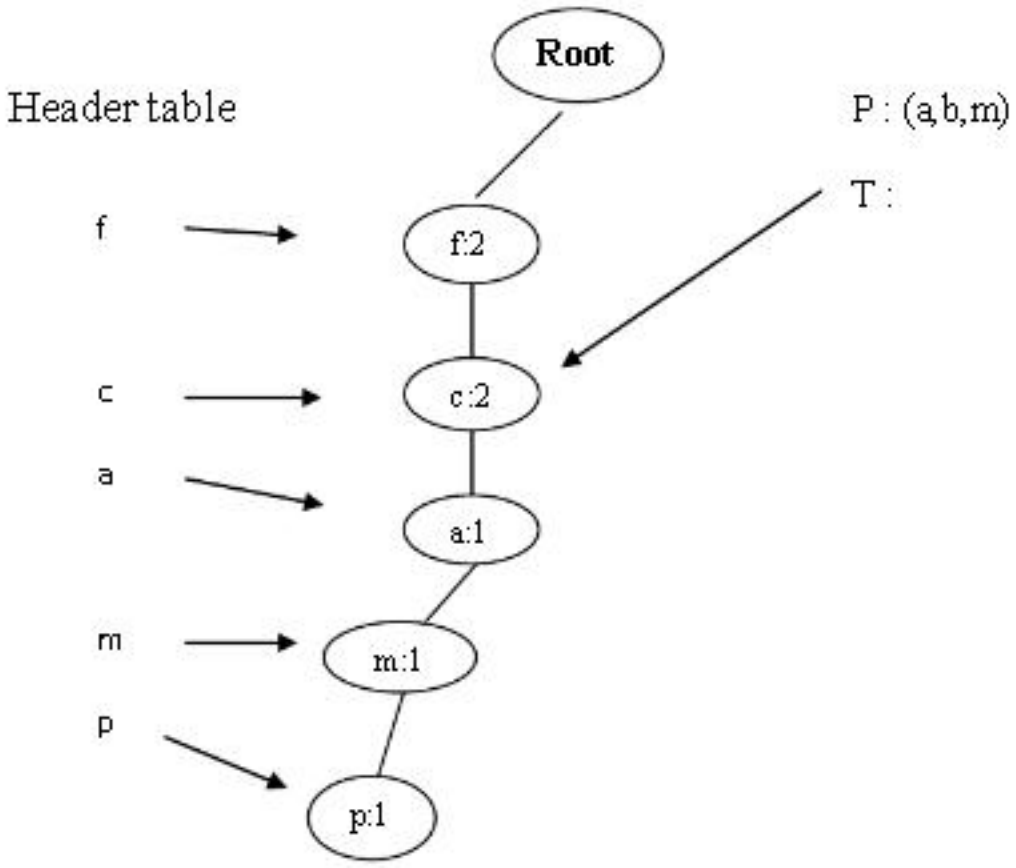


ii. Consider the first item in the second transaction i.e. f and add it in the tree.

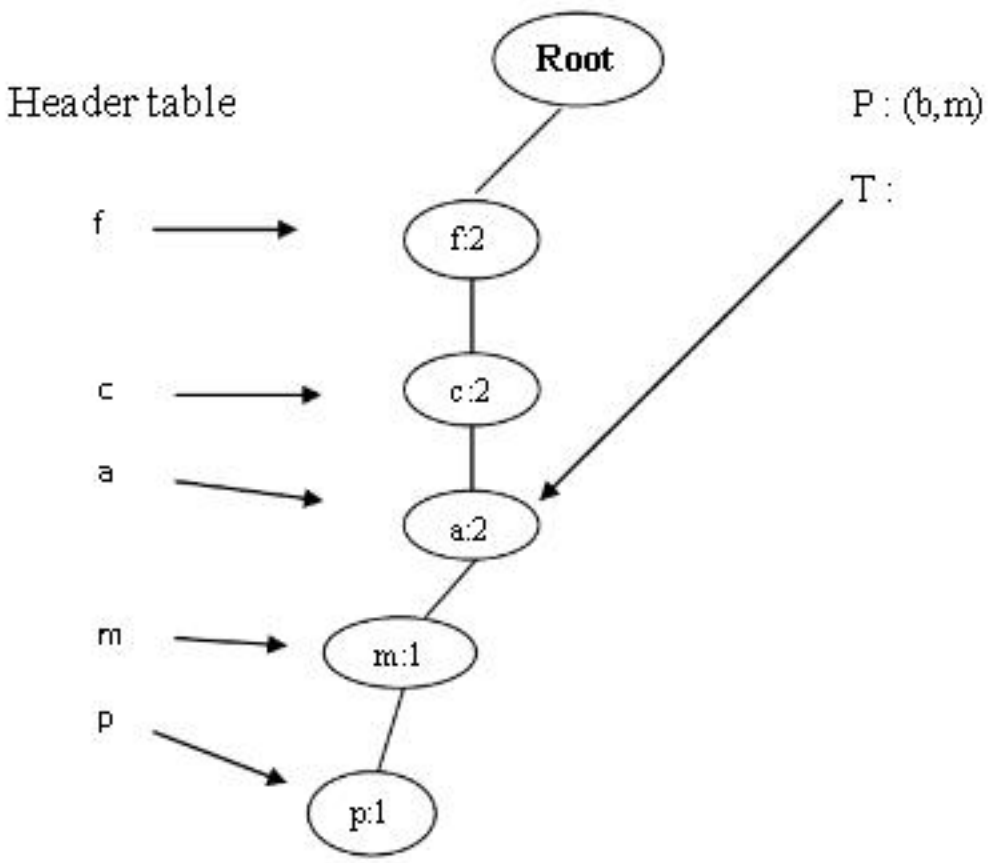


After this step we get f:2, finished adding f in the above tree.

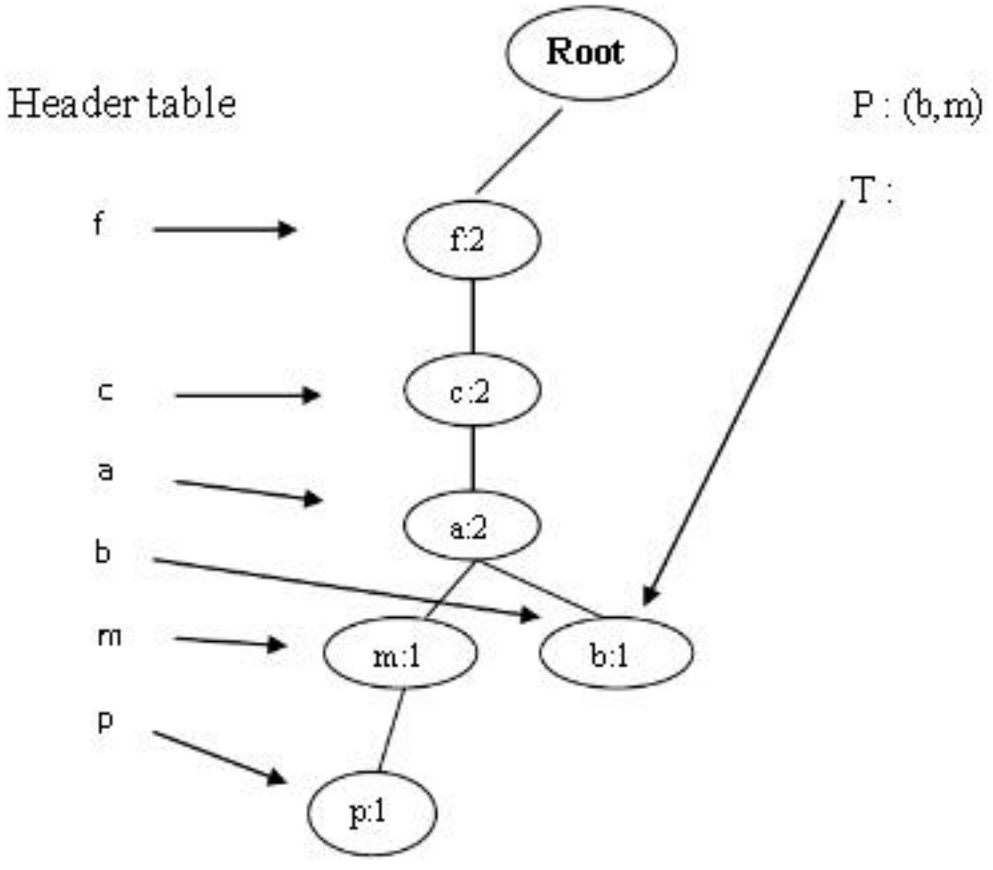
iii. Now consider the second item in the above transaction i.e. c.



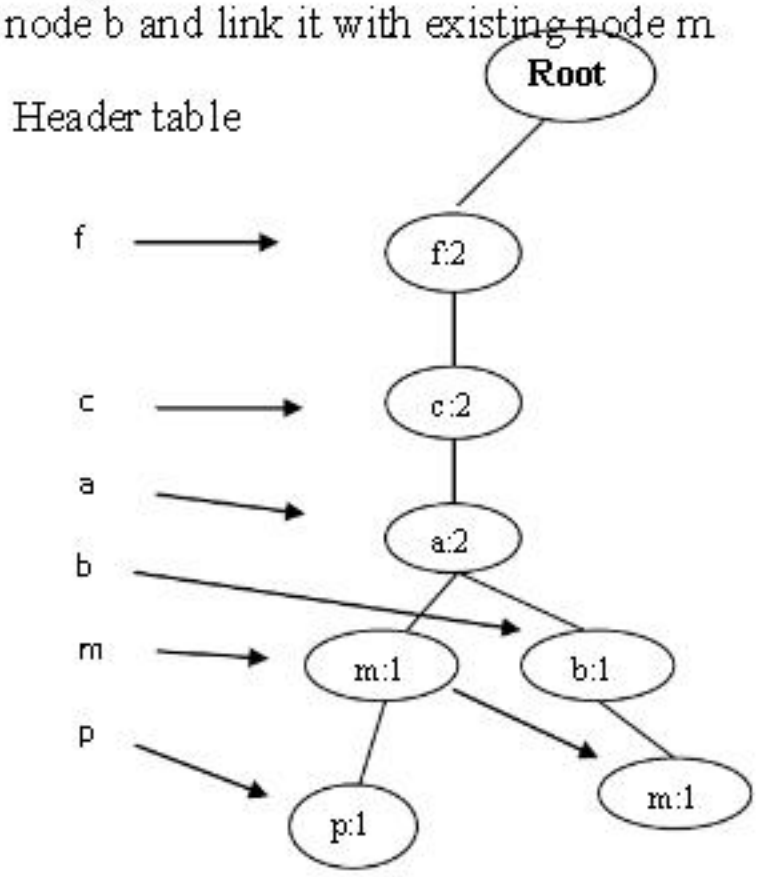
iv. Similarly consider the next item a.



v. Since we do not have a node b, we create one node for b below the node a(note : to maintain the path).

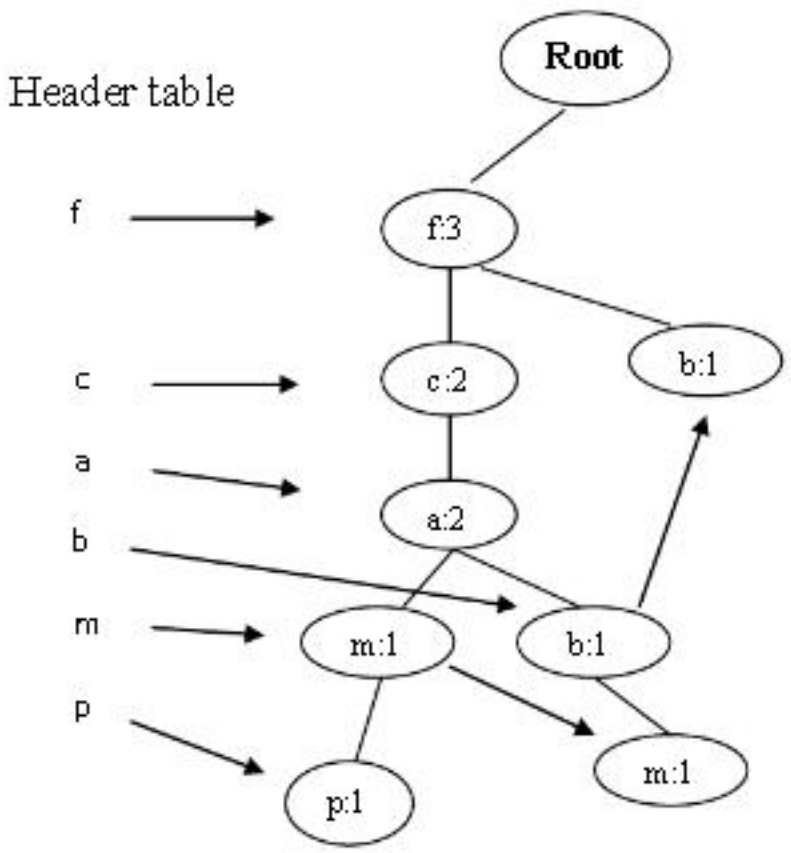


vi. Now only m of second transaction is left. Though a node m is already exists still we can't increase its count of the existing node m as we need to represent the second transaction in FP tree, so add new node m below node b and link it with existing node m

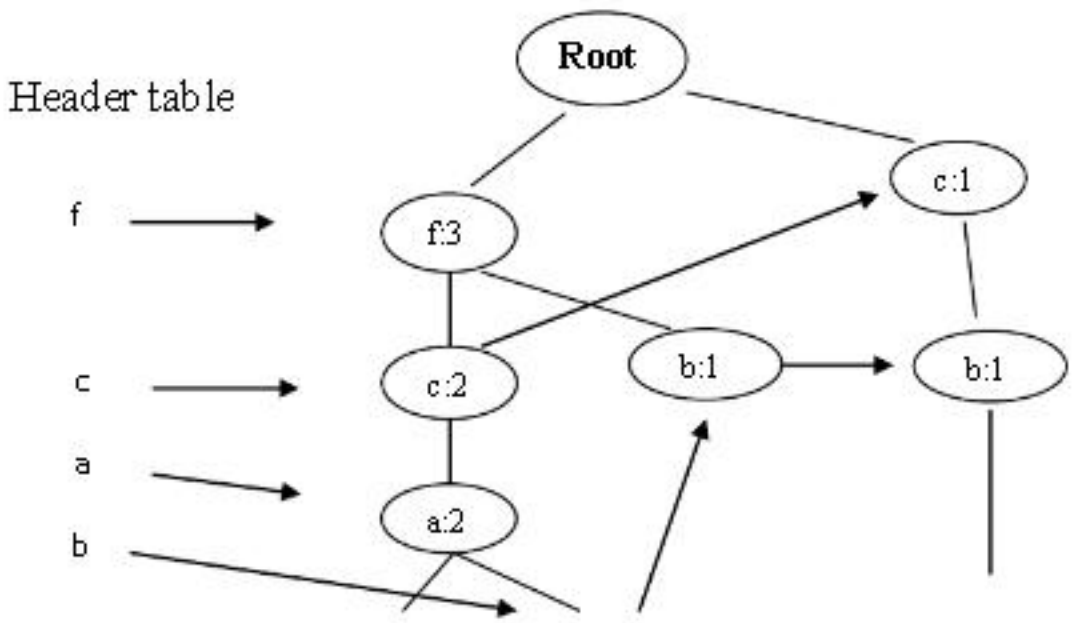


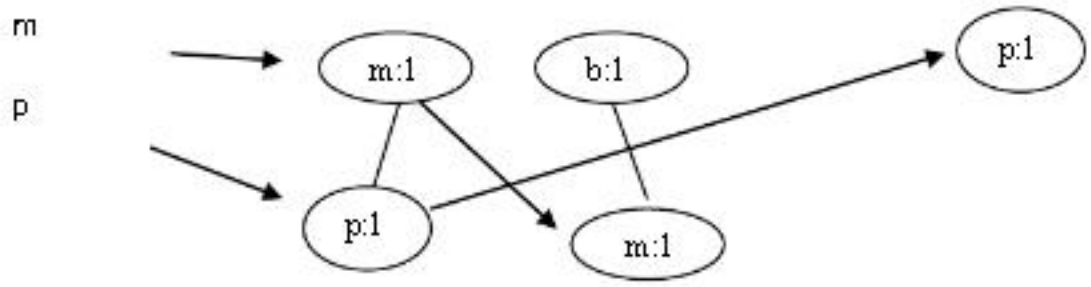
Second transaction is complete.

Step 6 : Similarly insert the third transaction(f,b) as explained in step 5.
So after the insertion of third transaction (f,b).

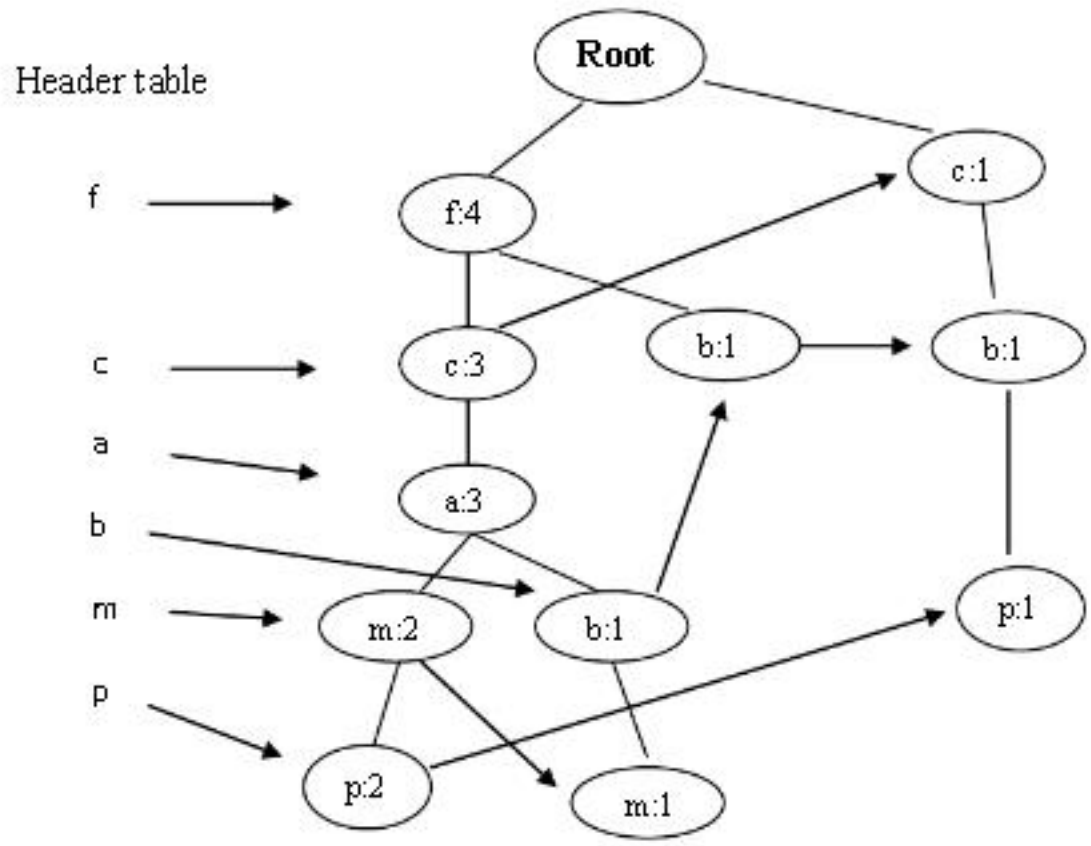


Step 7 : After the insertion of fourth transaction (c,b,p)





Step 8 : After the insertion of fifth transaction (f,c,a,m,p)



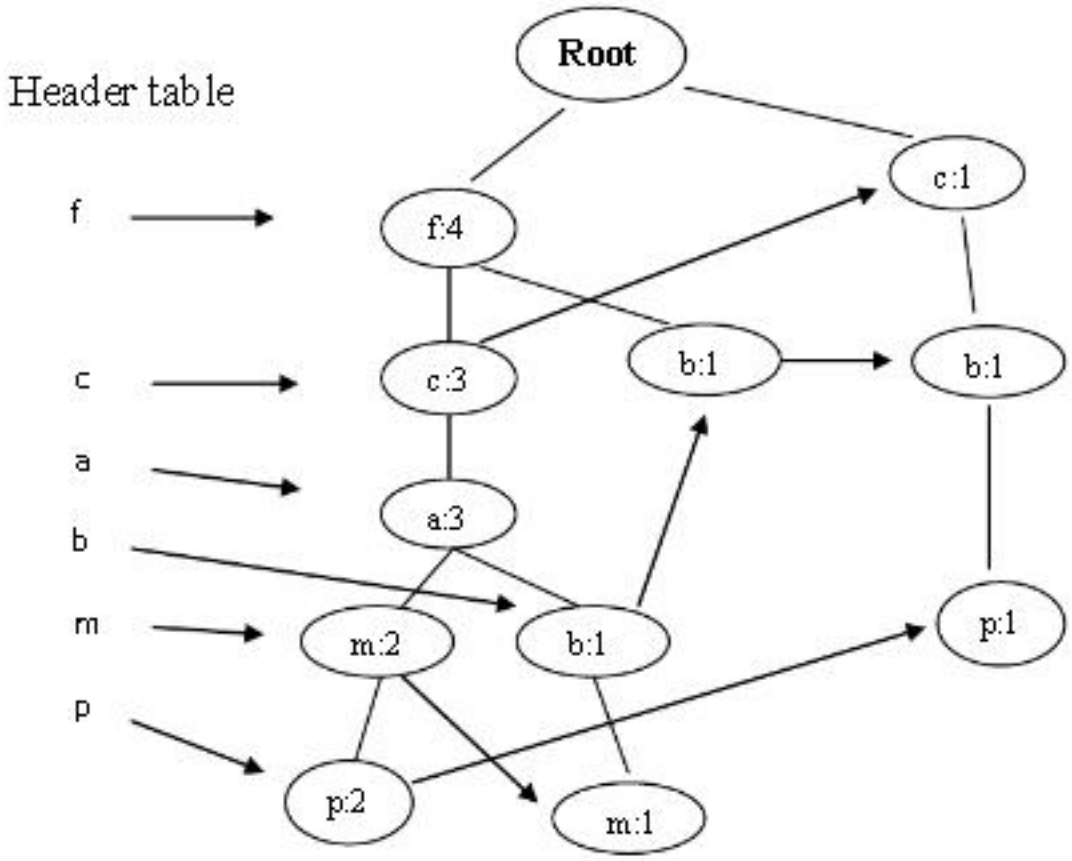
This is the final FP-Tree.

4.4.5 Mining Frequent Patterns from FP-Tree.

- **General idea (divide and conquer) :**
  - Use the FP Tree and recursively grow frequent pattern path.
- **Method :**
  - For each item, conditional pattern-base is constructed, and then its's conditional FP-tree.
  - On each newly created conditional FP-tree, repeat the process.
  - The process is repeated until the resulting FP-tree is empty, or it has only a single path (All the combinations of sub paths will be generated through that single path, each of which is a frequent pattern).

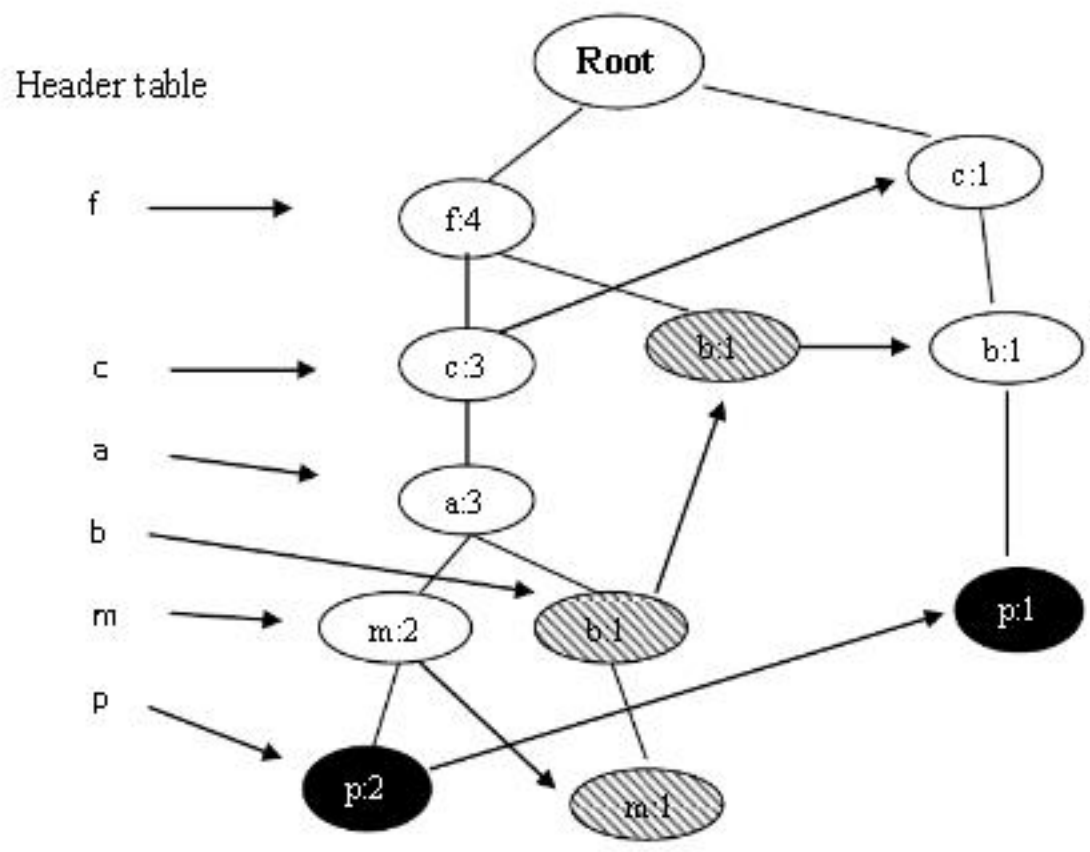
Examples of FP tree :

Ex 4.4.2 : Finding all the patterns with 'p' in the FP tree given below.



Solution :

- Starting from the bottom of the header table.

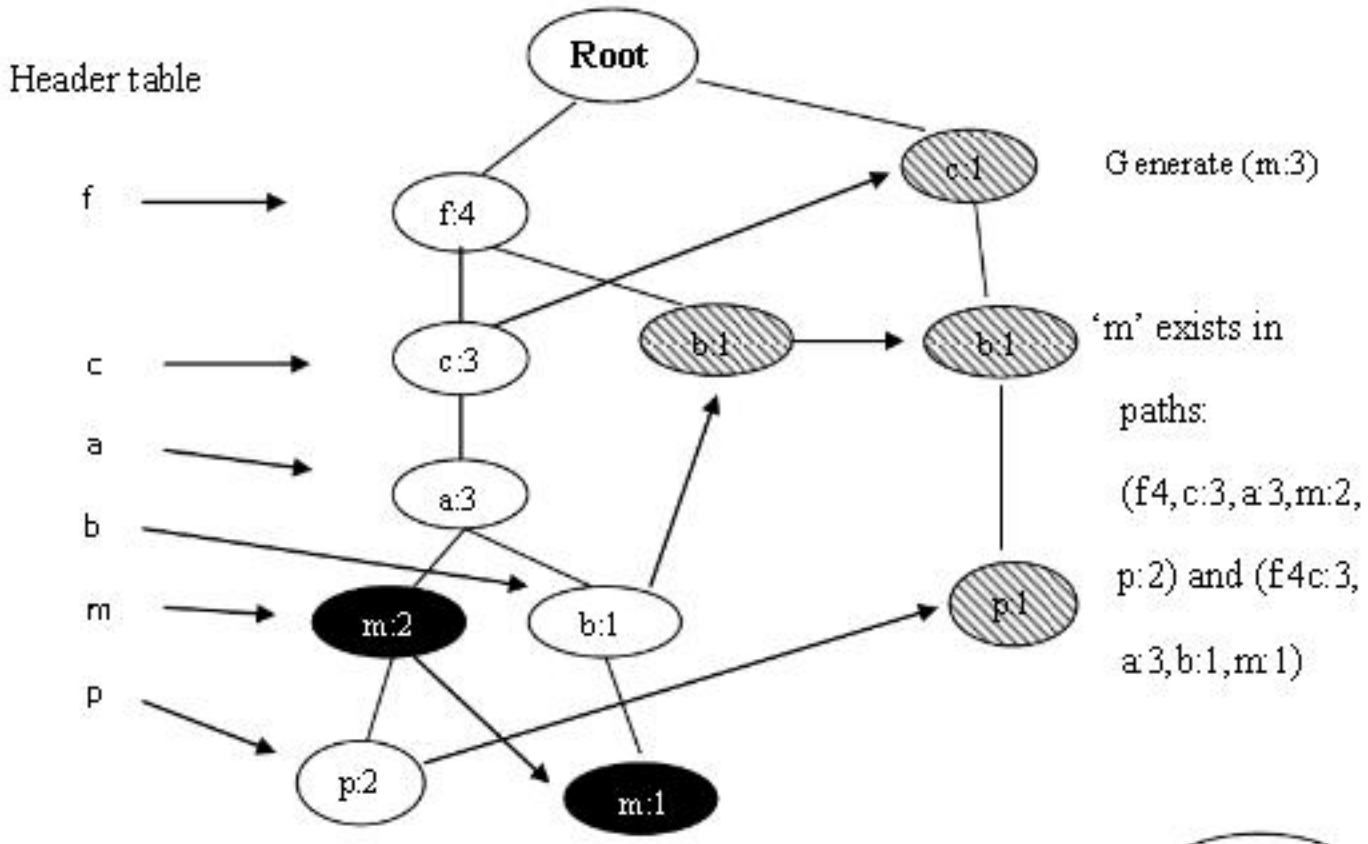


- Following are the paths with 'P'
  - We got (f:4,c:3,a:3,m:2,p:2) and (c:1,b:1,p:1)
  - The transaction containing 'p' have p.count.
  - Therefore we have (f:2,c:2,a:2,m:2,p:2) and (c:1,b:1,p:1).
  - Since 'p' is part of these we can remove 'p'.
- Conditional Pattern Base (CPB)
  - After removing P we get : (f:2,c:2,a:2,m:2) and (c:1,b:1)
  - Find all frequent patterns in the CPB and add 'p' to them, this will give us all frequent patterns containing 'p'.
  - This can be done by constructing a new FP-Tree for the CPB.
- Finding all patterns with 'P'.
  - We again filter away all items < minimum support threshold (i.e. 3)
  - (f:2,c:2,a:2,m:2),(c:1,b:1) => (c:3)
  - We generate (cp : 3) (Note : we are finding frequent patterns containing item p, so we append p to c as c is only item that has min support threshold).
  - Support value is taken from the sub-tree.
  - Frequent patterns thus far : (p:3, cp:3).

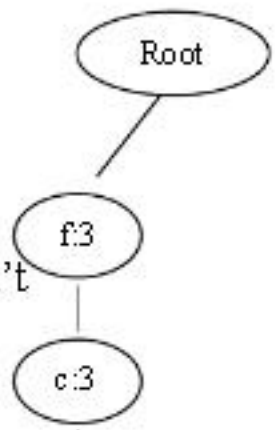
Ex. 4.4.3 : Finding patterns with 'm' but not 'p'.

Solution :

- Find 'm' from the header table.



- Conditional Pattern Base :
  - Path 1 : (f:2,c:2,a:3,m:2,p:2) -> (f:2,c:2,a:2)
  - In the above transaction we need to consider m:2, based on this we get f:2 and so on. Exclude p as we don't want p i.e. given in example.
  - Path 2 : (f:4,c:3,a:3,b:1,m:1) - > (f:1,c:1,a:1,b:1)
- Build FP tree using (f:2,c:2,a:2) and (f:1,c:1,a:1,b:1)
- Now we got (f:3,c:3,a:3,b:1)
- Initial filtering removes b:1 (We again filter away all items < Minimum support threshold).
- Mining Frequent Patterns by Creating Conditional Pattern-Bases.



Item	**Conditional pattern-base**	**Conditional FP-tree**
P	{(fcam:2),(cb:1)}	{(c:3)}|p
M	{(fca:2),(fcab:1)}	{(f:3,c:3a:3)}|m
B	{(fca:1),(f:1),(c:1)}	Empty
A	{(fc:3)}	{(f:3,c:3)}|a
C	{(f:3)}	{(f:3)}|c
f	Empty	Empty

- Support count of D = 1.

- Conditional Pattern Base (CPB)
- To find all frequent patterns containing 'D' we need to find all frequent patterns in the CPB and add 'D' tot them.
- We can do this by constructing a new FP-Tree for the CPB.
- Finding all patterns with 'D'.
    - We again filter away all items < minimum support threshold (i.e. 1)
    - {(A:1,B:1,C:1),(A:1,B:1),(A:1,C:1)(A:1)} => {(A:4,B:2,C:2)}
    - We generate ABCD:1
    - Similarly for other branch of the tree {(B:1,C:1)} => {(B:1,C:1)}
    - We generate BCD:1
    - Recursively apply FP-growth
    - So Frequent Itemsets found (with sup > 1) : AD,BD,CD,ACD,BCD which are generated from CPB on conditional node D.

Benefits of the FP-Tree Structure :

Completeness :

- The long pattern of any transaction is never broken.
- For frequent pattern mining complete information is preserved.
- The method can mine short as well as long frequent patterns and it is highly efficient.
- FP-Growth algorithm is much faster than Apriori Algorithm.
- The search cost is reduced.

4.5 Mining Multilevel Association Rules :

- Items are always in the form of hierarchy.
- Items which are at leaf nodes are having lower support.
- An item can be either generalized or specialized as per the described hierarchy of that item and its levels can be powerfully preset in transactions.
- Rules which combine associations with hierarchy of concepts are called Multilevel Association Rules.

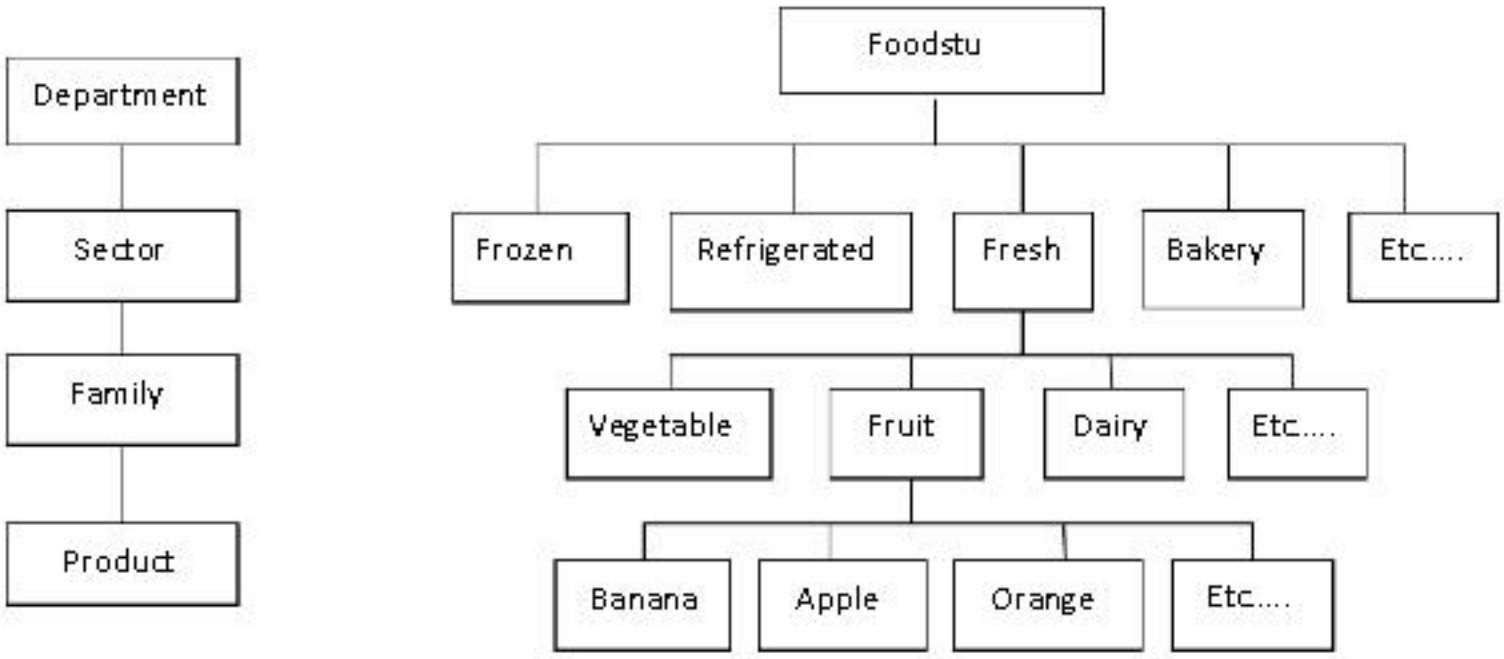


Fig. 4.5.1 Hierarchy of Concept

Support and confidence of multilevel association rules :

- The support and confidence of an item is affected due to its generalization or specialization value of attributes.
- The support of generalization item is more than the support of specialization item
- Similarly the support of rules increases from specialized to generalized itemsets.
- If the support is below the threshold value then that rule becomes invalid.
- Confidence is not affected for general or specialized.

Two approaches of multilevel association rule :

1. Using uniform minimum support for all levels.

- Consider the same minimum support for all levels of hierarchy.
- As only one minimum support is set, so there is no necessary to examine the items of itemset whose ancestors do not have minimum support.
- If very high support is considered then many low level association may get missed.
- If very low support is considered then many high level association rules are generated.

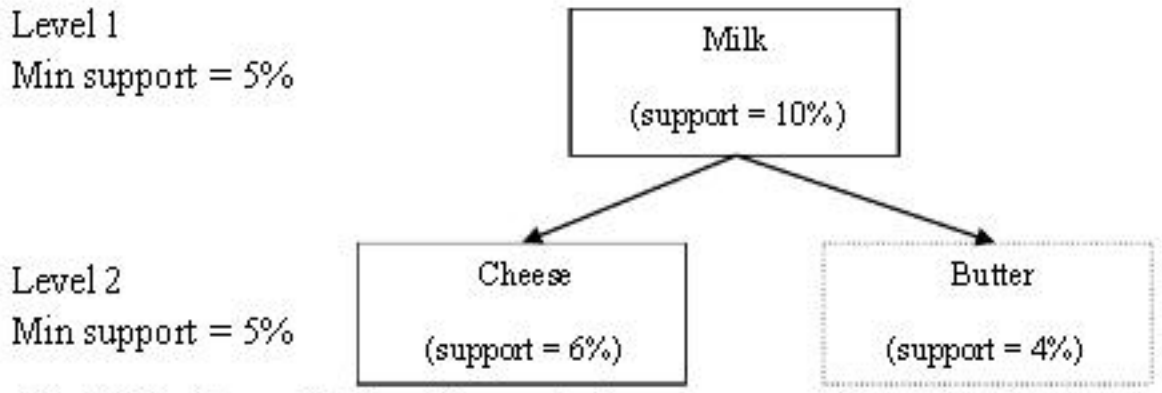


Fig 4.5.2 : Example of uniform minimum support for all levels

2. Using reduced minimum support at lower level :

- Consider separate minimum support at each level of hierarchy.
- As every level is having its own minimum support, the support at lower level reduces.

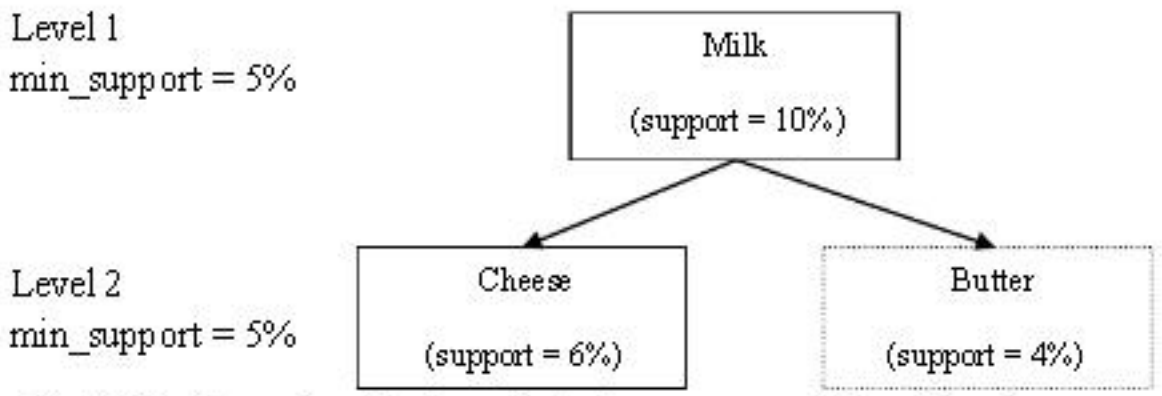


Fig 4.5.3 : Example of reduced minimum support for all levels

There are 4 search strategies.

i. **Level-by-level independent :**
   - It's a full-breadth search method.
   - The parent node is checked whether it's frequent or not frequent and based on that node is examined.

ii. **Level-cross filtering by single item :**
   The children of only frequent nodes are checked.

iii. **Level-cross filtering by k-itemset :**
   - Find the frequent k itemset at the parent level.
   - Only the k itemsest at next level is checked.

iv. **Controlled level-cross filtering by single item** :
   - This is the modified version of Level-cross filtering by single item.

- Some minimum support threshold is set for lower level.
- So the items which do not satisfy minimum support checked for minimum support threshold this is also called "Level Passage Threshold".

4.6. Constraint based Association Rule Mining :

Mining performed based on user specific constraints is called constraint-based mining.

Forms of constraints.

1. Knowledge type constraints.
2. Data constraints.
3. Dimension/Level constraints.
4. Interestingness constraints.
5. Rule constraint.

Mining query optimizer' must be incorporated in the mining process to exploit the constraints specified.

Metarule-Guided Mining of Association Rule :

- Specifies the syntactic form of the rules in which we are interested.
- Syntactic forms serves as the constraints.
- It is based on analysts experience or intuition regarding data.
- To analyze the customers behavior leading to the purchase of Apple Product, meta rule will P_1(C,Y) and P_2(C,Z) -> buys (C, "Apple Products").
  Where P_1, P_2 are the predicates on customer C for values Y and Z of predicates P_1 and P_2.
- Data mining system looks for the patterns which matches the given metarules. For example, if two predicates Age and Salary are given to analyze whether the customer buys "Apple Product".
  age(C,"30.....40")^ Salary (C,"30K.....50K") -> buys(C, "Apple Products").
- So generalize the metarule Guided association rule as a template like

P_1 ^P_2^.....^P_n -> Q_1^Q_2^.....^Q_r

where, each P_i's and Q_j's are predicates.

And the number of predicate in the metarule is p =n+r.

UNIT – V Classification

Introduction to: Classification and Regression for Predictive Analysis,
Decision Tree Induction,
Rule-Based Classification: using IF-THEN Rules for Classification,
Rule Induction Using a Sequential Covering Algorithm.
Bayesian Belief Networks,
Training Bayesian Belief Networks,
Classification Using Frequent Patterns,
Associative Classification,
Lazy Learners-k-Nearest-Neighbor Classifiers,
Case-Based Reasoning.

Introduction:

- Classification constructs the classification model based on training data set and using that model classifies the new data.
- It predicts the value of classifying attribute or class label.
- Typical application :
  - Classify credit approval based on customer data.
  - Target marketing of product.
  - Medical diagnosis based on symptoms of patient.
  - Treatment effectiveness analysis of patient based on their treatment given.
- Various classification techniques :
  - Regression
  - Decision trees
  - Rules
  - Neural networks.

Classification Problem:

- Suppose a database D is given as D = {t1, t2,……,tn} and a set of desired classed are C = {C1, C2,……,Cm}, the Classification problem is to define the mapping m in such a way that which tuple of database D belongs to which class of C. Actually divides D into equivalence classes.
- Prediction is similar, but may be viewed as having infinite number of classes.
- Prediction models continuous-valued functions, i.e., predicts unknown or missing values.

Classification Example:

How teacher gives grades to students based on their marks obtained.

- If x >= 90 then grade = A.
- If 80 <= x < 90 then grade = B.
- If 70 <= x < 80 then grade = C.
- If 60 <= x < 70 then grade = D.
- If x < 60 then grade = E.

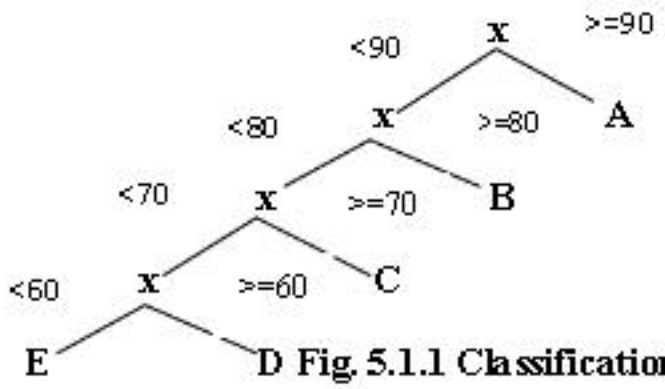


Fig. 5.1.1 Classification of grading

Classification Requirements:

Classification is a Two Step Process:

1. Model Construction:

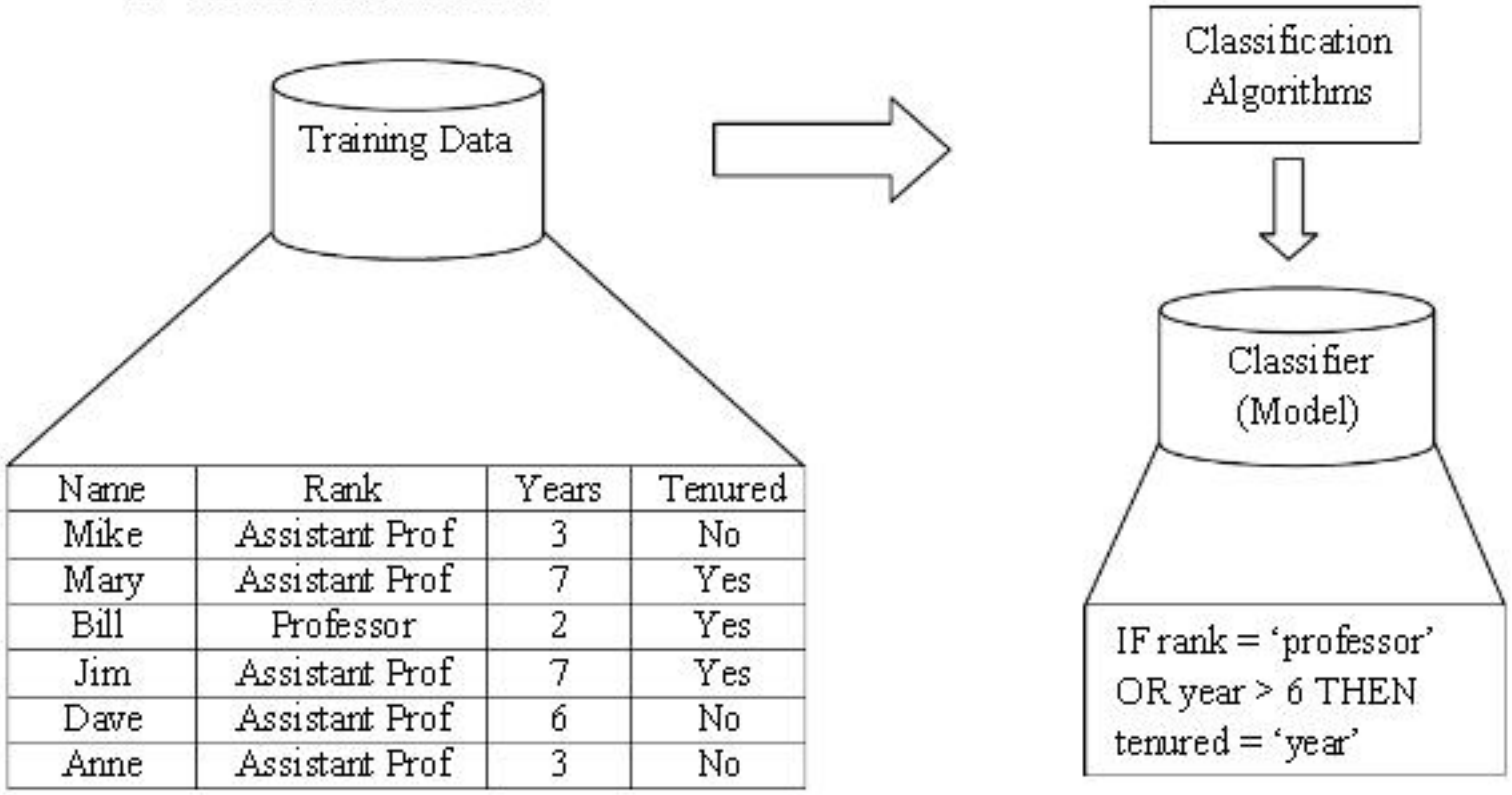
- Every sample tuple or object has assigned a predefined class label.
- Those set sample tuples or subset data set is known as training data set.
- The constructed model based on training data set is represented as classification rules, decision trees or mathematical formulae.

2. Model usage:

- For classifying unknown objects or new tuple use the constructed model.
- Compare the class label of test sample with the resultant class label.
- Estimate accuracy of the model by calculating the percentage of test set samples that are correctly classified by the model constructed.
- Test sample data and training data samples are always different, otherwise over-fitting will occur.

Example: Classification process:

1. **Model Construction :**

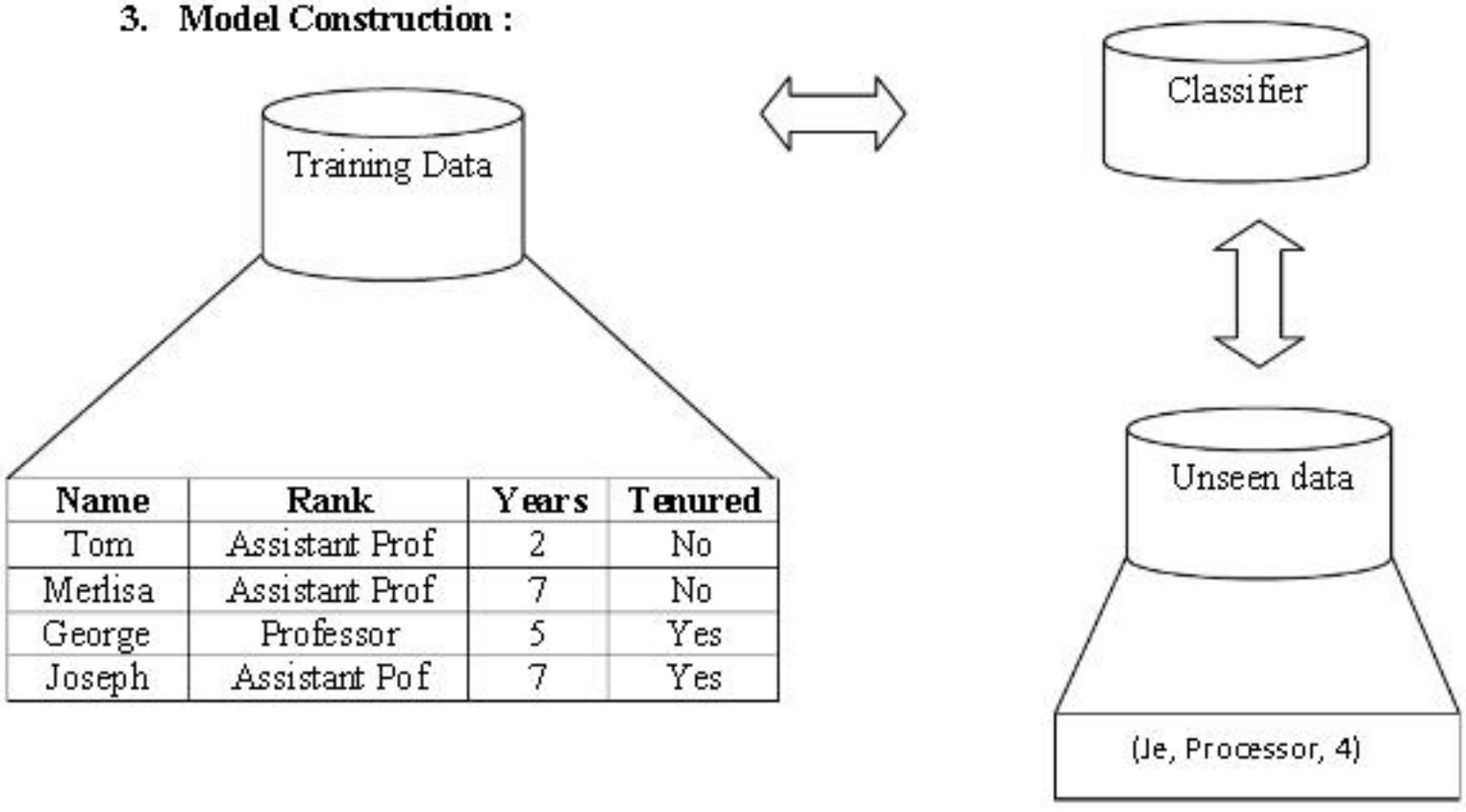


Name	Rank	Years	Tenured
Mike	Assistant Prof	3	No
Mary	Assistant Prof	7	Yes
Bill	Professor	2	Yes
Jim	Assistant Prof	7	Yes
Dave	Assistant Prof	6	No
Anne	Assistant Prof	3	No

Fig. 5.2.1 Learning: Training data are analyzed by a classification algorithm

Classification process:

2. **Model usage (Use the model in prediction)**
3. **Model Construction :**

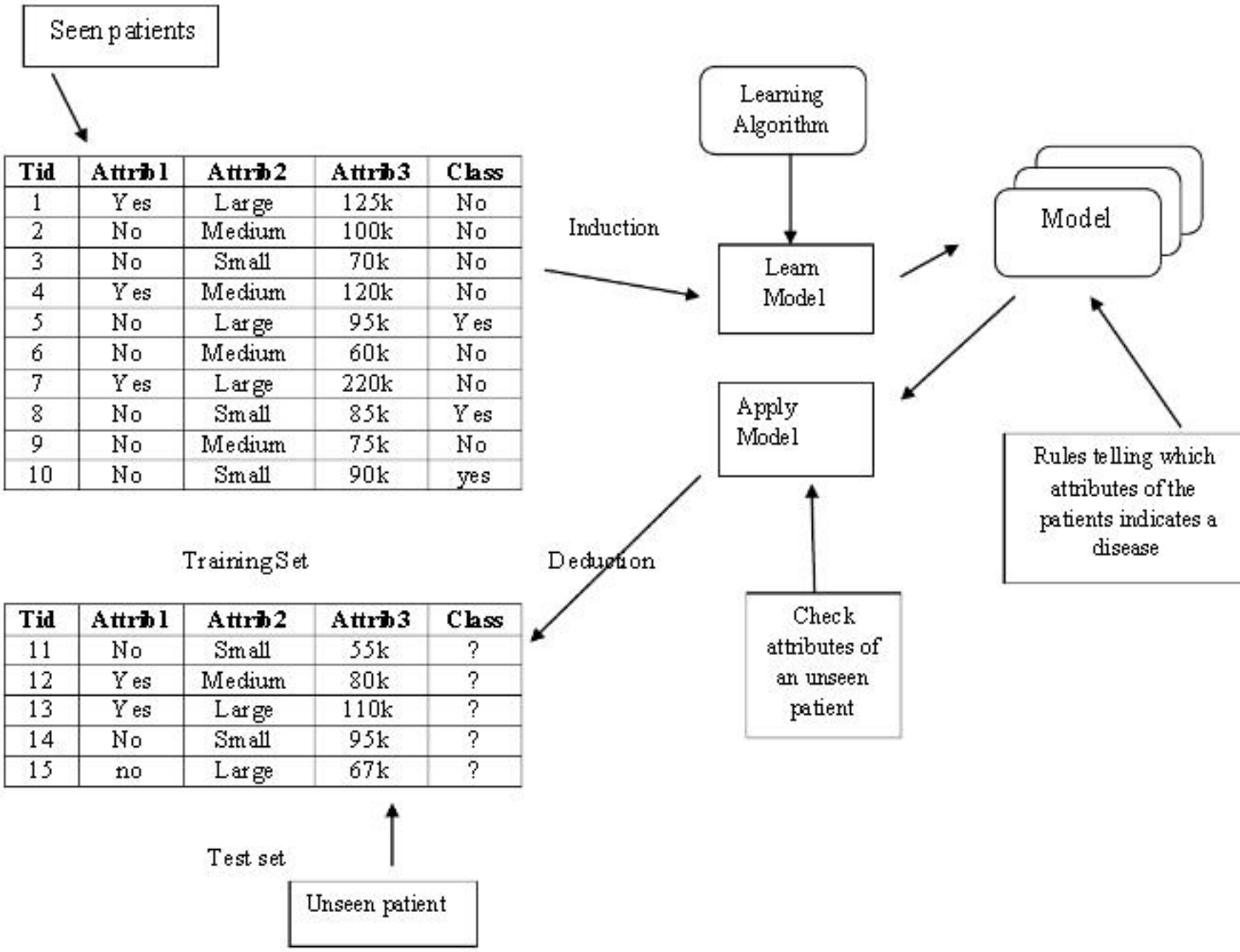


Name	Rank	Years	Tenured
Tom	Assistant Prof	2	No
Merlisa	Assistant Prof	7	No
George	Professor	5	Yes
Joseph	Assistant Pof	7	Yes

Fig. 5.2.2 Classification: Test data are used to estimate the accuracy of the classification rule.

For example:

How to perform classification task for classification of medical patients by their disease?



Tid	Attrib1	Attrib2	Attrib3	Class
1	Yes	Large	125k	No
2	No	Medium	100k	No
3	No	Small	70k	No
4	Yes	Medium	120k	No
5	No	Large	95k	Yes
6	No	Medium	60k	No
7	Yes	Large	220k	No
8	No	Small	85k	Yes
9	No	Medium	75k	No
10	No	Small	90k	yes

Training Set

Tid	Attrib1	Attrib2	Attrib3	Class
11	No	Small	55k	?
12	Yes	Medium	80k	?
13	Yes	Large	110k	?
14	No	Small	95k	?
15	no	Large	67k	?

Fig. 5.2.3 : Classification model for medical patients by their disease

Difference between Classification and Prediction:

Sr. No	Classification	Prediction
1.	Classification is a major type of prediction problem where classification is used to predict discrete or nominal values	Prediction can be viewed as the construction and use of a model to assess the class of an unlabeled sample.
2.	Classification is the use of prediction to predict class labels.	It is used to assess the values or value ranges of an attribute that a given sample if likely to have.

3.	E.g. Group patients based on their known medical data and treatment outcome then it's a classification	E.g. if a classification model is used to predict the treatment outcome for a new patient, then it would be a prediction.

Issues Regarding Classification and Prediction :

Data Preparation :

- **Data cleaning :** Pre-process data in order to reduce noise and handle missing values.
- Relevance **analysis(feature selection) :** Remove the irrelevant or redundant attributes.
- **Data transformation :** Generalize the data to higher level concepts using concept hierarchies and/or normalize data which involves scaling the values.

Evaluating classification methods :

- **Predictive accuracy :** This refers the ability of the model to correctly predict the class label of new or previously unseen data.
- **Speed and scalability :**
  - Time to construct the model
  - Time to use the model
  - Efficiency in disk-resident databases.
- **Robustness :** Handling noise and missing values
- **Interpretability :** Understanding and insight provided by the model
- **Goodness of rule :**
  - Decision tree size.
  - Compactness of classification rules.

Regression:

- Suppose an employee needs to predict how much rise he will get in his salary after 5 years, means he bother to predict the numeric value. In this case a model is constructed based on his previous salary values that predicts a continuous-values function or ordered value.
- Prediction is generally about the future values or the unknown events and it models continuous-valued functions.
- Most commonly used methods for prediction is regression.

Structure of Regression Model :

- Regression Model represents reality by using the system of equations.
- Regression model explains relationship between variables and also enables quantification of these relationships.

- It determines the strength of relationship between one dependent variable with the other independent variable using some statistical measure.
- Dependent variable is usually denoted by Y.
- The two basic types of regression :
  1. Linear regression.
  2. Multiple regression.

- The general form of regression is :

  Linear Regression : $Y = m+nX+u$

  Multiple Regression : $Y = m+n_1X_1 +n_2 X_2 +n_3X_3 +.....+n_tX_t +u$

  Where, Y = The dependent variable which we are trying to predict

  X = The independent variable that we are using to predict variable Y

  m = The intercept

  n = The slope

  u = The regression residual
- In multiple regressions each variable is differentiated with subscripted numbers.
- Regression uses a group of random variables for prediction and finds a mathematical relationship between them. This relationship is depicted in the form of a straight line (linear regression) that approximates all the points in the best way.
- Regression may be used to determine for e.g. price of a commodity, interest rates, the price movement of an asset influenced by industries or sectors.

Linear Regression :

Regression tries to find the mathematical relationship between variables, if it is a straight line then it is linear model an if it gives a curved line then it is non linear model.

Simple linear regression :

- The relationship between dependent and independent variable is described by straight line and it has only one independent variable.

$$\mathbf{Y = \alpha+\beta X}$$

- Two parameters, α and β specify the (Y-intercept and slope of the) line and are to be estimated by using the data at hand.
- The value of Y increases or decreases in a linear manner as the value of X changes accordingly.
- Draw a line relating to Y and X which is well fitted to given data set.
- The idea situation is that if the line which is well fitted for all the data points and no error for prediction.
- If there is random variation of data points, which are not fitted in a line then construct a probabilistic model related to X and Y.

- Simple linear regression model assumes that data points deviates about the line, as shown in below figure.

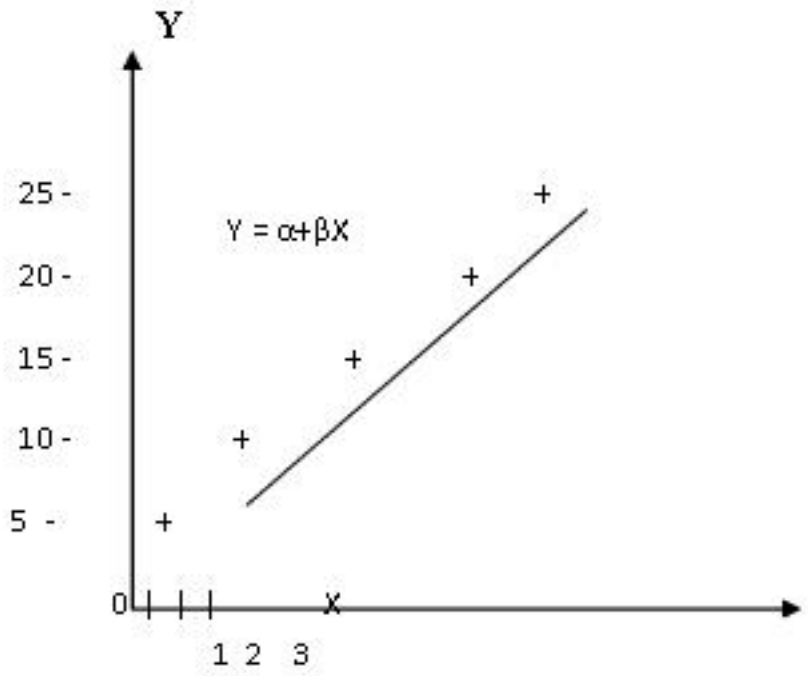


Figure: Linear Regression

Multiple Linear Regression :

- Multiple linear regression is an extension of simple linear regression analysis.
- It uses two or more independent variables to predict the outcome and a single continuous dependent variable.

$$Y = a_0 + a_1 X_1 + a_2 X_2 + \ldots\ldots + a_k X_k + e$$

Where, Y is the dependent variable or response variable

$X_1, X_2 \ldots\ldots X_k$ are the independent variables or predictors e is random error.

$a_0, a_1, a_2 \ldots\ldots a_k$ are the regression coefficients. **Other Regression Model :**

- In log linear regression a best fit between the data and a log linear model is found.
- Major assumption : A linear relationship exists between the log of the dependent and independent variable.
- Log linear models are models that postulate a linear relationship between the independent variables and the logarithm of the dependent variable, for example :

$$\log(y) = a_0 + a_1 X_1 + a_2 X_2 + \ldots\ldots + a_N X_N$$

where y is the dependent variable ; x_i, i=1,….,N, are independent variables and {a_i, i=0,……,N} are parameters (coefficients) of the model.

- For example, log linear models are widely used to analyze categorical data represented as a contingency table. In this case, the main reason to transform frequencies (counts) or probabilities to their log-values is that, provided the independent variables are not correlated with each other, the relationship between the new transformed dependent variable and the independent variables is a linear (addictive) one.

5.3 Methods of Supervised Learning:

Classification methods are given below :

1. Decision Tree Induction : Attribute selection measures, tree pruning
2. Bayesian Classification : Naïve Baye's Classifier

5.4 Decision Trees:

- Training dataset should be class-labeled for learning of decision trees in decision tree induction.
- A decision tree represents rules and it is very a popular tool for classification and prediction.
- Rules are easy to understand and can be directly used in SQL to retrieve the records from database.
- To recognize and approve the discovered knowledge got from decision model is very crucial task.
- There are a many algorithms to build decision trees:
  - ID3 (Iterative Dichotomiser 3)
  - C4.5 (Successor of ID3)
  - CART(Classification And Regression Tree)
  - CHAID(Chi-squared Automatic Interaction Detector).

5.4.1 Appropriate Problems for Decision Tree Learning :

Decision tree learning is appropriate for the problems having the characteristics given below :

- Instances are represented by a fixed set of attributes (e.g. gender) and their values (e.g. male, female) described as attribute-value pairs.
- If the attribute has small number of disjoint possible values (e.g. high, medium, low) or there are only two possible classes (e.g. true, false) then decision tree learning is easy.
- Extension of decision tree algorithm also handles real value attributes (e.g. salary).

- Decision tree gives a class label to each instance of dataset.
- Decision tree methods can be used even when some training examples, have unknown values (e.g. humidity is known for only a fraction of the examples).
- Learned functions are either represented by a decision tree or represented as sets of if-then rules to improve readability.

5.4.2 Decision Tree Representation :

Decision tree classifier has three type structure which has leaf nodes and decision nodes.

- A leaf node is the last node of each branch and indicates class label or value of target attribute.
- A decision node is the node of tree which has leaf node or sub-tree. Some test to be carried on the each value of decision node to get the decision of class label or to get next sub-tree.

Example : Decision tree representation for play tennis.

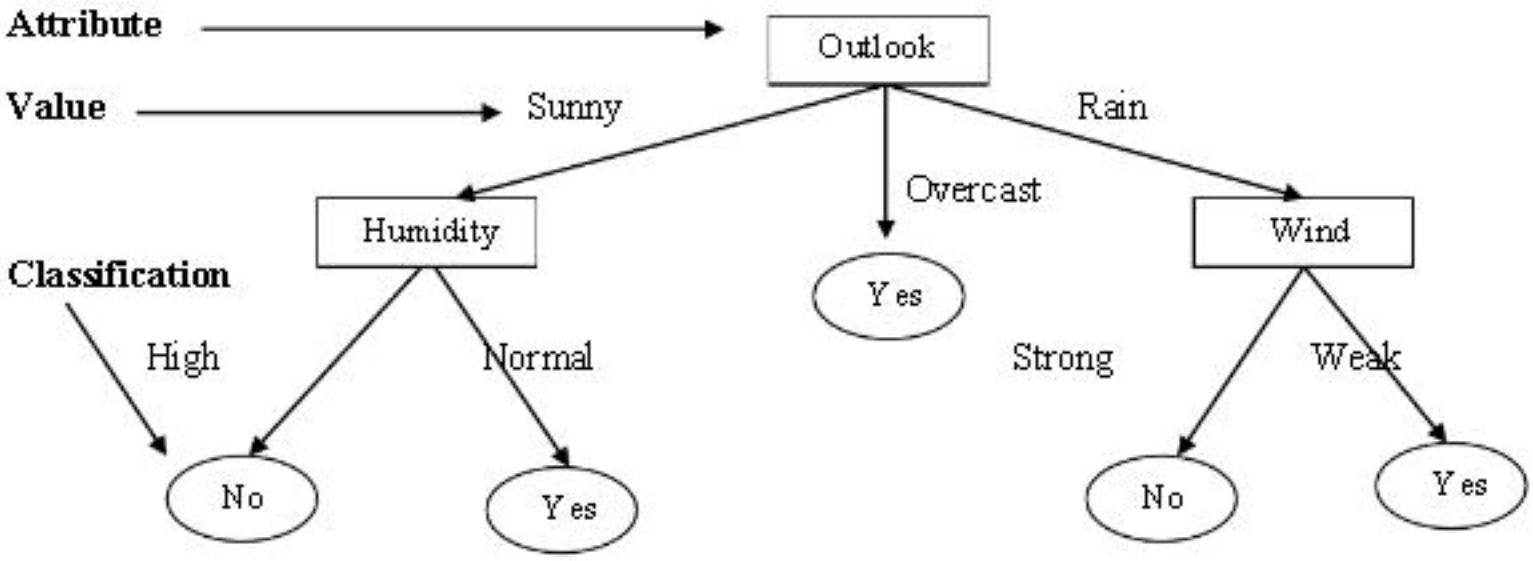


Fig. 5.4.1 Representation of decision trees

Other representation for play tennis :

- Logical expression for play tennis = Yes is given below,
  (Outlook = sunny ∧ humidity = normal) ∨ (outlook = overcast) ∨ outlook = rain ∧ wind = weak
- If-then rules :
  - If outlook =sunny ∧ humidity = normal then play tennis = Yes
  - If outlook =sunny ∧ humidity = high then play tennis = No
  - If outlook =overcast then play tennis = Yes
  - If outlook = rain ∧ wind = weak then play tennis = Yes
  - If outlook = rain ∧ wind = strong then play tennis = No

5.5.

Bayesian Classification :

5.6.1. Baye's Theorem :

- It is also known as Baye's Rule.
- Baye's theorem is used to find conditional probabilities.
- The conditional probability of an event is a likelihood obtained with the additional information that some other event has previously occurred.
- P(X|Y) is the conditional probability of event X occurring for the event Y which was already occurred.

$$P(X|Y) = P(X \text{ and } Y)/P(A)$$

- An initial probability is called as a priori probability which we get before any additional information is obtained.
- The probability is called as a posterior probability. Value which we get or revised after any additional information is obtained.

5.6.2 Naïve Bayes Classification :

- Probabilistic learning : Explicit probabilities are calculated for Hypothesis.
- Incremental : The probability of a hypothesis whether it is correct can be incrementally increased or decreased by each training example.
- Probabilistic prediction : Multiple hypothesis can be predicted by their probability weight.
- Meta-classification : The outputs of several classifiers can be obtained, e.g. by multiplying the probabilities that all classifiers predict for a given class.
- Standard : The computationally intractable Bayesian methods provide a standard of optimal decision making against which other methods can be measured.

Given training data D, posteriori probability of a hypothesis h, P(h|D) follows the Bayes theorem

$$P(h|D) = \frac{P(D|h)P(h)}{P(D)}$$

P(h) = Independent probability of h: prior probability

P(D) = Independent probability of D

P(D|h) = Conditional probability of D given h : likelihood

P(h|D) = Conditional probability of h given D : posteriori probability

Practical difficulties :

- Require initial knowledge of many probabilities
- Significant computational cost.

5.6.3 Naïve Bayes Classifier : Example

Ex5.6.1 Training set is given for play-tennis example.

Outlook	Temperature	Humidity	Windy	Class
Sunny	Hot	High	False	No
Sunny	Hot	High	True	No
Overcast	Hot	High	False	Yes
Rain	Mild	High	False	Yes
Rain	Cool	Normal	False	Yes
Rain	Cool	Normal	True	No
Overcast	Cool	Normal	True	Yes
Sunny	Mild	High	False	No
Sunny	Cool	Normal	False	Yes
Rain	Mild	Normal	False	Yes
Sunny	Mild	Normal	True	Yes
Overcast	Mild	High	True	Yes
Overcast	Hot	Normal	False	Yes
rain	mild	High	True	No

Solution :

Given a training set, we can compute the probabilities as follow:

- The classification problem may be formalized using a-posteriori probabilities:
  P(C|Y) is the probability that the sample tuple $Y=<y_1, \ldots\ldots, y_k>$ is of class C.
- Assign to sample Y the class label C such that P(C|Y) is maximal.
- From the above given sample data, calculate the probabilities for play tennis(P) and don't play tennis(N) for all attributes.

Outlook	
P(sunny|Yes) = 2/9	P(sunny|No) = 3/5
P(overall|Yes) = 4/9	P(overall|No) = 0
P(rain|Yes) = 3/9	P(rain|No) = 2/5
Temperature	
P(hot|Yes) = 2/9	P(hot|No) = 2/5
P(mild|Yes) = 4/9	P(mild|No) = 2/5
P(cool|Yes) = 3/9	P(cool|No) = 1/5
Humidity	
P(high|Yes) = 3/9	P(high|No) = 4/5
P(normal|Yes) = 6/9	P(normal|No) = 2/5
Windy	
P(true|Yes) = 3/9	P(true|No) = 3/5
P(false|Yes) = 6/9	P(false|No) = 2/5

P(No) = 5/14

- As unseen sample Y = <rain, hot,high, false>
- P(Y|Yes).P(Yes) = P(rain|Yes).P(hot|Yes).P(high|Yes).P(false|Yes).P(Yes)
  = 3/9.2/9.3/9.6/9.9/14 = 0.010582
  P(Y|No).P(No) = P(rain|No).P(hot|No).P(high|No).P(false|No).P(No)
  = 2/5.2/5.4/5.2/5.5/14 = 0.018286
- Choose the class so that it maximizes this probability. This means that the new instances will be classified as no. (don't play)
- Sample Y is classified in class No(i.e. don't play)

An unseen sample = <sunny, cool, high, true>

P(Y|Yes).P(Yes) = P(sunny|Yes).P(cool|Yes).P(high|Yes).P(true|Yes).P(Yes)
= 2/9.3/9.3/9.3/9.9/14 = 0.0053
P(Y|No).P(No) = P(sunny|No).P(cool|No).P(high|No).P(true|No).P(No)
= 3/5.1/5.4/5.3/5.5/14 = 0.0206

Now choose the class so that it maximizes this probability. This means that the new instance will be classified as no.(don't play)

Ex. 5.6.2 : Car theft example : Attributes are color, type, origin, and the subject, stolen can be either yes or no.

Data set:

Car. No	Color	Type	Origin	Stolen
1	Red	Sports	Domestic	Yes
2	Red	Sports	Domestic	No
3	Red	Sports	Domestic	Yes
4	Yellow	Sports	Domestic	No
5	Yellow	Sports	Imported	Yes
6	Yellow	SUV	Imported	No
7	Yellow	SUV	Imported	Yes
8	Yellow	SUV	Domestic	No
9	Red	SUV	Imported	No
10	Red	Sports	imported	yes

Solution :

We want to classify a <Red, Domestic, SUV> i.e. unseen sample. Note there is no example of a <Red, Domestic, SUV> in our data set.

P(Yes) = 5/10
P(No) = 5/10

Color	
P(Red|Yes) = 3/5	P(Red|No) = 2/5
P(Yellow|Yes) = 2/5	P(Yellow|No) = 3/5
Type	
P(SUV|Yes) = 1/5	P(SUV|No) = 3/5

P(Sports|Yes) = 4/5	P(Sports|No) = 2/5
Origin	
P(Domestic|Yes) = 2/5	P(Domestic|No) = 3/5
P(Imported|Yes) = 3/5	P(Imported|No) = 2/5

An unseen sample X = <Red, Domestic, SUV>

P(X|Yes).P(Yes) = P(Red|Yes).P(Domestic|Yes).P(SUV|Yes).P(Yes)

3/5.2/5.1/5.5/10 = 0.024

P(X|No).P(No) = P(Red|No).P(Domestic|No).P(SUV|No).P(No)

= 2/5.3/5.3/5.5/10 = 0.072

Since 0.072 > 0.024, our example gets classified as 'NO'.

Ex. 5.6.3 : Consider the following data set S, which contains observations of several cases of sunburn:

Name	Hair	Height	Weight	Dublin	Result
Sarah	Blonde	Average	Light	No	Sunburned
Dana	Blonde	Tall	Average	Yes	None
Alex	Brown	Short	Average	Yes	None
Annie	Blonde	Short	Average	No	Sunburned
Emily	Red	Average	Heavy	No	Sunburned
Pete	Brown	Tall	Heavy	No	None
John	Brown	Average	Heavy	No	None
Katie	Brown	Short	Light	Yes	None

Unseen sample x = <brown, tall, average no> Predict the results value as sunburned or none?

Solution :

Hair	
P(Blonde|Yes) = 2/3	P(Blonde|No) = 1/5
P(Brown|Yes) = 0	P(Brown|No) = 4/5
P(Red|Yes) = 1/3	P(Red|No) = 0
Height	
P(Average|Yes) = 2/3	P(Average|No) = 0
P(Tall|Yes) = 0	P(Tall|No) = 2/5
P(Short|Yes) = 1/3	P(Short|No) = 2/5
Weight	
P(Light|Yes) = 1/3	P(Light|No) = 1/5
P(Average|Yes) = 1/3	P(Average|No) = 2/5
P(Heavy|Sunburned) = 1/3	P(Heavy|None) = 2/5

Dublin	
P(No|Sunburned) = 3/3	P(No|None) = 2/5
P(Yes|Sunburned) = 0	P(Yes|None) = 3/5

P(Sunburned) = 3/8
P(None) = 5/8

- An unseen sample X = <brown, tall, average, no>

P(X|Sunburned).P(Sunburned) = P(Brown|Sunburned).P(Tall|Sunburned)
.P(Average|Sunburned). P(No|Sunburned).P(Sunburned)
= 0

P(X|None).P(None) = P(Brown|None).P(Tall|None)
.P(Average|None). P(No|None).P(None)
= 0.032

Since 0.032 > 0, our example gets classified as 'NONE'.

Ex. 5.6.4 : Predict a class label of an unknown sample using Naïve Bayesian Classification on the following training datasets from all electronics customer database.

Age	Income	Student	Credit_rating	Class:buys-computer
<=30	High	No	Fair	No
<=30	High	No	Excellent	No
31....40	High	No	Fair	Yes
>40	Medium	No	Fair	Yes
>40	Low	Yes	Fair	Yes
>40	Low	Yes	Excellent	No
31....40	Low	Yes	Excellent	Yes
<=30	Medium	No	Fair	No
<=30	Low	Yes	Fair	Yes
>40	Medium	Yes	Fair	Yes
<=30	Medium	Yes	Excellent	Yes
31....40	Medium	No	Excellent	Yes
31....40	High	Yes	Fair	Yes
>40	Medium	No	Excellent	No

Solution :

The unknown sample is x = {age="<=30", Income="Medium", Student="Yes", Credit_rating ="Fair"}

Age	
P(<=30|Yes) = 2/9	P(<=30|No) = 3/5

P(31….40|Yes) = 4/9	P(31….40|No) = 0
P(>40|Yes) = 3/9	P(>40|No) = 2/5
Income	
P(High|Yes) = 2/9	P(High|No) = 2/5
P(Medium|Yes) = 4/9	P(Low|No) = 2/5
P(Low|Yes) = 3/9	P(Medium|No) = 1/5
Student	
P(No|Yes) = 3/9	P(No|No) = 4/5
P(Yes|Yes) = 6/9	P(Yes|No) = 1/5
Credit_rating	
P(Fair|Yes)= 6/9	P(Fair|No) = 2/5
P(Excellent|Yes) = 3/9	P(Excellent|No) = 3/5

P(Yes) = 9/14
P(No) = 5/14

- An unseen sample X = <age="<=30", Income="Medium", Student="Yes ", Credit_rating="Fair ">

  P(X|Yes).P(Yes) = P(Age<=30|Yes).P(Income=Medium|Yes)
  .P(Student="yes"|Yes).P(credit_ranking="fair"|Yes).P(Yes)
  = 2/9.4/9.6/9.6/9.9/14
  = 0.028

  P(X|No).P(No) = P(Age<=30|No).P(Income=Medium|No)
  .P(Student="yes"|No).P(credit_ranking="fair"|No).P(No)
  = 0.007

  Since 0.028 > 0.007, therefore the Naïve Bayesian classifier predicts Buys computer = "Yes" for sample X.

Ex.5.6.5 : Using Naïve Bayesian Classification on the following given training set, classify the unseen tuple (Refund = No, Married, Income=120K)

Rid	Refund	Marital Status	Taxable	Evade
1	Yes	Single	125k	No
2	No	married	100k	No
3	No	Single	120k	No
4	Yes	Married	70k	No
5	No	Divorced	95k	Yes
6	No	Married	60k	No
7	Yes	Divorced	220k	No
8	No	Single	85k	Yes
9	No	Married	75k	No
10	No	Single	90k	yes

P(No) = 7/10

P(Yes) = 3/10

Solution : Given a test Record

X = (Refund = No, Married, Income = 120k)

P(X|Class=No) = P(Refund=No|Class=No).P(Married|Class=No)
.P(Income=120k|Class=No)
4/7.4/7.1/7 = 0.0466

P(X|Class=Yes) = P(Refund=No|Class=Yes).P(Married|Class=Yes)
.P(Income=120k|Class=Yes)

= 3/3.0.0=0

Since P(X|No) P(No) > P(X|Yes) P(Yes)

0.0466.7/10 > 0.3/10

Therefore, P(No|X) > P(Yes|X) => Class = No

Ex. 5.6.6 : Consider the training set for the class of mammals and non mammals, using Naïve Bayesian classification classify the unseen tuple (Give Birth=Yes, Can fly=No, Live in water=Yes, have legs=No).

Name	Give Birth	Can Fly	Live in water	Have Legs	Class
Human	Yes	No	No	Yes	Mammals
Python	No	No	No	No	Non-Mammals
Salmon	No	No	Yes	No	Non-Mammals
Whale	Yes	No	Yes	No	Mammals
Frog	No	No	Sometimes	Yes	Non-Mammals
Komodo	No	No	No	Yes	Non-Mammals
Bat	Yes	Yes	No	Yes	Non-Mammals
Pigeon	No	Yes	No	Yes	Mammals
Cat	Yes	No	No	Yes	Mammals
Leopard shark	Yes	No	Yes	No	Non-Mammals
Turtle	No	No	Sometimes	Yes	Non-Mammals
Penguin	No	No	Sometimes	Yes	Non-Mammals
Porcupine	Yes	No	No	Yes	Mammals
Eel	No	No	Yes	No	Non-Mammals
Salamander	No	No	Sometimes	Yes	Non-Mammals
Gila monster	No	No	No	Yes	Non-Mammals
Platypus	No	No	No	Yes	Mammals
Owl	No	yes	No	Yes	Non-Mammals
Dolphin	Yes	No	Yes	No	Mammals
Eagle	No	Yes	No	Yes	Non-Mammals

Solution : Unseen record is given as :

Table P.5.6.1(a)

Give Birth	Can fly	Live in water	Have legs	Class
Yes	No	Yes	No	?

A : attributes M : Mammals N : Non-Mammals

$$P(A|M) = 5/7.6/7.2/7.2/7 = 0.05$$

$$P(A|N) = 2/13.10/13.3/13.4/13 = 0.0084$$

$$P(A|M)\ P(M) = 0.05.7/10 = 0.0175$$

$$P(A|N)\ P(N) = 0.008.13/20 = 0.0052$$

$$P(A|M)\ P(M) > P(A|N)\ P(N)$$

Unseen record belongs to class mammals.

Ex. 5.6.7 : Consider the following dataset that helps to predict the RISK of a loan application based on the applicant's CREDIT HISTORY, DEBT and INCOME.

Table P. 5.6.7

Credit History	Debt	Income	Risk
Bad	Low	0 to 15	High
Bad	High	15 to 35	High
Bad	Low	0 to 15	High
Unknown	High	15 to 35	High
Unknown	High	0 to 15	High
Good	High	0 to 15	High
Bad	Low	Over 35	Moderate
Unknown	Low	15 to 35	Moderate
Good	High	15 to 35	Moderate
Unknown	Low	Over 35	Low
Unknown	Low	Over 35	Low
Good	Low	Over 35	Low
Good	High	Over 35	Low
Good	High	Over 35	Low

Solution :

- Predict the risk for unseen Tuple X = <unknown, high, over 35, moderate>.
- Write down the rule used by Naive Bayes to classify instances, and apply it to the following instance : <Credit History=bad, Debt=low, Income=15 to 35>. Which class will be returned by Naïve Bayes?

Ex. 5.6.8 : Using given table, create classification model using any algorithm and hence classify following tuple <income=medium, credit=good>.

Table P.5.6.8

TID	Income	Credit	Decision
1	Very High	Excellent	AUTHORIZE
2	High	Good	AUTHORIZE
3	Medium	Excellent	AUTHORIZE
4	High	Good	AUTHORIZE
5	Very High	Good	AUTHORIZE
6	Medium	Excellent	AUTHORIZE
7	High	Bad	REQUEST ID
8	Medium	Bad	REQUEST ID
9	High	Bad	REJECT
10	Low	Bad	CALL POLICE

Solution :

Income			
P(Very High|AUTHORIZE)=2/6	P(Very High|REQUEST ID)=0	P(Very High|REJECT)=0	P(Very High|CALL POLICE)=0
P(High|AUTHORIZE)=2/6	P(High|REQUEST ID)=1/2	P(High|REJECT)=1/1	P(High|CALL POLICE)=0
P(Medium|AUTHORIZE)= 2/6	P(Medium|REQUEST ID)=1/2	P(Medium|REJECT)= 0	P(Medium|CALL POLICE)=0
P(Low|AUTHORIZE)=0	P(Low|REQUEST ID)=0	P(Low|REJECT)=0	P(Low|CALL POLICE)=1/1
Credit			
P(Excellent|AUTHORIZE) =3/6	P(Excellent|REQUEST ID)=0	P(Excellent|REJECT) =0	P(Excellent|CALL POLICE)=0
P(Good|AUTHORIZE)=3/6	P(Good|REQUEST ID)=0	P(Good|REJECT)=0	P(Good|CALL POLICE)=0
P(Bad|AUTHORIZE)=0	P(Bad|REQUEST ID)=2/2	P(Bad|REJECT)=1/1	P(Bad|CALL POLICE)=1/1

P(AUTHORIZE) = 6/10

P(REQUEST ID) = 2/10

P(REJECT) = 1/10

P(CALL POLICE) = 1/10

P(X|AUTHORIZE) x P(AUTHORIZE) = 2/6.3/6.6/10 = 0.1

P(X|REQUEST ID) x P(REQUEST ID) = 0

P(X|REJECT) x P(REJECT) = 0

P(X|CALL POLICE) x P(CALL POLICE) = 0

Therefore the Naïve Bayesian classifier predicts decision = "AUTHORIZE" for sample X.

Rule-Based Classification: using IF-THEN Rules for Classification

A rule-based classifier uses a set of IF-THEN rules for classification. An IF-THEN rule is an expression of the form

IF condition THEN conclusion.

An example is rule R1,

R1: IF age = youth AND student = yes THEN buys computer = yes.

The "IF" part (or left side) of a rule is known as the rule antecedent or precondition. The "THEN" part (or right side) is the rule consequent. In the rule antecedent, the condition consists of one or more attribute tests (e.g., age = youth and student= yes) that are logically ANDed. The rule's consequent contains a class prediction *R*1 can also be written as

*R*1: (*age* = *youth*) ^ (*student* = *yes*)) → (*buys computer* = *yes*).

The above R1 predicts whether a customer will buy a computer.

If the condition (i.e., all the attribute tests) in a rule antecedent holds true for a given tuple, we say that the rule antecedent is satisfied (or simply, that the rule is satisfied) and

A rule *R* can be assessed by its coverage and accuracy. Given a tuple, *X*, from a class labeled data set, *D*, let n_{covers} be the number of tuples covered by *R*; $n_{corrects}$ be the number of tuples correctly classified by *R*; and |*D*| be the number of tuples in *D*. We can define the coverage and accuracy of *R* as

$$\text{Coverage}(R) = \frac{n_{covers}}{|D|}$$

$$\text{Accuracy}(R) = \frac{n_{corrects}}{n_{covers}}$$

Rule Induction Using a Sequential Covering Algorithm

Algorithm: Sequential covering. Learn a set of IF-THEN rules for classification.

Input:

- *D*, a data set of class-labeled tuples;
- *Att_vals*, the set of all attributes and their possible values.

Output: A set of IF-THEN rules.

Method:

```
(1)  Rule_set = {}; // initial set of rules learned is empty
(2)  for each class c do
(3)      repeat
(4)          Rule = Learn_One_Rule(D, Att_vals, c);
(5)          remove tuples covered by Rule from D;
(6)          Rule_set = Rule_set + Rule; // add new rule to rule set
(7)      until terminating condition;
(8)  endfor
(9)  return Rule_Set;
```

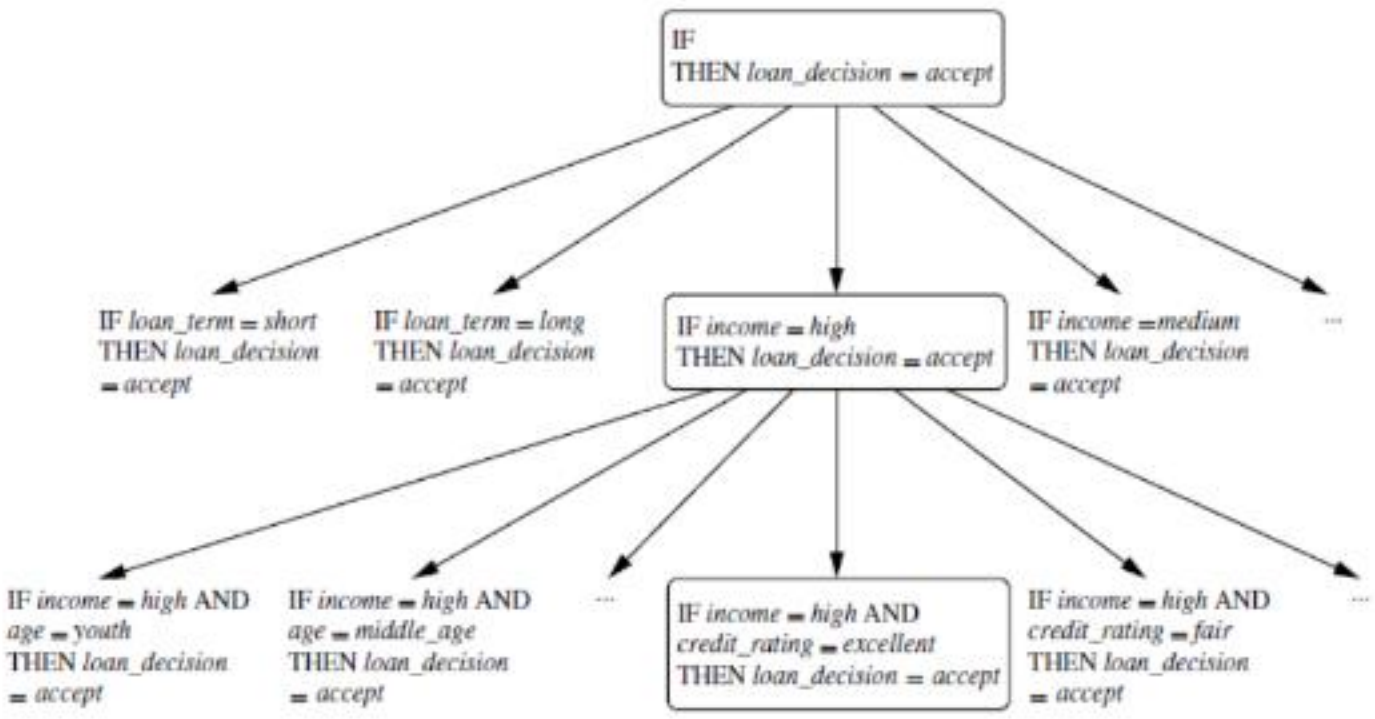


Figure: A general-to-specific search through rule space.

KNN (k-nearest-neighbor) Approach with Case Study :

- It is supervised learning method.
- In KNN data are represented in a vector space.
- The target function may be either discrete-valued or real-valued.
- For a given object E, get the top k dataset objects which are "nearest" to E by selecting distance measure.
- Then assign the class C to object E that represents the most objects after inspecting the class of these k objects.
- So for unknown tuple, KNN looks for pattern space for the k tuples which are closest to that tuple. These k tuples become the nearest neighbors of that unknown tuple.
- To find the Euclidean Distance between two points or tuples, the formula is given below.

  Let, $Y1 = = \{y_{11}, y_{12}, y_{13}, \ldots\ldots, y_{1n}\}$ and $Y2 = = \{y_{21}, y_{22}, y_{23}, \ldots, y_{2n}\}$

$$\text{Distance}(Y1, Y2) = \sqrt{\sum_{i=1}^{n} (y_{1i} - y_{2i})^2}$$

- KNN classifiers can be extremely slow when classifying test tuples O(n).
- By simple presorting and arranging the stored tuples into search tree, the number of comparisons can be reduced to O(logN).
- Example : if k =5, it selects the 5 nearest neighbor as shown in Fig. 5.7.1.

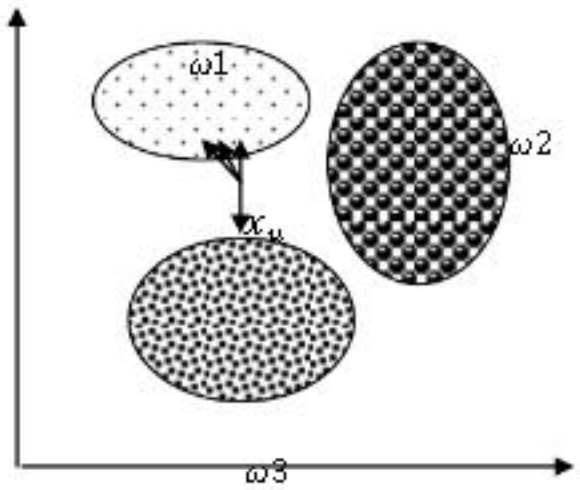


Fig. 5.7.1 KNN for k=5

Case Study :

Determining decision on scholarship application based on the following features :

- Income (in Rs)
- Number of siblings
- HSC grade
- Take the decision to award scholarships to high-performers.

- Normalize the data first.
- Apply Euclidean distance formula for the above given features.
- Combine the similar samples which are close to each other and assign the class to them.
- For unseen records, select k tuples from training data set, apply KNN and find the class, whether the student is eligible for scholarships or not.

UNIT – VI Multiclass Classification

6.1 Metrics for Performance Evaluation :

- Validation test data is very useful to estimate the accuracy of model.
- Various methods for estimating a classifier's accuracy are given below. All of them re based on randomly sampled partitions of data :
    - Holdout method
    - Random subsampling
    - Cross-validation
    - Bootstrap
- If we want to compare classifiers to select the best one then the following methods are used :
    - Confidence intervals
    - Cost-benefit analysis and Receiver Operating Characteristics (ROC) Curves.

6.1.1 Accuracy and Error Measures :

Accuracy of a classifier M, acc(M) is the percentage of test set tuples that are correctly classifier by the model M.

Basic concepts :

1. **Partition the data randomly into three sets :** Training sets, validation set and test set.
    - Training set is the subset of data used to train/build the model.
    - Test set is a set of instances that have not been used in the training process. The model's performance is evaluated on unseen data. Testing just estimates the probability of success on unknown data.
    - Validation data is used for parameter tuning but it cannot be the test data. Validation data can be the training data, or a subset of training data.
    - Generalization Error : Model error on the test data.
2. **Success :** Instance (record) class is predicted correctly.
3. **Error :** Instance class is predicted incorrectly.

4. **The confusion matrix :** It is a useful tool for analyzing how well your classifier can recognize tuples of different classes.
   - If we have only two way classification then only four classification outcomes are possible which are given in table 6.1.2 in the form of a confusion matrix.

Table 6.1.1

		Predicted class		
	Class Label	C1	C2	Total
Actual class	C1	True Positive (TP)	False Negatives (FN)	P
	C2	False Positives (FP)	True Negatives (TN)	N
	Total	P`	N`	All

   - TP : Class members which are classified as class members.
   - TN : Class non-members which are classified as non-members.
   - FP : Class non-members which are classified as class members.
   - FN : Class non-members which are classified as class non-members.
   - P : The number of positive tuples
   - N : The number of negative tuples
   - P` : The number of tuples that were labeled as positive.
   - N` : The number of tuples that were labeled as negative.
   - All : Total number of tuples i.e. TP+FN+FP+TN or P+N or P`+N`.
5. **Sensitivity :** True Positive recognition rate which is the proportion of positive tuples that are correctly identified.
6. **Specificity :** True Negative recognition rate which is the proportion of negative tuples that are correctly identified.

   Specificity = TN/N
7. **Classifier accuracy or recognition rate :** Percentage of test set tuples that are correctly classified.

   Accuracy = (TP + TN) / ALL

   OR

   $$\text{Accuracy} = \frac{(TP+TN)}{P+N}$$

Accuracy is also a function of sensitivity and specificity.

$$\textbf{Accuracy} = \textbf{Sensitivity}\ \frac{P}{(P+N)} + \textbf{Specificity}\ \frac{N}{(P+N)}$$

8. **Error rate :** A percentage of error made over the whole set of instances (records) used for testing.

Error rate = 1 – accuracy, of Error rate = (FP+FN)/ALL

OR

$$\textbf{Error rate} = \frac{FP+FN}{|TP|+|FP|}$$

9. **Precision :** Percentage of tuples which are correctly classified as positive are actual positive. It is a measure of exactness.

$$\textbf{Precision} = \frac{|TP|}{|TP|+|FP|}$$

10. **Recall :** Percentage of positive tuples which are classifier labeled as positive. It is a measure of completeness.

$$\textbf{Recall} = \frac{|TP|}{|TP|+|FP|}$$

11. **F measure(F_1 or F-score) :** Harmonic mean of precision and recall,

$$\text{F} = \frac{2 \times Precision \times Recall}{Precision+Recall}$$

12. F_β = Weighted measure of precision and recall and assigns β times as much weight to recall as to precision.

$$F \qquad \beta = \frac{2 \times Precision \times Recall}{Precision+Recall}$$

Where β is a non-negative real number.

13. **Classifiers can also be compared with respect to :**
    - Speed
    - Robustness
    - Scalability
    - Interpretability
14. **Re-substitution error rate :**
    - Re-substitution error rate is a performance measure and is equivalent to training data error rate.
    - It is difficult to get 0% error rate can be minimized, so low error rate is always preferable.

6.1.2 Holdout :

- In holdout method, data is divided into training data set and testing set and testing data set (usually 1/3 for testing, 2/3 for training).

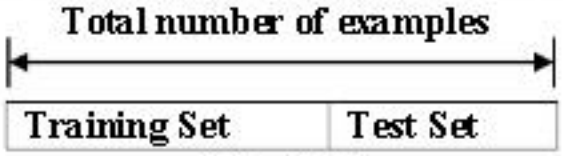


Fig. 6.1.2

- To train the classifier, training data set is used and once the classifier is constructed the use data set to estimate the error rate of the classifier.
- If the training is more than better model is constructed and if the test data is more than more accurate the error estimates.
- Problem : The samples might not be representative. For examples, some classes might be represented with very few instances or even with no instances at all.
- Solution : Stratification is the method which ensures that both training and testing data have equal number of samples of same class.

6.1.3 Random Subsampling :

- It is a variation of the holdout method.
- The holdout method is repeated k times.
- Each split randomly selects a fixed number examples without replacement.

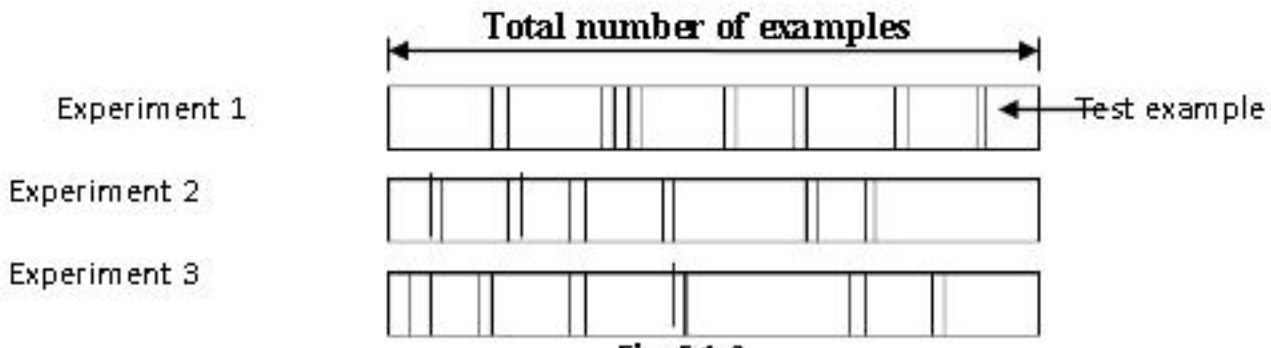


Fig. 6.1.3

- For each data split we retrain the classifier from scratch with the training examples and estimate E_i with the test examples.
- The overall accuracy is calculated by taking the average of the accuracies obtained from each iteration.

$$E = \frac{1}{K} \sum_{i=1}^{K} E_i$$

6.1.4 Cross-Validation (CV) :

- Avoids overlapping test sets.
- **k-fold cross-validation :**
  - **First step :** Data is split into k subsets of equal size (usually by random sampling).
  - **Second step :** Each subset in turn is used for testing and the remainder for training.
- The advantage is that all the examples are used for both training and testing.
- The error estimates are averaged to yield an overall error estimate.

$$E = \frac{1}{K} \sum_{i=1}^{K} E_i$$

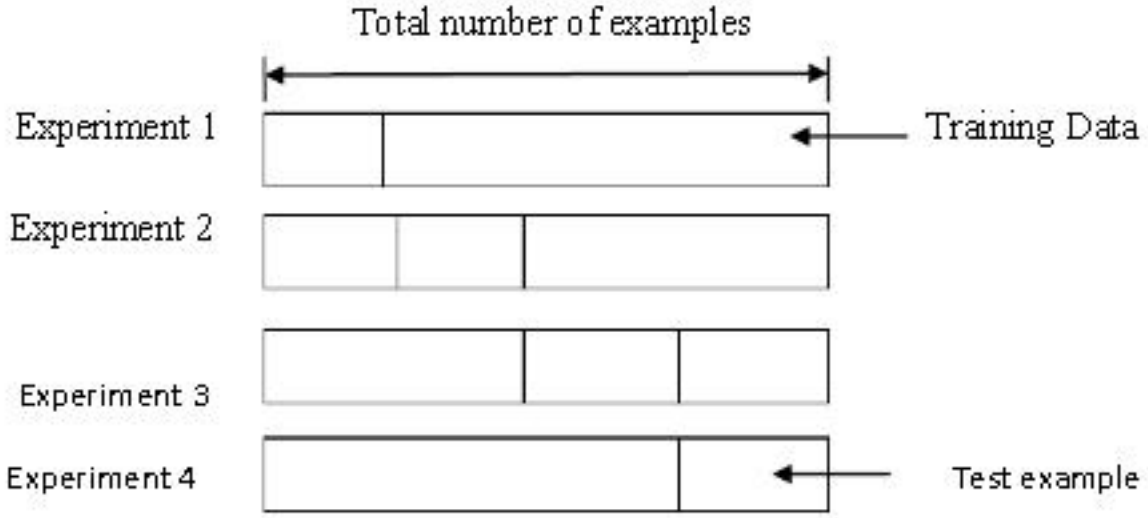


Fig. 6.1.4.1

- **Leave-one-out cross validation** :
  - If dataset has N examples, then N experiments to be performed for Leave-one-out cross validation.
  - For every experiment, training uses N-1 examples and remaining examples for testing.
- The average error rate on test examples gives the true error.

$$E = \frac{1}{N} \sum_{i=1}^{K} E_i$$

- **Stratified cross-validation :** Subsets are stratified before the cross-validation is performed.
- **Stratified ten-fold cross-validation :**
  - This gives accurate estimate of evaluation.
  - The estimate's variance get reduces due to stratification.
  - Ten-fold cross-validation is repeated ten times and finally the results are averaged based on the previous 10 results.

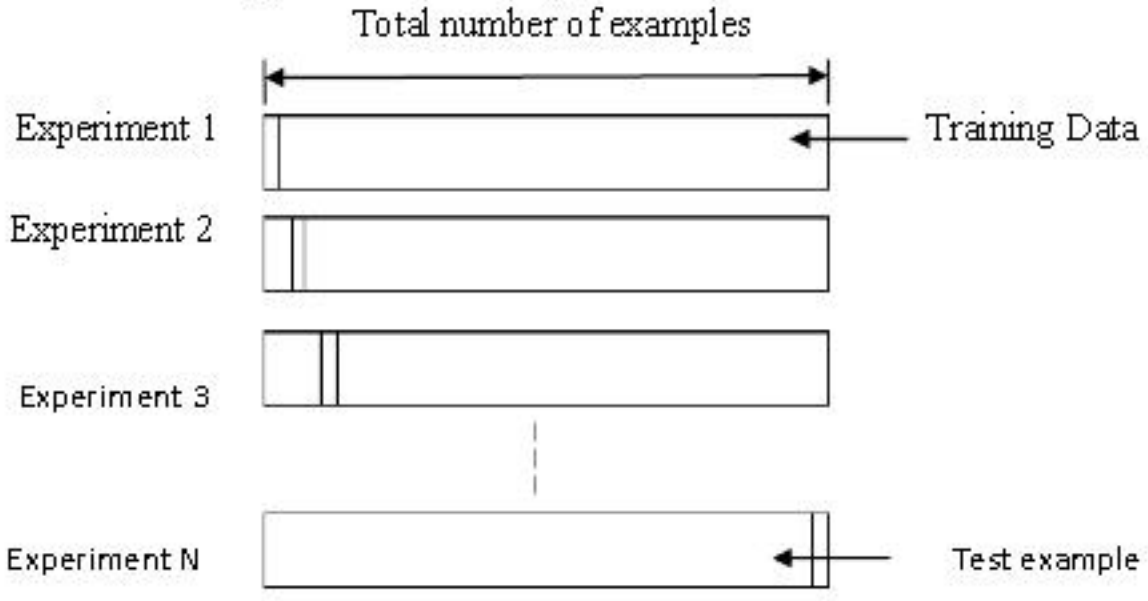


Fig. 6.1.4.2

6.1.5 Bootstapping :

- CV uses sampling of data set without replacement. Once the tuples or instances is selected, it cannot be selected again for training or test data.
- The bootstrap uses sampling with replacement to get the training set.
- **Training set :** A dataset of k instances is sampled with replacement k times to form the training set of k instances.
- **Test set :** This is separate data set from the original dataset which is not the part of training dataset.
- Bootstrapping is the best error estimator for small data sets.

6.1.6 Comparing Classifier Performance using ROC Curves :

- To compare two classification models, Receiver Operating Characteristics (ROC) curves are a useful visual tool. It shows the trade-off between the true positive rate and the false positive rate.

- The accuracy of the model is measured by the area unde4r the ROC curve.
- Tuples which are most likely to belong to positive class should appear at the top of the list, so accordingly rank all the test tuples in decreasing order.
- The closer to the diagonal line (i.e., the closer the area is to 0.5), the less accurate is the model.

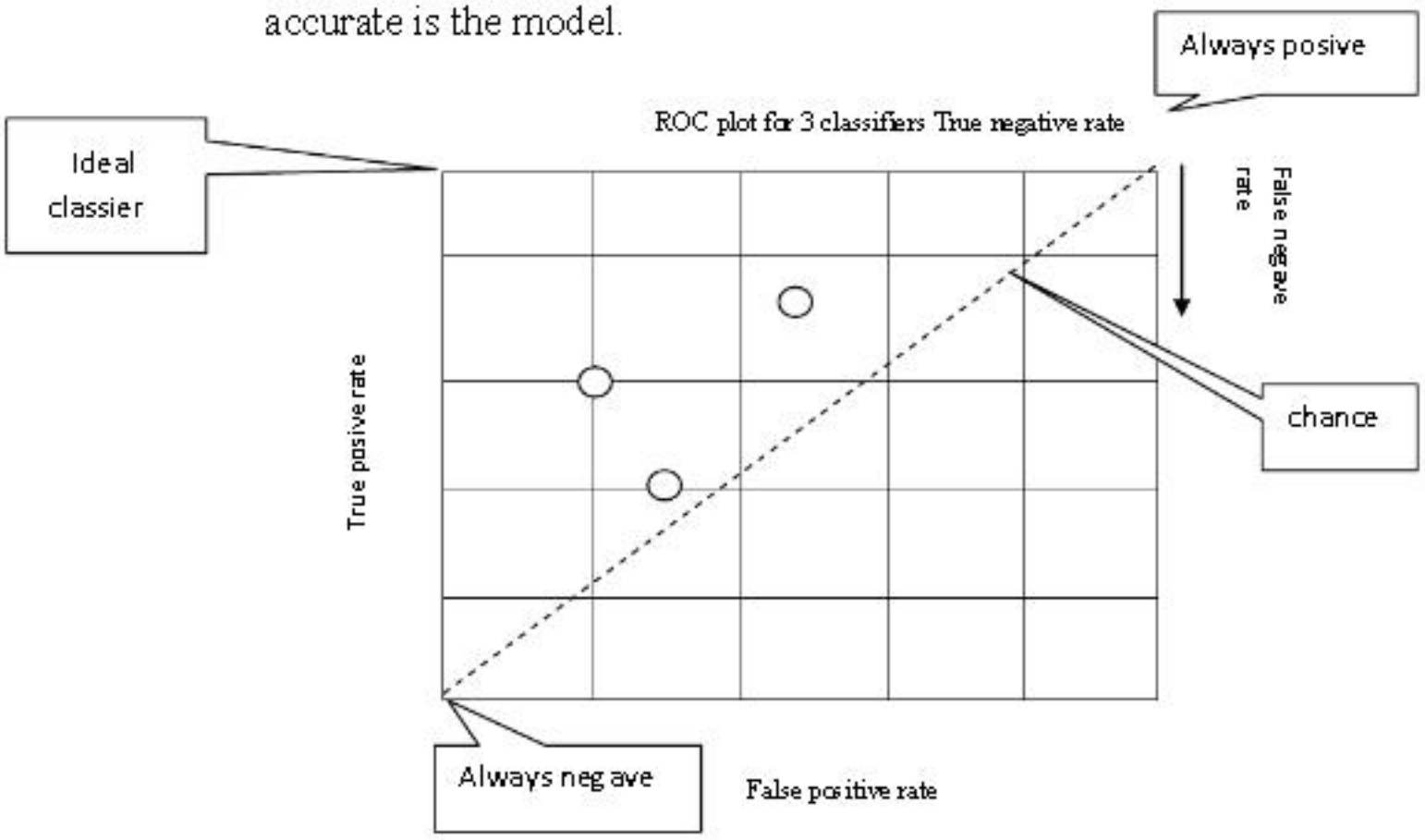


Fig 6.1.6 ROC curve for 3 model

- ROC curve plots sensitivity (true positive rate) vs. 1-specificity (false positive rate).
- Always goes from (0,0) to (1,1). The more area in the upper left, the model is better.
- Random model is on the diagonal.
- "Area Under the Curve" (AUC) is a common measure of predictive performance.