Data Mining and Business Intelligence

(Code - ITC602)

Semester VI - Information Technology

(Mumbai University)

Strictly as per the Choice Based Credit and Grading System (Revise 2016) of Mumbai University w.e.f. academic year 2018-2019

Dr. Arti Deshpande

Department of Computer Engineering

Thadomal Shahani Engineering College,

Mumbai.

Maharashtra, India.

Dr. Pallavi Halarnkar

Department of Computer Engineering

Thadomal Shahani Engineering College,

Mumbai.

Maharashtra, India.

Tech-Max Publications, Pune

Innovation Throughout Engineering Division

ME250A

Syllabus

Course Code	Course Name	Theory	Practical	Tutorial	Theory	Oral and Practical	Tutorial	Total
ITC602	Data Mining and Business Intelligence	04	--	--	04	--	--	04

Course Code	Course Name	Examination Scheme						
		Theory Marks				Term Work	Oral and Practical	Total
		Internal assessment			End Sem. Exam			
		Test 1	Test 2	Avg. of two Tests				
ITC602	Data Mining and Business Intelligence	20	20	20	80	--	--	100

Course Objectives : Students will try :

1. To introduce the concept of data Mining as an important tool for enterprise data management and as a cutting edge technology for building competitive advantage.
2. To enable students to effectively identify sources of data and process it for data mining.
3. To make students well versed in all data mining algorithms, methods of evaluation.
4. To impart knowledge of tools used for data mining.
5. To provide knowledge on how to gather and analyze large sets of data to gain useful business understanding.
6. To impart skills that can enable students to approach business problems analytically by identifying opportunities to derive business value from data.

Course Outcomes : Students will be able to :

1. Demonstrate an understanding of the importance of data mining and the principles of business intelligence.
2. Organize and Prepare the data needed for data mining using preprocessing techniques.
3. Perform exploratory analysis of the data to be used for mining.
4. Implement the appropriate data mining methods like classification, clustering or Frequent Pattern mining on large data sets.
5. Define and apply metrics to measure the performance of various data mining algorithms.
6. Apply BI to solve practical problems : Analyze the problem domain, use the data collected in enterprise apply the appropriate data mining technique, interpret and visualize the results and provide decision support.

Prerequisite : Database Management System, Advanced Data Management Technology.

Detailed syllabus

Sr. No.	Module	Detailed Content	Hours	CO Mapping
0	Prerequisites	Knowledge of databases, and Data warehousing, OLAP.	02	--
I	Introduction to Data Mining	What is Data Mining; Kind of patterns to be mined; Technologies used; Major issues in Data Mining. **(Refer Chapter 1)**	03	CO1
II	Data Exploration and Data Preprocessing	Types of Attributes; Statistical Description of Data; Data Visualization; Measuring similarity and dissimilarity. Why Preprocessing? Data Cleaning; Data Integration; Data Reduction : Attribute subset selection, Histograms, Clustering and Sampling; Data Transformation & Data Discretization : Normalization, Binning, Histogram Analysis and Concept hierarchy generation. **(Refer Chapter 2)**	09	CO2 CO3

Sr. No.	Module	Detailed Content	Hours	CO Mapping
III	Classification	Basic Concepts; Classification methods : 1. Decision Tree Induction : Attribute Selection Measures, Tree pruning. 2. Bayesian Classification : Naïve Bayes Classifier. Prediction : Structure of regression models; Simple linear regression, Multiple linear regression. Accuracy and Error measures, Precision, Recall, Holdout, Random Sampling, Cross Validation. **(Refer Chapter 3)**	09	CO4 CO5
IV	Clustering	Cluster Analysis : Basic Concepts; Partitioning Methods : K-Means, K-Mediods; Hierarchical Methods : Agglomerative, Divisive, BIRCH; Density-Based Methods : DBSCAN What are outliers? Types, Challenges; Outlier Detection Methods : Supervised, Semi Supervised, Unsupervised, Proximity based, Clustering Based. **(Refer Chapter 4)**	10	CO4 CO5
V	Frequent Pattern Mining	Market Basket Analysis, Frequent Itemsets, Closed Itemsets, and Association Rules; Frequent Pattern Mining, Efficient and Scalable Frequent Itemset Mining Methods, The Apriori Algorithm for finding Frequent Itemsets Using Candidate Generation, Generating Association Rules from Frequent Itemsets, Improving the Efficiency of Apriori, A pattern growth approach for mining Frequent Itemsets; Mining Frequent itemsets using vertical data formats; Introduction to Mining Multilevel Association Rules and Multidimensional Association Rules; From Association Mining to Correlation Analysis, lift; Introduction to Constraint-Based Association Mining. **(Refer Chapter 5)**	10	CO4 CO5
VI	Business Intelligence	What is BI? Business intelligence architectures; Definition of decision support system; Development of a business intelligence system using Data Mining for business Applications like Fraud Detection, Clickstream Mining, Market Segmentation, retail industry, telecommunications industry, banking & finance CRM etc. **(Refer Chapter 6)**	09	CO6

□□□

❑❑❑

CHAPTER 1

Introduction to Data Mining

Syllabus

What is Data Mining; Kind of patterns to be mined; Technologies used; Major issues in Data Mining.

Syllabus Topic : What is Data Mining ?

1.1 What is Data Mining?

- Data Mining is a new technology, which helps organizations to process data through algorithms to uncover meaningful patterns and correlations from large databases that may otherwise be not possible with standard analysis and reporting.
- Data mining tools can help to understand the business better and also improve future performance through predictive analytics and make them proactive and allow knowledge driven decisions.
- Issues related to information extraction from large databases, data mining field brings together methods from several domains like Machine Learning, Statistics, Pattern Recognition, Databases and Visualization.
- Data mining field finds its application in market analysis and management like for e.g. customer relationship management, cross selling, market segmentation. It can also be used in risk analysis and management for forecasting, customer retention, improved underwriting, quality control, competitive analysis and credit scoring.

1.1.1 Definition

MU - May 2016, Dec. 2016

- Data mining is processing data to identify patterns and establish relationships.
- Data mining is the process of analyzing large amounts of data stored in a data warehouse for useful information which makes use of artificial intelligence techniques, neural networks, and advanced statistical tools (such as cluster analysis) to reveal trends, patterns and relationships, which otherwise may be undetected.

- Data Mining is a non-trivial process of identifying :
  - Valid
  - Novel
  - Potentially useful, understandable patterns in data.

1.1.2 Applications of Data Mining

MU - May 2016, Dec. 2016

Data Mining has been used in numerous areas, which include both private as well as public sectors.

- The use of Data mining in major industry areas like Banking, Retail, Medicine, insurance can help reduce costs, increase their sales and enhance research and development.
- For example in banking sector data mining can be used for customer retention, fraud prevention by credit card approval and fraud detection.
- Prediction models can be developed to help analyze data collected over years. For e.g. customer data can used to find out whether the customer can avail loan from the bank, or an accident claim is fraudulent and needs further investigation.
- Effectiveness of a medicine or certain procedure may be predicted in medical domain by using data mining.
- Data mining can be used in Pharmaceutical firms as a guide to research on new treatments for diseases, by analyzing chemical compounds and genetic materials.
- A large amount of data in retail industry like purchasing history, transportation services may be collected for analysis purpose. This data can help multidimensional analysis, sales campaign effectiveness, customer retention and recommendation of products and much more.
- Telecommunication industry also uses data mining, for e.g. they may do analysis based on the customer data which of them are likely to remain as subscribers and which one will shift to competitors.

1.1.3 KDD Process (Knowledge Discovery in Databases)

MU - Dec. 2015

- The process of discovering knowledge in data and application of data mining methods refers to the term **Knowledge Discovery in Databases (KDD).**
- It includes a wide variety of application domains, which include Artificial Intelligence, Pattern Recognition, Machine Learning Statistics and Data Visualization.
- The main goal includes extracting knowledge from large databases, the goal is achieved by using various data mining algorithms to identify useful patterns according to some predefined measures and thresholds.

Outline steps of the KDD process

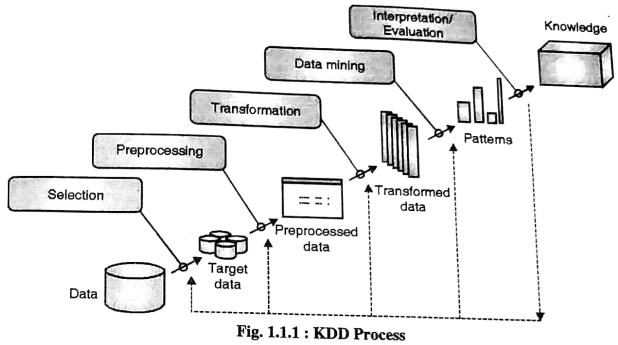


Fig. 1.1.1 : KDD Process

The overall process of finding and interpreting patterns from data involves the repeated application of the following steps:

1. **Developing an understanding of :**
   (1) The application domain
   (2) The relevant prior knowledge
   (3) The goals of the end-user.
2. **Creating a target data set :** Selecting a data set, or focusing on a subset of variables, or data samples, on which discovery is to be performed.
3. **Data cleaning and pre-processing:**
   (1) Noise or outliers are removed.
   (2) Essential information is collected for modelling or accounting for noise.
   (3) Missing data fields are handled by using appropriate strategies.
   (4) Time sequence information and changes are maintained.
4. **Data reduction and projection :**
   (1) Based on the goal of the task, useful features are found to represent the data.
   (2) The number of variables may be effectively reduced using methods like dimensionality reduction or transformation. Invariant representations for the data may also be found out.
5. **Choosing the data mining task :**
   Selecting the appropriate Data mining tasks like classification, clustering, regression based on the goal of the KDD process.

6. **Choosing the data mining algorithm(s) :**
   (1) Pattern search is done using the appropriate Data Mining method(s).
   (2) A decision is taken on which models and parameters may be appropriate.
   (3) Considering the overall criteria of the KDD process a match for the particular data mining method is done.
7. **Data mining:**
   Using a representational form or other representations like classification, rules or trees, regression clustering for searching patterns of interest.
8. **Interpreting mined patterns**
9. **Consolidating discovered knowledge**

The terms **knowledge discovery** and **data mining** are distinct.

KDD	Data Mining
KDD is a field of computer science, which helps humans in extracting useful, previously undiscovered knowledge from data. It makes use of tools and theories for the same.	Data Mining is one of the step in the KDD process, it applies the appropriate algorithm based on the goal of the KDD process for identifying patterns from data.

1.1.4 Architecture of a Typical Data Mining System

MU - Dec. 2015

Architecture of a typical data mining system may have the following major components as shown n Fig. 1.1.2.

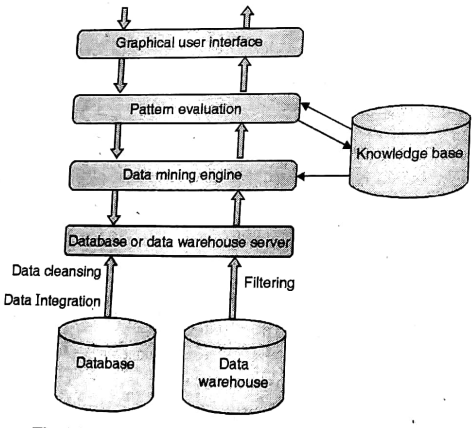


Fig. 1.1.2 : Architecture of typical data mining system

1. **Database, data warehouse, or other information repository**
   These are information repositories. Data cleaning and data integration techniques may be performed on the data.
2. **Databases or data warehouse server**
   It fetches the data as per the user's requirement which is need for data mining task.
3. **Knowledge base**
   This is used to guide the search, and gives the interesting and hidden patterns from data.
4. **Data mining engine**
   It performs the data mining task such as characterization, association, classification, cluster analysis etc.
5. **Pattern evaluation module**
   It is integrated with the mining module and it helps in searching only the interesting patterns.
6. **Graphical user interface**
   This module is used to communicate between user and the data mining system and allow users to browse database or data warehouse schemas.

Syllabus Topic : Kind of Patterns to be Mined

1.2 Kind of Patterns to be Mined

1.2.1 Data Mining Functionalities

(1) Association

- The items or objects in relational databases, transactional databases or other information repositories are considered for finding frequent patterns, associations, correlations, or causal 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 commonly purchased together. This knowledge can be used in advertising or for goods placement in stores.
- Association rules have the general form

$$I_1 \rightarrow I_2 \text{ (where } I_1 \cap I_2 = 0)$$

  Where, I_n are sets of items, for example can be purchased in a store.
- The rule should be read as "Given that someone has bought the items in the set I_1 they are likely to also buy the items in the set I_2".

(2) Classification

- Maps data into predefined groups or classes and searches for new patterns.

- Classification is a data mining technique used for systematic placement of group membership for data. For example, you may wish to use classification to predict whether the weather on a particular day will be "sunny", "rainy" or "cloudy". Popular classification techniques include decision trees and neural networks.

(3) Regression

Used to predict for individuals on the basis of information gained from a previous sample of similar individuals.

Example :

(i) A person wants to do some savings for future and then it will be based on his current values and several past values. He uses a linear regression formula to predict his future savings.

(ii) It may also be used in modelling the effect of doses in medicines or agriculture, response of a customer to a mail and evaluate the risk that the client will not pay back the loan taken from the bank.

(4) Cluster analysis

- Clustering is a data mining (machine learning) technique used to place data elements into related groups without advance knowledge of the group definitions.
- Maximizing intra-class similarity and minimizing interclass similarity.
- **Example :** A department store chain creates special catalogs targeted to various types of customer groups based on attributes such as income, location, etc.

(5) Forecasting

Discovering patterns in data that can lead to reasonable predictions about the future.

(6) Multi-dimensional concept description

- Characterization and discrimination.
- Generalize, summarize and contrast data characteristics, e.g. dry vs. wet regions.
- Concept hierarchies are used to reduce the amount of data by collecting and replacing low level detailed data (such as numeric values for the attribute age) by higher level data (such as young, middle-aged, or senior).

(7) Outlier analysis

- **Outlier :** Data object that does not comply with the general behavior of the data.
- It is useful in fraud detection, rare events analysis.
- Based on the above functionality Data Mining Tasks can be classified into two categories:

(a) **Descriptive mining :** To derive patterns like correlation, trends etc. which summarizes the underlying relationship between data.

Example : Identifying items which are purchased together frequently.

Some of Descriptive mining techniques :

- Class/Concept description
- Mining of frequent patterns
- Mining of associations
- Mining of correlations
- Mining of clusters

(b) **Predictive mining :** Predict the value of a specific attribute based on the value of other attributes.

Example: Predict the next year's profit or loss.

Some of Predictive Mining techniques:

- Classification (IF-THEN) Rules
- Decision Trees
- Mathematical Formulae
- Neural Networks

Syllabus Topic : Technologies Used

1.3 Technologies Used

Many techniques that strongly influence the development of data mining methods. Some of such technologies are given below :

1.3.1 Statistics

- Statistics is a discipline of science, which uses mathematical analysis to quantify representations, model and summarize empirical data or real world observations.
- Statistical analysis involves collection of methods and applying them on large amounts of data to draw conclusions and report the trend.

1.3.2 Machine Learning

- Machine learning is a type of artificial intelligence that provides computers with the ability to learn without being explicitly programmed. When new data is exposed, computer programs can teach themselves to grow or change due to machine learning.
- For example, Facebook's News Feed changes according to the user's personal interactions with other users.

1. **Supervised learning :** One standard formulation of the supervised learning task is the classification problem. In classification, the training dataset is having labelled examples. Based on training dataset, classification model is constructed which is used to give label to unseen data. Neural networks and decision trees techniques are highly dependent on the

previous information of classification where the classes are pre-determined by classifications techniques.

2. **Unsupervised learning :** One standard formulation of the unsupervised learning task is the Clustering problem where the examples of training dataset are not labelled. Based on similarity measures, clusters are formed.
3. **Semi-supervised learning :** This learning technique combines both labelled and unlabeled examples to generate an appropriate function or classifier. This method generally avoids the large number of labelled examples.
4. **Active learning :** The user play an important role in active learning. The algorithm itself decides which thing you should label. Active learning is a powerful approach in analyzing data effectively.

1.3.3 Information Retrieval (IR)

Information Retrieval deals with uncertainty and vagueness in information systems :

- Uncertain representations of the semantics of objects (text, images,...).
- Vague specifications of information need (iterative querying).
- **Example :** Find documents *relevant* to an information need from a large document set.

1.3.4 Database Systems and Data Warehouses

Database

1. Databases are used to record the data and also be used for data warehousing. Online Transactional Processing (OLTP) uses databases for day to day transaction purpose.
2. To remove the redundant data and save the storage space, data is normalized and stored in the form of tables.
3. Entity-Relational modeling techniques are used for Relational Database Management System design
4. Write operation on databases are optimized in databases.
5. Query writing is complex so performance is low for query analysis.

Data warehouse

1. Data Warehouses are 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.
2. Data is de-normalized so tables are not complex and reduces the response time for analytical queries.
3. Data-modeling techniques like star schema are used for the Data Warehouse design.
4. Read operation on data warehouse are optimized as it has been used frequently for analysis purpose so performance is high for queries.

1.3.5 Decision Support System

- Decision Support system is a category of information systems, which helps in decision making for business and organizations.
- It is an interactive software based systems which helps decision makers to extract useful information from raw data, documents and take appropriate decisions, modelling business by identifying problems and solving them.
- **Information that is gathered and presented by a DSS :**
  - A list Information assets like legacy, relational data sources, cubes, data warehouse, data marts.
  - Comparative statements of sales.
  - Projected revenue based on assumptions of new product sales.

Syllabus Topic : Major Issues in Data Mining

1.4 Major Issues in Data Mining

- **Mining methodology and user interaction issues**
  - Mining different kinds of knowledge in database.
  - Interactive mining of knowledge at multiple levels of abstraction.
  - Incorporation of background knowledge.
  - Data mining query language and ad hoc data mining.
  - Presentation and visualization of data mining results.
  - Handling noisy or incomplete data.
  - Pattern Evaluation.
- **Performance issues**
  - Efficiency and scalability of data mining algorithms.
  - Parallel, distributed and incremental mining algorithm.
- **Issues relating to the diversity of database types**
  - Handling of relational and complex types of data.
  - Mining information from heterogeneous databases and global information system.

Review Questions

Q. 1 What are the major issues in data mining ?

Q. 2 Explain the KDD process in detail.

Q. 3 Explain architecture of a typical data mining system.

1.5 University Questions and Answers

Dec. 2015

Q. 1 Explain data mining as a step in KDD. Give the architecture of typical Data Mining system. **(10 Marks)**
(Ans. : Refer Sections 1.1.3 and 1.1.4)

May 2016

Q. 2 Define "Data Mining". Enumerate five example applications that can benefit by using Data Mining. *(Ans. : Refer Sections 1.1.1 and 1.1.2)* **(5 Marks)**

Dec. 2016

Q. 3 Define "Data Mining". Enumerate five example applications that can benefit by using Data Mining. *(Ans. : Refer sections 1.1.1 and 1.1.2)* **(5 Marks)**

...Chapter Ends

❑❑❑

CHAPTER 2

Data Exploration and Data Preprocessing

Syllabus

Types of Attributes; Statistical Description of Data; Data Visualization; Measuring similarity and dissimilarity. Why Preprocessing? Data Cleaning; Data Integration; Data Reduction: Attribute subset selection, Histograms, Clustering and Sampling; Data Transformation& Data Discretization: Normalization, Binning, Histogram Analysis and Concept hierarchy generation.

Syllabus Topic : Types of Attributes

2.1 Types of Attributes

Data Objects

- A data object is a logical cluster of all tables in the data set which contains data related to the same entity. It also represents an object view of the same.
- **Example:** In a product manufacturing company, product, customer are objects. In a retail store, employee, customer, items and sales are objects.
- Every data object is described by its properties called as attributes and it is stored in the database in the form of a row or tuple. The columns of this data tuple are known to be attributes.

Attributes Types MU May 2015

- An attribute is a property or characteristic of a data object. For e.g. Gender is a characteristic of a data object person.
- The attributes may have values like :

(i) Nominal attributes (ii) Binary attributes
(iii) Ordinal attributes (iv) Numeric attributes
(v) Discrete versus continuous attributes

(I) Nominal attributes

- Nominal attributes are also called as Categorical attributes and allow for only qualitative classification.
- Every individual item has a certain distinct categories, but quantification or ranking the order of the categories is not possible.
- The nominal attribute categories can be numbered arbitrarily.
- Arithmetic and logical operations on the nominal data cannot be performed.
- **Typical examples of such attributes are:**

Car owner :	1. Yes 2. No
Employment status :	1. Unemployed 2. Employed

(II) Binary attributes

- A nominal attribute which has either of the two states 0 or 1 is called Binary attribute , where 0 means that the attribute is absent and 1 means that it is present.
- **Symmetric binary variable :**If both of its states i.e. 0 and 1 are equally valuable. Here we cannot decide which outcome should be 0 and which outcome should be 1.**Example:** Marital status of a person is "Married or Unmarried". In this case both are equally valuable and difficult to represent in terms of 0(absent) and 1(present).
- **Asymmetric binary variable :**If the outcome of the states are not equally important. An example of such a variable is the presence or absence of a relatively rare attribute. **Example :** Person is "handicapped or not handicapped". The most important outcome is usually coded as 1 (present) and the other is coded as 0 (absent).

(III) Ordinal attributes

- A discrete ordinal attribute is a nominal attribute, which have meaningful order or rank for its different states.
- The interval between different states is uneven due to which arithmetic operations are not possible, however logical operations may be applied.
- **Example :** Considering Age as an ordinal attribute, it can have three different states based on an uneven range of age value. Similarly income can also be considered as an ordinal attribute, which is categorised as low, medium, high based on the income value.

Age :	1. Teenage 2. Young 3. Old
Income :	1. Low 2. Medium 3. High

(Iv) Numeric attributes

Numeric Attributes are quantifiable. It can be measured in terms of a quantity, which can either have an integer or real value. They can be of two types,

1. Interval scaled attributes
2. Ratio scaled attributes

1. **Interval-scaled attributes**

Interval-scaled attributes are continuous measurement on a linear scale. **Example :** Weight, height and weather temperature. These attributes allow for ordering, comparing and quantifying the difference between the values. An interval-scaled attributes has values whose differences are interpretable.

2. **Ratio-scaled attributes**

- Ratio scaled attributes are continuous positive measurements on a non linear scale. They are also interval scaled data but are not measured on a linear scale.
- Operations like addition, subtraction can be performed but multiplication and division are not possible.
- **Example :** For instance, if a liquid is at 40 degrees and we add 10 degrees, it will be 50 degrees. However, a liquid at 40 degrees does not have twice the temperature of a liquid at 20 degrees because 0 degrees does not represent "no temperature"
- There are three different ways to handle the ratio-scaled variables :
  - As interval scale variables. The drawback of handling them as interval scaled is that it can distort the results.
  - As continuous ordinal scale.
  - Transforming the data (for example, logarithmic transformation) and then treating the results as interval scaled variables.

2.1.1 Numeric Attributes

Numeric Attributes are quantifiable. It can be measured in terms of a quantity, which can either have an integer or real value. They can be of two types,

1. Interval scaled attributes
2. Ratio scaled attributes

1. **Interval-scaled attributes**

Interval-scaled attributes are continuous measurement on a linear scale. **Example :** weight, height and weather temperature. These attributes allow for ordering, comparing and quantifying the difference between the values. An interval-scaled attributes has values whose differences are interpretable.

2. ***Ratio-scaled attributes***
   - Ratio scaled attributes are continuous positive measurements on a non linear scale. They are also interval scaled data but are not measured on a linear scale.
   - Operations like addition, subtraction can be performed but multiplication and division are not possible.
   - **Example :** For instance, if a liquid is at 40 degrees and we add 10 degrees, it will be 50 degrees. However, a liquid at 40 degrees does not have twice the temperature of a liquid at 20 degrees because 0 degrees does not represent "no temperature"
   - There are three different ways to handle the ratio-scaled variables :
     - As interval scale variables. The drawback of handling them as interval scaled is that it can distort the results.
     - As continuous ordinal scale.
     - Transforming the data (for example, logarithmic transformation) and then treating the results as interval scaled variables.

(V) Discrete versus continuous attributes

If an attribute can take any value between two specified values then it is called as continuous else it is discrete. An attribute will be continuous on one scale and discrete on another.

Example : If we try to measure the amount of water consumed by counting the individual water molecules then it will be discrete else it will be continuous.

- **Examples of continuous attributes** includes time spent waiting, direction of travel, water consumed etc.
- **Examples of discrete attributes** include voltage output of a digital device, a person's age in years.

Syllabus Topic : Statistical Description of Data

2.2 Statistical Description of Data

Statistical description of data is useful in finding the properties of data which can help identify in finding whether a particular data is noise or an outlier.

Following are some of the measures which can be used for measuring the statistical properties of the data.

2.2.1 Central Tendency

MU - May 2016

Central tendency is also known as measure of central location that describes the central position within the set of data. The various measure of central tendency is :

(I)	Mean
(ii)	Median
(iii)	Mode
(iv)	Midrange

Based on conditions, a suitable measure is applied to calculate central tendency.

(i) Mean

- It is one of the most common measure of central tendency. It can be used with both continuous and discrete attributes. It is mostly used with continuous attributes. Mean is equal to the sum of the all the values in a data set divided by the total number of values in the data set.
- Mean is given by $\bar{x}$ (pronounced as x bar)

$$\bar{x} = \frac{(x_1 + x_2 + \ldots + x_n)}{n}$$

Or

$$\mu = \frac{\Sigma x}{n}$$

Where, μ represents the mean.

- For example the mean for the following data is,

85	55	89	66	25	14	96	78	87	45	92

$$\mu = 64.7$$

- Mean is the only measure of central tendency in which the sum of the deviations of each value from mean is always zero.
- If the weights are associated with the value x_i then weighted arithmetic mean or the weighted average is,

$$\bar{x} = \frac{\sum_{i=1}^{n} w_i x_i}{\sum_{i=1} w_i}$$

- Mean has one disadvantage; it is highly susceptible to the influence of outliers. Under such types of situations median would be a better measure of central tendency.

(ii) Median

- This measure is suitable when the data is skewed (frequency distribution of data is skewed). As the data becomes skewed, it loses its ability to provide a central position under such a situation median can be used as it is not influenced by the skewed values.

- Median is the middle score for a data set, which has been arranged in the order of magnitude (smallest first).
- For example if we have the following data set :

85	55	89	66	25	14	96	78	87	45	92

Rearranging the above data in the order of magnitude:

14	25	45	55	66	78	85	87	89	92	96

The median for the above data is 78. It is the middle position element as there are five scores before it and five scores after it.

- The above example is for odd number of elements in the data set, but in case if we have even number of elements for example :

14	25	45	55	66	78	85	87	89	92	96	100

In the above case an average value of two values i.e. 78 and 85 is considered as median of the above data set. i.e. 81.5

- The median is a useful number in cases where the distribution has very large extreme values, which would otherwise skew the data.

$$\text{Median} = L_1 + \left(\frac{n/2 - (\sum f)l}{f_{median}}\right)C$$

Where,

L_1 = Lower class boundary

n = Number of values in the data

$(\sum f)l$ = The sum of the frequencies of all of the classes that are lower than the median class

f_{median} = Frequency of median class

C = Size of the median class interval

(iii) Mode

- The mode is the most frequently occurring value in the data set.
- If we consider a histogram then it is the highest bar in the chart. Mostly mode is used for categorical data, where the most common category is to be known.
- Data set with one mode is called uni-modal.
- Data sets with two modes are called bimodal.
- Data sets with three modes are called tri-modal.
- The empirical relation is :

$$\text{mean} - \text{mode} = 3 \times (\text{mean} - \text{median})$$

(iv) Midrange

The midrange is the average of the largest and smallest value in a data set.

2.2.2 Dispersion of Data

MU - May 2016

The degree to which numeric data tend to spread is called the dispersion or variance of the data.

(i) **Quartiles :** Q_1 (25^{th} percentile), Q_3 (75^{th} percentile)

(ii) **Inter-Quartile Range (IQR) :** The distance between the first and third quartile is a simple measure of spread that gives the range covered by the middle half of the data.

$$IQR = Q_3 - Q_1$$

(iii) **Five number summary :** A fuller summery of the shape of a distribution can be obtained by providing highest and lowest data values. This is known as five number summary and written in order as,

min, Q_1, M, Q_3, max

Where, M = Median

Q_1, Q_2 = Quartile

Min = Smallest individual observation

Max = Largest individual observation

(iv) **Box-plot :**

MU - May 2016

- The visual representation of a distribution is the box-plot.
- Box plots used for assigning location and variation information in data sets. It also detects and shows the location and variation changes between dissimilar groups of data.
- Box plots have Vertical axis which gives response variable and Horizontal axis which gives the factor of interest
- To create box plot, order the data in ascending order

Example : 7,8,3,1,5,6,10,13,5

1	3	5	5	6	7	8	10	13

- Calculate the median of that data which divides the data into two halves.

1	3	5	5	6	7	8	10	13

The median is $Q_2 = 6$

- Again find the median of those two halves which divides data into quarters.

1	3	5	5

7	8	10	13

- Calculate the median of both the halves as there are 4 values, the median is average of two middle values :

$$Q_1 = (3+5)/2 = 4$$

Similarly for second half,

$$Q_3 = (8+10)/2 = 9$$

- Now there are three points, Q_2(the first middle point), Q_1,Q_3 (the middle point of two halves).
- These three points divides the whole data set into quarters which are called as "quartiles".
- The minimum value is1and the maximum value is13, so we have:

  Min:1, Q_1: 4, Q_2: 6, Q_3: 9, max:13

  Then the box plot looks like this :

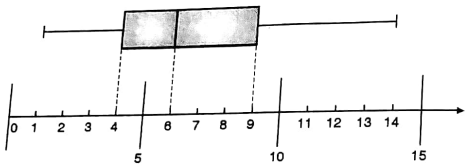


Fig. 2.2.1 : Box plot for dataset

(v) Outlier : Usually, a value higher/lower than 1.5 x IQR (**Inter-Quartile Range**)

(vi) Variance : The variance of n observations $x_1,x_2,\ldots\ldots,x_n$ is

$$s^2 = \frac{1}{n-1}\sum_{i=1}^{n}(x_i-\bar{x})^2 = \frac{1}{n-1}\left[\sum_{i=1}^{n}x_i^2 - \frac{1}{n}\left(\sum_{i=1}^{n}x_i\right)^2\right]$$

(vii) Standard deviation s is the square root of variance s^2

Measures spread about the mean. It is zero if and only if all the values are equal.

2.2.3 Graphic Displays of Basic Statistical Descriptions of Data

(i)	Bar charts, pie charts, line graphs
(ii)	Box-plots
(iii)	Histograms
(iv)	Quantile plots
(v)	Quantile-Quantile plots (Q-Q plots)
(vi)	Scatter plots
(vii)	Loess curves

(i) Bar charts, pie charts, line graphs

- **Line graph :** Line graphs are used to track changes over short and long periods of time.
- **Pie Chart :** Pie charts are best to use when you are trying to compare parts of a whole. They do not show changes over time.

- **Bar Chart :** Bar graphs are used to compare things between different groups or to track changes over time.

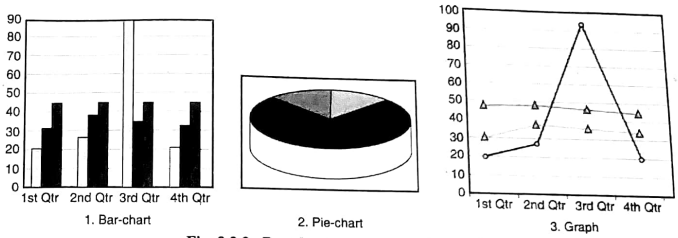


Fig. 2.2.2 : Bar charts, pie charts, line graphs

(ii) Box-plots

The information about box-plots is already given in Section 2.2.2.

(iii) Histograms

- Provide a first look at a univariate data distribution, i.e., a distribution of values of a single attribute.
- Consists of a set of rectangles that reflect the counts or frequencies of the classes present in the given data.
- **Example :** Distribution of salaries of the Acme Corporation

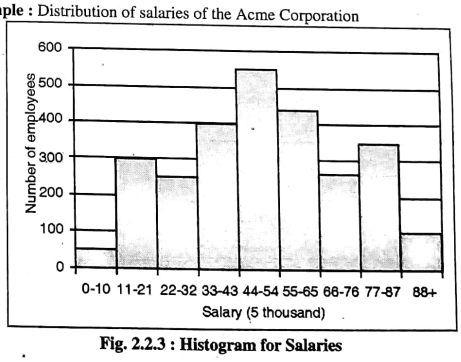


Fig. 2.2.3 : Histogram for Salaries

- A vertical bar graph and a histogram differ in these ways :
  - In a histogram, frequency is measured by the area of the column.
  - In a vertical bar graph, frequency is measured by the height of the bar.

(iv) Quantile plots

- The normal quantile plot is used to compare the data values with the values that can be predicted for a standard normal distribution.
- To draw normal quantile plot, compute two additional numbers for each value of variable.

1. Sort the data value from lowest to highest and in the column labeled x(i) place the data values is ascending order.
2. Next column shows the quartile for each data value representation.
3. Now calculate the value of the standard normal distribution that lies at the quantile just computed in previous column.
4. A normal quantile plot is formed by plotting each data value against the value of the standard normal distribution.

Example

i	x(i)	Quantile = i – 0.5/n	z = (x(i) – u)/σ
1	– 1.2	0.05	– 1.272108844
2	– 0.4	0.15	– 0.727891156
3	– 0.3	0.25	– 0.659863946
4	– 0.2	0.35	– 0.591836735
5	0	0.45	– 0.455782313
6	0.1	0.55	– 0.387755102
7	0.6	0.65	– 0.047619048
8	2.7	0.75	1.380952381
9	2.6	0.85	1.31292517
10	2.8	0.95	1.448979592

mean u = 0.67

StdDev σ = 1.471998

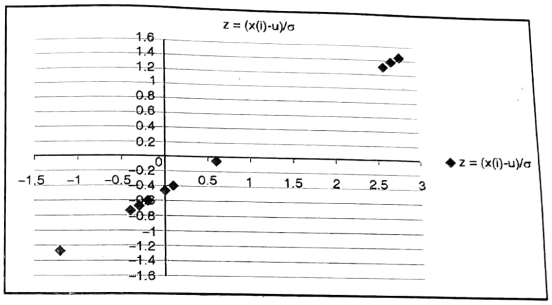


Fig. 2.2.4 :Quantile plot of z-score

(v) Quantile-Quantile plots (Q-Q plots)

- Quantile-quantile plots are useful to compare the quantiles of two sets of numbers.
- As compared to mean and medians, the comparison using Q-Q plots is more detailed but it needs more samples for comparison.
- One of the quantiles is your sample observations placed in ascending order.

Example :

- Consider marks of 30 students
- Order the marks in ascending order from smallest to largest. That will be your y-axis values.
- If the marks are normally distributed, then x-axis values would be the quantiles of a standard Normal distribution
- As we have marks of 30 students, we will need 30 Normal quantiles.
- Finally you plot marks versus the z-values.
- So a Q-Q plot is used to determine how well a theoretical distribution models a set of measurements.

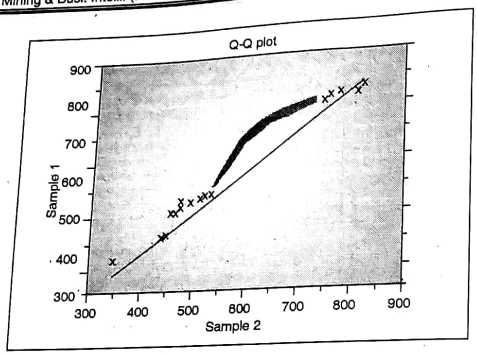


Fig. 2.2.5 :Quantile-Quantile plot for two samples

(vi) Scatter plot

- Provide a first look at bivariate data to see clusters of points, outliers, etc.
- Each pair of values is treated as a pair of coordinates and plotted as points in the plane.
- Scatter plots tell the relationships between two variables : Response variable is Y and it is usually on vertical axis.
- Variable which is related to response is X and it is on horizontal axis.
- **Example :** Assume that during a three-hour period spent outside, a person recorded the temperature and their water consumption. The experiment was conducted on 7 randomly selected days during the summer. The data is shown in the Table 2.2.1.

Table 2.2.1 : Temperature and Water consumption per day

Day	Temperature (F)	Water Consumption (oz)
1	99	48
2	85	27
3	97	48
4	75	16
5	92	32
6	85	25
7	83	20

- Corresponding Scatter plot of Water Consumption based on Temperature is given in the Fig. 2.2.6 :

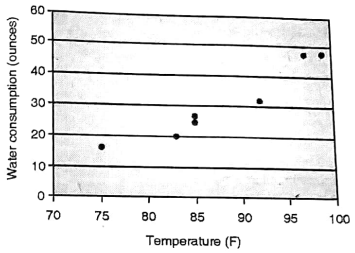


Fig. 2.2.6 : Scatter plot for Temperature and Water Consumption

(vii) Loess curve

- Loess curve is used for-fitting a smooth curve between two variables
- The procedure originated as LOWESS (Locally Weighted Scatter-plot Smoother).
- It is also called local regression.
- To get the better perception of pattern, a smooth curve is added in scatter plot as shown in Fig. 2.2.7.

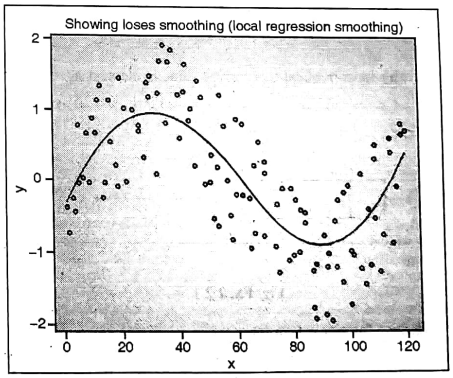


Fig. 2.2.7 : Loess Curve

Ex. 2.2.1 For the same set of data points

Data 11, 13, 13, 15, 15, 16, 19 20, 20, 20, 21, 21, 22, 23, 24, 30, 40, 45, 45, 45, 71, 72, 73, 75

(a)Find mean, Median and mode.

(b)Show a box plot of the data. Clearly indicating the five-number summary.

MU- Dec. 2013, 10 Marks

Soln. :

(a) Mean

Mean is given by $\bar{x}$ (pronounced as x bar)

$$\bar{x} = \frac{(x_1 + x_2 + \ldots + x_n)}{n}$$

Or

$$\mu = \frac{\Sigma x}{n}$$

Where, μ represents the mean.

$\bar{x}$ OR μ = (11+ 13+ 13+ 15+ 15+ 16+ 19+ 20+ 20+20+21+ 21+ 22+ 23+ 24+ 30+ 40+ 45+ 45+ 45+ 71+ 72+ 73+ 75)/24 = 769/24 = 32.041

Median

We have even number of elements so in this case an average value of two values i.e. 21 and 22 is considered as median of the above data set. i.e. 21.5

Mode

- The mode is the most frequently occurring value in the data set.
- Therefore in this case the mode value for data set is 45 and 20 as it is bimodal.

(b) Plot box of the data

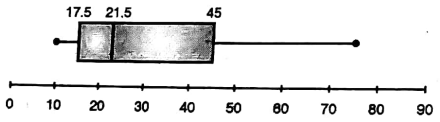


Fig. Ex. 2.2.1

Five-number summary for given data set :

The five number summary of a distribution consists of the minimum value, first quartile, median value, third quartile, and maximum value. It provides a good summary of the shape of the distribution and for this data is: 11, 17.5, 21.5, 45, and 75.

Ex. 2.2.2 Consider the following data points : 13, 15, 16, 19, 20, 20, 21, 22, 22, 25, 25, 25, 25, 30, 33, 33, 35, 35, 35, 35, 36, 40, 45, 46, 52, 70.

(a) What is the mean of the data? What is the median?

(b) What is the mode of the data?

(c) What is the midrange of the data?

(d) Can you find (roughly) the first quartile (Q1) and the third quartile (Q3) of the data ?

(e) Show a boxplot of the data.

MU- May 2015, 10 Marks

Soln :

(a) The (arithmetic) mean of the data is: $\bar{x} = \frac{1}{N} \sum_{i=1}^{N} x_i = 809/27 = 30$

The median (middle value of the ordered set, as the number of values in the set is odd) of the data is: 25.

(b) This data set has two values that occur with the same highest frequency and is, therefore, bimodal. The modes (values occurring with the greatest frequency) of the data are 25 and 35.

(c) The midrange (average of the largest and smallest values in the data set) of the data is: (70 + 13)/2 = 41.5.

(d) The first quartile (corresponding to the 25th percentile) of the data is: 20. The third quartile (corresponding to the 75th percentile) of the data is: 35.

(e) Boxplot of the data

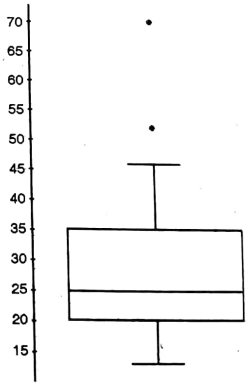


Fig. Ex. 2.2.2

Syllabus Topic : Data Visualisation

2.3 Data Visualisation

MU - May 2015, Dec. 2015

Data visualization is presenting the data in a graphical or pictorial format. Visualization techniques help people to analyze things which are otherwise not possible when the data is large. Patterns in the data can be marked very easily using the data visualization techniques. Some of the data visualization techniques are as follows :

1. Pixel-oriented visualization techniques

- In pixel based visualisation techniques, there is a separate sub-windows for the value of each attribute and is represented by one colored pixel.
- It maximises the amount of information represented at one time without any overlap.
- A tuple with m variables has different m colored pixel to represent each variable and each variable has sub window.
- Based on data characteristics and visualization task, the color mapping of the pixels is decided.

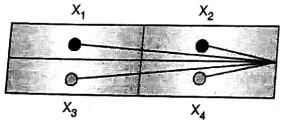


Fig. 2.3.1 : Pixel visualization with four variables

2. Geometric projection visualization techniques

Geometric transformations and projections of multidimensional data sets can be found using the following techniques :

(i) Scatter plot matrices
(ii) Hyper slice
(iii) Parallel coordinates

(i) **Scatter plot matrices:** It is composed of scatter plots of all possible pairs of variables in a dataset.

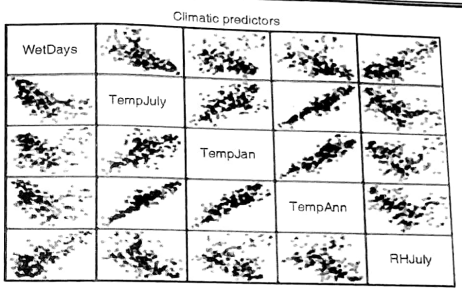


Fig. 2.3.2 : An example of scatter plot

(ii) **Hyper slice :** It is an extension to scatter plot matrices. They represent a multidimensional function as a matrix of orthogonal two dimensional slices.

(iii) **Parallel co-ordinates :** The parallel vertical lines separated define the axes. A point in the Cartesian coordinates corresponds to a poly line in parallel coordinates.

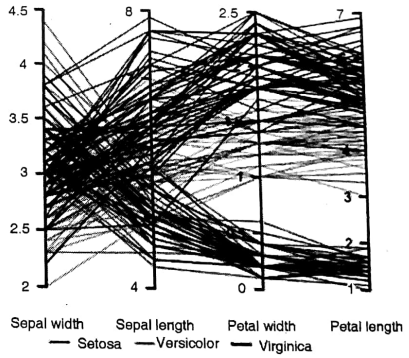


Fig. 2.3.3: An example of Parallel coordinates

3. Icon-based visualization techniques

- Icon based visualisation techniques also known as iconic display techniques.
- Each multidimensional data item is mapped to an icon.
- This technique allows visualisation of large amounts of data.

- Two most commonly used icon based techniques are :

(I)	Chernoff faces
(II)	Stick figures

(i) Chernoff faces

- Illustration of trends in multidimensional data can be done by using Chernoff faces. This concept was introduced by Herman Chernoff in the year 1973.
- The faces in Chernoff faces are related to facial expressions or features of human being. So to distinguish between them is easy.
- Different data dimensions were mapped to different facial features, for example the face width, the length or curvature of the mouth, the length of the nose etc.
- An example of Chernoff faces is shown below; they use facial features to represent trends in the values of the data, not the specific values themselves.
- They display multidimensional data of upto 18 variables or dimensions.
- In Fig. 2.3.4. Each face represents an n-dimensional data points(n<=18).

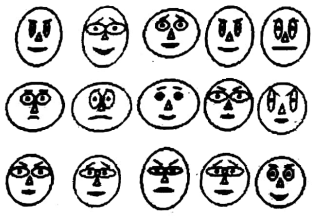
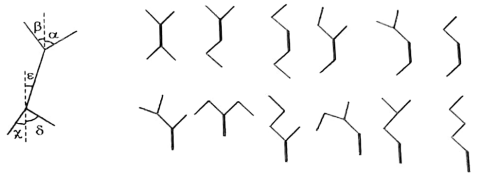


Fig. 2.3.4:An example of Chernoff faces

(ii) Stick figures

- Pickett and Grinstein introduced stick figure icon visualization technique.
- The Fig. 2.3.5 represents the original stick figure with five stick and family of twelve of them.
- This icon family is designed to display data with up to five variates.
- Stick icon can be used to display bivariate MRI data by using a two stick icon which helps to differentiate the texture in a complex image.



(a) A five stick figure icon with orientation **(b) A stick figure icon family with a body and four limb**

Fig. 2.3.5

4. Hierarchical visualization techniques

- The visualisation techniques discussed above display multiple dimensions simultaneously. However for a large data set having large number of dimensions the above techniques may not be useful.
- Hierarchical visualisation techniques partition all dimensions in to subset (subspaces).
- These subspaces are visualised in a hierarchical manner.
- Some of the visualisation techniques are :

(i)	Dimensional stacking
(ii)	Mosaic Plot
(iii)	Worlds-within-worlds
(iv)	Tree-map
(v)	Visualizing complex data and relations

(i) Dimensional stacking

- In dimension stacking, partition the n-dimensional attribute space in 2-dimenstinal subspaces.
- Attribute values are partitioned into various classes.
- Each element is a two dimensional space is a xy plot.
- Mark the important attributes and are used on the outer levels.

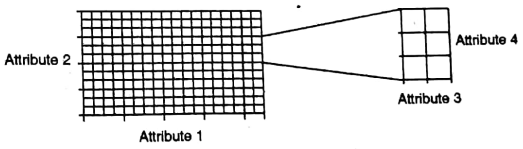


Fig. 2.3.6 : Data in dimension stacking

(ii) Mosaic plot

- Mosaic plots give a graphical illustration of the successive decompositions.
- Rectangles are used to represent the count of categorical data and at every stage rectangles are split parallel.
- To draw a mosaic plot, a contingency table of data and chosen ordering of variable with the response variable is required.
- **Example :** In titanic example , out of all women , 67% survived which is coded as 1 and 33% died which is coded as 0. So the women bar shows as 67/33 split. Among men, only 17% survived, so this bar shows a 17/83 split.

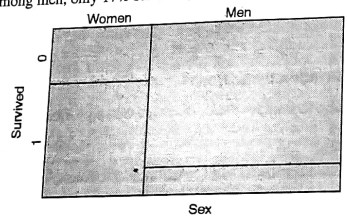


Fig.2.3.7 : Mosaic Plot for Titanic

(iii) Worlds within worlds

- Worlds within worlds are useful to generate an interactive hierarchy of display.
- Innermost word must have a function and two most important parameters
- Remaining parameters fix with constant value.
- Through this N-vision of data are possible like data glove and stereo displays, including rotation, scaling (inner) and translation (inner/outer)
- Using queries static interaction is also possible.

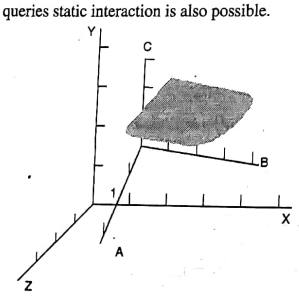


Fig. 2.3.8 : Worlds within worlds visualization

(iv) Tree-maps

- **Tree maps** visualization techniques are well suited for displaying large amounts of hierarchical structured data.
- The visualization space is divided into multiple rectangles that are sized and ordered according to a quantitative variable.
- The levels in the hierarchy are seen rectangles containing other rectangles.
- Each set of rectangles on the same level in the hierarchy represents a column or an expression in a data set.
- Each individual rectangle on a level in the hierarchy represents a category in a column.
- **Example :**In the Fig. 2.3.9, a rectangle representing global below which there are rectangles representing continents which contain several rectangles representing countries in that continent.
- Each rectangle representing a country may in turn contain rectangles representing states in these countries.

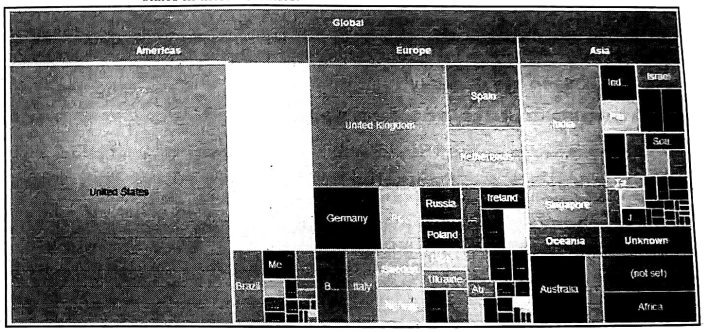


Fig. 2.3.9 : Web traffic by location Tree-map

(v) Visualizing complex data and relations

- This technique is useful to visualize non-numeric data such as text, pictures, blog entries and product reviews.
- A tag cloud is a visualization method which helps to understand the information of user generated tags.
- Arrange the tags alphabetically or with the user preferences with different font sizes and colors.
- Tag clouds are used in two ways that with the size of tag, we can find out that how many times that tag is applied on that item by different users or that tag has been applied to how many items.

Fig. 2.3.10 : Social data visualization

Syllabus Topic : Measuring Similarity and Dissimilarity

2.4 Measuring Similarity and Dissimilarity

Data Mining Applications such as Clustering, Classification, outlier Analysis needs a way to assess of how alike or unalike are the objects from one another. For this some measures of similarity and dissimilarity are needed and given below.

2.4.1 Data Matrix versus Dissimilarity Matrix

Let us consider a set of n objects with p attributes given by $X_1=(X_{11},X_{12},\ldots. X_{1p})$, $X_2= (X_{21}, X_{22},\ldots. X_{2p})$ and so on. Where X_{ij} is the value for i^{th} object with j^{th} attribute. These objects can be tuples in a relational database or feature vectors.

There are mainly two types of data structures for main memory-based clustering algorithms :

1. **Data matrix or object by variable structure**

$$\begin{bmatrix} x_{11} & \ldots & x_{1l} & \ldots & x_{1l} \\ \ldots & \ldots & \ldots & \ldots & \ldots \\ x_{i1} & \ldots & x_{il} & \ldots & x_{il} \\ \ldots & \ldots & \ldots & \ldots & \ldots \\ x_{n1} & \ldots & x_{nl} & \ldots & x_{nl} \end{bmatrix}$$

The Data matrix stores the n data objects in the form of a relational table or in the form of a matrix as shown above.

2. **Dissimilarity matrix or by object structure**

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

- In the above dissimilarity matrix d(i,j) refers to the measure of dissimilarity between objects i and j.
- d(i,j) is close to 0 when the objects i and j are similar
- The distance d(i,j)==d(j,i), hence not shown as a part of the above matrix as the matrix is symmetric.
- **Similarity:** Similarity in data mining context refers to how much alike two data objects are which can be described by the distance with dimensions representing features of objects where a small distance indicating that the objects are highly similar and a large indicates they are not.
- Similarity can also be expressed as, **sim(i,j)= 1-d(i,j).**
- **Two mode matrix :**Data Matrix is also called as two mode matrix as it represents two entities objects which are its features.
- **One mode matrix :**Dissimilarity matrix is called as one mode matrix as it only represents one dimension i.e. the distance.

2.4.2 Proximity Measures for Nominal Attributes

- Proximity refers to either similarity or dissimilarity. As defined above in Section 2.2 what are nominal attributes, let us see how to calculate similarity and dissimilarity of nominal attributes.
- Dissimilarity is given by,

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

where, p= Total number of attributes describing the objects

and m= Number of matches

- Similarity is given by,

$$sim(i,j) = 1 - d(i,j)=\frac{m}{p}$$

Table 2.4.1

Id	Types of Property
1	Houses
2	Condos
3	co-ops
4	bungalows

- The Table 2.4.1 represents nominal data for an estate agent classifying different types of property. The dissimilarity matrix for the above example can be calculated as follows :

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

- The value in the above matrix is 0 if the objects are similar and it is a 1 if the objects differ.

2.4.3 Proximity Measures for Binary Attributes

- Binary attributes are of two types, symmetric and asymmetric.
- A contingency Table 2.4.2 for binary data :

Table 2.4.2

	Object n		
Object m	1	0	Sum
1	A	b	a + b
0	C	d	c + d
Sum	a + c	b + d	P

- Here we are comparing two objects, object *m* and object *n*.

a would be the number of variables which are present for both objects.

b would be the number found in object m but not in object *n*.

c is just the opposite to b and *d* is the number that are not found in either object.

- Simple matching coefficient (invariant, if the binary variable is *symmetric*) as shown in Equation (2.4.1) :

$$d(i, j) = \frac{b + c}{a + b + c + d} \quad \ldots(2.4.1)$$

- Jaccard coefficient (non-invariant if the binary variable is *asymmetric*) as shown in Equation (2.4.2) :

$$d(i, j) = \frac{b + c}{a + b + c} \quad \ldots(2.4.2)$$

Example :

Table 2.4.3 : A Relational table containing mostly binary values

Name	Gender	Fever	Cough	Test-1	Test-2	Test-3	Test-4
Jai	M	Y	N	P	N	N	N
Raj	F	Y	N	P	N	P	N
Jaya	M	Y	P	N	N	N	N

- Gender is a symmetric attribute the remaining attributes are asymmetric binary.
- Let the values Y and P be set to 1, and the value N be set to 0 as shown in the Table 2.4.4.

Using Equation(2.4.2) of asymmetric variable.

Table 2.4.4

Name	Gender	Fever	Cough	Test-1	Test-2	Test-3	Test-4
Jai	M	1	0	1	0	0	0
Raj	F	1	0	1	0	1	0
Jaya	M	1	1	0	0	0	0

- Distance between Jai and Raj (i.e. *d(Jai, Raj)*) is calculated using Equation (2.4.2) and use contingency Table 2.4.4.

Consider attributes : Fever, cough, Test-1, Test-2, Test-3, Test-4

Consider Jai as object i and Raj as object j

a = Attribute values 1 in Jai and in Raj also = 2

b = Attribute values 1 in Jai but 0 in Raj = 0

c = Attribute values 0 in Jai but 1 in Raj = 1

$$d(i, j) = \frac{b + c}{a + b + c}$$

$$d(\text{Jai, Raj}) = \frac{0 + 1}{2 + 0 + 1} = 0.33$$

Similarly, calculate distance for other combination

$$d(\text{Jai, Jaya}) = \frac{1 + 1}{1 + 1 + 1} = 0.67$$

$$d(\text{Jaya, Raj}) = \frac{1 + 2}{1 + 1 + 2} = 0.75$$

- So, Jai and Raj are most likely to have a similar disease with lowest dissimilarity value.

2.4.4 Dissimilarity of Numeric Data : Minkowski Distance

- It is used to determine the similarity or dissimilarity between two data objects.
- Minkowski distance formula :

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

where

$i = (x_{i1}, x_{i2}, \ldots, x_{ip})$ and $j = (x_{j1}, x_{j2}, \ldots, x_{jp})$ are two objects with p number of attributes,

q is a positive integer

- If $q = 1$, then $d(i,j)$ is Manhattan distance

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

- If q = 2,then d(i,j) is Euclidean distance :

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

- Both the Euclidean distance and Manhattan distance satisfy the following mathematic requirements of a distance function :

$$d(i,j) \geq 0$$

$$d(i,i) = 0$$

- Supremum/Chebyshev (if $q = \infty$)

$$d(i,j) = \max_t |i_t - j_t|$$

- Let us consider the following data :

Customer ID	No. of Trans	Revenue	Tenure(Months)
101	30	1000	20
102	40	400	30
103	35	300	30
104	20	1000	35
105	50	500	1
106	80	100	10
107	10	1000	2

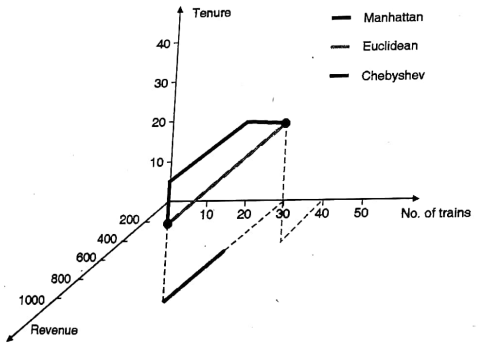


Fig. 2.4.1

$$d_1(\text{cust101, cust102}) = |30-40| + |1000-400| + |20-30| = 620$$

$$d_2(\text{cust101, cust102}) = \sqrt{(30-40)^2 + (1000-400)^2 + (20-30)^2} \approx 600.16$$

$$d_{max}(\text{cust101, cust102}) = |1000-400| = 600$$

2.4.5 Proximity Measures for Ordinal Attributes

- An ordinal attribute can be discrete or continuous. The ordering of it is important e.g. a rank. These attributes can be treated like interval scaled variables.
- Let us consider f as an ordinal attribute having M_f states. These ordered states define the ranking :

$$r_{if} \in \{1, \ldots M_f\}$$

- Map the range of each variable onto [0, 1] by replacing i^{th} object in the f^{th} variable by,

$$z_{if} = \frac{r_{if} - 1}{M_f - 1}$$

- Compute the dissimilarity using distance methods discussed in Section 2.4.4.
- Let us consider an example :

Emp Id	Income
1	High
2	Low
3	Medium
4	High

- The three states for the above income variable are low, medium and high, that is M_f=3.
- Next we can replace these values by ranks 3(low), 2(medium) and 1(High).
- We can now normalize the ranking by mapping rank 1 to 0.0, rank 2 to 0.5 and rank 3 to 1.0.
- Next to calculate the distance we can use the Euclidean distance that results in a dissimilarity matrix as :

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

- From the above matrix it can be seen that objects 1 and 2 are most dissimilar so are the object 2 and 4.

2.4.6 Dissimilarity for Attributes of Mixed Types

- In many of the applications, objects may be described by a mixture of attribute types.
- In such cases one of the most preferred approach is to combine all the attributes into a single dissimilarity matrix and computing on a common scale of [0.0 , 1.0]
- The dissimilarity may be calculated using

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

$$\delta_{ij}(f) = 0$$

Where if either

X_{if} or X_{jf} is missing

$X_{if} = X_{jf} = 0$ and attribute f is asymmetric binary

Otherwise

$$\delta_{ij}(f) = 1$$

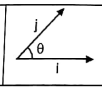
- The f attribute is computed based on the following :
  - Iff is binary or nominal:

    $d_{ij}^{(f)} = 0$ if $x_{if} = x_{jf}$, or $d_{ij}^{(f)} = 1$ other wise
  - If f is interval-based then use the normalized distance.
  - If f is ordinal or ratio-scaled then compute ranks r_{if} and treat z_{if} as interval-scaled.

$$z_{if} = \frac{r_{if} - 1}{M_f - 1}$$

2.4.7 Cosine Similarity

- Cosine similarity is a measure of similarity between two vectors. The data objects are treated as vectors. Similarity is measured as the angle θ between the two vectors. Similarity is 1 when θ = 0, and 0 when θ = 90°.
- Similarity function is given by,

$\cos(i,j) = \frac{i \bullet j}{\|i\| \times \|j\|}$	$i \bullet j = \sum_{k-1}^{n} i_k j_k$	$\|i\| = \sqrt{\sum_{k-1}^{n} i_k^2}$	

- Let us consider an example

  Given two data objects: x = (3, 2, 0, 5), and y = (1, 0, 0, 0)

  Since,

$$x \bullet y = 3*1 + 2*0 + 0*0 + 5*0 = 3$$

$$\|x\| = \text{sqrt}(3^2 + 2^2 + 0^2 + 5^2) \approx 6.16$$

$$\|y\| = \text{sqrt}(1^2 + 0^2 + 0^2 + 0^2) = 1$$

Then, the similarity between x and y: $\cos(x, y) = 3/(6.16 * 1) = 0.49$

The dissimilarity between x and y: $1 - \cos(x,y) = 0.51$

2.5 Data Processing

- 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.

2.6 Form of Data Pre-processing

MU - Dec. 2016

Syllabus Topic : Why Pre-processing is Required ?

2.6.1 Why Pre-processing is Required ?

MU - May 2015, Dec. 2015, May 2016

1. Real world data are generally

- **Incomplete :** The data is said to be incomplete when certain attributes or attributes values are missing or only aggregate data is available.
- **Noisy :**When the data contains errors or some outliers it is considered to be noisy data.
- **Inconsistent :** When the data contains differences in codes or names it is inconsistent data.

2. 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.

2.6.2 Different Forms of Data Pre-processing

MU - May 2015, Dec. 2015, May 2016

1. Data cleaning
2. Data integration and transformation
3. Data reduction
4. Data discretization and Concept hierarchy generation

Syllabus Topic : Data Cleaning

2.7 Data Cleaning

Data cleaning is also known as scrubbing. The data cleaning process detects and removes the errors and inconsistencies and improves the quality of the data. Data quality problems arise due to misspellings during data entry, missing values or any other invalid data.

2.7.1 Reasons for "Dirty" Data

- Dummy values
- Absence of data
- Multipurpose fields
- Cryptic data
- Contradicting data
- Inappropriate use of address lines
- Violation of business rules
- Reused primary keys
- Non-unique identifiers
- Data integration problems

Why data cleaning or cleansing is required ?

- Source Systems data is not clean; it contains certain errors and inconsistencies.
- Specialized tools are available which can be used for cleaning the data.
- Some of the Leading data cleansing vendors include Vality (Integrity), Harte-Hanks (Trillium) and First logic.

2.7.2 Steps in Data Cleansing

MU - May 2016

1. Parsing

- Parsing is a process in which individual data elements are located and identified in the source systems and then these elements are isolated in the target files.
- For example, parsing of name into First name, Middle name and Last name or parsing the address into street name, city, state and country.

2. Correcting

- This is the next phase after parsing, in which individual data elements are corrected using data algorithm and secondary data sources.
- For example, in the address attribute replacing a vanity address and adding a zip code.

3. Standardizing

- In standardizing process conversion routines are used to transform data into a consistent format using both standard and custom business rules.
- For example, addition of a pre-name, replacing a nickname and using a preferred street name.

4. Matching

- Matching process involves eliminating duplications by searching and matching records with parsed, corrected and standardised data using some standard business rules.
- For example, identification of similar names and addresses.

5. Consolidating

Consolidation involves merging the records into one representation by analyzing and identifying relationship between matched records.

6. Data cleansing must deal with many types of possible errors

- Data can have many errors like missing data, or incorrect data at one source.
- When more than one source is involved there is a possibility of inconsistency and conflicting data.

7. Data staging

- Data staging is an interim step between data extraction and remaining steps.
- Using different processes like native interfaces, flat files, FTP sessions, data is accumulated from asynchronous sources.
- After a certain predefined interval data is loaded into the warehouse after the transformation process.
- No end user access is available to the staging file.
- For data staging, operational data store may be used.

2.7.3 Missing Values

Missing data values

- This involves searching for empty fields where values 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 process by filling out the missing values, smoothening 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 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 difficult 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.

2.7.4 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 at 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

How to handle noisy data ?

Different data smoothing techniques are given below :

1. **Binning**
   - Considering the neighbourhood of the sorted data smoothening can be applied.
   - The sorted data is placed into bins or buckets.
   - Smoothing by bin means.
   - Smoothing by bin medians.
   - Smoothing by bin boundaries.

Different approaches of binning

(a) **Equal-width (distance) partitioning**
   - Divides the range into *N* intervals of equal size: uniform grid.

     bin width = (max value – min value) / N
   - **Example :** Consider a set of observed values in the range from 0 to 100.

     The data could be placed into 5 bins as follows :

     width = (100 – 0)/5 = 20

     Bins formed are : [0-20], (20-40], (40-60], (60-80], (80-100]
   - The first and the last bin is extended to allow values outside the range : (-infinity-20], (20-40], (40-60], (60-80], (80-infinity)

Disadvantages
   - Outliers in the data may be a problem.
   - Skewed data cannot be held with this method.

(b) Equal-depth (frequency) partitioning or Equal-height binning

- The entire range is divided into N intervals, each containing approximately the same number of samples.
- This results in good data scaling.
- Handling categorical attributes may be a problem.

Example1

- Let us consider sorted data for e.g. Price in INR
  4, 8, 9, 15, 21, 21, 24, 25, 26, 28, 29, 34
- Partition into (equal-depth) bins: (N=3)
  Bin 1: 4, 8, 9, 15
  Bin 2: 21, 21, 24, 25
  Bin 3: 26, 28, 29, 34
- Smoothing by bin means :
  Replace each value of bin with its mean value.
  Bin 1: 9, 9, 9, 9
  Bin 2: 23, 23, 23, 23
  Bin 3: 29, 29, 29, 29
- Smoothing by bin boundaries :
  In this method the minimum and maximum values of the bean boundaries is found and each value is replaced with its nearest value either minimum or maximum.
  Bin 1: 4, 4, 4, 15
  Bin 2: 21, 21, 25, 25
  Bin 3: 26, 26, 26, 34

Ex. 2.7.1: Partition the given data into 4 bins using Equi-depth binning method and perform smoothing according to the following methods. Smoothing by bin mean Smoothing by bin median Smoothing by bin boundaries.
Data : 11,13,13,15,15,16,19,20,20,20,21,21,22,23,24,30,40,45,45,45,71,72,73,75
MU - May 2016, Dec. 2016, 10 Marks

Soln. :

- Partition into (equal-depth) bins : (N=4)
  Bin 1: 11,13,13,15,15,16
  Bin 2: 19,20,20,20,21,21
  Bin 3: 22,23,24,30,40,45
  Bin 4: 45,45,71,72,73,75
- Smoothing by bin means :
  Replace each value of bin with its mean value.

Bin 1:13.83,13.83,13.83,13.83,13.83,13.83
Bin 2: 20.16,20.16,20.16,20.16,20.16,20.16
Bin 3: 30.67,30.67,30.67,30.67,30.67,30.67
Bin 4: 63.5,63.5,63.5,63.5,63.5,63.5

- Smoothing by bin boundaries :
  In this method the minimum and maximum values of the bean boundaries is found and each value is replaced with its nearest value either minimum or maximum.
  Bin 1: 11,11,11,16,16,16
  Bin 2: 19, 19, 19, 21, 21, 21
  Bin 3: 22, 22, 22, 22, 45, 45
  Bin 4: 45,45,75,75,75,75
- Smoothing by bin median :
  Replace each value with median. As number of elements are even in each bin , so find the median value by considering the average of middle two values of each bin.
  Bin 1: 14, 14, 14, 14, 14, 14
  Bin 2: 20, 20 ,20, 20, 20, 20
  Bin 3: 27, 27, 27, 27, 27, 27
  Bin 4: 71.5, 71.5, 71.5, 71.5, 71.5, 71.5

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.

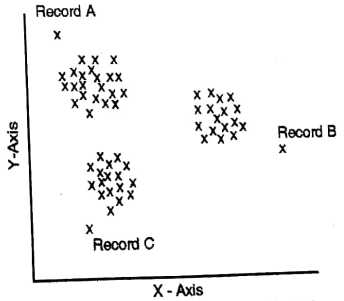


Fig. 2.7.1:Graphical Example of Clustering

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 variables to predict the outcome.
- The general form of each type of regression is :

Linear Regression : $Y = a + bX + u$

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

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.
- 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.

Log linear 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 variables.
- Log linear models are models that postulate a linear relationship between the independent variables and the logarithm of the dependent variable.

  For example : $\log(y) = a_0 + a_1 x_1 + a_2 x_2 ... + a_N x_N$

  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 (additive) one.

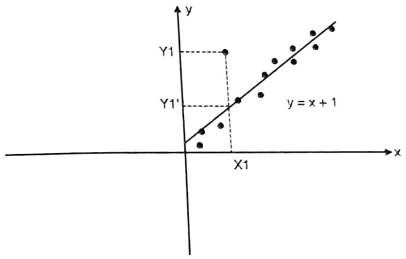


Fig. 2.7.2 : Regression example

2.7.5 Inconsistent Data

- The state in which the data quality of the existing data is understood and the desired quality of the data is known refers to consistent data quality.
- It is a state in which the existent data quality is been modified to meet the current and future business demands.

Syllabus Topic : Data Integration

2.8 Data Integration

2.8.1 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.

Issues in data integration

- **Schema integration**
  - Integrate metadata from different sources.
  - Entity identification problem: identify real world entities from multiple data sources, e.g. A.cust-id ≡B.cust-#.
- **Detecting and resolving data value conflicts**
  - As the data is collected from multiple sources, attribute values are different for the same real world entity.
  - Possible reasons include different representations, different scales, e.g. metric vs. British units.

- **Redundant data occur due to integration of multiple databases**
  - Attributes may be represented in different names in different sources of data.
  - An attribute may be derived attribute in another table, e.g. yearly income.
  - With the help of co-relational analysis, detection of redundant data is possible.
  - The redundancies or inconsistencies may be reduced by careful integration of the data from multiple sources, which will help in improving mining speed and quality.

2.8.1(A) Entity Identification Problem

- Schema integration is an issue as to integrate metadata from different sources is a difficult task.
- Identify real world entities from multiple data sources and their matching is 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.
- Detecting and resolving data value conflicts for the same real world entity, attribute values from different sources are different.

2.8.1(B) 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.
- X^2 *(Chi-square)* test can be carried out on nominal data to test how strongly the two attributes are related.
- Correlation coefficient and covariance may be used with numeric data, this will give a variation between the attributes.

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

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

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

$$DF = (r-1) * (c-1)$$

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 attribute 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.

Ex. 2.8.1 : By taking the random sample of 1000 buyers, a survey was conducted. Sample data were classified by gender (male or female) and by buying preference (Young Age, Middle Age, or Old Age). Results are shown in the contingency table below.

	Buying Preferences			Row Total
	Young Age	Middle Age	Old Age	
Male	200	150	50	400
Female	250	300	50	600
Column total	450	450	100	1000

Level of significance is 0.05. Interpret the result.

Soln. :

- **State the hypotheses**
  1. H_0: Null hypothesis : Gender and buying preferences are independent.
  2. H_a: Alternative hypothesis : Gender and buying preferences are not independent.

- **Analyze sample data**

  The degrees of freedom DF is,

  $$DF = (r-1) * (c-1) = (2-1) * (3-1) = 2$$

  r = 2 (the number of levels for row i.e. male and female)

  c = 3 (the number of levels for column categorical variable i.e. Young age, Middle age and Old age.)

- By using given contingency table, calculate the expected frequency $E_{r,c}$ for each level $E_{r,c} = (n_r * n_c) / n$.

	Buying preferences			Row total
	Young age	Middle age	Old age	
Male (O)	200	150	50	400
Male(E)	(400 * 450) / 1000 =180	(400 * 450) / 100 =180	(400 * 100) / 1000 =40	400
Male(O-E)	20	– 30	10	
Male(O-E)^2	400	900	100	
Male(O-E)^2/E	2.22	5	2.5	
Female(O)	250	300	50	600
Female(E)	(600 * 450) / 1000 =270	(600 * 450) / 1000 =270	(600 * 100) / 1000 =60	600
Female(O-E)	– 20	30	– 10	
Female(O-E)^2	400	900	100	
Female (O-E)^2/E	1.48	3.33	1.67	

- Using formula,

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

$$X^2 = 2.22 + 5.00 + 2.50 + 1.48 + 3.33 + 1.67 = 16.2$$

- The P-value is the probability that a chi-square statistic having 2 degrees of freedom is more extreme than 16.2.
- We use the Chi-Square distribution table to find $P(X^2 > 16.2) = 0.0003$.
- **Results Interpretation**

The significance level given in problem statement is 0.05 and the P-value (0.0003) is less than the significance level (0.05), So, the null hypothesis cannot be accepted and we can interpret that there is a relationship between gender and buying preference.

The correlation coefficient and covariance

- The correlation coefficient is given by,

$$r_{p,q} = \frac{\sum (p-\bar{p})(q-\bar{q})}{n \cdot \sigma_p \sigma_q} = \frac{\sum ((pq) - n\,\bar{p}\,\bar{q})}{n \cdot \sigma_p \sigma_q}$$

  Where n is the number of tuples, $\bar{p}$ and $\bar{q}$ are the respective means of p and q, σ_p and σ_p are the respective standard deviation of p and q and $\sum$ (pq) is the sum of the pq cross-product.

- If $r_{p,q} > 0$, p and q are positively correlated (p's values increase as q's). The higher, the stronger correlation. $r_{p,q} = 0$: independent ; $r_{pq} < 0$: negatively correlated.
- Correlation measures the linear relationship between objects. First standardize data objects, x and y and then calculate the correlation between them by taking dot product.

$$X'_k = (X_k - \text{mean}(X))/\text{std}(X)$$

$$Y'_k = (Y_k - \text{mean}(Y))/\text{std}(Y)$$

$$\text{Correlation}(X,Y) = X' \cdot Y'$$

- Covariance is given by

$$\text{Cov}(X,Y) = E((X-\bar{X})(Y-\bar{Y})) = \frac{\sum_{i=1}^{n} (x_i-\bar{X})(y_i-\bar{Y})}{n}$$

  It can be simplified in computation as

$$\text{Cov}(X,Y) = E(X \cdot Y) - \bar{X}\,\bar{Y}$$

2.8.1(C) Tuple Duplication

- In data integration tuple duplication should be checked. This type of duplication may occur due to incorrect data entry or updation of data.
- Detection of tuple duplication :
  - Specify the relevant attributes.
  - Compare tuples pair wisely using a similarity measure.
  - Objects with similarity above a given threshold are considered as duplicates.
  - The closures of duplicates are computed, in which each is assigned one <sourceID>.

2.8.1(D) Data Value Conflict Detection and Resolution

- Data conflicts can arise because of incomplete data, invalid data and out-of-date data.
- It is thus critical for data integration systems to resolve conflicts from various sources and identify true values from false ones.

- **Two kinds of data conflicts:**
  1. Uncertainty 2. Contradiction

Uncertainty	Contradiction
It is an attribute level data conflict.	It is an attribute level data conflict.
If there is missing information or null values, then Uncertainties occurs.	If there is contradicting information or different attribute values, then contradiction occurs.
If information for the attribute value is not available, then uncertainty is introduced.	When data is collected from various sources, then contradiction is introduced for the same properties of the same objects.
If the set of values includes special NULL value and one other NON NULL value, then uncertainty present.	If at least two different non-null values appear in the set of values, then the contradiction is present.
For age attribute, if the set of values for customer1 are{30,Null,30}={30,Null}. Then there is uncertainty as there is on Null value for age.	For age attribute, if the set of values for customer1 are{30,35,30}={30,35}. Then contradictions, as there are two different values for age.

Syllabus Topic : Data Reduction

2.9 Data Reduction

2.9.1 Need for Data Reduction

1. Reducing the number of attributes

- **Data cube aggregation :** This process involves applying OLAP operations 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

1. Data cube aggregation
2. Dimensionality reduction
3. Data compression
4. Numerosity reduction

2.9.2 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.2.9.1.

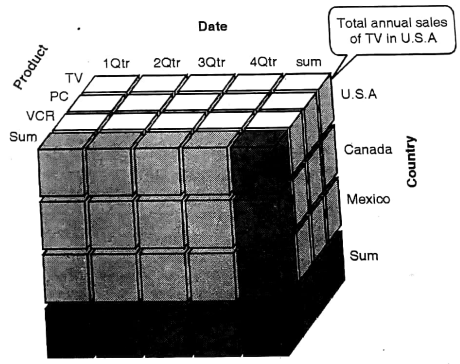


Fig. 2.9.1 : Example of data cube

2.9.3 Dimensionality Reduction

- In the mining task during analysis, the data sets of information may contain large number of attributes that may be irrelevant or redundant.
- Dimensionality reduction is a process in which attributes are removed and the resulting dataset is smaller in size.
- This process helps in reducing the time and space complexity required by a data mining technique.
- Data visualization becomes an easy task.

- It also involves deleting inappropriate features or reducing the noisy data.

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 procedure 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.

2.9.4 Data Compression

- Data compression is the process of reducing the number of bits needed to either store or transmit the data. This data can be text, graphics, video, audio, etc. This can be usually be done with the help of encoding techniques.
- Data compression techniques can be classified into either lossy or lossless techniques. In lossy technique there is a loss of information whereas in lossless there is no loss.

Lossless compression

- Lossless compression consists of those techniques guaranteed to generate an exact duplication of the input dataset after a compress/decompress cycle.
- Lossless compression is essentially a coding technique. There are many different kinds of coding algorithms, such as Huffman coding, run-length coding and arithmetic coding.

Loss compression

- In lossy compression techniques at the cost of data quality one can achieve higher compression ratio.
- These types of techniques are useful in applications where data loss is affordable. They are mostly applied to digitized representations of analog phenomenon.
- Two methods of lossy data compression :

1. Wavelet transforms
2. Principle component analysis

1. The wavelet transform

A clustering approach which applies wavelet transform to the feature space :

- The orthogonal wavelet transform when applied over a signal results in time scale decomposition through its multi-resolution aspect.
- It clusters the functional data into homogenous groups.
- Both grid-based and density-based.

Input parameters

- Number of grid cells for each dimension.
- The wavelet and the number of applications of wavelet transform.
- Clustering approach using Wavelet transform.
- Impose a multidimensional grid like structure on to the data for summarisation.
- Use an n-dimensional feature space for representing spatial data objects.
- Dense regions may be identified by applying the wavelet transform over the feature space.
- Applying wavelet transform multiple times results in clusters of different scales
- Clusters are identified by using hat-shape filters and also suppress weaker information in their boundary.

Major features

- It also results in Effective removal of outliers.
- The technique is Cost efficient.
- Complexity O(N).
- At difference scales arbitrary shaped clusters are detected.

- The method is not sensitive to noise or input order.
- It is applicable only to low dimensional data.

2. Principal components analysis

- Principal Component Analysis (PCA) creates a representation of the data with orthogonal basis vectors, i.e. eigenvectors of the covariance matrix of the data. This can also be derived using Singular value decomposition method. By this projection original dataset is reduced with little loss of information.
- PCA is often presented using the eigen value/eigenvector approach of the covariance matrices. But in efficient computation related to PCA, it is the Singular Value Decomposition (SVD) of the data matrix that is used.
- A few scores of the PCA and the corresponding loading vectors can be used to estimate the contents of a large data matrix.
- The idea behind this is that by reducing the number of eigenvectors used to reconstruct the original data matrix, the amount of required storage space is reduced.

2.9.5 Numerosity Reduction

Numerosity reduction technique refers to reducing the volume of data by choosing smaller forms for data representation

Different techniques used for numerosity reduction are :

1. Histograms

- It replaces data with an alternative, smaller data representation.
- Approximate data distributions.
- Divide data into buckets and store average (sum) for each bucket.
- A bucket represents an attribute-value/frequency pair.
- Can be constructed optimally in one dimension using dynamic programming.
- Related to quantization problems.

Different types of histogram

1. **Equal-width histograms:** It divides the range into N intervals of equal size.
2. **Equal-depth (frequency) partitioning :** It divides the range into *N* intervals, each containing approximately same number of samples.
3. **V-optimal:** Different Histogram types for a given number of buckets are considered and the one with least variance is chosen.
4. **MaxDiff:** After the sorting process applied to the data, borders of the buckets are defined where the adjacent values have maximum difference.

Example :

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

Histogram of above data sample is,

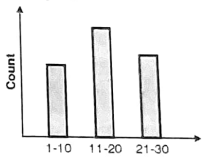


Fig. 2.9.2 : Example of histogram

2. Clustering

- Clustering is a data mining technique used to group the elements based on their similarity without prior knowledge of their class labels.
- It is a technique that belongs to undirected data mining tools.
- The goal of undirected data mining is to explore structure in the data. No target variable is to be predicted, therefore there is no difference been made between independent and dependent variables.
- Categorization of clusters based on clustering techniques is given below :
  - Any example belonging to a single cluster would be termed as exclusive cluster.
  - Any example may belong to many clusters in such a case it is said to be overlapping.
  - Any example belongs to a cluster with certain probability then it is said to be probabilistic.
  - A Hierarchical representation may be used for clusters in which clusters may be at highest level of hierarchy and subsequently refined at lower levels to form sub-clusters.

3. 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.

2.9.6 Data Transformation and Data Discretization

2.9.6(A) Data Transformation

- Operational databases keep changing with the requirements, a data warehouse integrating data from these multiple sources typically faces the problem of inconsistency.
- To deal with these inconsistent data, transformation process may be employed.
- The most commonly used process is "Attribute Naming Inconsistency", as it is very common to use different names to the same attribute in different sources of data.
- E.g. Manager Name may be MGM_NAME in one database, MNAME in the other.
- In this one set of data names is considered and used consistently in the data warehouse.
- Once the naming consistency is done, they must be converted to a common format.
- The conversion process involves the following :
  (i) ASCII to EBCDIC or vice versa conversion process may be used for characters.
  (ii) To ensure consistency uppercase representation may be used for mixed case text.
  (iii) A common format may be adopted for numerical data.
  (iv) Standardization must be applied for data format.
  (v) A common representation may be used for measurement e.g. (Rs/$).
  (vi) A common format must be used for coded data (e.g. Male/Female, M/F).
- The above conversions are automated and many tools are available for the transformation e.g. DataMapper.

Data transformation can have the following activities

- **Smoothing :** It involves removal of noise from the data.
- **Aggregation :** It involves summarization and data cube construction.
- **Generalization :** In generalization data is replaced by higher level concepts using concept hierarchy.
- **Normalization :** In normalization, attribute scaling is performed for a specified range.

Example : To transform V in [min, max] to V′ in [0,1], apply

$$V' = (V\text{-Min}) / (\text{Max-Min})$$

Scaling by using mean and standard deviation (useful when min and max are unknown or when there are outliers) :

$$V' = (V\text{-Mean}) / \text{Std. Dev.}$$

- **Attribute/feature construction :** In this process new attributes may be constructed and used for data mining process

2.9.6(B) Data Discretization

- 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.

2.9.6(C) Data Transformation by Normalization

- Data Transformation by Normalization or standardization is the process of making an entire set of values have 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 results in a linear alteration of the original data. The values are within a given range.

   Following formula may be used to perform mapping a v value, of an attribute A from range [minA,maxA] to a new range [new_minA, new_maxA],

   $$v' = (v - \text{minA})/(\text{maxA} - \text{minA}) * (\text{new\_maxA} - \text{new\_minA}) + \text{new\_minA}$$

   $$v = 73600 \text{ in } [12000,98000]$$

   $$v' = 0.716 \text{ in } [0,1] \text{ (new range)}$$

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.

   $$v' = (v - \text{meanA}) / \text{std\_devA}$$

   Where, MeanA = sum of the all attribute value of A
   std_devA= Standard deviation of all values of A

   Example :

   If sample data {10, 20, 30}, then
   Mean = 20
   std_dev= 10
   So v'= (– 1, 0, 1)

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

$$v'(i) = v(i)/10^k \text{ for the smallest } k \text{ such that } \max(|v'(i)|)<1.$$

Example : For the range between – 991 and 99,

10^k is 1000 (k=3 as we have maximum 3 digit number in the range)

$$v'(-991) = -0.991 \text{ and } v'(99) = 0.099$$

2.9.6(D) Discretization by Binning

- This is the data smoothing technique.
- Discretization by binning has two approaches :
  a) Equal-width (distance) partitioning
  b) Equal-depth (frequency) partitioning or Equal-height binning
- Both this binning approaches are given in section 2.7.4.

2.9.6(E) Discretization by Histogram Analysis

Discretization by Histogram divides data into buckets and store average (sum) for each bucket in smaller data representation.

Different types of histogram

1. Equal-width histograms
2. Equal-depth (frequency) partitioning
3. V-optimal
4. MaxDiff

All the above mentioned methods are given in section 2.9.5 in Numerosity Reduction.

2.9.6(F) Concept Hierarchies

The amount of data may be reduced using concept hierarchies. The low level detailed data (for example numerical values for age) may be represented by higher-level data(e.g. Young, Middle aged or Senior).

Concept hierarchy generation for categorical data

- The users or experts may perform a partial/total ordering of attributes explicitly at schema level :
  E.g. street < city < state < country
- Specification of a hierarchy for a set of values by explicit data grouping :
  E.g. {Acton, Canberra, ACT} < Australia
- Ordering of only a partial set of attributes :
  E.g. only street < city, not others
- By analyzing number of distinct values the hierarchies or attribute levels may be generated automatically.

E.g. for a set of attributes : {street, city, state, country}

E.g. weekday, month, quarter, year

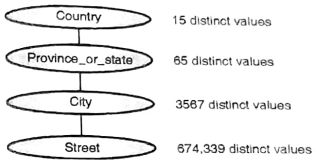


Fig. 2.9.3 : Concept hierarchy example

Review Questions

Q. 1 Explain the different types of attributes.

Q. 2 What are the different statistical parameters used for measuring the properties of data ?

Q. 3 Explain the various graphic displays used for statistical description of data.

Q. 4 Explain the different visualization techniques.

Q. 5 Explain the different measures of similarity and dissimilarity.

2.10 University Questions and Answers

May 2015

Q. 1 Describe the different types of attributes one may come across in a data mining data set with two examples of each type. *(Ans. : Refer section 2.1)* **(5 Marks)**

Q. 2 Consider the following data points : 13, 15, 16, 19, 20, 20, 21, 22, 22, 25, 25, 25, 25, 30, 33, 33, 35, 35, 35, 35, 36, 40, 45, 46, 52, 70.

(a) What is the mean of the data? What is the median?

(b) What is the mode of the data?

(c) What is the midrange of the data?

(d) Can you find (roughly) the first quartile (Q1) and the third quartile (Q3) of the data ?

(e) Show a boxplot of the data. **(10 Marks)**

(Ans. : Refer Ex. 2.2.2)

Q. 3 Why is Data Preprocessing required ? Explain the different steps involved in Data Preprocessing. *(Ans. :Sections 2.6.1 and 2.6.2)* **(10 Marks)**

Q. 4 Describe the different visualization techniques that can be used in data mining. *(Ans. : Sections 2.3)* **(10 Marks)**

Dec. 2015

Q. 5 Explain different visualization techniques that can be used in data mining. **(10 Marks)**
(Ans. : Refer Section 2.3)

Q. 6 Why is Data Preprocessing required ? Explain the different steps involved in data pre-processing. *(Ans. : Sections 2.6.1 and 2.6.2)* **(10 Marks)**

May 2016

Q. 7 What is data preprocessing? Explain the different methods for the data cleansing phase. **(5 Marks)**
(Ans. : Sections 2.6.1 and 2.6.2)

Q. 8 Partition the given data into 4 bins using Equi-depth binning method and perform smoothing according to the following methods. **(10 Marks)**

Smoothing by bin mean

Smoothing by bin median

Smoothing by bin boundaries

Data : 11,13,13,15,15,l6,19,20,20,20,21,21,22,23,24,30,40,45,45,45,71,72, 73, 75.

(Ans. : Refer Ex. 2.7.1)

Q. 9 For set of data points :

Data : 11,13,13,15,15,l6,19,20,20,20,21,21,22,23,24,30,40,45,45,45,71,72, 73, 75.

(a) For the Mean, Median and Mode. *(Ans. : Refer Section 2.2.1)*

(b) Show a boxplot of the data. Clearly indicating the five number summary.

(Ans. : Refer Section 2.2.2) **(10 Marks)**

Dec. 2016

Q. 10 Clearly explain the data preprocessing phase for data mining.
(Ans. : Refer section 2.6) **(5 Marks)**

Q. 11 Partition the given data into 4 bins using Equi-depth binning method and perform smoothing according to the following methods.

Smoothing by bin mean

Smoothing by bin median

Smoothing by bin boundaries

Data 11, 13, 13, 15, 15, 16, 19 20, 20, 20, 21, 21, 22, 23, 24, 30, 40, 45, 45, 45, 71, 72, 73, 75

(Ans. : Refer Ex. 2.7.1) **(10 Marks)**

Q. 12 For the same set of data points

Data 11, 13, 13, 15, 15, 16, 19 20, 20, 20, 21, 21, 22, 23, 24, 30, 40, 45, 45, 45, 71, 72, 73, 75

(a) Find mean, Median and mode.

(b) Show a box plot of the data. Clearly indicating the five-number summary.

(Ans. : Refer Ex. 2.2.1)

(10 Marks)

...Chapter Ends

❑❑❑

CHAPTER 3

Classification

Syllabus

Basic Concepts; Classification methods: 1. Decision Tree Induction: Attribute Selection Measures, Tree pruning. 2. Bayesian Classification: Naïve Bayes" Classifier. Prediction : Structure of regression models; Simple linear regression, Multiplelinear regression. Accuracy and Error measures, Precision, Recall, Holdout, Random Sampling, Cross Validation.

Syllabus Topic : Basic Concept

3.1 Basic Concept : Classification

- 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 applications**
  - 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

MU - Dec. 2015

3.1.1 Classification Problem

- Suppose a database D is given as $D=\{t_1, t_2,\ldots,t_n\}$ and a set of desired classes are $C=\{C_1,\ldots,C_m\}$, 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.

3.1.2 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.

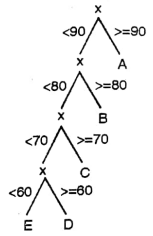


Fig. 3.1.1 : Classification of grading

MU - Dec. 2015

3.1.3 Classification is a Two Step Process

1. Model construction

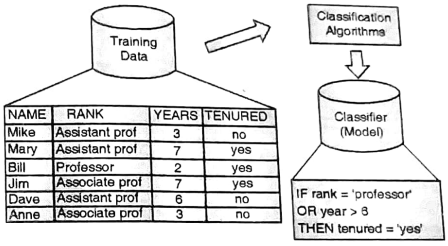
- Every sample tuple or object has assigned a predefined class label
- Those set of sample tuples or subset data set is known as training data set.
- The constructed model based on training data set is represented as classification rules, decision trees or mathematical formulae.

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

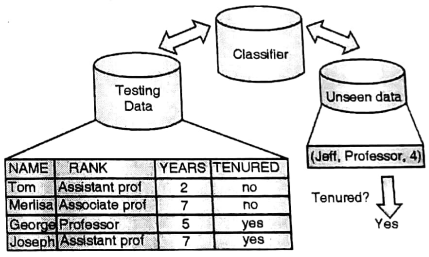
Classification process : (1) Model construction



NAME	RANK	YEARS	TENURED
Mike	Assistant prof	3	no
Mary	Assistant prof	7	yes
Bill	Professor	2	yes
Jim	Associate prof	7	yes
Dave	Assistant prof	6	no
Anne	Associate prof	3	no

Fig. 3.1.2 : Learning : Training data are analyzed by a classification algorithm

Classification process : (2) Model usage (Use the model in prediction)



NAME	RANK	YEARS	TENURED
Tom	Assistant prof	2	no
Merlisa	Associate prof	7	no
George	Professor	5	yes
Joseph	Assistant prof	7	yes

Fig. 3.1.3 : 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 ?

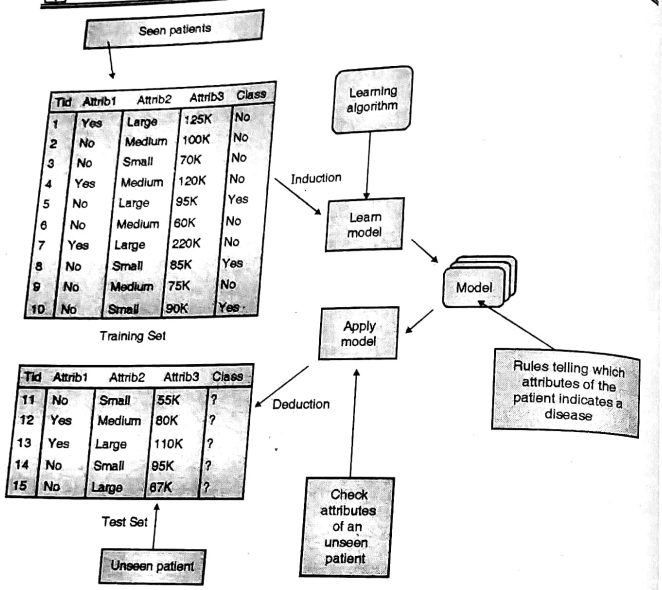


Fig. 3.1.4

3.1.4 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 is 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.

3.1.5 Issues Regarding Classification and Prediction

MU - Dec. 2015

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 rules :**
  - Decision tree size
  - Compactness of classification rules

Syllabus Topic : Classification Methods

3.2 Classification Methods

Classifications methods are given below :

1. **Decision Tree Induction :** Attribute selection measures, tree pruning
2. **Bayesian Classification :** Naïve Bayes' classifier

Syllabus Topic : Decision Tree Induction

3.2.1 Decision Tree Induction

- 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

3.2.1(A) 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 to 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 re-represented as sets of if-then rules to improve readability.

3.2.1(B) Decision Tree Representation

Decision tree classifier has tree 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.

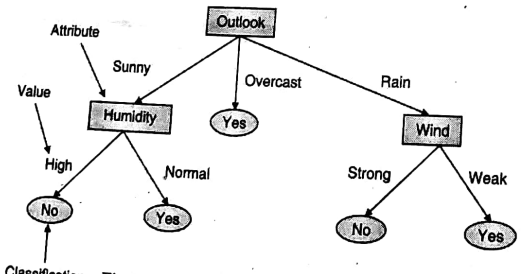


Fig. 3.2.1: Representation of decision tree

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 :
  - IFoutlook = sunny ∧ humidity = normal THEN play tennis = Yes
  - IFoutlook = sunny ∧ humidity = high THEN play tennis = No
  - IFoutlook = overcast THEN play tennis = Yes
  - IFoutlook = rain ∧ wind = weak THEN play tennis = Yes
  - IFoutlook = rain ∧ wind = strong THEN play tennis = No

Syllabus Topic : Attribute Selection Measure

3.2.1(C) Attribute Selection Measure

(1) Gini index (IBM Intelligent Miner)

- Suppose all attributes are continuous-valued.
- Assume that each value of attribute has many possible split.
- It can be adapted for categorical attributes.
- An alternative method to information gain is called the **gini index.**
- Gini is used in CART (Classification and Regression Trees), IBM's Intelligent Miner system, SPRINT (Scalable PaRallelizable INduction of decision Trees).
- If a data set T contains examples from n classes, gini index, gini(T) is defined as

$$\text{gini}(T) = 1 - \sum_{j=1}^{n} p_j^2$$

Where, p_j is the relative frequency of class j in T.

gini(T) is minimized if the classes in T are skewed.

- After splitting T into two subsets T_1 and T_2 with sizes N_1 and N_2, the gini index of the split data is defined as

$$\text{gini}_{split}(T) = \frac{N_1}{N}\text{gini}(T_1) + \frac{N_2}{N}\text{gini}(T_2)$$

- For each attribute, each of the possible binary splits is considered. For a discrete-valued attribute, the attribute providing smallest $\text{gini}_{split}(T)$ is chosen to split the node.For continuous-valued attributes, each possible split-point must be considered.

Example using gini index :

Consider the following dataset D,

Outlook	Temperature	Humidity	Windy	Play ?
Sunny	Hot	High	False	No
Sunny	Hot	High	True	No
Overcast	Hot	High	False	Yes
Rainy	Mild	High	False	Yes
Rainy	Cool	normal	False	Yes
Rainy	Cool	Normal	True	No
Overcast	Cool	Normal	True	Yes
Sunny	Mild	High	False	No
Sunny	Cool	Normal	False	Yes
Rainy	Mild	Normal	False	Yes
Sunny	Mild	Normal	True	Yes
overcast	Mild	High	True	Yes
overcast	Hot	Normal	False	Yes
rainy	Mild	High	True	No

- The total for node D is :

$$gini\ (D) = 1 - \text{sum}\ (p1^2, p2^2, \ldots pn^2) \quad \ldots(3.2.1)$$

Where, p1....n are the frequency ratios of class 1....n in D.

In the above example, there are only two classes : 1) Yes 2) No

Therefore n=2

So the gini index for the entire set :

$$= 1 - ([9/14]^2 + [5/14]^2)$$
$$= 1 - (0.413 + 0.127)$$
$$= 0.459$$

- To find the splitting criterion for the tuples in D, compute the gini index for each attribute.
- The gini value of a split of D into subsets D_1, D_2,.....,D_n is :

Split (D) = N_1/N gini (D_1) + N_2/N gini (D_2) +.....+ N_n/N gini (D_n) ...(3.2.2)

Where, N = Size of dataset D

$N_1, N_2, \ldots N_n$ = Size of each subset.

- Consider First attribute as outlook from dataset D

E.g. Outlook splits into 5 tuples for sunny, 4 tuples for overcast, 5 tuples for rainy :

So $N_1 = 5$, $N_2 = 4$ and $N_3 = 5$

Split = 5/14 gini (sunny) + 4/14 gini (overcast) + 5/14 gini (rainy) ...(3.2.3)

- Now consider the dataset D_1 with only tuples having outlook = "sunny"

Outlook	Temperature	Humidity	Windy	Play ?
Sunny	Hot	High	False	No
Sunny	Hot	High	True	No
Sunny	Mild	High	False	No
Sunny	Cool	Normal	False	Yes
Sunny	Mild	Normal	True	Yes

Using Equation(3.2.1),

$$\text{Gini(sunny)} = 1 - \text{sum}\ ([2/5]^2, [3/5]^2) = 1 - 0.376 = 0.624$$

Similarly calculate for "overcast" and "rainy"

$$\text{Overcast} = 1 - \text{sum}\ (4/4^2, 0/4^2) = 0.0$$
$$\text{Rainy} = 1 - \text{sum}\ ([3/5]^2, [2/5]^2) = 0.624$$

From Equation(3.2.3)

Split = 5/14 gini (sunny) + 4/14 gini (overcast) + 5/14 gini (rainy)

Split = (5/14 * 0.624) + 0 + * (5/14 * 0.624)

Split = 0.446

The attribute that generates the smallest gini split value is chosen to split the node on.

(2) Information gain (ID3/C4.5)

- All attributes are believed to be categorical.
- It can be adapted for continuous-valued attributes.
- The attribute which has the highest information gain is selected for split.
- Assume there are two classes, P and N
- Suppose we have S samples, out of these p samples belongs to class P and n samples belongs to class N.
- The amount of information, needed to decide if random example in S belongs to P or N is defined as

$$I(p, n) = -\frac{p}{p+n}\log_2\frac{p}{p+n} - \frac{n}{p+n}\log_2\frac{n}{p+n}$$

- Assume that using attribute A, a set S will be partitioned into sets $\{S_1, S_2, \ldots, S_v\}$
- If S_i contains p_i examples of P and n_i examples of N, the entropy, or the expected information needed to classify objects in all subtrees S_i is

$$E(A) = \sum_{i=1}^{v} \frac{p_i + n_i}{p+n} I(p_i, n_i)$$

Entropy (E) :

- Expected amount of information (in bits) needed to assign a class to a randomly drawn object in S under the optimal, shortest-length code.
- **Calculate information gain i.e. gain(A) :** Measures reduction in entropy achieved because of the split. Choose the split that achieves most reduction (maximizes GAIN)

$$Gain\ (A) = I(p, n) - E(A)$$

(3) Gain ratio

- It is an alteration of the information gain that reduces its favouritism on high-branch attributes.
- Gain ratio should be big when data is evenly spread and small when all data belong to one branch. So it considers number of branches and size of branches when it selects attribute to split.
- Intrinsic information is the entropy of distribution of instances into branches.

$$Intrinsic\ Info(S,A) = -\sum \frac{|S_i|}{|S|} \log_2 \frac{|S_i|}{|S|}$$

- Gain ratio normalizes info gain by using following formula:

$$Gain\ Ratio\ (S,A) = \frac{Gain(S,A)}{Intrinsic\ Info(S,A)}$$

3.2.1(D) Algorithm for Inducing a Decision Tree

The Basic ideas behind ID3:

- **C4.5** is an extension of ID3.
- **C4.5** accounts for unavailable values, continuous attribute value ranges, pruning of decision trees, rule derivation and so on.
- **C4.5** is a designed by Quinlan to address the following issues not given by ID3 :
  - It avoids over fitting the data.
  - It determines the depth of decision tree and reduces the error pruning.
  - It also handles continuous value attributes e.g. Salary or temperature.
  - It works for missing value attribute and handles suitable attribute selection measure.
  - It gives better the efficiency of computation. The algorithm to generate decision tree is given by Jiawei Han et al. as below :

Algorithm : Generate_decision_tree - Generate a decision tree from the training tuples of data partition, D.

Input :

- Data partition, D, which is a set of training tuples and their associated class labels;
- Attribute_list, the set of candiadate attributes;
- Attribute_selection_method, a procedure to determine the splitting criterion that "best" partitions the data tuples into individual classes. This criterion consists of a splitting_attribute and, possibly, either a split-point or splitting subset.

Output :

A decision tree.

Method :

```
1.  create a node N;
2.  if tuples in D are all of the same class, C, then
3.          return N as a leaf node labeled with the class C;
4.  if attribute_list is empty then
5.          return N as a leaf node labeled with the majority class in D; // majority voting
6.  apply Attribute_selection_method(D, attribute_list) to find the "best" splitting_criterion;
7.  label node N with splitting_criterion;
8.  if splitting_attribute is discrete-valued and
          Multiway splits allowed then // not restricted to binary trees
9.          Attribute_list ← attribute_list – splitting_attribute; // remove splitting_attribute
10. for each outcome j of splitting_criterion
    // partition the tuples and grow subtrees for each partition
11. let Dj be the set of data tuples in D satisfying outcome j; // a partition
12. if Dj is empty then
13.         attach a leaf labeled with the majority class in D to node N;
14.     else attach the node returned by Generate_decision_tree (Dj, attribute_list) to node N; end for
15. return N;
```

Syllabus Topic : Tree Pruning

3.2.2 Tree Pruning

- Because of noise or outliers, the generated tree may over-fit due to many branches.
- To avoid over-fitting, prune the tree so that it is not too specific.
  - **Pre-pruning :**
    - Start pruning in the beginning while building the tree itself.
    - Stop the tree construction in early stage.

- Avoid splitting a node by checking the threshold with the goodness measure falling below a threshold.
- Selection of correct threshold is difficult in pre-pruning.

o **Post-pruning :**

- Build the full tree then start pruning, remove the branches.
- Use different set of data than training data set to get the best pruned tree.

3.2.3 Examples of ID3

MU - Dec. 2015

Ex. 3.2.1 : Apply ID3 on the following training dataset from all electronics customer database and extract the classification rule from the tree

Table P. 3.2.1 : Training data of customer

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

Soln. :

Class P : buys_computer = "yes"

Class N: buys_computer = "no"

Total number of records 14.

Count the number of records with "yes" class and "no" class.

So number of records with "yes" class = 9 and "no" class = 5

So Information gain = $I(p, n) = -\frac{p}{p+n}\log_2\frac{p}{p+n} - \frac{n}{p+n}\log_2\frac{n}{p+n}$

$I(p, n) = I(9, 5) = -(9/14)\log_2(9/14) - (5/14)\log_2(5/14) = 0.940$

Step 1 : Compute the entropy for age :

For age <=30,

p_i = with "yes" class = 2 and n_i = with "no" class = 3

Therefore, $I(p_i, n_i) = I(2,3) = 0.971$.

Similarly for different age ranges $I(p_i, n_i)$ is calculated as given below :

Age	p_i	n_i	$I(p_i, n_i)$
<=30	2	3	0.971
31...40	4	0	0
>40	3	2	0.971

So, the expected information needed to classify a given sample if the samples are partitioned according to age is,

Calculate entropy using the values from the above table and the formula given below :

$$E(A) = \sum_{i=1}^{v} \frac{p_i + n_i}{p+n} I(p_i, n_i)$$

$$E(age) = \frac{5}{14} I(2, 3) + \frac{4}{14} I(4, 0) + \frac{5}{14} I(3, 2) = 0.694$$

Hence

$$Gain(age) = I(p, n) - E(age)$$
$$= 0.940 - 0.694 = 0.246$$

Similarly,

Gain (income) = 0.029

Gain (student) = 0.151

Gain (credit_rating) = 0.048

Now the age has the highest information gain among all the attributes, so select age as test attribute and create the node as age and show all possible values of age for further splitting.

Since Age has three possible values, the root node has three branches (<=30, 31...40, >40).

Step 2 :

The next question is "what attribute should be tested at the Age branch node?" Since we have used Age at the root, now we have to decide on the remaining three attributes: income, student, or credit_rating.

Consider Age: <=30 and count the number of tuples from the original given training set

$S_{<=30}$ = 5 (Age: <=30)

Age	Income	Student	Credit_rating	Buys_computer
<=30	High	No	Fair	No
<=30	High	No	Excellent	No
<=30	Medium	No	Fair	No
<=30	Low	Yes	Fair	Yes
<=30	Medium	Yes	Excellent	Yes

Note : Refer above table :
Total number of Yes tuple =2 and total number of No tuple = 3
$I(p, n) = I(2, 3) = -(2/5)\log_2(2/5) - (3/5)\log_2(3/5) = 0.971$

(i) Compute the entropy for income: (High, medium, low)

For Income = High,

p_i = with "yes" class = 0 and n_i = with "no" class = 2

Therefore , $I(p_i, n_i) = I(0,2) = -(0/2)\log_2(0/2) - (2/2)\log_2(2/2) = 0$.

Similarly for different age ranges $I(p_i, n_i)$ is calculated as given below :

Income	p_i	n_i	$I(p_i, n_i)$
High	0	2	0
Medium	1	1	1
Low	1	0	0

Calculate entropy using the values from the above table and the formula given as :

$$E(A) = \sum_{i=1}^{v} \frac{p_i + n_i}{p + n} I(p_i, n_i)$$

E (Income) = 2/5 * I (0, 2) + 2/5* I(1, 1) + 1/5 *I(1, 0) = 0.4

Note : $S_{<=30}$ is the total training set.

Hence

$$\text{Gain}(S_{<=30}, \text{Income}) = I(p, n) - E(\text{Income})$$
$$= 0.971 - 0.4 = 0.571$$

(ii) Compute the entropy for Student : (No , yes)

For Student = No,

p_i = with "yes" class = 0 and n_i = with "no" class = 3

Therefore, $I(p_i, n_i) = I(0,3) = -(0/3)\log_2(0/3) - (3/3)\log_2(3/3) = 0$.

Similarly for different outlook ranges $I(p_i, n_i)$ is calculated as given below :

Student	p_i	n_i	$I(p_i, n_i)$
No	0	3	0
Yes	2	0	0

Calculate Entropy using the values from the above table and the formula given below

$$E(A) = \sum_{i=1}^{v} \frac{p_i + n_i}{p + n} I(p_i, n_i)$$

E (Student) = 3/5 * I (0, 3) + 2/5* I(2, 0) = 0

Note : $S_{<=30}$ is the total training set.

Hence

$$\text{Gain}(S_{<=30}, \text{Student}) = I(p, n) - E(\text{Student})$$
$$= 0.971 - 0 = 0.971$$

(iii) Compute the entropy for credit_rating : (Fair, excellent)

For credit_rating = Fair,

p_i = with "yes" class = 1 and n_i = with "no" class = 2

Therefore $I(pi, ni) = I(1, 2) = -(1/3)\log2(1/3) - (2/3)\log2(2/3)$

$= 0.918$

Similarly for different outlook ranges $I(p_i, n_i)$ is calculated as given below :

Credit_rating	p_i	n_i	$I(p_i, n_i)$
Fair	1	2	0.918
Excellent	1	1	1

Calculate entropy using the values from the above table and the formula given below :

$$E(A) = \sum_{i=1}^{v} \frac{p_i + n_i}{p + n} I(p_i, n_i)$$

E (Credit_rating) = 3/5 * I (1, 2) + 2/5* I(1, 1) = 0.951

Note : $S_{<=30}$ is the total training set.

Hence

$$\text{Gain}(S_{<=30}, \text{credit\_rating}) = I(p, n) - E(\text{credit\_rating})$$
$$= 0.971 - 0.951 = 0.02$$

Therefore,

$\text{Gain}(S_{<=30}, \text{student}) = 0.970$

$\text{Gain}(S_{<=30}, \text{income}) = 0.570$

$\text{Gain}(S_{<=30}, \text{credit\_rating}) = 0.02$

Student has the highest gain; therefore, it is below Age: "<=30".

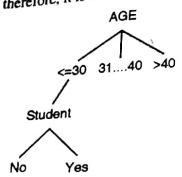


Fig. P. 3.2.1(a)

Step 3 :

Consider now only income and credit rating for age: 31...40 and count the number of tuples from the original given training set

$S_{31...40}$ = 4 (age : 31...40)

Age	Income	Student	Credit_rating	Buys_computer
31...40	High	No	Fair	Yes
31...40	Low	Yes	Excellent	Yes
31...40	Medium	No	Excellent	Yes
31...40	High	Yes	Fair	Yes

Since for the attributes income and credit_rating , buys_computer = yes, so assign class 'yes' to 31...40

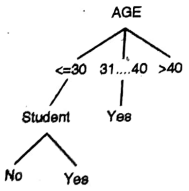


Fig. P. 3.2.1(b)

Step 4 :

Consider income and credit_rating for age: >40 and count the number of tuples from the original given training set

$S_{>40}$ = (age: >40)

Age	Income	Student	Credit_rating	Buys_computer
>40	Medium	No	Fair	Yes
>40	Low	Yes	Fair	Yes
>40	Low	Yes	Excellent	No
>40	Medium	Yes	Fair	Yes

Age	Income	Student	Credit_rating	Buys_computer
>40	Medium	No	Excellent	No

Consider the above table as the new training set and calculate the Gain for income and credit_rating

Class P : buys_computer = "yes"

Class N: buys_computer = "no"

Total number of records 5.

Count the number of records with "yes" class and "no" class.

So number of records with "yes" class = 3 and "no" class = 2

So Information gain = $I(p, n) = -\frac{p}{p+n}\log_2\frac{p}{p+n} - \frac{n}{p+n}\log_2\frac{n}{p+n}$

$I(p, n) = I(3, 2) = -(3/5)\log_2(3/5) - (2/5)\log_2(2/5) = 0.970$

(iv) Compute the entropy for credit_rating :

For credit_rating = Fair

p_i = with "yes" class = 3 and n_i = with "no" class = 0

Therefore , $I(p_i , n_i) = I(3,0) = 0$.

For credit_rating = Excellent

p_i = with "yes" class = 0 and n_i = with "no" class = 2

Therefore, $I(p_i , n_i) = I(0,2) = 0$

Similarly for different age ranges $I(p_i , n_i)$ is calculated as given below :

Credit_rating	p_i	n_i	$I(p_i, n_i)$
Fair	3	0	0
Excellent	0	2	0

Calculate entropy using the values from the above table and the formula given below

$$E(A) = \sum_{i=1}^{v} \frac{p_i + n_i}{p+n} I(p_i, n_i)$$

$$E(\text{Credit\_rating}) = \frac{3}{5} I(3,0) + \frac{2}{5} I(0,2) = 0$$

Hence

$$\text{Gain}(S_{>40}, \text{credit\_rating}) = I(p, n) - E(\text{credit\_rating})$$
$$= 0.970 - 0 = 0.970$$

(v) Compute the entropy for income: (High, medium, low)

For Income = High,

p_i = with "yes" class = 0 and n_i = with "no" class = 0

Therefore, $I(p_i , n_i) = I(0,0) = 0$

Similarly for different outlook ranges $I(p_i, n_i)$ is calculated as given below :

Income	p_i	n_i	$I(p_i, n_i)$
High	0	0	0
Medium	2	1	0.918
Low	1	1	1

Calculate Entropy using the values from the above table and the formula given below

$E(\text{Income}) = 0/5 * I(0, 0) + 3/5 * I(2, 1) + 2/5 * I(1, 1) = 0.951$

Note : $S_{>40}$ is the total training set.

Hence

$$Gain(S_{>40}, \text{income}) = I(p, n) - E(\text{income})$$
$$= 0.970 - 0.951 = 0.019$$

Therefore,

$$Gain(S_{>40}, \text{income}) = 0.019$$
$$Gain(S_{>40}, \text{Credit\_rating}) = 0.970$$

Credit_rating has the highest gain; therefore, it is below Age: ">40".

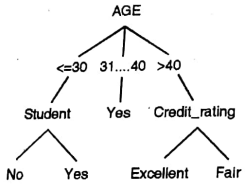


Fig. P. 3.2.1(c)

Output : A Decision Tree for "buys_computer"

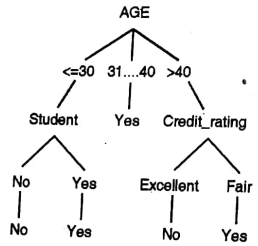


Fig. P. 3.2.1(d) : Decision tree for "buys computer"

Extracting classification rules from trees :

Example :

IF age = "<=30" AND student = "no" THEN buys_computer = "no"
IF age = "<=30" AND student = "yes" THEN buys_computer = "yes"
IF age = "31…40" THEN buys_computer = "yes"
IF age = ">40" AND credit_rating = "excellent" THEN buys_computer = "no"
IF age = ">40" AND credit_rating = "fair" THEN buys_computer = "yes"

Ex. 3.2.2 : The weather attributes are outlook, temperature, humidity, and wind speed. They can have the following values :

Outlook = {sunny, overcast, rain}
temperature = {hot, mild, cool}
humidity = {high, normal}
wind = {weak, strong}

Sample data set S are :

Table P. 3.2.2 : Training data set for Play Tennis

Day	Outlook	Temperature	Humidity	Wind	Play ball
1	Sunny	Hot	High	Weak	No
2	Sunny	Hot	High	Strong	No
3	Overcast	Hot	High	Weak	Yes
4	Rain	Mild	High	Weak	Yes
5	Rain	Cool	Normal	Weak	Yes
6	Rain	Cool	Normal	Strong	No
7	Overcast	Cool	Normal	Strong	Yes
8	Sunny	Mild	High	Weak	No
9	Sunny	Cool	Normal	Weak	Yes
10	Rain	Mild	Normal	Weak	Yes
11	Sunny	Mild	Normal	Strong	Yes
12	Overcast	Mild	High	Strong	Yes
13	Overcast	Hot	Normal	Weak	Yes
14	Rain	Mild	High	Strong	No

We need to find which attribute will be the root node in our decision tree. The gain is calculated for all four attributes using formula of gain(A) in Section 3.2.1(C).

Soln. :

Class P : Playball = "yes"

Class N : Playball = "no"

Total number of records 14.

Count the number of records with "yes" class and "no" class.

So number of records with "yes" class = 9 and "no" class = 5

So Information gain = $I(p, n) = -\frac{p}{p+n}\log_2\frac{p}{p+n} - \frac{n}{p+n}\log_2\frac{n}{p+n}$

$I(p, n) = I(9, 5) = -(9/14)\log_2(9/14) - (5/14)\log_2(5/14)$

$= (-0.643) * (-0.637) + (-0.357) * (-1.485)$

$= 0.409 + 0.530 = 0.940$

Step 1 : Compute the entropy for outlook : (Sunny, overcast , rain)

For outlook = sunny,

p_i = with "yes" class = 2 and n_i = with "no" class = 3

Therefore , $I(p_i, n_i) = I(2,3) = -(2/5)\log_2(2/5) - (3/5)\log_2(3/5) = 0.971$.

Similarly for different outlook ranges $I(p_i, n_i)$ is calculated as given below :

Outlook	p_i	n_i	$I(p_i, n_i)$
Sunny	2	3	0.971
Overcast	4	0	0
Rain	3	2	0.971

Calculate entropy using the values from the above table and the formula given below :

$$E(A) = \sum_{i=1}^{v} \frac{p_i + n_i}{p+n} I(p_i, n_i)$$

$$E(\text{outlook}) = \frac{5}{14} I(2, 3) + \frac{4}{14} I(4, 0) + \frac{5}{14} I(3, 2) = 0.694$$

Note : T is the total training set.

Hence $\text{Gain}(T, \text{outlook}) = I(p, n) - E(\text{outlook}) = 0.940 - 0.694 = 0.246$

Similarly, $\text{Gain}(T, \text{Temperature}) = 0.029$

$\text{Gain}(T, \text{Humidity}) = 0.151$

$\text{Gain}(T, \text{Wind}) = 0.048$

Outlook shows the highest gain, so it is used as the decision attribute in the root node.

As Outlook has only values "sunny, overcast, rain", the root node has three branches

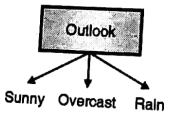


Fig. P. 3.2.2(a)

Step 2 : As attribute outlook at root, we have to decide on the remaining three attribute for sunny branch node.

Consider outlook = Sunny and count the number of tuples from the original given training set

$S_{sunny} = \{1, 2, 8, 9, 11\}$

= 5 (From Table. P.3.2.2, outlook = sunny)

Day	Outlook	Temperature	Humidity	Wind	Play ball
1	Sunny	Hot	High	Weak	No
2	Sunny	Hot	High	Strong	No
8	Sunny	Mild	High	Weak	No
9	Sunny	Cool	Normal	Weak	Yes
11	Sunny	Mild	Normal	Strong	Yes

Note : Refer Table P. 3.2.2 :

Total number of Yes tuple = 2 and total number of No tuple= 3

$I(p, n) = I(2, 3) = -(2/5)\log_2(2/5) - (3/5)\log_2(3/5) = 0.971$

(i) Compute the entropy for temperature : (Hot, mild, cool)

For Temperature = Hot,

p_i = with "yes" class = 0 and n_i = with "no" class = 2

Therefore , $I(p_i, n_i) = I(0,2) = -(0/2)\log_2(0/2) - (2/2)\log_2(2/2) = 0$

Similarly for different outlook ranges $I(p_i, n_i)$ is calculated as given below :

Temperature	p_i	n_i	$I(p_i, n_i)$
Hot	0	2	0
Mild	1	1	1
Cool	1	0	0

Calculate Entropy using the values from the above table and the formula given below :

$$E(A) = \sum_{i=1}^{v} \frac{p_i + n_i}{p+n} I(p_i, n_i)$$

$E(\text{Temperature}) = 2/5 * I(0, 2) + 2/5 * I(1, 1) + 1/5 * I(1, 0) = 0.4$

Note : T_{sunny} is the total training set.

Hence

$\text{Gain}(T_{sunny}, \text{temperature}) = I(p, n) - E(\text{temperature})$

$= 0.971 - 0.4 = 0.571$

(ii) Compute the entropy for humidity : (High, normal)

For Humidity = High,

p_i = with "yes" class = 0 and n_i = with "no" class = 3

Therefore, $I(pi, ni) = I(0,3) = -(0/3)\log2(0/3) - (3/3)\log2(3/3) = 0$

Similarly for different outlook ranges $I(p_i, n_i)$ is calculated as given below :

Humidity	p_i	n_i	$I(p_i, n_i)$
High	0	3	0
Normal	2	0	0

Calculate Entropy using the values from the above table and the formula given below

$$E(A) = \sum_{i=1}^{v} \frac{p_i + n_i}{p + n} I(p_i, n_i)$$

$$E(\text{Humidity}) = 3/5 * I(0, 3) + 2/5 * I(2, 0) = 0$$

Note : T_{sunny} is the total training set.

Hence

$$\text{Gain}(T_{sunny}, \text{Humidity}) = I(p, n) - E(\text{Humidity})$$
$$= 0.971 - 0 = 0.971$$

(iii) Compute the entropy for wind : (Weak, strong)

For wind = weak,

p_i = with "yes" class = 1 and n_i = with "no" class = 2

Therefore, $I(p_i, n_i) = I(1,2) = -(1/3)\log_2(1/3) - (2/3)\log_2(2/3) = 0.918$

Similarly for different outlook ranges $I(p_i, n_i)$ is calculated as given below :

Wind	p_i	n_i	$I(p_i, n_i)$
Weak	1	2	0.918
Strong	1	1	1

Calculate Entropy using the values from the above table and the formula given as:

$$E(A) = \sum_{i=1}^{v} \frac{p_i + n_i}{p + n} I(p_1, n_1)$$

$$E(\text{Wind}) = 3/5 * I(1, 2) + 2/5 * I(1, 1) = 0.951$$

Note : T_{sunny} is the total training set.

Hence $\text{Gain}(T_{sunny}, \text{Wind}) = I(p, n) - E(\text{Wind})$

$$= 0.971 - 0.951 = 0.02$$

Therefore,

$\text{Gain}(T_{sunny}, \text{Humidity}) = 0.970$

$\text{Gain}(T_{sunny}, \text{Temperature}) = 0.570$

$\text{Gain}(T_{sunny}, \text{Wind}) = 0.02$

Humidity has the highest gain; therefore, it is below Outlook = "sunny".

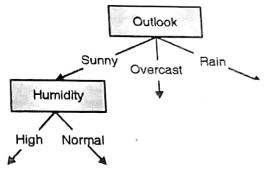


Fig. P. 3.2.2(b)

Step 3 : Consider now only temperature and wind for outlook = Overcast and count the number of tuples from the original given training set

$T_{overcast} = \{3,7,12,13\}$

= 4 (From Table. P.3.2.2, outlook = overcast)

Day	Outlook	Temperature	Humidity	Wind	Play ball
3	Overcast	Hot	High	Weak	Yes
7	Overcast	Cool	Normal	Strong	Yes
12	Overcast	Mild	High	Strong	Yes
13	Overcast	Hot	Normal	Weak	Yes

Since for the attributes temperature and wind, playball = yes, so assign class 'yes' to overcast.

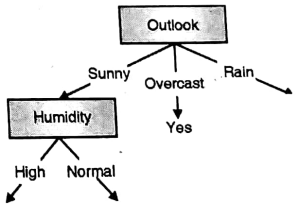


Fig. P. 3.2.2(c)

Step 4 : Consider temperature and wind for outlook = Rain and count the number of tuples from the original given training set

T_{rain} = {4, 5, 6, 10, 14}

= 5 (From Table. P.3.2.2, outlook = rain)

Day	Outlook	Temperature	Humidity	Wind	Play ball
4	Rain	Mild	High	Weak	Yes
5	Rain	Cool	Normal	Weak	Yes
6	Rain	Cool	Normal	Strong	No
10	Rain	Mild	Normal	Weak	Yes
14	Rain	Mild	High	Strong	No

Consider the above table as the new training set and calculate the Gain for temperature and Wind.

Class P : Playball = "yes"

Class N : Playball = "no"

Total number of records 5

Count the number of records with "yes" class and "no" class.

So number of records with "yes" class = 3 and "no" class = 2

So Information gain = $I(p, n) = -\frac{p}{p+n}\log_2\frac{p}{p+n} - \frac{n}{p+n}\log_2\frac{n}{p+n}$

$I(p, n) = I(3, 2) = -(3/5)\log_2(3/5) - (2/5)\log_2(2/5) = 0.970$

(iv) Compute the entropy for Wind :

For Wind = Weak

p_i = with "yes" class = 3 and n_i = with "no" class = 0

Therefore, $I(p_i, n_i) = I(3,0) = 0$.

For Wind = strong

p_i = with "yes" class = 0 and n_i = with "no" class = 2

Therefore, $I(p_i, n_i) = I(0, 2) = 0$

Similarly for different outlook ranges $I(p_i, n_i)$ is calculated as given below :

Wind	p_i	n_i	$I(p_i, n_i)$
Weak	3	0	0
Strong	0	2	0

Calculate entropy using the values from the above table and the formula given below :

$$E(A) = \sum_{i=1}^{v} \frac{p_i + n_i}{p+n} I(p_i, n_i)$$

$$E(\text{Wind}) = \frac{3}{5} I(3, 0) + \frac{2}{5} I(0, 2) = 0$$

Hence $\text{Gain}(T_{rain}, \text{Wind}) = I(p, n) - E(\text{Wind})$

$= 0.970 - 0 = 0.970$

(v) Compute the entropy for Temperature : (Hot, mild , cool)

For Temperature = Hot.

p_i = with "yes" class = 0 and n_i = with "no" class = 0

Therefore, $I(p_i, n_i) = I(0,0) = 0$

Similarly for different outlook ranges $I(p_i, n_i)$ is calculated as given below :

Temperature	p_i	n_i	$I(p_i, n_i)$
Hot	0	0	0
Mild	2	1	0.918
Cool	1	1	1

Calculate Entropy using the values from the above table and the formula given below :

$$E(A) = \sum_{i=1}^{v} \frac{p_i + n_i}{p+n} I(p_i, n_i)$$

$E(\text{Temperature}) = 0/5 * I(0, 0) + 3/5 * I(2, 1) + 2/5 * I(1, 1)$

$= 0.951$

Note : T_{Rain} is the total training set.

Hence

$\text{Gain}(T_{Rain}, \text{temperature}) = I(p, n) - E(\text{temperature})$

$= 0.970 - 0.951 = 0.019$

Therefore,

$\text{Gain}(T_{rain}, \text{Temperature}) = 0.019$

$\text{Gain}(T_{rain}, \text{Wind}) = 0.970$

Wind has the highest gain; therefore, it is below outlook = "rain".

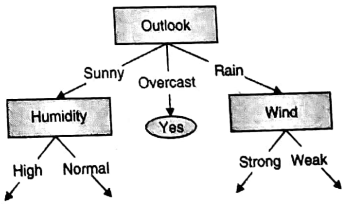


Fig. P. 3.2.2(d)

Therefore the final decision tree is :

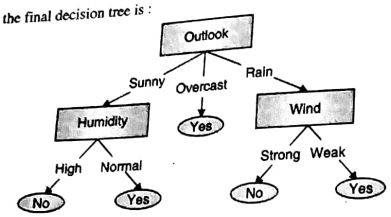


Fig. P. 3.2.2(e) : Decision tree for play tennis

The decision tree can also be expressed in rule format :

- IF outlook = sunny AND humidity = high THEN playball = no
- IF outlook = Sunny AND humidity = normal THEN playball = yes
- IF outlook = overcast THEN playball = yes
- IF outlook = rain AND wind = strong THEN playball = no
- IF outlook = rain AND wind = weak THEN playball = yes

Ex. 3.2.3 : A sample training dataset for stock market is given below. Profit is the class attribute and value is based on age, contest and type.

Age	Contest	Type	Profit
Old	Yes	Swr	Down
Old	No	Swr	Down
Old	No	Hwr	Down
Mid	Yes	Swr	Down
Mid	Yes	Hwr	Down
Mid	No	Hwr	Up
Mid	No	Swr	Up
New	Yes	Swr	Up
New	No	Hwr	Up
New	No	Swr	Up

Soln. :

In the stock market case the decision tree is :

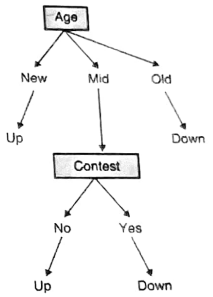


Fig. P. 3.2.3

Ex. 3.2.4 : Using the following training data set. Create classification model using decision-tree and hence classify following tuple :

Tid	Income	Age	Own House
1.	Very High	Young	Yes
2.	High	Medium	Yes
3.	Low	Young	Rented
4.	High	Medium	Yes
5.	Very high	Medium	Yes
6.	Medium	Young	Yes
7.	High	Old	Yes
8.	Medium	Medium	Rented
9.	Low	Medium	Rented
10.	Low	Old	Rented
11.	High	Young	Yes
12.	medium	Old	Rented

Soln. :

Class P : Own house = "yes"

Class N: Own house = "rented"

Total number of records 12

Count the number of records with "yes" class and "rented" class.

So number of records with "yes" class = 7 and "no" class = 5

So Information gain = $I(p, n) = -\frac{p}{p+n}\log_2\frac{p}{p+n} - \frac{n}{p+n}\log_2\frac{n}{p+n}$

$I(p, n) = I(7, 5) = -(7/12)\log_2(7/12) - (5/12)\log_2(5/12) = 0.979$

Step 1 : Compute the entropy for Income : (Very high, high, medium, low)

For Income =Very high,

p_i = with "yes" class = 2 and n_i = with "no" class = 0

Therefore, $I(p_i, n_i) = I(2,0) = 0$

Similarly for different Income ranges $I(p_i, n_i)$ is calculated as given below :

Income	p_i	n_i	$I(p_i, n_i)$
Very high	2	0	0
High	4	0	0
Medium	1	2	0.918
Low	0	3	0

Calculate entropy using the values from the above table and the formula given below :

$$E(A) = \sum_{i=1}^{v} \frac{p_i + n_i}{p+n} I(p_i, n_i)$$

$$E(\text{Income}) = 2/12 * I(2,0) + 4/12 * I(4,0) + 3/12 * I(0,3) = 0.229$$

Note : S is the total training set.

Hence, $\text{Gain}(S, \text{Income}) = I(p, n) - E(\text{Income})$

$= 0.979 - 0.229 = 0.75$

Step 2 : Compute the entropy for Age : (Young , medium, old)

Similarly for different age ranges $I(p_i, n_i)$ is calculated as given below :

Age	p_i	n_i	$I(p_i, n_i)$
Young	3	1	0.811
Medium	3	2	0.971
Old	1	2	0.918

Calculate entropy using the values from the above table and the formula given below :

$$E(A) = \sum_{i=1}^{v} \frac{p_i + n_i}{p+n} I(p_i, n_i)$$

$$E(\text{Age}) = 4/12 * I(3,1) + 5/12 * I(3,2) + 3/12 * I(1,2)$$

$$= 0.904$$

Note : S is the total training set.

Hence

$\text{Gain}(S, \text{age}) = I(p, n) - E(\text{age})$

$= 0.979 - 0.904$

$= 0.075$

Income attribute has the highest gain, therefore it is used as the decision attribute in the root node. Since income has four possible values, the root node has four branches (very high, high, medium, low).

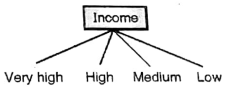


Fig. P. 3.2.4(a)

Step 3 : Since we have used income at the root, now we have to decide on the age attribute.

Consider income = "very high" and count the number of tuples from the original given training set

$S_{\text{very high}} = 2$

Since both the tuples have class label ="yes" , so directly give "yes" as a class label below "very high".

Similarly check the tuples for income="high" and income ="low" , are having the class label "yes" and "rented" respectively.

Now check for income="medium", where number of tuples having "yes" class label is 1 and tuples having "rented" class label are 2.

So put the age label below income="medium".

So the final decision tree is :

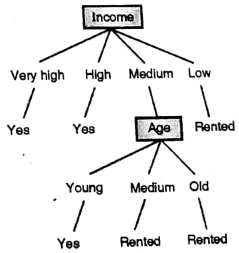
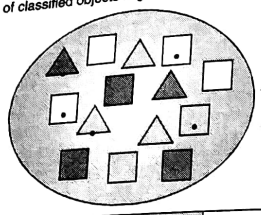


Fig. P. 3.2.4(b)

Ex. 3.2.5 : Data Set: A set of classified objects is given as below. Apply ID3 to generate tree.



	Attribute			
Sr.No	**Colour**	**Outline**	**Dot**	**Shape**
1	Green	Dashed	No	Triangle
2	Green	Dashed	Yes	Triangle
3	Yellow	Dashed	No	Square
4	Red	Dashed	No	Square
5	Red	Solid	No	Square
6	Red	Solid	Yes	Triangle
7	Green	Solid	No	Square
8	Green	Dashed	No	Triangle
9	Yellow	Solid	Yes	Square
10	Red	Solid	No	Square
11	Green	Solid	Yes	Square
12	Yellow	Dashed	Yes	Square
13	Yellow	Soild	No	Square
14	Red	Dashed	yes	Triangle

Soln. :

Class N : Shape = "Triangle"

Class P: Shape = "Square"

Total number of records 14

Count the number of records with "triangle" class and "square" class.

So number of records with "triangle" class = 5 and "square" class = 9

P(square) = 9/14

P(triangle) = 5/14

$$\text{So information gain} = I(p, n) = -\frac{p}{p+n}\log_2\frac{p}{p+n} - \frac{n}{p+n}\log_2\frac{n}{p+n}$$

$$I(p, n) = I(9,5)$$

$$= -(9/14)\log_2(9/14) - (5/14)\log_2(5/14)$$

$$= 0.940$$

Step 1 : Compute the entropy for Color : (Red, green, yellow)

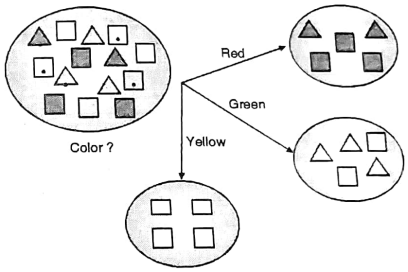


Fig. P. 3.2.5(a)

For color=Red,

p_i = with "square" class = 3 and n_i = with "triangle" class = 2

Therefore, $I(p_i, n_i) = I(3,2) = 0.971$

Similarly for different Color values, $I(p_i, n_i)$ is calculated as given below :

Color	p_i	n_i	$I(p_i, n_i)$
Red	3	2	0.971
Green	2	3	0.971
Yellow	4	0	0

Calculate Entropy using the values from the above table and the formula given below :

$$E(A) = \sum_{i=1}^{v} \frac{p_i + n_i}{p+n} I(p_i, n_i)$$

$$E(\text{Color}) = 5/14 * I(3,2) + 5/14 * I(2,3) + 4/14 * I(4,0)$$

$$= 0.694$$

Note : S is the total training set.

Hence $\text{Gain}(S, \text{color}) = I(p, n) - E(\text{Color}) = 0.940 - 0.694 = 0.246$

Step 2 : Compute the entropy for outline: (Dashed, solid)

Similarly for different outline values, $I(p_i , n_i)$ is calculated as given below :

Outline	p_i	n_i	$I(p_i, n_i)$
Dashed	3	4	0.985
Solid	6	1	0.621

Calculate Entropy using the values from the above table and the formula given below :

$$E(A) = \sum_{i=1}^{v} \frac{p_1 + n_1}{p + n} I(p_1, n_1)$$

E (Outline) = 7/14 * I(3,4)+7/14*I(6,1)=0.803

Note : S is the total training set.

Hence, Gain(S, Outline) = I (p, n) – E (Outline)

= 0.940– 0.803 = 0.137

Step 3 : Compute the entropy for dot: (no, yes)

Similarly for different dot values, $I(p_i , n_i)$ is calculated as given below :

Outline	p_i	n_i	$I(p_i, n_i)$
No	6	2	0.811
Yes	3	3	1

Calculate entropy using the values from the above table and the formula given below :

$$E(A) = \sum_{i=1}^{v} \frac{p_1 + n_1}{p + n} I(p_1, n_1)$$

E (Dot) = 8/14 * I(6,2)+6/14*I(3,3)

= 0.892

Note : S is the total training set.

Hence, Gain(S, dot) = I (p, n) – E (dot)

= 0.940 – 0.892 = 0.048

Therefore, Gain(S, color) = 0.246

Gain(S, outline) = 0.137

Gain(S, dot) = 0.048

As color has highest gain, it should be the root node.

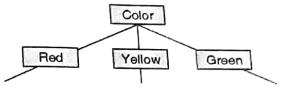


Fig. P. 3.2.5(b)

Step 4 : As attribute color is at the root, we have to decide on the remaining two attribute for red branch node.

Consider color = red and count the number of tuples from the original given training set

	Attribute			Shape
	Color	Outline	Dot	
1.	Red	Dashed	No	Square
2.	Red	Solid	No	Square
3.	Red	Solid	Yes	Triangle
4.	Red	Solid	No	Square
5.	Red	Dashed	Yes	Triangle

Note : Refer above table :

Total number of tuple with "square" class = 3 and total number of No tuple with "triangle" class = 2

$I(p, n) = I(3,2) = -(3/5)\log_2(3/5) - (2/5)\log_2(2/5) = 0.971$

Compute the entropy for outline: (Dashed, solid)

Similarly for different outline values, $I(p_i , n_i)$ is calculated as given below.

Outline	p_i	n_i	$I(p_i, n_i)$
Dashed	1	1	1
Solid	2	1	0.918

Calculate Entropy using the values from the above table and the formula given as :

$$E(A) = \sum_{i=1}^{v} \frac{p_1 + n_1}{p + n} I(p_1, n_1)$$

E (Outline) = 2/5 * I(1,1)+3/5*I(2,1)

= 0.951

Hence, $Gain(S_{red}, Outline)$ = I (p, n) – E (Outline)

= 0.971– 0.951 = 0.02

Compute the entropy for Dot : **(no, yes)**

Similarly for different Dot values, $I(p_i, n_i)$ is calculated as given below :

Outline	p_i	n_i	$I(p_i, n_i)$
No	3	0	0
Yes	0	2	0

Calculate entropy using the values from the above table and the formula given below

$$E(A) = \sum_{i=1}^{v} \frac{p_i + n_i}{p + n} I(p_i, n_i)$$

$$E(Dot) = 3/5 * I(3,0) + 2/5 * I(0,2)$$

$$= 0$$

Hence, $Gain(S_{red}, Dot) = I(p, n) - E(Dot)$

$$= 0.971 - 0 = 0.971$$

Dot has the highest gain; therefore, it is below Color = "Red"

Check the tuples with Dot = "yes" from sample S_{red} , it has class triangle

Check the tuples with Dot = "no" from sample S_{red} , it has class square

So the partial tree for red color sample is as given in Fig. P. 3.2.5(c).

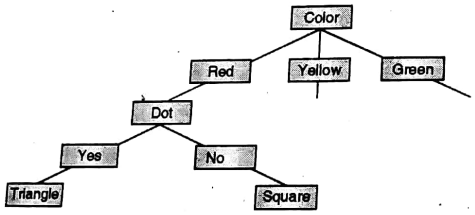


Fig. P. 3.2.5(c)

Step 5 : Consider Color = Yellow and count the number of tuples from the original given training set.

	Attribute			Shape
	Color	Outline	Dot	
1.	Yellow	Dashed	No	Square
2.	Yellow	Solid	Yes	Square
3.	Yellow	Dashed	Yes	Square
4.	Yellow	Solid	No	Square

As all the tuples belong to yellow color have class label square, so directly assign a class label below the node color = "yellow" as square.

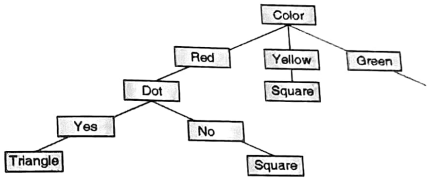


Fig. P. 3.2.5(d)

Step 6 : Consider Color = green and count the number of tuples from the original given training set, as only attribute outline has left, it becomes a node below color ="green".

	Attribute			Shape
	Color	Outline	Dot	
1.	green	Dashed	no	Triangle
2.	green	Dashed	Yes	triangle
3.	green	Solid	No	square
4.	green	dashed	no	triangle
5.	green	Solid	yes	Square

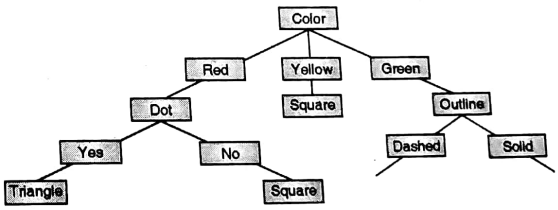


Fig. P. 3.2.5(e)

Check the tuples with Outline = "dashed" from sample S_{green} , it has class triangle

Check the tuples with outline = "solid" from sample S_{green} , it has class square.

Therefore the final Decision Tree is

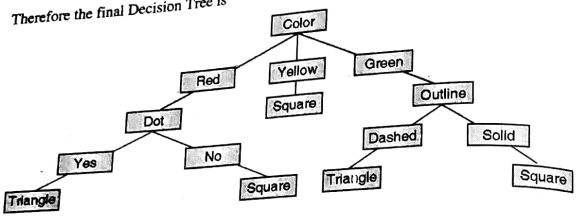


Fig. P. 3.2.5(f)

Ex. 3.2.6 : Apply statistical based algorithm to obtain the actual probabilities of each event to classify the new tuple as a tall. Use the following data

Person ID	Name	Gender	Height	Class
1	Kristina	Female	1.6m	Short
2	Jim	Male	2m	Tall
3	Maggie	Female	1.9m	Medium
4	Martha	Female	1.85m	Medium
5	John	Male	2.8m	Tall
6	Bob	Male	1.7m	Short
7	Clinton	Male	1.8m	Medium
8	Nyssa	Female	1.6m	Short
9	Kathy	Female	1.65m	Short

Soln. :

P(Short) = 4/9 P(Medium) = 3/9 P(Tall) = 2/9

Divide the height attribute into six ranges as given below :

[0,1.6] ,[1.6,1.7],[1.7,1.8],[1.8,1.9],[1.9,2.0],[2.0,infinity]

Gender attribute has only two values Male and Female.

Total Number of short person = 4, Medium = 3, Tall = 2

Prepare the probability table as given below :

Attribute	Value	Count			Probabilities		
		Short	Medium	Tall	Short	Medium	Tall
Gender	Male	1	1	2	1 / 4	1/3	2/2
	Female	3	2	0	3 /4	2/3	0/2
Height	[0,1.6]	2	0	0	2 /4	0	0
	[1.6,1.7]	2	0	0	2/4	0	0
	[1.7,1.8]	0	1	0	0	1/3	0
	[1.8,1.9]	0	2	0	0	2 /3	0
	[1.9,2.0]	0	0	1	0	0	1 /2
	[2.0,infinity]	0	0	1	0	0	1 /2

Use above values to classify new tuple as a tall :

Consider new tuple as t = {Adam, M, 1.95m}

$$P(t \mid Short) = 1/4 * 0 = 0$$

$$P(t \mid Medium) = 1/3 * 0 = 0$$

$$P(t \mid Tall) = 2/2 * 1/2 = 0.5$$

Therefore likelihood of being short = $p(t \mid short) * P(short) = 0 * 4/9 = 0$

Likelihood of being Medium = $0 * 3/9 = 0$

Likelihood of being Tall = $2/9 * 1/2 = 0.11$

Then estimate P(t) by adding individual likelihood values since t will be either short or medium or tall.

$$P(t) = 0 + 0 + 0.11 = 0.11$$

Finally Actually probabilities of each event

$$P(Short \mid t) = (P(t \mid short) * p(short)) / P(t) = (0 * 4/9)/0.11 = 0$$

Similarly

$$P(Medium \mid t) = (0 * 3/9)/0.11 = 0$$

$$P(Tall \mid t) = (0.5 * 2/9)/0.11 = 1$$

New tuple is a Tall as it has the highest probability.

Ex. 3.2.7 : The training data is supposed to be a part of a transportation study regarding mode choice to select Bus, Car or Train among commuters along a major route in a city.

Attributes				Classes
Gender	Car ownership	Travel cost ($)/km	Income level	Transportation mode
Male	0	Cheap	Low	Bus
Male	1	Cheap	Medium	Bus
Female	1	Cheap	Medium	Train

Attributes				Classes
Gender	Car ownership	Travel cost ($)/km	Income level	Transportation mode
Female	0	Cheap	Low	Bus
Male	1	Cheap	Medium	Bus
Male	0	Standard	Medium	Train
Female	1	Standard	Medium	Train
Female	1	Expensive	High	Car
Male	2	Expensive	Medium	Car
Female	2	Expensive	High	Car

Suppose we have new unseen records of a person from the same location where the data sample was taken. The following data are called test data (in contrast to training data) because we would like to examine the classes of these data.

Person name	Gender	Car ownership	Travel cost ($)/km	Income level	Transportation mode
Alex	Male	1	Standard	High	?
Buddy	Male	0	Cheap	Medium	?
Cherry	Female	1	Cheap	High	?

The question is what transportation mode would Alex, Buddy and Cheery use?

Soln. :

Class P: Transportation mode= "Bus"

Class Q: Transportation mode= "Train"

Class N: Transportation mode= "Car"

Total no. of records: 10

No. of records with "Bus" class = 4

No. if records with "Train" class = 3

No. of records with "Car" class = 3

So,

$$\text{Information Gain} = I(p, q, n) = -(p/(p+q+n))\log_2(p/(p+q+n)) - (q/(p+q+n))\log_2(q/(p+q+n)) - (n/(p+q+n))\log_2(n/(p+q+n))$$

$$I(p, q, n) = I(4,3,3) = -(0.4)(-1.322) - (0.3)(-1.737) - (0.3)(-1.737)$$

$$I(4,3,3) = 0.5288 + 0.5211 + 0.5211$$

$$\mathbf{I(4,3,3) = 1.571}$$

Step 1 : Compute the entropy of gender : (Male, Female)

For gender = Male $p_i = 3$

$q_i = 1$ $n_i = 1$

Therefore, $I(p_i, q_i, n_i) = I(3,1,1)$

$= -(3/5)\log_2(3/5) - (1/5)\log_2(1/5) - (1/5)\log_2(1/5)$

$= 1.371$

Similarly for different gender $I(p_i, q_i, n_i)$ is calculated as given below :

Gender	p_i	q_i	n_i	$I(p_i, n_i)$
Male	3	1	1	1.371
Female	1	2	2	1.522

Calculate Entropy using the values from the above table and the formula given below

$$E(A) = \sum_{i=1}^{v} \frac{p_i + n_i}{p + n} I(p_i, n_i)$$

$$E(\text{gender}) = (5/10) * I(3,1,1) + (5/10) * I(1,2,2)$$

$$= 1.447$$

Note : S is the total training set.

Hence

$$\text{Gain(S, gender)} = I(p, q, n) - E(\text{gender}) = 1.571 - 1.447 = 0.124$$

Similarly,

Gain (S, Car Ownership) = 0.535

Gain (S, Travel Cost ($)/Km) = 1.21

Gain (S, Income Level) = 0.696

Travel cost ($)/Km attribute has the highest gain, therefore it is used as the decision attribute in the root node.

Since travel cost ($)/Km has three possible values, the root node has three branches (Cheap, Standard, Expensive).

Since for all the attributes of Travel Cost ($)/Km = expensive, Transportation mode= "Car" , so assign class 'Car' to expensive.

Since for all the attributes of Travel Cost ($)/Km = Standard, Transportation mode= "Train", so assign class 'Train' to standard.

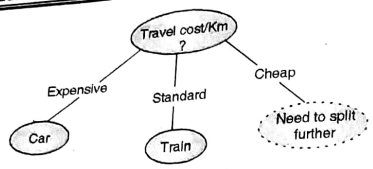


Fig. P. 3.2.7(a)

Consider travel cost ($)/Km = Cheap and count the number of tuples from the original given training set

S_{cheap} = 5

Attributes				Classes
Gender	**Car ownership**	**Travel cost ($)/km**	**Income level**	**Transportation mode**
Male	0	Cheap	Low	Bus
Male	1	Cheap	Medium	Bus
Female	1	Cheap	Medium	Train
Female	0	Cheap	Low	Bus
Male	1	Cheap	Medium	Bus

Note : *Refer above table :*

Total No. of Bus tuple = 4 and total no of Train tuple = 1,

and total no of Car tuple = 0

$$I(p, q, n) = I(4, 1, 0)$$
$$= -(4/5)\log_2(4/5) - (1/5)\log_2(1/5) - (0/5)\log_2(0/5)$$
$$= 0.722$$

(i) Compute the entropy for gender : (Male, female)

For gender = Male,

p_i = with "Bus" class = 3, q_i = with "Train" class = 0 and n_i with "car" class= 0

Therefore ,

$$I(p_i, q_i, n_i) = I(3,0,0) = -(3/3)\log_2(3/3) - (0/3)\log_2(0/3) - (0/3)\log_2(0/3)$$
$$= 0.$$

Similarly for different genders $I(p_i, q_i)$ is calculated as given below :

Gender	p_i	q_i	n_i	$I(p_i, q_i, n_i)$
Male	3	0	0	0
Female	1	1	0	1

Calculate entropy using the values from the above table and the formula given below

$$E(A) = \sum_{i=1}^{v} \frac{p_i + n_i}{p + n} I(p_i, n_i)$$

$$E(\text{gender}) = 3/5 * I(3, 0, 0) + 2/5 * I(1, 1, 0) = 0.4$$

Note : S_{cheap} is the total training set.

Hence

$$\text{Gain}(S_{cheap}, \text{gender}) = I(p, q, n) - E(\text{gender})$$
$$= 0.722 - 0.4 = 0.322$$

(ii) Compute the entropy for Car ownership: (0, 1, 2)

For Car ownership = 0,

p_i = with "bus" class = 2 ,q_i = with "train" class = 0 and n_i with "car" class = 0

Therefore,

$I(p_i, q_i, n_i) = I(2,0,0) = -(2/2)\log_2(2/2) - (0/2)\log_2(0/2) - (0/2)\log_2(0/2) = 0.$

Similarly for different outlook ranges $I(p_i, q_i, n_i)$ is calculated as given below :

Car ownership	p_i	q_i	n_i	$I(p_i, q_i, n_i)$
0	2	0	0	0
1	2	1	0	0.918
2	0	0	0	0

Calculate Entropy using the values from the above table and the formula given below

$$E(A) = \sum_{i=1}^{v} \frac{p_i + n_i}{p + n} I(p_i, n_i)$$

$$E(\text{Car ownership}) = 2/5 * I(2, 0, 0) + 3/5 * I(2, 1, 0) + 0/5 * I(0, 0, 0)$$
$$= 0.551$$

Note : S_{cheap} is the total training set.

Hence

$$\text{Gain}(S_{cheap}, \text{car ownership}) = I(p, q, n) - E(\text{car ownership})$$
$$= 0.722 - 0.551 = 0.171$$

(iii) Compute the entropy for income level : (Low, medium, high)

For income level = Low,

p_i = with "Bus" class = 2 ,q_i = with "Train" class = 0 and n_i with "car" class = 0

Therefore,

$I(p_i, q_i, n_i)$ = $I(2,0,0) = -(2/2)\log_2(2/2) - (0/2)\log_2(0/2) - (0/2)\log_2(0/2)$

= 0.

Similarly for different outlook ranges $I(p_i, q_i, n_i)$ is calculated as given below :

Income level	p_i	q_i	n_i	$I(p_i, q_i, n_i)$
Low	2	0	0	0
Medium	2	1	0	0.918
High	0	0	0	0

Calculate Entropy using the values from the above table and the formula given below

$$E(A) = \sum_{i=1}^{v} \frac{p_i + n_i}{p+n} I(p_i, n_i)$$

E (Income Level) = $2/5 * I(2, 0, 0) + 3/5 * I(2, 1, 0) + 0/5 * I(0,0,0)$

= 0.551

Note : S_{cheap} is the total training set.

Hence

$Gain(S_{cheap}, \text{Income level})$ = I (p, q, n) – E (Income level)

= 0.722 – 0.551 = 0.171

Therefore, since gender has the highest gain, it comes below cheap.

For all gender = Male, Transportation mode= bus

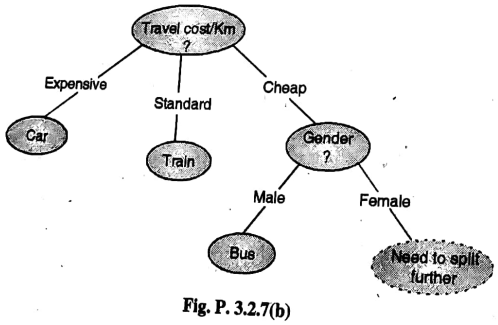


Fig. P. 3.2.7(b)

S_{female} = 2

Gender	Car ownership	Income level	Transportation mode
Female	1	Medium	Train
Female	0	Low	Bus

Suppose we select attribute car ownership, we can update our decision tree into the final version.

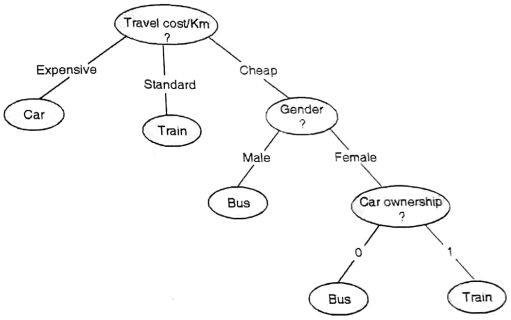


Fig. P. 3.2.7(c)

Ex. 3.2.8 : The table below shows a sample dataset of whether a customer responds to a survey or not. **"Outcome"** is the class label. Construct a Decision Tree Classifier for the dataset. For a new example **(Rural, semidetached, low, No)**, what will be the predicted class label ?

MU - May 2016, Dec. 2016, 10 Marks

District	House type	Income	Previous customer	Outcome
Suburban	Detached	High	No	Nothing
Suburban	Detached	High	Yes	Nothing
Rural	Detached	High	No	Responded
Urban	Semi-detached	High	No	Responded
Urban	Semi-detached	Low	No	Responded
Urban	Semi-detached	Low	Yes	Nothing
Rural	Semi-detached	Low	Yes	Responded
Suburban	Terrace	High	No	Nothing
Suburban	Semi-detached	Low	No	Responded

District	House type	Income	Previous customer	Outcome
Urban	Terrace	Low	No	Responded
Suburban	Terrace	Low	Yes	Responded
Rural	Terrace	High	Yes	Responded
Rural	Detached	Low	No	Responded
Urban	Terrace	High	Yes	Nothing

Soln. :

Sr.No.	District	House_Type	Income	Previous_Customer	Outcome
1	Suburban	Detached	High	No	Nothing
2	Suburban	Detached	High	Yes	Nothing
3	Rural	Detached	High	No	Responded
4	Urban	Terrace	High	No	Responded
5	Urban	Semi-detached	Low	No	Responded
6	Urban	Semi-detached	Low	Yes	Nothing
7	Rural	Semi-detached	Low	Yes	Responded
8	Suburban	Terrace	High	No	Nothing
9	Suburban	Semi-detached	Low	No	Responded
10	Urban	Terrace	Low	No	Responded
11	Suburban	Terrace	Low	Yes	Responded
12	Rural	Terrace	High	Yes	Responded
13	Rural	Detached	Low	No	Responded
14	Urban	Terrace	High	Yes	Nothing

Class P : Outcome = "Responded"

Class N : Outcome = "Nothing"

Total number of records 14.

Count the number of records with "Responded" class and "Nothing" class.

So number of records with "Responded" class = 9 and "Nothing" class = 5

So Information gain $= I(p, n) = -\frac{p}{p+n}\log_2\frac{p}{p+n} - \frac{n}{p+n}\log_2\frac{n}{p+n}$

$I(p, n) = I(9, 5) = -(9/14)\log_2(9/14) - (5/14)\log_2(5/14)$

$= (-0.643) * (-0.637) + (-0.357) * (-1.485)$

$I(p, n) = 0.409 + 0.530 = 0.940$

Step 1 : Compute the entropy for District: (Suburban, Rural , Urban)

For District = Suburban,

Pi = with "Responded" class = 2 and ni = with "Nothing" class = 3

Therefore, I(pi , ni) = $I(2,3) = -(2/5)\log_2(2/5) - (3/5)\log_2(3/5) = 0.971$.

Similarly for different District ranges I(pi , ni) is calculated as given below :

District	pi	ni	I(pi, ni)
Suburban	2	3	0.971
Rural	4	0	0
Urban	3	2	0.971

Calculate entropy using the values from the above table and the formula given below :

$$E(A) = \sum_{i=1}^{v} \frac{pi + ni}{p + n} I(pi, ni)$$

$$E(\text{District}) = \frac{5}{14} I(2, 3) + \frac{4}{14} I(4, 0) + \frac{5}{14} I(3, 2) = 0.694$$

T is the total training set.

Hence Gain(T, District) = I (p, n) – E (District) = 0.940 – 0.694 = 0.246

Similarly, Gain (T, House_Type) = 0.029

Gain (T, Income) = 0.151

Gain (T, Previous_Customer) = 0.048

District shows the highest gain, so it is used as the decision attribute in the root.

As District has only values "Suburban, Rural, Urban", the root node has three branches

Step 2 :

As attribute District at root, we have to decide on the remaining three attribute for Suburban branch

Consider District = Suburban and count the number of tuples from the original given training set

SSuburban = {1, 2, 8, 9, 11} = 5

Sr.No.	District	House_Type	Income	Previous_Customer	Outcome
1	Suburban	Detached	High	No	Nothing
2	Suburban	Detached	High	Yes	Nothing
8	Suburban	Terrace	High	No	Nothing
9	Suburban	Semi-detached	Low	No	Responded

Sr.No.	District	House_Type	Income	Previous_Customer	Outcome
11	Suburban	Terrace	Low	Yes	Responded

Total number of Responded tuple = 2 and total number of Nothing tuple= 3

$$I(p, n)=I(2, 3) = -(2/5)\log2(2/5) - (3/5)\log2(3/5) = 0.971$$

(i) Compute the entropy for House_Type: (Detached, Terrace, Semi-detached)

For House_Type = Detached,

pi = with "Responded" class = 0 and ni = with "Nothing" class = 2

Therefore , $I(pi , ni) = I(0,2) = -(0/2)\log2(0/2) - (2/2)\log2(2/2) = 0$

Similarly for different District ranges I(pi , ni) is calculated as given below :

House_Type	pi	ni	I(pi, ni)
Detached	0	2	0
Terrace	1	1	1
Semi-detached	1	0	0

Calculate Entropy using the values from the above table and the formula given below :

$$E(A) = \sum_{i=1}^{v} \frac{pi + ni}{p + n} I(pi, ni)$$

E (House_Type) = 2/5 * I (0, 2) + 2/5* I(1, 1) + 1/5 *I(1, 0) = 0.4

Note : TSuburban is the total training set.

Hence,

Gain(TSuburban, House_Type) = I (p, n) – E (House_Type)

= 0.971 – 0.4 = 0.571

(ii) Compute the entropy for Income : (High, Low)

For Income = High,

pi = with "Responded" class = 0 and ni = with "Nothing" class = 3

Therefore , $I(pi , ni) = I(0,3) = -(0/3)\log2(0/3) - (3/3)\log2(3/3) = 0$

Similarly for different District ranges I(pi , ni) is calculated as given below :

Income	pi	ni	I(pi, ni)
High	0	3	0
Low	2	0	0

Calculate Entropy using the values from the above table and the formula given below

$$E(A) = \sum_{i=1}^{v} \frac{pi + ni}{p + n} I(pi, ni)$$

E (Income) = 3/5 * I (0, 3) + 2/5* I(2, 0) = 0

Note : TSuburban is the total training set.

Hence,

Gain(TSuburban, Income) = I (p, n) – E (Income)

= 0.971 – 0 = 0.971

(iii) Compute the entropy for Previous_Customer : (No, Yes)

For Previous_Customer = No,

pi = with "Responded" class = 1 and ni = with "Nothing" class = 2

Therefore, $I(pi , ni) = I(1,2) = -(1/3)\log2(1/3) - (2/3)\log2(2/3) = 0.918$

Similarly for different District ranges I(pi , ni) is calculated as given below :

Previous_Customer	pi	ni	I(pi, ni)
No	1	2	0.918
Yes	1	1	1

Calculate Entropy using the values from the above table and the formula given as:

$$E(A) = \sum_{i=1}^{v} \frac{pi + ni}{p + n} I(p1, n1)$$

E (Previous_Customer) = 3/5 * I (1, 2) + 2/5* I (1, 1) = 0.951

Note : TSuburban is the total training set.

HenceGain (TSuburban, Previous_Customer)

= I (p, n) – E (Previous_Customer)

= 0.971 – 0.951 = 0.02

Therefore,

Gain (TSuburban, Income) = 0.970

Gain (TSuburban, House_Type) = 0.570

Gain (TSuburban, Previous_Customer) = 0.02

Income has the highest gain; therefore, it is below District = "Suburban".

Step 3 :

Consider only House_Type and Previous_Customer for District = Rural and count the number of tuples from the original given tUrbaning set

TRural = {3,7,12,13} = 4

Sr.No.	District	House_Type	Income	Previous_Customer	Outcome
3	Rural	Detached	High	No	Responded
7	Rural	Semi-detached	Low	Yes	Responded
12	Rural	Terrace	High	Yes	Responded
13	Rural	Detached	Low	No	Responded

Since for the attributes House_Type and Previous_Customer, Outcome = Responded, so assign class 'Responded' to Rural.

Step 4 :

Consider House_Type and Previous_Customer for District = Urban and count the number of tuplesfrom the original given training set

TUrban = {4, 5, 6, 10, 14}

= 5

Sr.No.	District	House_Type	Income	Previous_Customer	Outcome
4	Urban	Terrace	High	No	Responded
5	Urban	Semi-detached	Low	No	Responded
6	Urban	Semi-detached	Low	Yes	Nothing
10	Urban	Terrace	Low	No	Responded
14	Urban	Terrace	High	Yes	Nothing

Consider the above table as the new training set and calculate the Gain for House_Type and Previous_Customer.

Class P : Outcome = "Responded"

Class N : Outcome = "Nothing"

Total number of records 5

Count the number of records with "Responded" class and "Nothing" class:

So number of records with "Responded" class = 3 and "Nothing" class = 2

$$\text{So Information gain} = I(p, n) = -\frac{p}{p+n}\log_2\frac{p}{p+n} - \frac{n}{p+n}\log_2\frac{n}{p+n}$$

$$I(p, n) = I(3, 2) = -(3/5)\log_2(3/5) - (2/5)\log_2(2/5) = 0.970$$

(iv) Compute the entropy for Previous_Customer :

For Previous_Customer = No

pi = with "Responded" class = 3 and ni = with "Nothing" class = 0

Therefore, I(pi , ni) = I(3,0) = 0.

For Previous_Customer = Yes

pi = with "Responded" class = 0 and ni = with "Nothing" class = 2

Therefore, I(pi, ni) = I(0, 2) = 0

Similarly for different District ranges I(pi , ni) is calculated as given below :

Previous_Customer	pi	ni	I(pi, ni)
No	3	0	0
Yes	0	2	0

Calculate entropy using the values from the above table and the formula given below :

$$E(A) = \sum_{i=1}^{v} \frac{pi + ni}{p + n} I(pi, ni)$$

$$E(\text{Previous\_Customer}) = \frac{3}{5} I(3, 0) + \frac{2}{5} I(0, 2) = 0$$

Hence

Gain(TUrban, Previous_Customer) = I (p, n) – E (Previous_Customer)

= 0.970 – 0 = 0.970

(v) Compute the entropy for House_Type : (Detached, Terrace , Semi-detached)

For House_Type = Detached,

pi = with "Responded" class = 0 and ni = with "Nothing" class = 0

Therefore, I(pi , ni) = I(0,0) = 0

Similarly for different District ranges I(pi , ni) is calculated as given below :

House_Type	pi	ni	I(pi, ni)
Detached	0	0	0
Terrace	2	1	0.918
Semi-detached	1	1	1

Calculate Entropy using the values from the above table and the formula given below :

$$E(A) = \sum_{i=1}^{v} \frac{pi + ni}{p + n} I(pi, ni)$$

E (House_Type) = 0/5 * I (0, 0) + 3/5* I(2, 1) + 2/5 *I(1, 1)

= 0.951

Note : TUrban is the total training set.

Hence

Gain(TUrban, House_Type) = I (p, n) – E (House_Type)

= 0.970 – 0.951 = 0.019

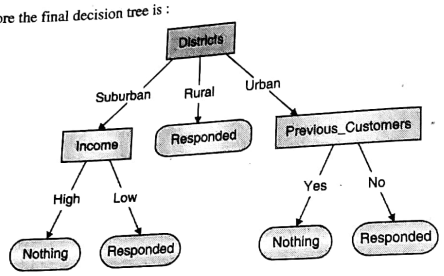
Therefore,

Gain(TUrban, House_Type) = 0.019

Gain(TUrban, Previous_Customer) = 0.970

Previous_Customer has the highest gain; therefore, it is below District = "Urban".

Therefore the final decision tree is :



The decision tree can also be expressed in rule format :

IF District = Suburban AND Income = high THEN Outcome = Nothing

IF District = Suburban AND Income = Low THEN Outcome = Responded

IF District = Rural THEN Outcome = Responded

IF District = Urban AND Previous_Customer = Yes THEN Outcome = Nothing

IF District = Urban AND Previous_Customer = No THEN Outcome = Responded

Syllabus Topic : Bayesian Classification - Naïve Bayes Classifier

3.3 Bayesian Classification : Naïve Bayes Classifier

3.3.1 Bayes' Theorem

- It is also known as Bayes' Rule.
- Bayes' 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 has 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.

3.3.2 Basics of Bayesian 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 combined, 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

$$P(h \mid D) = \frac{P(D \mid h)\, P(h)}{P(D)}$$

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.

3.3.3 Naive Bayes Classifier : Examples

Ex. 3.3.1 : 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

Soln. :

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

- The classification problem may be formalized using a-posteriori probabilities :
  P(C|Y) is the probability that the sample tuple $Y=<y_1,\ldots,y_k>$ 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(overcast|Yes) = 4/9	P(overcast|No) = 0	P(Yes) = 9/14
P(rain|Yes) = 3/9	P(rain|No) = 2/5	P(No) = 5/14
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	

- An unseen sample Y = <rain, hot, high, false>

$P(Y|Yes) \cdot P(Yes) = P(rain|Yes) \cdot P(hot|Yes) \cdot P(high|Yes) \cdot P(false|Yes) \cdot P(Yes)$

$= 3/9 \cdot 2/9 \cdot 3/9 \cdot 6/9 \cdot 9/14 = 0.010582$

$P(Y|No) \cdot P(No) = P(rain|No) \cdot P(hot|No) \cdot P(high|No) \cdot P(false|No) \cdot P(No)$

$= 2/5 \cdot 2/5 \cdot 4/5 \cdot 2/5 \cdot 5/14 = 0.018286$

- Choose the class so that it **maximizes** this probability. This means that the new instance 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 >

$P(Y|Yes) \cdot P(Yes) = P(sunny|Yes) \cdot P(cool|Yes) \cdot P(high|Yes) \cdot P(true|Yes) \cdot P(Yes)$

$= 2/9 \cdot 3/9 \cdot 3/9 \cdot 3/9 \cdot 9/14 = 0.0053$

$P(Y|No) \cdot P(No) = P(sunny|No) \cdot P(cool|No) \cdot P(high|No) \cdot P(true|No) \cdot P(No)$

$= 3/5 \cdot 1/5 \cdot 4/5 \cdot 3/5 \cdot 5/14 = 0.0206$

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. 3.3.2 : 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

Soln. :

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(DomesticlYes) =2/5	P(DomesticlNo) =3/5
P(ImportedlYes) =3/5	P(ImportedlNo) = 2/5

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

P(XlYes) · P(Yes) = P(Redl Yes) · P(Domesticl Yes) · P(Suvl Yes) · P(Yes)
= 3/5 × 2/5 × 1/5 × 5/10 = 0.024

P(XlNo) · P(No) = P(Redl No) · P(Domesticl No) · P(Suvl 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. 3.3.3 : 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 result value as sunburned or none ?

Soln. :

Hair	
P(BlondelSunburned) = 2/3	P(Blondel None) = 1/5
P(Brownl Sunburned) = 0	P(Brownl None) = 4/5
P(Redl Sunburned) = 1/3	P(Redl None) = 0
Height	
P(Averagel Sunburned) = 2/3	P(Averagel None) = 0
P(Talll Sunburned) = 0	P(Talll None) = 2/5
P(Shortl Sunburned) = 1/3	P(Shortl None) = 2/5
Weight	
P(Lightl Sunburned) =1/3	P(Lightl None) = 1/5
P(Averagel Sunburned) =1/3	P(Averagel None) = 2/5
P(Heavyl Sunburned) =1/3	P(Heavyl None) = 2/5

Dublin	
P(Nol Sunburned) = 3/3	P(Nol None) = 2/5
P(Yesl Sunburned) = 0	P(Yesl None) = 3/5

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

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

P(XlSunburned) · P(Sunburned) = P(Brownl Sunburned) · P(talll Sunburned) · P(averagel Sunburned)· P(Nol Sunburned) ·P(sunburned) = 0

P(XlNone) · P(None) = P(Brownl None) · P(talll None) · P(averagel None) · P(Nol None) ·P(None) = 0.032

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

Ex. 3.3.4 : Predict a class label of an unknown sample using Naive Bayesian classification on the following training dataset 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

Soln. :

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(Medium| No) = 2/5
P(low|yes) = 3/9	P(low| 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 Rating="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 Rating="fair"| No) ·P(No) = 0.007

- Since 0.028 > 0.007, Therefore the naive Bayesian classifier predicts Buys computer= "Yes" for sample X.

Ex. 3.3.5 : Using Naive Bayesian classification on the following given training set, classify the unseen tuple (Refund = No, Married, Income = 120 K)

rid	Refund	Marital status	taxable	Evade
1	Yes	Single	125 K	No
2	No	Married	100 K	No
3	No	Single	120 K	No
4	Yes	Married	70 K	No
5	No	Divorced	95 K	Yes
6	No	Married	60 K	No
7	Yes	Divorced	220 K	No
8	No	Single	85 K	Yes
9	No	Married	75 K	No
10	No	Single	90 K	Yes

P(No) = 7/10

P(Yes) = 3/10

Soln. :

Given a test Record

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

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. 3.3.6 : Consider the training set for the class of mammals and non mammals, using Naive 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

Soln. :

Unseen record is given as :

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

A: attributes M: mammals N: non-mammals

$$P(A \mid M) = 5/7 \times \frac{6}{7} \times \frac{2}{7} \times \frac{2}{7} = 0.05$$

$$P(A \mid N) = 2/13 \times \frac{10}{13} \times \frac{3}{13} \times \frac{4}{13} = 0.0084$$

$$P(A \mid M)\,P(M) = 0.05 \times \frac{7}{20} = 0.0175$$

$$P(A \mid N)\,P(N) = 0.008 \times \frac{13}{20} = 0.0052$$

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

Unseen record belongs to class mammals.

Ex. 3.3.7 : Consider the following dataset that helps to predict the RISK of a loan application based on the applicant's CREDIT HISTORY, DEBT and INCOME.

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

Soln. :

- Predict the risk for unseen Tuple X = <unknown, high, over35, 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=15to35>. Which class will be returned by Naive Bayes?

Ex. 3.3.8 : Using given table, create classification model using any algorithm and hence classify following tuple <income = medium, credit = good>.

Transaction Id	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

Transaction Id	Income	Credit	Decision
7	High	Bad	REQUEST ID
8	Medium	Bad	REQUEST ID
9	High	Bad	REJECT
10	Low	Bad	CALL POLICE

Soln. :

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) = ½	P(High| REJECT) = 1/1	P(High| CALL POLICE) = 0
P(Medium| AUTHORIZE) = 2/6	P(Medium| REQUEST ID) = ½	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

$$P(\text{AUTHORIZE}) = 6/10$$
$$P(\text{REQUEST ID}) = 2/10$$
$$P(\text{REJECT}) = 1/10$$
$$P(\text{CALL POLICE}) = 1/10$$
$$P(X|\text{ AUTHORIZE}) \times P(\text{AUTHORIZE}) = 2/6 \times 3/6 \times 6/10 = 0.1$$
$$P(X|\text{ REQUEST ID}) \times P(\text{REQUEST ID}) = 0$$
$$P(X|\text{ REJECT}) \times P(\text{REJECT}) = 0$$
$$P(X|\text{ CALL POLICE}) \times P(\text{CALL POLICE}) = 0$$

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

Ex. 3.3.9 : Illustrate any one classification technique for the above data set. Show how we can classify a new tuple, with (Homeowner = Yes; Status = Employed; Income = Average).

MU - May 2015, 10 Marks

Id	Homeowner	Status	Income	Defaulted
1	Yes	Employed	High	No
2	No	Business	Average	No
3	No	Employed	Low	No
4	Yes	Business	High	No
5	No	Unemployed	Average	No
6	No	Business	Low	No
7	Yes	Unemployed	High	No
8	No	Employed	Average	No
9	No	Business	Low	No
10	No	Employed	Average	No

Ans. :

The unknown sample is

x = { Homeowner = Yes; Status = Employed; Income = Average}

Homeowner	
P(Yes|Yes) = 0/3	P(Yes|No) = 3/7
P(No|Yes) = 3/3	P(No|No) = 4/7
Status	
P(Employed | yes) = 2/3	P(Employed | No) = 2/7
P(Unemployed | yes) = 1/3	P(Unemployed | No) = 1/7
P(Business | yes) = 0/3	P(Business | No) = 4/7
Income	
P(High | yes) = 0/3	P(High | No) = 3/7
P(Average | yes) = 3/3	P(Average | No) = 1/7
P(Low | yes) = 0/3	P(Low | No) = 3/7

P(Yes) = 3/10
P(No) = 7/10

An unseen sample x = { Homeowner = Yes; Status = Employed; Income = Average}

$P(X|Yes) \cdot P(Yes) = P(\text{Homeowner} = Yes|Yes) \times P(\text{Status} = \text{Employed}|Yes) \times P(\text{Income} = \text{Average} | Yes) \times P(Yes)$

$= 0/3 \times 2/3 \times 3/3 \times 3/10 = 0$

$P(X|No) \cdot P(No) = P(\text{Homeowner} = Yes|No) \times P(\text{Status} = \text{Employed}|No) \times P(\text{Income} = \text{Average} | No) \times P(No)$

$= 3/7 \times 2/7 \times 1/7 \times 7/10 = 0.012$

Since 0.012> 0, Therefore the naive Bayesian classifier predicts new tuple X is classified as Defaulted = "No".

3.3.4 Rule based Classification

- Rule-based classification is featured by building rules based on object attributes. Rule-based classification is a powerful tool for feature extraction, often performing better than supervised classification for many feature types.
- Learned model is represented as a set of IF-THEN rules.
- IF-THEN rule is expressed in the form.
- IF condition THEN conclusion.
- The LHS or "IF" part of the rule is called as "rule antecedent" or "precondition".
- The RHS or "THEN" part is called as "rule consequent".
- Example : IF age = young AND salary = high THEN loan = yes
- This can also be written as :

(age = young) ^ (salary = high) => loan = yes

Rule R can be accessed by its coverage and accuracy

$$\text{coverage}(R) = n_{covers} / |D|$$

$$\text{accuracy}(R) = n_{correct} / n_{covers}$$

where n_{covers} = number of tuples covered by R

$n_{correct}$ = number of tuples correctly classified by R

|D| = number of tuples in data set D

- If more than one rule is triggered then it need **conflict resolution.**
- Based on size, it has to order. So give highest priority to that triggering rule which has the maximum attribute test.
- Make the decision list based on the ordering of the rules. Rules are organized based on some measure of rule quality or by taking expert opinion.

Extract the rules from decision tree

- Once the decision tree is created, list the rules which are easy to understand than big and complex tree.
- For every path of the tree, create a rule from root node to a leaf node.
- The last node or leaf node gives the class label.

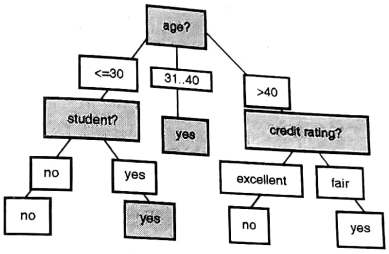


Fig. 3.3.1 : Decision Tree for "Buys_Computer"

Rule extraction from above buys_computer decision-tree

1. IF age = "<=30" AND student = "no" THEN buys_computer = "no"
2. IF age = "<=30" AND student = "yes" THEN buys_computer = "yes"
3. IF age = "31…40" THEN buys_computer = "yes"
4. IF age = ">40" AND credit_rating = "excellent" THEN buys_computer = "no"
5. IF age = ">40" AND credit_rating = "fair" THEN buys_computer = "yes"

3.3.5 Other Classification Methods

- k-nearest neighbor classifier
- Case-based reasoning
- Genetic algorithm
- Rough set approach
- Fuzzy set approaches

Syllabus Topic : Prediction

3.4 Prediction

- 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-valued 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.

Syllabus Topic : Structure of Regression Model

3.4.1 Structure of Regression Model

MU - Dec. 2016

- 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 regressions

- The general form of regression is :

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

Multiple regression : $Y=m+n_1X_1+n_2X_2+n_3X_3+\ldots+n_tX_t+u$

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.

3.4.2 Linear Regression

MU - Dec. 2016

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

Syllabus Topic : Simple linear regression

Simple linear regression

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

$$Y = \alpha + \beta X$$

- 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 the Fig. 3.4.1.

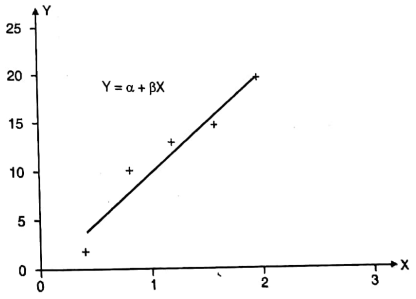


Fig. 3.4.1 : Linear regression

Syllabus Topic : Multiple Linear Regression

MU - Dec. 2016

3.4.3 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

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

Where, Y is the dependent variable or response variable

$X_1, X_2 .....X_k$ are the independent variables or predictors

e is random error.

$a_0, a_1, a_2.....a_k$ are the regression coefficients

3.4.4 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 variables.
- Log linear models are models that postulate a linear relationship between the independent variables and the logarithm of the dependent variable, for example :

$$\log(y) = a_0 + a_1 x_1 + a_2 x_2 ... + a_N x_N$$

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 (additive) one.

3.5 Model Evaluation and Selection

MU - May 2015

- 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 are based on randomly sampled partitions of data :
  - Holdout method
  - Random sub-sampling
  - 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 Characteristic (ROC) Curves

Syllabus Topic : Accuracy and Error Measures, Precision, Recall

3.5.1 Accuracy and Error Measures

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

Basic concepts

1. **Partition the data randomly into three sets :** Training set, 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 below in the form of a confusion matrix :

		Predicted class		
	Class Label	**C_1**	**C_2**	**Total**
Actual class	C_1	True Positives (TP)	False Negatives (FN)	**P**
	C_2	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 members which are classified as class non-members.
- **P :** 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 tuple 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

$$Sensitivity = TP/P$$

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

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

Accuracy is also a function of sensitivity and specificity :

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

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

$$Error\ rate = 1 - accuracy, \text{ or } Error\ rate = (FP + FN)/All$$

Or

$$Error\ rate = \frac{FP + FN}{P + N}$$

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

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

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

$$Recall = \frac{|TP|}{|TP| + |FN|}$$

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

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

12. **F_β :** Weighted measure of precision and recall and assigns ß times as much weight to recall as to precision

$$F_\beta = \frac{(1 + \beta^2) \times Precision \times Recall}{\beta^2 \times Precision + Recall}$$ 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.

Syllabus Topic : Holdout

3.5.2 Holdout

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

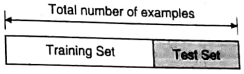


Fig. 3.5.1

- To train the classifier, training data set is used and once the classifier is constructed the use test 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 example, 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.

Syllabus Topic : Random Sampling

3.5.3 Random Sub-sampling

- It is a variation of the holdout method.
- The holdout method is repeated k times.
- Each split randomly selects a fixed number example without replacement.
- For each data split we retrain the classifier from scratch with the training examples and estimate E_i with the test examples.

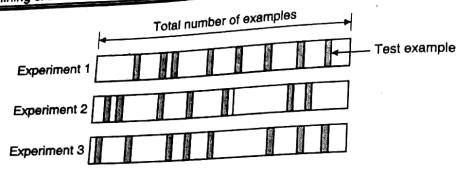


Fig. 3.5.2

- The overall accuracy is calculated by taking the average of the accuracies obtained from each iteration.

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

Syllabus Topic : Cross-Validation (CV)

3.5.4 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).

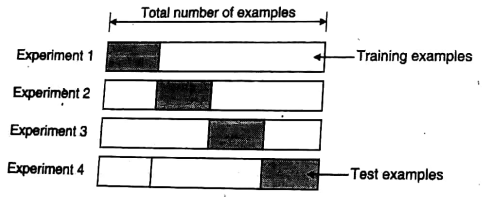


Fig. 3.5.3

  - **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.

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

- **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 example for testing.
- The average error rate on test examples gives the true error.

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

- **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 reduced due to stratification.
  - Ten-fold cross-validation is repeated ten times and finally the results are averaged based on the previous 10 results.

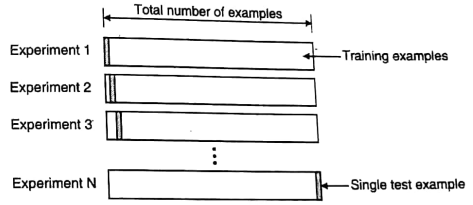


Fig. 3.5.4

3.5.5 Bootstrapping

- CV uses sampling of data set without replacement. Once the tuple or instance 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 dataset from the original dataset which is not the part of training dataset.
- Bootstrapping is the best error estimator for small datasets.

3.6 University Questions and Answers

May 2015

Q. 1

Id	Homeowner	Status	Income	Defaulted
1	Yes	Employed	High	No
2	No	Business	Average	No
3	No	Employed	Low	No
4	Yes	Business	High	No
5	No	Unemployed	Average	No
6	No	Business	Low	No
7	Yes	Unemployed	High	No
8	No	Employed	Average	No
9	No	Business	Low	No
10	No	Employed	Average	No

Illustrate any one classification technique for the above data set. Show how we can classify a new tuple, with (Homeowner = Yes; Status = Employed; Income = Average).
(Ans. : Refer Ex. 3.3.9) **(10 Marks)**

Q. 2 Explain different methods that can be used to evaluate and compare the accuracy of different classification algorithms. *(Ans. : Refer section 3.5)* **(10 Marks)**

Dec. 2015

Q. 3 Define classification, issues of classification and explain ID3 classification with example.
(Ans. : Refer Sections 3.1.1, 3.1.3, 3.1.5 and 3.2.3) **(10 Marks)**

May 2016

Q. 4 The table below shows a sample dataset of whether a customer responds to a survey or not. **"Outcome"** is the class label. Construct a Decision Tree Classifier for the dataset. For a new example **(Rural, semidetached, low, No)**, what will be the predicted class label ? **(10 Marks)**

District	House type	Income	Previous customer	Outcome
Suburban	Detached	High	No	Nothing
Suburban	Detached	High	Yes	Nothing
Rural	Detached	High	No	Responded
Urban	Semi-detached	High	No	Responded
Urban	Semi-detached	Low	No	Responded
Urban	Semi-detached	Low	Yes	Nothing
Rural	Semi-detached	Low	Yes	Responded
Suburban	Terrace	High	No	Nothing

District	House type	Income	Previous customer	Outcome
Suburban	Semi-detached	Low	No	Responded
Urban	Terrace	Low	No	Responded
Suburban	Terrace	Low	Yes	Responded
Rural	Terrace	High	Yes	Responded
Rural	Detached	Low	No	Responded
Urban	Terrace	High	Yes	Nothing

(Ans. : Refer Ex. 3.2.8)

Dec. 2016

Q. 5 The table below shows a sample dataset of whether a customer responds to a survey or not. "Outcome" is the class table.
Construct a Navie Bayes 'Classifier for the dataset. For a new example (Rural, semidetached, low, No),What will be the predicted class label ? *(Ans. : Refer Ex. 3.2.8)* **(10 Marks)**

District	House Type	Income	Previous Customer	Outcome
Suburban	Detached	High	No	Nothing
Suburban	Detached	High	Yes	Nothing
Rural	Detached	High	No	Responded
Urban	Semi- Detached	High	No	Responded
Urban	Semi- Detached	Low	No	Responded
Urban	Semi- Detached	Low	Yes	Nothing
Rural	Semi- Detached	Low	Yes	Responded
Suburban	Terrace	High	No	Nothing
Suburban	Semi- Detached	Low	No	Responded
Urban	Terrace	Low	No	Responded
Suburban	Terrace	Low	Yes	Responded
Rural	Terrace	High	Yes	Responded
Rural	Detached	Low	No	Responded
Urban	Terrace	High	Yes	Nothing

Q. 6 Briefly explain Regression based Classifiers.
(Ans. : Refer sections 3.4.1, 3.4.2 and 3.4.3) **(10 Marks)**

...Chapter Ends

□□□

CHAPTER 4

Clustering

Syllabus

Cluster Analysis : Basic Concepts;
Partitioning Methods : K-Means, K-Medoids;
Hierarchical Methods : Agglomerative, Divisive, BIRCH;
Density-Based Methods : DBSCAN, OPTICS
What are outliers? Types, Challenges; Outlier Detection Methods: Supervised, Semi-Supervised, Unsupervised, Proximity based, Clustering based..

Syllabus Topic : Cluster Analysis - Basic Concepts

4.1 Cluster Analysis

4.1.1 What is Clustering ?

MU - Dec. 2015, Dec. 2016

- Clustering is an unsupervised learning problem.
- Clustering is a data mining (machine learning) technique used to place data elements into related groups without advance knowledge of the group definitions.
- It is a process of partitioning data objects into sub classes, which are called as clusters.
- A cluster contains data objects which have high inter similarity and low intra similarity.
- We can show this with a simple graphical example :

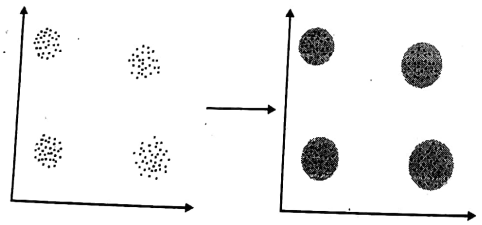


Fig. 4.1.1 : Clustering graphical example

- From Fig. 4.1.1, it can be seen that the data objects belong to 4 different clusters. Geometrical distance can be used as similarity criteria to find out which data object would belong to which of the four clusters. This type of clustering is called as distance based clustering.
- The other kind of clustering known as conceptual clustering in which data objects in a cluster are a part of it based on a common concept. In other words they fit based on some descriptive concept and not on a similarity measure.

Applications

Clustering algorithms can be applied in many disciplines :

- **Marketing :** Clustering can be used for targeted marketing. For e.g. Given a customer database, containing properties and past buying records. Similar groups of customers can be identified and grouped into one cluster.
- **Biology :** Clustering can also be used in classifying plants and animals into different classes based on their features.
- **Libraries :** Based on different details about books clustering can be used for book ordering.
- **Insurance :** With the help of clustering different groups of policy holders can be identified. For e.g. policy holders with high average claim cost or identifying some frauds.
- **City-planning :** Using details like house type, geographical locations, groups of houses can be identified using clustering.
- **Earthquake studies :** Clustering can also be used to identify dangerous zones based on earthquake epicenters.
- **WWW :** Clustering can be used to find groups of similar access patterns using weblog data. It can also be used for classification of documents.

Requirements

The main requirements that a clustering algorithm should satisfy are :

- Scalability.
- Dealing with different types of attributes.
- Discovering clusters with arbitrary shape.
- Minimal requirements for domain knowledge to determine input parameters.
- Ability to deal with noise and outliers.
- Insensitivity to order of input records.
- High dimensionality.
- Interpretability and usability.

Problems

- Because of time complexity, it creates problem to deal with large amount of data items.
- For distance based clustering, the effectiveness of method depends on distance definition.

4.1.2 Categories of Clustering Methods

- A good clustering method will produce high quality clusters with :
  - High intra-class similarity
  - Low inter-class similarity
- Major clustering methods can be classified into the following categories :
  1. **Partitioning methods :** In Partitioning based approach, various partitions are created and then they are evaluated based on certain criteria.
  2. **Hierarchical methods :** The set of data objects are decomposed hierarchically using certain criteria.
  3. **Density-based methods :** This approach is based on density (local cluster criteria) For e.g. Density connected points
  4. **Grid-based methods :** This approach is based on multi-resolution grid data structure.
- Jiawei Han and Kamber has given the overview of above mentioned clustering methods as shown in the Table 4.1.1.

Table 4.1.1 : Overview of various clustering methods

Method	General characteristics
Partitioning methods	• Find mutually exclusive clusters of spherical shape. • Distance-based. • May use mean or medoid (etc.) to represent cluster center. • Effective for small to medium size data sets.
Hierarchical methods	• Clustering is a hierarchical decomposition (i.e., multiple levels). • Cannot correct erroneous merges or splits. • May incorporate other techniques like micro-clustering or consider object "linkages".
Density-based methods	• Can find arbitrarily shaped clusters. • Clusters are dense regions of objects in space that are separated by low-density regions. • **Cluster density :** Each point must have a minimum number of points within its "neighborhood". • May filter out outliers.
Grid-based methods	• Use a multi-resolution grid data structure. • Fast processing time (typically independent of the number of data objects, yet dependent on grid size).

4.1.3 Different Distance Measures that can be used to Compute Distances between Two Clusters

MU - May 2016

- From the scientific and mathematical point of view, *distance* is defined as a quantitative degree of how far apart two objects are. Synonyms for *distance* include dissimilarity.
- Those distance measures satisfying the metric properties are simply called *metric* while other non-metric distance measures are occasionally called *divergence*. Synonyms for *similarity* include proximity and similarity measures are often called *similarity coefficients*. The choice of distance/similarity measures depends on the measurement type or representation of objects.

Table 4.1.2 : L_p Minkowski family

1.	Enclidean L_2	$d_{Euc} = \sqrt{\sum_{i=1}^{d} \lvert P_i - Q_i \rvert^2}$
2.	City block L_1	$d_{CB} = \sum_{i=1}^{d} \lvert P_i - Q_i \rvert$
3.	Minkowski L_p	$d_{Mk} = \sqrt[p]{\sum_{i=1}^{d} \lvert P_i - Q_i \rvert^p}$
4.	Chebyshev L_∞	$d_{Cheb} = \max_i \lvert P_i - Q_i \rvert$

- Euclid stated that the shortest distance between two points is a line and thus the equation (1) is predominantly known as Euclidean distance. It was often called Pythagorean metric since it is derived from the Pythagorean Theorem. In the late 19th century, Hermann Minkowski considered the city block distance.
- Other names for the equation (2) include rectilinear distance, taxicab norm, and Manhattan distance. Hermann also generalized the formulae (1) and (2) to the equation (3) which is coined after Minkowski. When p goes to infinite, the equation (4) can be derived and it is called the chessboard distance in 2D, the minimax approximation, or the Chebyshev distance named after Pafnuty Lvovich Chebyshev
- These distances (similarities) can be based on a single dimension or multiple dimensions, with each dimension representing a rule or condition for grouping objects.
- For example, if we were to cluster fast foods, we could take into account the number of calories they contain, their price, subjective ratings of taste, etc. The most straightforward way of computing distances between objects in a multi-dimensional space is to compute Euclidean distances.
- If we had a two or three-dimensional space this measure is the actual geometric distance between objects in the space (i.e., as if measured with a ruler).

4.1.4 Difference between Classification and Clustering

Sr. No.	Classification	Clustering
1.	In classification, a Training set containing data that have been previously categorized and based on this training set, the algorithms finds the category that the new data points belong to.	In clustering, the characteristics of similarity of data is not known in advance so using statistical concepts, we split the datasets into sub-datasets such that the Sub-datasets have "Similar" data called as clusters.
2.	Classification is supervised learning.	Clustering is unsupervised learning.
3.	You're given an unseen tuple and you are suppose to set a label or a class to that tuple For example, a company wants to classify their customers. When the company launches a product they want to classify which of their customers will buy their product and which ones will not buy it.	You're given a set of transaction history which recorded which customer bought what. By using clustering techniques, you can tell the segmentation of your customers.
4.	A common approach for classifiers is to use decision trees to partition and segment records.	There are a variety of algorithms used for clustering, but they all share the property of iteratively assigning records to a cluster unless it indicates that the process has converged to stable segments.

Syllabus Topic : Partitioning Methods

4.2 Partitioning Methods

Partitioning methods construct a partition of a database D of n objects into a set of K clusters.

Different partitioning methods

1. **Global optimal method :** Exhaustively enumerate all partitions.
2. **Heuristic methods :** K-means and K-medoids algorithms.
   - **K-means :** Each cluster is represented by the centre of the cluster.
   - **K-medoids or PAM (Partitioning Around Medoids) :** Each cluster is represented by one of the objects in the cluster.

Syllabus Topic : K-Means

4.2.1 K-means Clustering : (Centroid Based Technique)

MU - Dec. 2015, Dec. 2016

- In 1967, J. MacQueen and then in 1975 J. A. Hartigan and M. A. Wong developed K-means clustering algorithm.
- In K-means approach the data objects are classified based on their attributes or features into k number of clusters. The number of clusters i.e. K is an input given by the user.
- K-means is one of the simplest unsupervised learning algorithms.
- Define K centroids for K clusters which are generally far away from each other.
- Then Group the elements into clusters, which are nearer to the centroid of that cluster.
- After this first step, again calculate the new centroid for each cluster based on the elements of that cluster.
- Follow the same method and group the elements based on new centroid.
- In every step, the centroid changes and elements move from one cluster to another.
- Do the same process till no element is moving from one cluster to another i.e. till two consecutive steps with same centroid and same elements are obtained.
- Finally, this algorithm aims at minimizing an *objective function*, in this case a squared error function. The objective function is given below,

$$J = \sum_{j=1}^{k} \sum_{i=1}^{n} \|x_i^{(j)} - c_j\|^2$$

Where,

$x_i^{(j)}$ = A data point

c_j = The cluster centre

n = Number of data points

k = Number of clusters

$\|x_i^{(j)} - c_j\|^2$ = Distance measure between a data point $x_i^{(j)}$ and the cluster centre c_j

K-means algorithm

k : number of clusters

n : sample feature vectors $x_1, x_2, \ldots, x_n$

m_i : the mean of the vectors in cluster i

- Assume $k < n$.

- Make initial guesses for the means $m_1, m_2, ..., m_k$.
- Until there are no changes in any mean.
- Use the estimated means to classify the samples into clusters.

for i = 1 to k

Replace m_i with the mean of all of the samples for cluster i

end_for

end_until

- Following three steps are repeated until convergence :
- Iterate till no object moves to a different group :

1. Find the centroid coordinate.
2. Find the distance of each object to the centroids.
3. Based on minimum distance group the objects.

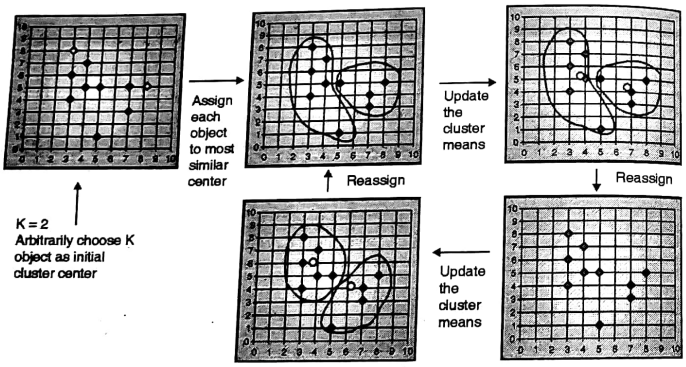


Fig. 4.2.1 : K-means graphical example

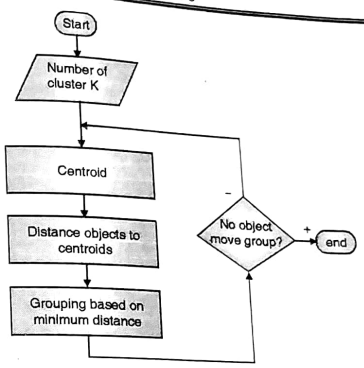


Fig. 4.2.2 : Basic steps for K-means clustering

Given a cluster $K_i = \{t_{i1}, t_{i2}, ..., t_{im}\}$, the cluster mean is $m_i = (1/m)(t_{i1} + ... + t_{im})$

Ex. 4.2.1 : Given : {2,4,10,12,3,20,30,11,25}, Assume number of cluster i.e. k = 2. **MU - Dec. 2015, Dec. 2016, 5 Marks**

Soln. :

Method 1 :

1. Randomly assign means : $m_1 = 3$, $m_2 = 4$.
2. The numbers which are close to mean $m_1 = 3$ are grouped into cluster K_1 and numbers which are close to mean $m_2 = 4$ are grouped into cluster K_2.
3. Again calculate the new mean for new cluster groups.
4. $K_1 = \{2,3\}$, $K_2 = \{4,10,12,20,30,11,25\}$, $m_1 = 2.5$, $m_2 = 16$
5. $K_1 = \{2,3,4\}$, $K_2 = \{10,12,20,30,11,25\}$, $m_1 = 3$, $m_2 = 18$
6. $K_1 = \{2,3,4,10\}$, $K_2 = \{12,20,30,11,25\}$, $m_1 = 4.75$, $m_2 = 19.6$
7. $K_1 = \{2,3,4,10,11,12\}$, $K_2 = \{20,30,25\}$, $m_1 = 7$, $m_2 = 25$
8. $K_1 = \{2,3,4,10,11,12\}$, $K_2 = \{20,30,25\}$
9. Stop as the clusters with these means (in step 7 and 8) are the same. The clusters in the last two groups are identical.
10. So the final answer is $K_1 = \{2,3,4,10,11,12\}$, $K_2 = \{20,30,25\}$.

Method 2 :

1. Given : {2,4,10,12,3,20,30,11,25}, Randomly assign alternative values to each cluster.
2. Number of cluster = 2, therefore
   K_1 = {2,10,3,30,25}, Mean = 14
   K_2 = {4,12,20,11}, Mean = 11.75
3. Re-assign
   K_1 = {20,30,25}, Mean = 25
   K_2 = {2,4,10,12,3,11}, Mean = 7
4. Re-assign
   K_1 = {20,30,25}, Mean = 25
   K_2 = {2,4,10,12,3,11}, Mean = 7

So the final answer is K_1 = {2,3,4,10,11,12}, K_2 = {20,30,25}

Ex. 4.2.2 : Use K-means algorithm to create 3 - clusters for given set of values : {2, 3, 6, 8, 9, 12, 15, 18, 22}

Soln. :

1. 2, 3, 6, 8, 9, 12, 15, 18, 22 – break into 3 clusters (Randomly assign data to three clusters) and calculate the mean value.
   K_1 = 2, 8, 15 – mean = 8.3
   K_2 = 3, 9, 18 – mean = 10
   K_3 = 6, 12, 22 – mean = 13.3
2. Re-assign
   K_1 = 2, 3, 6, 8, 9 – mean = 5.6
   K_2 = mean = 0
   K_3 = 12, 15, 18, 22 – mean = 16.75
3. Re-assign
   K_1 = 3, 6, 8, 9 – mean = 6.5
   K_2 = 2 – mean = 2
   K_3 = 12, 15, 18, 22 – mean = 16.75
4. Re-assign
   K_1 = 6, 8, 9 – mean = 7.6
   K_2 = 2, 3 – mean = 2.5
   K_3 = 12, 15, 18, 22 – mean = 16.75

5. Re-assign
   K_1 = 6, 8, 9 – mean = 7.6
   K_2 = 2, 3 - mean = 2.5
   K_3 = 12, 15, 18, 22 – mean = 16.75
6. Last two groups are same. So finally we got 3 clusters
   Cluster 1 = {6,8,9} , Cluster 2 = { 2,3 } , Cluster 3 = { 12,15,18,22 }

Ex. 4.2.3 : Consider four objects with two attribute (X and Y). These four objects are to be grouped together into two clusters. Following are the objects with their attribute values.

Object	X	Y
A	1	1
B	2	1
C	4	3
D	5	4

Soln. :

Each object represents one point with two attributes (X, Y) that can be represented as a coordinate in an attribute space as shown below.

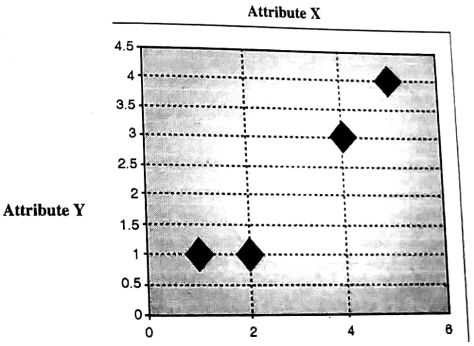


Fig. P. 4.2.3(a) : Graphical representation of Data Points

1. **Take initial centroids :** Consider object A and object B as the first centroids, then $C_1 = (1,1)$ and $C_2 = (2,1)$

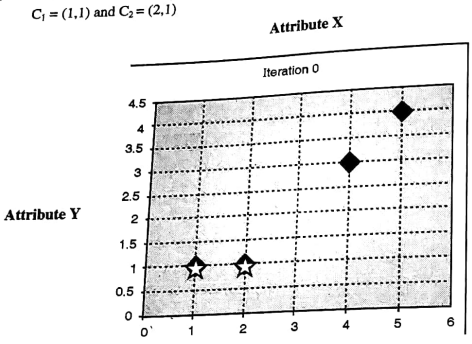


Fig. P. 4.2.3(b) : Randomly selected centroids C_1 and C_2 for two clusters

2. **Objects-centroids distance :** Using Euclidean distance, calculate the distance between cluster centroid to each object. For Iteration 0 we have,

$$D^0 = \begin{bmatrix} 0 & 1 & 3.61 & 5 \\ 1 & 0 & 2.83 & 4.24 \end{bmatrix} \begin{matrix} C_1 = (1,1) \text{ group - 1} \\ C_2 = (2,1) \text{ group - 2} \end{matrix}$$

A B C D

The above distance matrix shows the distance of each object with respect to the centroid of cluster C_1 and C_2.

For example, distance from object D = (5,4) to the first centroid C_1 = (1, 1) is 5 and to the second centroid C_2 = (2, 1) is 4.24, etc.

3. **Make the clusters of Objects :** Each object is assigned a group, based on the minimum distance of that object with respect to centroid of group. So object A is assigned to group-1 and object B,C,D are assigned grpup-2 and represented by 1.

$$G^0 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 \end{bmatrix} \begin{matrix} \text{group - 1} \\ \text{group - 2} \end{matrix}$$

A B C D

4. **Determine new centroids :** Based on the object belong to group, calculate the new centroid. As group-1 has only one object, so centroid of group-1 is $C_1 = (1, 1)$.

Group-2 has 3 objects so centroid is

$$C_2 = \left(\frac{2+4+5}{3}, \frac{1+3+4}{3}\right) = \left(\frac{11}{3}, \frac{8}{3}\right)$$

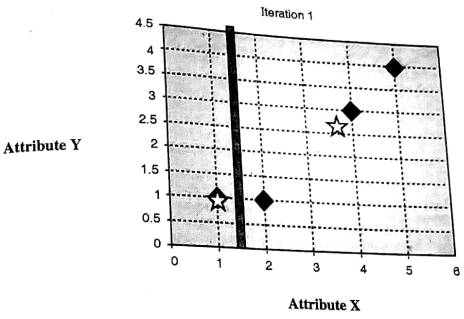


Fig. P. 4.2.3(c) : Cluster formation after First Iteration

5. Calculate **objects-centroids distances :** Compute the distance of all objects with respect to new centroids. So new distance matrix will be D^1 is given below :

$$D^1 = \begin{bmatrix} 0 & 1 & 3.61 & 5 \\ 3.14 & 2.36 & 0.47 & 1.89 \end{bmatrix} \begin{matrix} C_1 = (1\ 1) & \text{group - 1} \\ C_1 = \left(\frac{11}{3}, \frac{8}{3}\right) & \text{group - 2} \end{matrix}$$

A B C D

6. Make the new clusters of **objects :** Follow the step 3 given above. Based on minimum distance assign the group to each object. Now an object A and B belongs to group-1 and objects C and D belongs to group-2 as given below:

$$G^1 = \begin{bmatrix} 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \end{bmatrix} \begin{matrix} \text{group - 1} \\ \text{group - 2} \end{matrix}$$

A B C D

7. **Again determine centroids :** Repeat the step 4 to calculate new centroid.

$$C_1 = \left(\frac{1+2}{2}, \frac{1+1}{2}\right) = \left(1\frac{1}{2}, 1\right)$$

and

$$C_2 = \left(\frac{4+5}{2}, \frac{3+4}{2}\right) = \left(4\frac{1}{2}, 3\frac{1}{2}\right)$$

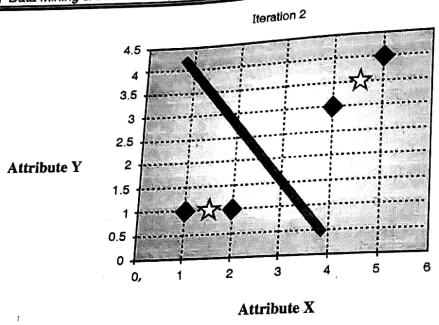


Fig. P. 4.2.3(d) : Two clusters with centroids

8. **Compute the objects-centroids distances :** Repeat step no 2, a new distance matrix for iteration 2 is obtained as shown below :

$$D^2 = \begin{bmatrix} 0.5 & 0.5 & 3.20 & 4.61 \\ 4.30 & 3.54 & 0.71 & 0.71 \end{bmatrix} \begin{matrix} C_1 = \left(1\frac{1}{2}, 1\right) & \text{group - 1} \\ C_2 = \left(4\frac{1}{2}, 3\frac{1}{2}\right) & \text{group - 2} \end{matrix}$$

A B C D

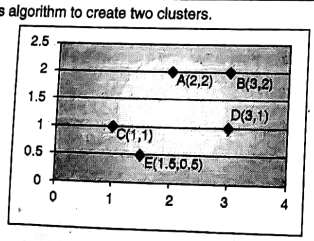
9. **Make the clusters of objects :** Assign each object based on the minimum distance calculated using Euclidean distance.

$$G^2 = \begin{bmatrix} 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \end{bmatrix} \begin{matrix} \text{group - 1} \\ \text{group - 2} \end{matrix}$$

A B C D

Last two iterations shows that object does not move from groups, so stop the iteration of k-means and that will be the final clusters as clustering has reached its stability.

Ex. 4.2.4 : Use K-means algorithm to create two clusters.



Soln. :

Object	attribute 1 (X)	attribute 2 (Y)
A	2	2
B	3	2
C	1	1
D	3	1
E	1.5	0.5

For simplicity we can find the adjacency matrix which gives distances of all object from each other.

Using Euclidean Distance we have

$$D(i, j) = \sqrt{|x_2 - x_1|^2 + |y_2 - y_1|^2}$$

$$D(i, j) = D(A,B) = \sqrt{(2-3)^2 + (2-2)^2} = 1,$$

Similarly we can compute for the rest.

	A	B	C	D	E
A	0	1	1.41	1.41	1.58
B	1	0	2.24	1	2.12
C	1.41	2.24	0	2	0.71
D	1.41	1	2	0	1.58
E	1.58	2.12	0.71	1.58	0

1. **Initial value of centroids :** Assume A and C as the first centroids. So the centroids are $C_1(2,2)$ and $C_2(1,1)$

2. **Objects-centroids distance :** Using Euclidean distance formula , the distance of each object with respect to centroid C_1 and C_2 is given below :

$D^0 =$

A	B	C	D	E	Cluster centroid
0	1	1.41	1.41	1.58	$C_1(2,2)$ – Group 1
1.41	2.24	0	2	0.71	$C_2(1,1)$ – Group 2

Note : Use adjacency matrix to get the distances or use the Euclidean distance formula for calculation of distances.

The object is represented by column in the distance matrix. The first row represents the distance of each object to the first centroid and the second row to the second centroid.

3. **Objects clustering :** Each object is assigned based on the minimum distance. Thus object A, B and D is assigned to group 1 and C and E to group 2. A value of 1 is assigned in the distance matrix if an object belongs to that group.

G^0 =

A	B	C	D	E	Cluster centroid
1	1	0	1	0	C_1(2,2) – Group 1
0	0	1	0	1	C_2(1,1) – Group 2

4. **Iteration-1, determine centroids :** After assigning the objects to their appropriate groups , now new centroids are calculated. Group 1 has three member thus the centroid C_1 is the average of the coordinates of those three members similarly Group 2 now has two members, thus the centroid is the average coordinate among the two members:

$$C_1 = \left(\frac{2+3+3}{3}, \frac{2+2+1}{3}\right) = (2.67, 1.67)$$

$$C_2 = \left(\frac{1+1.5}{2}, \frac{1+0.5}{2}\right) = (1.25, 0.75)$$

5. **Iteration-1, objects-centroids distances :** The next step is to compute the distance of all objects to the new centroids. Similar to step 2, we have distance matrix at iteration 1 is

D^1 =

A	B	C	D	E	Cluster Centroid
0.75	0.47	1.79	0.75	1.65	C_1(2.67,1.67) – Group 1
1.45	2.15	0.32	1.76	0.35	C_2(1.25,0.75) – Group 2

For example, distance from A = (2,2) to the first centroid C_1 (2.67,1.67) is $\sqrt{(2.67-2)^2+(1.67-2)^2} = 0.75$, and its distance to the second centroid C_2(1.25,0.75) is $\sqrt{(1.25-2)^2+(0.75-2)^2} = 1.45$, similarly calculate for the points B,C,D,E.

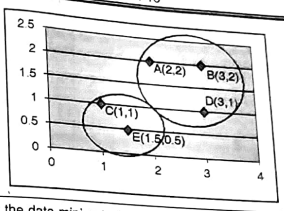
6. **Iteration-1, objects clustering :** Similar to step 3, we assign each object based on the minimum distance. The Group matrix is shown below

G^1 =

A	B	C	D	E	Cluster centroid
1	1	0	1	0	C_1(2.67,1.67) – Group 1
0	0	1	0	1	C_2 (1.25,0.75) – Group 2

By comparing the above results we observe that $G^0 = G^1$, this shows that object do not move any more to a different group. Thus K-means clustering has reached its stability.

So final clusters are Group 1= { A, B, D } and Group 2 = { C, E }



Ex. 4.2.5 : Suppose that the data mining task is to cluster the following points (with (x,y) representing locations) into 3 clusters.

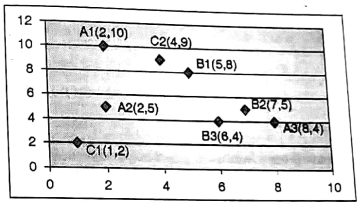
A1(2,10), A2(2,5), A3(8,4), B1(5,8), B2(7,5), B3(6,4), C1(1,2), C2(4,9)

The distance function is Euclidean distance. Suppose initially we assign A1, B1 and C1 as the centre of each cluster respectively, use the K-means algorithm to show only

(a) The three cluster centres after the first round execution.

(b) The final three clusters.

Soln. :



1. **Initial value of centroids :** In this we use A1 , B1and C1 as the first centroids. Let X1 and X2, X3 denote the coordinate of the centroids, then X1 = A1(2,10), X2 = B1(5,8) and X3 = C1(1,2)

2. **Objects-centroids distance :** We calculate the distance between cluster centroid to each object. Let us use Euclidean distance, then we have distance matrix at iteration 0 is

D^0 =

A1	A2	A3	B1	B2	B3	C1	C2	
0	5	8.48	3.61	7.07	7.21	8.06	2.24	X1=A1(2,10)
3.61	4.24	5	0	3.61	4.12	7.21	1.41	X2=B1(5,8)
8.06	3.16	7.28	7.21	6.71	5.39	0	7.62	X3=C1(1,2)

3. **Objects clustering :** We assign each object based on the minimum distance. Thus, A1 is assigned to group 1, point A3, B1, B2, B3, C2 are assigned to group2 and A2 and C1 are assigned to group3. The element of Group matrix below is 1 if and only if the object is assigned to that group.

G^0 =

A1	A2	A3	B1	B2	B3	C1	C2	
1	0	0	0	0	0	0	0	X1=A1(2,10)
0	0	1	1	1	1	0	1	X2=B1(5,8)
0	1	0	0	0	0	1	0	X3=C1(1,2)

4. **Iteration-1, determine centroids :**

$X1 = (2,10)$

$X2 = \left(\frac{8+5+7+6+4}{5}, \frac{4+8+5+4+9}{5}\right) = (6,6)$

$X3 = \left(\frac{2+1}{2}, \frac{5+2}{2}\right) = (1.5,3.5)$

5. **Iteration-1, objects-centroids distances :** The next step is to compute the distance of all objects to the new centroids. Similar to step 2, we have distance matrix at iteration 1 is

D^1 =

A1	A2	A3	B1	B2	B3	C1	C2	
0	5	8.48	3.61	7.07	7.21	8.06	2.24	X1(2,10)
5.66	4.12	2.83	2.24	1.41	2	6.40	3.61	X2(6,6)
6.52	1.58	6.52	5.70	5.70	4.52	1.58	6.04	X3(1.5,3.5)

6. **Iteration-1, objects clustering :** Similar to step 3, we assign each object based on the minimum distance. The Group matrix is shown below.

G^1 =

A1	A2	A3	B1	B2	B3	C1	C2	
1	0	0	0	0	0	0	1	X1(2,10)
0	0	1	1	1	1	0	0	X2(6,6)
0	1	0	0	0	0	1	0	X3(1.5,3.5)

7. **Iteration-2, determine centroids :**

$X1 = ((2+4)/2, (10+9)/2) = (3, 9.5)$

$X2 = ((8+5+7+6)/4, (4+8+5+4)/4) = (6.5, 5.25)$

$X3 = ((2+1)/2, (5+2)/2) = (1.5, 3.5)$

8. **Iteration-2, objects-centroids distances :**

D^2 =

A1	A2	A3	B1	B2	B3	C1	C2	
1.12	2.35	7.43	2.5	6.02	6.26	7.76	1.12	X1(3,9.5)
6.54	4.51	1.95	3.13	0.56	1.35	6.38	7.68	X2(6.5 ,5.25)
6.52	1.58	6.52	5.70	5.70	4.52	1.58	6.04	X3(1.5,3.5)

9. **Iteration-2, objects clustering :** We assign each object based on the minimum distance. The Group matrix is shown below.

G^2 =

A1	A2	A3	B1	B2	B3	C1	C2	
1	0	0	1	0	0	0	1	X1(3,9.5)
0	0	1	0	1	1	0	0	X2(6.5 ,5.25)
0	1	0	0	0	0	1	0	X3(1.5,3.5)

10. **Iteration-3, determine centroids :**

$X1 = ((2+5+4)/3, (10+9+8)/3) = (3.67, 9)$

$X2 = ((8+7+6)/3, (4+5+4)/3) = (7, 4.33)$

$X3 = ((2+1)/2, (5+2)/2) = (1.5, 3.5)$

11. **Iteration-3, objects-centroids distances :**

D^3 =

A1	A2	A3	B1	B2	B3	C1	C2	
1.95	4.33	6.61	1.66	5.20	5.52	7.49	0.33	X1(3.67,9)
6.01	5.04	1.05	4.17	0.67	1.05	6.44	5.55	X2(7 ,4.33)
6.52	1.58	6.52	5.70	5.70	4.52	1.58	6.04	X3(1.5,3.5)

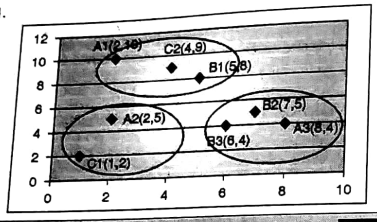
12. **Iteration-3, objects clustering :** We assign each object based on the minimum distance. The Group matrix is shown below.

G^3 =

A1	A2	A3	B1	B2	B3	C1	C2	
1	0	0	1	0	0	0	1	X1(3.67,9)
0	0	1	0	1	1	0	0	X2(7 ,4.33)
0	1	0	0	0	0	1	0	X3(1.5,3.5)

By comparing $G^3 = G^2$ we see that the objects do not move to new group therefore we can say that K means has reached its stability.

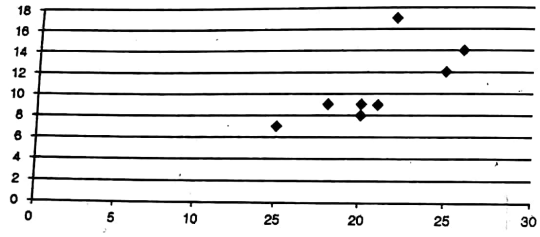
So final clusters are Group 1= { A1, B1, C2 } and Group 2 = { A3, B2 , B3} and Group 3 = {A2, C1}.



Ex. 4.2.3 : Use K-means to cluster the following data set into 3 clusters. **MU - May 2016, 10 Marks**

Protein	20	21	15	22	20	25	26	20	18	20
Fat	9	9	7	17	8	12	14	9	9	9

Soln. :



Sr. No.	Protein	Fat	
1	20	9	
2	21	9	c1
3	15	7	c2
4	22	17	
5	20	8	
6	25	12	
7	26	14	c3
8	20	9	

Sr. No.	Protein	Fat	
9	18	9	
10	20	9	

1. **Initial value of centroids :** In this we use 2 , 3 and 7 as the first centroids. Let X1 and X2, X3 denote the coordinate of the centroids, then X1 = A1(21,9), X2 = B1(15,7) and X3 = C1(26,14)

2. **Objects-centroids distance :** We calculate the distance between cluster centroid to each object. Let us use Euclidean distance, then we have distance matrix at iteration 0 is

D^0 =

	1	2	3	4	5	6	7	8	9	10
c1	1	0	6.32	8.06	1.41	5.00	7.07	1.00	3.00	1.00
c2	5.39	6.32	0.00	12.21	5.10	11.18	13.04	5.39	3.61	5.39
c3	7.81	7.07	13.04	5.00	8.49	2.24	0.00	7.81	9.43	7.81

3. **Objects clustering :** Assign each object based on the minimum distance. The element of Group matrix below is 1 if and only if the object is assigned to that group.

G^0 =

	1	2	3	4	5	6	7	8	9	10
c1	1	1	0	0	1	0	0	1	0	1
c2	0	0	1	0	0	0	0	0	1	0
c3	0	0	0	1	0	1	1	0	0	0

4. **Iteration-1, determine centroids :**

c1	19.8	9
c2	17.5	7.5
c3	24.33333	14.33333

5. **Iteration-1, objects-centroids distances :** The next step is to compute the distance of all objects to the new centroids. Similar to step 2, we have distance matrix at iteration 1 is

D^1 =

	1	2	3	4	5	6	7	8	9	10
c1	0.20	1.20	5.20	8.30	1.02	6.00	7.96	0.20	1.80	0.20
c2	2.92	3.81	2.55	10.51	2.55	8.75	10.70	2.92	1.58	2.92
c3	6.87	6.29	11.87	3.54	7.67	2.43	1.70	6.87	8.28	6.87

6. **Iteration-1, objects clustering :** Similar to step 3, we assign each object based on the minimum distance. The Group matrix is shown below.

G^1 =

	1	2	3	4	5	6	7	8	9	10
c1	1	1	0	0	1	0	0	1	0	1
c2	0	0	1	0	0	0	0	0	1	0
c3	0	0	0	1	0	1	1	0	0	0

7. **Iteration-2, determine centroids :**

c1	20.2	9
c2	16.5	8
c3	24.33333	14.33333

8. **Iteration-2, objects-centroids distances :** The next step is to compute the distance of all objects to the new centroids. Similar to step 2, we have distance matrix at iteration 2 is

D^2 =

	1	2	3	4	5	6	7	8	9	10
c1	0.20	0.80	5.57	8.20	1.02	5.66	7.66	0.20	2.20	0.20
c2	3.64	4.61	1.80	10.55	3.50	9.39	11.24	3.64	1.80	3.64
c3	6.87	6.29	11.87	3.54	7.67	2.43	1.70	6.87	8.28	6.87

9. **Iteration-2, objects clustering :** Similar to step 3, we assign each object based on the minimum distance. The Group matrix is shown below.

G^2 =

	1	2	3	4	5	6	7	8	9	10
c1	1	1	0	0	1	0	0	1	0	1
c2	0	0	1	0	0	0	0	1	0	0
c3	0	0	0	1	0	1	1	0	0	0

By comparing $G^2 = G^1$ we see that the objects do not move to new group therefore we can say that K means has reached its stability.

C1 = {1, 2, 5, 8, 10}

C2 = {3, 8}

C3 = {4, 6, 7}

Strength of K-means clustering

- **Relatively efficient** : O(ikn),

where *n* is number of objects,

k is number of clusters,

i is number of iterations.

Normally, k, i<<n.

- K-means often terminates at a **local optimum**.
- Techniques like **deterministic annealing** and **genetic algorithms are used to get the global optimum solution.**

Weakness of K-means clustering

- Applicable only when **mean** is defined.
- Need to specify *k*, the number of clusters, in advance.
- Unable to handle noisy data and **outliers** (outlier : objects with extremely large values)
- Not suitable to discover clusters with **non-convex shapes.**

Syllabus Topic : K-Medoids

4.2.2 K-Medoids (Representative Object-based Technique)

Instead of taking the mean value of the object in a cluster as a reference point, medoids can be used, which is the most centrally located object in a cluster.

- Also called as Partitioning Around Medoids (PAM).
- Handles outliers well.
- Ordering of input does not impact results.
- Does not scale well.
- Each cluster represented by one item, called the medoid.
- Initial set of K medoids randomly chosen.

In a single partition of data into K clusters where each cluster has a representative point that is centrally located point of the cluster based on some distance measure.

These representative points are called medoids.

Basic K-medoid algorithm :

1. Select K points as the initial medoids.
2. Assign all points to the closest medoid.
3. See if any other point is a "better" medoid.
4. Repeat steps 2 and 3 until the medoids don't change.

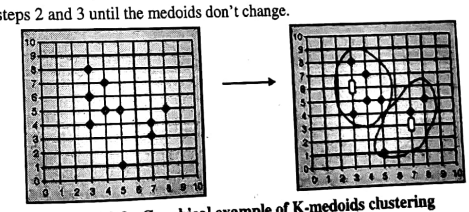


Fig. 4.2.3 : Graphical example of K-medoids clustering

- At each step in algorithm, medoids are changed if the overall cost is improved.
- C_{jih} – cost change for an item t_j associated with swapping medoid t_i with non-medoid t_h.

Algorithm given by Margaret H. Dunham

```
Input :
D = {t1, t2,...,tn} // set of elements
A  // Adjacency matrix showing distance between elements.
k  // Number of desired clusters.
Output :
K  // set of clusters.
PAM Algorithm :
Arbitrarily select k medoids from D;
Repeat
   For each th not a medoid do
      For each medoid t, do
         Calculate TCih;
find i,h where TCih is the smallest;
if TCih< 0 then
      replace medoid ti with th ;
until TCih≤ 0;
for each tie D do
   assign ti to Kj where dis(ti, tj) is the smallest over all medoids;
```

Calculation of swapping cost :

$$TC_{ih} = \sum_{j=1}^{n} C_{jih}$$

Advantages of PAM (Partitioning Around Medoids) :

- PAM works effectively for small data sets, but does not handle large data sets well.
- Complexity is $O(n(n-k)^2)$ for each iteration where n is number of data, k is number of clusters.
- PAM is more robust than k-means in the presence of noise and outliers.

Ex. 4.2.7 : Coordinates of objects are given below. Apply K-medoids (PAM). Number of clusters = 2.

Number	x co-ordinate	y co-ordinate
1	1.0	4.0
2	5.0	1.0
3	5.0	2.0

Number	x co-ordinate	y co-ordinate
4	5.0	4.0
5	10.0	4.0
6	25.0	4.0
7	25.0	6.0
8	25.0	7.0
9	25.0	8.0
10	29.0	7.0

Soln. :

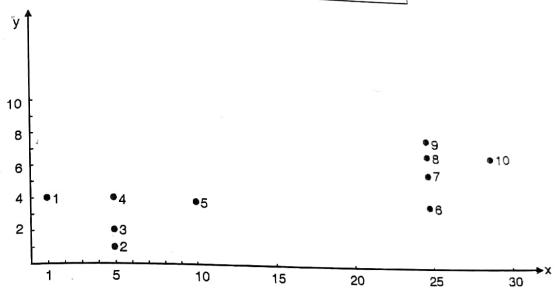


Fig. P. 4.2.7(a) : Two-dimensional example with 10 objects

Step I :

- Objects 1 and 5 are the selected representative objects initially. (Random selection)
- Calculate the distance of every object with respect to selected object 1 and 5.
- Find the closest representative object with respect to the selected object.
- Calculate the average value of minimal dissimilarity.
- Cost = Average value of minimal dissimilarity = 9.37.

Table P. 4.2.7(a) : Assignment of objects to two representative objects

Object number	Dissimilarity from object 1	Dissimilarity from object 5	Minimal dissimilarity	Closest representative object
1	0.00	9.00	0.00	1
2	5.00	5.83	5.00	1
3	4.47	5.39	4.47	1
4	4.00	5.00	4.00	1
5	9.00	0.00	0.00	5

Object number	Dissimilarity from object 1	Dissimilarity from object 5	Minimal dissimilarity	Closest representative object
6	24.00	15.00	15.00	5
7	24.08	15.13	15.13	5
8	24.19	15.30	15.30	5
9	24.33	15.52	15.52	5
10	28.16	19.24	19.24	5
			Average 9.37	

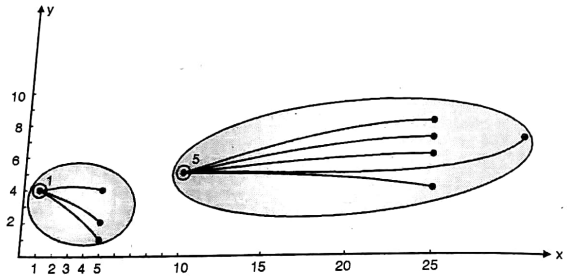


Fig. P. 4.2.7(b)

Step 2 :

- Select two objects randomly.
- Objects 4 and 8 are the selected representative objects.
- Repeat the step I.
- Cost = average value of minimal dissimilarity = 2.30.

Table P. 4.2.7(b) : Assignment of objects to two representative objects

Object number	Dissimilarity from object 4	Dissimilarity from object 8	Minimal dissimilarity	Closest representative object
1	4.00	24.19	4.00	4
2	3.00	20.88	3.00	4
3	2.00	20.62	2.00	4
4	0.00	20.22	0.00	4
5	5.00	15.30	5.00	4

Object number	Dissimilarity from object 4	Dissimilarity from object 8	Minimal dissimilarity	Closest representative object
6	20.00	3.00	3.00	8
7	20.10	1.00	1.00	8
8	20.22	0.00	0.00	8
9	20.40	1.00	1.00	8
10	24.19	4.00	4.00	8
			Average 2.30	

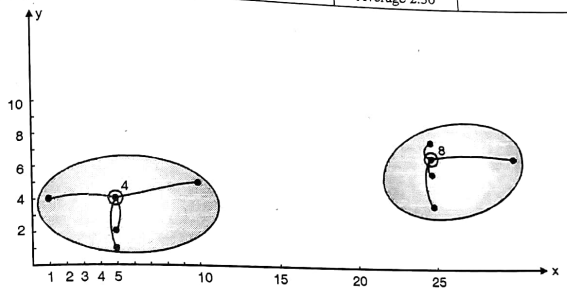


Fig. P. 4.2.7(c) : Clustering corresponding to the selections described in Table P. 4.2.6(a) and (b)

Step 3 :

- Calculation of swapping cost.
- Swapping cost = New cost – Old cost

  = 2.30 – 9.32 = – 7.02
- If swapping cost < 0

  replace medoid with new selected object.
- So new medoids are object 4 and object 8.

Step 4 :

Repeat step 2 and 3 until swapping cost <= 0

4.2.3 Sampling Based Method

1. CLARA

- Clustering large Applications.
- Built in statistical analysis packages, such as S+.
- Draw multiple samples of the data set, apply PAM on each sample.

- Performs better than PAM in larger data sets.
- Efficiency depends on the sample size.

2. CLARANS

- CLARANS is applicable to large applications which are based upon randomized search.
- CLARANS is more efficient than PAM and CLARA clustering algorithms.
- Search is over the sample of the neighbours of a node.
- It draws a sample of neighbours in each search step.
- It has main two parameters for clustering are maximum number of neighbours and number of local minima obtained.

Syllabus Topic : Hierarchical Clustering

4.3 Hierarchical Clustering

MU - May 2016, Dec. 2016

Various hierarchical clustering algorithms are :

- Single-linkage clustering, nearest-neighbour.
- Complete-linkage, furthest neighbour.
- Average-linkage, Unweighted Pair-Group Method Average (UPGMA).
- Weighted-pair group average, UPGMA weighted by cluster sizes.
- Within-groups clustering.
- Ward's method.

Hierarchical clustering technique (Basic algorithm)

1. Compute the proximity matrix (i.e. distance matrix).
2. Let each data point be a cluster.
3. Repeat.
4. Merge the two closest clusters.
5. Update the proximity matrix.
6. Until only a single cluster remains.

Note : Proximity matrix means the matrix which is symmetric, meaning that the numbers on the lower half of the diagonal will be the same as the numbers on the top half of the diagonal.

Different approaches to defining the distance between clusters distinguish the different algorithms i.e.

- **Single-linkage clustering :**

Single Linkage clustering is also called as minimum method, the minimum distance from any object of one cluster to any object of another cluster is considered. In the single linkage method, **D(A,B)** is computed as D(A,B) = Min { d(i,j) : Where object i is in cluster A and object j is in cluster B}

This measure of inter-group distance is illustrated in the Fig. 4.3.1.

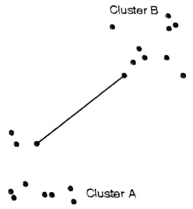


Fig. 4.3.1 : Single-linkage clustering

- **Complete-linkage clustering :**

Complete linkage also called as maximum method, the maximum distance between any object of one cluster to any object of another cluster is considered.

In the complete linkage method, **D(A,B)** is computed as D(A,B) = Max { d(i,j) : Where object i is in cluster A and object j is in cluster B}

The measure is illustrated in the Fig. 4.3.2.

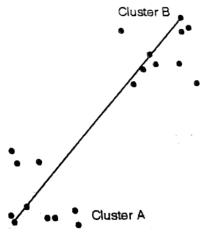


Fig. 4.3.2 : Complete-linkage clustering

- **Average-linkage clustering :**

In average-linkage clustering, we consider the distance between any two clusters A and B is taken to be the average of all distances between pairs of objects "i" in A and "j" in B, that is, the mean distance between elements of each cluster.

In the average linkage method, **D(A,B)** is computed as D(A,B) = mean{ d(i,j) : Where object i is in cluster A and object j is in cluster B}

$$\text{proximity}(\text{Cluster}_i, \text{Cluster}_j) = \frac{\sum_{p_i \in \text{Cluster}_i} \sum_{p_j \in \text{Cluster}_i} \text{proximity}(p_i, p_j)}{|\text{Cluster}_i| * |\text{Cluster}_j|}$$

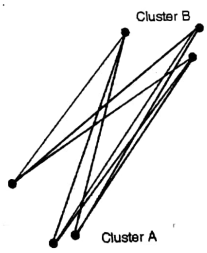


Fig. 4.3.3 : Average-linkage clustering

The Fig. 4.3.3 illustrates average linkage clustering.

- **Centroid clustering :**

  In centroid method, the distance between two clusters is calculated by finding the distance between two centroids (i.e. mean value of cluster) of the clusters. At every step, two clusters are combined that have minumum centroid distance.

  In the centroid clustering method, D(A,B) is computed as

  $D(A,B) = d(A_{centroid}, B_{centroid})$ where $A_{centroid}$ is the mean value or centroid of cluster A and $B_{centroid}$ is the mean value or centroid of cluster B. The Fig. 4.3.4 illustrates centroid clustering.

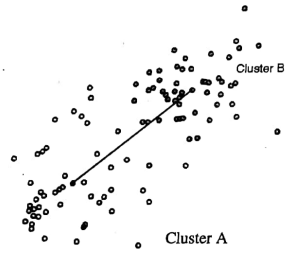


Fig. 4.3.4 : Centroid clustering

Syllabus Topic : Agglomerative Hierarchical Clustering

4.3.1 Agglomerative Hierarchical Clustering

In Hierarchical clustering algorithms, either top down or bottom up approach is followed. In Bottom up approach, every object is considered to be a cluster and in subsequent iterations they are merged in to single cluster. Therefore it is also called as Hierarchical Agglomerative Clustering (HAC).

An HAC clustering is typically visualized as a *dendrogram* as shown in Fig. 4.3.5 where each merge is represented by a horizontal line.

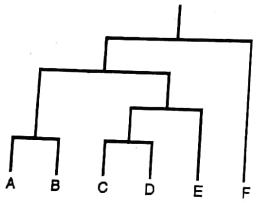


Fig. 4.3.5 : Dendrogram

What is Dendrogram ?

- **Dendrogram :** A tree data structure which illustrates hierarchical clustering techniques.
- Each level shows clusters for that level.
  - *Leaf – individual clusters*
  - *Root – one cluster*
- A cluster at level i is the union of its children clusters at level i + 1.
- A clustering of the data objects is obtained by cutting the dendrogram at the desired level, then each connected component forms a cluster.
- The flow chart of agglomerative hierarchical clustering algorithm is shown in Fig. 4.3.6.

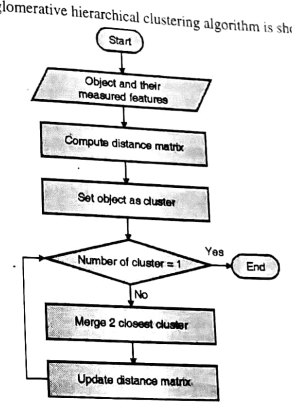


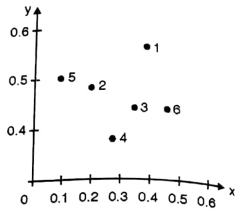
Fig. 4.3.6 : Flowchart of agglomerative hierarchical clustering

Ex. 4.3.1 : Assume that the database D is given by the table below. Follow single link technique to find clusters in D. Use Euclidean distance measure.

		X	Y
	p1	0.40	0.53
	p2	0.22	0.38
D =	p3	0.35	0.32
	p4	0.26	0.19
	p5	0.08	0.41
	p6	0.45	0.30

Soln. :

Step 1 : Plot the objects in n-dimensional space (where n is the number of attributes). In our case we have 2 attributes x and y, so we plot the objects p1, p2, ... p6 in 2-dimensional space :



Step 2 : Calculate the distance from each object (point) to all other points, using Euclidean distance measure, and place the numbers in a distance matrix.

The formula for Euclidean distance between two points i and j is :

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

where x_{i1} is the value of attribute 1 for i and x_{j1} is the value of attribute 1 for j and so on, as many attributes we have ... shown up to p i.e. x_{ip} in the formula.

In our case, we only have 2 attributes. So, the Euclidean distance between our points p1and p2, which have attributes x and y would be calculated as follows:

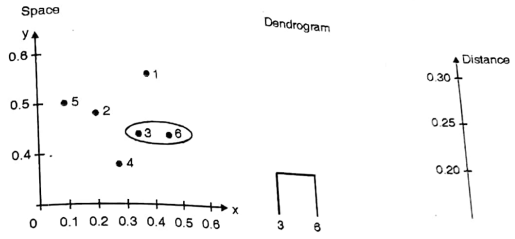
$$d(p1, p2) = \sqrt{|xp_1 - xp_1|^2 + |yp_1 - yp_2|^2}$$
$$= \sqrt{|0.40 - 0.22|^2 + |0.53 - 0.38|^2}$$
$$= \sqrt{|0.18|^2 + |0.15|^2} = \sqrt{0.0324 + 0.0225}$$
$$= \sqrt{0.0549}$$
$$= 0.2343$$

Analogically, we calculate the distance to the remaining points and we will receive the following values :

Distance matrix :

p1	0					
p2	0.24	0				
p3	0.22	0.15	0			
p4	0.37	0.20	0.15	0		
p5	0.34	0.14	0.28	0.29	0	
p6	0.23	0.25	**0.11**	0.22	0.39	0
	p1	p2	p3	p4	p5	p6

Step 3 : In the above matrix, p6 and p3 are two clusters with shortest distance 0.11, so merge p6 and p3 and make single cluster (p3,p6). Now re-compute the distance matrix.



Distance matrix :

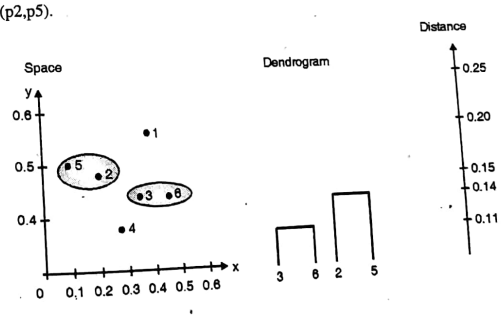
p1	0				
p2	0.24	0			
(p3, p6)	**0.22**	**0.15**	0		
p4	0.37	0.20	**0.15**	0	
p5	0.34	0.14	**0.28**	0.29	0
	p1	p2	**(p3, p6)**	p4	p5

To calculate the distance of p1 from (p3,p6) :

dist((p3, p6), p1) = MIN (dist(p3, p1) , dist(p6, p1))
= MIN (0.22, 0.23) //from original matrix
= 0.22

Step 4 : Repeat Step 3 until one single cluster is formed i.e. merge all the clusters.

a. Now p2 and p5 have the smallest distance from above matrix, so merge p2 and p5 to get cluster(p2,p5).



p1	0			
(p2, p5)	**0.24**	0		
(p3, p6)	0.22	**0.15**	0	
p4	0.37	**0.20**	0.15	0
	p1	**(p2, p5)**	(p3, p6)	p4

The distance between (p3, p6) and (p2, p5) is calculated as given below :

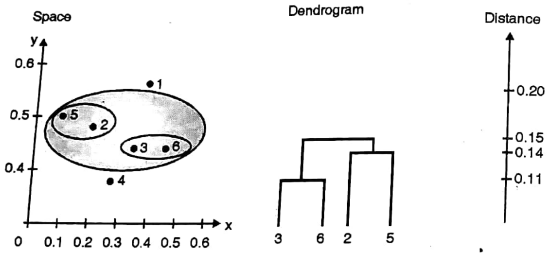
dist((p3, p6), (p2, p5)) = MIN (dist(p3, p2) , dist(p6, p2), dist(p3, p5), dist(p6, p5))

= MIN (0.15, 0.25, 0.28, 0.39) //from original matrix

= 0.15

b. Repeat Step 3.

Merge (p2, p5) and (p3, p6) as having minimum distance i.e.0.15 and again compute distance matrix.



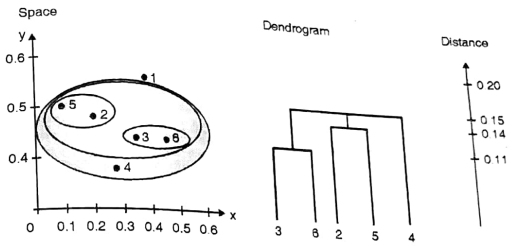
Distance matrix :

p1	0		
(p2, p5, p3, p6)	**0.22**	0	
p4	0.37	**0.15**	0
	p1	**(p2, p5, p3, p6)**	p4

c. Since we have more clusters to merge, we continue to repeat Step 3.

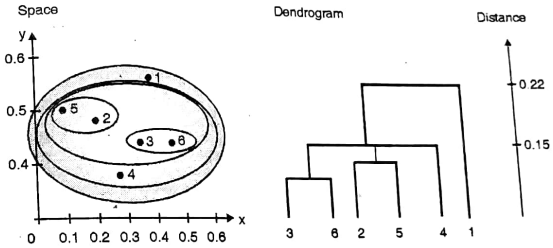
So, looking at the last distance matrix above, we see that (p2, p5, p3, p6) and p4 have the smallest distance from all i.e. 0.15. So, we merge those two in a single cluster, and re-compute the distance matrix.

Space dendrogram :



d. Since we have more clusters to merge, we continue to repeat Step 3.

So, looking at the last distance matrix above, we see that (p2, p5, p3, p6, p4) and p1 have the smallest distance - 0.22 (the only one left). So, we merge those two in a single cluster. There is no need to re-compute the distance matrix, as there are no more clusters to merge.



Stopping condition :

We indicated that "each object is placed in a separate cluster, and at each step we merge the closest pair of clusters, *until certain termination conditions are satisfied*".

To stop clustering either user has to specify the number of clusters he wants or algorithm has to make decision to stop clustering at which level. Through dendrogram, we can notice the merging of clusters at various distances. If merging of clusters is at high distance then we can stop clustering at that level.

Complete link :

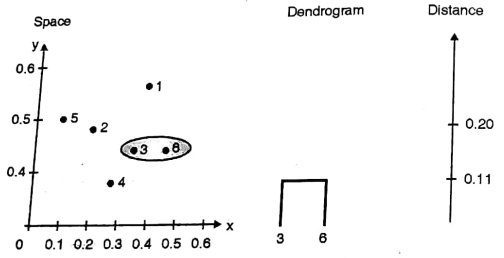
Step 1 and step 2 :

Refer the single link solution (same)

Distance matrix :

	p1	p2	p3	p4	p5	p6
p1	0					
p2	0.24	0				
p3	0.22	0.15	0			
p4	0.37	0.20	0.15	0		
p5	0.34	0.14	0.28	0.29	0	
p6	0.23	0.25	0.11	0.22	0.39	0

Step 3 : Identify the two clusters with the shortest distance in the matrix, and merge them together. Re-compute the distance matrix, as those two clusters are now in a single cluster.



By looking at the distance matrix above, we see that p3 and p6 have the smallest distance from all i.e. 0.11 So, we merge those two in a single cluster and re-compute the distance matrix.

$dist$((p3, p6), p1) = MAX ($dist$(p3, p1) , $dist$(p6, p1))
= MAX (0.22, 0.23) //from original matrix
= 0.23

$dist$((p3, p6), p2) = MAX ($dist$(p3, p2) , $dist$(p6, p2))
= MAX (0.15, 0.25) //from original matrix
= 0.25

$dist$((p3, p6), p4) = MAX ($dist$(p3, p4) , $dist$(p6, p4))
= MAX (0.15, 0.22) //from original matrix
= 0.22

$dist$((p3, p6), p5) = MAX ($dist$(p3, p5) , $dist$(p6, p5))
= MAX (0.28, 0.39) //from original matrix
= 0.39

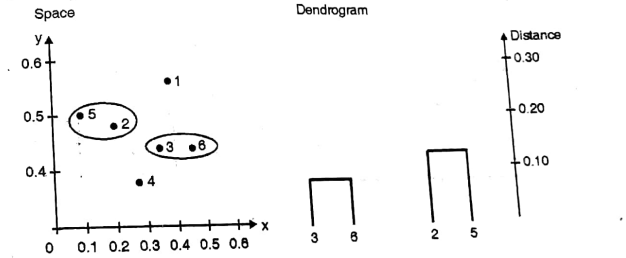
New distance matrix :

	p1	p2	(p3, p6)	p4	p5
p1	0				
p2	0.24	0			
(p3, p6)	**0.23**	**0.25**	0		
p4	0.37	0.20	**0.22**	0	
p5	0.34	0.14	**0.39**	0.29	0

Step 4 : Consider the following distance matrix

	p1	p2	(p3, p6)	p4	p5
p1	0				
p2	0.24	0			
(p3, p6)	**0.23**	**0.25**	0		
p4	0.37	0.20	**0.22**	0	
p5	0.34	0.14	**0.39**	0.29	0

So, looking at the above distance matrix, we see that p2 and p5 have the smallest distance from all = 0.14. So, we merge those two in a single cluster, and re-compute the distance matrix using the following calculations.



$dist$((p2, p5), p1) = MAX ($dist$(p2, p1) , $dist$(p5, p1))
= MAX (0.24, 0.34) //from original matrix
= 0.34

$dist$((p2, p5), (p3,p6)) = MAX ($dist$(p2, p3) , $dist$(p2, p6), $dist$(p5, p3), $dist$(p5, p6))
= MAX (0.15, 0.25, 0.28, 0.39) //from original matrix
= 0.39

$dist$((p2, p5), p4) = MAX ($dist$(p2, p4) , $dist$(p5, p4))

= MAX (0.20, 0.29) //from original matrix

= 0.29

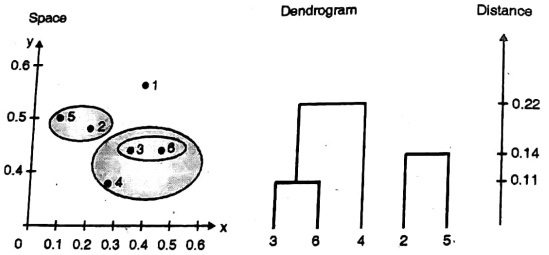
Therefore new distance matrix is :

	p1	(p2, p5)	(p3, p6)	p4
p1	0			
(p2, p5)	**0.34**	0		
(p3, p6)	0.23	**0.39**	0	
p4	0.37	**0.29**	0.22	0

Step 5 : Consider the following matrix

	p1	(p2, p5)	(p3, p6)	p4
p1	0			
(p2, p5)	**0.34**	0		
(p3, p6)	0.23	**0.39**	0	
p4	0.37	**0.29**	0.22	0

The minimum distance is 0.22, so merge (p3, p6) and p4



dist((p3, p6, p4) , p1) = MAX (*dist*(p3, p1) , *dist*(p6, p1), dist(p4, p1))

= MAX (0.22, 0.23, 0.37) //from original matrix

= 0.37

dist((p3, p6, p4) , (p2,p5)) = MAX (*dist*(p3, p2) , *dist*(p3, p5),
dist(p6, p2),dist(p6, p5),
dist(p4, p2) , dist(p4, p5))

= MAX (0.15, 0.28, 0.25, 0.39, 0.20, 0.29)

= 0.39

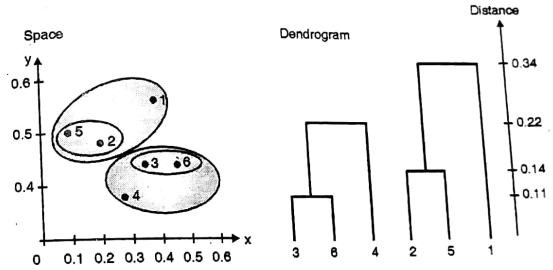
Therefore new distance matrix

	p1	(p2, p5)	(p3, p6, p4)
p1	0		
(p2, p5)	**0.34**	0	
(p3, p6, p4)	0.37	**0.39**	0

Step 6 : Now consider the following distance matrix

	p1	(p2, p5)	(p3, p6, P4)
p1	0		
(p2, p5)	**0.34**	0	
(p3, p6, p4)	0.37	**0.39**	0

Since the minimum distance is 0.34, merge (p2, p5) with p1.

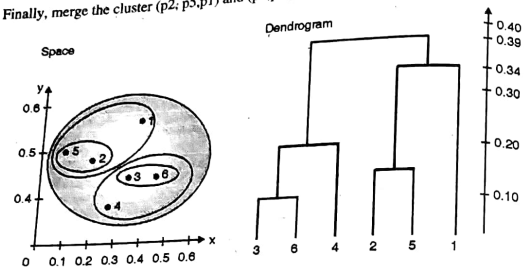


dist ((p2, p5, p1), (p3, p6, p4) = 0.39

Therefore new distance matrix

	(p2, p5,p1)	(p3, p6, P4)
(p2, p5,p1)	0	
(p3, p6, p4)	0.39	0

Finally, merge the cluster (p2, p5,p1) and (p3,p6,p4)



Average link :

***Step 1* and *Step 2* :**

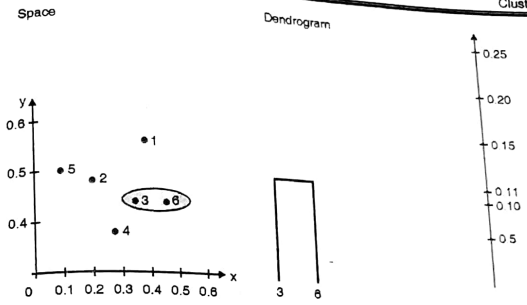
Refer the single link solution

Distance matrix :

p1	0					
p2	0.24	0				
p3	0.22	0.15	0			
p4	0.37	0.20	0.15	0		
p5	0.34	0.14	0.28	0.29	0	
p6	0.23	0.25	0.11	0.22	0.39	0
	p1	p2	p3	p4	p5	p6

***Step 3* :** Identify the two clusters with the shortest distance in the matrix, and merge them together. Re-compute the distance matrix, as those two clusters are now in a single cluster.

By looking at the distance matrix above, we see that p3 and p6 have the smallest distance from all – 0.11 So, we merge those two in a single cluster, and re-compute the distance matrix.



dist((p3 p6), p1) = 1/2 (dist(p3,p1) + dist(p6,p1))
= 0.5 × (0.22 + 0.23) = 0.23

dist((p3 p6), p2) = 1/2 (dist(p3,p2) + dist(p6,p2))
= 0.5 × (0.15 + 0.25) = 0.2

dist((p3 p6), p4) = 1/2 (dist(p3,p4) + dist(p6,p4))
= 0.5 × (0.15 + 0.22) = 0.19

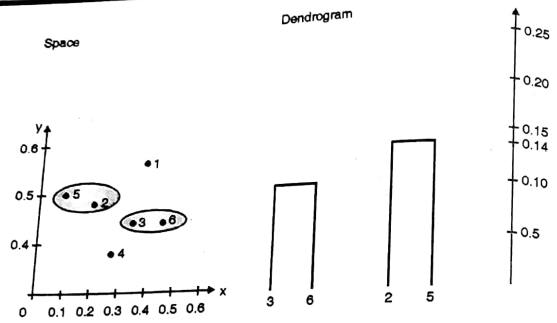
dist((p3 p6), p5) = 1/2 (dist(p3,p5) + dist(p6,p5))
= 0.5 × (0.28 + 0.39) = 0.34

Distance matrix :

p1	0				
p2	0.24	0			
(p3, p6)	**0.23**	**0.2**	0		
p4	0.37	0.20	**0.19**	0	
p5	0.34	0.14	**0.34**	0.29	0
	p1	p2	**(p3, p6)**	p4	p5

Step 4 :

So, looking at the above distance matrix above, we see that p2 and p5 have the smallest distance from all – 0.14. So, we merge those two in a single cluster, and re-compute the distance matrix.



dist((p2, p5), p1) = 1/2 (dist(p2,p1) + dist(p5,p1))

= 0.5 × (0.24 + 0.34)

= 0.29

dist((p2, p5), (p3,p6)) = 1/4 (dist(p2,p3) + dist(p2,p6) + dist(p5,p3) + dist(p5,p6))

= 1/4 × (0.15 + 0.25 + 0.28 + 0.39)

= 0.27

dist((p2, p5), p4) = 1/2 (dist(p2,p4) + dist(p5,p4))

= 0.5 × (0.14 + 0.29)

= 0.22

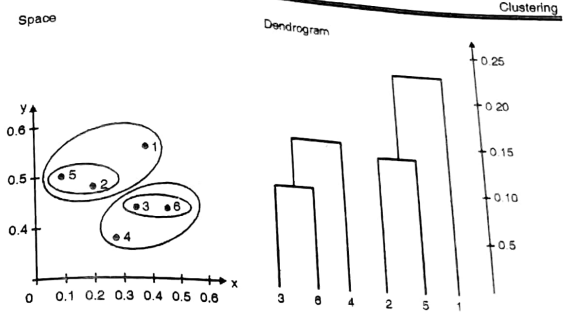
Distance matrix :

p1	0			
(p2, p5)	**0.29**	0		
(p3, p6)	0.22	**0.27**	0	
p4	0.37	**0.22**	0.15	0
	p1	**(p2, p5)**	(p3, p6)	p4

Since, we have merged (p2, p5) together in a cluster, we now have one entry for (p2, p5) in the table, and no longer have p2 or p5 separately.

Step 5 :

Now the closest clusters are merged, where distance is the smallest measure by looking at the maximum distance between any two points.



dist((p3, p6, p4), (p2, p5)) = 1/6 × (dist(p3,p2) + dist(p3,p5) + dist(p6,p2) + dist(p6,p5) + dist(p4,p2) + dist(p4,p5))

= 1/6 × (0.15 + 0.28 + 0.25 + 0.39 + 0.20 + 0.29)

= 0.26

dist((p3, p6, p4), p1) = 1/3 × (dist(p3,p1) + dist(p6,p1) + dist(p4,p1))

= 1/3 × (0.22 + 0.23 + 0.37)

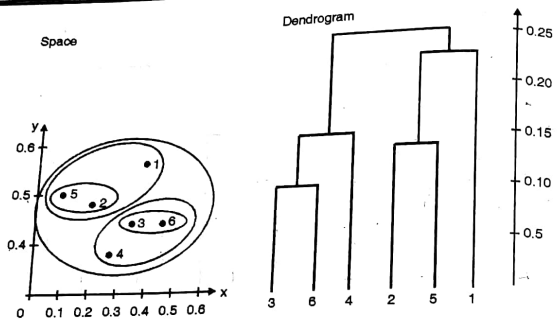
= 0.27

Distance matrix :

p1	0		
(p2, p5)	**0.24**	0	
(p3, p6, p4)	0.27	**0.26**	0
	p1	**(p2, p5)**	(p3, p6, P4)

Step 6 :

So merge the cluster (p2,p5) and p1.



$$dist(\text{(p3 p6 p4), (p2,p5,p1)}) = 1/9 \times (dist(p3,p2) + dist(p3,p5) + dist(p3,p1) + dist(p6,p2) + dist(p6,p5) + dist(p6,p1) + dist(p4,p2) + dist(p4,p5) + dist(p4+p1))$$

$$= 1/9 \times (0.15 + 0.28 + 0.22 + 0.25 + 0.39 + 0.23 + 0.20 + 0.29 + 0.37)$$

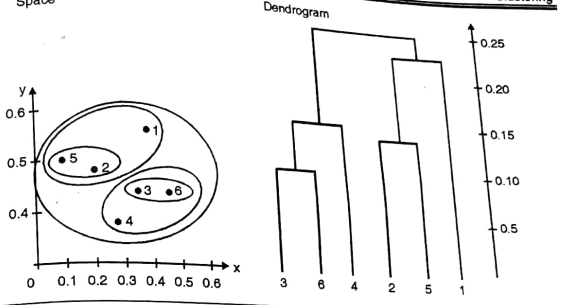
$$= 0.26$$

Distance matrix :

(p2, p5,p1)	0	
(p3, p6, p4)	0.26	0
	(p2, p5,p1)	(p3, p6, P4)

We need to re-compute the distance from all other points / clusters to our new cluster - (p2, p5,p1) to (p3,p6,p4) and enter the maximum distance in the above matrix (use original distance matrix).

Finally merge the cluster (p2, p5,p1) and (p3,p6,p4).



Ex. 4.3.2 : For given distance matrix, draw single link, complete link and average link dendrogram.

Soln. :

(I) Single link :

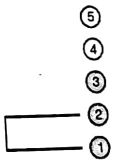
Step 1 :

$$\begin{array}{c|ccccc} & 1 & 2 & 3 & 4 & 5 \\ 1 & 0 & & & & \\ 2 & 2 & 0 & & & \\ 3 & 6 & 3 & 0 & & \\ 4 & 10 & 9 & 7 & 0 & \\ 5 & 9 & 8 & 5 & 4 & 0 \end{array} \Rightarrow \begin{array}{c|cccc} & (1,2) & 3 & 4 & 5 \\ (1,2) & 0 & & & \\ 3 & 3 & 0 & & \\ 4 & 9 & 7 & 0 & \\ 5 & 8 & 5 & 4 & 0 \end{array}$$

$d_{(1,2)3} = \min \{d_{1,3}, d_{2,3}\} = \min \{6, 3\} = 3$

$d_{(1,2)4} = \min \{d_{1,4}, d_{2,4}\} = \min \{10, 9\} = 9$

$d_{(1,2)5} = \min \{d_{1,5}, d_{2,5}\} = \min \{9, 8\} = 8$

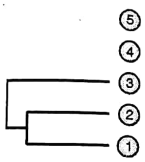


Step 2 :

$$\begin{array}{c|ccccc} & 1 & 2 & 3 & 4 & 5 \\ 1 & 0 & & & & \\ 2 & 2 & 0 & & & \\ 3 & 6 & 3 & 0 & & \\ 4 & 10 & 9 & 7 & 0 & \\ 5 & 9 & 8 & 5 & 4 & 0 \end{array} \Rightarrow \begin{array}{c|cccc} & (1,2) & 3 & 4 & 5 \\ (1,2) & 0 & & & \\ 3 & 3 & 0 & & \\ 4 & 9 & 7 & 0 & \\ 5 & 8 & 5 & 4 & 0 \end{array} \Rightarrow \begin{array}{c|ccc} & (1,2,3) & 4 & 5 \\ (1,2,3) & 0 & & \\ 4 & 7 & 0 & \\ 5 & 5 & 4 & 0 \end{array}$$

$d_{(1,2,3),4} = \min \{d_{(1,2),4}, d_{3,4}\} = \min \{9, 7\} = 7$

$d_{(1,2,3)5} = \min \{d_{(1,2),5}, d_{3,5}\} = \min \{8, 5\} = 5$



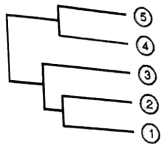
Step 3 :

$$\begin{array}{c|ccccc} & 1 & 2 & 3 & 4 & 5 \\ 1 & 0 & & & & \\ 2 & 2 & 0 & & & \\ 3 & 6 & 3 & 0 & & \\ 4 & 10 & 9 & 7 & 0 & \\ 5 & 9 & 8 & 5 & 4 & 0 \end{array} \Rightarrow \begin{array}{c|cccc} & (1,2) & 3 & 4 & 5 \\ (1,2) & 0 & & & \\ 3 & 3 & 0 & & \\ 4 & 9 & 7 & 0 & \\ 5 & 8 & 5 & 4 & 0 \end{array} \Rightarrow \begin{array}{c|ccc} & (1,2,3) & 4 & 5 \\ (1,2,3) & 0 & & \\ 4 & 7 & 0 & \\ 5 & 5 & 4 & 0 \end{array}$$

$$\Downarrow$$

$$\begin{array}{c|cc} & (1,2,3) & 4,5 \\ (1,2,3) & 0 & \\ (4,5) & 5 & 0 \end{array}$$

$d_{(1,2,3),(4,5)} = \min \{d_{(1,2,3),4}, d_{(1,2,3),5}\} = \min \{7, 5\} = 5$



(ii) Complete link :

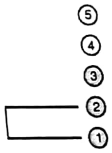
Step 1 :

$$\begin{array}{c|ccccc} & 1 & 2 & 3 & 4 & 5 \\ 1 & 0 & & & & \\ 2 & 2 & 0 & & & \\ 3 & 6 & 3 & 0 & & \\ 4 & 10 & 9 & 7 & 0 & \\ 5 & 9 & 8 & 5 & 4 & 0 \end{array} \Rightarrow \begin{array}{c|cccc} & (1,2) & 3 & 4 & 5 \\ (1,2) & 0 & & & \\ 3 & 6 & 0 & & \\ 4 & 10 & 7 & 0 & \\ 5 & 9 & 5 & 4 & 0 \end{array}$$

$d_{(1,2),3} = \max \{d_{1,3}, d_{2,3}\} = \max \{6, 3\} = 6$

$d_{(1,2)4} = \max \{d_{1,4}, d_{2,4}\} = \max \{10, 9\} = 10$

$d_{(1,2)5} = \max \{d_{1,5}, d_{2,5}\} = \max \{9, 8\} = 9$

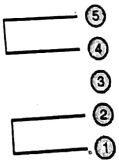


Step 2 :

$$\begin{array}{c|ccccc} & 1 & 2 & 3 & 4 & 5 \\ 1 & 0 & & & & \\ 2 & 2 & 0 & & & \\ 3 & 6 & 3 & 0 & & \\ 4 & 10 & 9 & 7 & 0 & \\ 5 & 9 & 8 & 5 & 4 & 0 \end{array} \Rightarrow \begin{array}{c|cccc} & (1,2) & 3 & 4 & 5 \\ (1,2) & 0 & & & \\ 3 & 6 & 0 & & \\ 4 & 10 & 7 & 0 & \\ 5 & 9 & 5 & 4 & 0 \end{array} \Rightarrow \begin{array}{c|ccc} & (1,2) & 3 & (4,5) \\ (1,2) & 0 & & \\ 3 & 6 & 0 & \\ (4,5) & 10 & 7 & 0 \end{array}$$

$d_{(1,2),(4,5)} = \max \{d_{(1,2),4}, d_{(1,2),5}\} = \max \{10, 9\} = 10$

$d_{3,(4,5)} = \max \{d_{3,4}, d_{3,5}\} = \max \{7, 5\} = 7$

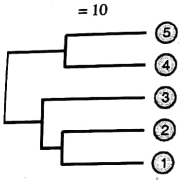


Step 3 :

$$\begin{array}{c} \\ 1 \\ 2 \\ 3 \\ 4 \\ 5 \end{array}\begin{array}{c} \begin{array}{ccccc} 1 & 2 & 3 & 4 & 5 \end{array} \\ \begin{bmatrix} 0 & & & & \\ 2 & 0 & & & \\ 6 & 3 & 0 & & \\ 10 & 9 & 7 & 0 & \\ 9 & 8 & 5 & 4 & 0 \end{bmatrix} \end{array} \Rightarrow \begin{array}{c} \\ (1,2) \\ 3 \\ 4 \\ 5 \end{array}\begin{array}{c} \begin{array}{cccc} (1,2) & 3 & 4 & 5 \end{array} \\ \begin{bmatrix} 0 & & & \\ 6 & 0 & & \\ 10 & 7 & 0 & \\ 9 & 5 & 4 & 0 \end{bmatrix} \end{array} \Rightarrow \begin{array}{c} \\ (1,2) \\ 3 \\ (4,5) \end{array}\begin{array}{c} \begin{array}{ccc} (1,2) & 3 & (4,5) \end{array} \\ \begin{bmatrix} 0 & & \\ 6 & 0 & \\ 10 & 7 & 0 \end{bmatrix} \end{array}$$

$$d_{(1,2,3),(4,5)} = \max\{d_{(1,2),(4,5)}, d_{3,(4,5)}\} = \max\{10, 7\} = 10$$

$$\begin{array}{c} \\ (1,2,3) \\ (4,5) \end{array}\begin{array}{c} \begin{array}{cc} (1,2,3) & (4,5) \end{array} \\ \begin{bmatrix} 0 & \\ 10 & 0 \end{bmatrix} \end{array}$$



(iii) Average link :

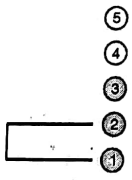
Step 1 :

$$\begin{array}{c} \\ 1 \\ 2 \\ 3 \\ 4 \\ 5 \end{array}\begin{array}{c} \begin{array}{ccccc} 1 & 2 & 3 & 4 & 5 \end{array} \\ \begin{bmatrix} 0 & & & & \\ 2 & 0 & & & \\ 6 & 3 & 0 & & \\ 10 & 9 & 7 & 0 & \\ 9 & 8 & 5 & 4 & 0 \end{bmatrix} \end{array} \Rightarrow \begin{array}{c} \\ (1,2) \\ 3 \\ 4 \\ 5 \end{array}\begin{array}{c} \begin{array}{cccc} (1,2) & 3 & 4 & 5 \end{array} \\ \begin{bmatrix} 0 & & & \\ 4.5 & 0 & & \\ 9.5 & 7 & 0 & \\ 8.5 & 5 & 4 & 0 \end{bmatrix} \end{array}$$

$$d_{(1,2),3} = \frac{1}{2}(d_{1,3} + d_{2,3}) = \frac{6+3}{2} = 4.5$$

$$d_{(1,2),4} = \frac{1}{2}(d_{1,4} + d_{2,4}) = \frac{10+9}{2} = 9.5$$

$$d_{(1,2),5} = \frac{1}{2}(d_{1,5} + d_{2,5}) = \frac{9+8}{2} = 8.5$$



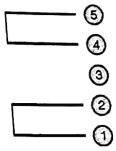
Step 2 :

$$\begin{array}{c} \\ 1 \\ 2 \\ 3 \\ 4 \\ 5 \end{array}\begin{array}{c} \begin{array}{ccccc} 1 & 2 & 3 & 4 & 5 \end{array} \\ \begin{bmatrix} 0 & & & & \\ 2 & 0 & & & \\ 6 & 3 & 0 & & \\ 10 & 9 & 7 & 0 & \\ 9 & 8 & 5 & 4 & 0 \end{bmatrix} \end{array} \Rightarrow \begin{array}{c} \\ (1,2) \\ 3 \\ 4 \\ 5 \end{array}\begin{array}{c} \begin{array}{cccc} (1,2) & 3 & 4 & 5 \end{array} \\ \begin{bmatrix} 0 & & & \\ 4.5 & 0 & & \\ 9.5 & 7 & 0 & \\ 8.5 & 5 & 4 & 0 \end{bmatrix} \end{array} \Rightarrow \begin{array}{c} \\ (1,2) \\ 3 \\ (4,5) \end{array}\begin{array}{c} \begin{array}{ccc} (1,2) & 3 & (4,5) \end{array} \\ \begin{bmatrix} 0 & & \\ 4.5 & 0 & \\ 9 & 6 & 0 \end{bmatrix} \end{array}$$

$$d_{(1,2),(4,5)} = \frac{1}{4}\{d_{1,4} + d_{1,5} + d_{2,4} + d_{2,5}\} = 1/4\{10 + 9 + 9 + 8\}$$

$$= 9$$

$$d_{3,(4,5)} = \frac{1}{2}\{d_{3,4} + d_{3,5}\} = 6$$



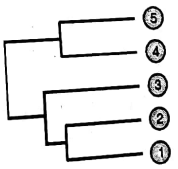
Step 3 :

$$\begin{array}{c} \\ 1 \\ 2 \\ 3 \\ 4 \\ 5 \end{array}\begin{array}{c} \begin{array}{ccccc} 1 & 2 & 3 & 4 & 5 \end{array} \\ \begin{bmatrix} 0 & & & & \\ 2 & 0 & & & \\ 6 & 3 & 0 & & \\ 10 & 9 & 7 & 0 & \\ 9 & 8 & 5 & 4 & 0 \end{bmatrix} \end{array} \Rightarrow \begin{array}{c} \\ (1,2) \\ 3 \\ 4 \\ 5 \end{array}\begin{array}{c} \begin{array}{cccc} (1,2) & 3 & 4 & 5 \end{array} \\ \begin{bmatrix} 0 & & & \\ 4.5 & 0 & & \\ 9.5 & 7 & 0 & \\ 8.5 & 5 & 4 & 0 \end{bmatrix} \end{array} \Rightarrow \begin{array}{c} \\ (1,2) \\ 3 \\ (4,5) \end{array}\begin{array}{c} \begin{array}{ccc} (1,2) & 3 & (4,5) \end{array} \\ \begin{bmatrix} 0 & & \\ 4.5 & 0 & \\ 9 & 6 & 0 \end{bmatrix} \end{array}$$

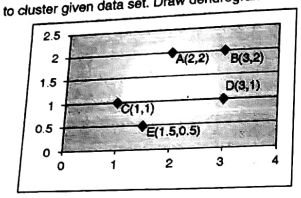
$$\begin{array}{c} \\ (1,2,3) \\ (4,5) \end{array}\begin{array}{c} \begin{array}{cc} (1,2,3) & (4,5) \end{array} \\ \begin{bmatrix} 0 & \\ 8 & 0 \end{bmatrix} \end{array}$$

$$d_{(1,2,3),(4,5)} = \frac{1}{6}(d_{1,4} + d_{1,5} + d_{2,4} + d_{2,5} + d_{3,4} + d_{3,5}) = 1/6\{10 + 9 + 9 + 8 + 7 + 5\}$$

$$= 8$$



Ex. 4.3.3 : Use the data given below. Create adjacency matrix. Use single link or complete link algorithm to cluster given data set. Draw dendrogram.



Soln. :

Object	Attribute 1 (X)	Attribute 2 (Y)
A	2	2
B	3	2
C	1	1
D	3	1
E	1.5	0.5

For simplicity we can find the adjacency matrix which gives distances of all object from each other. Using Euclidean distance we have

$$D(i,j) = \sqrt{|x_2 - x_1|^2 + |y_2 - y_1|^2}$$

$$D(A,B) = \sqrt{(2-3)^2 + (2-2)^2} = 1,$$

Similarly we can compute for the rest.

	A	B	C	D	E
A	0				
B	1	0			
C	1.41	2.24	0		
D	1.41	1	2	0	
E	1.58	2.12	0.71	1.58	0

(I) Single link :

Step 1 : Since C, E is minimum we can combine clusters C, E

	A	B	(C,E)	D
A	0			
B	1	0		
(C,E)	1.41	2.12	0	
D	1.41	1	1.58	0

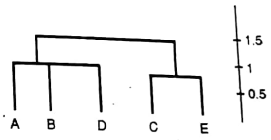
Step 2 : Now A and B is having minimum value therefore we merge these two clusters.

	(A, B)	(C,E)	D
(A,B)	0		
(C, E)	1.41	0	
D	1	1.58	0

Step 3 : Cluster (A,B) and D can be merged together as they are having minimum distance value

	(A, B,D)	(C,E)
(A,B,D)	0	
(C, E)	1.41	0

Step 4 : In the last step there are only two clusters to be combined they are, (A,B,D) and (C,E). Now the final dendrogram is



(II) Complete link :

Step 1 : Closest clusters are merged where the distance is the smallest measured by looking at the maximum distance between any two point.

Since C, E is minimum we can combine clusters C, E.

	A	B	(C, E)	D
A	0			
B	1	0		
(C,E)	1.58	2.24	0	
D	1.41	1	2	0

Step 2 : Now A and B is having minimum closest measure value therefore we merge these two clusters.

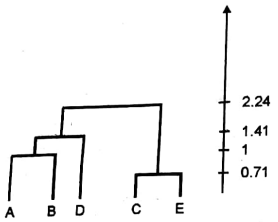
	(A, B)	(C,E)	D
(A,B)	0		
(C, E)	2.24	0	
D	1.41	2	0

Step 3 : Cluster (A,B) and D can be merged together as they are having minimum distance value.

	(A,B,D)	(C,E)
(A,B,D)	0	
(C,E)	2.24	0

Step 4 : In the last step there are only two clusters to be combined they are, (A,B,D) and (C,E).

Final dendrogram :



Ex. 4.3.4 : Discuss the agglomerative algorithm using following data and plot a dendrogram using single link approach. The following figure contains sample data items indicating the distance between the elements.

Item	E	A	C	B	D
E	0	1	2	2	3
A	1	0	2	5	3
C	2	2	0	1	6
B	2	5	1	0	3
D	3	3	6	3	0

Soln. :

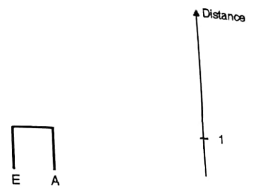
Given :

Distance matrix :

	E	A	C	B	D
E	0				
A	1	0			
C	2	2	0		
B	2	5	1	0	
D	3	3	6	3	0

Step 1 : From above given distance matrix, E and A clusters are having minimum distance, so merge them together to form cluster (E,A).

Dendrogram



Distance matrix :

dist((E A), C) = MIN (dist(E,C), dist(A,C))
= MIN (2,2)
= 2

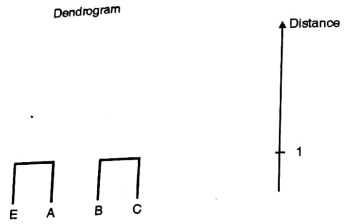
dist((E A), B) = MIN (dist(E,B), dist(A,B))
= MIN (2,5)
= 2

dist((E A), D) = MIN (dist(E,D), dist(A,D))
= MIN (3,3)
= 3

	E, A	C	B	D
E, A	0.			
C	2	0		
B	2	1	0	
D	3	6	3	0

Step 2 : Consider the distance matrix obtained in step 1 (given above)

Since B,C distance is minimum, we combine B and C.



Distance matrix :

dist((B C), (E A)) = MIN (dist(B E), dist(B A), dist(C E), dist(C A))

= MIN (2, 5, 2, 2)

= 2

dist((B C), D) = MIN (dist(B,D), dist(C D))

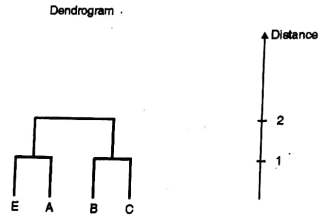
= MIN (3, 6)

= 3

	E, A	B, C	D
E, A	0		
B, C	2	0	
D	3	3	0

Step 3 : Consider the distance matrix obtained in step 2 (given above)

Since (E,A) and (B,C) distance is minimum, we combine them



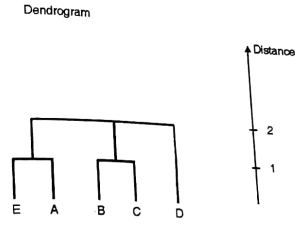
dist((E A), (B C)) = MIN (dist(E,B), dist(E C), dist(A,B), dist(A C))

= MIN (2, 2, 2, 5, 2) = 2

	E,A, B, C	D
E, A, B,C	0	
D	2	0

Step 4 : Finally combine D with (E A B C)

Final dendrogram :



Ex. 4.3.5 : Following table gives fat and proteins content of items. Apply single linkage clustering and construct dendrogram.

Food Item	Protein	Fat
1	1.1	60
2	8.2	20
3	4.2	35
4	1.5	21
5	7.6	15
6	2.0	55
7	3.9	39

Soln. :

Step 1 : Calculate the distance matrix using Euclidian distance formula :

Cluster number	C1	C2	C3	C4	C5	C6	C7
C1	0	40.62	25.19	39.00	45.46	5.08	21.18
C2		0	15.52	6.77	5.03	35.54	19.48
C3			0	14.25	20.28	20.12	4.01
C4				0	8.55	34.00	18.19
C5					0	40.39	24.28
C6						0	16.11

From the above table, the minimum distance between any two points is 4.01 and this distance is between C3 and C7. So, these two points can be merged into a single point (cluster) and is called the C37.

Step 2 : Calculate the new distance matrix with C37 using single linkage clustering.

Therefore,

dis(C37,4) = min(dis(3,4), dis(7,4)) = min(14.25, 18.19) = 14.25

Similarly, calculate the other distances to get the distance matrix.

Cluster number	C1	C2	C37	C4	C5	C6
C1	0	40.62	21.18	39.00	45.46	5.08
C2		0	15.52	6.77	5.03	35.54
C37			0	14.25	20.28	16.11
C4				0	8.55	34.00
C5					0	40.39
C6						0

In the above matrix distance between points 2 and 5 is minimum i.e. 5.03. So combine the cluster as C25.

Step 3 : Calculate the new distance matrix with C25 using single linkage clustering.

Cluster number	C1	C25	C37	C4	C6
C1	0	40.62	21.18	39.00	5.08
C25		0	15.52	6.77	35.54
C37			0	14.25	16.11
C4				0	34.00
C6					0

In the above matrix distance between C1 and C6 in minimum which is 5.08. So newly formed new cluster is C16.

Step 4 : Calculate the new distance matrix with cluster C16.

Cluster number	C16	C25	C37	C4
C16	0	35.54	16.11	34.00
C25		0	15.52	6.77
C37			0	14.25
C4				0

The minimum distance is 6.77. So combine clusters C25 and C4

Step 5 : Calculate new distance matrix.

Cluster number	C16	C254	C37
C16	0	34.00	16.11
C254		0	14.25
C37			0

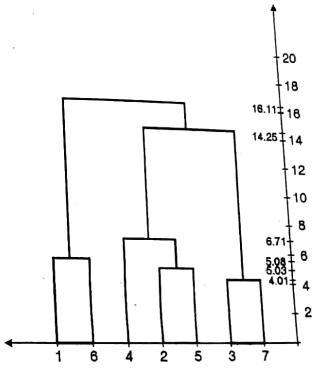
Combine the clusters C254 and C37 which has minimum distance 14.25.

Step 6 : New distance matrix is

Cluster number	C16	C25437
C16	0	16.11
C25437		0

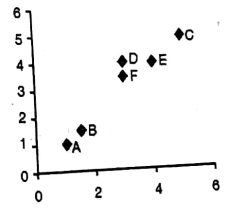
So finally combine the clusters C25437 and C16.

Dendrogram is given below :



Ex. 4.3.6 : Suppose we have 6 objects (with name A, B, C, D, E and F) and each object have two measured features (X1 and X2).

	X1	X2
A	1	1
B	1.5	1.5
C	5	5
D	3	4
E	4	4
F	3	3.5



Apply Single linkage clustering and draw Dendrogram.

Soln. :

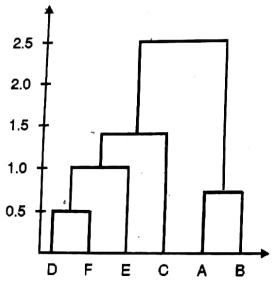
We have given an input distance matrix of size 6 by 6.

Dist	A	B	C	D	E	F
A	0.00	0.71	5.66	3.61	4.24	3.20
B	0.71	0.00	4.95	2.92	3.54	2.50
C	5.66	4.95	0.00	2.24	1.41	2.50
D	3.61	2.92	2.24	0.00	1.00	0.50
E	4.24	3.54	1.41	1.00	0.00	1.12
F	3.20	2.50	2.50	0.50	1.12	0.00

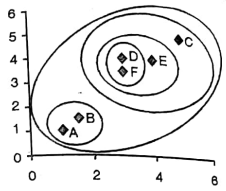
We summarized the results of computation as follow :

1. In the beginning we have 6 clusters: A, B, C, D, E and F.
2. We merge cluster D and F into cluster (D, F) at distance 0.50.
3. We merge cluster A and cluster B into (A, B) at distance 0.71.
4. We merge cluster E and (D, F) into ((D, F), E) at distance 1.00.
5. We merge cluster ((D, F), E) and C into (((D, F), E), C) at distance 1.41.
6. We merge cluster (((D, F), E), C) and (A, B) into ((((D, F), E), C), (A, B)) at distance 2.50.
7. The last cluster contain all the objects, thus conclude the computation.

Using this information, we can now draw the final results of a dendogram.



We can also plot the clustering hierarchy into XY space.



	X1	X2
A	1	1
B	1.5	1.5
C	5	5
D	3	4
E	4	4
F	3	3.5

Ex. 4.3.7 Use any hierarchical clustering algorithm to cluster the following 8 examples into 3 clusters :
A1 = (2, 10), A2 = (2, 5), A3 = (8, 4), A4 = (5, 8),
A5 = (7, 5), A6 = (6, 4), A7 = (1, 2), A8 = (4, 9)
MU - May 2016, 10 Marks

Soln. :

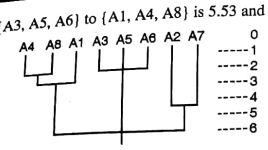
The distance matrix based on the Euclidean distance is given below :

	A1	A2	A3	A4	A5	A6	A7	A8
A1	0	$\sqrt{25}$	$\sqrt{36}$	$\sqrt{13}$	$\sqrt{50}$	$\sqrt{52}$	$\sqrt{65}$	$\sqrt{5}$
A2		0	$\sqrt{37}$	$\sqrt{18}$	$\sqrt{25}$	$\sqrt{17}$	$\sqrt{10}$	$\sqrt{20}$
A3			0	$\sqrt{25}$	$\sqrt{2}$	$\sqrt{2}$	$\sqrt{53}$	$\sqrt{41}$
A4				0	$\sqrt{13}$	$\sqrt{17}$	$\sqrt{52}$	$\sqrt{2}$
A5					0	$\sqrt{2}$	$\sqrt{45}$	$\sqrt{25}$
A6						0	$\sqrt{29}$	$\sqrt{29}$
A7							0	$\sqrt{58}$
A8								0

Average link

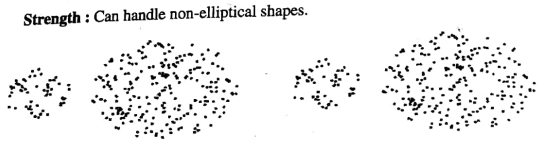
d	k	K
0	8	{A1}, {A2}, {A3}, {A4}, {A5}, {A6}, {A7}, {A8}
1	8	{A1}, {A2}, {A3}, {A4}, {A5}, {A6}, {A7}, {A8}
2	5	{A4, A8}, {A1}, {A3, A5, A6}, {A2}, {A7}
3	4	{A4, A8, A1}, {A3, A5, A6}, {A2}, {A7}
4	3	{A4, A8, A1}, {A3, A5, A6}, {A2, A7}
5	3	{A4, A8, A1}, {A3, A5, A6}, {A2, A7}
6	1	{A4, A8, A1, A3, A5, A6, A2, A7}

Average distance from {A3, A5, A6} to {A1, A4, A8} is 5.53 and 5.57 to {A2, A7}



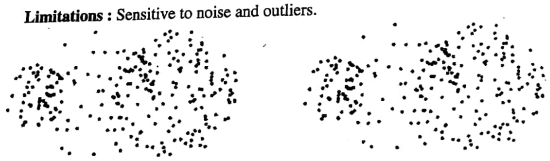
Comparison of the three methods (based on distance formula)

1. ***Single-link :***
   - "Loose" clusters.
   - **Strength :** Can handle non-elliptical shapes.



Original points Two clusters

   - **Limitations :** Sensitive to noise and outliers.

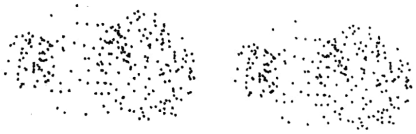


Original points Two clusters

2. **Complete-link :**
   - "Tight" clusters.
   - **Strength :** Less susceptible to noise and outliers.

Original points Two clusters

Limitations :
- Tends to break large clusters.
- Biased towards globular clusters.



Original points Two clusters

3. **Average-link :**
   - "In between".
   - Compromise between Single and complete link.
   - **Strengths :** Less susceptible to noise and outliers.
   - **Limitations :** Biased towards globular clusters.

Note : Which one is the best? Depends on what you need!

Agglomerative algorithm given by Margaret H. Dunham :

Input :

D = {$t_1, t_2, \ldots, t_n$} // set of elements
A // Adjacency matrix showing distance between elements

Output

DE // Dendrogram represented as a set of ordered triples
Agglomerative algorithm:
D = 0;
k = n ;
K = {{t_1},.....,{t_n}}
DE = <d, k, K >; // Initially dendrogram contains each element in its own cluster.
Repeat
  Oldk = k;
  d = d + 1 ;
  for each pair of K_i, K_j K do
    ave = average distance between all t_i K and t_j Kd,k,K
    if ave <= d, then
      K = K – {Ki} – {Kj} U { Ki U Kj}

```
K = oldk - 1
DE = DE U < d,k, K> // New set of clusters added to dendrogram.
Until k = 1
```

Syllabus Topic : Divisive Hierarchical Clustering

4.3.2 Divisive Hierarchical Clustering

- In this data objects are grouped in a top down manner.
- Initially all objects are in one cluster.
- Then the cluster is subdivided into smaller and smaller pieces, until each object forms a cluster on its own or until it satisfies certain termination conditions as the desired numbers of clusters are obtained.
- Divisive methods are not generally available, and rarely have been applied.

AGNES (AGglomerative NESting) and DIANA (DIvisive ANAlysis) :

- Agglomerative and divisive hierarchical clustering on data objects {1,2,3,4,5}.

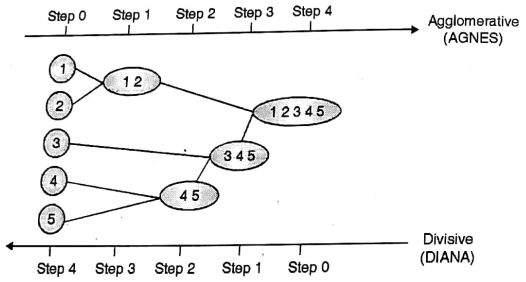


Fig. 4.3.7 : Agglomerative v/s Divisive hierarchical clustering

Advantages

- Is simple and outputs a hierarchy, a structure that is more informative.
- It does not require us to pre-specify the number of clusters.

Disadvantages

- Selection of merge or split points is critical as once a group of objects is merged or split, it will operate on the newly generated clusters and will not undo what was done previously.
- Thus merge or split decisions if not well chosen may lead to low-quality clusters.

Difference between agglomerative and divisive

Agglomerative	Divisive
Start with the points as individual clusters.	Start with one, all-inclusive cluster.
At each step, merge the closest pair of clusters until only one cluster (or k clusters) left.	At each step, split a cluster until each cluster contains a point (or there are k clusters).

Syllabus Topic : BIRCH

4.3.3 BIRCH (Balanced Iterative Reducing and Clustering using Hierarchies)

MU - Dec. 2015

- It is a scalable clustering method.
- BIRCH technique place significant emphasis on scalability of very large data sets.
- This technique is efficient for data where averages make sense.
- It is based on the notation of CF (Clustering Feature).
- CF tree is a height-balanced tree that stores the clustering features for Hierarchical clustering.
- Cluster of data points (vector) is represented by a triple numbers (N,LS,SS).

  Where,

  N = Number of items in the subcluster,

  LS = Linear sum of the points,

  SS = Sum of the square of points
- A CF Tree is a height-balanced tree that stores the clustering features in a hierarchy.
- Internal nodes store the sums of their descendants.

A CF tree structure is given as below :

- Each non-leaf node has at most B entries.
- Each leaf node has at most L CF entries which each satisfy threshold T, a maximum diameter or radius.
- P (page size in bytes) is the maximum size of a node.
- **Compact :** Each leaf node is a sub-cluster, not a data point.

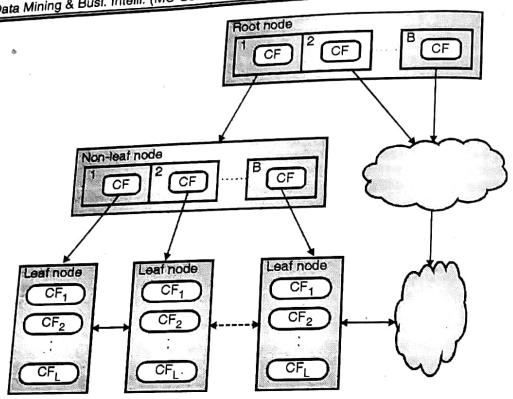


Fig. 4.3.8 : A CF tree structure

Detail algorithm as a flow-graph :

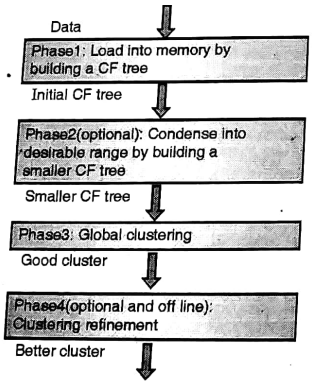


Fig. 4.3.9

Algorithm :

Step 1: The data is loaded into the memory.

- In one scan a CF tree in memory is built with the data.

- Subsequent phases are :
  - Fast (The processing is carried out on sub clusters and not on individual data points, no more I/O needed)
  - Accurate (outliers are separated)
  - Less order Sensitive (As initial ordering of data is done by the CF-Tree)

Step 2 : Condense data

- Data resizing is done, which helps step 3 to be executed on optimally sized data.
- With a Larger T, CF tree is rebuilt.
- More outliers are removed.
- Crowded sub clusters are grouped together.
- Condensing is optional.

Step 3 : Global clustering

- Use clustering algorithm (e.g., HC, KMEANS, CLARANS) on CF entries.
- The problem is fixed where natural clusters span nodes.

Step 4 : Cluster refining

- Extra passes over the dataset are carried out and the data points are reassigned to the closest centroid from step 3.
- Refining is optional.
- The problem with CF trees is fixed where different leaf entries are assigned the same valued data points.
- Always converges to a minimum.
- Allows to discard more outliers.

Example :

Clustering feature :

$$CF = (N, LS, SS)$$

N : Number of data points

$$LS : \sum_{i=1}^{N} = X_i$$

$$SS : \sum_{i=1}^{N} = X_i^2$$

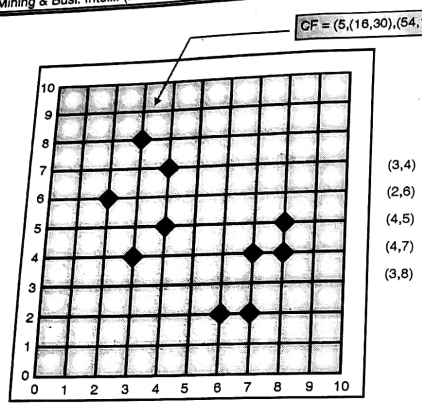


Fig. 4.3.10 : Clustering feature example

$N = 5$

$NS = (16,30)$ i.e. $3 + 2 + 4 + 4 + 3 = 16$ and $4 + 6 + 5 + 7 + 8 = 30$

$SS = (54,190)$ i.e. $3^2 + 2^2 + 4^2 + 4^2 + 3^2 = 54$ and $4^2 + 6^2 + 5^2 + 7^2 + 8^2 = 190$

- **Advantage :** Finds a good clustering with a single scan and improves the quality with a few additional scans.
- **Weakness :** Handles only numeric data, and sensitive to the order of the data record.
- **Complexity :** Algorithm complexity O(n) where n is number of objects to be clustered.

Practical use of BIRCH :

1. Pixel classification in images :
   - From top to bottom
   - BIRCH classification
   - Visible wavelength band
   - Near-infrared band
2. Image compression using vector quantization :
   - Generate codebook for frequently occurring patterns.
   - BIRCH performs faster than CLARANS or LBG, while getting better compression and nearly as good quality.
3. BIRCH works with very large data sets.

Fig. 4.3.11

4. Explicitly bounded by computational resource i.e. it runs with specified amount of memory.
5. Superior to CLARANS and KMEANS in terms of quality, speed, stability and scalability.

4.3.4 Advantages and Disadvantages of Hierarchical Clustering

Advantages

- It is simple and outputs a hierarchy, a structure that is more informative.
- It does not require pre-specifying the number of clusters.

Disadvantages

- Combined clusters cannot be splited.
- It doesn't give proper clustering if data is noisy as sensitive to noise and outliers.
- Breaking of big cluster into smaller pieces is one of the problem.

Syllabus Topic : Density-Based Methods

4.4 Density-Based Methods

- Density based method is used to find clusters of arbitrary shape.
- In Density-based approaches clusters are the region where objects are dense and separated from the low density object regions.
- It is mainly used to find the outliers, i.e. the rare events, e.g. detecting fraud activities in E-commerce.

Major features

- Discover clusters of arbitrary shape
- Handles noise
- One scan needed
- Need density parameters

Syllabus Topic : DBSCAN

4.4.1 DBSCAN (Density Based Methods)

MU - May 2015, May 2016, Dec. 2016

DBSCAN : Density Based Spatial Clustering of Applications with Noise

The algorithm DBSCAN, based on the formal notion of density-reachability for *k*-dimensional points, is designed to discover clusters of arbitrary shape. The runtime of the algorithm is of the order O(n log n) if region queries are efficiently supported by spatial index structures, i.e. at least in moderately dimensional spaces.

Explanation of DBSCAN Steps :

- Epsilon (Eps) and Minimum points (MinPts) are the two parameters needed by DBSCAN. An unvisited point is chosen as the starting point. Eps between the starting point and its neighbors is calculated and the points within it are considered.
- If the number of data points in the neighbourhood is greater than or equal to MinPts then a cluster is formed. The starting point chosen is marked as visited.
- The above steps are then repeated for all the remaining neighbours.
- If the data points found in the neighbourhood is less than MinPts then they are marked as noise.
- If all the points within reach in a cluster are visited then the algorithm proceeds by choosing other remaining unvisited points in the dataset.

Basic concept :

For any cluster, we have :

- A central point (p) i.e. core point.
- A distance from the core point (Eps).
- Minimum number of points within the specified distance (MinPts).

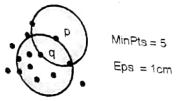
Major features :

- Discover clusters of arbitrary shape.
- Handles noise.
- One scan.
- Need density parameters as termination condition.

DBSCAN method :

- Clusters of arbitrary shape and size are grown which are dense.
- Algorithm is as follows :
  (1) Core objects and density reachable objects are found out, merge these density reachable core objects and their clusters are discovered.
  (2) When no new points can be added to any of the clusters the execution may be stopped.
- Clusters are dense regions of objects separated by regions of low density (noise).
- Outliers will not affect creation of cluster.
- **Input :**
  (1) **MinPts :** Minimum number of points in any cluster.
  (2) **ϵ :** For each point in cluster there must be another point in it less than this distance away.
- **ϵ-neighborhood :** Points within ϵ distance of a point.
- **$N\epsilon(p)$:** {q belongs to D | dist(p,q) <= ϵ }

- **Core point :** ϵ - neighborhood dense enough (MinPts)
- **Directly density-reachable :** A point p is directly density-reachable from a point q if the distance is small (ϵ) and q is a core point.
  (1) p belongs to $N\epsilon$ (q)
  (2) core point condition :
  $|N\epsilon(q)| >= MinPts$



Issues :

- One of the limitations is that, user has to set the MinPts and Eps threshold. This needs a good knowledge of the dataset. Sometimes in high dimensional data set it is difficult to decide.
- Some of the datasets distribution may be globally inconsistent. E.g. some of the area in the dataset may be too dense compared to the other areas, some of the sections may not have clusters or noise may be present.
- The process is extremely sensitive to noise which leads to very different clusters.

DBSCAN Algorithm :

- DBSCAN(D, ϵ , Minpts)
- **Input :** Database of objects D

```
Do for all O ∈ D
if O is unclassified
        call function expand_cluster(O,D, Є, Minpts)
enddo
Function expand_cluster(O,D, Є, Minpts)
Get the Є-neighbourhood of O as Nє(O)
  If | Nє(O)| < Minpts
     Mark O as noise
     Return
else
  Select a new cluster_id and mark all objects of Nє(O) with this cluster_id and put them into
candidate-objects
  Do while candidate-objects is not empty
     Select an object from candidate-objects as current object
     Delete current-object from candidate-objects
     Retrieve Nє(current-object)
     if| Nє(current-object)|≥Minpts
        Select all objects in Nє(current-object) not yet classified or marked as noise,
        Mark all of the objects with cluster_id,
```

```
    Include the unclassified objects into candidate-objects
  Enddo
return
```

4.4.2 OPTICS (Ordering Points to Identify the Clustering Structure)

- DBSCAN clusters the data objects with the help of two parameters the maximum radius of a neighborhood ϵ and minimum number of points MinPts required in the neighborhood of a core object, for this the user has to set the parameter values for clustering process.
- Setting of parameter values is also done in other clustering algorithms, these types of parameter settings are difficult to determine when it comes to real world high dimensional data.
- Real world data have skewed distribution and cannot be characterized with a single set of global density parameters.
- To overcome this limitation, clustering algorithm called as OPTICS was proposed.
- The output of OPTICS is cluster ordering, which represent density based clustering structure of data.
- Objects belonging to denser clusters are listed close to each other.
- The method does not require any density specific threshold from the user.
- The cluster ordering can be useful in extracting information like cluster centre or arbitrary shaped clusters.
- OPTICS needs two important parameters with respect to each object :

(i) Core distance :

- The core distance of an object p can be defined as the minimum distance represented as ϵ' that makes p as the core object having MinPts in the ϵ' neighborhood.
- The core distance of p is undefined if p is not a core object with respect to ϵ' and MinPts.

(ii) Reachability distance :

- It is the minimum radius from object q to object p, that makes p density reachable from object q.
- An object say p is said to be density reachable from q if q is the core object and p is in the neighborhood of q.
- Reachability Distance is given by,

  Max{core-distance(q),dist(p,q)}
- The reachability distance to p from q is undefined if q is not a core object with respect to ε and MinPts.

Example :

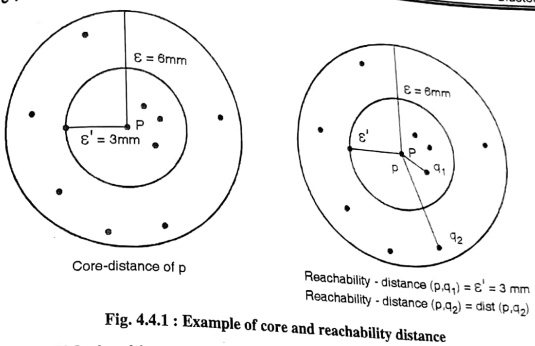


Fig. 4.4.1 : Example of core and reachability distance

Steps for OPTICS algorithm

1) An ordering of all the objects in the database is computed, for each object two parameters are calculated, core distance and reachability distance.
2) A list called as orderseeds is used to generate the output ordering.
3) Objects in this list are sorted based on reachability distance, (minimum distance is considered).
4) An arbitrary object is chosen and neighborhood of the chosen object is retrieved, core distance is calculated and reachability distance of the chosen object is set to undefined. The current object chosen is written to output.
5) If the chosen object is not a core object, then next object is chosen from the order seeds list.
6) If the chosen object is a core object, then for each object for e.g. say q in the neighborhood of the chosen object, it updates its reachability distance from the chosen object say p and inserts q in to the list incase q has not been processed.
7) This process is continued till the input is fully consumed and orders seeds list is empty.

Some point of difference between DBSCAN and OPTICS

	DBSCAN	OPTICS
Density	Boolean value (high/low)	Numerical value (core distance)
Density connected	Boolean value (yes/no)	Numerical value (reachability distance)
Searching strategy	Random	Greedy

Syllabus Topic : What is an Outlier ?

4.5 What is an Outlier ?

MU - May 2015, May 2016, Dec. 2016

An outlier is an observation or a subset of observation, which is different from the remaining set of data under analysis. It can have an unusually large or a small value compared to the rest of the data. These can be present in most of the real world data. They usually don't follow the same distribution as rest of the objects in the data.

Outliers arise due to a number of reasons a few of them include instrumental error, Human mistake, deviation in population, systems behavioral changes or system faults.

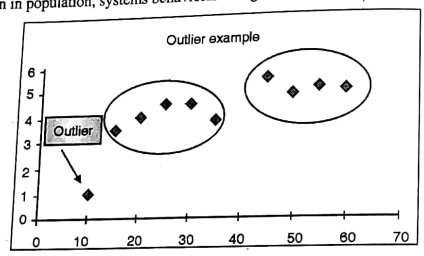


Fig. 4.5.1 : Example of outlier

The detection of these potential outliers is sometimes important for the following reasons :

1. An outlier can indicate an erroneous data, this can be due to human error or some experimental error, this type of data needs to be deleted either, from further analysis or needs to be corrected before further analysis.
2. Not always an outlier may indicate bad data, it may be due to some variation that the outlier has occurred, in such cases some statistical analysis can be used to analyze the outlier obtained.

Outliers should not be confused with noise. Noise is a random error or variance measured in a variable. In outlier detection noise should be removed first.

4.5.1 Applications

A number of applications can make use of outlier detection, as outliers can be a bad indication to further proceed with analysis or it may be considered as something interesting and can be considered for further analysis. A few of the applications are listed below :

- **Fraud detection :** Outlier mining can be used for detecting the fraud in the credit card transaction based on distance of infrequency and unconventionality in the transactional data. The fraudulent transactional attribute values are significantly different from the typical attribute values.
- **Loan application processing :** Outlier detection can be used to find fraudulent loan applications or potential problematic customers.
- **Intrusion detection :** Outlier detection can help to detect an unauthorized access into a computer network.
- **Activity monitoring :** Phone usage pattern or suspicious trading in equity can help in fraud detection in phone fraud or stock market fraud.
- **Network performance :** The computer networks performance can be monitored this can help detect network bottlenecks.
- **Fault diagnosis :** Different processes may be monitored for faults for e.g. in motors, generators, pipelines or space instruments.
- **Structural defect detection :** Manufacturing processes can also be monitored for defects for example cracked beams.
- **Satellite image analysis :** Satellite images captured can be analyzed to identify novel features or misclassified features.
- **Detecting innovations in images :** This can be used in applications like robot neotaxis or surveillance systems.
- **Segmentation of motion :** Excluding background and detecting image features.
- **Time-series monitoring :** Critical applications safety monitoring like drilling or high speed milling.
- **Medical condition monitoring :** Heart bit rate monitoring.
- **Research in pharmaceutical domain :** Identification of innovative structures of molecules.
- **Detecting unexpected entries in databases :** Detection of errors or frauds or some unexpected entries.

Syllabus Topic : Types of Outliers

4.6 Types of Outliers

MU - May 2015

Outliers can be classified as :

1. Global outliers
2. Contextual (or conditional) outliers
3. Collective outliers

4.6.1 Global Outliers

- When a Data object considerably differs from the rest of the given data set, it is considered to be a Global Outlier.
- This is considered to be one of the simplest outlier detection techniques and is a part of many outlier Detection techniques.
- Global Outlier Detection techniques analyze the relation of an individual data object with the rest of the data objects.

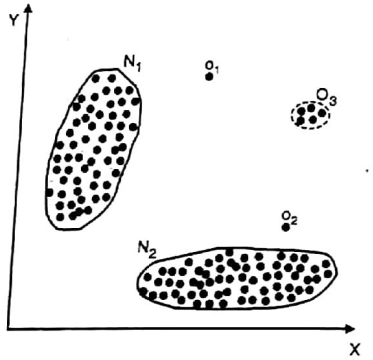


Fig. 4.6.1 : Point Anomalies

- Global Outliers are also known as Point Anomalies.
- For Example, let us consider two features for credit card transaction.
- Time of the day represented along x- axis and the amount spent represented on y-axis in the Fig. 4.6.1. o1, o2, and o3 lie outside the boundary of the normal region.

4.6.2 Contextual (or Conditional) Outliers

- A contextual (or conditional outlier) is a data object which anomalous within its context or its neighborhood.
- These are also known as conditional anomalies.
- These types of outliers are mostly seen in time series data.
- As an example, suppose it is winter season and suddenly the temperature rises to 35°C at some instance of time say t1 on a day this can be considered as a contextual outlier. The context considered being time and there is a sudden rise in the temperature on some day.

The contextual outliers has the following two properties :

1. The data object under consideration has spatial / sequential nature: The data object has two sets of attributes contextual and behavioral attributes. The contextual attributes defines the position of the object which is used to determine the context or neighborhood of the data object. For example time in time series related data. The behavioral attributes define the non-contextual characteristics of a data object. For example the rise in the temperature is a behavioral attribute.
2. The behavioral attributes within a specific context are used to determine the outlier detection. A data object may be an outlier in a particular context having some behavioral attribute where as in some other context it may be considered to be a normal data object having the same behavioral attributes.

4.6.3 Collective Outliers

- A subset of data objects when outlying with respect to the entire data set, they are called as collective outliers.
- The individual data objects in collective outliers are not outliers but their occurrence together as a substructure is considered to be an outlier.
- These types of outliers are significant only when the data object have a sequential or spatial nature.
- These are anomalous sub graphs or sub sequences occurring in the data.
- Fig. 4.6.2 shows one such example for human electrocardiogram output.

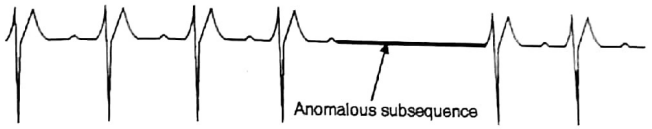


Fig. 4.6.2 : Anomalous Subsequencefor human electrocardiogram

Syllabus Topic : Challenges of Outlier Detection

4.7 Challenges of Outlier Detection

An outlier can be defined as a pattern that does not correspond to the expected normal behavior.

- Defining a representative normal region is challenging.
- The boundary between normal and outlying behaviour is often not precise. Thus an outlying observation lying close to boundary may actually be a normal data object.
- The exact notion of an outlier is different for different application domains. The set of requirements and constraints keep varying from one application to another, this makes the process of outlier detection more challenging.
- Availability of labelled data for training/validation is often difficult while developing an outlier detection technique.
- Outliers may also be a result of malicious actions, this may result in the Malicious adversaries make the outlying observations appear like normal data objects which makes the task of outlier detection more difficult.

- Data might contain noise which may be similar to the outliers making the task of detecting the actual outliers more difficult.

Syllabus Topic : Outlier Detection Methods - Supervised Methods, Semi-supervised Methods, Unsupervised Methods

4.8 Outlier Detection Methods

MU - May 2016, Dec. 2016

The process of outlier detection can also be termed as Novelty, Anomaly, Noise, Deviation detection or exception mining.

4.8.1 Supervised, Semi - Supervised, Unsupervised Methods

Supervised methods

- In Supervised methods, the training data set has instances which are labelled as normal and outlier class.
- Two predictive models can be built up for both normal as well as outlier class.
- An unseen data can then be compared against the two models to find out which class it belongs to normal or outlier.
- One of the limitations of the method is that it is very expensive to get the accurately labeled training data as the labeling method is manual and requires human efforts.
- Artificial outlier may also be injected into the normal data and supervised outlier detection techniques may be applied to detect outliers in the test data.
- In semi-supervised methods the data is labeled for only one type of class as it is difficult to get data for the other class.
- E.g. in Airline system fault detection, an outlier would be an accident which is not easy to model. In this approach the test instance which does not fit the model would be assumed to belong to the other class.
- In cases where the data is not labeled as normal or outlier, unsupervised outlier detection methods are used.
- The unsupervised technique uses a notion of outliers and then makes use of the same to detect the outliers.
- The major problem associated with unsupervised method is how good is the notion assumed?

4.8.2 Statistical Methods, Proximity-based Methods and Clustering-based Methods

Following is the classification of outlier detection methods :

Statistical approaches

- In this approach an outlier is assumed to be an irrelevant observation which is not generated by the stochastic model which is assumed.
- Thus the statistical approach is estimating the probability distribution function for the data and then testing whether the data instance is generated by that model or not.

Proximity-based approaches

- In this approach, a data point is an outlier if its proximity deviates significantly from other data points in the set.
- Proximity based outlier detection techniques can be classified into two types :
  - **Distance based :** In distance based approach, using distance as a metric, a data point is considered to be an outlier if its neighbourhood does not have enough other points within the specified distance.
  - **Density based :** In Density based approach, Local Outlier Factor (LOF) is used as parameter. A data point is considered to be an outlier if its density is relatively much lower than that of its neighbours.

Clustering based approaches

In clustering based approaches a data point which belongs either to a remote cluster or does not belong to any of the clusters is called an outlier.

Classification based approaches

- In classification based approach training data with class labels are used.
- A classification model may be trained and used to classify normal data and outliers.

Syllabus Topic : Proximity based Approaches

4.9 Proximity based Approaches

- The main idea behind proximity based approaches is to consider a point as an outlier that is far away from the rest of the data in the set.
- There are basically two types of approaches for the same, Density based approach and distance based approaches.

4.9.1 Distance-based Outlier Detection and a Nested Loop Method

- A threshold distance D could be defined for a neighborhood of an object. An object O in a dataset S is an outlier if most of the data objects in the data set S are far away from the distance D that means they do not belong to the neighborhood of the object O then it is considered to be an outlier.
- Or in other words, A DB(p, D)-outlier is an object O in a dataset T such that at least fraction p of the objects in T lies at a distance greater than distance D from O.
- A nested loop approach may be used to detect whether a point O is an outlier.

Steps for nested loop algorithm

- In Nested-loop Algorithm, the first step is to calculate the distance (D_k) between p and its k^{th} nearest neighbor.
- Maximum D_k is considered based on which data points are sorted to select the top n points.
- To compute D_k for points, each point p from the database is scanned.
- The List of k nearest points of p is maintained.
- Then a point q is considered, the distance between point p and q is calculated and checked whether the distance is smaller than the k^{th} nearest neighbor found so far.
- If the distance turns out to be smaller, then q is included in the list, if at this point the list contains more than k neighbors than the point that is furthest is deleted from the list.

The above steps results in scanning of one point at a time from the database, this would result in scanning the database N times, which results in $O(N^2)$ time complexity for the algorithm.

4.9.2 A Grid based Method

- Another Distance based outlier detection method is CELL which is grid based.
- In a Grid based method a set of grid cells are defined.
- Every individual object in the data set is assigned to the appropriate grid cell.
- The objects inside each cell form a group.
- The length of each cell can be based on l which is given by,

$$L = \frac{D}{2\sqrt{k}}$$

Where, D is the distance and k is the dimensionality

- Let us consider a cell c in a grid like structure, the neighboring area around c can be divided into two types of groups :
  - **Layer-1 neighbors** – These are the intermediate neighbor cells. Considering a point p of c and another point q, then the dist(p, q) ≤ D, then point q is said to be in Layer 1.

  - **Layer-2 neighbors** – Considering a point p of c and another point q , then the dist(p, q) ≥ D, then point q is said to be in Layer 2.
- To search for an outlier we can apply the following steps :
  - A cell can be searched internally. If the numbers of object present in the cell are M then they are not considered as outliers.
  - A search is conducted for Layer-1 neighbors, if at this level there are M objects inside the cell then all these are also not considered as outliers.
  - Then layer-2 neighbors are considered, if the numbers of objects are less then M, then all the neighbor cells in Layer-1 and Layer -2 are outliers.
  - Else the objects in the cell could be an outlier, which can found out by calculating the distance between the objects in this cell and the objects in the layer 2 neighbor cells. The total distance D is considered and checked whether it is more than M.
- **For example :**

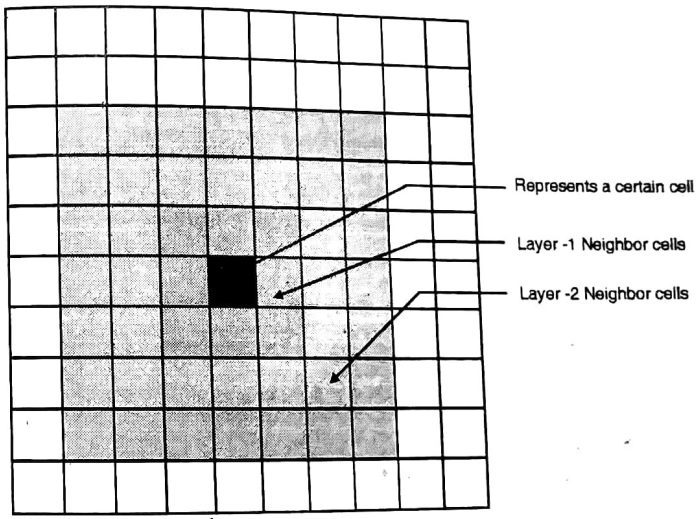


Fig. 4.9.1 : Grid based structure

The Above grid like structure represent the above discussed cell based method for outlier detection :

- **Green Cell :** Represents a certain cell
- **Red cells :** Layer -1 Neighbor cells
- **Orange cells :** Layer-2 Neighbor cells

4.9.3 Density based Outlier Detection

- In Distance based outlier detection method, a global view of the dataset is considered for outlier detection.
- In real world, data sets can have a more complex structure due to which distance based outlier may not be suitable rather outliers may be detected considering their local neighborhood.
- In density based outlier detection techniques, density of an object is considered to find out whether the object can be declared as outlier or not.
- Local Outlier Factor (LOF) of an object is the minimum number of nearest neighbors used for defining its local neighborhood.

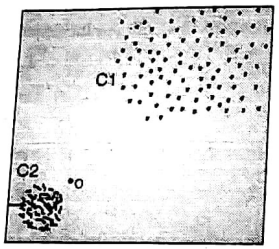


Fig. 4.9.2 : Advantage of LOF over Distance based method for Outlier Detection

In the above Fig. 4.9.2 it can be observed that outlier o cannot be detected using distance based method whatsoever threshold may be set. In such cases, density based outlier detection play an important role.

Let us define a few terms which can useful in defining the LOF.

- k-Distance of an Object o.
- Consider a Data set S, the k-distance of p, which is denoted as $dist_k(o)$ is defined as the distance between o and its k^{th} Nearest Neighbor.
- k-distance neighborhood of o.

It is denoted as $N_k(o)$ is given by,

$$\{o' | o' \text{ in } S, dist(o, o') \leq dist_k(o)\}$$

- Reachability Distance from o' to o.

$$\text{reach } dist_k(o' \to o) = \max\{dist_k(o), dist(o,o')\}$$

- Local Reachability distance of o.

$$ird_k(0) = \frac{\| N_k(0) \|}{\sum 0' \in N_k(0) \text{ reach dist}_k(0' \leftarrow 0)}$$

- Considering all the above terms we can now define the Local Outlier Factor(LOF) as

$$LOF_k(0) = \frac{\Sigma 0' \in N_k(0) \frac{lrd_k(0')}{lrd_k(0)}}{\| N_k(0) \|}$$

$$= \sum_{0' \in N_k(0)} (lrd_k(0') \sum_{0' \in N_k(0)} \text{reach dist}_k(0' \leftarrow 0)$$

- From the above formula it can be seen that LOF is the average ratio of Local reachability of o and o's k- NN.
- For a High LOF, the local reachability density of p should be low and local reachability density of k-NN of o should be high.

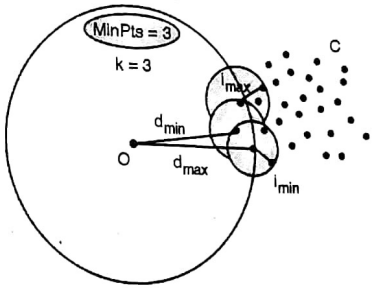


Fig. 4.9.3 : An example LOF

Syllabus Topic : Clustering based Approaches

4.10 Clustering based Approaches

- Clustering is an unsupervised technique, although semi-supervised techniques are also been explored.
- Clustering is a machine learning technique which is used to group the similar data items together into a cluster.

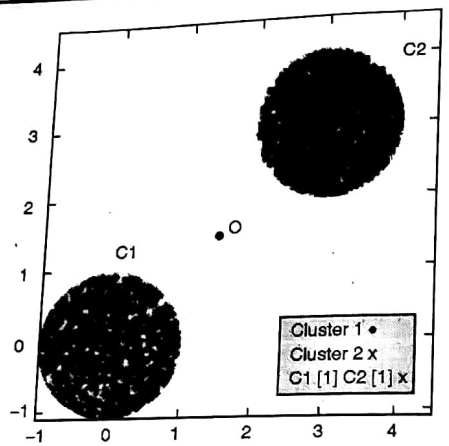


Fig. 4.10.1 : Clustering based outlier Detection

- It can be either used as a standalone tool or as a further analysis in some other techniques as a preprocessing step which is used to detect clusters.
- In this approach the outliers will be a very few number of data instances which may not belong to any of the clusters formed or they may themselves be a small cluster whose properties may not be similar to those clusters who have normal instances.
- Clustering based outlier detection can be broadly classified as follows :

If a Data instance does not belong to any of the clusters then it is an outlier. Fig. 4.10.2 shows a point which does not belong to any cluster. Using density based clustering method it can be seen that two clusters are formed and a single point o does not belongs to either of them thus it may be declared as an outlier.

- If the distance is large from a data instance to a cluster to which it is closest then it is an outlier.

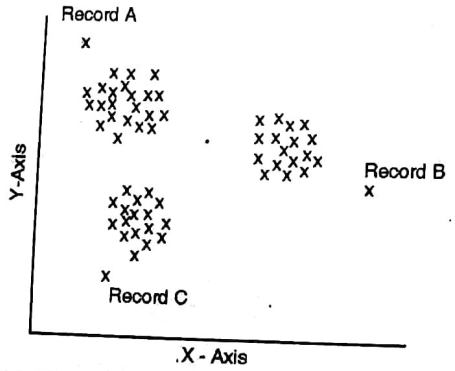


Fig. 4.10.2 : Graphical presentation of clustering based outlier

Using K means clustering, we can partition the above data set to form clusters. However from Fig. 4.10.2, we can note that Record A, B and C are far from the clusters to which they are closest hence can be considered to be outliers.

- A set of data instances form a small cluster and have properties which are not similar to other clusters then it is an outlier.

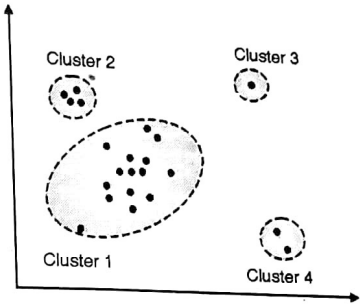


Fig. 4.10.3 : Graphical presentation of clustering based outlier

In the Fig. 4.10.3 as we can see the cluster 2, cluster 3 and cluster 4 have properties which are not similar to cluster 1, hence these can be possible outliers.

Cluster based local outlier factor :

- Use a suitable clustering method (e.g. K means) to cluster the data set.
- Sort these clusters based on decreasing size.
- A parameter α (range between 0 to 1) is used to distinguish Large Clusters (LC) from small clusters (SC).
- If a cluster contains at least a percentage for e.g. α=90% then that cluster is considered to be large.
- The remaining is considered to be small clusters.
- CBLOF is calculated as given below :

$$CBLOF(p) = \begin{cases} |C_i| \cdot \min(d(p, C_j)) \text{ if } C_i \in SC \text{ where } p \in C_i \text{ and } C_j \in LC \\ |C_i| \cdot \min(d(p, C_i)) \text{ if } C_i \in LC \text{ where } p \in C_i \end{cases}$$

- This parameter can be used to detect the outliers.
- Any small cluster that is far away from the large cluster is considered to be an outlier.
- Lower value of CBLOF is considered as a parameter for suspected outliers.

Advantages of clustering based outlier detection

- This approach does not require any kind of supervision.
- This approach can be used for incremental mode, which is used for anomaly detection from temporal data.

Disadvantages of clustering based outlier detection

- This approach is computationally costly.
- The technique may fail if the normal points do not form any clusters.
- Data is sparse in high dimensional spaces due to which distance between two records may be very less.

Review Questions

Q. 1 Explain the different categories of clustering methods.

Q. 2 State the difference between clustering and classification.

Q. 3 Write Short Notes on : (a) BIRCH (b) OPTICS

Q. 4 Explain Clustering based Approaches.

Q. 5 Explain various outlier detection methods.

4.11 University Questions and Answers

May 2015

Q. 1 Explain the different distance measures that can be used to compute distances between two clusters. *(Ans. : Refer section 4.1.3)* **(5 Marks)**

Q. 2 Use K-means to cluster the following data set into 3 clusters.

Protein	20	21	15	22	20	25	26	20	18	20
Fat	9	9	7	17	8	12	14	9	9	9

(Ans. : Refer Ex. 4.2.6) **(10 Marks)**

Q. 3 Explain : DBSCAN clustering algorithm with an example.
(Ans. : Refer section 4.4.1) **(10 Marks)**

Q. 4 Define "Outlier". What are the different types of Outliers that occur in a dataset ?
(Ans. : Refer sections 4.5 and 4.6) **(5 Marks)**

Dec. 2015

Q. 5 Explain BIRCH algorithm with example.
(Ans. : Refer section 4.3.3) **(10 Marks)**

Q. 6 What is clustering ? Explain k-means clustering algorithm. Suppose the data for clustering – {2, 4, 10, 12, 3, 20, 11, 25}
Consider k = 2, cluster the given data using above algorithm.
(Ans. : Refer sections 4.1.1, 4.2.1 and Ex. 4.2.1) **(10 Marks)**

May 2016

Q. 7 What is hierarchical clustering? Explain any two techniques for finding distance between the clusters in hierarchical clustering.
(Ans. : Refer section 4.3) **(5 Marks)**

Q. 8 Use any hierarchical clustering algorithm to cluster the following 8 examples into 3 clusters :
A1 = (2, 10), A2 = (2, 5), A3 = (8, 4), A4 = (5, 8),
A5 = (7, 5), A6 = (6, 4), A7 = (1, 2), A8 = (4, 9)
(Ans. : Refer Ex. 4.3.7) **(10 Marks)**

Q. 9 Clearly explain the working of the DB_SCAN algorithm using appropriate diagrams.
(Ans. : Refer section 4.4.1) **(10 Marks)**

Q. 10 What is an outlier? Describe methods that can be used for outlier analysis.
(Ans. : Refer sections 4.5 and 4.8) **(10 Marks)**

Dec. 2016

Q. 11 Describe one hierarchical clustering algorithm using an example dendrogram.
(Ans. : Refer section 4.3) **(5 Marks)**

Q. 12 What is clustering? Explain k-means clustering algorithm. Suppose the data for clustering is {2, 4, 10, 12, 3, 20, 11, 25}
Consider K = 2, cluster the given data using K-Means algorithm
(Ans. : Refer sections 4.1.1, 4.2.1 and Ex. 4.2.1) **(10 Marks)**

Q. 13 Clearly explain the working of the DBSCAN algorithm using appropriate diagrams.
(Ans. : Refer section 4.4.1) **(10 Marks)**

Q. 14 What is an outlier ? Describe methods that can be sued for outlier analysis.
(Ans. : Refer sections 4.5 and 4.8) **(10 Marks)**

...Chapter Ends

❑❑❑

CHAPTER 5

Frequent Pattern Mining

Syllabus

Market Basket Analysis, Frequent Itemsets, Closed Itemsets, and Association Rules;Frequent Pattern Mining, Efficient and Scalable Frequent Itemset Mining Methods, The Apriori Algorithm for finding Frequent Itemsets Using Candidate Generation, Generating Association Rules from Frequent Itemsets, Improving the Efficiency of Apriori, A pattern growth approach for mining Frequent Itemsets; Mining Frequent itemsets using vertical data formats; Mining closed and maximal patterns; Introduction to Mining Multilevel Association Rules and Multidimensional Association Rules; From Association Mining to Correlation Analysis, Pattern Evaluation Measures; Introduction to Constraint-based Association Mining.

Syllabus Topic : Market Basket Analysis

5.1 Market Basket Analysis

5.1.1 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.
- The market-basket problem assumes we 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 algorithms 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.

5.1.2 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 items 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.

5.1.3 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 services/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 result in customers getting tempted to buy one product with the other.
  - The layout of catalog of an ecommerce site may be defined.
  - Inventory may be managed based on product demands.

Syllabus Topic : Frequent Itemsets, Closed Itemsets and Association Rules

5.2 Frequent Itemsets, Closed Itemsets and Association Rules

5.2.1 Frequent Itemsets

- An itemset X is *frequent* if X's support is no less than a *minimum support* threshold.
- A frequent itemset is a set of items that appears at least in a pre-specified number of transactions. Frequent itemsets are typically used to generate association rules.
- Consider a data set S, frequent itemset in S are those items that appear in at least a fraction s of the basket, where s is a chosen constant with a value of 0.01 or 1%.
- To find frequent itemsets one can use the monotonicity principle or a-priori trick which is given as,

  If a set of items say S is frequent then all its subsets are also frequent.
- The procedure to find frequent itemsets :
  - A level wise search may be conducted to find the frequent-1 items(set of size 1), then proceed to find frequent -2 items and so on.
  - Next search all maximal frequent itemsets.

5.2.2 Closed Itemsets

- An itemset is closed if none of its immediate supersets has the same support as the itemset.
- Consider two itemsets X and Y, if every item of X is in Y but there is at least one item of Y, which is not in X, then Y is not a proper super-itemset of X. In this case, itemset X is closed.
- If X is both closed and frequent, it is known as closed frequent itemset.
- An itemset is maximal frequent if none of its immediate supersets is frequent.
- An itemset X is *maximal frequent itemset or max-itemset* if X is frequent and there exist no super itemset Y such that X is subset of Y and Y is frequent.

Example

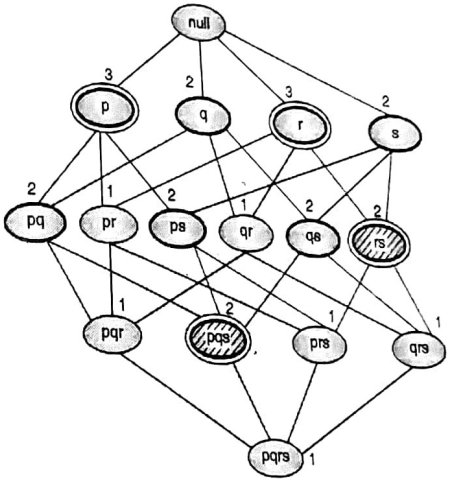


Fig.5.2.1 : Lattice diagram for maximal, closed and frequent itemsets

Let us consider minimum support =2.

- The itemsets that are circled with thick lines are the frequent itemsets as they satisfy the minimum support. From Fig.5.2.1 Frequent itemsets are { p, q, r, s, pq, ps, qs, rs, pqs}
- The itemsets that are circled with the double lines are closed frequent itemsets. From Fig. 5.2.1, closed frequent itemsets are { p,r,rs,pqs}. For example {rs} is closed frequent itemset as all of its superset {prs, qrs} have support less than 2.
- The itemsets that are circled with the double lines and shaded are maximal frequent itemsets. From Fig.5.2.1, maximal frequent itemsets are {rs,pqs}. For example, {rs} is maximal frequent itemset as none of its immediate supersets like {prs, qrs} is frequent.

5.2.3 Association Rules

- The items or objects in Relational databases, transactional databases or other information repositories are considered for finding frequent patterns, associations, correlations, or causal 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 commonly purchased together. This knowledge can be used in advertising or in goods placement in stores.
- Association rules have the general form

  $$I_1 \rightarrow I_2 \text{ (where } I_1 \cap I_2 = 0)$$

  Where, I_n are sets of items, for example can be purchased in a store.

- The rule should be read as "Given that someone has bought the items in the set I_1 they are likely to also buy the items in the set I_2".

5.2.3(A) Large Itemsets

- An itemset is a set of single items from the database of transactions.
- If some items often occur together they can form an association rule.
- **Support**
  - The support of an itemset is the count of that itemset in the total number of transactions, or in other words it is the percentage of the transactions in which the items appear.

    If A =>B

$$\text{Support } (A \Rightarrow B) = \frac{\#\_\text{tuples\_containing\_both\_A\_and\_B}}{\text{total\_\#\_of\_tuples}}$$

  - The support(s) for an association rule X =>Y is the percentage of transactions in the database that contains X U Y.
  - An itemset is considered to be a *large itemset* if its support is above some threshold.
- **Confidence**
  - The confidence or strength for an association rule A=> B is the ratio of the number of transactions that contain A U B to the number of transactions that contain A.
  - Consider a rule A => B, it is measure of ratio of the number of tuples containing both A and B to the number of tuples containing A

$$\text{Confidence } (A \Rightarrow B) = \frac{\#\_\text{tuples\_containing\_both\_A\_and\_B}}{\#\_\text{tuples\_containing\_A}}$$

Finding the large itemsets

1. **The Brute Force approach**
   - Find all the possible association rules.
   - Calculate the support and confidence for each rule generated in the above step.
   - The Rules that fail the minsup and minconf are prunned from the above list.
   - The above steps would be a time consuming process, we can have a better approach as given below.
2. **A better approach :** The Apriori Algorithm.

Syllabus Topic : Frequent Pattern Mining

5.3 Frequent Pattern Mining

Frequent pattern mining is classified in the various ways based on following criteria :

1. **Completeness of the pattern to be mined :** Here we can mine the complete set of frequent itemset, closed frequent itemset, constrained frequent itemsets.

2. **Levels of abstraction involved in the rule set :** Here we use multilevel association rules based on the levels of abstraction of data.
3. **Number of data dimensions involved in the rule :** Here we use single dimensional association rule, there is only one dimension or multidimensional association rule if there is more than one dimension.
4. **Types of the values handled in the rule :** Here we use Boolean and quantitative association rules.
5. **Kinds of the rules to be mined :** Here we use association rules and correlation rules based on the kinds of the rules to be mined.
6. **Kinds of pattern to be mined :** Here we use frequent itemset mining, sequential pattern mining and structured pattern mining.

Syllabus Topic : Efficient and Scalable Frequent Itemset Mining Method

5.4 Efficient and Scalable Frequent Itemset Mining Method

1. Apriori Algorithm
2. FP Tree

Syllabus Topic : Apriori Algorithm for Finding Frequent Itemsets using Candidate Generation, Generating Association Rules from Frequent Itemsets

5.4.1 Apriori Algorithm for Finding Frequent Itemsets using Candidate Generation

- The Apriori Algorithm solves the frequent item sets problem.
- The algorithm analyzes a data set to determine which combinations of items occur together frequently.
- The Apriori algorithm is at the core of various algorithms for data mining problems. The best known problem is finding the association rules that hold in a basket - item relation.

Basic Idea

- An itemset can only be a large itemset if all its subsets are large itemsets.
- Frequent itemsets: The sets of items that have minimum support.
- All the subsets of a frequent itemset must be frequent for e.g. {PQ} is a frequent itemset {P} and {Q} must also be frequent.
- Find frequent itemsets frequently with cardinality 1 to k(k-itemset).
- Generate association rules from frequent itemsets.

Apriori Algorithm given by Jiawei Han et al.

Input :
D, a database of transactions;
min_sup, the minimum support count threshold.
Output : L, frequent itemsets in D.
Method :

```
(1)  L1 = find_frequent_1-itemsets(D);
(2)  for (k = 2; Lk-1 ≠ φ; k++){
(3)      Ck = apriori_gen (Lk-1);
(4)      for each transaction t ∈ D { // scan D for counts
(5)          Ct = subset (Ck, t); // get the subsets of t that are candidates
(6)          for each candidate c ∈ Ct
(7)              c.count ++ ;
(8)      }
(9)      Lk = {c ∈ Ck| c.count≥ min_sup}
(10) }
(11) return L = Uk Lk;
Procedure apriori_gen (Lk-1:frequent (k – 1) - itemsets)
(1)  for each itemset l1∈Lk-1
(2)  for each itemset l2∈Lk-1
(3)      if (l1 (1) = l2 (l2) ∧ (l1 (2) = l2 (2))
         ∧...∧ (l1 [k – 2] = l2 [k – 2]) ∧ (l1 [k – 1] <l2 [k – 1]) then {
(4)          c = l1⋈l2; // join step:generate candidates
(5)          if has_infrequent_subset(c, Lk-1) then
(6)              delete c; // prune step: remove unfruitful candidate
(7)          else add c to Ck
(8)      }
(9)  return Ck;
Procedure has_infrequent_subset(c: candidate k-intemset;
         Lk-1: frequent (k – 1) –itemsets); // use prior knowledge
(1)  for each (k – 1)-subset s of c
(2)      if s ∉ Lk-1 then
(3)          return TRUE;
(4)  return FALSE;
```


5.4.2 Advantages and Disadvantages of Apriori Algorithm

Some of the advantages and disadvantages of Apriori Algorithm are listed as :

Advantages

1. The algorithm makes use of large itemset property.
2. The method can be easily parallelized.
3. The algorithm is easy from implementation point of view.

Disadvantages

1. Although the algorithm is easy to implement it needs many database scans which reduces the overall performance.
2. Due to Database scans, the algorithm assumes transaction database is memory resident.

5.4.3 Solved Examples on Apriori Algorithm

Ex. 5.4.1 : Given the following data, apply the Apriori algorithm. Min support = 50 % Database D.

TID	Items
100	1 3 4
200	2 3 5
300	1 2 3 5
400	2 5

Soln. :

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	Sup.
{1}	2
{2}	3
{3}	3
{4}	1
{5}	3

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

L_1=

Itemset	Sup.
{1}	2
{2}	3
{3}	3
{5}	3

Step 3 : Generate candidate C_2 from L_1

C_2 =

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

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

C_2 =

Itemset	Sup.
{1 2}	1
{1 3}	2
{1 5}	1
{2 3}	2
{2 5}	3
{3 5}	2

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

L_2=

Itemset	Sup.
{1 3}	2
{2 3}	2
{2 5}	3
{3 5}	2

Step 6 : generate candidate C_3 from L_2

C_3 =

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

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

C_3 =

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

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

L_3 =

Itemset	sup
{2,3,5}	2

Step 9 : So data contain the frequent itemset(2,3,5)

Therefore the association rule 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%

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 :

Rule 1: 2^3=>5 and Rule 2 : 3^5=>2

Ex. 5.4.2 : 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

Soln. :

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=

Items	Sup.
{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=

Items	Sup.
{A}	3
{B}	2
{C}	2

Step 3: Generate candidate C_2 from L_1

C_2=

Items
{A,B}
{A,C}
{B,C}

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

C_2=

Items	Sup.
{A,B}	1
{A,C}	2
{B,C}	1

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

L_2=

Items	Sup.
{A,C}	2

Step 6 : So data contain the frequent item l(A,C)

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

Ex. 5.4.3 : 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

Soln. :

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

C_1=

1-Itemsets	Sup-count
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 given is 2).

L_1=

1-Itemsets	Sup-count
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-itemsets.

C_2=

2-Itemsets	Sup-count
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 =

Frequent 2-Itemsets	Sup-count
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 =

Frequent 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 =

Frequent 3-Itemset	Sup-count
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 =

Frequent 3-Itemset	Sup-count
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 itemsets ={I1, I2, I5}. Following are the Association rules that can be generated shown below with the support and confidence.

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

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

Rule	Confidence
I1^I5=>I2	100%
I2^I5=>I1	100%
I5=>I1^I2	100%

Ex. 5.4.4 : 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.

Soln. :

Step 1 : Scan the transaction Database D and find the count for item-1 set 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	Sup-count
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 transactions (As support count given is 30%).

L_1 =

Itemset	Sup-count
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	Sup-count
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	Sup-count
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	Sup-count
1,2,3	1
2,3,5	2
1,3,4	1

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

L_3 =

Itemset	Sup-count
2,3,5	2

Therefore the database contains the frequent itemset{2,3,5}.

Following are the association rules that can be generated from L3 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%

Given 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 :

Rule 1: 2^3=>5 and Rule 2 : 3^5=>2

Ex. 5.4.5 : 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).

Soln. :

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	Sup-count
A	4
B	4
C	2
D	3
E	2
K	1

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

L_1=

Itemset	Sup-count
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	Sup-count
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	Sup-count
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	Sup
A,B,D	3

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

L_3 =

Itemset	Sup
A,B,D	3

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

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

Association Rule	Support	Confidence	Confidence %
A^B=>D	3	3/4=0.75	75%
A^D=>B	3	3/3=1	100%
B^D=>A	3	3/3=1	100%
A=>B^D	3	3/4=0.75	75%
B=>A^D	3	3/4=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 these are the only ones generated that are strong.

Ex.5.4.6 : 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

TID	List of item_IDs
T800	I1,I2,I3
T900	I2,I3
T1000	I3,I5

Soln. :

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

C_1=

1-Itemsets	Sup-count
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=

1-Itemsets	Sup-count
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=

2-Itemsets	Sup-count
1,2	4
1,3	4
1,4	1
1,5	3
2,3	4
2,4	2
2,5	2
3,4	0

2-Itemsets	Sup-count
3,5	3
4,5	0

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

L_2 =

2-Itemsets	Sup-count
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 =

Frequent 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 =

Frequent 3-Itemset	Sup-count
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 =

Frequent 3-Itemset	Sup-count
1,2,3	2
1,2,5	2
1,3,5	2

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

Let us assume that the data contains the frequent itemset ={I1,I2,I5} then the association rules that can be generated from frequent itemset are as shown below with the support and confidence.

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

If the minimum confidence threshold is 70% (Given), then only the SECOND,THIRD AND 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}.

Ex. 5.4.7 : A Database has four transactions. Let Minimum support and confidence be 50%.

T_{id}	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

Soln. :

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	Sup-count
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	Sup-count
1	6
2	5
3	7
5	6

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

C_2 =

Itemset	Sup-count
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	Sup-count
1,3	5

So data contain the frequent itemset = {1,5}.

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

Final rules are :

Rule 1: 1=>3 and Rule 2 : 3=>1

Ex. 5.4.8 : Use the Apriori algorithm to identify the frequent item-sets in the following database. Then extract the strong association rules from these sets. Minimum support - 30%, Minimum confidence = 75%. MU - May 2016, Dec. 2016

TID	Items
01	A, B, D, E, F
02	B, C, E,
04	A, B, D, E,
04	A, B, C, E,
05	A, B, C, D, E, F
06	B, C, D
07	A,B, D, E

Soln.:

Step I : Generate single item set

Items	Support
A	5
B	7
C	4
D	5
E	6
F	2

Item set above 30 % support	
A	5
B	7
C	4
D	5
E	6

Step II : Generate 2 item set

Item	Support
AB	5
AC	2
AD	4
AE	5
BC	4
BD	4
BE	6

Item set above 30 % support	
AB	5
AD	4
AE	5
BC	4
BD	4
BE	6
DE	4

Item	Support
CE	3
DE	4

Item set above 30 % support

Step III : Generate 3 item set

Item sets of 3 items

Item set	Support
ABD	4
ABC	2
ABE	5
ADE	4
BCD	2
BCE	2
BDE	4

Item set above 30 % support

Item set	Support
ABD	4
ABE	5
ADE	4
BDE	4

Step IV : Generate 4 item set

Item set	Support
ABDE	4

Therefore ABDE is the large item set with minimum support 30%.

Following Rules generated

Rule	Confidence	Confidence %
A → BED	4/5=0.8	80%
B → AED	4/7=0.57	57%
E → ABD	4/5=0.8	80%
D → ABE	4/6=0.66	66%
AB → ED	4/5=0.8	80%

Rule	Confidence	Confidence %
BE → AD	4/6=0.66	66%
ED → AB	4/4=1	100%
AE → BD	4/5=0.8	80%
AD → BE	4/4=1	100%
BED → A	4/4=1	100%
AED → B	4/4=1	100%
ABD → E	4/4=1	100%
ABE → D	4/5=0.8	80%

From the above Rules generated, only the rules having greater than 75% are considered as final rules. So final Rules are,

Rule
A → BED
E → ABD
AB → ED
ED → AB
AE → BD
AD → BE
BED → A
AED → B
ABD → E
ABE → D

Ex. 5.4.9 : Consider the following transaction database :
Apply the Apriori algorithm with minimum support of 30% and minimum confidence of 70%, and find all the association rules in the data set. MU May 2015

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, G, 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

Soln. :

Step I : Generate single item set :

Items	Support
A	6
B	7
C	6
D	7
E	6
F	3
G	2
H	3
I	1
K	4
L	4

Item set above 30 % support	
A	6
B	7
C	6
D	7
E	6
F	3
H	3
K	4
L	4

Step II : Generate 2 item set :

Item	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

Item	Support
CH	1
CK	2
CL	1
DE	4
DF	2
DH	1
DK	2
DL	2
EF	2
EH	2
EK	3
EL	3
FH	2
FK	0

Item set 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

Item	Support
BL	2
CD	5
CE	2
CF	1

Item	Support
FL	2
HK	1
HL	2
KL	1

Item set above 30 % support

Step III : Generate 3 item set :

Item sets of 3 items

Item set	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

Item set above 30 % support

Item set	Support
ABC	3
ABD	4
ACD	3
BCD	5
BDE	3

Step IV : Generate 4 item set

Item set	Support
ABCD	3
ABDE	2
BCDE	2

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

Following Rules generated

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	3/4 = 0.75	75%
BC → AD	3/5 = 0.6	60%
CD → AB	3/5 = 0.6	60%
AC → BD	3/4 = 0.75	75%
AD → BC	3/4 = 0.75	75%
BCD → A	3/5 = 0.6	60%
ACD → B	3/3 = 1	100%
ABD → C	3/4 = 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

Syllabus Topic : Improving the Efficiency of Apriori

5.4.4 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 itemset counting :** The itemsets can be hashed into corresponding buckets, For a particular iteration a k-itemset 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 itemset 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 itemsets. 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 itemsets are found out in each partition. The local frequent itemsets may or may not be frequent with respect to the entire database however a frequent itemset from database has to be frequent in atleast one of the partitions. All the frequent itemsets with respect to each partition forms the global candidate itemsets. In the second phase of the algorithm, a second scan of database for actual support of each item is found, these are global frequent itemsets.
- **Sampling :** Rather than finding the frequent itemsets in the entire database D, a subset of transactions are picked up and searched for frequent itemsets. A lower threshold of minimum support is considered as this reduces the possibility of missing the actual frequent itemset due to a higher support count
- **Dynamic itemset 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 itemset is added to the frequent itemset collection which can be further used to generate longer candidate itemset.

Syllabus Topic : A Pattern Growth Approach for Mining Frequent Itemsets

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

5.5.1 Definition of FP-tree

An FP-tree is a 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 the 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.

5.5.2 FP-Tree Algorithm

FP-tree construction algorithm given by Jiawei Han et al.

Algorithm : F.P. growth, Mine frequent itemsets 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 :

A FP tree is constructed in the following steps :

1. (a) Scan the transaction database D once, Collect F, the set of frequent items, and their support counts. Sort F by support count in 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 combination (denoted as β) of the nodes in the path P
(3) generate pattern β∪ α with support_count = minimum support count of nodes in β;
(4) else for each a_i in the header of Tree {
(5) generate pattern β = a_i∪ αwith support_count = a_i.support_count;
(6) construct β's conditional pattern base and then β's conditional FP_tree Tree_β;
(7) if Tree_β≠ϕ then
(8) call FP_growth (Tree_β, β) ; }
```

5.5.3 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 size is dependent on the order of the items. Ordering of items by decreasing support will not always result in a smaller FP -Tree size (it's heuristic).

5.5.4 Example of FP Tree

Ex. 5.5.1 : 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

Soln. :

Step 1: Find the minimum support of each item.

Item	Sup.
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	Sup.
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)

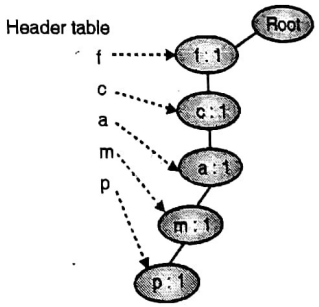
TID	Items Bought	(Ordered frequent items)
1	f, a, c, d, g, i, m, p	f, c, a, m, p
2	a, b, c, f, l, m, o	f, c, a, b, m
3	b, f, h, j, o	f, b
4	b, c, k, s, p	c, b, p
5	a, f, c, e, l, p, m, n	f, c, a, m, p

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

Step 3 : FP Tree construction

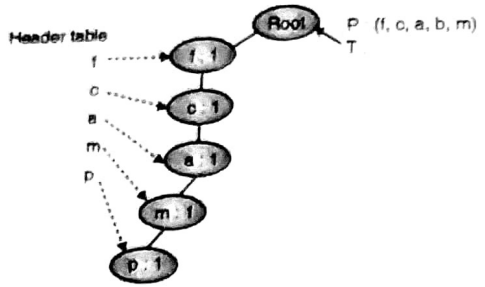
Originally Empty

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

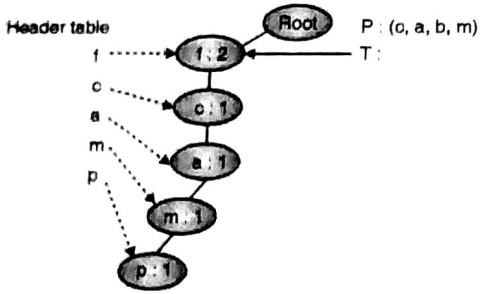


Step 5 : Start the insertion of Second transaction (f,c,a,b,m)

(i) The transaction T is pointing to the root node,

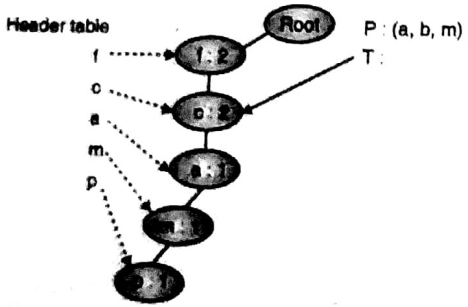


(ii) Consider the first item in the second transaction i.e. f and add it in the tree.

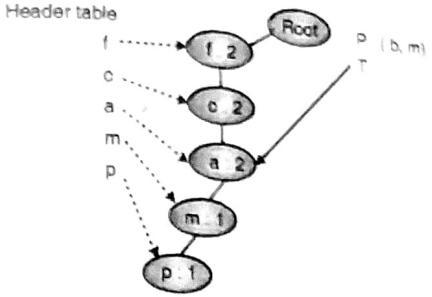


After this step we get f:2, finished adding f in the above tree.

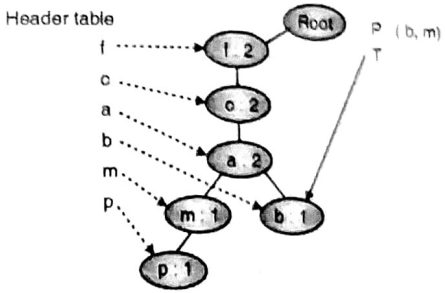
(iii) Now consider the second item in the above transaction .i.e. c.



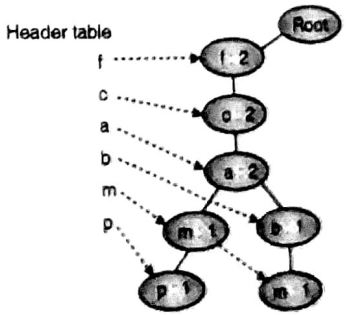
(iv) Similarly consider the next item a.



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

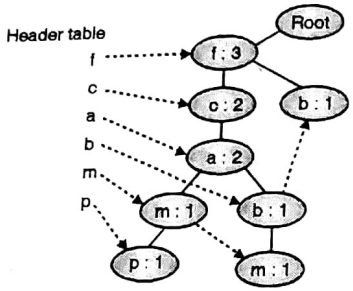


(vi) Now only m of second transaction is left. Though a node m is already exists still we can't increase its count of the existing node m as we need to represent the second transaction in FP tree, so add new node m below node b and link it with existing node m.

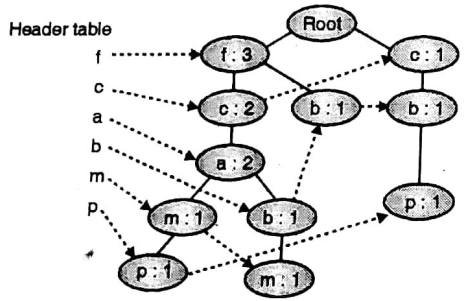


Second transaction is complete.

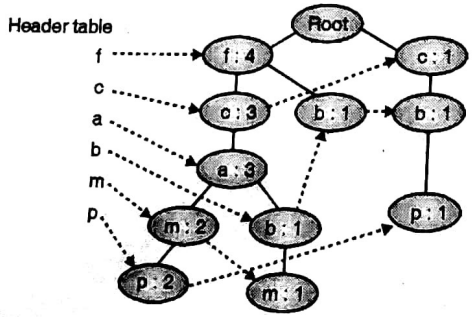
Step 6 : Similarly insert the third transaction(f, b) as explained in step 5. So After the insertion of third transaction (f, b)



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.

5.5.5 Mining Frequent Patterns from FP Tree

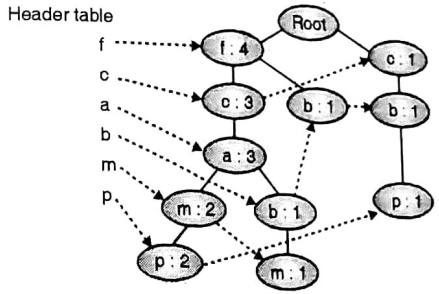
General idea (divide-and-conquer)

- Use the FP Tree and recursively grow frequent pattern path.

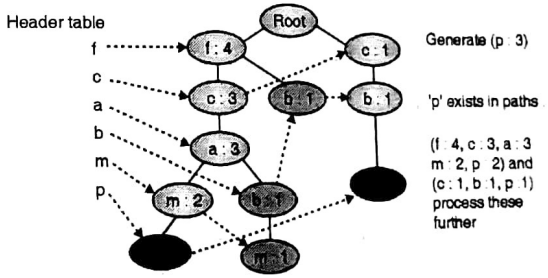
Method:

- For each item, conditional pattern-base is constructed, and then it'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).

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



- Starting from the bottom of the header table.



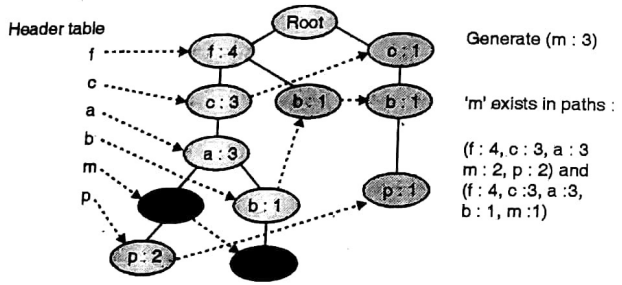
- Following are the paths with 'P'
  - We got (f:4, c:3, a:3, m:2, p:2) and (c:1, b:1, p:1)
  - The transactions containing 'p' have p.count
  - Therefore we have (f:2, c:2, a:2, m:2, p:2) and (c:1, b:1, p:1)
  - Since 'p' is part of these we can remove 'p'
- Conditional Pattern Base (CPB)
  - After 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.

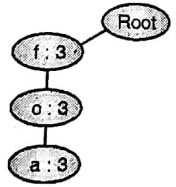
- Finding all patterns with 'P'.
  - We again filter away all items < minimum support threshold (i.e. 3)
  - (f:2, c:2, a:2, m:2), (c:1, b:1) => (c:3)
  - We generate (cp:3) (Note : we are finding frequent patterns containing item p, so we append p to c as c is only item that has min support threshold.)
  - Support value is taken from the sub-tree
  - Frequent patterns thus far: (p:3, cp:3)

Example : Finding Patterns with 'm' but not 'p'.

- Find 'm' from the header table



- Conditional Pattern Base:
  - Path1 : (f:4, c:3, 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.

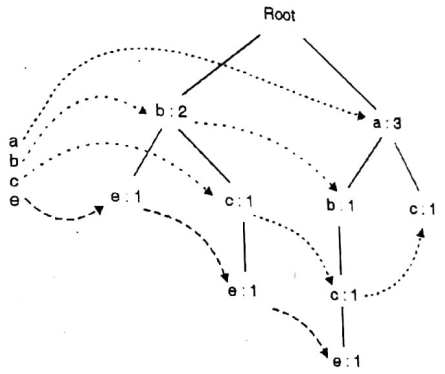
Item	Conditional pattern-base	Conditional FP-tree
P	{(fcam:2), (cb:1)}	{(c:3)}|p
M	{(fca:2), (fcab:1)}	{(f:3, c:3, a:3)}|m
B	{(fca:1), (f:1), (c:1)}	Empty

Item	Conditional pattern-base	Conditional FP-tree
A	{(fc:3)}	{(f:3, c:3)}|a
C	{(f:3)}	{(f:3)}|c
f	Empty	Empty

Ex. 5.5.2 : Transaction item list is given below. Draw FP tree.

T1 = b, e
T2 = a, b, c, e
T3 = b, c, e
T4 = a, c
T5 = a
Given : minimum support = 2

Soln. :

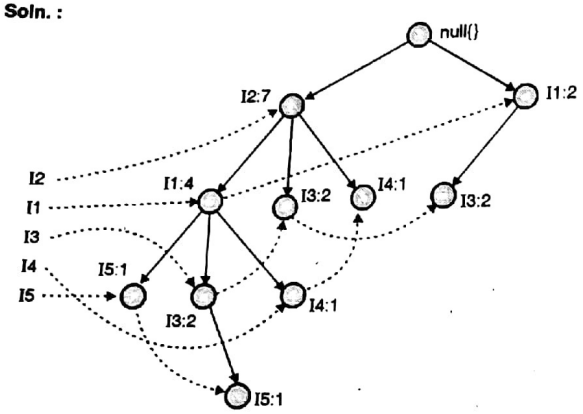


Ex. 5.5.3 : Transaction database is

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

Min support = 2

Soln. :



Item ID	Support Count
I2	7
I1	6
I3	6
I4	2
I5	2

Mining the FP-Tree by creating conditional (sub) pattern bases.

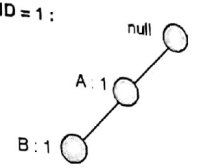
Item	Conditional pattern base	Conditional FP-tree	Frequent patterns generated
15	{(I2 I1 : 1),(I2 I1 I3 : 1) }	{I2 : 2, I1 : 2}	I2 I5 : 2, I1 I5 : 2, I2 I1 I5 : 2
14	{(I2 I1 : 1),(I2 : 1)}	{I2 : 2}	I2 I4 : 2
13	{(I2 I1 : 2),(I2 : 2), (I1 : 2) }	(I2 : 4, I1 : 2), (I1 : 2)	I2 I3 : 4, I1, I3 : 2, I2 I1 I3 : 2
11	{(I2 : 4)}	{(I2 : 4)}	I2 I1 : 4

Ex. 5.5.4 : Consider the following dataset of frequent itemsets. All are sorted according to their support count. Construct the FP-Tree and find Conditional Pattern base for D.

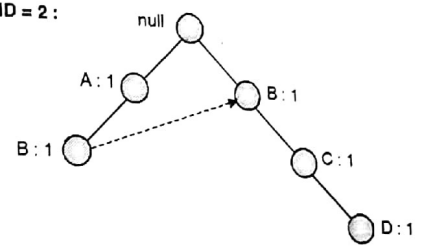
TID	Items
1	{A, B}
2	{B, C, D}
3	{A, C, D, E}
4	{A, D, E}
5	{A, B, C}
6	{A, B, C, D}
7	{B, C}
8	{A, B, C}
9	{A, B, D}
10	{B, C, E}

Soln. :

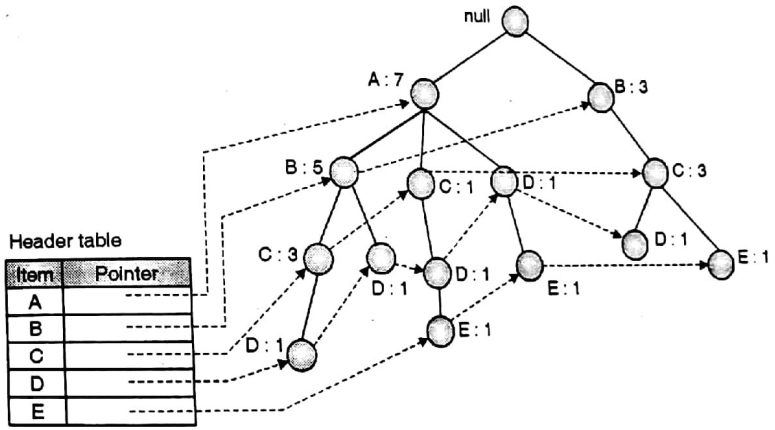
After reading TID = 1 :



After reading TID = 2 :



Similarly for all the remaining transactions, FP tree is given below.



Header table

Item	Pointer
A	
B	
C	
D	
E	

Conditional Pattern base for D :

P = {(A:1,B:1,C:1),
(A:1,B:1),
(A:1,C:1),
(A:1),
(B:1,C:1)}

- We have the following paths with 'D'

P = {(A:1,B:1,C:1),(A:1,B:1), (A:1,C:1), (A:1)] and {(B:1,C:1)}

- 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' to 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.

5.5.6 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.

Syllabus Topic : Mining Frequent Itemsets using Vertical Data Formats

5.6 Mining Frequent Itemsets using Vertical Data Formats

- There are usually two ways of representing transactional data, Horizontal Data format and Vertical data format.
- In Horizontal data format the transactional data is represented as TID-itemset where TID is the transaction id and itemset is the set of items bought in that particular transaction.

E.g. of Horizontal data Format

Transaction ID	Items Bought
T100	1,2,3
T200	1,3
T300	1,4
T400	2,3

- In Vertical data format, transactional data is represented as item-TID-set. Item is the item name and TID-set is the set of transactions containing that item
  - E.g. of Vertical Data Format

Items Bought	Transaction ID set
1	T100, T200, T300
2	T100, T400
3	T100,T200, T400
4	T300

  - 2-itemset in vertical data format

Items Bought	Transaction ID set
12	T100
13	T100, T200
14	T300
23	T100, T400

  - 3-itemset in vertical data format

Items Bought	Transaction ID set
123	T100

Therefore there is only one frequent 3 -itemset {1,2,3}.

- The above process is repeated by the intersection of TID_sets of the frequent k-itemsets to compute the TID_sets of the corresponding (k+1) itemsets. The process is stopped when until no frequent itemsets or candidate itemsets can be found.

5.7 Mining Closed and Maximal Patterns

- Applications like web content mining, network intrusion, usage mining contains very long patterns, these patterns are computationally infeasible, mining becomes a problem.
- To overcome this problem two solutions may be adopted one is mining maximal frequent itemsets and the other is mining closed frequent itemsets.
- Maximal Pattern mining is helpful in dense domains for understanding long patterns but they lead to a loss of information, as subset frequency is not available maximal sets cannot be used for generating rules.
- On the other hand closed sets are lossless, it is possible to find all frequent itemsets and their exact frequency.

Syllabus Topic : Introduction to Mining Multilevel Association Rules

5.8 Introduction to Mining Multilevel Association Rules

MU - May 2015, Dec. 2015, May 2016, Dec. 2016

- 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.

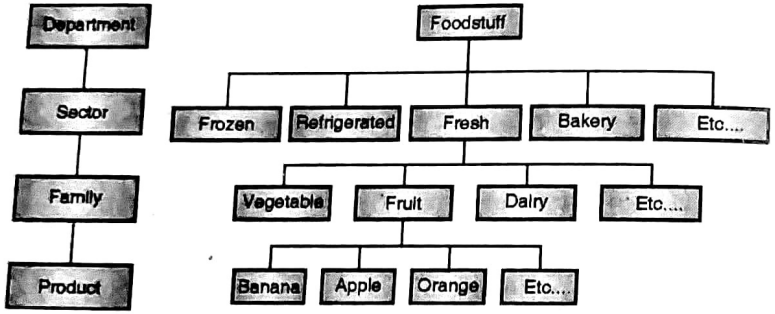


Fig. 5.8.1 : Hierarchy of concept

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 generalized item is more than the support of specialized 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 necessity 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.

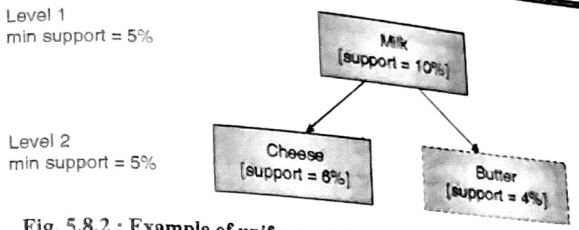


Fig. 5.8.2 : Example of uniform minimum support for all levels

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.

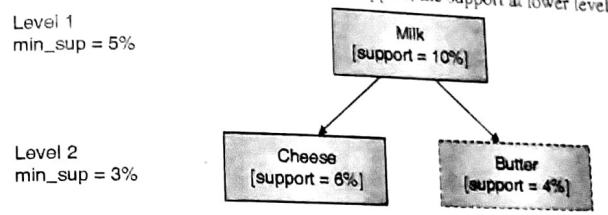


Fig. 5.8.3 : Example of reduced minimum support for lower levels

There are 4 search strategies :

(i) **Level-by-level independent**
   - Its 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 itemset 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 are checked for minimum support threshold this is also called "Level Passage Threshold".

Syllabus Topic : Introduction to Mining Multidimensional (MD) Association Rules

5.9 Mining Multidimensional (MD) Association Rules

MU - May 2015, May 2016, Dec. 2016

- **Single-dimensional rules :** The rule contains only one distinct predicate. In the following example the rule has only one predicate "buys".

  buys(X, "Butter")⇒buys(X, "Milk")

- **Multi-dimensional rules :** The rule contains two or more dimensions or predicates.
- **Inter-dimension association rules :** The rule doesn't have any repeated predicate

  gender(X,"Male") ∧ salary(X, "High") ⇒ buys(X, "Computer")

- **Hybrid-dimension association rules :** The rule have many occurrences of same predicate i.e. buys.

  gender(X,"Male") ∧ buys(X, "TV") ⇒ buys(X, "DVD")

- **Categorical attributes :** This have finite number of possible values and there is no ordering among values. Example : brand, color.
- **Quantitative attributes :** These are numeric values and there is implicit ordering among values. Example: age, income.

Techniques for Mining MD Associations

1. **Using static discretization of quantitative attributes**
   - Using concept hierarchy, discretize the quantitative attributes.
   - Convert the numeric values by ranges or categorical values.

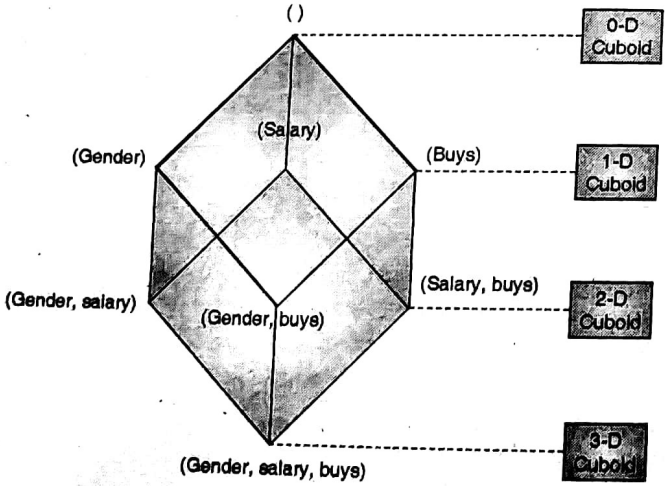


Fig. 5.9.1 : Lattice of cuboids making up a 3-D data cube

- To get the all frequent k-predicate , k or k + 1 table scans are required and generate a cuboid which is more suited for mining
- Mining is always more faster on data cube.
- Predicate sets are determined by the cells of n- dimensional cuboid.
- In the example below, the base cuboid contains the three predicates gender, salary and buys. Each cuboid represents a different group-by.

2. **Quantitative association rules :**
   - Numeric attributes are dynamically discretized such that the confidence or compactness of the rules mined is maximized.
   - In quantitative association rule, the left hand side of rule contains 2-D quantitative attributes and right hand side of rule contains one categorical attribute.

     $A_{quantitative1} \wedge A_{quantitative2} \Rightarrow C_{categorical}$

   - Example : If we are interested in association rule where two quantitative measures are age and salary and the type of the phone that customer buy.

     Age(CUST,"30-34") ∧ Salary (CUST,"24K – 48K") ⇒ buys(CUST, "Sony Xperia Z")

   - The approach used to find such rules is the Association Rule Clustering System (ARCS).
   - Steps involved in ARCS are :

     (a) **Binning:** Partition the ranges of quantitative attributes into intervals. These intervals are considered as bins. ARCS are equi-width binning method where each bin has same interval size.

     (b) **Finding frequent predicate set :** It finds the frequent predicate sets which satisfy the minimum support and also minimum confidence. Rule Generation algorithm is used to generate strong association rules.

     (c) **Clustering the association rule:** Strong association rules obtained from above step is mapped to a 2-D grid as shown in Fig. 5.9.2 :

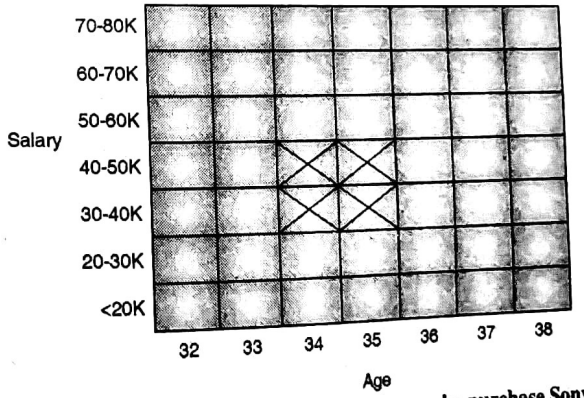


Fig. 5.9.2 : A 2-D grid for tuples representing customers who purchase Sony Xperia Z

Following four CUSTs correspond to the rules :

age(CUST,34) ∧Salary(CUST,"30 – 40K") → buys(CUST," Sony Xperia Z")
age(CUST,35) ∧Salary (CUST,"30 – 40K") → buys(CUST," Sony Xperia Z")
age(CUST,34) ∧Salary (CUST,"40 – 50K") → buys(CUST," Sony Xperia Z")
age(CUST,35) ∧Salary (CUST,"40 – 50K") → buys(CUST," Sony Xperia Z")

- As above rules are close to each other, they can be clustered together to form the following rule :
  age(CUST, "34 – 35") ∧Salary (CUST,"30 – 50K") → buys(CUST," Sony Xperia Z")

Limitations of ARCS

- It works only for quantitative attributes on LHS of rules.
- The limitation is 2D i.e. two attributes on LHS only so doesn't work for more dimensions.

3. Distance-based association rules

- This is a dynamic discretization process that considers the distance between data points.
- This mining process has only two steps :
  - Perform clustering to find the interval of attributes involved.
  - Obtain association rules by searching for groups of clusters that occur together.
- The resultant rules of this method must satisfy :
  - Clusters in the rule antecedent are strongly associated with clusters of rules in the consequent.
  - Clusters in the antecedent occur together.
  - Clusters in the consequent occur together.

Syllabus Topic : From Association Mining to Correlation

5.10 From Association Mining to Correlation Analysis

- Correlation Analysis provides another framework to get the interesting relationship of association rules which is also called as lift.
- Two item sets X and Y are independent, if $P(X \cup Y) = P(X) \cdot P(Y)$ Otherwise X and Y are dependent and correlated.
- Correlation between X and Y is given by the formula:

$$\text{Correlation}(X,Y) = P(X \cup Y) / (P(X) \cdot P(Y))$$

OR

$$\text{Correlation}(X,Y) = P(Y|X) / P(Y)$$

- This is also known as the lift of the association rule A => B
  - Correlation(X, Y) > 1 means that X and Y are positively correlated i.e. the occurrence of one implies the occurrence of the other.
  - Correlation(X, Y) < 1 means that the occurrence of X is negatively correlated with the occurrence of Y.
  - Correlation(X, Y) = 0 means that X and Y are independent and there is no correlation between them.

- The correlations gives an additional information about the association rule (A →B).
- A correlation rule is a set of items $\{i_1, i_2, \ldots i_n\}$, where the items occurrences are correlated.

5.11 Pattern Evaluation Measures

Syllabus Topic : Lift

1. Lift

The lift value of the rule is the another interestingness measures on the rules. Lift is used to filter out the rules based on their importance.

$$\text{Lift}(A \rightarrow B) = \text{confidence}(A \rightarrow B) / \text{support}(B)$$
$$= \text{Support}(A \rightarrow B) / \text{Support}(A).\ \text{Support}(B)$$

if lift > 1, then A and B are positively correlated.

lift< 1, then A are B are negatively correlated.

lift = 1, then A and B are independent

2. all_confidence

- Given an itemset $X = \{X_1,\ldots,X_k\}$ the lowest confidence rule that can be generated from X is the one with the highest support item on the left-hand side.
- The lowest confidence (or all-confidence) can be used to measure the potential of the itemset to generate cross-support patterns.
- A cross-support pattern is an itemset

$$X = \{X_1,\ldots,X_k\} \text{ with low ratio}$$

$$r(X) = \frac{\min [s(X_1), \ldots, s(X_k)]}{\max [s(X_1), \ldots, s(X_k)]}$$

- The all-confidence (or lowest confidence) can be used to measure the potential of the itemset to generate cross-support patterns.
- Itemsets with low all-confidence can be filtered out before rule generation.
- Suppose we have two itemsets X and Y ,then all_Confidence can be calculated as

$$\text{all\_confidence} = \text{support}(X \cup Y) / \max\{\text{support}(X), \text{support}(Y)\}$$

$$= \min\{\text{confidence}(Y \rightarrow X, \text{Confidence } X \rightarrow Y)\}$$

- So all_confidence is the minimum confidence of the two association rules Y→X and X →Y for the itemsets X and Y.

- This measure is very useful for pruning the rules generated as it is anti-monotonic means if item set is not passing minimum association threshold then its superset will also not pass it.

3. max_confidence

- Suppose we have two itemsets X and Y ,then max_confidence can be calculated as,

$$\text{max\_confidence} = \text{support}(X \cup Y) / \min\{\text{support}(X), \text{support}(Y)\}$$
$$= \max\{\text{confidence}(Y \rightarrow X, \text{Confidence } X \rightarrow Y)\}$$

- So max_confidence is the maximum confidence of the two association rules $Y \rightarrow X$ and $X \rightarrow Y$ for the itemsets X and Y.
- Property of this measure is monotonic as if an itemset's association is having the support which is less than minimum threshold, then all of its super sets will be having the support which is greater than that itemsets.

4. Kulczynski

- Suppose we have two itemsets X and Y ,then Kulc can be calculated as ,

  Kulc = 1/2 (confidence($Y \rightarrow X$) + Confidence ($X \rightarrow Y$)).
- So Kulczynski is the average confidence of the two association rules $Y \rightarrow X$ and $X \rightarrow Y$ for the itemsets X and Y which is an arithmetic mean.
- This measure does not have monotonic as well as anti-monotonic property.

5. cosine

- It measures the angle between the two vectors X and Y.
- If $\cos(X \rightarrow Y)$ value is zero the angle between the two vectors is 90 degrees and they don't share any terms.
- If $\cos(X \rightarrow Y)$ value is 1 then the two vectors are the same means they share common terms except for magnitude.
- Cosine is used when data is sparse, asymmetric and there is a similarity of lacking characteristics.

$$\text{Cos}(X \rightarrow Y) = \frac{\text{sup}(X \cup Y)}{\sqrt{\text{sup}(X) \times \text{sup}(Y)}}$$

**6. Imbalance Ratio **

- The Imbalance Ratio(IR) measure the imbalance of the two itemsets X and Y.
- The formula to find IR is given

$$\text{IR}(X, Y) = \frac{|\text{Support}(X) - \text{Support}(Y)|}{\text{Support}(X) + \text{Support}(Y) - \text{Support}(XY)}$$

- If $\text{conf}(X \rightarrow Y) = \text{conf}(Y \rightarrow X)$ then $\text{IR}(X,Y) = 0$.
- Total number of transactions or null transactions doesn't affect the IR.

Syllabus Topic : Introduction to Constraint based Association Mining

5.12 Introduction to Constraint based Association 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 constraints

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

Meta rule-Guided Mining of Association Rule

- Specifies the *syntactic form* of the rules in which we are interested.
- Syntactic forms serves as the constraint.
- It is based on analysts experience, expectation, or intuition regarding data.
- To analyze the customers behavior leading to the purchase of *Apple Products*, meta rule will be $P_1(C,Y)$ and $P_2(C,Z) \rightarrow$ 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 meta-rules. For example, if two predicates Age and Salary are given to analyze whether the customer buys "Apple Product"

  age (C, "30..40") $\wedge$ Salary (C, "30K..50K") $\rightarrow$buys (C, "Apple Product")
- So generalize the meta-rule Guided association rule as a template like

  $P_1 \wedge P_2 \wedge \ldots \wedge P_n \rightarrow Q_1 \wedge Q_2 \wedge \ldots \wedge Q_r$

  Where, each P_i's and Q_j's are predicates

  And the number of predicates in the meta rule is $p = n + r$

Review Questions

Q. 1 What is Market Basket Analysis?

Q. 2 Define the following terms :

a) Frequent Itemsets b) Closed Itemsets

c) Association Rules d) Support

e) Confidence

Q. 3 Explain the Apriori Algorithm

Q. 4 State what are the limitations of Apriori Algorithm and how the efficiency can be improved.

Q. 5 Explain the FP Tree algorithm.

Q. 6 Explain the different Pattern Evaluation Measures.

Q. 7 Write a short note on Constraint Based Association mining.

5.13 University Questions and Answers

May 2015

Q. 1 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. *(Ans. : Refer Ex. 5.4.9)* **(10 Marks)**

Q. 2 Explain : Multilevel and multidimensional association rules.
(Ans. : Refer sections 5.8 and 5.9) **(10 Marks)**

Dec. 2015

Q. 3 Explain Multilevel association rules with suitable examples.
(Ans. : Refer Section 5.8) **(10 Marks)**

May 2016

Q. 4 Use the Apriori algorithm to identify the frequent item-sets in the following database. Then extract the strong association rules from these sets. .

Minimum support - 30% ,Minimum confidence = 75%.

TID	Items
01	A, B, D, E, F
02	B, C, E,
04	A, B, D, E,
04	A, B, C, E,
05	A, B, C, D, E, F
06	B, C, D
07	A, B, D, E

(Ans. : Refer Ex 5.4.8) **(10 Marks)**

Q. 5 Explain multidimensional and multi level association rules with examples.
(Ans. : Refer Sections 5.8 and 5.9) **(10 Marks)**

Dec. 2016

Q. 6 Use the Apriori algorithm to identify the frequent item-sets in the following database. Then extract the strong association rules from these sets
Min. Support – 30% , Min. Confidence – 75%

TID	Items
01	A, B, D, E, F,
02	B, C, E,
03	A, B, D, E,
04	A, B, C, E,
05	A, B, C, D, E, F
06	B, C, D,
07	A, B, D, E

(Ans. : Refer Ex. 5.4.8) **(10 Marks)**

Q. 7 Explain multidimensional and multi level Association rules with examples.
(Ans. : Refer sections 5.8 and 5.9) **(10 Marks)**

...Chapter Ends

❑❑❑

CHAPTER 6

Business Intelligence

Syllabus

What is BI? Business intelligence architectures; Definition of decision support system; Development of a business intelligence system using Data Mining for business Applications like Fraud Detection, Clickstream Mining, Market Segmentation, retail industry, telecommunications industry, banking & finance CRM etc.

Syllabus Topic : What is Business Intelligence ?

6.1 What is Business Intelligence?

MU - May 2015, Dec. 2015

The large amount of data which we get through internet are often heterogeneous in origin,represented in different formats, contents are also varied with respect to origin. So there is a need to convert such data into knowledgeable information, which can be used by decision makers to take decision for the improvement of enterprise or their business.

Carlo Vercell is defined Business intelligence as "a set of mathematical models andanalysis methodologies that exploit the available data to generate informationand knowledge useful for complex decision-making processes".

Benefits of a business intelligence system

1. BI systems reduce labour costs by generating reports automatically.
2. Make information actionable as user can get data as per their requirement to get the knowledge.
3. Decision maker can take better decision as exact and up-to-date information is provided.
4. Multiple data sources can be combined through BI, so decision can be taken faster.
5. Business metrics reports are available and can be accessed from anywhere whenever there is a need.
6. Get insight into customer behavior.

Syllabus Topic : Business Intelligence Architectures

6.2 Business Intelligence Architectures

- A typical architecture of a business intelligence given by **Carlo Vercellis** is as shown in Fig. 6.2.1.

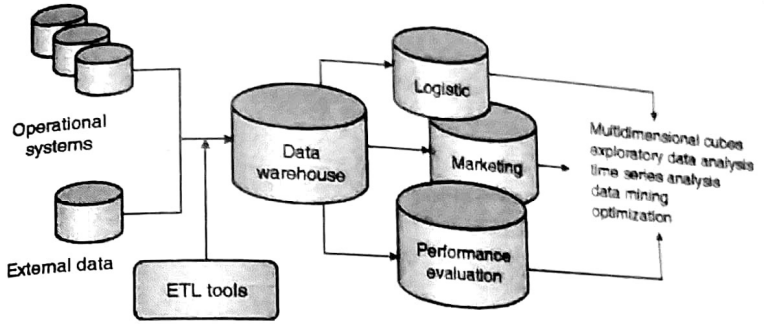


Fig. 6.2.1 : A typical architecture of business Intelligence

6.2.1 The Three Major Components of BI Architecture

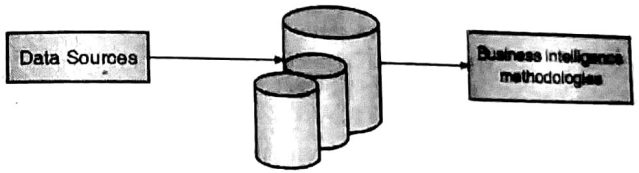


Fig. 6.2.2 : Major components of BI

1. Data sources

- Different data sources can be relational DBMS like Oracle, Informix.
- In addition to these internal data, operational data also includes external data obtained from commercial databases and databases associated with suppliers and customers, which include unstructured documents like emails.
- It is necessary to perform all the operations associated with extraction and integration of the data from different heterogeneous sources.

2. Data warehouses and Data marts

- An ETL tool (Extract,Transform,Load) is a tool that reads data from one or more sources, transforms the data so that it is compatible with destination and loads the data to the destination databases.

- ETL function transforms the relevant data collected from different source systems into useful information and then stores it in the data warehouse or data mart, which can be used for strategic decision making.

3. Business Intelligence methodologies

- The main purpose of a data warehouse is to provide information to the business managers for strategic decision-making.
- Extracted data from data warehouse or data mart is used to supply the mathematical models and analysis methodologies.
- These users interact with the warehouse using end user access tools.

Some of the examples of end user access tools can be:

- Reporting and Query Tools
- Application Development Tools
- Executive Information System Tools
- Online Analytical Processing Tools
- Data Mining Tools

6.2.2 Different Components of a Business Intelligent System

- **Data sources : Various data sources from where the data is collected for BI system is important component of BI system. This is one of the major** Components of BI Architecture and explained in the Section 6.2.1.
- **Data warehouses and data marts :** Data warehouse and data marts keeps the historical information of business related data. This information helps the business people to take future decisions for business and also do analysis based on past data.
- **Business intelligence methodologies :** This is also the part of BI architecture and described in above section.
- **Data exploration**
  - It helps to understand the characteristics of data.
  - Consist of query and reporting systems and statistical methods, which help to recognize unseen patterns.
  - It is used to generate prior hypotheses or to define the selection criteria of a dataset.
- **Data mining**
  - Data Mining is a technology, which helps organisations to process data through algorithms or mathematical models to uncover meaningful patterns and correlations from large databases that otherwise may not be possible with standard analysis and reporting.
  - Data Mining tools can help one to understand the business better and also exploited to improve future performance through predictive analytics and make them proactive and allow knowledge driven decisions.

  - Data Mining field brings together techniques from machine learning, pattern recognition, statistics, databases and visualization to deal with the issues of information extraction from large databases.
- **Optimization**
  - To determine the best solution out of a set of alternative actions, which is usually fairly extensive and sometimes even infinite?
  - Under the given constraints and alternatives, finding cost effective and high performance solution.
- **Decisions**

Finally, the decision for the enterprise is taken by the decision maker using business intelligence methodologies.

Syllabus Topic : Definition of Decision Support System

6.3 Definition of Decision Support System

- The structure of a decision support system is been shown in Fig. 6.3.1.
- The DSS is used to solve semi-structured and unstructured problems.
- It consists of a database, mathematical models and a user interface for communicating with the end users.

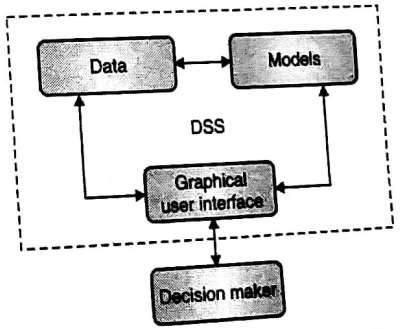


Fig. 6.3.1 : Structure of a Decision Support System

Features of DSS

1. Effectiveness

- Decision support systems are important component of Business Intelligence architecture.
- Using a DSS can give more accurate analysis, which needs more time investment from decision makers.

1. **Analysis**
   - Need to develop BI is identified first.
   - Interviews of knowledge workers are conducted who are performing different roles in organization.
   - The general objectives and priorities of BI project are described.
   - Cost of the development of project is calculated.
   - Set out the benefits coming from thedevelopment of the BI system.
2. **Design**
   - Derive a provisional plan of the overall architecture by considering development in near future and also the evolution during the midterm.
   - All the processes supported by BI should be thoroughly studied to get the information requirement.
   - First, it is necessary to make an assessment of the existing information.
   - Use classicalproject management methodologies to get the project plan.
   - Identify development phases of project.
   - Find out the expected execution times and costs.
   - Planning of required roles and resources.
3. **Planning**
   - The functions of the BI system are defined and described in greater detail.
   - Identification of the data definition, which is needed for data warehouse and data mart.
   - Externally retrieved data are evaluated.
   - Simultaneously the mathematical models needed are defined.
   - Availability of data required for mathematical model and the efficiency of the algorithms are verified.
   - Finally develop a system prototype by considering the cost factor with limited capabilities.
   - Actual needs and project specifications should be considered while developing prototype.
4. **Implementation and Control**

There are five main sub-phases in this phase.

1. **Development of data warehouse data mart :** Information infrastructure is given by this phase which is required by BI system.
2. **Development of metadata :** Meaning and relations of data is described by metadata. Also the transformation of data is mentioned in this phase.
3. **Development of ETL tools :** Extraction and transformation of data collected from primary sources and also the loading of data into data warehouse and data mart are given by ETL procedures.

4. **Development of application :** Development of essential BI applications are given in this phase.
5. **Release and Testing :** Finally the system is released for test and then used by the users

6.5 Business Intelligence

- Business intelligence is a collection of refined techniques that combine understandings from risk analysis, business strategy, organizational behavior, cognitive psychology and political science.
- In a rational management framework, Business intelligence makes use of information systems and transactional databases to offer decision making support and convert data into intelligence.

Why business intelligence ?

- Constantly changing circumstances and challenges are faced by a Business or an organization, nothing remains static for a longtime.
- Due to this changing environment, decisions have to be continuously taken by the business or organizations to adjust their profitable actions or enhancement in the services they provide.
- By making use of the data held within the organization BI can be helpful in two ways :
  - Detecting trends and early warning system.
  - Relevant patterns and insights can be found.

The BI system can be used to monitor or track following measures

1) **Achieving top performance**
2) **Unexpected patterns or trends**
   - Using BI, unexpected patterns or trends can be discovered, for this purpose a strong and easy to ad hoc querying or graph presentation system may be needed.
   - The data in a BI system can be reviewed and readily ad hoc questions may be posed, with this you can discover a difference in performance, which requires investigation.
   - These types of discoveries lead to breakthrough in performance. For e.g. an unnoticed market segment growth, outstanding performance by one of the business unit.

Differences between data mining tools and business intelligence/reporting tools

- OLAP tools have a new name called as Business Intelligence. BI tools can be used for list management, graphical reporting, query functionality management. BI tools extend the functionality to OLAP operations like drilling up, drilling down and across the data with some level of multidimensional analysis and aggregation.
- Data Mining is one step ahead of a simple data warehouse. A Data warehouse is a simple method for organizing the data, data mining is used to find hidden patterns in the data. For example customers with common interest can be grouped together with the help of data mining.

Business Intelligence Issues

Following are the key business intelligence issues divided into two categories :

1. Organizations and People
2. Data and Technology

Organizations and People

1) Management within an organization do not agree that the decision taken based on data or evidence work for them, they prefer to run the operation from instinct.
2) In order to assess business progress there is no overall business strategy laid out with objectives and measures for those objectives.
3) The data needed for Business Intelligence system cannot be obtained from IT personnel as they are overloaded and they have no resources available.
4) For performance improvement of the business either making use of BI or not, there is no incentives provided to the staff within the organization.
5) There is no obvious time to establish a BI system. The Business is in a state of high change or flux.
6) Until the results of a BI system are seen, the users aren't aware about what is expected from a BI system. This results in a lot of changes to the solution before it can be accepted.
7) The Business is not understood by the IT experts due to which many changes are expected to the system before the organization can accept it.
8) To manage project implementation on time and within budget or to design the system adequately, the company fails to have sufficient expertise or is unable to hire such expertise.

Data and Technology

1) The Data of the organization is not clean. The time and effort needed to correct or handle this type of data leads to an unsuccessful delivery of the BI project.
   For example, there could be many different definitions for the same item. E.g. a customer may be coded differently on sales system to that held in accounts system.
2) The Technology chosen for BI turns out to be so particular that it ultimately results in time consuming process which leads to delay in project completion.
3) The BI technology discourages the use of system due to the following reasons :
   - The information presentation quality is poor or limited.
   - Response time for data presentation is too slow or not acceptable.
   - It is too difficult or limited to ask new questions of BI technology for either of them, end users or BI expert.

Syllabus Topic : Data Mining for Business Applications like Fraud Detection

6.6 Fraud Detection

Fraud Detection for Telecommunications Industry

- In the last few years there has been huge expansion in telecommunications industry with development in reasonable mobile phone technology. The global mobile phone fraud is increasing with the increase in mobile phone users.
- There are various types of telecom fraud which occurs at different level.
- Common types of frauds are :
  - Subscription fraud
  - Superimposed or "surfing" fraud.
- When the fraud occur while subscribing to a service, then that fraud is called as subscription fraud. In this type of fraud, the fraudster wants to hide his/her identity by providing wrong documents with false identity. The intension of fraudster is not to pay bill amount, so all transaction with this number will be fraudulent.
- When the unauthorized person uses the service without having any authority, then that fraud is called as superimposed fraud. This type of fraud is detected when fake or phantom calls appear on a bill.
- Superimposed frauds can be carried out in a number of ways, which includes mobile phone cloning and getting authorization details of calling card.
- These types of frauds occur at the level of individual calls, the calls that are fraudulent will be mixed in with authentic ones.
- Frauds at subscription can be detected at billing process, though the aim involves in detecting it much before that, since huge costs can run up.
- Superimposed fraud can remain undetected for a long time.
- Telecommunications networks generate vast quantities of data, sometimes on the order of several gigabytes per day, so that data mining techniques are of particular importance.

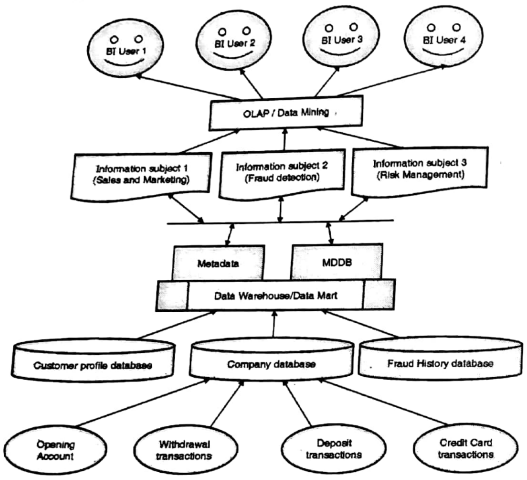


Fig. 6.6.1 : BI System for Credit Card Fraud detection

- Various data mining techniques like rule based detection system, statistical summarization etc are used to detect such telecommunication fraud :

  Simple rule-based detection systems use rules like :

  - Checks the distant geographical location with respect to time,
  - Overlapping calls at the same time,
  - Long calls or High value calls.
- At a higher level, statistical summaries of call distributions, supervised learning methods or threshold techniques are used to detect the frauds.
- Statistical and Artificial intelligence techniques are used for fraud detection.
- Statistical data analysis techniques are used for :
  1. For detection, validation, error correction, and filling up of missing or incorrect data Data preprocessing techniques are used.
  2. Various statistical parameters such as averages, performance metrics can be calculated.
  3. User profile computation.
  4. Analysis of time-dependent data.
  5. Finding patterns and association among groups of data using clustering and classification.

- AI techniques can be used for :
  1. Find associations and rules in the data automatically to signify interesting patterns which are related to fraud transactions.
  2. Detection of frauds in the form of rules using expert systems.
  3. Automatic Detection of approximate classes, clusters or patterns of suspicious behavior using pattern recognition.

Syllabus Topic : Click-stream Mining

6.7 Clickstream Mining

- The methodology adopted by users for surfing web is difficult to be analysed and understand.
- A lack of information is observed in quantitative data about what the user intends to do, while qualitative data is difficult to be gathered for large samples.
- Once a website is made live i.e. published, the user is the final controller of their own navigation. This involves a variety of strategies for browsing.
- The user's goals are not the only factor dependent on these strategies but it involves other factors such as proficient, understanding of the site, time pressures and observed cost of information
- How can this changing nature of browsing strategiescan be helpful in identifying the use made in the existing website.
- One solution is to make use of clickstream logs. These logs have information like address of each web page visited, date and time of the visit and the referring page. All this information is considered to be very valuable from Internet user activity point of view.
- Clickstream logs can be generated from the client side or from the server logs.

6.7.1 Clickstream Data : Collection and Restoration

- The server side clickstream data can be used as the most common tool for collecting data of the pages visited by the website users
- The clickstream log contains information about the pages sent as a response by the server on client's request. These logs are huge and present an incomplete picture of the activity
- For example, server side logs do not contain information that involves client side browser like browser caching (use of back button), network caching (pages held in an intermediate server's cache), or navigation of pages that are a part of the site but are kept on another server
- In spite of these server side limitations, algorithm like Pattern Restore Method (PRM) can be used to capture some aspects of user behaviour like use of back button, opening of new /additional windows within the same website.

6.7.2 Clickstream Data : Visualisation and Categorisation

- Once the clickstream data have been processed, analysing and categorising these data into usage pattern is needed using a suitable technique.
- Using visualisation techniques i.e. by producing a 'Footstep' graphs it is possible to do the analysis. The graph involves a 2D x-y plot, where x-axis shows the browsing time between two web pages and the y-axis shows the web pages in the users browsing routine.
- Thus the amount of time spent by the user on browsing is shown on x-axis and a transition from one web page to another is shown on y-axis.

Syllabus Topic : Market Segmentation

6.8 Market Segmentation

- Market segmentation is a process into which a market is divided into smaller sub-markets called segments. They are homogenous or have similar attributes. Purchasing Patterns and trends appear significantly in some segments.
- A Good market segmentation is the one in which segments with significant patterns can arise.
- Using Market segmentation the following may be analysed :
  - **Analysis of Market responsiveness :** This is very useful in direct marketing as product offerings responsiveness can be easily made available
  - **Analysis of Market Trend :** Market trends can be revealed by analysing segment by segment changes in sales revenues. This information is very important for every changing markets

6.8.1 Market Segmentation for Market Trend Analysis

- In market segmentation, group the similar attributes to create the segments. Various attributes are used to generate the market segment based on the area of interest of business. For example,
  - Geographical locations like regions, countries, states, zip-codes, etc.
  - Demographics of person like gender, age, income, education, etc.
  - Life style classification of person.
  - Sales related information like sales channels, branches, and departments.
  - Product and product categories.
  - Various offers and types of offers.
- Market segmentation helps to find trend analysis.
- Trent analysis can be done for various segment trend information like :
  - Sales revenues for next quarters or year.
  - Segments growth in dollar and growth rate in terms of percentage.
  - Highest revenue decline in dollar and decline rates in terms of percentage.
  - Exponential growth or decline of segments.

6.8.2 Sales Trend Analysis

Identification of newly emerging trends is very important for a business from time to time. Market trends are indicated by sales pattern of customer segments. Significance of new trends can be obtained from upward and downward trends. For identifying market trends embedded in change of sales revenue, time series predictive modeling can be used. Sales trends are important for customer retention as well as marketing point of view.

Typical sales trend analysis includes

- Customer segments having highest growth or highest revenue decline in terms of rupee.
- Customer segments having highest growth rates.
- Customer segments having highest revenue decline rates in terms of percentage.
- The growth or decline trend.
- Exponential growth or decline of customer segment.

Trends may be categorized as

- Short term trends capture rapidly emerging trends.
- Midterm trends capture trends developing in between.
- Long term trends capture trends developing over long periods.

Syllabus Topic : Retail Industry

6.9 Retail Industry

- The retail industry is realizing the possibilities of advantages by using data mining.
- A rich source is provided by the Retail industry for Data mining.
- Just like other industry for e.g. Banking, retail industry is also collecting huge amounts of data throughout the years and can find useful pieces of information through various tools.
- In Retail industry, data mining can be used for multidimensional analysis of sales, customers and products. It can be helpful in customer retention, the effectiveness of sales campaign may be analyzed.

Following are some examples of how data mining is useful in retail industry.

1. **Performing market basket analysis**

   Finding which items customers buy together. This information can be helpful in Layout strategies inside the store, stocking and promotions.

2. **Sales forecasting**

   Retailers can make stocking decisions by exploring time based patterns. For e.g. if a customer is buying a certain item today, when likely he will buy a complimentary item.

3. **Database marketing**

   Customer profiles with certain behaviours can be developed and used for cost effective promotions. For e.g. some customers may purchase designer label clothings or some may attend sales on regular basis.

4. **Merchandise planning and allocation**

   Merchandise planning and allocation for new retail stores can be improved by exploring patterns in stores with similar demographic characteristics. Ideal layout for a specific store can also be determined with the help of data mining.

5. **Risk management**

   - Risk management is another important area that needs to be considered in the retail industry along with data mining.
   - Retail industry use data mining to find out which products may not be suitable for competitive offers or changing customer buying patterns.

6. **Fraud detection**

   Fraud detection is important in retail industry, for e.g. at a PoS (Point of Sale) terminal a fraud can take place which can be reduced by use of data mining.

Syllabus Topic : Telecommunications Industry

6.10 Telecommunications Industry

- One of the first industries to adopt data mining was telecommunications industry. This industry maintains data related to phone calls, which contains detailed information about each phone call.
- The data stored is in a huge amount, it is of high quality and has a very large customer base and it operates in a rapidly changing and highly competitive environment.
- Data mining plays an important role in telecommunication industry in improving their marketing efforts, identifying fraud and managing their networks.
- One of the challenging issues in using data mining in this industry is the huge amount of data, the sequential and temporal aspects of their data, and prediction of a rare event like a fraud or network failures in real time.
- Use of data mining in telecommunications can be viewed as an extension of the use of expert systems.
- These types of systems can be used to address the complexity associated with the maintenance of a huge network infrastructure, maximizing network reliability by reducing the labor costs.
- One of the drawback of these systems is the expenses incurred in developing them from complexity and time point of view to elicit the requisite domain knowledge from experts.
- Data mining can be viewed as a means of generating this knowledge from the data automatically.

The telecommunication industry faces a number of data mining challenges

- Telecommunication companies have very large databases due to which the scalability of the data mining methods is a key concern.
- The data collected is in the form of transactions/events, hence not at a proper level of semantic for data mining. For example, the call details are to be mined at customer level but the raw data represents individual phone calls. This makes it necessary to aggregate the data at a proper semantic level before data mining.
- Many of its applications like fraud detection and network fault detection needs to be operated in real time. This is yet another issue that needs to be considered.
- The effort needed to handle these issues makes Telecommunication industry a leader in mining data streams.
- Maintaining signatures of data streams, which include its description, which can be updated quickly, is one of the ways in which a data stream can be handled.
- The occurrence of a fraud or a network equipment failure is a rare event in telecommunications industry, predicting and identifying these rare events is a major challenge and it is a difficult task for many data mining algorithms, this types of issues must be handled very carefully to ensure reasonably good results.

Syllabus Topic : Banking and Finance

6.11 Banking and Finance

- The **Banking Industry** at its core offers access to credit. For Lenders it includes access to their own investments and interest payments on the amount. For Borrowers, it gives an access to loans for the creditworthy, at a good interest rate.
- For managing individual and institution finances, banking industry includes services like verification of account details, balance details, funds transfer and advisory services
- The huge amount of data collected over the years by the banking industry can greatly influence the success of data mining efforts.
- Data mining can be used to analyze patterns and trends, which includes prediction with increased accuracy, how customers will be reacting to adjustment in interest rates, which of the customers will be likely to accept new product offers, which of them will be at a higher risk for non-payment of a loan, making customer relationships more profitable.
- With the help of data mining tools, bank can also subsequently offer 'tailor-made' products and services to those customers.
- Other areas include customer segmentation and profitability, credit scoring and approval, predicting payment default, marketing, detecting fraud transactions, cash management and forecasting operations, optimising stock portfolios and ranking investments. It can also be used in finding the most profitable credit card customers or high risk loan applicants

Some of the examples of how data mining has been effectively utilized by banking industry are given as :

1. **Cross-selling**
2. **Card marketing**

   Customer segmentation can help the card issuers and acquirers to improve profitability with effective acquisition and retention programs, targeted product development and customized pricing.

3. **Cardholder pricing and profitability**

   Data mining technology can be used to price the products, which will maximize the profit and minimize the loss of customers. This includes risk-based pricing.

4. **Fraud detection**

   Analyses of past transactions that were later identified to be fraudulent, banks can identify patterns. Frauds are considered to be extremely costly.

5. **Predictive life-cycle management**

   With the help of Data mining prediction of each customer's lifetime value and to service each of the segments appropriately by giving special deals and discounts is possible.

Syllabus Topic : CRM

6.12 CRM

- Customer Relationship Management (CRM) has appeared in the last decade. It was developed to reflect the role of customer in the strategic positioning of a company.
- It includes measures for customer understanding and using this knowledge for designing and implementing marketing strategies, production alignment and coordinating the supply chain.

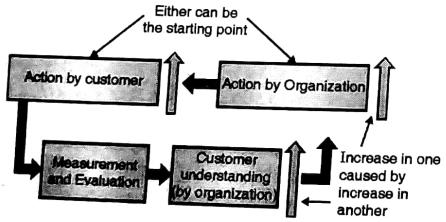


Fig. 6.12.1 : The basic CRM Cycle

- CRM lays emphasis on the coordination of these types of measures and implies the integration of customer-related data, meta-data and knowledge, centralized planning and evaluation of measures to increase customer lifetime value.
- It gains importance in companies that serve multiple groups of customers and exploits interaction channels for them.
- CRM offers a wide variety of functions, not all of them requires data mining.
- These functions include *marketing automation* (e.g. campaign management, cross and up-sell, customer segmentation, customer retention), *Automation of sales force* (e.g. contact management, lead generation, sales analytics, generation of quotes, product configuration), and *Management of contact center* (e.g. call management, integration of multiple contact channels, problem escalation and resolution, metrics and monitoring, logging interactions and auditing).

6.12.1 Data Mining Challenges and Opportunities in CRM

- **Data integration requirement before data mining :** Data source is from multiple sources for both CRM and data mining. In a CRM, data may be collected over various departments in an organization. Interesting patterns may span multiple sources due to which there is a need to integrate them before actual data mining exploration can be applied.
- **Different data types :** Due to different data types, integrated mining of diverse and heterogeneous data is needed. In CRM this issue is not important. Customer data comes in the form of structured records of different data types (e.g. demographic data), temporal data (e.g. weblogs), text (e.g. emails, consumer reviews, blogs and chat-room data), audio (e.g. recorded phone conversations of service representative with customers).
- **Noisy data in CRM :** Weblog contains a lot of noise due to crawlers, missed hits due to caching problem. Data pertaining to customer "touch points" has cleaning problems seen in any business related data.
- **Privacy and confidentiality for data and analysis results :** This is a major issue. In CRM there is a huge amount of data that is confidential, like email and phone logs. Concern about inference capabilities makes other forms of data sensitive as well. E.g. someone can recover Personally Identifiable Information (PII) from web logs.

Case Study :

Consider the following case study : An international chain of hotels wants to analyse and improve its performance using several performance indicators quality of room, service facilities, check in, breakfast, popular time of visits, duration of stay etc. For this case study design a BI system, clearly explaining all steps from data collection to decision making.

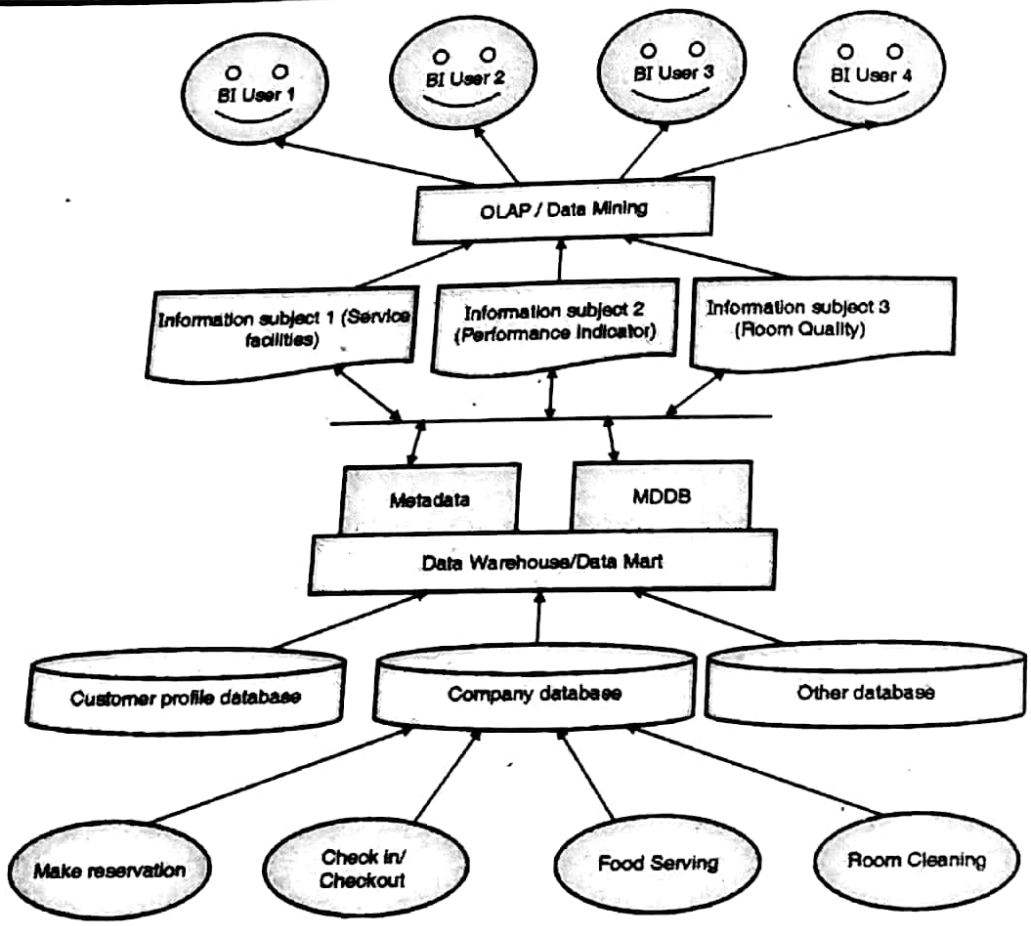


Fig. 6.12.1 : BI System for Chain of Hotels

Review Questions

Q. 1 What is Business Intelligence ?

Q. 2 Explain the architecture of a BI system.

Q. 3 Explain the different phases in the development of a BI system.

6.13 University Questions and Answers

May 2015

Q. 1 Define "Business Intelligence" with examples. *(Ans. : Refer Section 6.1)* **(5 Marks)**

Dec. 2015

Q. 2 Define business intelligence system with examples. *(Ans. : Refer Section 6.1)* **(5 Marks)**

...Chapter Ends

□□□