UNIT 1

Syllabus :

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

UNIT III

Syllabus :

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

UNIT IV

Syllabus :

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

UNIT V

Syllabus :

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

❑❑❑

CHAPTER 1

Unit I

Introduction

Syllabus Topics

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

Syllabus Topic : Data Mining

1.1 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 otherwise may not be 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.

☞ **Definition of Data Mining**

➔ **(SPPU - Dec. 15)**

Q. Define Data Mining. Dec. 15, 2 Marks

- Data mining is processing data to identify patterns and establish relationships.
- Data mining is the process of analysing 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.

1.1.1 Applications of Data Mining

- 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.

1.1.2 Challenges to Data Mining

→ (SPPU - Oct. 16)

Q. Describe three challenges to data mining regarding data mining methodology. **Oct. 16, 6 Marks**

1) Mining different kinds of knowledge in databases

- Different users are interested in different kinds of knowledge and will require a wide range of data analysis and knowledge discovery tasks such as data characterization, discrimination, association, classification, clustering, trend and deviation analysis, and similarity analysis.
- Each of these tasks will use the same database in different ways and will require different data mining techniques.

2) Interactive mining of knowledge at multiple levels of abstraction

- Interactive mining, with the use of OLAP operations on a data cube, allows users to focus the search for patterns, providing and refining data mining requests based on returned results.
- The user can then interactively view the data and discover patterns at multiple granularities and from different angles.

3) Incorporation of background knowledge

- Background knowledge, or information regarding the domain under study such as integrity constraints and deduction rules, may be used to guide the discovery process and allow discovered patterns to be expressed in concise terms and at different levels of abstraction.
- This helps to focus and speed up a data mining process or judge the interestingness of discovered patterns.

1.1.3 KDD Process (Knowledge Discovery in Databases)

→ (SPPU - Aug. 17)

Q. Explain the knowledge discovery in database (KDD) with diagram. What is the role of data mining steps in KDD? **Aug. 17, 6 Marks**

- 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 Visualisation.
- 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

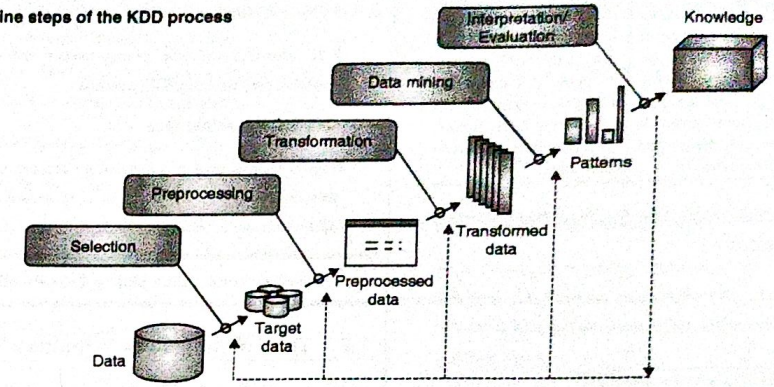


Fig. 1.1.1 : KDD Process

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

1. Developing an understanding of

(i) The application domain

(ii) The relevant prior knowledge

(iii) 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

(i) Noise or outliers are removed.

(ii) Essential information is collected for modeling or accounting for noise.

(iii) Missing data fields are handled by using appropriate strategies.

(iv) Time sequence information and changes are maintained.

4. Data reduction and projection

(i) Based on the goal of the task, useful features are found to represent the data.

(ii) 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)

(i) Pattern search is done using the appropriate Data Mining method(s).

(ii) A decision is taken on which models and parameters may be appropriate.

(iii) 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

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

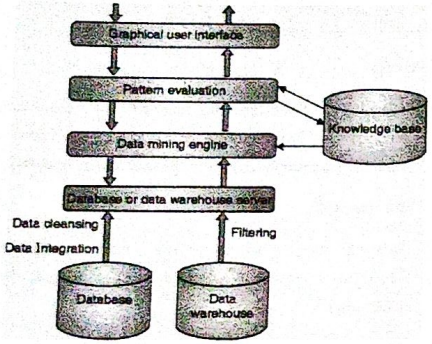


Fig. 1.1.2 : Architecture of typical data mining system

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 : Data Mining Task Primitives

1.2 Data Mining Task Primitives

Data mining primitives define a data mining task, which can be specified in the form of a data mining query.

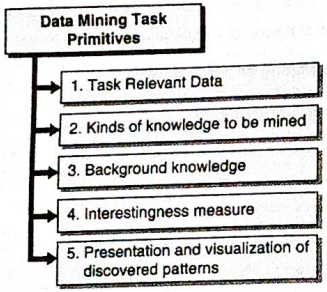


Fig. 1.2.1 : Data Mining Task Primitives

➜ **1. Task relevant data**

- Specify the data on which the data mining function to be performed.
- Using relational query, a set of task relevant data can be collected.
- Before data mining analysis, data can be cleaned or transformed.
- Minable view is created i.e. the set of task relevant data for data mining.

➜ **2. The kind of knowledge to be mined**

- Specify the knowledge to be mined.

- Kinds of knowledge include concept description, association, classification, prediction and clustering.
- User can also provide pattern templates. Also called metapatterns or metarules or metaqueries.

➜ **3. Background knowledge**

- It is the information about the domain to be mined.
- Concept hierarchies is the form of background knowledge which helps to discover the knowledge at multiple levels of abstraction.

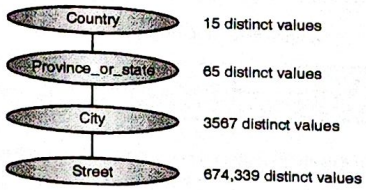


Fig. 1.2.2 : Concept hierarchy for the dimension location

☞ **Four major types of concept hierarchies**

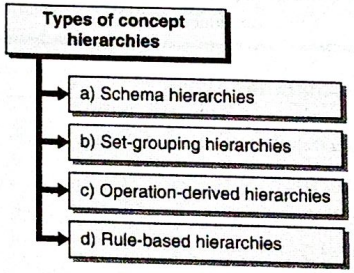


Fig. 1.2.3 : Types of hierarchies

➜ **a) Schema hierarchies**

It is the total or partial order among attributes in the database schema.

Example : Location hierarchy as street < city < province/state < country

➜ **b) Set-grouping hierarchies**

It organizes values into sets or groups of constants.

Example : For attribute salary, a set-grouping hierarchy can be specified in terms of ranges as in the following :

{low, avg, high} € all (salary)

{1000....5000} € low

{5001....10000} € avg

{10001....15000} € high

➜ **c) Operation-derived hierarchies**

It is based on operation specified which may include decoding of information-encoded strings, information extraction from complex data objects, data clustering.

Example : URL or email address

xyz@cs.mu.in gives login name

< dept. <univ.< country

➜ **d) Rule-based hierarchies**

It occurs when either whole or portion of a concept hierarchy is defined as a set of rules and is evaluated dynamically based on current database data and rule definition.

☞ **Example**

Following rules are used to categorize items as *low_profit, medium_profit* and *high_profit_margin.*

low_profit_margin (Z) <= price (Z, A1)^cost (Z, A2)^((A1-A2)<80)

medium_profit_margin

(Z) <= price (Z, A1)^cost (Z, A2)^((A1-A2) ≥ 80)^((A1-A2)≤350)

high_profit_margin (Z) <= price (Z, A1)^cost (Z, A2)^(A1-A2)>350)

➜ **4. Interestingness measures**

- It is used to confine the number of uninteresting patterns returned by the process.
- Based on the structure of patterns and statistics underlying them.
- Each measure is associated a threshold which can be controlled by the user.
- Patterns not meeting the threshold are not presented to the user.

- **Objective measures of pattern interestingness :**
  - **Simplicity :** A patterns interestingness is based on its overall simplicity for human comprehension.
  - **Example :** Rule length is a simplicity measure
  - **Certainty (confidence) :** Assesses the validity or trustworthiness of a pattern. Confidence is a certainty measure.

$$\text{Confidence } (A \Rightarrow B) = \left(\frac{\text{(Number of tuples containing both A and B)}}{\text{(Number of tuples containing A)}} \right)$$

  - **Utility (support) :** It is the usefulness of a pattern support.

$$(A \Rightarrow B) = \frac{\text{(Number of tuples containing both A and B)}}{\text{(total number of tuples)}}$$

  - **Novelty :** Patterns contributing new information to the given pattern set are called **novel patterns**. (**Example :** Data exception).

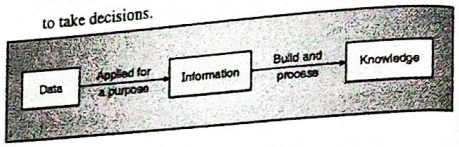
➜ **5. Presentation and visualization of discovered patterns**

- Data mining systems should be able to display the discovered patterns in multiple forms, such as rules, tables, crosstabs (cross-tabulations), pie or bar charts, decision trees, cubes, or other visual representations.
- User must be able to specify the forms of presentation to be used for displaying the discovered patterns.

Syllabus Topic : Data - Data, Information and Knowledge

1.3 Data : Data, Information and Knowledge

- **Data** represents a single primary entities and the related transaction of that entity. Data are facts, which are not processed or analyzed. Example : "The price of petrol is Rs. 80 per litre".
- **Information** is obtained after processing the data and then data has been interpreted and analysed. Such information is meaningful and useful to the user. Example : "The price of petrol is increased from Rs. 80 to Rs. 85 in last 3 months". This information is useful for user who keeps a track of the petrol prices.
- **Knowledge** is useful to take decisions and actions for business. Information is transformed into knowledge. Example "When the petrol prices increases, it is likely that transportation cost also increases".
- So to get boundaries between data, information and knowledge is not easy. Sometimes data may be information for others. But finally knowledge helps to take action for business and delivers the matrix or value for decision makers to take decisions.



(a) From data to information to knowledge

(b) Knowledge leads to action and BI delivers value for decision makers

Fig 1.3.1

Syllabus Topic : Attributes Types - Nominal, Binary, Ordinal and Numeric Attributes, Discrete Versus Continuous Attributes

1.4 Attributes Types

☞ **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**

- 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 :

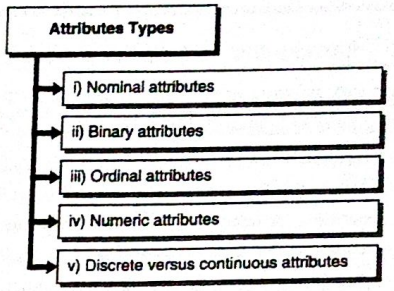


Fig. 1.4.1 : Attributes Types

➜ **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.

Examples

- 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. For 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.

Examples

- 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,

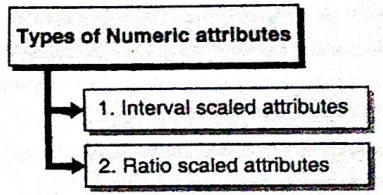


Fig. 1.4.2 : Types of Numeric attributes

➜ **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.
- **For 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*** includes voltage output of a digital device, a person's age in years.

Syllabus Topic : Introduction to Data Pre-processing

1.5 Introduction to Data Pre-processing

Q. What is data pre-processing ? (2 Marks)

- 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 organisations and businesses.
- Computer data processing refers to a process that takes the data input via a program and summarizes, analyse 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.**

1.6 Different Forms of Data Pre-processing

➜ **(SPPU - Oct. 16, Dec. 16)**

Q. What are the major tasks in data preprocessing ? Explain them in brief. Oct. 16, Dec. 16, 6 Marks

Q. Explain different steps in data preprocessing. (6 Marks)

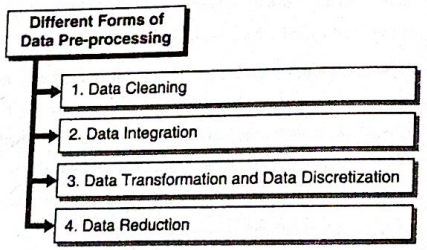


Fig. 1.6.1 : Different Forms of Data Pre-processing

Syllabus Topic : Data Cleaning

1.6.1 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.

☞ **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.
- Specialised tools are available which can be used for cleaning the data.
- Some of the Leading data cleansing vendors include Validity (Integrity), Harte-Hanks (Trillium) and First logic.

1.6.1(A) Steps in Data Cleansing

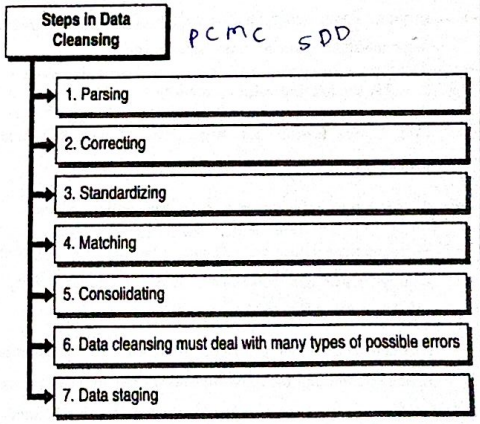


Fig. 1.6.2 : Steps in Data Cleansing

➜ **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.
- **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.
- **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.
- **Example :** addition of a prename, 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 analysing 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.

Syllabus Topic : Missing Values

1.6.1(B) Missing Values

➜ **(SPPU - May 16, Dec. 16, Dec. 17)**

Q. Describe the various methods for handling the missing values. **May 16, 6 Marks**

Q. In real-world data, tuples with missing values for some attributes are a common occurrence. Describe various methods for handling this problem. **Dec. 16, 6 Marks**

Q. What are missing values? Explain methods to handle missing values. **Dec. 17, 6 Marks**

☞ **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 :

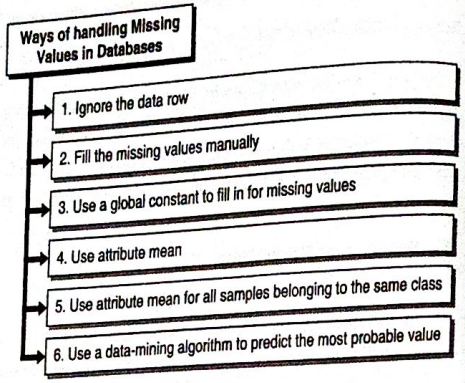


Fig. 1.6.3 : Ways of handling Missing Values in Databases

➜ **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.
- **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.
- **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.
- 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.
- **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.

Syllabus Topic : Noisy Data

1.6.1(C) 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**

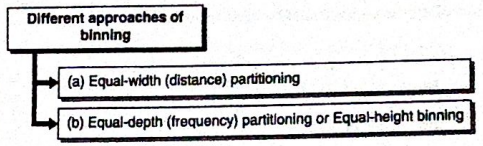


Fig. 1.6.4 : Different approaches of binning

➜ **(a) Equal-width (distance) partitioning**

- Divides the range into *N* intervals of equal size: uniform grid.

$$\text{bin width} = (\text{max value} - \text{min value}) / N$$

- **Example :** Consider a set of observed values in the range from 0 to 100.

The data could be placed into 5 bins as follows :

$$\text{width} = (100 - 0)/5 = 20$$

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.
- **Example :** 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

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.*

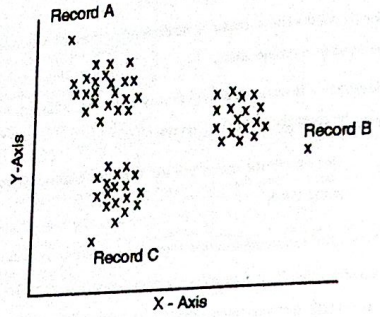


Fig. 1.6.5 : Graphical Example of Clustering

- Perform clustering on attributes values and replace all values in the cluster by a cluster representative.

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.
- 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.

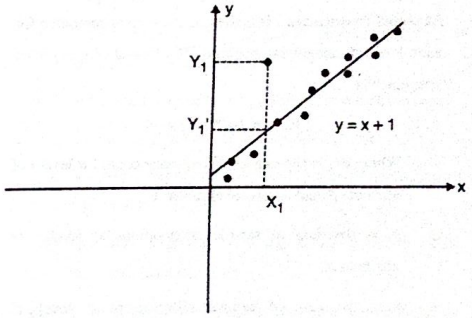


Fig. 1.6.6 : Regression example

1.6.1(D) 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

1.6.2 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.

1.6.2(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.

Syllabus Topic : Correlation Analysis

1.6.2(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 Er_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.

Syllabus Topic : Data Transformation and Data Discretization - Min-max Normalization, Z-Score Normalization and Decimal Scaling

1.6.3 Data Transformation and Data Discretization

1.6.3(A) Data Transformation

Q. Explain the data transformation in detail. (4 Marks)

- 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) Standardisation 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 summarisation 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

1.6.3(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.

1.6.3(C) Data Transformation by Normalization

➔ **(SPPU - May 17)**

Q. What are the different data normalization methods? Explain them in brief. May 17, 6 Marks

- 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 :

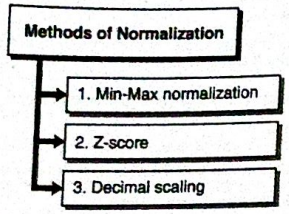


Fig. 1.6.7 : Methods of Normalization

➜ 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],

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

v = 73600 in [12000,98000]

v' = 0.716 in [0,1] (new range)

Ex. 1.6.1 : Consider the following group of data : 200, 300, 400, 600, 1000

(i) Use the min-max normalization to transform value 600 onto the range [0.0, 1.0]

(ii) Use the decimal scaling to transfer value 600. **SPPU - Oct. 16, 4 Marks**

Soln. :

(i) Min = Minimum value of the given data = 200

Max = Maximum value of the given data = 1000

$$V = 600 = \left(\frac{(V - min)}{(max - min)}\right) * (1 - 0) + 0$$

$$= \frac{600 - 200}{1000 - 200} = \left(\frac{400}{800}\right) * 1 = 0.5$$

(ii) Decimal scaling for 600

10^k is 10^3 = 1000

$\frac{600}{1000}$ = 0.6

➜ 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.

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

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)

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

Syllabus Topic : Discretization by Binning

1.6.3(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 1.6.1(C).

Syllabus Topic : Discretization by Histogram Analysis

1.6.3(E) Discretization by Histogram Analysis

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

☞ Different types of histogram

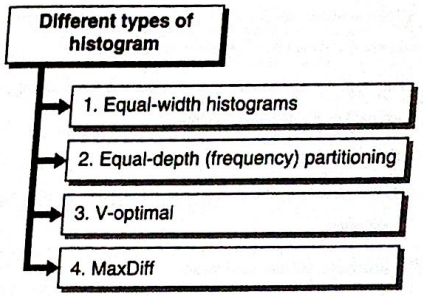


Fig. 1.6.8 : Different types of histogram

➜ 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,

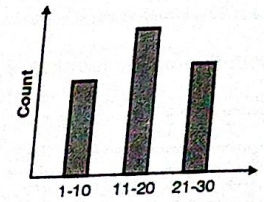


Fig. 1.6.9 : Example of histogram

Syllabus Topic : Data Reduction

1.6.4 Data Reduction

1.6.4(A) Need for Data Reduction

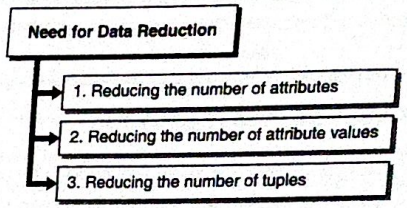


Fig. 1.6.10 : Need for Data Reduction

➜ 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.

1.6.4(B) Data Reduction Technique

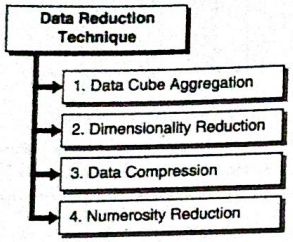


Fig. 1.6.11 : Data Reduction Technique

Syllabus Topic : Data Cube Aggregation, Attribute Subset Selection

1.6.4(B).1 Data Cube Aggregation

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

☞ **Example**

Total annual sales of TV in USA is aggregated quarterly as shown in Fig. 1.6.12.

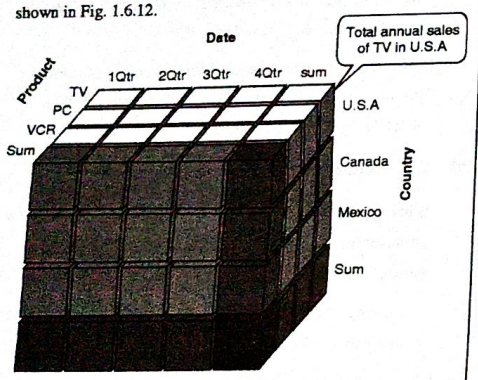


Fig. 1.6.12 : Example of data cube

1.6.4(B).2 Dimensionality Reduction

➜ **(SPPU - May 16, Dec. 17)**

Q. Enlist the dimensionality reduction techniques for text. Explain any one of them in brief. May 16, Dec. 17, 6 Marks

- 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**

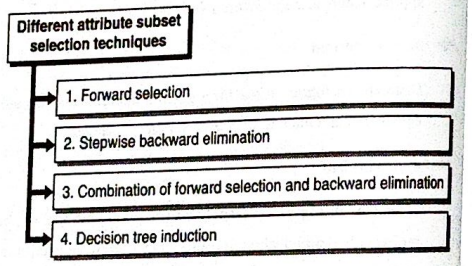


Fig. 1.6.13 : Different attribute subset selection techniques

➜ **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.

1.6.4(B).3 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.

☞ **Lossy 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 :

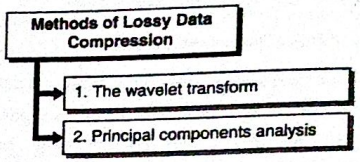


Fig. 1.6.14 : Methods of Lossy Data Compression

➜ **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 of data compression

- It also results in Effective removal of outliers.
- The technique is Cost efficient.
- Complexity O(N).
- At different 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(SVD) 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.

Syllabus Topic : Sampling

1.6.4(B).4 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 :

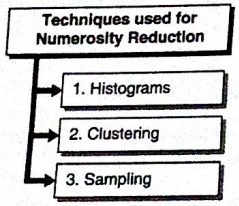


Fig. 1.6.15 : Techniques used for Numerosity Reduction

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 [Refer 1.6.3(E)]

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

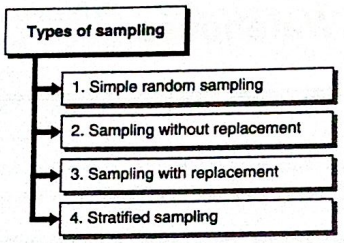


Fig. 1.6.16 : Types of sampling

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.

1.7 Solved University Questions and Answers

Q. 1 Discuss whether or not each of the following activities is a data mining task. **(May 16, 6 Marks)**

(i) Computing the total sales of a company.

(ii) Predicting the future stock price of a company using historical records.

(iii) Predicting the outcomes of tossing a pair of dice.

Ans. :

(i) Computing the total sales of a company

This activity is not a data mining task because the total sales can be computed by using simple calculations.

(ii) Predicting the future stock price of a company using historical records

This activity is a data mining task. Historical records of stock price can be used to create a predictive model called regression, one of the predictive modeling tasks that is used for continuous variables

(iii) Predicting the outcomes of tossing a pair of dice :

This activity is not a data mining task because predicting the outcome of tossing a fair pair of dice is a probability calculation, which doesn't have to deal with large amount of data or use complicate calculations or techniques.

Q. 2 Differentiate between Descriptive and Predictive data mining tasks **(Oct. 16, 2 Marks)**

Ans. :

(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

□□□

CHAPTER 2

Unit II

Data Warehouse

Syllabus Topics

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

Syllabus Topic : Data Warehouse

2.1 Data Warehouse

Q. Define Data Warehouse. (2 Marks)

- Precisely, a data warehouse system proves to be helpful in providing collective information to all its users. It is mainly created to support different analysis, queries that need extensive searching on a larger scale.
- With the help of Data warehousing technology, every industry right from retail industry to financial institutions, manufacturing enterprises, government department, airline companies people are changing the way they perform business analysis and strategic decision making.

The term Data Warehouse was defined by Bill Inmon in 1990, in the following way: "A warehouse is a subject-oriented, integrated, time-variant and non-volatile collection of data in support of management's decision making process". He defined the terms in the sentence as follows :

Subject Oriented

Data that gives information about a particular subject instead of about a company's ongoing operations.

Integrated

Data that is gathered into the data warehouse from a variety of sources and merged into a coherent whole.

Time-variant

All data in the data warehouse is identified with a particular time period.

Non-volatile

- Data is stable in a data warehouse. More data is added but data is never removed. This enables management to gain a consistent picture of the business.
- Ralph Kimball provided a much simpler definition of a data warehouse i.e. "data warehouse is a copy of transaction data specifically structured for query and analysis". This is a functional view of a data warehouse. Kimball did not address how the data warehouse is built like Inmondid, rather he focused on the functionality of a data warehouse.

Benefits of Data Warehousing

- **Potential high returns on investment and delivers enhanced business intelligence :** Implementation of data warehouse requires a huge investment in lakhs of rupees. But it helps the organization to take strategic decisions based on past historical data and organization can improve the results of various processes like marketing segmentation, inventory management and sales.
- **Competitive advantage :** As previously unknown and unavailable data is available in data warehouse, decision makers can access that data to take decisions to gain the competitive advantage.

- **Saves Time :** As the data from multiple sources is available in integrated form, business users can access data from one place. There is no need to retrieve the data from multiple sources.
- **Better enterprise intelligence :** It improves the customer service and productivity.
- **High quality data :** Data in data warehouse is cleaned and transferred into desired format. So data quality is high.

2.1.1 Features of Data Warehouse

Characteristics/ Features of a Data Warehouse

A common way of introducing data warehousing is to refer to the characteristics of a data warehouse :

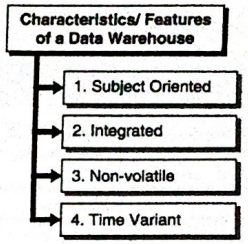


Fig. 2.1.1 : Characteristics/ Features of a Data Warehouse

1. Subject Oriented

- Data warehouses are designed to help analyze data. For example, to learn more about banking data, a warehouse can be built that concentrates on transactions, loans, etc.
- This warehouse can be used to answer questions like "Which customer has taken maximum loan amount for last year?" This ability to define a data warehouse by subject matter, loan in this case, makes the data warehouse subject oriented.

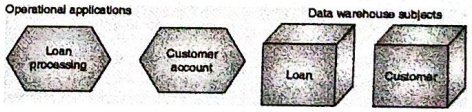


Fig. 2.1.2 : Data Warehouse is subject Oriented

2. Integrated

- A data warehouse is constructed by integrating multiple, heterogeneous data sources like, relational databases, flat files, on-line transaction records.
- The data collected is cleaned and then data integration techniques are applied, which ensures consistency in naming conventions, encoding structures, attribute measures etc. among different data sources.

Example

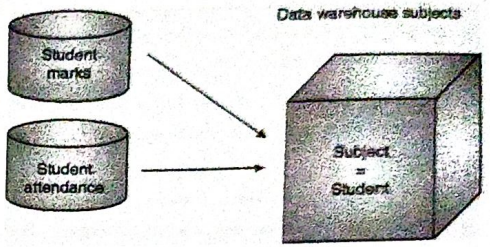


Fig. 2.1.3 : Integrated Data Warehouse

3. Non-volatile

Nonvolatile means that, once data entered into the warehouse, it cannot be removed or changed because the purpose of a warehouse is to analyze the data.

4. Time Variant

A data warehouse maintains historical data. For e.g. A customer record has details of his job, a data warehouse would maintain all his previous jobs (historical information) when compared to a transactional system which only maintains current job due to which its not possible to retrieve older records.

Syllabus Topic : Operational Database Systems and Data Warehouses (OLTP Vs OLAP)

2.2 Operational Database Systems and Data Warehouses (OLTP Vs OLAP)

2.2.1 Why are Operational Systems not Suitable for Providing Strategic Information?

- The fundamental reason for the inability to provide strategic information is that strategic information has been extracted from the existing operational systems.

- These operational systems such as University Record system, inventory management, claims processing, outpatient billing, and so on are not designed in a way to provide strategic information.
- If we need the strategic information, the information must be collected from altogether different types of systems. Only specially designed decision support systems or informational systems can provide strategic information.
- Operational systems are tuned for known transactions and workloads, while workload is not known a priori in a data warehouse.
- Special data organization, access methods and implementation methods are needed to support data warehouse queries (typically multidimensional queries)e.g., average amount spent on phone calls between 9AM-5PM in Pune during the month of December.

Operational Database System	Data Warehouse (or DSS -Decision Support System)
Application oriented	Subject oriented
Used to run business	Used to analyze business
Detailed data	Summarized and refined
Current up to date	Snapshot data
Isolated data	Integrated data
Repetitive access	Ad-hoc access
Clerical user	Knowledge user (manager)
Performance sensitive	Performance relaxed
Few records accessed at a time (tens)	Large volumes accessed at a time (millions)
Read/update access	Mostly read (batch update)
No data redundancy	Redundancy present
Database size 100 MB-100 GB	Database size 100GB – few terabytes

2.2.2 OLAP Vs OLTP

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

Definition

Definition given by OLAP council (www.olapcouncil.org) On-Line Analytical Processing (OLAP) is a category of software technology that enables analysts, managers and executives to gain insight into data through fast, consistent, interactive access in a wide variety of possible views of information that has been transformed from raw data to reflect the real dimensionality of the enterprise as understood by the user.

Application Differences

OLTP (On Line Transaction Processing)	OLAP (On-Line Analytical Processing)
Transaction oriented	Subject oriented
High Create/Read/Update/ Delete (CRUD) activity	High Read activity
Many users	Few users
Continuous updates – many sources	Batch updates – single source
Real-time information	Historical information
Tactical decision-making	Strategic planning
Controlled, customized delivery	"Uncontrolled", generalized delivery
RDBMS	RDBMS and/or MDBMS
Operational database	Informational database

Modeling Objectives Differences

OLTP	OLAP
High transaction volumes using few records at a time.	Low transaction volumes using many records at a time.
Balancing needs of online v/s scheduled batch processing.	Design for on-demand online processing.
Highly volatile data.	Non-volatile data.
Data redundancy – *BAD*.	Data redundancy – *GOOD*.
Few levels of granularity.	Multiple levels of granularity.
Complex database designs used by IT personnel.	Simpler database designs with business-friendly constructs.

Model Differences

OLTP	OLAP
Single purpose model – supports Operational System.	Multiple models – support Informational Systems.
Full set of Enterprise data.	Subset of Enterprise data.
Eliminate redundancy.	Plan for redundancy.
Natural or surrogate keys.	Surrogate keys.
Validate Model against business Function Analysis.	Validate Model against reporting requirements.
Technical metadata depends on business requirements.	Technical metadata depends on data mapping results.
This moment in time is important.	Many moments in time are essential elements.

Syllabus Topic : A Multidimensional Data Model

2.3 A Multidimensional Data Model

2.3.1 What is Dimensional Modeling ?

- It is a logical design technique used for data warehouses.
- Dimensional model is the underlying data model used by many of the commercial OLAP products available today in the market.
- Dimensional model uses the relational model with some important restrictions.
- It is one of the most feasible technique for delivering data to the end users in a data warehouse.
- Every dimensional model is composed of at least one table with a multipart key called the *fact table* and a set of other related tables called *dimension tables.*

Syllabus Topic : Data Cubes

2.3.2 Data Cubes

- Multidimensional models is used to inhabit data in multi-dimensional matrices like Data Cubes or Hypercubes. A standard spreadsheet, signifying a conventional database, is a two-dimensional matrix. One example would be a spreadsheet of regional sales by product for a particular time period. Products sold with respect to region can be shown in 2 dimensional matrix but as one more dimension like time is added then it produces 3 dimensional matrix as shown in Fig. 2.3.1.

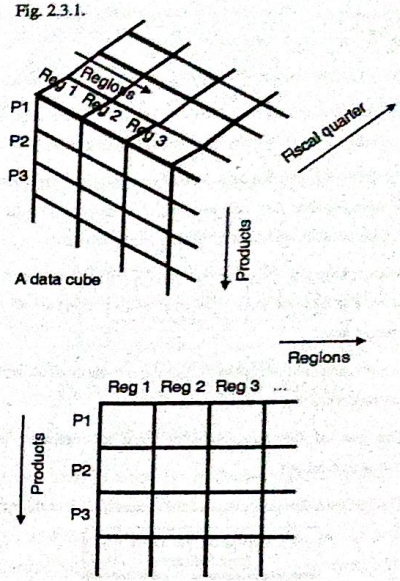


Fig. 2.3.1 : Pictorial view of data cube and 2D database

- A multidimensional model has two types of tables :
  1. **Dimension tables :** contains attributes of dimensions
  2. **Fact tables :** contains facts or measures

Syllabus Topic : Star Schema

2.3.3 Star Schema

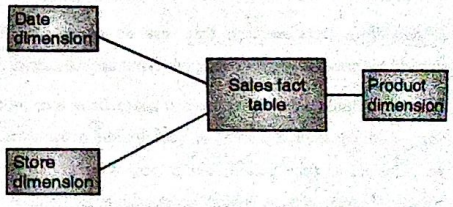


Fig. 2.3.2 : Examples of Star Schema

- Star Schema is the most popular schema design for a data warehouse.
- Dimensions are stored in a Dimension table and every entry has its own unique identifier.
- Every Dimension table is related to one or more fact tables.
- All the unique identifiers (primary keys) from the dimension tables make up for a composite key in the fact table.
- The fact table also contains facts. For example, a combination of store_id, date_key and product_id giving the amount of a certain product sold on a given day at a given store.
- Foreign keys for the dimension tables are contained in a fact table. For eg.(date key, product id and store_id) are all three foreign keys.
- In a dimensional modeling fact tables are normalised, whereas dimension tables are not.
- The size of the fact tables is large as compared to the dimension tables.
- The Facts in the star schema can be classified into three types.

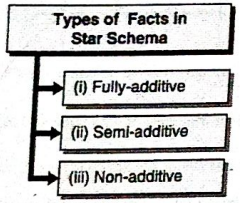


Fig. 2.3.3 : Types of Facts in Star Schema

➜ **(i) Fully-additive**

- Additive facts are facts that can be summed up through all of the dimensions in the fact table.

➜ **(ii) Semi-additive**

- Semi-additive facts are facts that can be summed up for some of the dimensions in the fact table, but not the others.

Example : Bank Balances : You can take a bank account as Semi- Additive since a current balance for the account can't be summed as time period; but if you want see current balance of a bank you can sum all accounts current balance.

➜ **(iii) Non-additive**

- Non-additive facts are facts that cannot be summed up for any of the dimensions present in the fact table.

E.g. : Ratios, Averages and Variance

☞ **Advantages of Star Schema**

- A star schema describes aspects of a business. It is made up of multiple dimension tables and one fact table. For e.g. if you have a book selling business, some of the dimension tables would be customer, book, catalog and year. The fact table would contain information about the books that are ordered from each catalog by each customer during a particular year.
- Reduced Joins, Faster Query Operation.
- It is fully denormalized schema.
- Simplest DW schema.
- Easy to understand.
- Easy to Navigate between the tables due to less number of joins.
- Most suitable for Query processing.

Syllabus Topic : The Snowflake Schema

2.3.4 The Snowflake Schema

- A snowflake schema is used to remove the low cardinality i.e attributes having low distinct values, textual attributes from a dimension table and placing them in a secondary dimension table.
- For e.g. in Sales Schema, the product category in the product dimension table can be removed and placed in a secondary dimension table by normalizing the product dimension table. This process is carried out on large dimension tables.
- It is a normalization process carried out to manage the size of the dimension tables. But this may affect its performance as joins needs to be performed.
- In a star schema, if all the dimension tables are normalised then this schema is called as snowflake schema, and if only few of the dimensions in a star schema are normalised then it is called as star flake schema.

2.3.5 Star Flake Schema

- It is a hybrid structure (i.e. star schema + snowflake schema).
- Every fact points to one tuple in each of the dimensions and has additional attributes.
- Does not capture hierarchies directly.
- Straightforward means of capturing a multiple dimension data model using relations.

2.3.6 Differentiate between Star Schema and Snowflake Schema

Sr. No.	Star Schema	Snowflake Schema
1.	Star schema contains the dimension tables mapped around one or more fact tables.	A Snowflake schema contains in-depth joins because the tables are split in to many pieces.
2.	It is a de-normalized model.	It is the normalised form of Star schema.
3.	No need to use complicated joins.	Have to use complicated joins, since it has more tables.
4.	Queries results fast.	There will be some delay in processing the Query.
5.	Star Schemas are usually not in BCNF form. All the primary keys of the dimension tables are in the fact table.	In Snowflake schema, dimension tables are in 3NF, so there are more dimension tables which are linked by primary – foreign key relation.

2.3.7 Factless Fact Table

- Factless table means only the key available in the Fact there is no measures available.
- Used only to put relation between the elements of various dimensions.
- Are useful to describe events and coverage, i.e. the tables contain information that something has/has not happened.
- Often used to represent many-to-many relationships.
- The only thing they contain is a concatenated key, they do still however represent a focal event which is identified by the combination of conditions referenced in the dimension tables.
- An Example of Factless fact table can be seen in the Fig. 2.3.4.

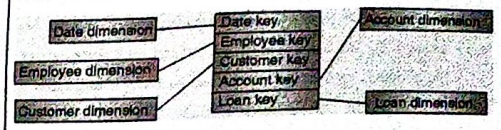


Fig. 2.3.4 : A Factless Fact Table

- There are two main types of factless fact tables :

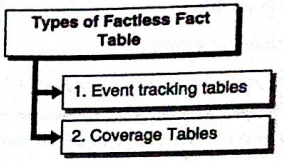


Fig. 2.3.5 : Types of Factless Fact Table

➜ **1. Event tracking tables**

- Use a factless fact table to track events of interest to the organization. For example, attendance at a cultural event can be tracked by creating a fact table containing the following foreign keys (i.e. links to dimension tables): event identifier, speaker/entertainer identifier, participant identifier, event type, date. This table can then be queried to find out information; such as which cultural events or event types are the most popular.
- Following example shows factless fact table which records every time a student attends a course or Which class has the maximum attendance? Or What is the average number of attendance of a given course?
- All the queries are based on the COUNT() with the GROUP BY queries. So we can first count and then apply other aggregate functions such as AVERAGE, MAX, MIN.

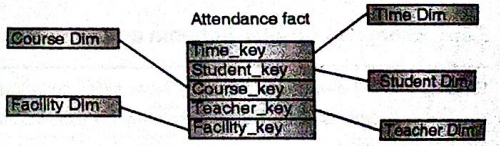


Fig. 2.3.6 : Example of Event Tracking Tables

2. Coverage Tables

The other type of factless fact table is called Coverage table by Ralph. It is used to support negative analysis report. For example a Store that did not sell a product for a given period. To produce such report, you need to have a fact table to capture all the possible combinations. You can then figure out what is missing.

Common examples of factless fact table

- Ex-Visitors to the office.
- List of people for the web click.
- Tracking student attendance or registration events.

Syllabus Topic : Fact Constellation Schema or Families of Star

2.3.8 Fact Constellation Schema or Families of Star

Fact Constellation

- As its name implies, it is shaped like a constellation of stars (i.e., star schemas).
- This schema is more complex than star or snowflake varieties, which is due to the fact that it contains multiple fact tables.
- This allows dimension tables to be shared amongst the fact tables.
- A schema of this type should only be used for applications that need a high level of sophistication.
- For each star schema or snowflake schema it is possible to construct a fact constellation schema.
- That solution is very flexible, however it may be hard to manage and support.
- The main disadvantage of the fact constellation schema is a more complicated design because many variants of aggregation must be considered.
- In a fact constellation schema, different fact tables are explicitly assigned to the dimensions, which are for given facts relevant.
- This may be useful in cases when some facts are associated with a given dimension level and other facts with a deeper dimension level.
- Use of that model should be reasonable when for example, there is a sales fact table (with details down to the exact date and invoice header id) and a fact table with sales forecast which is calculated based on month, client id and product id.
- In that case using two different fact tables on a different level of grouping is realized through a fact constellation model.

Family of stars

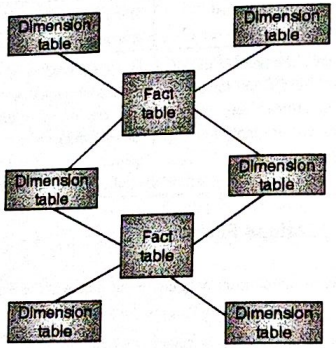


Fig. 2.3.7 : Family of stars

2.3.9 Examples on Star Schema and Snowflake Schema

Ex. 2.3.1 : *All electronics company have sales department. Sales consider four dimensions namely time, item, branch and location. The schema contains a central fact table sales with two measures dollars_sold and unit_sold. Design star schema, snowflake schema and fact constellation for same.*

Soln. :

(a) Star Schema

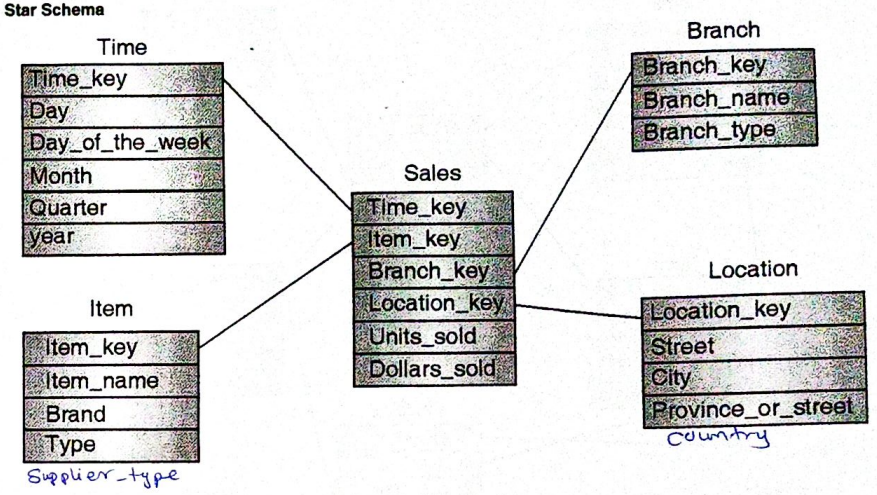


Fig. P. 2.3.1 : Sales Star Schema

(b) Snowflake Schema

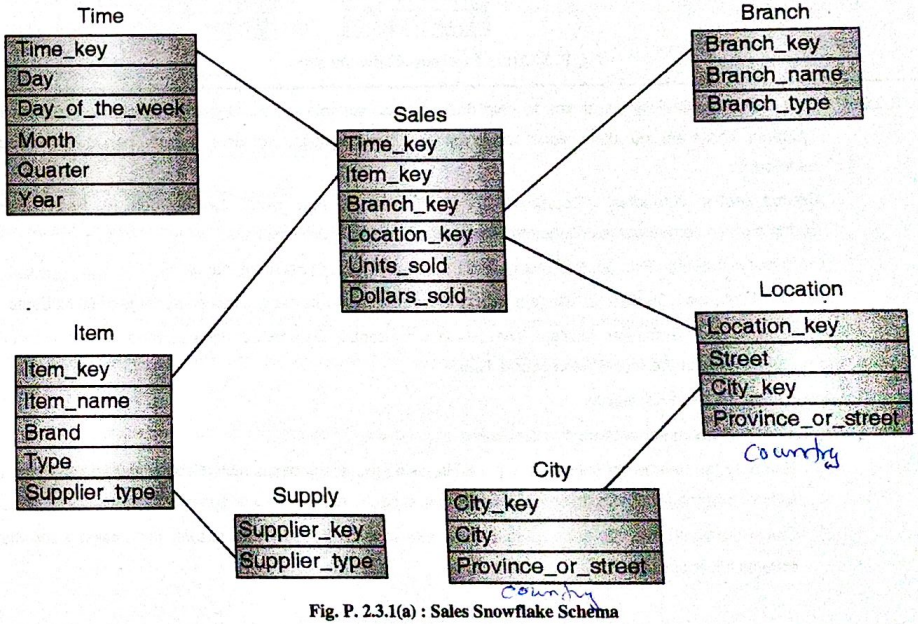


Fig. P. 2.3.1(a) : Sales Snowflake Schema

(c) Fact Constellation

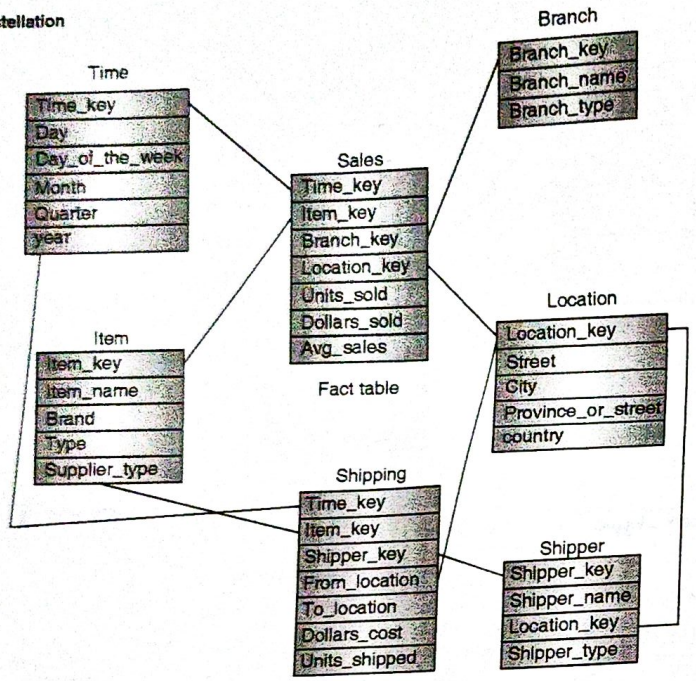


Fig. P. 2.3.1(b) : Fact constellation for sales

Ex. 2.3.2 : The Mumbai university wants you to help design a star schema to record grades for course completed by students. There are four dimensional tables namely course_section, professor, student, period with attributes as follows :

Course_section Attributes : Course_Id, Section_number, Course_name, Units, Room_id, Roomcapacity. During a given semester the college offers an average of 500 course sections

Professor Attributes : Prof_id, Prof_Name, Title, Department_id, department_name

Student Attributes : Student_id, Student_name, Major. Each Course section has an average of 60 students

Period Attributes : Semester_id, Year. The database will contain Data for 30 months periods. The only fact that is to be recorded in the fact table is course Grade

Answer the following Questions

(a) Design the star schema for this problem

(b) Estimate the number of rows in the fact table, using the assumptions stated above and also estimate the total size of the fact table (in bytes) assuming that each field has an average of 5 bytes.

(c) Can you convert this star schema to a snowflake schema ? Justify your answer and design a snowflake schema if it is possible.

Soln. :

(a) Star Schema

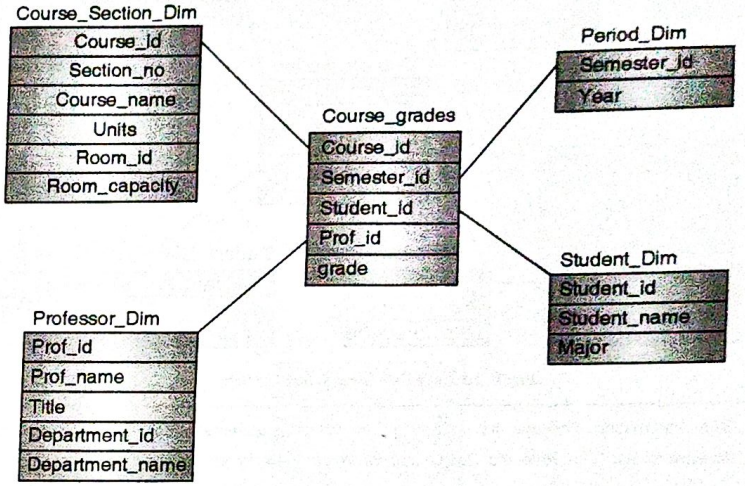


Fig. P. 2.3.2 : University Star Schema

(b) Total Courses Conducted by university = 500

Each Course has average students = 60

University stores data for 30 months

Total Student in University for all courses in 30 months = 500*60 = 30000

Time Dimension = 30 months = 5 Semesters (Assume 1 semester = 6 months)

Now, Number of rows of fact table = 30000*5 = 150000 (one student has 5 grades for 5 semesters)

(c) Snowflake Schema

- Yes, the above star schema can be converted to a snowflake schema, considering the following assumptions
- Courses are conducted in different rooms, so course dimension can be further normalized to rooms dimension as shown in the Fig. P. 2.3.2(a).
- Professor belongs to a department, and department dimension is not added in the star schema, so professor dimension can be further normalized to department dimension.
- Similarly students can have different major subjects, so it can also be normalized as shown in the Fig. P. 2.3.2(a).

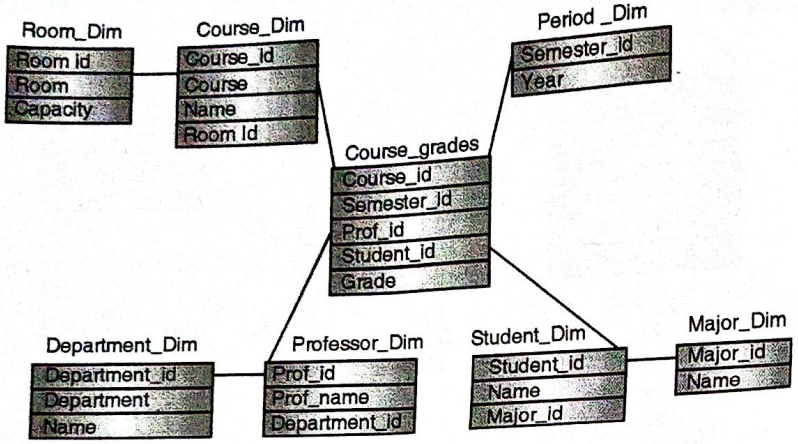


Fig. P. 2.3.2(a) : University Snowflake Schema

Ex. 2.3.3 : Give Information Package for recording information requirements for "Hotel Occupancy" considering dimensions like Time, Hotel etc. Design star schema from the information package.

Soln. :

Information package diagram

- Information package diagram is the approach to determine the requirement of data warehouse.
- It gives the metrics which specifies the business units and business dimensions.
- The information package diagram defines the relationship between the subject or dimension matter and key performance measures (facts).
- The information package diagram shows the details that users want so its effective for communication between the user and technical staff.

Table P. 2.3.3 : Information Package for Hotel Occupancy

Hotel	Room Type	Time	Room Status
Hotel Id	Room id	Time id	Status id
Branch Name	room type	Year	Status Description
Branch Code	room size	Quarter	
Region	number of beds	Month	
Address	type of bed	Date	
city/stat/zip	max occupants	day of week	
construction year	Suite	day of month,	
renovation year		holiday flag	

Facts

(a) Occupied Rooms

(b) Vacant Rooms

(c) Unavailable Rooms

(d) No of occupants

(e) Revenue

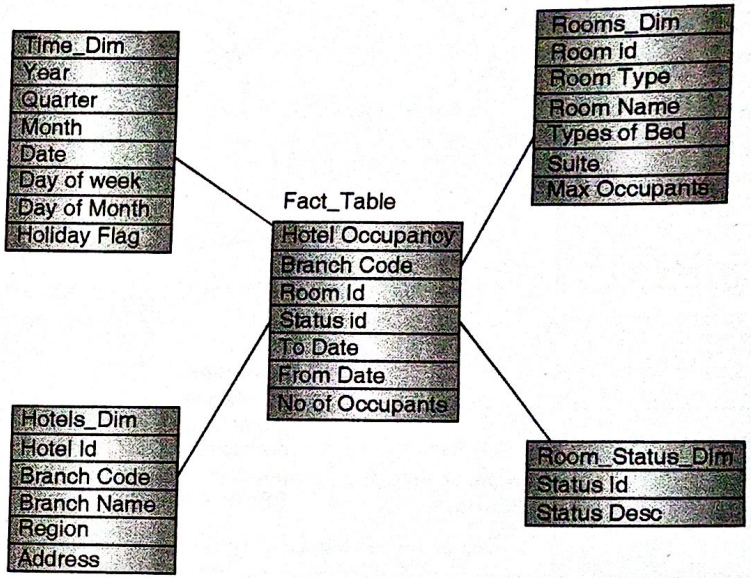


Fig. P. 2.3.3 : Hotel Occupancy Star Schema

Ex. 2.3.4 : For a Supermarket Chain consider the following dimensions, namely Product, store, time , promotion. The schema contains a central fact tables sales facts with three measures unit_sales, dollars_sales and dollar_cost.

Design star schema and calculate the maximum number of base fact table records for the values given below :

Time period : 5 years

Store : 300 stores reporting daily sales

Product : 40,000 products in each store(about 4000 sell in each store daily)

Promotion : a sold item may be in only one promotion in a store on a given day

Soln. :

(a) Star schema

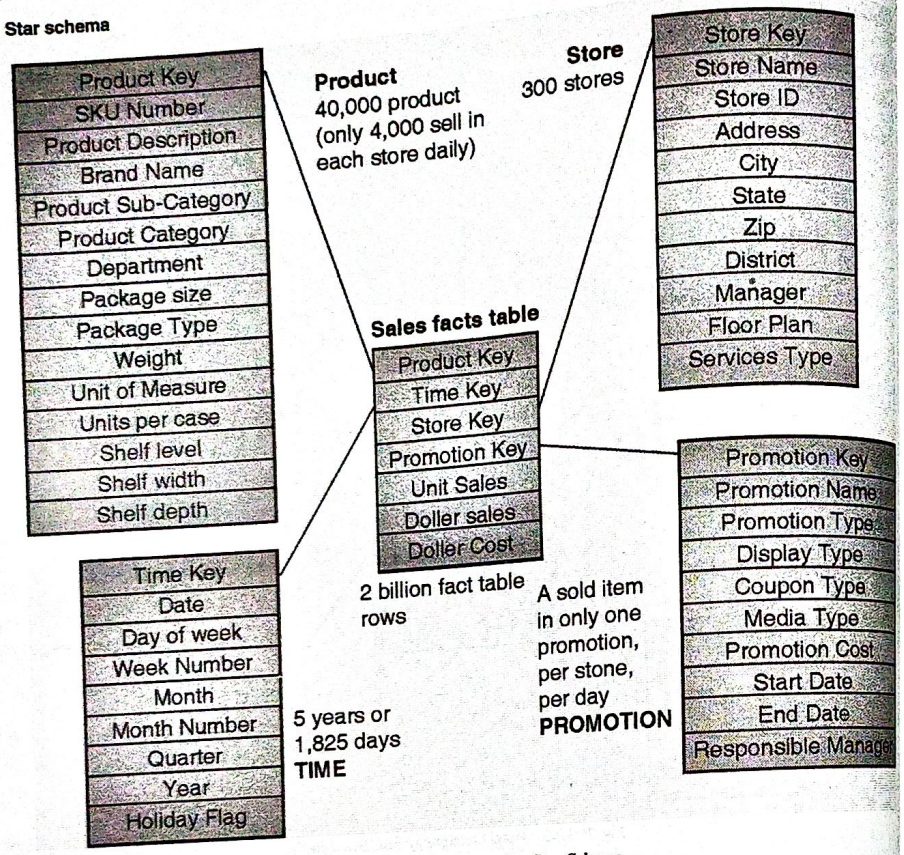


Fig. P. 2.3.4 : Sales Promotion Star Schema

(b) Time period = 5 years × 365 days = 1825

There are 300 stores.

Each stores daily sale = 4000 ; Promotion = 1

Maximum number of fact table records : 1825 × 300 × 4000 × 1= 2 billion

Ex. 2.3.5 : Draw a Star Schema for Student academic fact database.

Soln. :

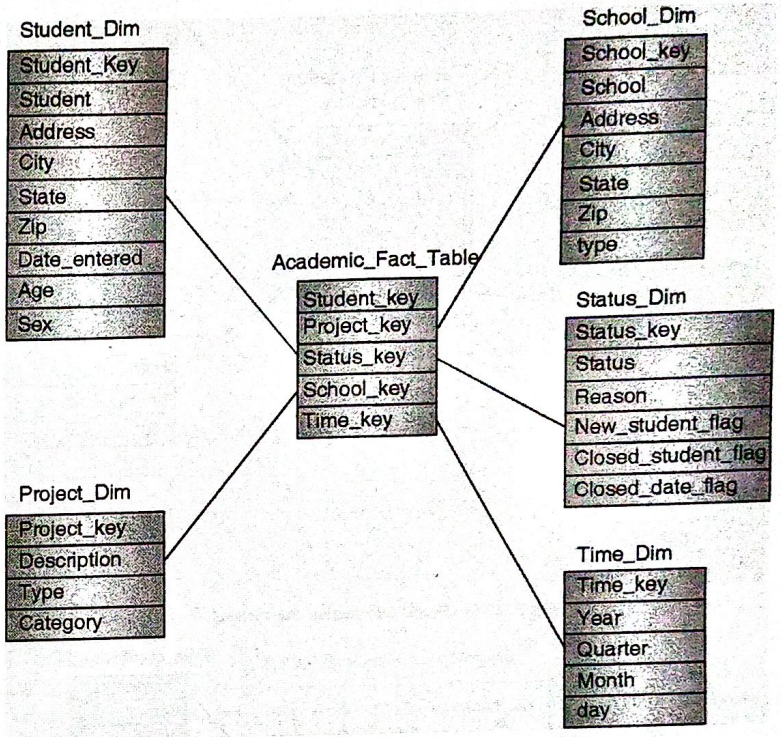


Fig. P. 2.3.5 : Student Academic Star Schema

Ex. 2.3.6 : List the dimensions and facts for the Clinical Information System and Design Star and Snow Flake Schema.

Soln. :

Dimensions

1. Patient
2. Doctor
3. Procedure
4. Diagnose
5. Date of Service
6. Location
7. Provider

Facts

1. Adjustment
2. Charge
3. Age

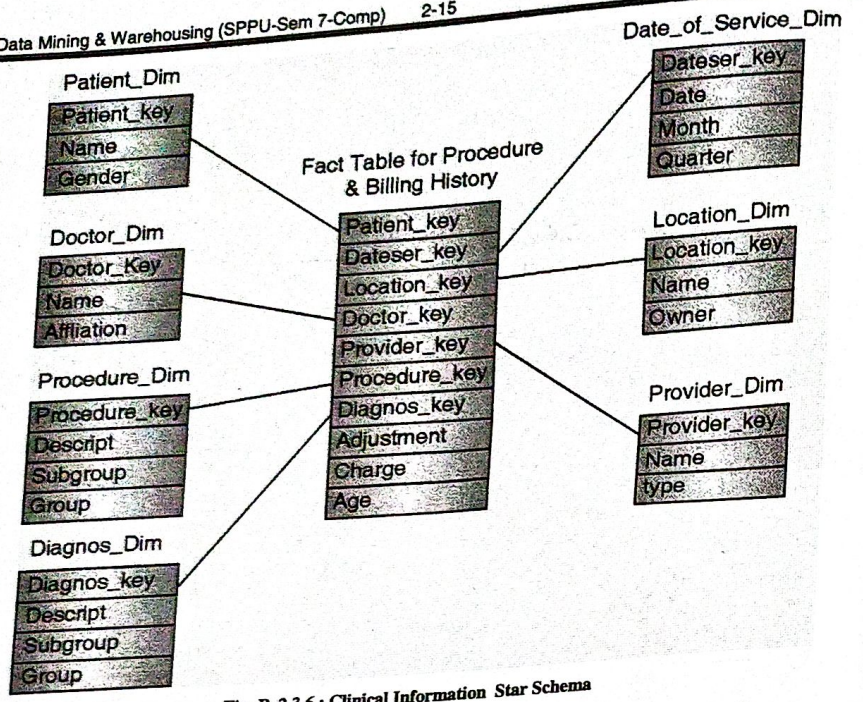


Fig. P. 2.3.6 : Clinical Information Star Schema

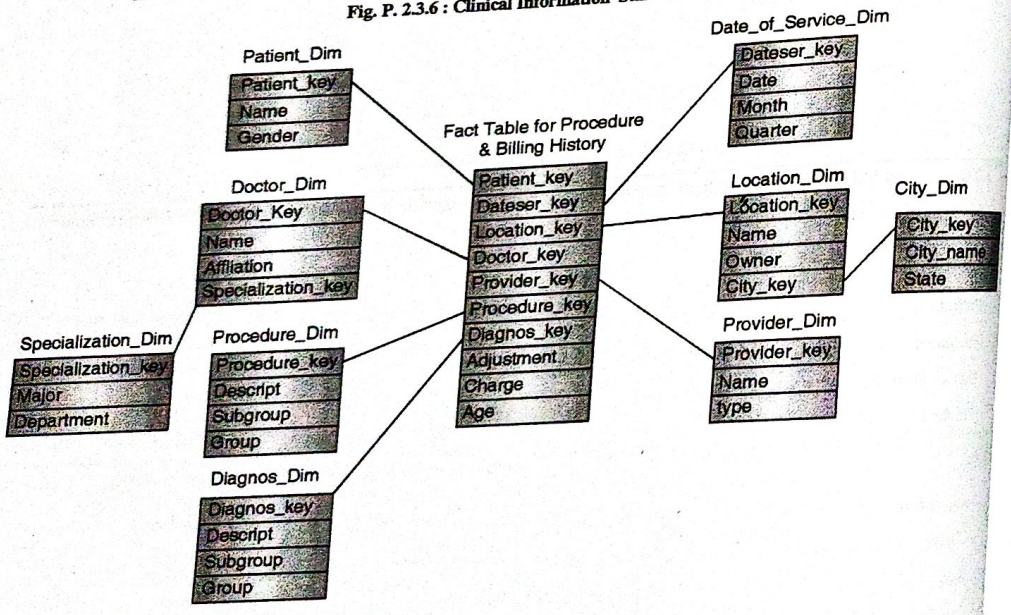


Fig. P. 2.3.6(a) : Clinical Information Snow Flake Schema

Ex. 2.3.7 : Draw a Star Schema for Library Management.

Soln. :

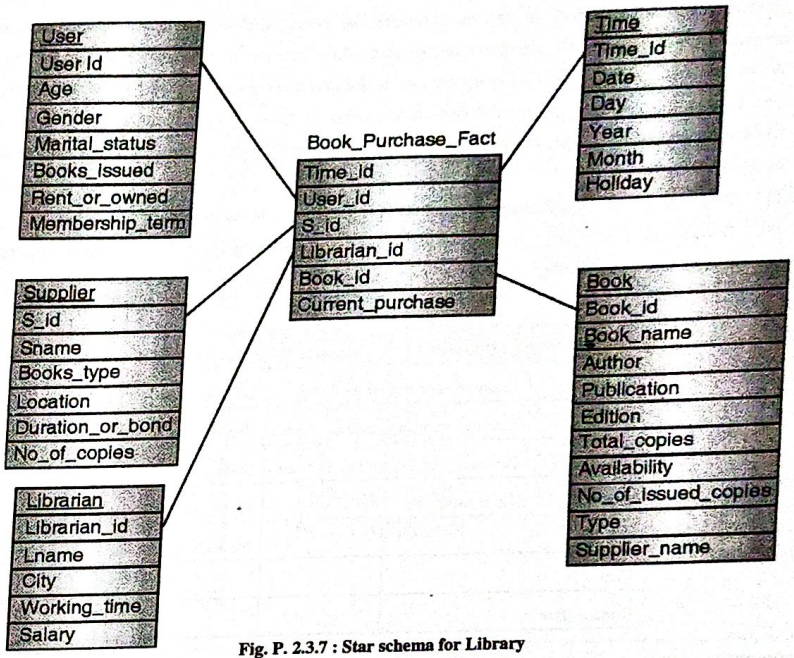


Fig. P. 2.3.7 : Star schema for Library

Ex. 2.3.8 : A manufacturing company has a huge sales network. To control the sales, it is divided in the regions. Each region has multiple zones. Each zone has different cities. Each sales person is allocated different cities. The object is to track sales figure at different granularity levels of region. Also to count no. Of products sold. Create data warehouse schema to take into consideration of above granularity levels for region, sales person and the quarterly, yearly and monthly sales.

Soln. :

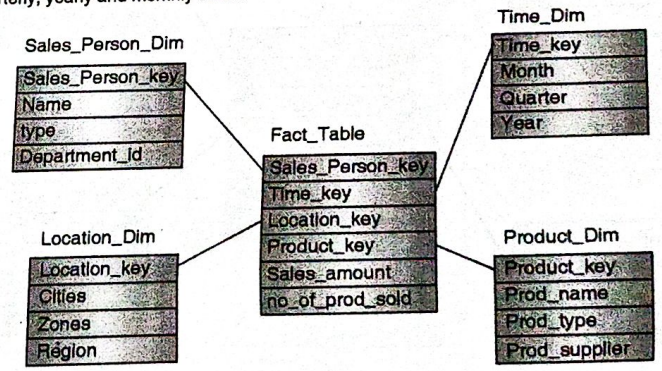


Fig. P. 2.3.8 : Star schema for Sales

Ex. 2.3.9 : A bank wants to develop a data warehouse for effective decision-making about their loan schemes. The ba provides loans to customers for various purposes like House Building loan, car loan, educational loa personal loan etc. The whole country is categorized into a number of regions, namely, North, South, Ea West. Each region consists of a set of states; loan is disbursed to customers at interest rates that change fr time to time. Also, at any given point of time, the different types of loans have different rates. That di warehouse should record an entry for each disbursement of loan to customer. With respect to the abo business scenario.

(i) Design an information package diagram. Clearly explain all aspects of the diagram.

(ii) Draw a star schema for the data warehouse clearly identifying the fact tables, dimension tables, a attributes and measures.

Soln. : (i)

Time	Customer	Branch	Location
Time_key	Customer_key	Branch_key	Location_key
Day	Account_number	Branch_Area	Region
Day_of_week	Account_type	Branch_home	State
Month	Loan_type		City
Quarter			Street
Year			
Holiday_flag			

(ii) Star Schema for a Bank

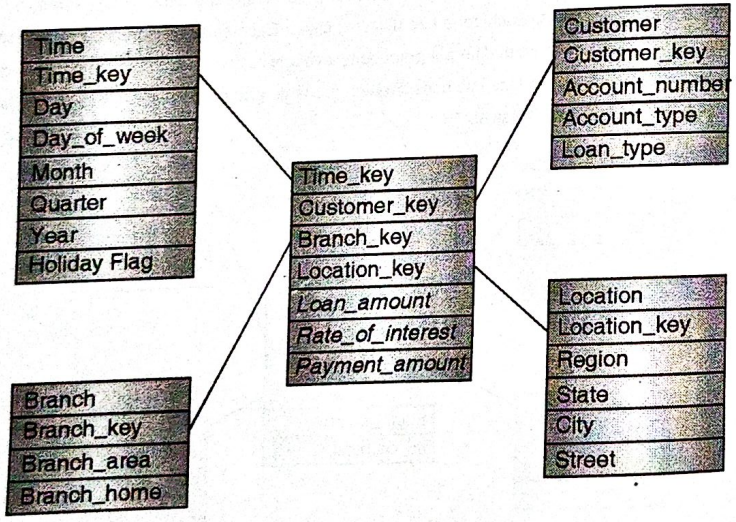
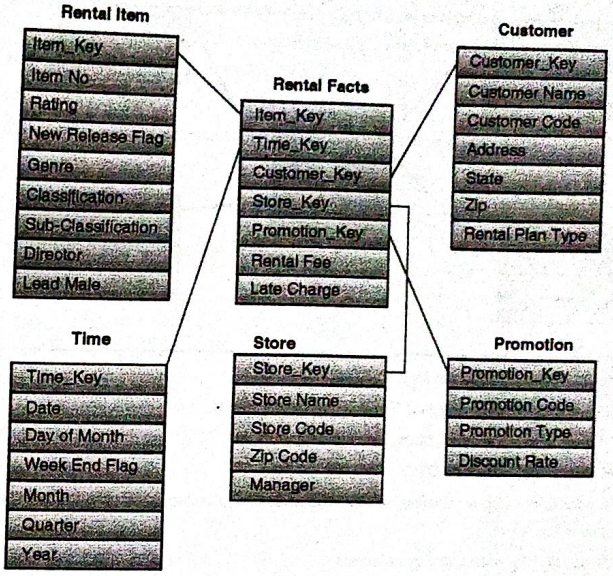


Fig. P. 2.3.9 : Star schema for Bank

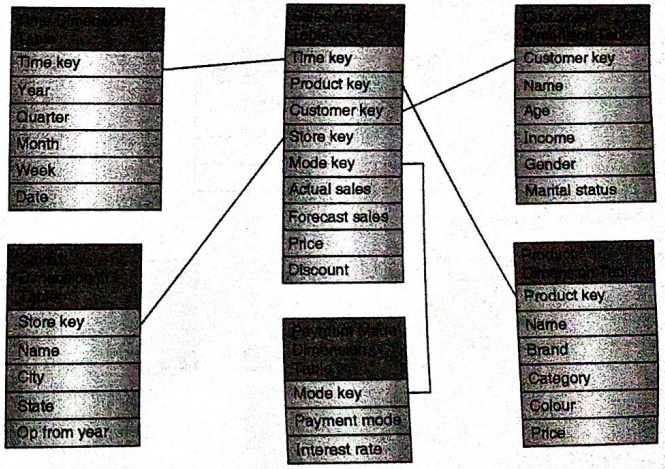
Ex.2.3.10 : Draw star schema for video Rental.

Soln. :



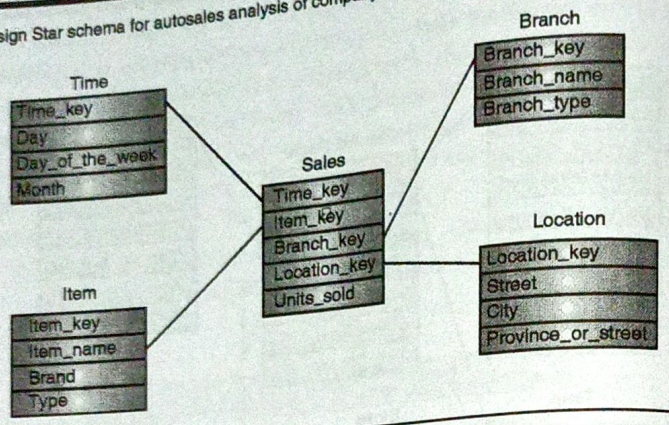
Ex. 2.3.11 : Draw star schema for retail chain.

Soln. :



Ex. 2.3.12 : Design Star schema for autosales analysis of company.

Soln. :



Ex. 2.3.13 : Consider the following database for a chain of bookstores.

BOOKS (Booknum, Primary_Author, Topic, Total_Stock, Price)

BOOKSTORE (Storenum, City, State, Zip, inventory_Value)

STOCK (Storenum, Booknum, QTY)

With respect to the above business scenario, answer the following questions. Clearly state any reason assumptions you make.

(a) Design an information package diagram.

(b) Design a star schema for the data warehouse clearly identifying the fact tables(s), Dimension tabl their attributes and measures.

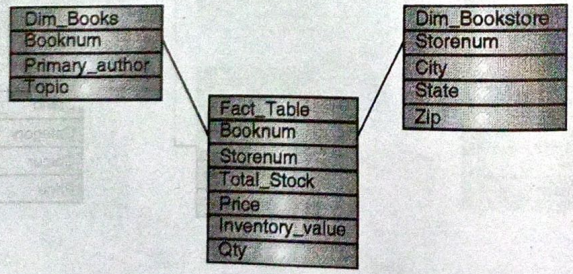
Soln. :

a) Information Package Diagram

BOOKS	BOOKSTORE
Booknum	Storenum
Primary_author	City
Topic	State
	Zip

Facts : Total_Stock , Price, Inventory_value , Qty

b) Star Schema



Ex. 2.3.14 : One of India's large retail departmental chains, with annual revenues touching $2.5 billion mark and having over 3600 employees working at diverse locations, was keenly interested in a business intelligence solution that can bring clear insights on operations and performance of departmental stores across the retail chain. The company needed to support a data warehouse that exceeds daily sales data from Point of Sales (POS) across all locations, with 80 million rows and 71 columns.

(a) List the dimensions and facts for above application.

(b) Design star schema and snow flake schema for the above application.

Soln. :

a) Dimensions : Product, Store, Time, Location

Facts : Unit Sales, Dollar Sales, Dollar Cost

b) Star Schema and snowflake Schema

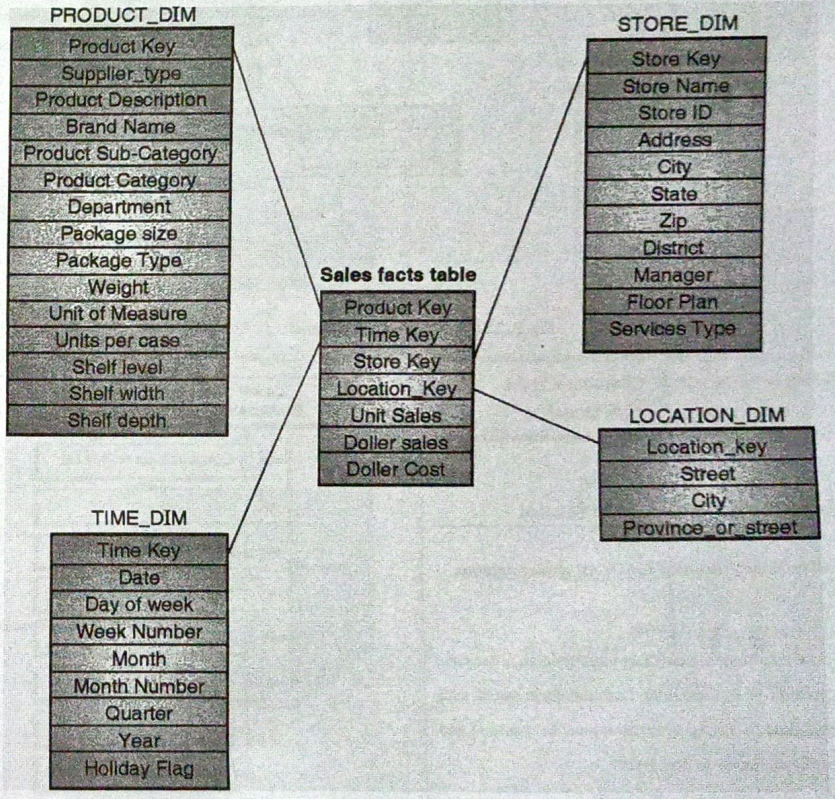


Fig. P. 2.3.14 : Star Schema

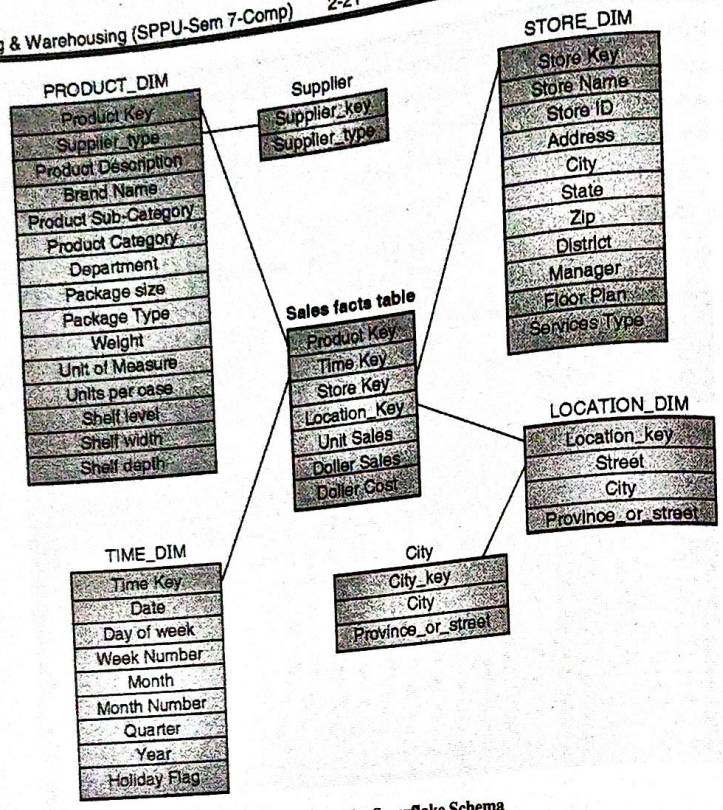


Fig. P. 2.3.14(a) : Snowflake Schema

Syllabus Topic : OLAP Operations in the Multidimensional Data Model

2.4 OLAP Operations in the Multidimensional Data Model

The following techniques are used for OLAP implementation

Example

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

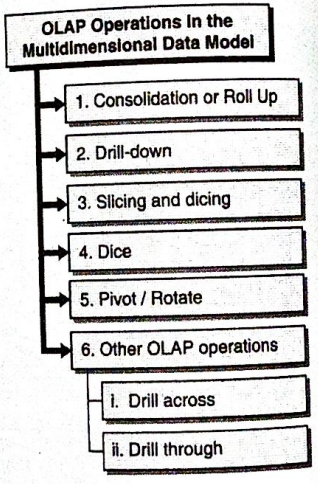


Fig. 2.4.1 : OLAP Operations in the Multidimensional Data Model

1. Consolidation or Roll Up

- Multi-dimensional databases generally have hierarchies with respect to dimensions.
- Consolidation is rolling up or adding data relationship with respect to one or more dimensions.
- For example, adding up all product sales to get total City data.
- For example, Fig. 2.4.2 shows the result of roll up operation performed on the central cube by climbing up the concept hierarchy for location.
- This hierarchy was defined as the total order street <city <province_or_state<country.
- The roll up operation shown aggregates the data by city to the country by location hierarchy.

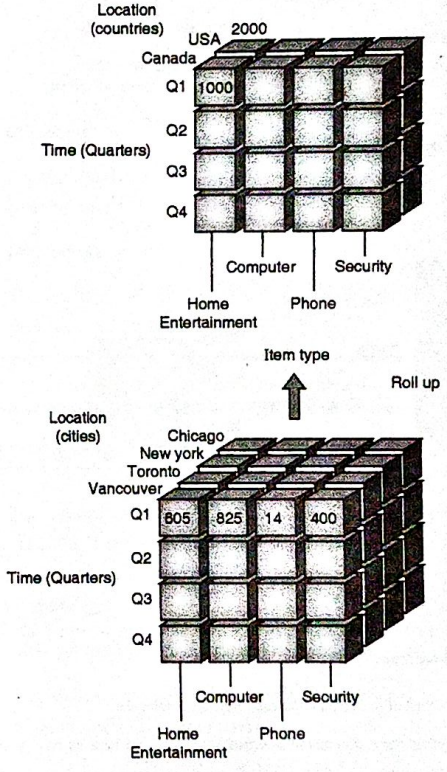


Fig. 2.4.2 : Roll -up or drill up

2. Drill-down

- Drill Down is defined as changing the view of data to a greater level of detail.
- For example, the Fig. 2.4.3 shows the result of drill down operations performed on the upper cube by stepping down a concept hierarchy for time defined as day<month<quarter<year.

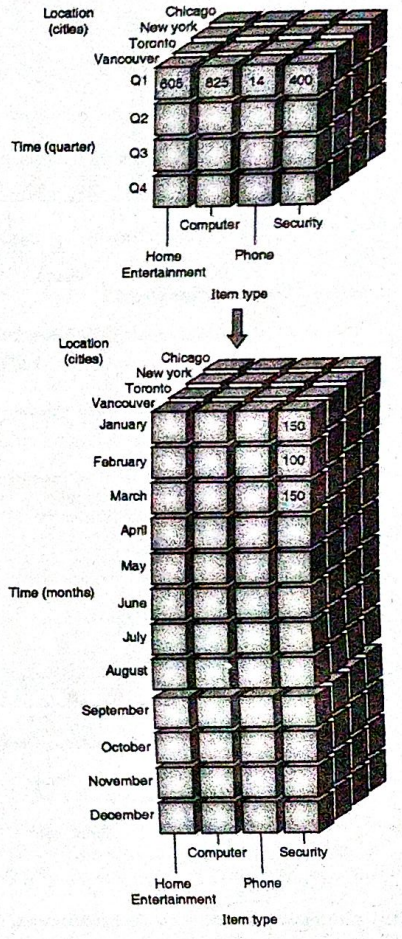


Fig. 2.4.3 : Drill Down

3. Slicing and dicing

- Slicing and dicing refers to the ability to look at a database from various viewpoints.
- Slice operation carry out selection with respect to one dimension of the given cube and produces a sub cube.
- For example, Fig. 2.4.4 shows the slice operation where the sales data are selected from the left cube for the dimension time using criterion time = "Q1"

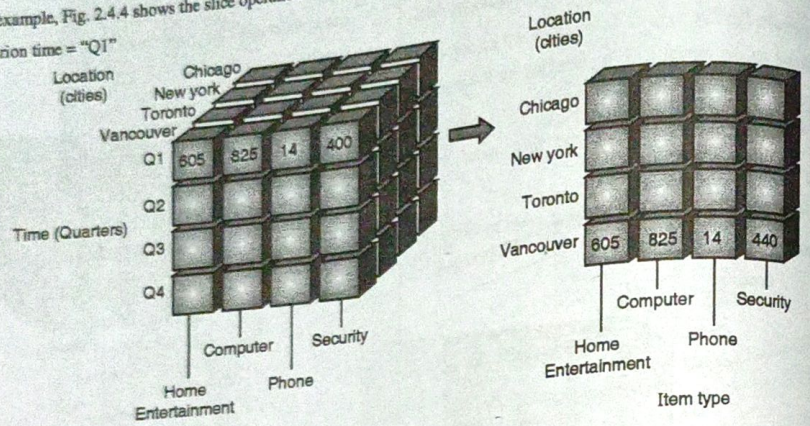


Fig. 2.4.4 : Slice operation

4. Dice

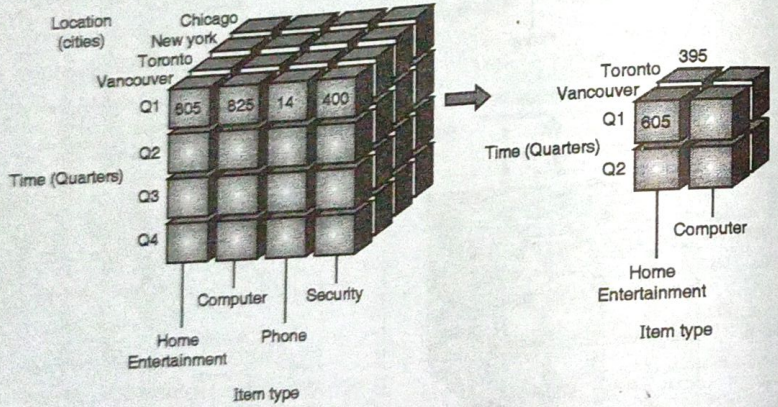


Fig. 2.4.5 : Dice operation

- Dice operation carry out selection with respect to two or more dimensions of the given cube and produces a sub cube.
- For example, the dice operation is performed on the left cube based on three dimension as Location, Time and Item as shown in Fig. 2.4.5 where the criteria is (location= "Torrento" or "Vancouver") and (time = "Q1" or "Q2") and (item= "home entertainment" or "computer").

5. Pivot / Rotate

- Pivot technique is used for visualization of data. This operation rotates the data axis to give another presentation of the data.
- For example Fig. 2.4.6 shows the pivot operation where the item and location axis in a 2-D slice are rotated.

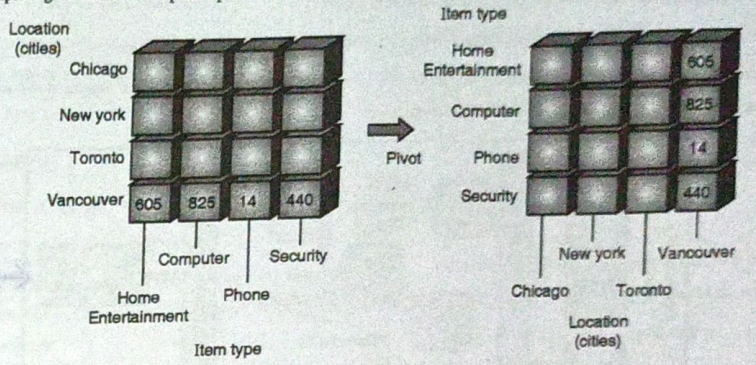


Fig. 2.4.6 : Pivot operation

6. Other OLAP operations

Drill across

This technique is used when there is need to execute a query involving more than one fact table.

Drill through

This technique uses relational SQL facilities to drill through the bottom level of the data cube.

Syllabus Topic : Concept Hierarchies

2.5 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).

Detailed data → more generalized data

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 analysing 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

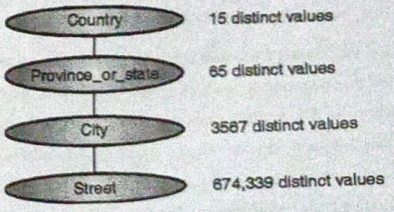


Fig. 2.5.1 : Concept hierarchy example

Syllabus Topic : Data Warehouse Architecture

2.6 Data Warehouse Architecture

- The data in a data warehouse comes from operational systems of the organization as well as from other external sources. These are collectively referred to as ***source systems***.

- The data *extracted* from source systems is stored in an area called *data staging area*, where the data is cleaned, *transformed*, combined, and duplicated to prepare the data in the data warehouse.
- The data staging area is generally a collection of machines where simple activities like sorting and sequential processing takes place. The data staging area does not provide any query or presentation services.
- As soon as a system provides query or presentation services, it is categorized as a *presentation server*.
- A presentation server is the target machine on which the data is *loaded* from the data staging area organized and stored for direct querying by end users, report writers and other applications.

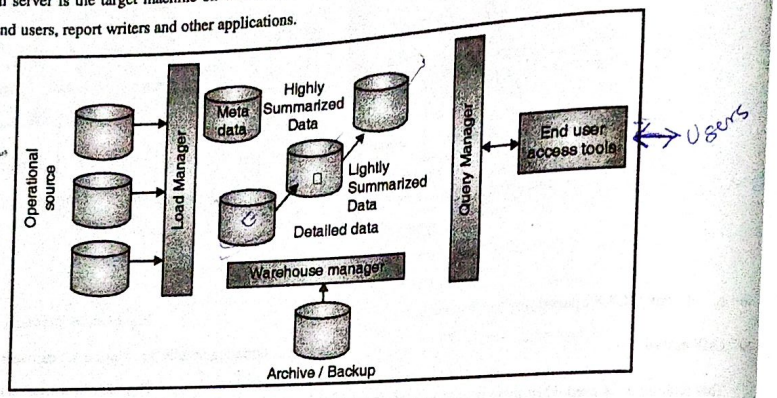


Fig. 2.6.1 : Data Warehouse Architecture

- The three different kinds of systems that are required for a data warehouse are :
  (i) Source Systems (ii) Data Staging Area
  (iii) Presentation servers
- The data travels from source systems to presentation servers via the data staging area. The entire process is popularly known as ETL (extract, transform, and load) or ETT (extract, transform, and transfer). Oracle's ETL tool is called Oracle Warehouse Builder (OWB) and MS SQL Server's ETL tool is called Data Transformation Services (DTS).
- A typical architecture of a data warehouse is shown in Fig. 2.6.1.

Each component and the tasks performed by them are explained below :

1. Operational Source

The sources of data for the data warehouse are supplied from :

- The data from the mainframe systems in the traditional network and hierarchical format.
- Data can also come from the 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 supplier and customers.

2. Load Manager

- The load manager performs all the operations associated with extraction and loading data into the data warehouse.
- These operations include simple transformations of the data to prepare the data for entry into the warehouse.
- The size and complexity of this component will vary between data warehouses and may be constructed using a combination of vendor data loading tools and custom built programs.

3. Warehouse Manager

- The warehouse manager performs all the operations associated with the management of data in the warehouse.

- This component is built using vendor data management tools and custom built programs.
- The operations performed by warehouse manager include.
  - Analysis of data to ensure consistency.
  - Transformation and merging the source data from temporary storage into data warehouse tables.
  - Create indexes and views on the base table.

Denormalization

- Generation of aggregation.
- Backing up and archiving of data.
- In certain situations, the warehouse manager also generates query profiles to determine which indexes and aggregations are appropriate.

4. Query Manager

- The query manager performs all operations associated with management of user queries.
- This component is usually constructed using vendor end-user access tools, data warehousing monitoring tools, database facilities and custom-built programs.
- The complexity of a query manager is determined by facilities provided by the end-user access tools and database.

5. Detailed Data

- This area of the warehouse stores all the detailed data in the database schema.
- In the majority of cases detailed data is not stored online but aggregated to the next level of details.
- The detailed data is added regularly to the warehouse to supplement the aggregated data.

6. Lightly and Highly Summarized Data

- This stores all the predefined lightly and highly summarized (aggregated) data generated by the warehouse manager.
- This area of the warehouse is transient as it will be subject to change on an ongoing basis in order to respond to the changing query profiles.
- The main goal of the summarized information is to speed up the query performance.
- As the new data is loaded into the warehouse, the summarized data is updated continuously.

7. Archive and Back up Data

- The detailed and summarized data are stored for the purpose of archiving and back up.
- The data is transferred to storage archives such as magnetic tapes or optical disks.

8. Meta Data

- The data warehouse also stores all the Meta data (data about data) definitions used by all processes in the warehouse.
- It is used for variety of purpose including:
- The extraction and loading process-Meta data is used to map data sources to a common view of information within the warehouse.
- The warehouse management process-Meta data is used to automate the production of summary tables.
- As part of Query Management process - Meta data is used to direct a query to the most appropriate data source.
- The structure of Meta data will differ in each process, because the purpose is different.

9. End-User Access Tools

- The main purpose of a data warehouse is to provide information to the business managers for strategic decision-making.
- 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 Systems Tools
  - Online Analytical Processing Tools
  - Data Mining Tools

Syllabus Topic : The Process of Data Warehouse Design

2.7 The Process of Data Warehouse Design

Data Warehouse Design Process

- Choose the grain (atomic level of data) of the business process.
- Choose a business process to model, e.g., orders, invoices, etc.
- Choose the dimensions that will apply to each fact table record.
- Choose the measure that will populate each fact table record

2.8 Data Warehousing Design Strategies or Approaches for Building a Data Warehouse

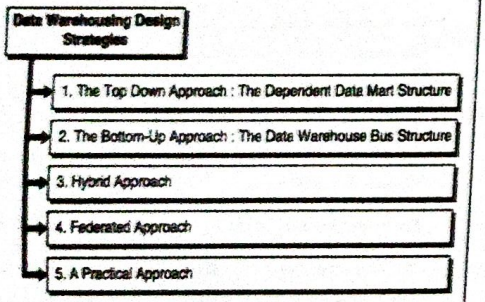


Fig. 2.8.1 : Data Warehousing Design Strategies

2.8.1 The Top Down Approach : The Dependent Data Mart Structure

- The data flow in the top down OLAP environment begins with data extraction from the operational data sources.
- This data is loaded into the staging area and validated and consolidated for ensuring a level of accuracy and then transferred to the Operational Data Store (ODS).
  - The ODS stage is sometimes skipped if it is a replication of the operational databases.

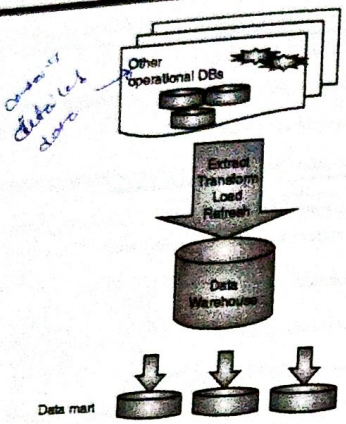


Fig. 2.8.2 : Top Down Approach

- Data is also loaded into the Data warehouse in a parallel process to avoid extracting it from the ODS.
  - Detailed data is regularly extracted from the ODS and temporarily hosted in the staging area for aggregation, summarization and then extracted and loaded into the Data warehouse.
  - The need to have an ODS is determined by the needs of the business.
  - If there is a need for detailed data in the Data warehouse then, the existence of an ODS is considered justified.
  - Else organizations may do away with the ODS altogether.
  - Once the Data warehouse aggregation and summarization processes are complete, the data mart refresh cycles will extract the data from the Data warehouse into the staging area and perform a new set of transformations on them.
  - This will help organize the data in particular structures required by data marts.
- Then the data marts can be loaded with the data and the OLAP environment becomes available to the users.

Advantages of top down approach

- It is not just a union of disparate data marts but it is inherently architected.

- The data about the content is centrally stored and the rules and control are also centralized.
- The results are obtained quickly if it is implemented with iterations.

Disadvantages of top down approach

- Times consuming process with an iterative method.
- The failure risk is very high.
- As it is integrated a high level of cross functional skills are required.

2.8.2 The Bottom-Up Approach : The Data Warehouse Bus Structure

- This architecture makes the data warehouse more of a virtual reality than a physical reality. All data marts could be located in one server or could be located on different servers across the enterprise while the data warehouse would be a virtual entity being nothing more than a sum total of all the data marts.
- In this context even the cubes constructed by using OLAP tools could be considered as data marts. In both cases the shared dimensions can be used for the conformed dimensions.
- The bottom-up approach reverses the positions of the Data warehouse and the Data marts.
- Data marts are directly loaded with the data from the operational systems through the staging area.
- The ODS may or may not exist depending on the business requirements. However, this approach increases the complexity of process coordination.
- The data flow in the bottom up approach starts with extraction of data from operational databases into the staging area where it is processed and consolidated and then loaded into the ODS.
- The data in the ODS is appended to or replaced by the fresh data being loaded.
- After the ODS is refreshed the current data is once again extracted into the staging area and processed to fit into the Data mart structure.

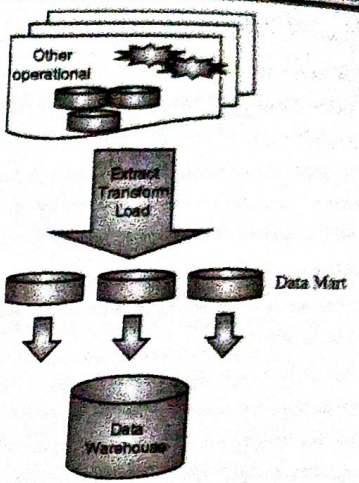


Fig. 2.8.3 : Bottom Up Approach

- The data from the Data mart then is extracted to the staging area aggregated, summarized and so on and loaded into the Data Warehouse and made available to the end user for analysis.

Advantages of bottom up approach

- This model strikes a good balance between centralized and localized flexibility.
- Data marts can be delivered more quickly and shared data structures along the bus eliminate the repeated effort expended when building multiple data marts in a non-architected structure.
- The standard procedure where data marts are refreshed from the ODS and not from the operational databases ensures data consolidation and hence it is generally recommended approach.
- Manageable pieces are faster and are easily implemented.
- Risk of failure is low.
- Allows one to create important data mart first.

Disadvantages of bottom up approach

- Allows redundancy of data in every data mart.
- Preserves inconsistent and incompatible data.
- Grows unmanageable interfaces.

2.8.3 Hybrid Approach

- The Hybrid approach aims to harness the speed and user orientation of the Bottom up approach to the integration of the top-down approach.
- The Hybrid approach begins with an Entity Relationship diagram of the data marts and a gradual extension of the data marts to extend the enterprise model in a consistent, linear fashion.
- These data marts are developed using the star schema or dimensional models.
- The Extract, Transform and Load (ETL) tool is deployed to extract data from the source into a non persistent staging area and then into dimensional data marts that contain both atomic and summary data.
- The data from the various data marts are then transferred to the data warehouse and query tools are reprogrammed to request summary data from the marts and atomic data from the Data Warehouse.

Advantages of hybrid approach

- Provides rapid development within an enterprise architecture framework.
- Avoids creation of renegade "independent" data marts.
- Instantiates enterprise model and architecture only when needed and once data marts deliver real value.
- Synchronizes meta data and database models between enterprise and local definitions.
- Backfilled DW eliminates redundant extracts.

Disadvantages of hybrid approach

- Requires organizations to enforce standard use of entities and rules.
- Backfilling a DW is disruptive, requiring corporate commitment, funding, and application rewrites.
- Few query tools can dynamically query atomic and summary data in different databases.

2.8.4 Federated Approach

- This is a hub-and-spoke architecture often described as the "architecture of architectures". It recommends an integration of heterogeneous data warehouses, data marts and packaged applications that already exist in the enterprise.
- The goal is to integrate existing analytic structures wherever possible and to define the "highest value" metrics, dimensions and measures and share and reuse them within existing analytic structures.
- This may result in the creation of a common staging area to eliminate redundant data feeds or building of a data warehouse that sources data from multiple data marts, data warehouses or analytic applications.
- Hackney-a vocal proponent of this architecture claims that it is not an elegant architecture but it is an architecture that is in keeping with the political and implementation reality of the enterprise.

Advantages of federated approach

- Provides a rationale for "band aid" approaches that solve real business problems.
- Alleviates the guilt and stress data warehousing managers might experience by not adhering to formalized architectures.
- Provides pragmatic way to share data and resources.

Disadvantages of federated approach

- The approach is not fully articulated.
- With no predefined end-state or architecture in mind, it may give way to unfettered chaos.
- It might encourage rather than dominant in independent development and perpetuate the disintegration of standards and controls.

2.8.5 A Practical Approach

The Steps in the Practical Approach are :

1. The first step is to do Planning and defining requirements at the overall corporate level.

2. An architecture is created for a complete warehouse.
3. The data content is conformed and standardized.
4. Consider the series of supermarts one at a time and implement the data warehouse.
5. In this practical approach, first the organizations needs are determined. The key to this approach is that planning is done first at the enterprise level. The requirements are gathered at the overall level.
6. The architecture is established for the complete warehouse. Then the data content for each supermart is determined. Supermarts are carefully architected data marts. Supermart is implemented one at a time.
7. Before implementation checks the data types, field length etc. from the various supermarts, which helps to avoid spreading of different data across several data marts.
8. Finally a data warehouse is created which is a union of all data marts. Each data mart belongs to a business process in the enterprise, and the collection of all the data marts form an enterprise data warehouse.

Syllabus Topic : A Three-Tier Data Warehousing Architecture

2.9 A Three-Tier Data Warehousing Architecture

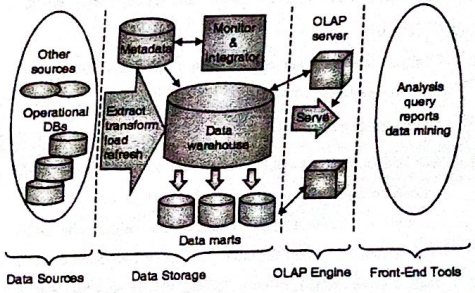


Fig. 2.9.1 : Multi-tier Data warehouse Architecture

1. Bottom Tier (Data Sources and Data Storage)

- It is a warehouse database server, that is generally a RDBMS.
- Using Application Program interfaces (called as gateways), data is extracted from operational and external sources.
- Gateways like, ODBC (Open Database connection), OLE-DB (Open linking and embedding for database), JDBC (Java Database Connection) is supported by underlying DBMS.

2. Middle Tier (OLAP Engine)

OLAP Engine is either implemented using ROLAP (Relational online Analytical Processing) or MOLAP(Multidimensional OLAP).

3. Top Tier (Front End Tools)

- This tier is a client which contains query and reporting tools, Analysis tools, and /or data mining tools.
- From the Architecture Point of view there are three data warehouse Models:

(a) Enterprise Warehouse

- The information of the entire organization is collected related to various subjects in enterprise warehouse.

(b) Data Mart

- A subset of Warehouse that is useful to a specific group of users.
- It can be categorized as Independent vs. dependent data mart.

(c) Virtual warehouse

- A set of views over operational databases.
- Only some of the possible summary views may be materialized.

2.9.1 Data Warehouse and Data Marts

Data Mart defined

- A data mart is oriented to a specific purpose or major data subject that may be distributed to support business needs. It is a subset of the data resource.

- A data mart is a repository of a business organization's data implemented to answer very specific questions for a specific group of data consumers such as organizational divisions of marketing, sales, operations, collections and others.
- A data mart is typically established as one dimensional model or star schema which is composed of a fact table and multi-dimensional table.
- A data mart is a small warehouse which is designed for the department level.
- It is often a way to gain entry and provide an opportunity to learn.
- **Major problem :** If they differ from department to department, they can be difficult to integrate enterprise-wide.

Table 2.9.1 : Differences between Data Warehouse and Data Mart

Sr. No.	Data Warehouse	Data Mart
1.	A data warehouse is application independent.	A data mart is a dependent on specific DSS application.
2.	It is centralized, and enterprise wide.	It is decentralized by user area.
3.	It is well planned.	It is possibly not planned.
4.	The data is historical, detailed and summarized.	The data consists of some history, detailed and summarized.
5.	It consists of multiple subjects.	It consists of a single subject of concern to the user.
6.	It is highly flexible.	It is restrictive.
7.	Implementation takes months to year.	Implementation is done usually in months.
8.	Generally size is from 100 GB to 1 TB.	Generally size is less than 100 GB.

Syllabus Topic : Types of OLAP Servers : ROLAP versus MOLAP versus HOLAP

2.10 Types of OLAP Servers : ROLAP versus MOLAP versus HOLAP

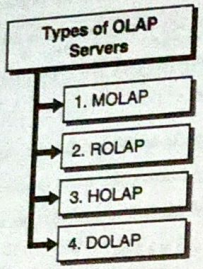


Fig. 2.10.1 : Types of OLAP Servers

Approaches to OLAP Servers

In the OLAP world, there are mainly two different types of OLAP servers: Multidimensional OLAP (MOLAP) and Relational OLAP (ROLAP). Hybrid OLAP (HOLAP) refers to technologies that combine MOLAP and ROLAP.

2.10.1 MOLAP

This is the more traditional way of OLAP analysis. In MOLAP, data is stored in a multidimensional cube. The storage is not in the relational database, but in proprietary formats.

Advantages of MOLAP

1. **Excellent performance :** MOLAP cubes are built for fast data retrieval, and are optimal for slicing and dicing operations.
2. **Can perform complex calculations :** All calculations have been pre-generated when the cube is created.

Disadvantages of MOLAP

1. **Limited in the amount of data it can handle :** Because all calculations are performed when the cube is built, it is not possible to include a large amount of data in the cube itself. This is not to say that the data in the cube cannot be derived from a large amount of data. Indeed, this is possible. But in this case, only summary-level information will be included in the cube itself.

2. **Requires additional investment :** Cube technology are often proprietary and do not already exists in the organization. Therefore, to adopt MOLAP technology, chances of additional investments in human and capital resources are needed.

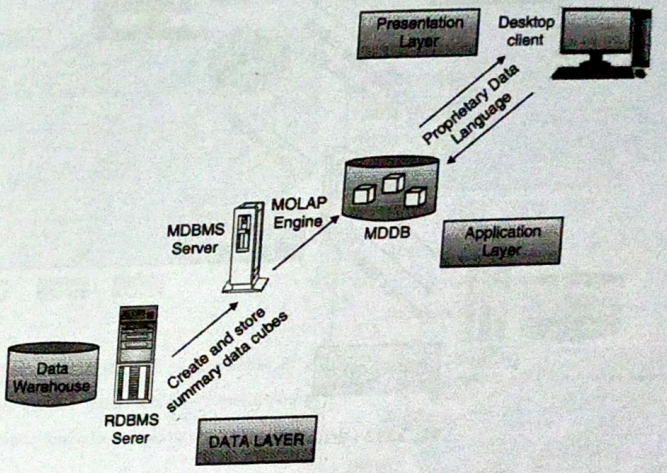


Fig. 2.10.2 : MOLAP Process

2.10.2 ROLAP

This methodology relies on manipulating the data stored in the relational database to give the appearance of traditional OLAP's slicing and dicing functionality. In essence, each action of slicing and dicing is equivalent to adding a "WHERE" clause in the SQL statement.

Advantages of ROLAP

1. **Can handle large amounts of data**

   The data size limitation of ROLAP technology is the limitation on data size of the underlying relational database. In other words, ROLAP itself places no limitation on amount of data.

2. **Can leverage functionalities inherent in the relational database**

   Often, relational database already comes with a host of functionalities. ROLAP technologies, since they sit on top of the relational database, can therefore leverage these functionalities.

Disadvantages of ROLAP

1. **Performance can be slow :** Because each ROLAP report is essentially a SQL query (or multiple SQL queries) in the relational database, the query time can be long if the underlying data size is large.
2. **Limited by SQL functionalities :** Because ROLAP technology mainly relies on generating SQL statements to query the relational database, and SQL statements do not fit all needs (for example, it is difficult to perform complex calculations using SQL), ROLAP technologies are therefore traditionally limited by what SQL can do.
3. ROLAP vendors have mitigated this risk by building into the tool out-of-the-box complex functions as well as the ability to allow users to define their own functions.

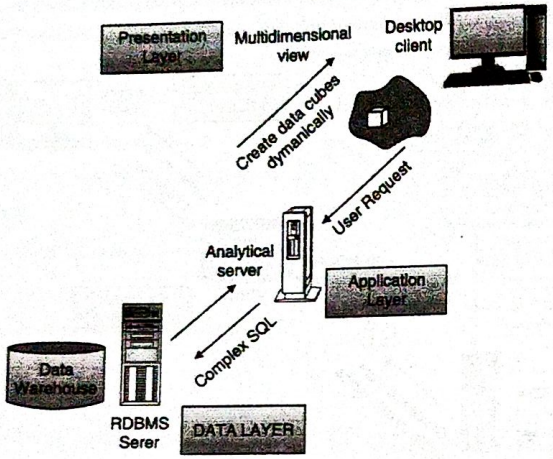


Fig. 2.10.3 : ROLAP Process

2.10.3 HOLAP

- HOLAP technologies attempt to combine the advantages of MOLAP and ROLAP. For summary-type information, HOLAP leverages cube technology for faster performance.
- When detail information is needed, HOLAP can "drill through" from the cube into the underlying relational data.
- For example, a HOLAP server may allow large volumes of detail data to be stored in a relational database, while aggregations are kept in a separate MOLAP store. The Microsoft SQL Server 7.0 OLAP Services supports a hybrid OLAP server.

2.10.4 DOLAP

It is Desktop Online Analytical Processing and variation of ROLAP. It offers portability to users of OLAP. For DOLAP, it needs only DOLAP software to be present on machine. Through this software, multidimensional datasets are formed and transferred to desktop machine.

2.11 Examples of OLAP

Ex. 2.11.1 : Consider a data warehouse for a hospital where there are three dimension (a) Doctor (b) Patient (c) Time and two measures i) count ii) charge where charge is the fee that the doctor charges a patient for a visit.

Using the above example describe the following OLAP operations.

1) Slice 2) Dice
3) Rollup 4) Drill down
5) Pivot

Soln. :

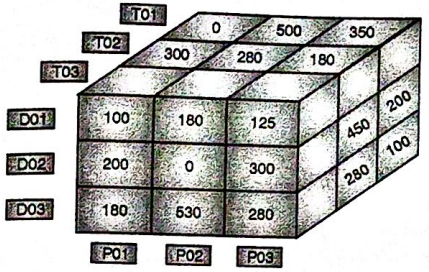
There are four tables, out of 3 dimension tables and 1 fact table.

Dimension tables

1. Doctor (DID, name, phone, location,pin,specialisation)
2. Patient (PID,name, phone , state, city,location, pin)
3. Time (TID,day, month, quarter , year)

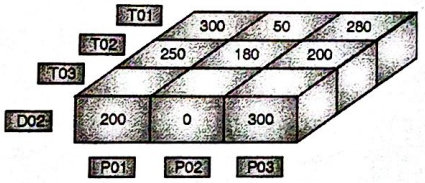
Fact Table

Fact_table (DID,PID,TID, count, charge)

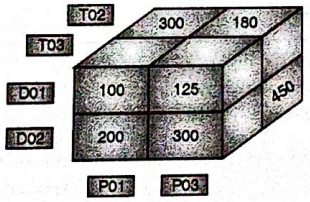


Operations

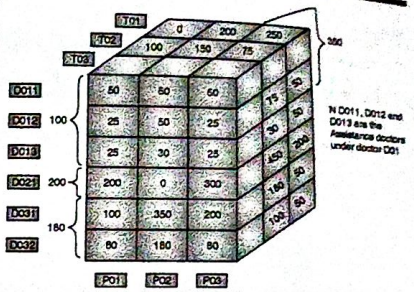
1. **Slice :** Slice on fact table with DID = 2 , this cuts the cube at DID = 2 along the time and patient axis thus it will display a slice of cube, in which time on x and patient on y axis.



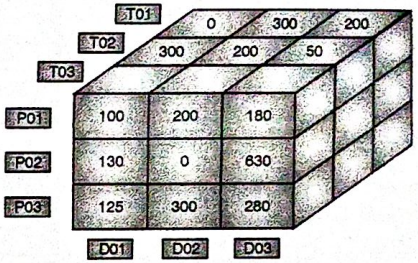
2. **Dice :** It is a sub cube of main cube. Thus it cuts the cube with more than one predicate like dice on cube with DID = 2, and DID = 01 and PID = 01 and PID = 03 and TID = 02, 03



3. **Roll up :** It gives summary based on concept hierarchies. Assuming there exists concept hierarchy in patient table as state->city->location. Then roll up will summarise the charges or count in terms of city or further roll up will give charges for a particular state etc.



4. **Drill down :** It is opposite to roll up that means if currently cube is summarised with respect to city then drill down will also show summarisation with respect to location.



5. **Pivot :** It rotates the cube, sub cube or rolled-up or drilled-down cube, thus changing the view of the cube.

Ex. 2.11.2 : All Electronics Company have sales department consider three dimensions namely

(i) Time
(ii) Product
(iii) store

The Schema Contains a central fact table sales with two measures

(i) dollars-cost and
(ii) units-sold

Using the above example describe the following OLAP operations :

(i) Dice (ii) Slice
(iii) Roll-up (iv) drill Down.

Soln. :

There are four tables, out of these 3 dimension tables and 1 fact table.

For OLAP operations refer Example 2.11.1.

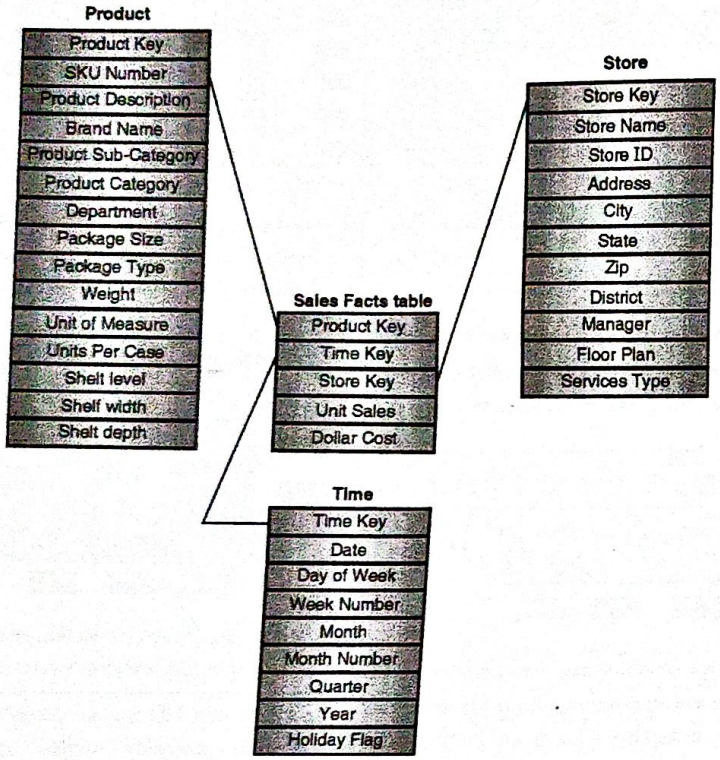


Fig. P. 2.11.2 : Star Schema for Electronics Company sales department

□□□

CHAPTER 3

Unit III

Measuring Data Similarity and Dissimilarity

Syllabus Topics

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

Syllabus Topic : Measuring Data Similarity and Dissimilarity

3.1 Measuring Data 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 given below.

3.1.1 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 :

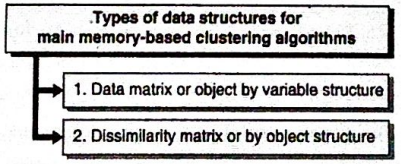


Fig. 3.1.1 : Types of data structures for main memory-based clustering algorithms

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

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

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.

Syllabus Topic : Proximity Measures for Nominal Attributes and Binary Attributes, Interval Scaled

3.2 Proximity Measures for Nominal Attributes and Binary Attributes, Interval Scaled

3.2.1 Proximity Measures for 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

- Proximity refers to either similarity or dissimilarity. As defined in Section 3.1 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 3.2.1

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

- The Table 3.2.1 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.

3.2.2 Proximity Measures for Binary Attributes

- Binary attributes are of two types, symmetric and asymmetric.
- 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.

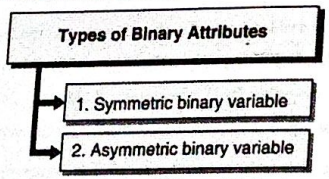


Fig. 3.2.1 : Types of Binary Attributes

➔ **1. 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.

For 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).

➔ **2. 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.

For 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).

- A contingency Table 3.2.2 for binary data :

Table 3.2.2

		Object n		
		1	0	Sum
Object m	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 (3.2.1) :

$$d(i,j) = \frac{b+c}{a+b+c+d} \quad \ldots(3.2.1)$$

- Jaccard coefficient (non-invariant if the binary variable is asymmetric) as shown in Equation (3.2.2) :

$$d(i,j) = \frac{b+c}{a+b+c} \quad \ldots(3.2.2)$$

☞ **Example**

Table 3.2.3 : 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 3.2.4.
- Using Equation (3.2.2) of asymmetric variable.

Table 3.2.4

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 (3.2.2) and use contingency Table 3.2.4.

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

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

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

Similarly, calculate distance for other combination

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

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

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

3.2.3 Interval Scaled

➔ **(SPPU - May 17)**

Q. What are interval-scaled variables ? Describe the distance measures that are commonly used for computing the dissimilarity of objects described by such variables. **May 17, 8 Marks**

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.

- These measures include the Euclidean, Manhattan, and Minkowski distances.

1.	Euclidean 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	$d_{Mk} = \sqrt[p]{\sum_{i=1}^{d} \lvert P_i - Q_i \rvert^p}$

- The measurement unit can affect the clustering analysis.
- For example, changing measurement units for weight form kilograms to pounds or for height from meters to inches, may lead to a very dissimilar clustering structure. In general, state a variable in minor unit will lead to a larger range for that variable, and thus a larger effect on the resultant clustering structure. To assist avoid belief on the choice of measurement units, the data must be standardized. Standardizing measurements attempts to give all variables an equal weight. This is mainly helpful when given no previous knowledge of the data. However, in some applications, users can intentionally want to grant more weight to a certain set of variables than to others.
- For example, when clustering basketball player candidates, we may favor to give more weight to the variable height.

Syllabus Topic : Dissimilarity of Numeric Data : Minkowski Distance, Euclidean Distance and Manhattan Distance

3.3 Dissimilarity of Numeric Data : Minkowski Distance, Euclidean Distance and Manhattan Distance

Minkowski Distance

- *It is used to determine the similarity or dissimilarity between two data objects.*

Minkowski distance formula

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

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

Euclidean distance and Manhattan distance

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

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

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

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

- 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 $q = \infty$)

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

- 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

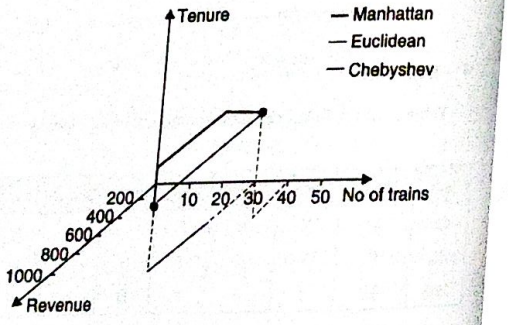


Fig. 3.3.1

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

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

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

Syllabus Topic : Proximity Measures for Categorical, Ordinal Attributes, Ratio Scaled Variables

3.4 Proximity Measures for Categorical, Ordinal Attributes, Ratio Scaled Variables

3.4.1 Categorical Attributes

Nominal attributes are also called as Categorical attributes. Described in section 3.2.1

3.4.2 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.
- For 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.
- 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 3.2.3.
- 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 normalise 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 :

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

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

3.4.3 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.
- **For 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.

3.4.4 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.
- **For 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** includes voltage output of a digital device, a person's age in years.

Syllabus Topic : Dissimilarity for Attributes of Mixed Types

3.5 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 :
  - If *f* is binary or nominal:

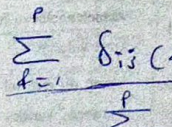
    $d_{ij}^{(f)} = 0$ if $x_{if} = x_{jf}$, or $d_{ij}^{(f)} = 1$ otherwise
  - If *f* is interval-based then use the normalized distance.
  - *If f* is ordinal or ratio-scaled then compute ranks r_{if} and treat z_{if} as interval-scaled.

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

Syllabus Topic : Cosine Similarity

3.6 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

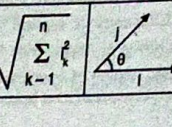
The dissimilarity between x and y : 1 – cos(x,y) = 0.51

Ex. 3.6.1 : Consider the following vectors x and y. x = [1, 1, 1, 1] y = [2, 2, 2, 2]. Calculate :
(I) Cosine similarity
(II) Euclidean distance **May 17, 3 Marks**

Soln. :

(I) 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

(II) Euclidean Distance

To find the Euclidean Distance between two points or tuples, the formula is given below.

Let $Y_1 = \{y_{11}, y_{12}, y_{13}, \ldots\ldots, y_{1n}\}$ and $Y_2 = \{y_{21}, y_{22}, y_{23}, \ldots\ldots, y_{2n}\}$

$$\text{distance}(Y_1, Y_2) = \sqrt{\sum_{i=1}^{n} (y_{1i} - y_{2i})^2}$$

Here,

$x_1 = \{1,1,1,1\}$ and $x_2 = \{2,2,2,2\}$

$$\text{distance}(x_1, y_2) = \sqrt{\sum_{i=1}^{n} y_{1i} - y_{2i})^2}$$

$= \sqrt{(2-1)^2 + (2-1)^2 + (2-1)^2 + (2-1)^2}$

$= \sqrt{4}$

$= 2$

□□□

$$\frac{x \bullet y}{\|x\| * \|y\|}$$

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

$$\delta_{ij} = \frac{x_{if} - x_{jf}}{\max(x_{if}) - \min(x_{if})}$$

CHAPTER 4

Unit IV

Association Rules Mining

Syllabus Topics

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

Syllabus Topic : Market Basket Analysis

4.1 Market Basket Analysis

4.1.1 What is Market Basket Analysis?

- Market basket analysis is a modelling 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. Customers buy the subset of items as per their need and marketer gets the information that which things customers have 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 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.*

4.1.2 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.

4.1.3 Applications of Market Basket Analysis

➔ **(SPPU - Dec. 17)**

Q. Explain applications of Market baskets analysis.

Dec. 17. 4 Marks

- 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 catalogue of an ecommerce site may be defined.
  - Inventory may be managed based on product demands.

Syllabus Topic : Frequent Itemsets

4.2 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 for all maximal frequent itemsets.

Syllabus Topic : Closed Itemsets

4.3 Closed Itemsets

➔ **(SPPU - May 16, Dec. 16, Aug. 17)**

Q. Explain the following terms : Closed and maximal frequent itemsets.

May 16, Dec. 16, Aug. 17, 3 Marks

- 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

Let us consider minimum support = 2.

- The itemsets that are circled with thick lines are the frequent itemsets as they satisfy the minimum support. Fig. 4.3.1, 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. Fig. 4.3.1, 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. Fig. 4.3.1, maximal frequent itemsets are {rs,pqs}. For example {rs} is maximal frequent itemset as none of its immediate supersets like {prs, qrs} is frequent.

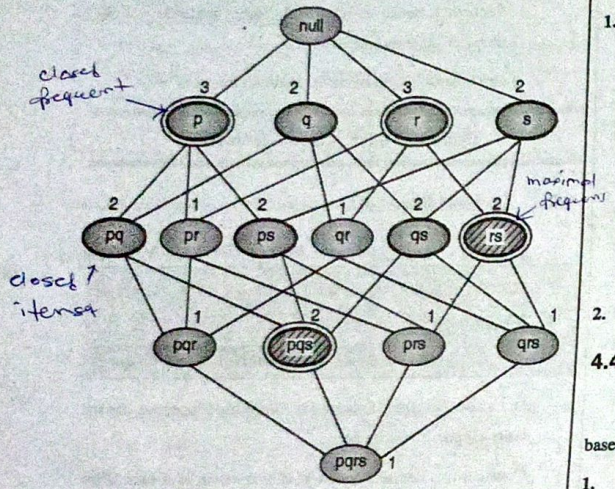


Fig. 4.3.1 : Lattice diagram for maximal, closed and frequent itemsets

Syllabus Topic : Association Rules

4.4 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".

4.4.1 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.

4.4.2 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.

4.4.3 Efficient and Scalable Frequent Itemset Mining Method

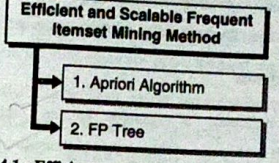


Fig. 4.4.1 : Efficient and Scalable Frequent Itemset Mining Method

Syllabus Topic : A-priori Algorithm

4.5 A-priori Algorithm

→ **(SPPU - Aug. 17)**

Q. Explain the Apriori algorithm for generation of association rules. How candidate keys are generated using apriori algorithm. **Aug. 17, 6 Marks**

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.

Q. Write a pseudo code for Apriori algorithm and explain. **Dec. 15, 6 Marks**

Q. Write Apriori Algorithm and explain it with suitable example. **Dec. 17, 6 Marks**

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) L_1 = find_frequent_1-itemsets(D);
(2) for ($k = 2; L_{k-1} \neq \phi; k++$){
(3) C_k = apriori_gen (L_{k-1});
(4) for each transaction $t \in D$ { // scan D for counts
(5) C_t = subset (C_k, t); // get the subsets of t that are candidates
(6) for each candidate $c \in C_t$
(7) c.count ++ ;
(8) }
(9) $L_k = \{c \in C_k \mid c.count \geq min\_sup\}$
(10) }
(11) return $L = \cup_k L_k$;

Procedure apriori_gen (L_{k-1}:frequent (k – 1) - itemsets)

(1) for each itemset $l_1 \in L_{k-1}$
(2) for each itemset $l_2 \in L_{k-1}$
(3) if (l_1 (1) = l_2 (l_1) $\wedge$ (l_1 (2) = l_2 (2)) $\wedge \ldots \wedge$ (l_1 [k – 2] = l_2 [k – 2]) $\wedge$ (l_1 [k – 1] <l_2 [k – 1]) then {
(4) $c = l_1 \bowtie l_2$; // join step:generate candidates
(5) if has_infrequent_subset(c, L_{k-1}) then
(6) delete c; // prune step: remove unfruitful candidate
(7) else add c to C_k
(8) }

```
(9)     return C_k;

Procedure has_infrequent_subset(c: candidate k-itemset;
L_k-1: frequent (k – 1) –itemsets); // use prior knowledge

(1)     for each (k – 1)-subset s of c
(2)         if s ∉ L_k-1 then
(3)             return TRUE;
(4)     return FALSE;
```

4.5.1 Advantages and Disadvantages of Apriori Algorithm

Some of the advantages and disadvantages of Apriori Algorithm are as follows :

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.

Syllabus Topic : Generating Association Rules from Frequent Item Sets

4.6 Generating Association Rules from Frequent Item Sets

- Find the frequent itemsets from transaction database.
- Using confidence formula generate strong association rules which satisfy both minimum support and minimum confidence.

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 as well as Y i. e.(X and Y together).
- An itemset is considered to be a *large itemset* if its support is above some threshold called minimum support.

Confidence

- The confidence or strength for an association rule A => B is the ratio of the number of transactions that contain A as well as 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

$$\text{Confidence } (A \Rightarrow B) = \frac{\#\_\text{tuples\_containing\_both\_A\_and\_B}}{\#\_\text{tuples\_containing\_A}}$$

Syllabus Topic : Improving the Efficiency of a-priori

4.7 Improving the Efficiency of a-priori

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.

4.8 Solved Example on Apriori Algorithm

Ex. 4.8.1 : 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. 4.8.2 : *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 I(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. 4.8.3 : *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.

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

Ex. 4.8.4 : 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. 4.8.5 : 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. 4.8.6 : Apply the Apriori algorithm on the following data with Minimum support = 2

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

Soln. :

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

C_1 =

I-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 =

I-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
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 datacontains 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. 4.8.7 : 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

50 x 9 / 100 = 4.5 ≈ 5

T_{id}	Items
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. 4.8.8 : Consider the five transactions given below. If minimum support is 30% and minimum confidence is 80%, determine the frequent itemsets and association rules using the a priori algorithm.

Transaction	items
T1	Bread, Jelly, Butter
T2	Bread, Butter
T3	Bread, Milk, Butter
T4	Coke, Bread
T5	Coke, Milk

Soln. :

Step 1 : Scan D for Count of each candidate.

The candidate list is {Bread, Jelly, Butter, Milk, Coke}

C_1 =

I-Itemlist	Sup-Count
Bread	4
Jelly	1
Butter	3
Milk	F2
Coke	2

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

I-Itemlist	Sup-Count
Bread	4
Butter	3
Milk	2
Coke	2

Step 3 : Generate C2 from L1 and find the support

C_2 =

I-Itemlist	Sup Count
{Bread, Butter}	3
{Bread, Milk}	1
{Bread, Coke}	1
{Butter, Milk}	1
{Butter, Coke}	0
{Milk, Coke}	1

Step 4 : Compare candidate (C2) support count with the minimum support count

L_2 =

Frequent 2 - Itemset	Sup – Count
{Bread, Butter}	3

Step 5 : So data contain the frequent itemset is {Bread, Butter}

Association Rule	Support	Confidence	Confidence %
Bread⟶Butter	3	3/4	75%
Butter⟶Bread	1	3/3	100%

Minimum confidence threshold is 80% (Given)

Final rule is

Butter→Bread

Ex. 4.8.9 : 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.

Soln. :

$\frac{10 \times 30}{100} = 3$

Step 1 : 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 2 : 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
BL	2
CD	5
CE	2
CF	1

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
FL	2
HK	1
HL	2
KL	1

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

Step 3 : 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 4 : 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

Ex. 4.8.10 : Consider the following :

Transaction	Items
t_1	Bread, Jelly, Peanut Butter
t_2	Bread, Peanut Butter
t_3	Bread, Milk, Peanut Butter
t_4	Beer, Bread
t_5	Beer, Milk

Calculate the support and confidence for the following association rules :

i) Bread → Peanut Butter

ii) Jelly → Milk,

iii) Beer → Bread.

SPPU - Dec. 15, 6 Marks

Soln. :

Consider Minimum support count = 2 and Minimum confidence = 80%

Step 1 : Scan D for Count of each candidate.

The candidate list is {Bread, Jelly, Peanut Butter, Milk, Beer}

C1 =

I-Itemlist	Sup-Count
Bread	4
Jelly	1
Peanut Butter	3
Milk	2
Beer	2

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

I-Itemlist	Sup-Count
Bread	4
Peanut Butter	3
Milk	2
Beer	2

Step 3 : Generate C2 from L1 and find the support

C2 =

I-Itemlist	Sup Count
{Bread, Peanut Butter}	3
{Bread, Milk}	1
{Bread, Beer}	1
{Peanut Butter, Milk}	1
{Peanut Butter, Beer}	0
{Milk, Beer}	1

Step 4 : Compare candidate (C2) support count with the minimum support count

L2 =

Frequent 2 - Itemset	Sup – Count
{Bread, Peanut Butter}	3

Step 5 : So data contain the frequent itemset is {Bread, Peanut Butter}

Association Rule	Support	Confidence	Confidence %
Bread ⟶ Peanut Butter	3	3/4	75%
Bread ⟶ Peanut Butter	1	3/3	100%

Minimum confidence threshold is 80%

Final rule is : Peanut Butter ⟶ Bread

Similarly Confidence and support for the following association rules are :

Association Rule	Support	Confidence	Confidence %
Bread →Peanut Butter	3	3/4	75%
Jelly → Milk	0	0/1	0 %
Beer → Bread	1	1/2	50%

Ex. 4.8.11 : Consider the market basket transactions shown below :

Transaction ID	Items-bought
T1	{Mango, Apple, Banana, Dates}
T2	{Apple, Dates, Coconut, Banana, Fig}
T3	{Apple, Coconut, Banana, Fig}
T4	{Apple, Banana, Dates}

Assuming the minimum support of 50% and minimum confidence of 80%

(i) Find all frequent itemsets using Apriori algorithm.

(ii) Find all association rules using Apriori algorithm. SPPU - May 16, 6 Marks

Soln. :

(i) Let us assume, Mango = M, Apple = A, Banana = B, Dates = D, Coconut = C and Fig = F

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

C_1 =

Itemset	Sup-count
A	4
B	4
C	2
D	3
F	2
M	1

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

L_1 =

Itemset	Sup-count
A	4
B	4
C	2
D	3
F	2

Step 3 : Generate candidate C_2 from L_1

C_2 =

Itemset
A,B
A,C
A,D
A,F
B,C
B,D
B,F
C,D
C,F
D,F

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

L_2 =

Itemset	Sup-count
A,B	4
A,C	2
A,D	3
A,F	2
B,C	2
B,D	2
B,F	2
C,F	2

Step 5 : Generate candidate C_3 from L_2.

C_3 =

Itemset
A,B,C
A,B,D
A,B,F
A,D,F
B,C,D
B,C,F
B,D,F
A,C,F

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

L_3 =

Itemset	Sup-count
A,B,C	2
A,B,D	3
A,B,F	2
B,C,F	2
A,C,F	2

Step 7 : Generate candidate C_4 from L_3.

C_4 =

Itemset
A,B,C,D
A,B,C,F

Step 8 : Compare candidate (C_4) support count with the minimum support count.

L_4 =

Itemset	Sup-count
A,B,C,D	1
A,B,C,F	2

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

(ii) 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,C → F	2	2/2	100
A,C,F → B	2	2/2	100
B,C,F→ A	2	2/2	100
A,B,F → C	2	2/2	100

Ex. 4.8.12 : A database has five transactions. Let minimum support is 60%.

TID	Items
1	Butter, Milk
2	Butter, Dates, Balloon, Eggs
3	Milk, Dates, Balloon, Cake
4	Butter, Milk, Dates, Balloon
5	Butter, Milk, Dates, Cake

Find all the frequent item sets using Apriori algorithm. Show each step.

SPPU - Oct. 16, 6 Marks

Soln. :

Step 1 : Scan database for count of each candidate. The candidate list is {Butter, milk, Dates, Balloon, Eggs, cake} and find the support

c_1 =

Itemset	Support
{ Butter }	4
{ Milk }	4
{ Dates }	4
{ Balloon }	3
{ Eggs }	1
{ Cake }	2

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

L_1 =

Itemset	Support
{ Butter }	4
{ Milk }	4
{ Dates }	4
{ Balloon }	3

Step 3 : Generate candidate C_2 from L_1

C_2 =

Itemset
{ Butter, Milk }
{ Butter, Dates}
{Butter, Balloon }
{Milk, Dates}
{ Milk, Balloon }
{Dates, Balloon }

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

Itemset	Support
{ Butter, Milk }	3
{ Butter, Dates}	3
{Butter, Balloon }	2
{Milk, Dates}	3
{ Milk, Balloon }	2
{Dates, Balloon }	3

Step 5 : Compare candidate C_2 support count with minimum support count

Itemset	Support
{ Butter, Milk }	3
{ Butter, Dates}	3
{Milk, Dates}	3
{ Dates, Balloon }	3

Step 6 : Generate candidate C_3 from L_2

{Balloon, Milk, Dates }

Associative rules from this –

Itemset	Confidence %
{Milk, Dates } → {Balloon }	0.67
{ Milk, Balloon} → {Dates }	1.00
{Dates, Balloon} → {Milk }	0.67
{ Balloon } → { Milk, Dates }	0.67
{ Dates } → { Milk, Balloon }	0.5
{ Milk} → { Dates, Balloon }	0.5

Ex. 4.8.13 : Consider the market basket transaction shown below :

Transaction ID	Items bought
T1	{M, A, B, D}
T2	{A, D, C, B, F}
T3	{A, C, B, F}
T4	{A, B, D}

Assuming the minimum support of 50% and minimum confidence of 80%

Find all frequent items using Apriori algorithm.

Find all association rules using Apriori algorithm.

Soln. :

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

C_1 =

Itemset	Sup-count
A	4
B	4
C	2
D	3
F	2
M	1

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

L_1 =

Itemset	Sup-count
A	4
B	4
C	2
D	3
F	2

Step 3 : Generate candidate C_2 from L_1.

C_2 =

Itemset
A,B
A,C
A,D
A,F
B,C
B,D
B,F
C,D
C,F
D,F

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

L_2 =

Itemset	Sup-count
A,B	4
A,C	2
A,D	3
A,F	2
B,C	2
B,D	2
B,F	2
C,F	2

Step 5 : Generate candidate C_3 from L_2.

C_3 =

Itemset
A,B,C
A,B,D
A,B,F
A,D,F
B,C,D
B,C,F
B,D,F
A,C,F

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

L_3 =

Itemset	Sup-count
A,B,C	2
A,B,D	3
A,B,F	2
B,C,F	2
A,C,F	2

Step 7 : Generate candidate C_4 from L_3.

C_4 =

Itemset
A,B,C,D
A,B,C,F

Step 8 : Compare candidate (C_4) support count with the minimum support count.

L_4 =

Itemset	Sup-count
A,B,C,D	1
A,B,C,F	2

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

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,C → F	2	2/2	100
A,C,F → B	2	2/2	100
B,C,F→ A	2	2/2	100
A,B,F → C	2	2/2	100

Syllabus Topic : Mining Frequent Item sets without Candidate Generation : FP Growth Algorithm

4.9 Mining Frequent Item sets without Candidate Generation : FP Growth Algorithm

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 ≤ the candidate sets generated in the association rule mining.
- This approach is very efficient due to :
  - Compression of a large database into a smaller data structure.
  - It is a frequent 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.

4.9.1 FP-Tree Algorithm

FP-tree construction 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).
- **Algorithm :** FP 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

1. **A FP tree is constructed in the following steps**

(a) Scan the transaction database D once. Collect F, the set of frequent items, and their support 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.

2. **The FP-tree is mined by calling FP growth. FP tree, null/, which is implemented as follows :**

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 Treeβ;
(7) if Treeβ≠ϕ then
(8) call FP_growth (Treeβ, β) ; }
```

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(|Transl), where |Transl is the number of frequent items in Trans.

4.9.2 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).

4.9.3 Example of FP Tree

Ex. 4.9.1 : 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

SPPU - Oct. 16, 5 Marks

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

Item	Sup.
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)

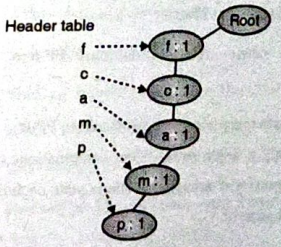
TID	Items Bought	(Ordered frequent items)
1	f, a, c, d, g, i, m, p	f, c, a, m, p
2	a, b, c, f, l, m, o	f, c, a, b, m
3	b, f, h, j, o	f, b
4	b, c, k, s, p	c, b, p
5	a, f, c, e, l, p, m, n	f, c, a, m, p

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

Step 3 : FP Tree construction

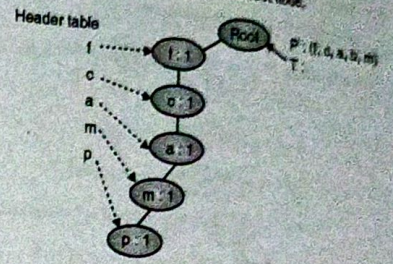
Originally Empty 

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

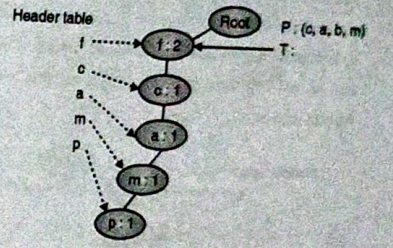


Step 5 : Start the insertion of Second transaction (f,c,a,b,m)

(i) The transaction T is pointing to the root node.

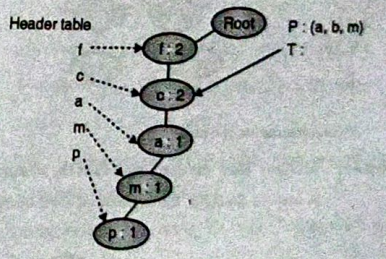


(ii) Consider the first item in the second transaction i.e. f and add it in the tree.

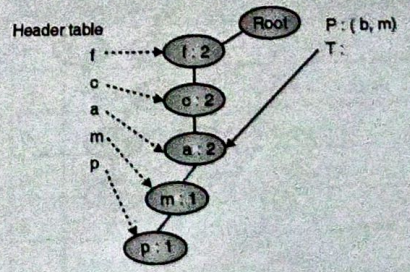


After this step we get f:2, finished adding f in the above tree.

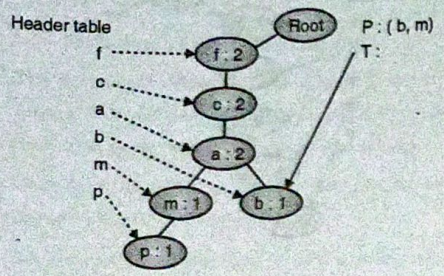
(iii) Now consider the second item in the above transaction .i.e. c.



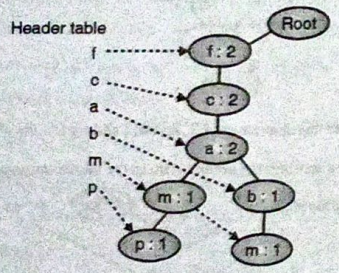
(iv) Similarly consider the next item a.



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

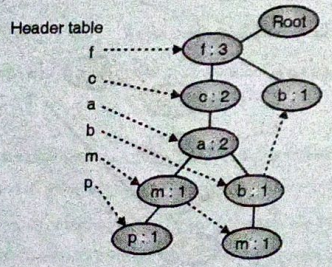


(vi) Now only m of second transaction is left. Though a node m is already exists still we can't increase its count of the existing node m as we need to represent the second transaction in FP tree, so add new node m below node b and link it with existing node m.

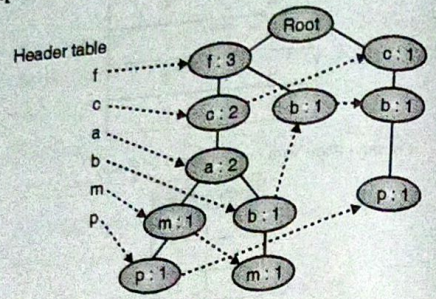


Second transaction is complete.

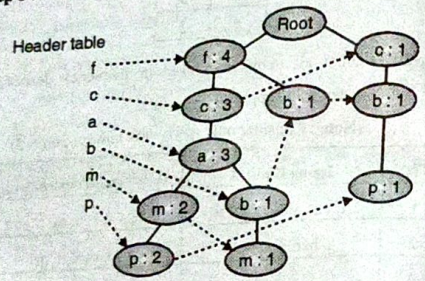
Step 6 : Similarly insert the third transaction(f,b) as explained in step 5. So After the insertion of third transaction (f,b)



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.

4.9.4 Mining Frequent Patterns from FP Tree

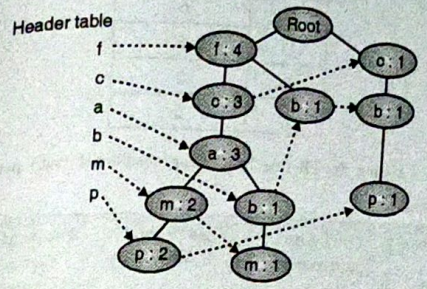
General idea (divide-and-conquer)

- Use the FP Tree and recursively grow frequent pattern path.

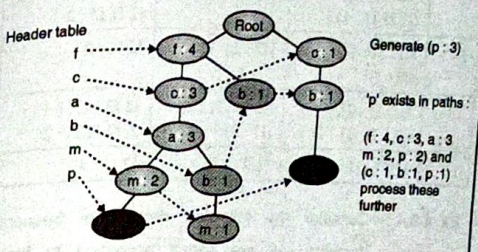
Method

- 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 :



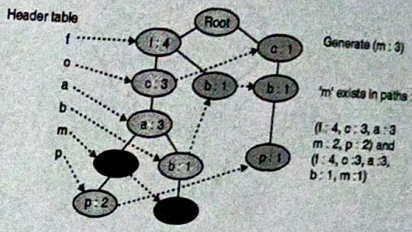
- Starting from the bottom of the header table.



- 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 :
  - Path 1 : (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.

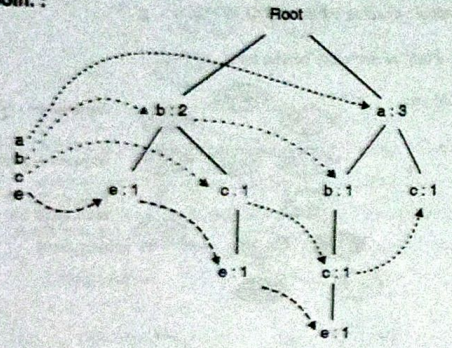
Item	Conditional pattern-base	Conditional FP-tree
P	{(fcam:2), (cb:1)}	{(c:3)}|p
M	{(fca:2), (fcab:1)}	{(f:3, c:3, a:3)}|m
B	{(fca:1), (f:1), (c:1)}	Empty
A	{(fc:3)}	{(f:3, c:3)}|a
C	{(f:3)}	{(f:3)}|c
f	Empty	Empty

Ex. 4.9.2 : 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

a - 3
b - 3
c - 3
e - 2

Soln. :

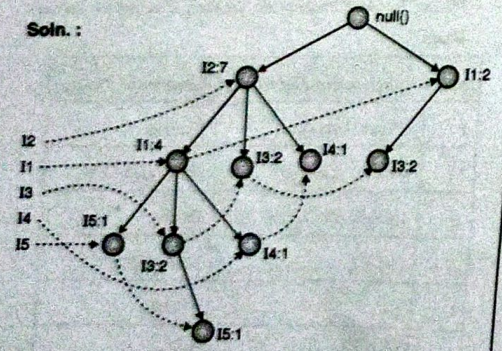


Ex. 4.9.3 : Transaction database is

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.

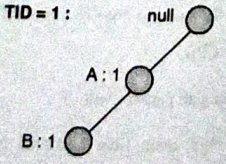
Item	Conditional pattern base	Conditional FP-tree	Frequent patterns generated
I5	{(I2 I1 : 1), (I2 I1 I3 : 1) }	{I2 : 2, I1 : 2}	I2 I5 : 2, I1 I5 : 2, I2 I1 I5 : 2
I4	{(I2 I1 : 1), (I2 : 1)}	{I2 : 2}	I2 I4 : 2
I3	{(I2 I1 : 2), (I2 : 2), (I1 : 2) }	(I2 : 4, I1 : 2), (I1 : 2)	I2 I3 : 4, I1, I3 : 2, I2 I1 I3 : 2
I1	{(I2 : 4)}	{(I2 : 4)}	I2 I1 : 4

Ex. 4.9.4 : 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.

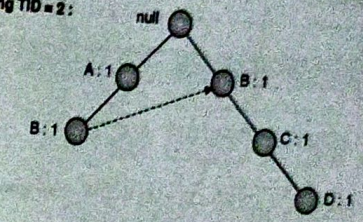
TID	Items
1	{A, B}
2	{B, C, D}
3	{A, C, D, E}
4	{A, D, E}
5	{A, B, C}
6	{A, B, C, D}
7	{B, C}
8	{A, B, C}
9	{A, B, D}
10	{B, C, E}

Soln. :

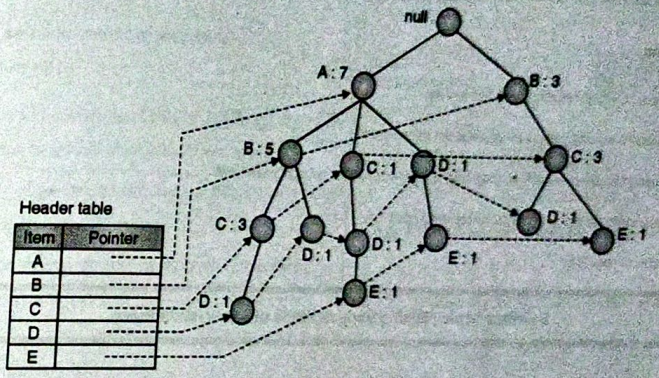
After reading TID = 1 :



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'
  - Again filter away all items < minimum support threshold

    (i.e.1 as Support of D =1)
  - Consider First Branch

    {(A:1,B:1,C:1),(A:1,B:1),

    (A:1,C:1), (A:1)} => {(A:4,B:2,C:2)}

    So append ABC with D

    We generate ABCD:1

- Similarly for other branch of the tree

  {(B:1,C:1)} => {(B:1,C:1)}

  So append BC with D

  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.

4.9.5 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 Various Kinds of Association Rules

4.10 Mining Various Kinds of Association Rules

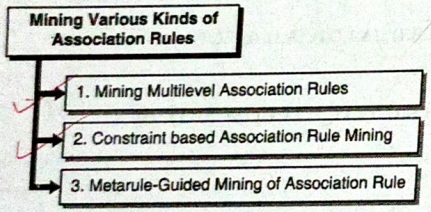


Fig. 4.10.1 : Mining Various Kinds of Association Rules

4.10.1 Mining Multilevel Association Rules

→ **(SPPU - May 16)**

Q. Explain the following terms : Multilevel association rules. **May 16, 3 Marks**

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

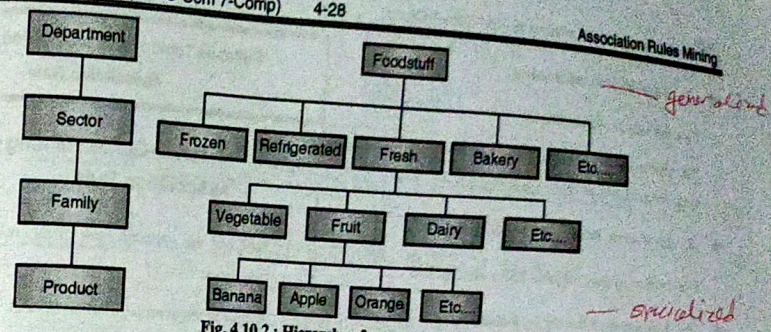


Fig. 4.10.2 : Hierarchy of concept

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

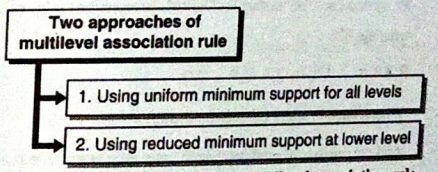


Fig. 4.10.3 : Two approaches of multilevel association rule

→ 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.

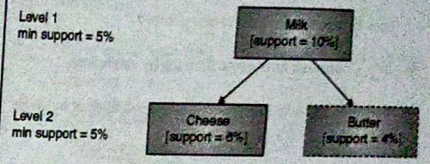


Fig. 4.10.4 : Example of uniform minimum support for all levels

→ 2. Using reduced minimum support at lower level

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

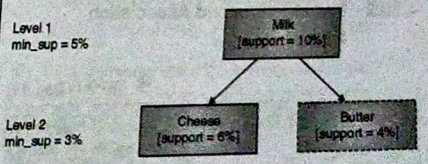


Fig. 4.10.5 : Example of reduced minimum support for lower levels

There are 4 search strategies :

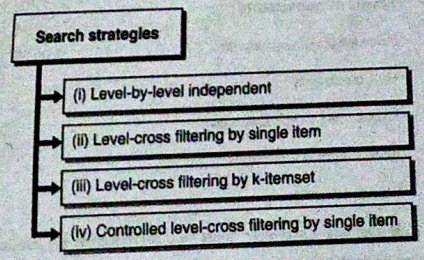


Fig. 4.10.6 : Search strategies

➔ **(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 : Constraint based Association Rule Mining

4.10.2 Constraint based Association Rule Mining

➔ **(SPPU - Dec. 16)**

Q. Explain the following terms : Constraints-based rule mining. **Dec. 16, 3 Marks**

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.

Syllabus Topic : Metarule-Guided Mining of Association Rule

4.10.3 Metarule-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 behaviour 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 metarules. For example if two predicates Age and Salary are given to analyse whether the customer buys "Apple Product"

  age (C, "30..40") Λ Salary (C, "30K..50K") $\rightarrow$buys (C, "Apple Product")
- So generalise the metarule Guided association rule as a template like

  $P_1 \Lambda P_2 \Lambda \ldots \Lambda P_n \rightarrow Q_1 \Lambda Q_2 \Lambda \ldots \Lambda Q_r$

  Where, each P_i's and Q_j's are predicates

  And the number of predicates in the merarule is $p = n + r$

4.11 Solved University Question and Answer

Q. 1 Differentiate between : **(Oct. 16, 4 Marks)**

(i) Multilevel and multidimensional associations

(ii) Pattern-pruning and data-pruning constraints

Ans. :

(I) Multilevel and multidimensional associations

Multilevel associations

- Items are always in the form of hierarchy.
- tems which are at leaf nodes are having lower support.

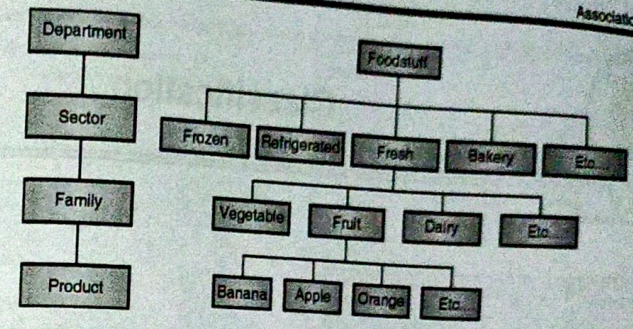


Fig. 1 : Hierarchy of concept

- 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.

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.

Multidimensional associations

- **Single-dimensional rules :** The rule contains only one distinct predicate. In the following example the rule has only one predicate "buys".

  buys(X, "Butter") $\Rightarrow$ 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") $\wedge$ salary(X, "High") $\Rightarrow$ buys(X, "Computer")
  - **Hybrid-dimension association rules :** The rule have many occurrences of same predicate i.e. buys.

  gender(X,"Male") $\wedge$ buys(X, "TV") $\Rightarrow$ 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.

(ii) Pattern-pruning and data-pruning constraints

Pattern-pruning	data-pruning
If we can prune a frequent pattern P after checking constraints on it, then the entire subtree rooted at P in the pattern tree model will not be grown.	If we can prune Graph G from the data space search of P after data pruning checking G, will be pruned from the data search space of all nodes in the subtree rooted at P.
Pattern-pruning should be performed when $T_c(P) <=_p Tp$	We would perform data pruning checking for G if $T_d(P,G) <_q T_p$

□□□

CHAPTER 5

Unit V

Classification

Syllabus Topics

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

Syllabus Topic : Introduction to : Classification and Regression for Predictive Analysis

5.1 Introduction to : Classification and Regression for Predictive Analysis

- 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. or target variable

☞ **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

5.1.1 Classification is a Two Step Process

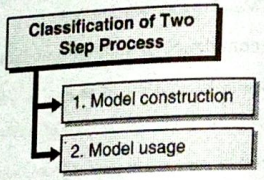


Fig. 5.1.1 : Classification of two step process

➜ **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.

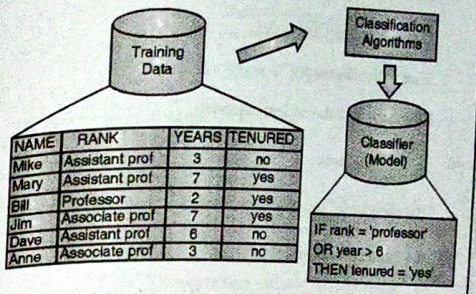
➜ **2. Model usage**

- 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**

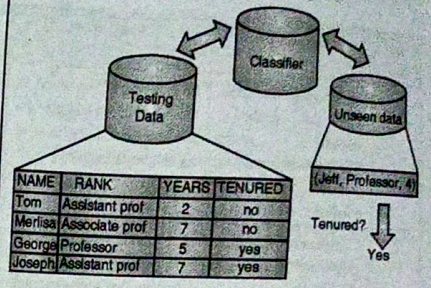
Classification process : (1) Model construction



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. 5.1.2 : Learning : Training data are analyzed by a classification algorithm

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

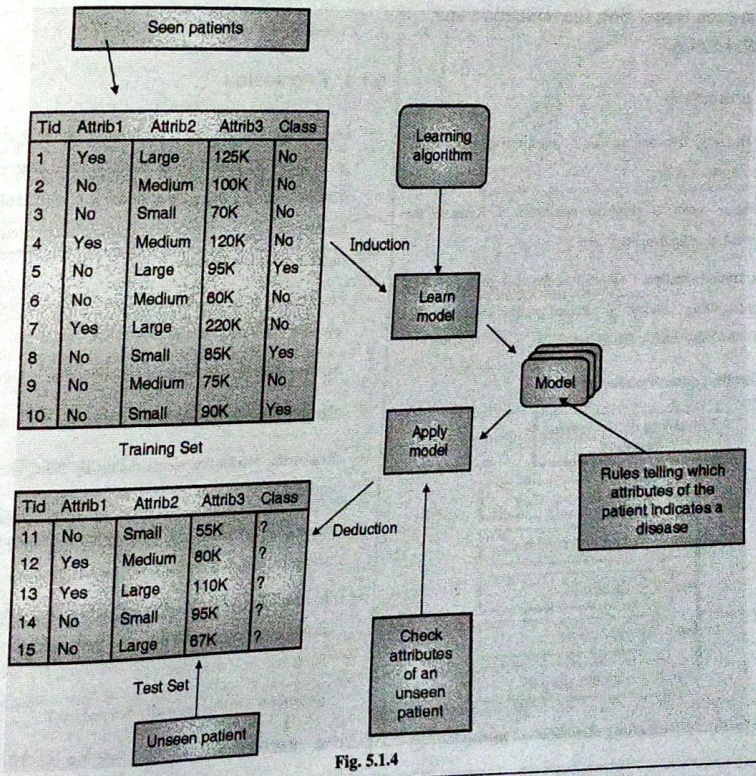


NAME	RANK	YEARS	TENURED
Tom	Assistant prof	2	no
Merlisa	Associate prof	7	no
George	Professor	5	yes
Joseph	Assistant prof	7	yes

Fig. 5.1.3 : 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 ?



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

Training Set

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

Fig. 5.1.4

5.1.2 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.

5.1.3 Issues Regarding Classification and Prediction

☞ **Data preparation**

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

☞ **Evaluating classification methods**

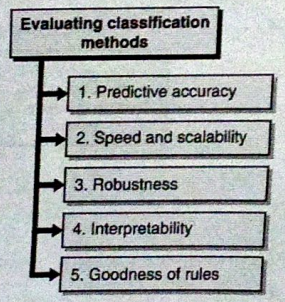


Fig. 5.1.5 : Evaluating classification methods

➜ **1. Predictive accuracy**

- This refers the ability of the model to correctly predict the class label of new or previously unseen data.

➜ **2. Speed and scalability**

- Time to construct the model.
- Time to use the model.
- Efficiency in disk-resident databases.

➜ **3. Robustness**

Handling noise and missing values.

➜ **4. Interpretability**

Understanding and insight provided by the model.

➜ **5. Goodness of rules**

- Decision tree size.
- Compactness of classification rules.

5.1.4 Regression

- Suppose an employee needs to predict how much rise he will get in his salary after 5 years, means he bother to predict the numeric value. In this case a model is constructed based on his previous salary values that predicts a continuous-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.

☞ **Structure of Regression Model**

- Regression Model represents reality by using the system of equations.
- Regression model explains relationship between variables and also enables quantification of these relationships.
- It determines the strength of relationship between one dependent variable with the other independent variable using some statistical measure.
- Dependent variable is usually denoted by Y.

the variable which we have to predict

- The two basic types of regression :

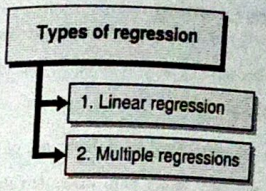


Fig. 5.1.6 : Types of Regression

The general form of regression is :

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

Multiple regression : $Y = m + n_1X_1 + n_2X_2 + n_3X_3 + ... + n_tX_t + u$

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

ϵ or u = The regression residual. / error term

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

☞ **Linear Regression**

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

(A) 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. 5.1.7.

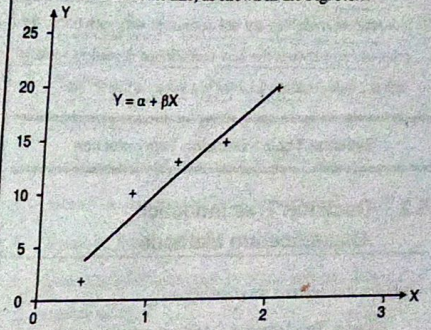


Fig. 5.1.7 : Linear regression

(B) 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.

$$Y = a_0 + a_1X_1 + a_2X_2 + ... + a_kX_k + e$$

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

e = error term

☞ **Other Regression Model**

- In log linear regression a best fit between the data and a loglinear model is found.

y = mx + c (x: independent, y-intercept)

- Major assumption : A linear relationship exists between the log of the dependent and independent variables.
- Loglinear models are models that postulate a linear relationship between the independent variables and the logarithm of the dependent variable, for example :

$$\log(y) = a_0 + a_1 x_1 + a_2 x_2 \ldots + a_N x_N$$

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.

Syllabus Topic : Decision Tree Induction

5.2 Decision Tree Induction Classification Methods

Classifications methods are given below :

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

- 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 builddecision trees :
  - ID3 (Iterative Dichotomiser 3)
  - C4.5 (Successor of ID3)
  - CART (Classification And Regression Tree)
  - CHAID (CHi-squared Automatic Interaction Detector

5.2.1 Appropriate Problems for Decision Tree Learning

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

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

5.2.2 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.

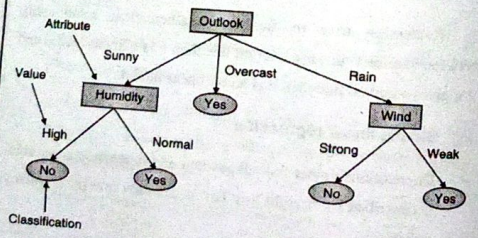


Fig. 5.2.1 : Representation of decision tree

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

5.2.3 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 :

Q. Write a pseudo code for the construction of Decision Tree State and justify its time complexity also. **Dec. 15. 4 Marks**

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. applyAttribute_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 D_j be the set of data tuples in D satisfying outcome j; // a partition
12. if D_j 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 (D_j, attribute_list) to node N; endfor

- **Time complexity :** For a normal style decision tree such as C4.5 the time complexity is O(N D^2), where D is the number of features. A single level decision tree would be O(N D)
- It gives better the efficiency of computation.

5.2.4 Tree Pruning

- Because of noise or outliers, the generated tree may overfit due to many branches.
- To avoid overfitting, prune the tree so that it is not too specific.

Prepruning

- 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 prepruning.

Postpruning

- 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.

5.2.5 Examples of ID3

Ex. 5.2.1 : Apply ID3 on the following training dataset from all electronics customer database and extract the classification rule from the tree.

Table P. 5.2.1 : 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

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

$$= 0.940 - 0.694 = 0.246$$

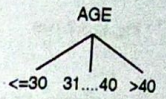
Similarly, Gain (income) = 0.029

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

$$Gain(S_{<=30}, Income) = I(p, n) - E(Income)$$

$$= 0.971 - 0.4 = 0.571$$

(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

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

$$= 0.971 - 0 = 0.971$$

(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 :

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

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

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".

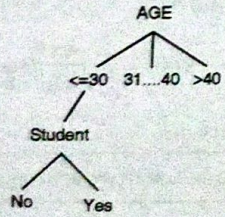


Fig. P. 5.2.1(a)

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

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

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

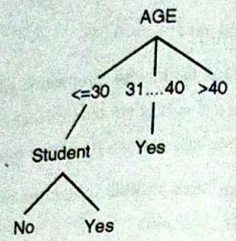


Fig. P. 5.2.1(b)

Step 4 :

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

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

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
>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

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

Therefore,

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

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

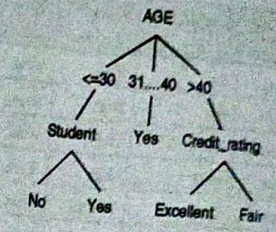


Fig. P. 5.2.1(c)

Output : A Decision Tree for "buys_computer"

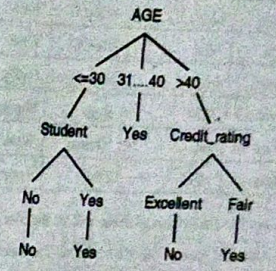


Fig. P. 5.2.1(d) : Decision tree for "buys computer"

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. 5.2.2 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. 5.2.2 : 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).

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 Gain(T, outlook) = I (p, n) – E (outlook)

= 0.940 – 0.694 = 0.246

Similarly,

Gain (T, Temperature) = 0.029

Gain (T, Humidity) = 0.151

Gain (T, 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

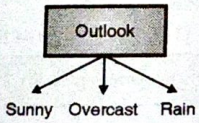


Fig. P. 5.2.2(a)

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.5.2.2, 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. 5.2.2 :

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 (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

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

$= 0.971 - 0.4 = 0.571$

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

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

Hence

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

$= 0.971 - 0 = 0.971$

(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:

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

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

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

Hence Gain(T_{sunny}, Wind) = I (p, n) – E (Wind)

= 0.971 – 0.951 = 0.02

Therefore,

Gain(T_{sunny}, Humidity) = 0.970

Gain(T_{sunny}, Temperature) = 0.570

Gain(T_{sunny}, Wind) = 0.02

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

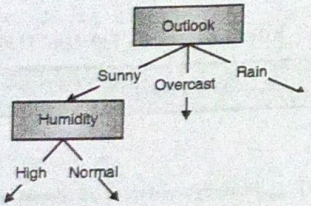


Fig. P. 5.2.2(b)

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.5.2.2, 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.

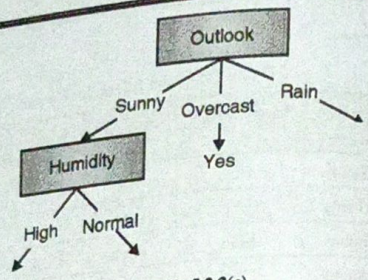


Fig. P. 5.2.2(c)

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. 5.2.2, 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 :

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

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

Hence Gain(T_{rain}, Wind) = I (p, n) – E (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 (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

Gain(T_{Rain}, temperature) = I (p, n) – E (temperature)

= 0.970 – 0.951 = 0.019

Therefore,

Gain(T_{rain}, Temperature) = 0.019

Gain(T_{rain}, Wind) = 0.970

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

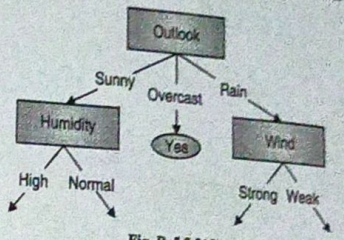


Fig. P. 5.2.2(d)

Therefore the final decision tree is :

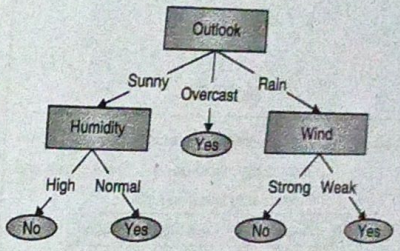


Fig. P. 5.2.2(e) : Decision tree for play tennis

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. 5.2.3 : 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 :

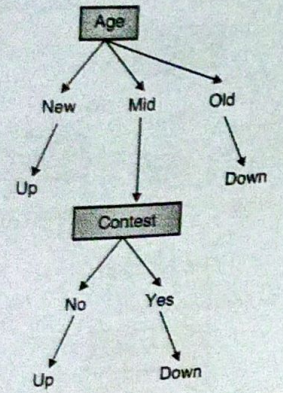


Fig. P. 5.2.3

Ex. 5.2.4 : 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 (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

Gain(S, Income) = I (p, n) – E (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 (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

Gain(S, age) = I (p, n) – E (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).

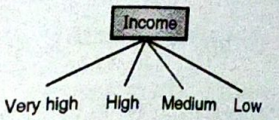


Fig. P. 5.2.4(a)

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

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

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 :

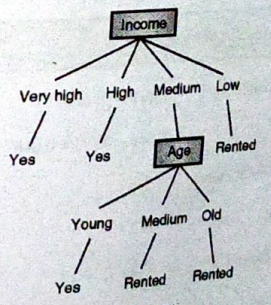
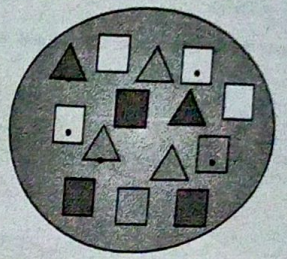


Fig. P. 5.2.4(b)

Ex. 5.2.5 : 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

So information gain = I (p, n)

$$= -\frac{p}{p+n}\log_2\frac{p}{p+n} - \frac{n}{p+n}\log_2\frac{n}{p+n}$$

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

$= -(9/14)\log2(9/14) - (5/14)\log2(5/14)$

$= 0.940$

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

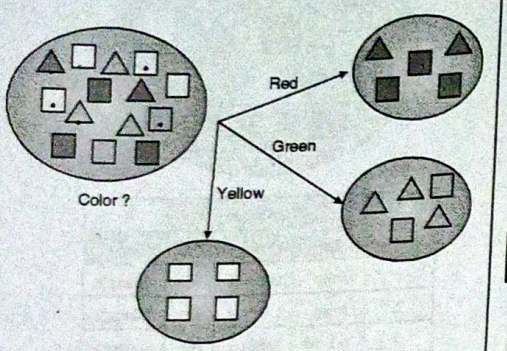


Fig. P. 5.2.5(a)

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 :

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

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

$= 0.694$

Note : S is the total training set.

Hence $Gain(S, color) = I(p, n) - E(Color)$

$= 0.940 - 0.694 = 0.246$

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 :

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

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

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 :

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

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

$= 0.892$

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.

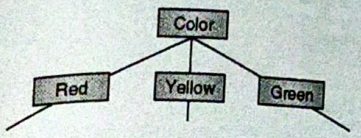


Fig. P. 5.2.5(b)

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 :

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

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

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. 5.2.5(c).

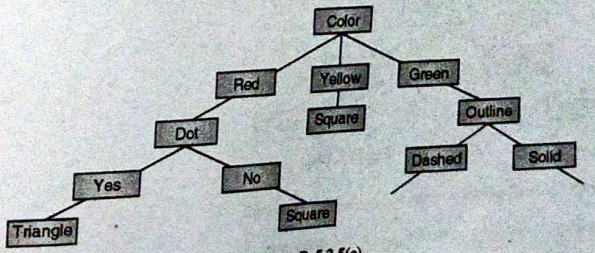


Fig. P. 5.2.5(c)

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.

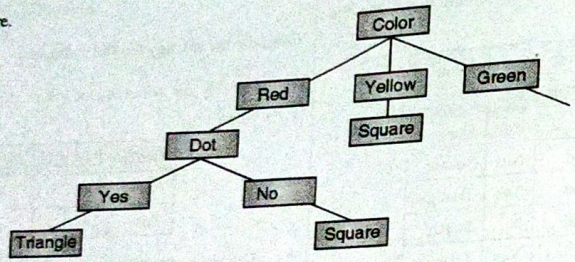


Fig. P. 4.2.5(d)

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

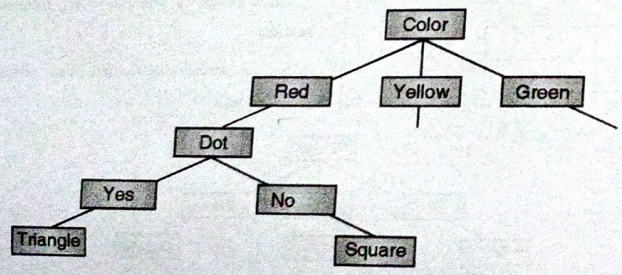


Fig. P. 5.2.5(e)

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

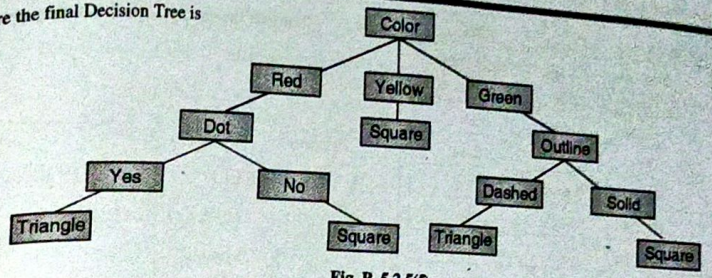


Fig. P. 5.2.5(f)

Ex. 5.2.6 : Apply statistical based algorithm to obtain the actual probabilities of each event to classify the new tuple as a tall. Use the following data

NB

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|Short) = 1/4 * 0 = 0

P(t|Medium) = 1/3 * 0 = 0

P(t|Tall) = 2/2 * 1/2 = 0.5

Therefore likelihood of being short

= p(t|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 / t) = (P(t|short) * p(short))/ P(t)

= (0 * 4 /9)/0.11 = 0

Similarly P(Medium|t) = (0 * 3/9) /0.11 = 0

P(Tall|t) = (0.5 * 2/9)/0.11 = 1

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

Ex. 5.2.7 : 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,

Information Gain = I(p,q,n)

$$= -(p/(p+q+n))\log_2(p/(p+q+n)) - (q/(p+q+n))\log_2(q/(p+q+n)) - (n/(p+q+n))\log_2(n/(p+q+n))$$

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

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

$$I(4,3,3) = 1.571$$

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

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

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.

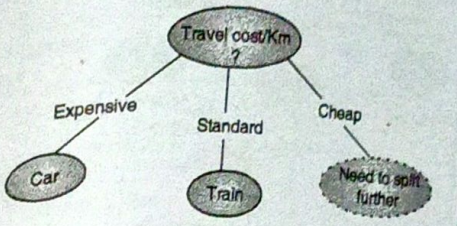


Fig. P. 5.2.7(a)

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 $I(p_i, q_i, n_i)$ 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

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

(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

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

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

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

Hence

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

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

For all gender = Male, Transportation mode= bus

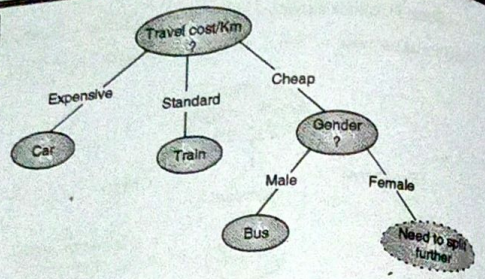


Fig. P. 5.2.7(b)

$S_{female} = 2$

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.

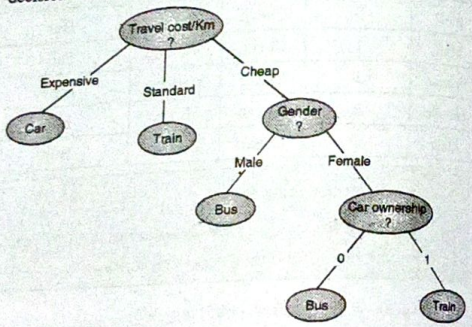


Fig. P. 5.2.7(c)

Ex. 5.2.8 : 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?

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

District	House type	Income	Previous customer	Outcome
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

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
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

So Information gain $= I(p, n)$

$$= -\frac{p}{p+n}\log2\frac{p}{p+n} - \frac{n}{p+n}\log2\frac{n}{p+n}$$

$I(p, n) = I(3, 2)$

$= -(3/5)\log2(3/5) - (2/5)\log2(2/5)$

$= 0.970$

(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

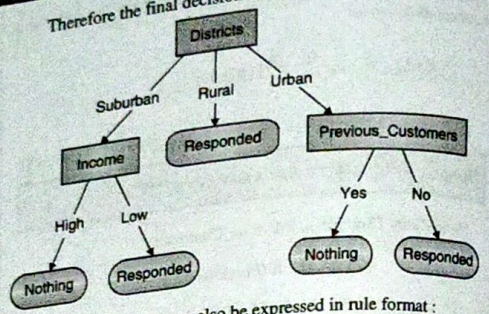
Therefore,

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

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

A set of IT-THEN rules is used for classification in Rule Based Classification. It classifies the record based on collection of IF-THEN rules. The syntax for rules is

"IF condition THEN conclusion"

Example

- If Rule is X → Y where X is condition.
- X is conjunctions of attributes and Y the class label of the rule
- LHS of rule is rule antecedent and RHS is consequent.

Example

- (Refund=No) ∧(Status=Married) → (Cheat=N0)
- (Status = Student) ∧ (age< 30) → (buys_coomputer = YES)

5.3.1 Rule Coverage and Accuracy

- Coverage of a rule: Percentage of tuples that satisfies the antecedent of a rule.
- Accuracy of a rule: Percentage of tuples that satisfy both the antecedent and consequent of a rule. i.e. percentage of tuples which are correctly classified.

Formulae

Coverage (Rule) = Number of tuples covered by Rule / number of tuples in dataset D

Accuracy (Rule) = Number of tuples correctly classified by Rule / Number of tuples covered by Rule

Example

Tid	Refund	Marital Status	Taxable Income	Class
1	Yes	Single	125K	No
2	No	Married	100K	No
3	No	Single	70K	No
4	Yes	Married	120K	No
5	No	Divorced	95K	Yes
6	No	Married	60K	No
7	Yes	Divorced	220K	No
8	No	Single	85K	Yes
9	No	Married	75K	No
10	No	Single	90K	Yes

(Status=Single) → No

Coverage = 4/10 = 0.4 = 40%

Accuracy = 2/4 = 0.5 = 50%

5.3.2 Characteristics of Rule-Based Classifier

1. **Mutually exclusive rules :** Every record of dataset is covered by at most one rule of classifier and the rules are independent of each other
2. **Exhaustive rules :** Classifier generates rules for every possible combination of attribute values and each record is covered by at least one rule.

Example

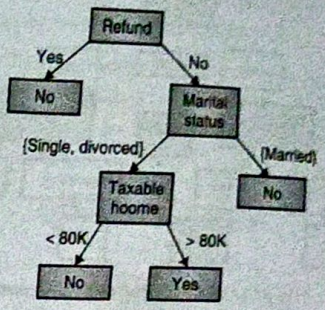


Fig. 5.3.1

Classification rules

(Refund=Yes)==> No

(Refund = No, Marital Status = {Single, Divorced}, Taxable Income<80K) ==> No

(Refund = No, Marital Status = {Single, Divorced}, Taxable Income>80K)==> Yes

(Refund=No, Marital Status={Married}) ==> No

In the above example

- Rules are mutually exclusive and exhaustive.
- Rule set contains as much information as the tree.

Extract the rules from decision tree

- 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.
- 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.

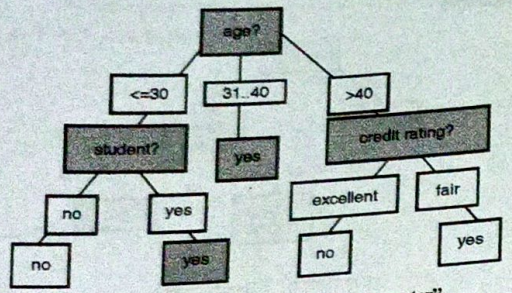


Fig. 5.3.2 : Decision Tree for "Buys_Computer"

Rule extraction from above buys_computerdecision-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"

Syllabus Topic : Rule Induction Using a Sequential Covering Algorithm

5.4 Rule Induction Using a Sequential Covering Algorithm

- Using sequential covering algorithm , IF-THEN rules can be extracted without generating a decision tree from training data.
- The rules can be learned one at a time.
- Sequential covering algorithms are the most widely used approach to mining disjunctive sets of classification rules
- Some of sequential covering algorithms are
  - AQ,
  - CN2
  - more recent RIPPER
- A basic sequential covering algorithm given by MichelineKamber is given below :

Algorithm

Sequential covering. Learn a set of IF-THEN rule for classification.

Input

D, a data set of class-labeled tuples;

Att_vals, the set of all attributes and their possible values.

Output

A set of IF-THEN rules.

Method

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

This algorithm basic functionality is :

1. Start from an empty rule
2. Grow a rule using the Learn-One-Rule function
3. Remove training records covered by the rule
4. Repeat Step (2) and (3) until stopping criterion is met

Syllabus Topic : Bayesian Belief Networks

5.5 Bayesian Belief Networks

- In Bayesian Belief network, conditional independence is defined between subsets of variables
- The network provides a graphical model of causal relationship on which learning can be performed

- A trained network can be used for classification.
- Bayesian Belief networks are also known as belief networks, bayesian networks and probabilistic networks.
- A belief network is defined by following two components.
  - A directed acyclic graph.
  - A set of conditional probability tables.

A directed acyclic graph

- Each node represents a random variable.
- Variables may be discrete or continuous valued.
- Variables may correspond to actual attributes or to hidden variables.
- Each arc represents a probabilistic dependence.
- if an arc is drawn from a node A to node B, then A is a parent or immediate predecessor of B and B is a descendent of A.
- Each variable is conditionally independent of its non descendants in the graph, given its parents.
- An example of a portion of Belief network is shown in Fig. 5.5.1 :

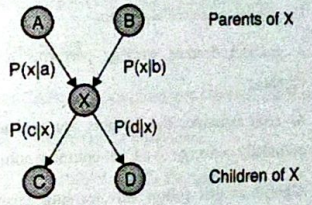


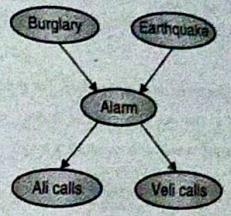
Fig. 5.5.1

- A portion of a belief network, consisting of a node X, having variable values (x1, x2, ...), its parents (A and B), and its children (C and D)

Example 1

- You have a new burglar alarm installed at home.
- It is fairly reliable at detecting burglary, but also sometimes responds to minor earthquakes.
- You have two neighbors, Ali and Veli, who promised to call you at work when they hear the alarm.
- Ali always calls when he hears the alarm, but sometimes confuses telephone ringing with the alarm and calls too.
- Veli likes loud music and sometimes misses the alarm.
- Given the evidence of who has or has not called, we would like to estimate the probability of a burglary.

The Bayesian network for the burglar alarm example. Buglary (B) and earthquake (E) directly affect the probability of the alarm (A) going off, but whether or not Ali calls (AC) or Veli calls (VC) depends only on the alarm.



Burglary true

P(B = T)	P(B = F)
0.001	0.999

P(E = T)	P(E = F)
0.002	0.998

B	E	P(A = T)	P(A = F)
T	T	0.95	0.05
T	F	0.94	0.06
F	T	0.29	0.71
F	F	0.001	0.999

A	P(VC = T)	P(VC = F)
T	0.70	0.30
F	0.01	0.99

A	P(AC = T)	P(AC = F)
T	0.90	0.10
F	0.05	0.95

- What is the probability that the alarm has sounded but neither a burglary nor an earthquake has occurred, and both Ali and Veli Call ?

$$P(AC, VC, A, \neg B, \neg E)$$
$$= P(AC \wedge A)\, P(VC \wedge A)\, P(A \wedge \neg B, \neg E)\, P(\neg B)\, P(\neg E)$$
$$= 0.90 \times 0.70 \times 0.001 \times 0.999 \times 0.998$$
$$= 0.00062$$

(Capital letter represent variables having the value true and $\neg$ represents negation)

Example 2

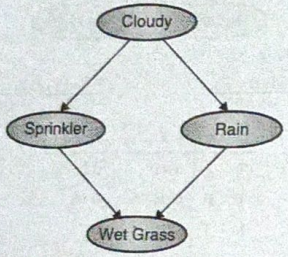
- Suppose we observe the fact that the grass is wet. There are two possible causes for this : either it rained, or the sprinkler was on. Which one is more likely ?

$$P(S|W) = \frac{P(S, W)}{P(W)} = \frac{0.2781}{0.6471} = 0.430$$

$$P(R|W) = \frac{P(R, W)}{P(W)} = \frac{0.4581}{0.6471} = 0.708$$

- We see that it is more likely that the grass is wet because it rained.

Another Bayesian network example. The event that the grass being wet (W = true) has two possible causes : either the water sprinkler was on (S = true) or it rained (R = true).



P(C = T)	P(C = F)
0.50	0.50

C	P(S = T)	P(S = F)
T	0.10	0.90
F	0.50	0.50

C	P(R = T)	P(R = F)
T	0.80	0.20
F	0.20	0.80

S	R	P(W = T)	P(W = F)
T	T	0.99	0.01
T	F	0.90	0.10
F	T	0.90	0.10
F	F	0.00	1.00

Applications of Bayesian Networks

- Machine learning
- Statistics
- Computer vision
- Natural language Processing
- Speech recognition
- Error-control codes
- Bioinformatics Medical diagnosis
- Weather forecasting

Syllabus Topic : Training Bayesian Belief Networks

5.6 Training Bayesian Belief Networks

- Training a network has a number of possible scenarios
  - The network topology (nodes and arcs) may be constructed by a human experts or inferred from the data
  - The network variables may be observable or hidden in all or some of the training tuples.
- Training the Network if the network topology is known and the variables are observable
  - Compute the CPT (Conditional Probability table) entries
- Training the Network if the network topology is known and some of the variables are hidden
  - Use gradient descent algorithm.
  - A gradient descent strategy performs a greedy hill climbing.
  - At each iteration, the weights are updated and will eventually converge to a local optimum solution.
  - Algorithms that follow this learning form are called Adaptive probabilistic networks.

Syllabus Topic : Classification Using Frequent Patterns : Associative Classification

5.7 Classification Using Frequent Patterns : Associative Classification

- Frequent patterns generated from association can be used for classification is called **associative classification**. Initially association rules are generated from frequent patterns and used for classification.

Steps involved in associative classification :

1. **Find** frequent itemsets, i.e commonly occurring attribute–valuepairs in the data.
2. **Generate association rules by analysing** frequent itemsets as per the class by considering class confidence and support criteria.
3. Organize the rules to form a rule-based classifier.

5.7.1 CBA

- One of the earliest and simplest algorithms for associative classification is **CBA** (Classification Based on Associations).

Steps in CBA

- **Mine** for CARs satisfying support and confidence thresholds
- **Sort** all CARs based on confidence
- **Classify** using the rule that satisfies the query and has the highest confidence

5.7.2 CMAR

Classification based on Multiple ARs (CMAR)

Steps in CMAR

- **Mine** for CARs satisfying support and confidence thresholds
- **Sort** all CARs based on confidence
- **Find** all CARs which satisfy the given query
- **Group** them based on their class label
- **Classify** the query to the class whose group of CARs has the maximum **weight**.

Syllabus Topic : Lazy Learners

5.8 Lazy Learners : (or Learning from your Neighbors)

- Classification methods can be classifies as Eager Learners and Lazy Learners.
- Eager learners are those classification techniques in which a given set of training tuples contructs a generalised model and then uses the same to classify a previously unseen tuple.
- A learned model as being ready and eager to classify an unseen tuple.
- Examples of Eager learners are Decision tree induction, Bayesian Classification, Rule based classification, classification by backpropogation, support vector machines and classification based on Association rule Mining.
- In lazy Learner approach, the learner waits until the last minute before doing model construction to classify a given test tuple.
- A lazy learner approach performs generalization only when its sees a test tuple. Until then it only stores training tuple or does very little processing.
- Lazy learners do very less amount of processing when training tuples are presented and does more amount of work when a classification or numeric prediction is to be done.
- Since Lazy learners stores training tuples or instances, it is also known as instance -based learners.

Disadvantages of lazy learners

- Lazy Learners are computationally expensive.
- Require efficient storage techniques
- Offers less insight into the data structures

Advantages of lazy learners

- Well suited to implementation on parallel
- Supports incremental learning
- Able to model complex decision spaces having hyperpolygonal shapes that may not be describable by any other learning algorithms
- Two examples of Lazy learners , K-nearest neighbors and case based reasoning

Syllabus Topic : K-Nearest-Neighbor Classifiers

5.8.1 K-Nearest-Neighbor Classifiers

→ **(SPPU - May 16, May 17)**

Q. Explain with suitable example.
(i) K-Nearest-Neighbor Classifier
May 16, May 17, 4 Marks

Lazy learner.

- K-Nearest Neighbors is used in the field of Pattern Recognition
- It learns by analogy, i.e. by comparing a given test tuple with training tuples that are similar to it
- The training tuples have n attributes, every tuple represents a point in n-dimensional space
- All training tuples are stored in an n-dimensional pattern space
- K-nearest neighbors searches the pattern space for the k training tuples that are closest to the unknown test tuple.
- The k training tuples are the k "nearest neighbors" of the unknow tuple.
- Closeness is defined using distance metrics such as Euclidean distance
- The Manhattan (city block) distance or other distance measurements, may also be used.
- **To find the** Euclidean Distance between two points or tuples, the formula is given below.

Let Y1 = $\{y_{11}, y_{12}, y_{13}, \ldots\ldots y_{1n}\}$

and Y2 = $\{y_{21}, y_{22}, y_{23}, \ldots\ldots y_{2n}\}$

distance(Y1,Y2) = $(y_{11} - y_{21})^2$

- KNN classifiers can be extremely slow when classifying test tuples O(n).
- By simple presorting and arranging the stored tuples into search tree, the number of comparisons can be reduced to O(logN).
- Example : if k = 5, it selects the 5 nearest neighbor as shown in Fig. 5.8.1.

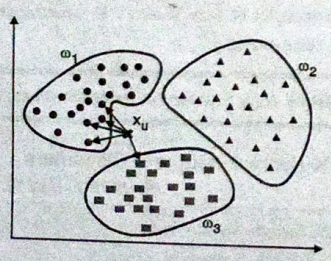


Fig. 5.8.1 : knn for k = 5

5.8.2 CBR (Case Based Reasoning)

- Case-based reasoning means using old experiences to understand and solve new problems. In case-based reasoning, a reasoner remembers a previous situation similar to the current one and uses that to solve the new problem.
- Case-based reasoning can mean adapting old solutions to meet new demands; using old cases to explain new situations; using old cases to critique new solutions; or reasoning from precedents to interpret a new situation (much like lawyers do) or create an equitable solution to a new problem (much like labor mediators do).
- To solve a current problem: the problem is matched against the cases in the case base, and similar cases are retrieved. The retrieved cases are used to suggest a solution which is reused and tested for success. If necessary, the solution is then revised. Finally, the current problem and the final solution are retained as part of a new case.
- All case-based reasoning methods have in common the following process :
  - retrieve the most similar case (or cases) comparing the case to the library of past cases;
  - reuse the retrieved case to try to solve the current problem;
  - revise and adapt the proposed solution if necessary;
  - retain the final solution as part of a new case.
- Retrieving a case starts with a (possibly partial) problem description and ends when a best matching case has been found. The subtasks involve :
  - identifying a set of relevant problem descriptors;
  - matching the case and returning a set of sufficiently similar cases (given a similarity threshold of some kind);
  - selecting the best case from the set of cases returned.
  - Some systems retrieve cases based largely on superficial syntactic similarities among problem descriptors, while advanced systems use semantic similarities.
- Reusing the retrieved case solution in the context of the new case focuses on: identifying the differences between the retrieved and the current case; and identifying the part of a retrieved case which can be transferred to the new case. Generally, the solution of the retrieved case is transferred to the new case directly as its solution case.
- A CBR tool should support the four main processes of CBR: retrieval, reuse, revision and retention. A good tool should support a variety of retrieval mechanisms and allow them to be mixed when necessary. In addition, the tool should be able to handle large case libraries with retrieval time increasing linearly (at worst) with the number of cases.

Applications of CBR

Case based reasoning first appeared in commercial tools in the early 1990's and since then has been sued to create numerous applications in a wide range of domains :

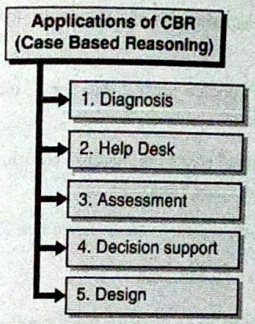


Fig. 5.8.2 : Applications of CBR

➜ 1. Diagnosis

Case-based diagnosis systems try to retrieve past cases whose symptom lists are similar in nature to that of the new case and suggest diagnoses based on the best matching retrieved cases. The majority of installed systems are of this type and there are many medical CBR diagnostic systems.

➜ 2. Help Desk

Case-based diagnostic systems are used in the customer service area dealing with handling problems with a product or service.

➜ 3. Assessment

Case-based systems are used to determine values for variables by comparing it to the known value of something similar. Assessment tasks are quite common in the finance and marketing domains.

➜ 4. Decision support

- In decision making, when faced with a complex problem, people often look for analogous problems for possible solutions. CBR systems have been developed to support in this problem retrieval process (often at the level of document retrieval) to find relevant similar problems. CBR is particularly good at querying structured, modular and non-homogeneous documents.

➜ 5. Design

- Systems to support human designers in architectural and industrial design have been developed. These systems assist the user in only one part of the design process, that of retrieving past cases, and would need to be combined with other forms of reasoning to support the full design process.

□□□

CHAPTER 6

Unit VI

Multiclass Classification

error correcting codes can be used to improve accuracy of multiclass classification.
e.g. Hamming distance method — guess closest class in case of errors.

Syllabus Topics

Multiclass Classification, Semi-Supervised Classification, Reinforcement learning, Systematic, Learning, Wholistic learning and multi-perspective learning. Metrics for Evaluating Classifier Performance : Accuracy, Error Rate, precision, Recall, Sensitivity, Specificity; Evaluating the Accuracy of a Classifier : Holdout Method, Random Sub sampling and Cross-Validation.

Syllabus Topic : Multiclass Classification

Kamber - 438

6.1 Multiclass Classification

6.1.1 Introduction to Multiclass Classification

- In Multiclass classification, there are N different classes.
- Each of the training point belongs to one of N different classes.
- The goal is to predict a class label to an Unknown tuple.

Two Approaches in multiclass classification

eg part of speech tagging, Topic classfn

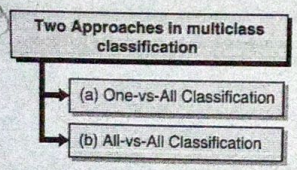


Fig. 6.1.1 : Two Approaches in multiclass classification

(a) One-vs-All Classification

- Select a good technique for building a Binary Classifier (e.g. SVM).
- Build N different Binary classifiers.
- Classifier I is trained using tuples of class i as the positive class and the remaining tuples as the negative class.
- To classify an unknown tuple, X, the set of classifiers vote collectively.
- If classifier i predicts positive class for X, then class I gets one vote.
- If the classifier i predicts the negative class for X, then each of the classes except I gets one vote.
- The class with maximum votes is assigned to X.

(b) All-vs-All Classification

- This is an alternative approach that learns a classifier for each pair of classes.
- Given N classes, construct n(n-1)/2 classifiers.
- A classifier is trained using tuples of the two classes.
- For classifying an unknown tuple X, each classifier votes.
- The class with maximum number of votes is assigned to the unknown tuple X.
- All-vs-All approach is better as compared to One-vs-All.

Syllabus Topic : Semi-Supervised Classification

6.2 Semi-Supervised Classification

- A semi-supervised classification uses labeled and unlabeled data to build a classifier.
- Following are the two forms of Semi-Supervised classification :

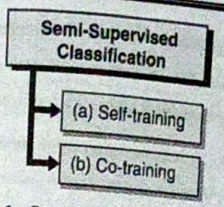


Fig. 6.2.1 : Semi-Supervised Classification

(a) Self-training

- It is one of the simplest form of semi-supervised classification.
- It first builds the classifier using the labeled data.
- Then the classifier tries to label the unlabeled data.
- The tuple with most confident label prediction is added to the labeled data.
- This process is repeated.
- One of the drawback of the method is that it may reinforce errors.

(b) Co-training

- This is another form of semi-supervised classification.
- In this approach two or more classifiers teach each other.
- Each learner uses different and independent set of features for each tuple.
- If the feature set is split into two sets and train two classifiers f1 and f2.
- Then f1 and f2 are used to predict the class labels for the unlabeled data.
- Each classifier then teaches the other in that tuple having the most confident prediction from f1 is added to the set of labeled data for f2(along with its label).
- The tuple having the most confident prediction from f2 is added to the set of labeled data for f1.
- Cotraining is less error prone as compared to self training.

Syllabus Topic : Reinforcement Learning

6.3 Reinforcement Learning

(SPPU - Dec. 16, May 17)

Q. Briefly explain the reinforcement learning. **Dec. 16, May 17, 6 Marks**

6.3.1 Introduction to Reinforcement

- Reinforcement learning is based on goal-directed learning from interaction.
- Reinforcement learning maximizes a numerical reward signal by mapping the situations to actions.
- In machine learning learner knows which action to take but in reinforcement learning, learner doesn't know the action, it discover which action gives most reward signal.
- Reinforcement learning is by characterizing a learning problem not by method.
- Reinforcement learning is different from supervised learning, as alone it is not adequate for learning from interaction.
- In this case, the agent has to act and get the learning through experience.
- All reinforcement learning agents have explicit goals and are intelligence to find the aspects of their environments, so accordingly they can select the actions to control their environments.

Example

- In a chess game, player makes a move based on planning of move, expecting possible replies and even counter replies. Then player takes immediate and spontaneous judgment and plays the move.
- In this example the agent (player) uses its experience to improve its performance and evaluate positions to improve his play over the time.

6.3.2 Elements of Reinforcement Learning

Main sub-elements of a reinforcement learning system are :

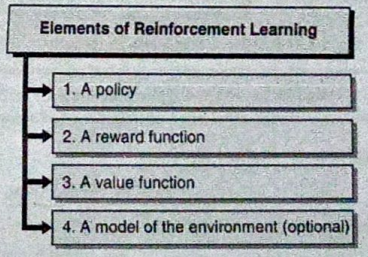


Fig. 6.3.1 : Elements of Reinforcement Learning

➔ **1. A policy**

The learning agent's manner of behaving at a given time.

➔ **2. A reward function**

The purpose in a reinforcement learning problem.

➔ **3. A value function**

What is good over the future or in the long run?

➔ **4. A model of the environment (optional)**

It is used for planning and predict the resultant next state and next reward.

6.3.3 Reinforcement Function and Environment Function

- It uses knowledge acquired so far and while exploring, the action leads to learning through either rewards or penalties.
- Rewards are related to specific actions and value function is the collective effect.
- To get the correct responses, environment needs to be model so that it can accept the inputs from changing scenarios and finally can produce the optimized value.
- Fig 6.3.2 shows the typical reinforcement-learning scenario where action lead to reward.
- So for every action, there are environment as well as reinforcement functions.

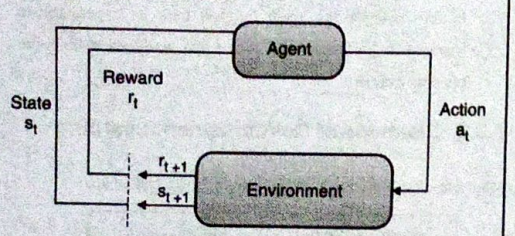


Fig. 6.3.2 : Reinforcement-learning scenario

Syllabus Topic : Wholistic Learning

6.3.4 Whole System Learning

➔ **(SPPU - Dec. 16)**

Q. What is meant by whole system learning ?

Dec. 16. 4 Marks

- Systematic learning considers complete system, its subsystem and the interactions between the systems for learning. Based on this it makes the decisions.
- It builds the systematic information which is useful for analysis.
- Systematic learning is interactive and driven by environment which is specific to the problem.

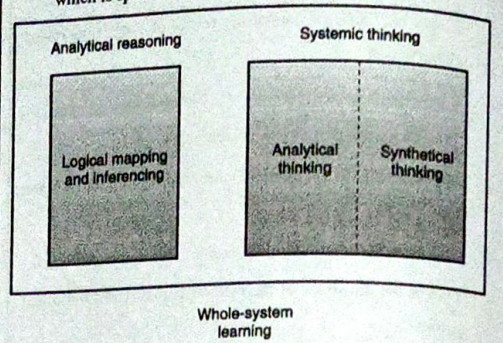


Fig. 6.3.3 : Whole-System learning

Syllabus Topic : Systematic Learning

6.4 Systematic Learning

- Systemic learning is about understanding systems, subsystems, and systemic impact of various actions, decisions within the system, and decisions in a systemic environment.
- It is more about learning about the actions and interactions from a systemic perspective.
- It identifies a system and builds systemic information.
- Building of this information is done from the analysis of perspectives with reference to systemic impact.
- This learning includes multiple perspectives and collection of data from all parts of system.
- It also includes data and decision analysis related to impact.
- Decisions are part of a learning process, and the learning takes place with every decision and its outcome.
- The knowledge augmentation takes place with every decision and decision-based learning.

- This learning is interactive and driven by environment, which include different parts of the system.
- The system dependency of learning is controlled and is specific to the problem and system.

Syllabus Topic : Multi-Perspective Learning

6.5 Multi-Perspective Decision Making for Big Data and Multi-Perspective Learning for Big Data

➔ **(SPPU - Dec. 15, May 16, Dec. 16, May 17, Dec. 17)**

Q. Write a note on multi-perspective learning.

Dec. 15, May 17, Dec. 17, 4 Marks

Q. Write short note on multi-perspective decision making.

May 16, 6 Marks

Q. What is meant by multi perspective decision making? Explain.

Dec. 16, 6 Marks

6.5.1 Fundamental of Multi-perspective Decision Making and Multi-perspective Learning

- Multi-perspective Learning is needed for Multi-perspective Decision making
- Multi-perspective Learning refers to learning from knowledge and information collected from different perspectives.
- Multi-perspective Learning builds knowledge from various perspectives so that it can be used for decision making process.

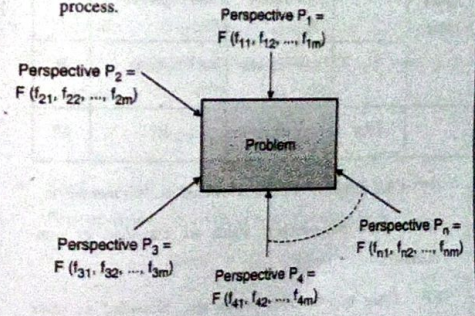


Fig. 6.5.1 : Multi-perspective Learning

- The perspective includes context, scenario and situation, the way we look at a particular decision problem.
- In Fig. 6.5.1, P_1, P_2, P_3,..., P_n refers to different perspective in the Learning process.
- Each of this perspective is represented as a function of features.
- There may be an overlap among the perspectives.
- Feature difference may be there as some features which possibly visible form one perspective may not be visible from the other perspective.
- The representative feature set should contain all the possible features.

6.5.2 Influence Diagram

- Perspective based information can be represented as an influence diagram.
- Helps in getting the context of the decision making
- It is a graphical representation of the decision situation
- It shows the relationships among objects and actions.
- These relationships may be mapped to probabilities.
- The Fig. 6.5.2 represents an influence diagram for a market scenario and relationship between marketing and budget, product, price, cost and profit.

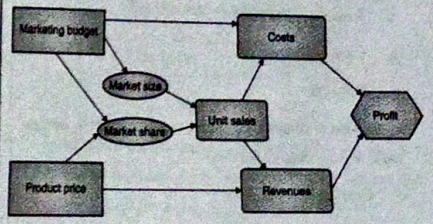


Fig. 6.5.2 : Influence Diagram

- The relationships may also be represented using a decision tree is shown in Fig. 6.5.3.
- Based on parameters measurements, the decision path of decision tree is decided.
- Decision rules can be represented on a decision tree.

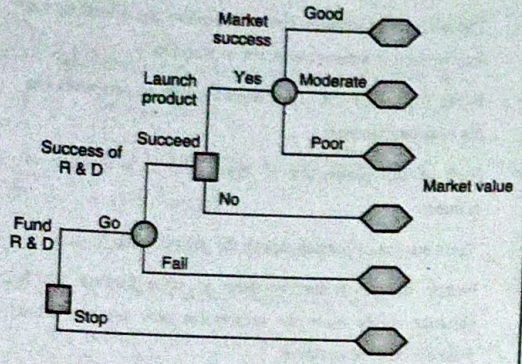


Fig. 6.5.3 : Decision tree

Syllabus Topic : Metrics for Evaluating Classifier Performance : Accuracy, Error Rate, Precision, Recall, Sensitivity, Specificity

6.6 Model Evaluation and Selection

- 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 subsampling
  - Cross-validation
  - Bootstrap
- If we want to compare classifiers to select the best one then the following methods are used :
  - Confidence intervals
  - Cost-benefit analysis and Receiver Operating Characteristic (ROC) Curves

6.6.1 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**
   - Trainingset, 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

$$\text{Sensitivity} = TP/P$$

6. **Specificity :** True Negative recognition rate which is the proportion of negative tuples that are correctly identified

$$\text{Specificity} = TN/N$$

7. **Classifier accuracy or recognition rate :** Percentage of test set tuples that are correctly classified

$$\text{Accuracy} = (TP + TN)/All$$

OR

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

Accuracy is also a function of sensitivity and specificity :

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

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

Error rate = 1 – accuracy, or Error rate = (FP + FN)/All

Or

$$\text{Error rate} = \frac{FP + FN}{P + N}$$

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

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

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

$$\text{Recall} = \frac{|TP|}{|TP| + |FN|}$$

11. **Fmeasure (F_1orF-score) :** Harmonic mean of precision and recall,

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

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

$$F_\beta = \frac{(1+\beta^2) \times \text{Precision} \times \text{Recall}}{\beta^2 \times \text{Precision} + \text{Recall}}$$

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 but it can be minimized, so low error rate is always preferable.

Syllabus Topic : Evaluating the Accuracy of a Classifier : Holdout Method

6.6.2 Holdout

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

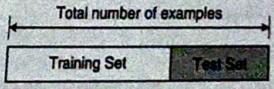


Fig. 6.6.1

- To train the classifier, training data set is used and once the classifier is constructed then 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 Sub-sampling

6.6.3 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.

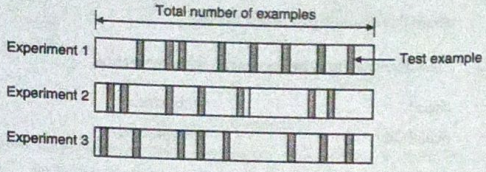


Fig. 6.6.2

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

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

Syllabus Topic : Cross-Validation (CV)

6.6.4 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).

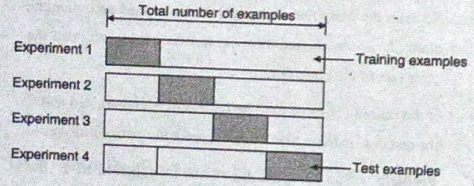


Fig. 6.6.3

- **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.

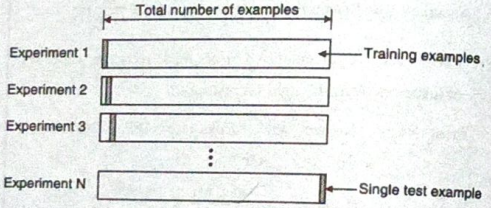


Fig. 6.6.4

6.7 Solved University Questions and Answers

Q. 1 For each of the following queries, identify and write the type of data mining task.

i) Find all credit applicants who are poor credit risks.

ii) Identify customers with similar buying habits,

iii) Find all items which are frequently purchased with milk. **(Dec. 15, 6 Marks)**

Ans. :

i) **Classification :** Based on credit applications , customers can be classified in various classes like poor, medium and high credit risk types of customers

ii) **Clustering :** Clusters can be formed based on similar type of buying patterns. Then the customers belongs to those clusters can be identified.

iii) **Association :** Various items which has been frequently purchased with milk can be identified with association data mining task. Based on the support and confidence, milk can be associated with those frequent items.

Q. 2 Consider the ten records given below : **(Dec. 15, 8 Marks)**

ID	Income	Credit	Class	X_1
1	4	Excellent	h_1	X_4
2	3	Good	h_1	X_7
3	2	Excellent	h_1	X_2
4	3	Good	h_1	X_7
5	4	Good	h_1	X_8
6	2	Excellent	h_1	X_2
7	3	Bad	h_2	X_{11}
8	2	Bad	h_2	X_{10}
9	3	Bad	h_3	X_{11}
10	1	Bad	h_4	X_9

Calculate the prior probabilities of each of the class h_1, h_2, h_3, h_4 and probabilities for data points X_1, X_4, X_7 and X_8, belonging to the class h_1.

Ans. :

Assign ten data values for all combinations of credit and income :

	1	2	3
Excellent	x_1	x_3	x_6
Good	x_2	x_4	x_5
Bad	x_7	$x_{8/9}$	x_{10}

From training data

$$P(h_1) = \frac{6}{10} = 60\ \%$$

$$P(h_2) = \frac{2}{10} = 20\ \%$$

$$P(h_3) = \frac{1}{10} = 10\ \%$$

$$P(h_4) = \frac{1}{10} = 10\ \%$$

Q. 3 What are similarities and differences between reinforcement learning and artificial intelligence algorithms ? **(Dec. 16, 5 Marks)**

Ans. :

Reinforcement Learning	Artificial Intelligence
Reinforcement learning is a branch of Artificial Intelligence and type of machine learning algorithms.	Artificial Intelligence (AI) is an area in computer science that emphasizes the creation of intelligent machines.
To maximize its performance, it allows software agents and machines to automatically determine the ideal behavior within a specific context.	One of the fundamental building blocks of artificial intelligence (AI) solutions is Learning. From a conceptual standpoint, learning is a process that improves the knowledge of an AI program by making observations about its environment.
The feedback received from the environment is used by the machine or software agent to learn its behavior.	Based on the feedback characteristics, AI Learning models can be classified into supervised, unsupervised, semi-supervised or reinforced.
Applications : Manufacturing, inventory management, delivery management, finance sector.	**Applications :** Expert systems, Speech recognition and Machine vision, Natural Language processing.

Similarity between reinforcement learning and systematic machine learning :

The similarity between reinforcement learning and systematic machine learning is both are Adapts to evolving environment.

□□□

① Sequential covering algorithm
② a) specificity
b) sensitivity
③ Wholistic learning vs multiperspective learning