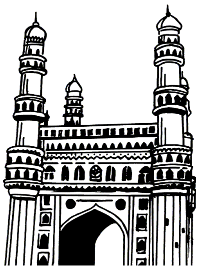
Rahul's ✔

Topper's Voice

NEW SYLLABUS



M.Sc.

(COMPUTER SCIENCE)

II Year III Sem

(Osmania University)

LATEST EDITION
2018-2019

DATA MINING

- ☞ **Study Manual**
- ☞ **Solved Model Papers**
- ☞ **Viva Questions**

Price
` . ~~199-00~~
` . 169-00

- by -

WELL EXPERIENCED LECTURER

Rahul Publications™
Hyderabad. Ph : 66550071, 9391018098

M.Sc.

II Year III Sem

DATA MINING

☞ **Study Manual**

☞ **Solved Model Papers**

☞ **Viva Questions**

Inspite of many efforts taken to present this book without errors, some errors might have crept in. Therefore we do not take any legal responsibility for such errors and omissions. However, if they are brought to our notice, they will be corrected in the next edition.

Price
` . ~~199-00~~
` . 169-00

Sole Distributors : **☎ : 66550071, Cell : 9391018098**

VASU BOOK CENTRE

Shop No. 3, Beside Gokul Chat, Koti, Hyderabad.

Maternity Hospital Opp. Lane, Narayan Naik Complex, Koti, Hyderabad.
Near Andhra Bank, Subway, Sultan Bazar, Koti, Hyderabad -195.

CONTENTS

DATA MINING

SYLLABUS

UNIT – I

Introduction to Data Mining : Why data mining? What is data mining? What kinds of data can be mined? What kinds of patterns can be mined? Which technologies are used? Major issues in data mining. Getting to Know Your Data: data objects and attribute types, basic statistical description of data, data visualization, measuring data similarity and dissimilarity. Data Processing: an overview, data cleaning, data integration, data reduction, data transformation and data discretization.

UNIT – II

Data Warehousing and Online Analytical Processing (OLAP): Basic concepts of data warehouse, data warehouse modelling–data cube and OLAP, data warehouse design and usage, data warehouse implementation, data generalization by attribute–oriented.Data Cube Technology: data cube computation preliminary concepts, data cube computation methods, processing advanced kinds of queries by exploring cube technology, multidimensional data analysis in cube space. Mining Frequent Patterns, Associations, and Correlations: basic concepts, frequent itemset mining methods, mining various kinds of association rules, from association mining to correlation analysis, constraint-based association mining.

UNIT – III

Classification : Basic Concepts, Decision Tree Induction, Bayes Classification Methods, Rule-Based Classification, classification by backpropagation, support vector machines, associative classification, lazy learners, other classification methods.Cluster Analysis: basic concepts of cluster analysis, partitioning methods, hierarchical methods, density-based methods, evaluation of clustering.

UNIT – IV

Outlier Detection : Outliers and outlier analysis, outlier detection methods, statistical approaches, proximitybased approaches, clustering-based approaches,classification-based approaches.Data Mining Trends and Research Frontiers: mining complex data types, other methodologies of data mining,data mining applications, data mining and society, data mining trends.

Contents

Topic No. **Page No.**

UNIT - I

UNIT - II

UNIT - III

Topic No. **Page No.**

UNIT - IV

FACULTY OF SCIENCE

M.Sc. II Year III Semester Examination

DATA MINING

Time : 3 Hours] **ANSWERS TO MODEL PAPER - I** [Max. Marks : 80

PART - A (8 × 4 = 32)

Answer all questions

Each Question Carries Equal Marks **ANSWERS :**

1. Define the tasks of data mining ? **(Unit - I, S.A. - 3)**

2. Discuss the issues related to datamining. **(Unit - I, S.A. - 6)**

3. What are the different types of OLAP tools ? **(Unit - II, S.A. - 4)**

4. What is association rule in dataminings ? **(Unit - II, S.A. - 13)**

5. Define classification and prediction. **(Unit - III, S.A. - 1)**

6. What is Rule Base classification **(Unit - III, S.A. - 4)**

7. Define the following : **(Unit - IV, S.A. - 2)**

   a) Space based approval b) Graph based approval

8. Different types of mining systems. **(Unit - IV, S.A. - 13)**

PART - B (4 × 12 = 48)

Answer any four from the following

9. a) On what kind of data is used in datamining ? **(Unit - I, Q. 4)**

Or

b) What are the issues in data mining ? **(Unit - I, Q. 8)**

10. a) Mention few approaches to mining multiway computation. **(Unit - II, Q. 23)**

Or

b) Explain FP growth algorithm. **(Unit - II, Q. 22)**

11. a) What are the steps followed in Back propagation ? **(Unit - III, Q. 9)**

Or

b) What are the different clustering methods ? **(Unit - III, Q. 11)**

12. a) Explain the classification of models in outlier detection system. **(Unit - IV, Q. 8)**

Or

b) What is sequential pattern mining in transactional database. **(Unit - IV, Q. 11)**

FACULTY OF SCIENCE

M.Sc. II Year III Semester Examination

DATA MINING

Time : 3 Hours] **ANSWERS TO MODEL PAPER - II** [Max. Marks : 80

PART - A (8 × 4 = 32)

Answer all questions

Each Question Carries Equal Marks **ANSWERS :**

1. Explain the scope of data mining ? **(Unit - I, Q. 2)**

2. Explain the issues in data integration. **(Unit - I, Q. 9)**

3. Write a short notes on star schema in data warehouse. **(Unit - II, Q. 5)**

4. How to implement data cube technology ? **(Unit - II, Q. 9)**

5. Explain partioning by hierarchical methods. **(Unit - III, Q. 9)**

6. Discuss about back propagation classification. **(Unit - III, Q. 8)**

7. Define sampling. **(Unit - IV, Q. 4)**

8. Write a short note on visual mining ? **(Unit - IV, Q. 10)**

PART - B (4 × 12 = 48)

Answer any four from the following

9. a) Discuss in detail different types of data mining techniques. **(Unit - I, Q. 7)**

Or

b) What is an attribute ? What are the different types of attributes. **(Unit - I, Q. 10)**

10. a) Explain in detail about OLAP ? **(Unit - II, Q. 4)**

Or

b) What are the alternative methods used in frequent item sets. **(Unit - II, Q. 25)**

11. a) Explain lazy learners ? **(Unit - III, Q. 7)**

Or

b) Explain different classification methods ? **(Unit - III, Q. 5)**

12. a) What does outlier detection mean ? **(Unit - IV, Q. 2)**

Or

b) Write in brief about the visual and audio datamining techniques. **(Unit - IV, Q. 14)**

FACULTY OF SCIENCE

M.Sc. II Year III Semester Examination

DATA MINING

Time : 3 Hours] **ANSWERS TO MODEL PAPER - III** [Max. Marks : 80

PART - A (8 × 4 = 32)

Answer all questions

Each Question Carries Equal Marks **ANSWERS :**

1. Explain the process of datamining. **(Unit - I, Q. 5)**
2. What is Dataware house and Discuss about knowledge database. **(Unit - I, Q. 8)**
3. What is apriori algorithm in data mining ? **(Unit - II, Q. 15)**
4. What are the benefits of data ware housing ? **(Unit - II, Q. 2)**
5. What is decision tree induction ? **(Unit - III, Q. 2)**
6. Discuss about centroid - based classification with k- mean method. **(Unit - III, Q. 7)**
7. What is information extraction ? **(Unit - IV, Q. 9)**
8. Explain the following : **(Unit - IV, Q. 8)**

   a) Term based method

   b) Phase based method

   c) Concept based method.

PART - B (4 × 12 = 48)

Answer any four from the following

9. a) What is motivated datamining ? What is its importance ? **(Unit - I, Q. 2)**

   Or

   b) Explain about advanced database and its applications with functionalities. **(Unit - I, Q. 6)**

10. a) Discuss in detail about data cube computation method. **(Unit - II, Q. 15)**

    Or

    b) Explain applications of information processing ? **(Unit - II, Q. 8)**

11. a) Explain decision tree induction with algorithm. **(Unit - III, Q. 2)**

    Or

    b) Explain in detail constraint based cluster analysis ? **(Unit - III, Q. 13)**

12. a) Explain statistical approach in data mining. **(Unit - IV, Q. 6)**

    Or

    b) Explain the trends in data mining. **(Unit - IV, Q. 16)**

UNIT I

Introduction to Data Mining : Why data mining? What is data mining? What kinds of data can be mined? What kinds of patterns can be mined? Which technologies are used? Major issues in data mining. Getting to Know Your Data: data objects and attribute types, basic statistical description of data, data visualization, measuring data similarity and dissimilarity. Data Processing: an overview, data cleaning, data integration, data reduction, data transformation and data discretization.

1.1 Introduction

Q1. Explain Data Mining in detail ?

Ans :

We are in an age often referred to as the information age. In this information age, because we believe that information leads to power and success, and thanks to sophisticated technologies such as computers, satellites, etc., we have been collecting tremendous amounts of information. Initially, with the advent of computers and means for mass digital storage, we started collecting and storing all sorts of data, counting on the power of computers to help sort through this amalgam of information. Unfortunately, these massive collections of data stored on disparate structures very rapidly became overwhelming. This initial chaos has led to the creation of structured databases and database management systems (DBMS).

The efficient database management systems have been very important assets for management of a large corpus of data and especially for effective and efficient retrieval of particular information from a large collection whenever needed. The proliferation of database management systems has also contributed to recent massive gathering of all sorts of information. Today, we have far more information than we can handle: from business transactions and scientific data, to satellite pictures, text reports and military intelligence. Information retrieval is simply not enough anymore for decision-making. Confronted with huge collections of data, we have now created new needs to help us make better managerial choices. These needs are automatic summarization of data, extraction of the "essence" of information stored, and the discovery of patterns in raw data.

Data mining is a powerful new technology with great potential to help companies focus on the most important information in their data warehouses. It has been defined as :

- The automated analysis of large or complex data sets in order to discover significant patterns or trends that would otherwise go unrecognised. The key elements that make data mining tools a distinct form of software are :

Automated Analysis

Data mining automates the process of sifting through historical data in order to discover new information. This is one of the main differences between data mining and statistics, where a model is usually devised by a statistician to deal with a specific analysis problem. It also distinguishes data mining from expert systems, where the model is built by a knowledge engineer from rules extracted from the experience of an expert.

The emphasis on automated discovery also separates data mining from OLAP and simpler query and reporting tools, which are used to verify hypotheses formulated by the user. Data mining does not rely on a user to define a specific query, merely to formulate a goal - such as the identification of fraudulent claims.

Large or Complex Data Sets

One of the attractions of data mining is that it makes it possible to analyse very large data sets in a reasonable time scale. Data mining is also suitable for complex problems involving relatively small amounts of data but where there are many fields or variables to analyse. However, for small, relatively simple data analysis problems there may be simpler, cheaper and more effective solutions.

Discovering significant patterns or trends that would otherwise go unrecognized. The goal of data mining is to unearth relationships in data that may provide useful insights.

Data mining tools can sweep through databases and identify previously hidden patterns in one step. An example of pattern discovery is the analysis of retail sales data to identify seemingly unrelated products that are often purchased together. Other pattern discovery problems include detecting fraudulent credit card transactions, performance bottlenecks in a network system and identifying anomalous data that could represent data entry keying errors. The ultimate significance of these patterns will be assessed by a domain expert - a marketing manager or network supervisor - so the results must be presented in a way that human experts can understand.

Data mining tools can also automate the process of finding predictive information in large databases. Questions that traditionally required extensive hands-on analysis can now be answered directly from the data - quickly. A typical example of a predictive problem is targeted marketing. Data mining uses data on past promotional mailings to identify the targets most likely to maximize return on investment in future mailings. Other predictive problems include forecasting bankruptcy and other forms of default, and identifying segments of a population likely to respond similarly to given events.

Data mining techniques can yield the benefits of automation on existing software and hardware platforms to enhance the value of existing information resources, and can be implemented on new products and systems as they are brought on-line. When implemented on high performance client/server or parallel processing systems, they can analyse massive databases to deliver answers to questions such as :

“Which clients are most likely to respond to my next promotional mailing, and why?”

Data mining is ready for application because it is supported by three technologies that are now sufficiently mature :

1. Massive data collection
2. Powerful multiprocessor computers
3. Data mining algorithms

Commercial databases are growing at unprecedented rates, especially in the retail sector. The accompanying need for improved computational engines can now be met in a cost-effective manner with parallel multiprocessor computer technology. Data mining algorithms embody techniques that have existed for at least 10 years, but have only recently been implemented as mature, reliable, understandable tools that consistently outperform older statistical methods.

The core components of data mining technology have been under development for decades, in research areas such as statistics, artificial intelligence, and machine learning. Today, the maturity of these techniques, coupled with high-performance relational database engines and broad data integration efforts, make these technologies practical for current data warehouse environments.

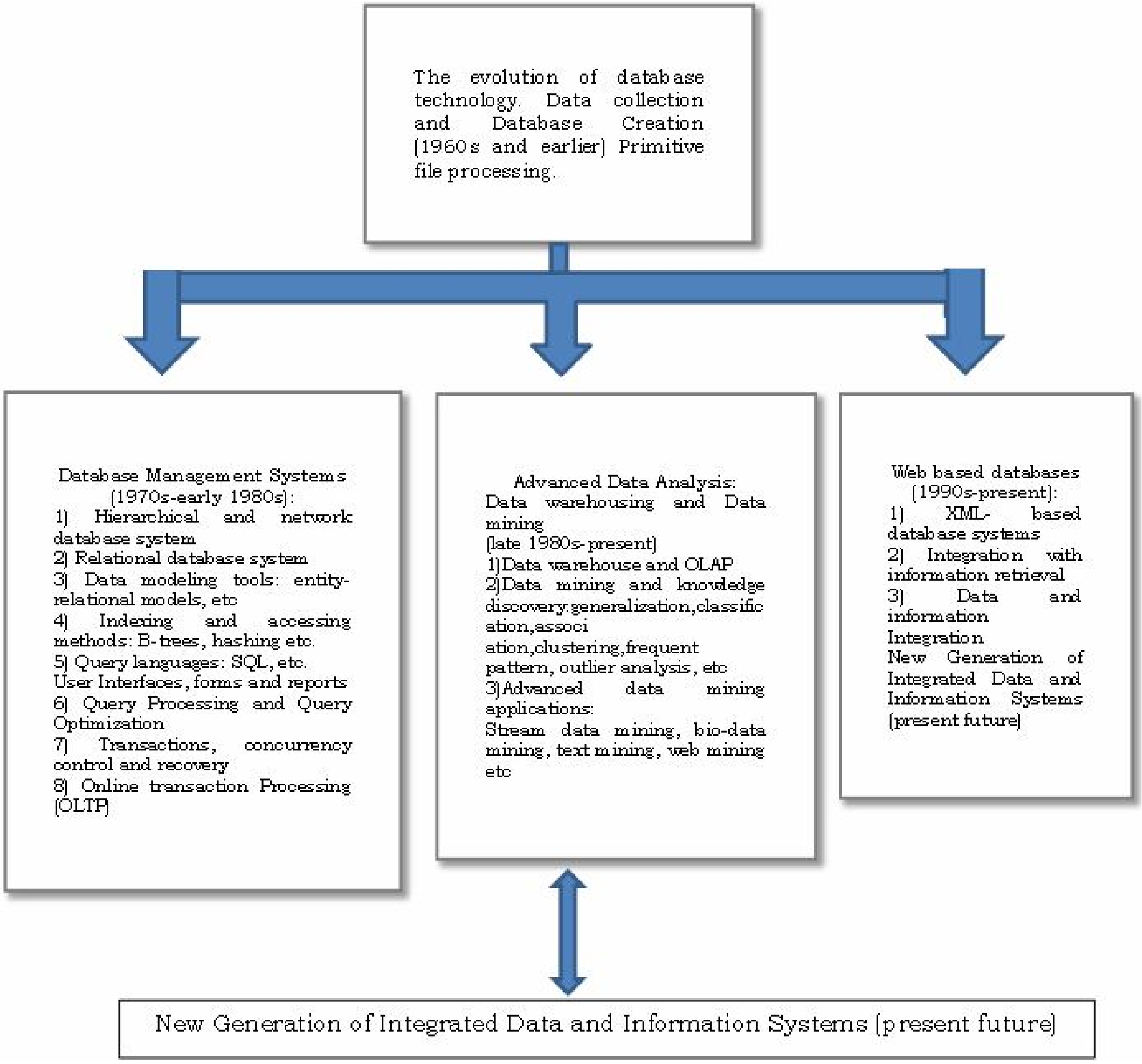
The key to understanding the different facets of data mining is to distinguish between data mining applications, operations, techniques and algorithms.

1.1.1 Motivation in Data Mining

Q2. What motivated data mining ? Why is it important ?

Ans :

The major reason that data mining has attracted a great deal of attention in information industry in recent years is due to the wide availability of huge amounts of data and the imminent need for turning such data into useful information and knowledge. The information and knowledge gained can be used for applications ranging from business management, production control, and market analysis, to engineering design and science exploration.


The evolution of database technology. Data collection and Database Creation (1960s and earlier) Primitive file processing.
Database Management Systems (1970s-early 1980s):
1) Hierarchical and network database system
2) Relational database system
3) Data modeling tools: entity-relational models, etc
4) Indexing and accessing methods: B-trees, hashing etc.
5) Query languages: SQL, etc.
User Interfaces, forms and reports
6) Query Processing and Query Optimization
7) Transactions, concurrency control and recovery
8) Online transaction Processing (OLTP)
Advanced Data Analysis:
Data warehousing and Data mining
(late 1980s-present)
1)Data warehouse and OLAP
2)Data mining and knowledge discovery.generalization,classific ation,associ ation,clustering,frequent pattern, outlier analysis, etc
3)Advanced data mining applications:
Stream data mining, bio-data mining, text mining, web mining etc
Web based databases (1990s-present):
1) XML- based database systems
2) Integration with information retrieval
3) Data and information Integration
New Generation of Integrated Data and Information Systems (present future)
New Generation of Integrated Data and Information Systems (present future)

1.2 Data Mining

Q3. What is data minings ?

Ans :

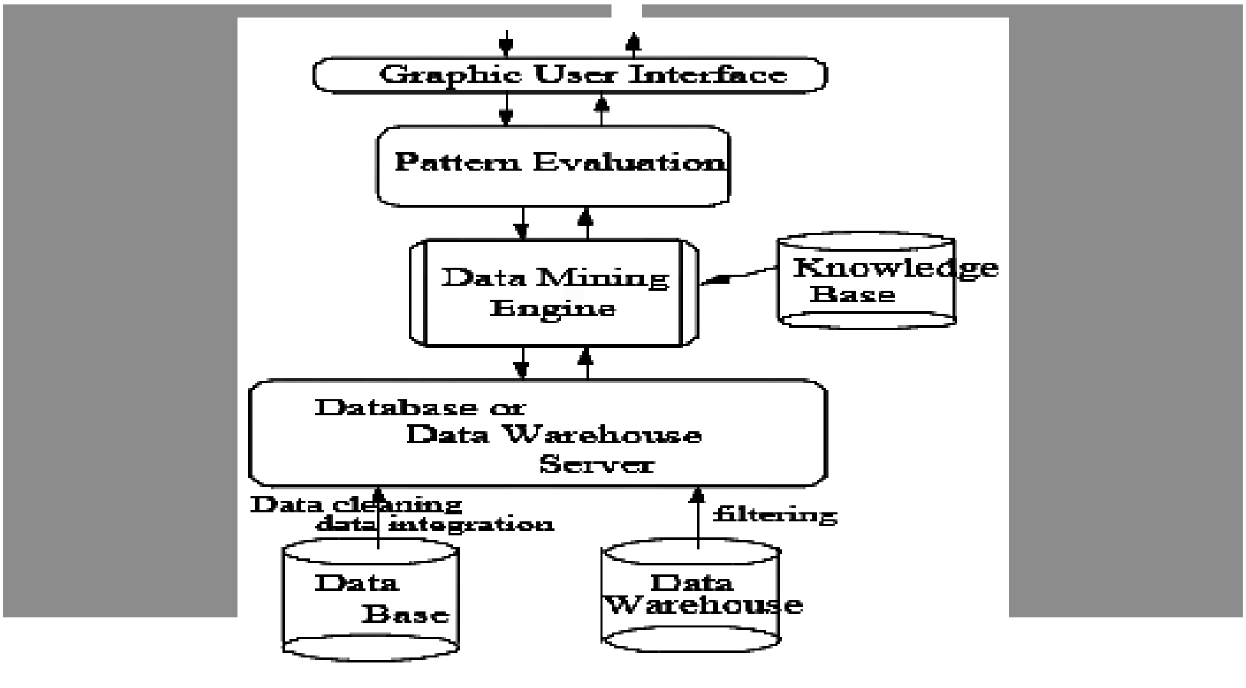
Data mining refers to extracting or mining" knowledge from large amounts of data. There are many other terms related to data mining, such as knowledge mining, knowledge extraction, data/pattern analysis, data archaeology, and data dredging. Many people treat data mining as a synonym for another popularly used term, Knowledge Discovery in Databases", or KDD.

Essential step in the process of knowledge discovery in databases

Knowledge discovery as a process is depicted in following figure and consists of an iterative sequence of the following steps :

- **Data Cleaning:** to remove noise or irrelevant data
- **Data Integration:** where multiple data sources may be combined
- **Data Selection:** where data relevant to the analysis task are retrieved from the database
- **Data Transformation:** where data are transformed or consolidated into forms appropriate for mining by performing summary or aggregation operations
- **Data Mining :** an essential process where intelligent methods are applied in order to extract data patterns
- Pattern Evaluation to identify the truly interesting patterns representing knowledge based on some interestingness measures
- **Knowledge Presentation:** where visualization and knowledge representation techniques are used to present the mined knowledge to the user.

Architecture of a Typical Data Mining System/Major Components



Architecture of a typical data mining system.

Data mining is the process of discovering interesting knowledge from large amounts of data stored either in databases, data warehouses, or other information repositories. Based on this view, the architecture of a typical data mining system may have the following major components :

1. A database, data warehouse, or other information repository, which consists of the set of databases, data warehouses, spreadsheets, or other kinds of information repositories containing the student and course information.
2. A database or data warehouse server which fetches the relevant data based on users' data mining requests.
3. A knowledge base that contains the domain knowledge used to guide the search or to evaluate the interestingness of resulting patterns. For example, the knowledge base may contain metadata which describes data from multiple heterogeneous sources.

4. A data mining engine, which consists of a set of functional modules for tasks such as classification, association, classification, cluster analysis, and evolution and deviation analysis.
5. A pattern evaluation module that works in tandem with the data mining modules by employing interestingness measures to help focus the search towards interestingness patterns.

A graphical user interface that allows the user an interactive approach to the data mining system.

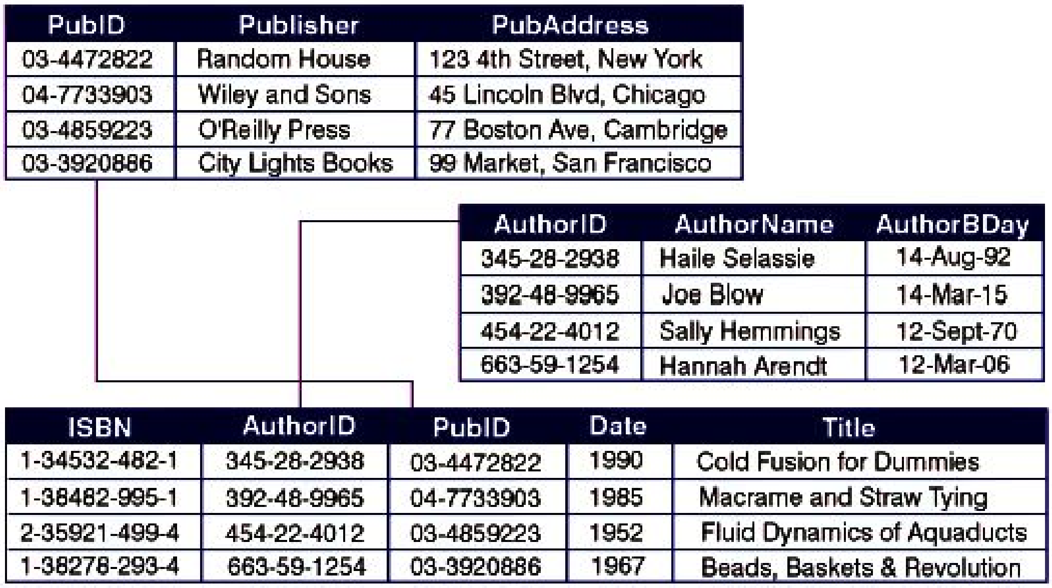
1.3 Types of data for Data Mining

Q4. On what kind of data is used in Data Mining? / Describe the following advanced database systems and applications: object-relational databases, spatial databases, text databases, multimedia databases, the World Wide Web ?

Ans :

In principle, data mining should be applicable to any kind of information repository. This includes relational databases, data warehouses, transactional databases, advanced database systems, flat files, and the World-Wide Web. Advanced database systems include object-oriented and object-relational databases, and special c application-oriented databases, such as spatial databases, time-series databases, text databases, and multimedia databases.

- **Flat files:** Flat files are actually the most common data source for data mining algorithms, especially at the research level. Flat files are simple data files in text or binary format with a structure known by the data mining algorithm to be applied. The data in these files can be transactions, time-series data, scientific measurements, etc.
- **Relational Databases:** a relational database consists of a set of tables containing either values of entity attributes, or values of attributes from entity relationships. Tables have columns and rows, where columns represent attributes and rows represent tuples. A tuple in a relational table corresponds to either an object or a relationship between objects and is identified by a set of attribute values representing a unique key. In following figure it presents some relations Customer, Items, and Borrow representing business activity in a video store. These relations are just a subset of what could be a database for the video store.



PubID	Publisher	PubAddress
03-4472822	Random House	123 4th Street, New York
04-7733903	Wiley and Sons	45 Lincoln Blvd, Chicago
03-4859223	O'Reilly Press	77 Boston Ave, Cambridge
03-3920886	City Lights Books	99 Market, San Francisco

AuthorID	AuthorName	AuthorBDay
345-28-2938	Haile Selassie	14-Aug-92
392-48-9965	Joe Blow	14-Mar-15
454-22-4012	Sally Hemmings	12-Sept-70
663-59-1254	Hannah Arendt	12-Mar-06

ISBN	AuthorID	PubID	Date	Title
1-34532-482-1	345-28-2938	03-4472822	1990	Cold Fusion for Dummies
1-38482-995-1	392-48-9965	04-7733903	1985	Macrame and Straw Tying
2-35921-499-4	454-22-4012	03-4859223	1952	Fluid Dynamics of Aquaducts
1-38278-293-4	663-59-1254	03-3920886	1967	Beads, Baskets & Revolution

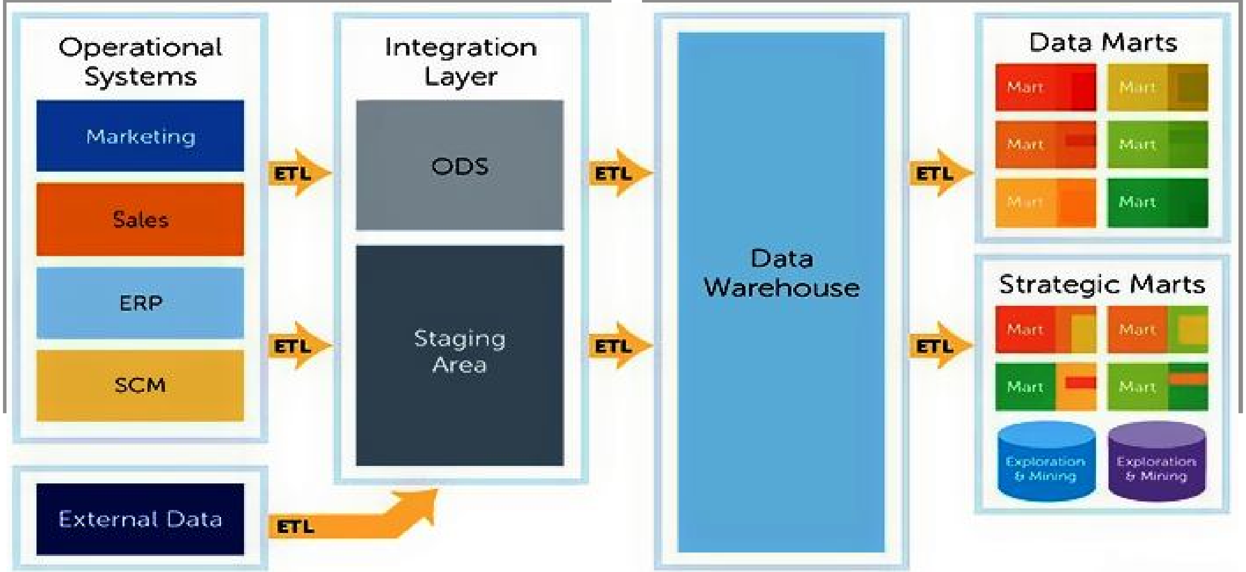
The most commonly used query language for relational database is SQL, which allows retrieval and manipulation of the data stored in the tables, as well as the calculation of aggregate functions such as average, sum, min, max and count. For instance, an SQL query to select the videos grouped by category would be:

SELECT count(*) FROM Items WHERE type=video GROUP BY category.

Data mining algorithms using relational databases can be more versatile than data mining algorithms specifically written for flat files, since they can take advantage of the structure inherent to relational databases. While data mining can benefit from SQL for data selection, transformation and consolidation, it goes beyond what SQL could provide, such as predicting, comparing, detecting deviations, etc.

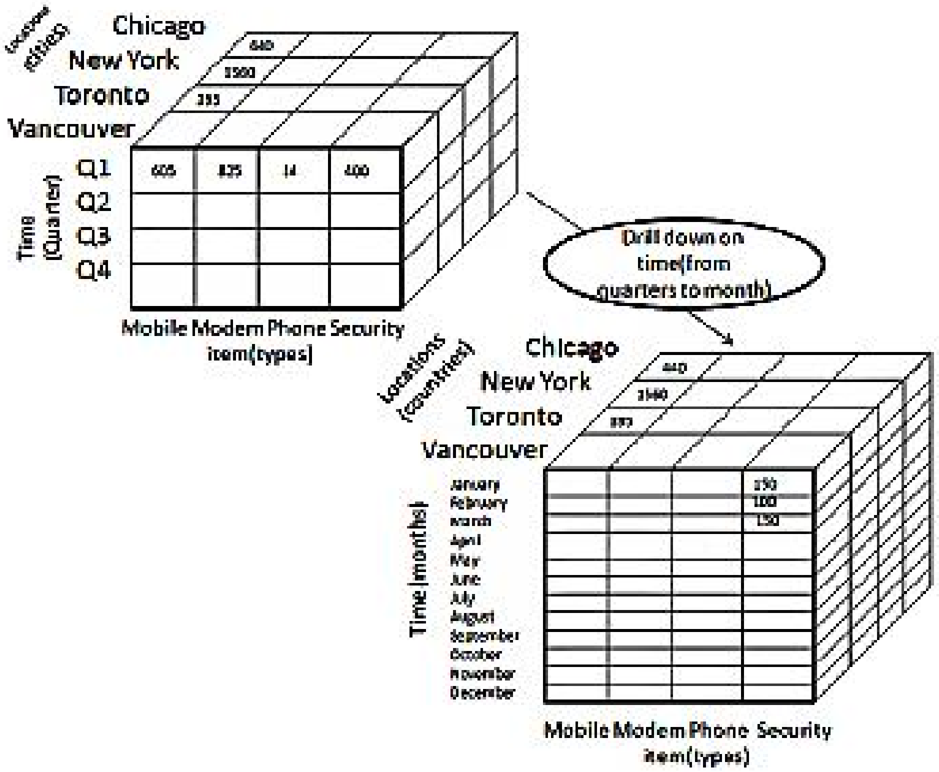
- **Data warehouses:** A data warehouse is a repository of information collected from multiple sources, stored under a unified schema, and which usually resides at a single site. Data warehouses are constructed via a process of data cleansing, data transformation, data integration, data loading, and periodic data refreshing. The figure shows the basic architecture of a data warehouse.

Data Warehouse



In order to facilitate decision making, the data in a data warehouse are organized around major subjects, such as customer, item, supplier, and activity. The data are stored to provide information from a historical perspective and are typically summarized.

A data warehouse is usually modeled by a multidimensional database structure, where each dimension corresponds to an attribute or a set of attributes in the schema, and each cell stores the value of some aggregate measure, such as count or sales amount. The actual physical structure of a data warehouse may be a relational data store or a multidimensional data cube. It provides a multidimensional view of data and allows the pre computation and fast accessing of summarized data.



The data cube structure that stores the primitive or lowest level of information is called a base cuboid. Its corresponding higher level multidimensional (cube) structures are called (non-base) cuboids. A base cuboid together with all of its corresponding higher level cuboids form a data cube. By providing multidimensional data views and the pre computation of summarized data, data warehouse systems are well suited for On-Line Analytical Processing, or OLAP. OLAP operations make use of background knowledge regarding the domain of the data being studied in order to allow the presentation of data at different levels of abstraction. Such operations accommodate different user viewpoints. Examples of OLAP operations include drill-down and roll-up, which allow the user to view the data at differing degrees of summarization, as illustrated in above figure.

- **Transactional databases:** In general, a transactional database consists of a flat file where each record represents a transaction. A transaction typically includes a unique transaction identity number (trans ID), and a list of the items making up the transaction (such as items purchased in a store) as shown below :

SALES Trans-ID	List of item_ID's
T100	I1,I3,I8 ....

- **Time-Series Databases**: Time-series databases contain time related data such stock market data or logged activities. These databases usually have a continuous flow of new data coming in, which

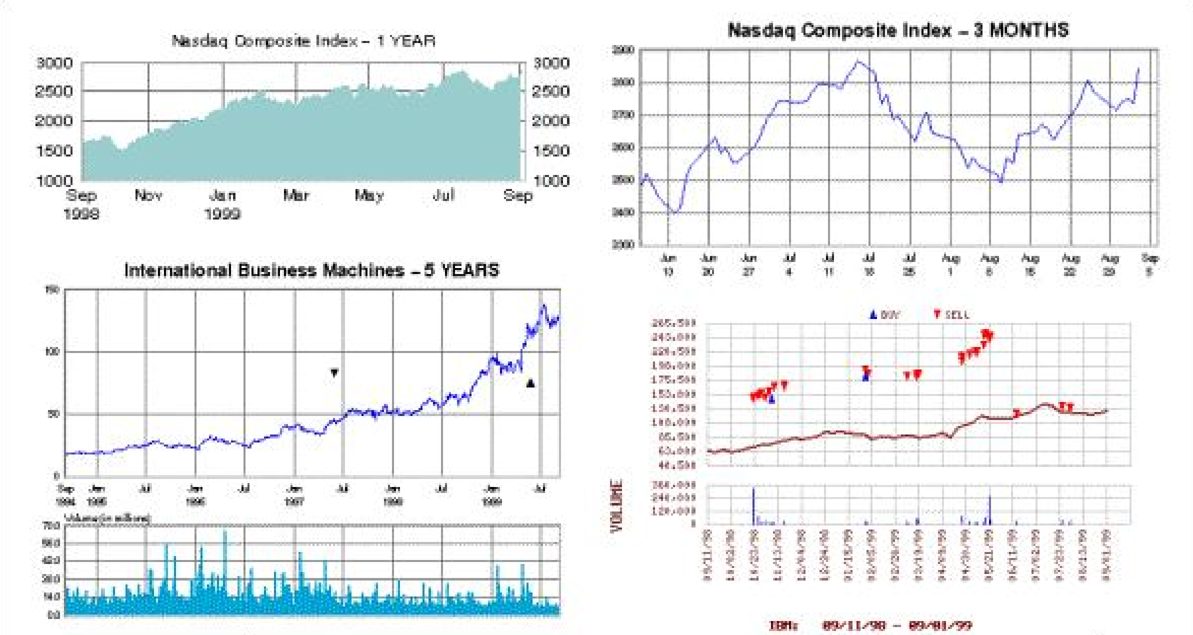


Figure 1.7 : Examples of Time-Series Data (Source: Thompson Investors Group)

- **World Wide Web**: The World Wide Web is the most heterogeneous and dynamic repository available. A very large number of authors and publishers are continuously contributing to its growth and metamorphosis, and a massive number of users are accessing its resources daily. Data in the World Wide Web is organized in inter-connected documents. These documents can be text, audio, video, raw data, and even applications. Conceptually, the World Wide Web is comprised of three major components: The content of the Web, which encompasses documents available; the structure of the Web, which covers the hyperlinks and the relationships between documents; and the usage of the web, describing how and when the resources are accessed. A fourth dimension can be added relating the dynamic nature or evolution of the documents. Data mining in the World Wide Web, or web mining, tries to address all these issues and is often divided into web content mining, web structure mining and web usage mining.

1.3.1 Applications Of Data Mining

Q5. What can be discovered ?

Ans :

The kinds of patterns that can be discovered depend upon the data mining tasks employed. By and large, there are two types of data mining tasks: *descriptive data mining* tasks that describe the general properties of the existing data, and *predictive data mining* tasks that attempt to do predictions based on inference on available data. The data mining functionalities and the variety of knowledge they discover are briefly presented in the following list :

- **Characterization :** Data characterization is a summarization of general features of objects in a target class, and produces what is called *characteristic rules*. The data relevant to a user-specified class are normally retrieved by a database query and run through a summarization module to extract the essence of the data at different levels of abstractions. For example, one may want to characterize the OurVideoStore customers who regularly rent more than 30 movies a year. With concept hierarchies on the attributes describing the target class, the *attribute-oriented induction* method can be used, for example, to carry out data summarization. Note that with a data cube containing summarization of data, simple OLAP operations fit the purpose of data characterization.

- **Discrimination :** Data discrimination produces what are called *discriminant rules* and is basically the comparison of the general features of objects between two classes referred to as the *target class* and the *contrasting class*. For example, one may want to compare the general characteristics of the customers who rented more than 30 movies in the last year with those whose rental account is lower than 5. The techniques used for data discrimination are very similar to the techniques used for data characterization with the exception that data discrimination results include comparative measures.
- **Association analysis**: Association analysis is the discovery of what are commonly called *association rules*. It studies the frequency of items occurring together in transactional databases, and based on a threshold called *support*, identifies the frequent item sets. Another threshold, *confidence*, which is the conditional probability than an item appears in a transaction when another item appears, is used to pinpoint association rules. Association analysis is commonly used for market basket analysis. For example, it could be useful for the OurVideoStore manager to know what movies are often rented together or if there is a relationship between renting a certain type of movies and buying popcorn or pop. The discovered association rules are of the form: P -> Q [s,c], where P and Q are conjunctions of attribute value-pairs, and s (for support) is the probability that P and Q appear together in a transaction and c (for confidence) is the conditional probability that Q appears in a transaction when P is present.
- **Classification**: Classification analysis is the organization of data in given classes. Also known as *supervised classification*, the classification uses given class labels to order the objects in the data collection. Classification approaches normally use a *training set* where all objects are already associated with known class labels. The classification algorithm learns from the training set and builds a model. The model is used to classify new objects. For example, after starting a credit policy, the OurVideoStore managers could analyze the customers behaviours vis-à-vis their credit, and label accordingly the customers who received credits with three possible labels "safe", "risky" and "very risky". The classification analysis would generate a model that could be used to either accept or reject credit requests in the future.
- **Prediction**: Prediction has attracted considerable attention given the potential implications of successful forecasting in a business context. There are two major types of predictions: one can either try to predict some unavailable data values or pending trends, or predict a class label for some data. The latter is tied to classification. Once a classification model is built based on a training set, the class label of an object can be foreseen based on the attribute values of the object and the attribute values of the classes. Prediction is however more often referred to the forecast of missing numerical values, or increase/ decrease trends in time related data. The major idea is to use a large number of past values to consider probable future values.
- **Clustering**: Similar to classification, clustering is the organization of data in classes. However, unlike classification, in clustering, class labels are unknown and it is up to the clustering algorithm to discover acceptable classes. Clustering is also called *unsupervised classification*, because the classification is not dictated by given class labels. There are many clustering approaches all based on the principle of maximizing the similarity between objects in a same class (*intra-class similarity*) and minimizing the similarity between objects of different classes (*inter-class similarity*).
- **Outlier analysis**: Outliers are data elements that cannot be grouped in a given class or cluster. Also known as *exceptions* or *surprises*, they are often very important to identify. While outliers can be considered noise and discarded

in some applications, they can reveal important knowledge in other domains, and thus can be very significant and their analysis valuable.

- **Evolution and deviation analysis**: Evolution and deviation analysis pertain to the study of time related data that changes in time. Evolution analysis models evolutionary trends in data, which consent to characterizing, comparing, classifying or clustering of time related data. Deviation analysis, on the other hand, considers differences between measured values and expected values, and attempts to find the cause of the deviations from the anticipated values.

- It is common that users do not have a clear idea of the kind of patterns they can discover or need to discover from the data at hand. It is therefore important to have a versatile and inclusive data mining system that allows the discovery of different kinds of knowledge and at different levels of abstraction. This also makes interactivity an important attribute of a data mining system.

1.3.2 Advanced Database Systems and Advanced Database Applications

Q6. Explain about advanced Data Base Systems and its Applications and explain the functionalities of Data Mining in detail?

Ans :

- **An objected-oriented database** is designed based on the object-oriented programming paradigm where data are a large number of objects organized into classes and class hierarchies. Each entity in the database is considered as an object. The object contains a set of variables that describe the object, a set of messages that the object can use to communicate with other objects or with the rest of the database system and a set of methods where each method holds the code to implement a message.

- **A spatial database** contains spatial-related data, which may be represented in the form of raster or vector data. Raster data consists of n-dimensional bit maps or pixel maps, and vector data are represented by lines, points, polygons or other kinds of processed primitives, Some examples of spatial databases include geographical (map) databases, VLSI chip designs, and medical and satellite images databases.

- **Time-Series Databases:** Time-series databases contain time related data such stock market data or logged activities. These databases usually have a continuous flow of new data coming in, which sometimes causes the need for a challenging real time analysis. Data mining in such databases commonly includes the study of trends and correlations between evolutions of different variables, as well as the prediction of trends and movements of the variables in time.

- **A text database** is a database that contains text documents or other word descriptions in the form of long sentences or paragraphs, such as product specifications, error or bug reports, warning messages, summary reports, notes, or other documents.

- **A multimedia database** stores images, audio, and video data, and is used in applications such as picture content-based retrieval, voice-mail systems, video-on-demand systems, the World Wide Web, and speech-based user interfaces.

- **The World-Wide Web** provides rich, world-wide, on-line information services, where data objects are linked together to facilitate interactive access. Some examples of distributed information services associated with the World-Wide Web include America Online, Yahoo!, AltaVista, and Prodigy.

Data Mining involves Six Common Classes of Tasks

- **Anomaly detection (Outlier/change/ deviation detection)** – The identification of unusual data records, that might be interesting or data errors that require further investigation.

- **Association rule learning (Dependency modeling)** – Searches for relationships between variables. For example a supermarket might gather data on customer purchasing habits. Using association rule learning, the supermarket can determine which products are frequently bought together and use this information for marketing purposes. This is sometimes referred to as market basket analysis.

- **Clustering** – is the task of discovering groups and structures in the data that are in some way or another "similar", without using known structures in the data.
- **Classification** – is the task of generalizing known structure to apply to new data. For example, an e-mail program might attempt to classify an e-mail as "legitimate" or as "spam".
- **Regression** – attempts to find a function which models the data with the least error.
- **Summarization** – providing a more compact representation of the data set, including visualization and report generation.

Classification of Data mining Systems :

The data mining system can be classified according to the following criteria :

- Database Technology
- Statistics
- Machine Learning
- Information Science
- Visualization
- Other Disciplines

Some Other Classification Criteria :

- Classification according to kind of databases mined
- Classification according to kind of knowledge mined
- Classification according to kinds of techniques utilized
- Classification according to applications adapted

Classification According to Kind of Databases Mined

We can classify the data mining system according to kind of databases mined. Database system can be classified according to different criteria such as data models, types of data etc. And the data mining system can be classified accordingly. For example if we classify the database according to data model then we may have a relational, transactional, object- relational, or data warehouse mining system.

Classification According to Kind of Knowledge Mined

We can classify the data mining system according to kind of knowledge mined. It is means data mining system are classified on the basis of functionalities such as :

1. Characterization
2. Discrimination
3. Association and Correlation Analysis
4. Classification
5. Prediction
6. Clustering
7. Outlier Analysis
8. Evolution Analysis

Classification According to kinds of Techniques Utilized

We can classify the data mining system according to kind of techniques used. We can describes these techniques according to degree of user interaction involved or the methods of analysis employed.

Classification According to Applications Adapted

We can classify the data mining system according to application adapted. These applications are as follows :

1. Finance
2. Telecommunications
3. DNA
4. Stock Markets
5. E-mail

1.4 Data Mining Techniques

Q7. Discuss in detail different types of Data Mining techniques ?

Ans :

One of the most important tasks in Data Mining is to select the correct data mining technique. Data Mining technique has to be chosen based on the type of business and the type of problem your business faces. A generalized approach has to be used to improve the accuracy and cost effectiveness of using data mining techniques. There are basically seven main Data Mining techniques which is discussed in this article. There are also a lot of other Data Mining techniques but these seven are considered more frequently used by business people.

- Statistics
- Clustering
- Visualization
- Decision Tree
- Association Rules
- Neural Networks
- Classification

1. Statistical Techniques

Data mining techniques statistics is a branch of mathematics which relates to the collection and description of data. Statistical technique is not considered as a data mining technique by many analysts. But still it helps to discover the patterns and build predictive models. For this reason data analyst should possess some knowledge about the different statistical techniques. In today's world people have to deal with large amount of data and derive important patterns from it. Statistics can help you to a greater extent to get answers for questions about their data like

- What are the patterns in their database?
- What is the probability of an event to occur?
- Which patterns are more useful to the business?
- What is the high level summary that can give you a detailed view of what is there in the database?

Statistics not only answers these questions they help in summarizing the data and count it. It also helps in providing information about the data with ease. Through statistical reports people can take smart decisions. There are different forms of statistics but the most important and useful technique is the collection and counting of data. There are a lot of ways to collect data like

- Histogram
- Mean
- Median
- Mode
- Variance
- Max
- Min
- Linear Regression

2. Clustering Technique

Clustering is one among the oldest techniques used in Data Mining. Clustering analysis is the process of identifying data that are similar to each other. This will help to understand the differences and similarities between the data. This is sometimes called segmentation and helps the users to understand what is going on within the database. For example, an insurance company can group its customers based on their income, age, nature of policy and type of claims.

There are different types of clustering methods. They are as follows :

- Partitioning Methods
- Hierarchical Agglomerative methods
- Density Based Methods
- Grid Based Methods
- Model Based Methods

The most popular clustering algorithm is Nearest Neighbor. Nearest neighbor technique is very similar to clustering. It is a prediction technique where in order to predict what a estimated value is in one record look for records with similar estimated values in historical database and use the prediction value from the record which is near to the unclassified record. This technique simply states that the objects which are closer to each other will have similar prediction values. Through this method you can easily predict the values of nearest objects very easily. Nearest Neighbor is the most easy to use technique because they work as per the thought of the people. They also work very well in terms of automation. They perform complex ROI calculations with ease. The level of accuracy in this technique is as good as the other Data Mining techniques.

In business Nearest Neighbor technique is most often used in the process of Text Retrieval. They are used to find the documents that share the important characteristics with that main document that have been marked as interesting.

3. Visualization

Visualization is the most useful technique which is used to discover data patterns. This technique is used at the beginning of the Data Mining process. Many researches are going on these days to produce interesting projection of databases, which is called Projection Pursuit. There are a lot of data mining technique which will produce useful patterns for good data. But visualization is a technique which converts Poor data into good data letting different kinds of Data Mining methods to be used in discovering hidden patterns.

4. Induction Decision Tree Technique

A decision tree is a predictive model and the name itself implies that it looks like a tree. In this technique, each branch of the tree is viewed as a classification question and the leaves of the trees are considered as partitions of the dataset related to that particular classification. This technique can be used for exploration analysis, data pre-processing and prediction work.

Decision tree can be considered as a segmentation of the original dataset where segmentation is done for a particular reason. Each data that comes under a segment has some similarities in their information being predicted. Decision trees provides results that can be easily understood by the user.

Decision tree technique is mostly used by statisticians to find out which database is more related to the problem of the business. Decision tree technique can be used for Prediction and Data pre-processing.

The first and foremost step in this technique is growing the tree. The basic of growing the tree depends on finding the best possible question to be asked at each branch of the tree. The decision tree stops growing under any one of the below circumstances

- If the segment contains only one record
- All the records contain identical features
- The growth is not enough to make any further spilt

CART which stands for Classification and Regression Trees is a data exploration and prediction algorithm which picks the questions in a more complex way. It tries them all and then selects one best question which is used to split the data into two or more segments. After deciding on the segments it again asks questions on each of the new segment individually.

Another popular decision tree technology is CHAID (Chi-Square Automatic Interaction Detector). It is similar to CART but it differs in one way. CART helps in choosing the best questions whereas CHAID helps in choosing the splits.

5. Neural Network

Neural Network is another important technique used by people these days. This technique is most often used in the starting stages of the data mining technology. Artificial neural network was formed out of the community of Artificial intelligence.

Neural networks are very easy to use as they are automated to a particular extent and because of this the user is not expected to have much knowledge about the work or database. But to make the neural network work efficiently you need to know

- How the nodes are connected?
- How many processing units to be used?
- When should the training process to be stopped?

There are two main parts of this technique – the node and the link

- **The node** – which freely matches to the neuron in the human brain
- **The link** – which freely matches to the connections between the neurons in the human brain
- A neural network is a collection of interconnected neurons. which could form a single layer or multiple layer. The formation of neurons and their interconnections are called architecture of the network. There are a wide variety of neural network models and each model has its own advantages and disadvantages.

Every neural network model has different architectures and these architectures use different learning procedures.

- Neural networks are very strong predictive modeling technique. But it is not very easy to understand even by experts. It creates very complex models which is impossible to understand fully. Thus to understand the Neural network technique companies are finding out new solutions. Two solutions have already been suggested
- First solution is Neural network is packaged up into a complete solution which will let it to be used for a single application
- Second solution is it is bonded with expert consulting services.
- Neural network has been used in various kinds of applications. This has been used in the business to detect frauds taking place in the business.

6. Association Rule Technique

This technique helps to find the association between two or more items. It helps to know the relations between the different variables in databases. It discovers the hidden patterns in the data sets which is used to identify the variables and the frequent occurrence of different variables that appear with the highest frequencies.

Association rule offers two major information

- **Support** – Hoe often is the rule applied ?
- **Confidence** – How often the rule is correct ?

This technique follows a two step process

- Find all the frequently occurring data sets
- Create strong association rules from the frequent data sets

There are three types of association rule. They are

- Multilevel Association Rule
- Multidimensional Association Rule
- Quantitative Association Rule

This technique is most often used in retail industry to find patterns in sales. This will help increase the conversion rate and thus increases profit.

7. Classification

Data mining techniques classification is the most commonly used data mining technique which contains a set of pre classified samples to create a model which can classify the large set of data. This technique helps in deriving important information about data and metadata (data about data). This technique is closely related to cluster analysis technique and it uses decision tree or neural network system. There are two main processes involved in this technique

- **Learning** – In this process the data are analyzed by classification algorithm
- **Classification** – In this process the data is used to measure the precision of the classification rules

There are different types of classification models. They are as follows

- Classification by decision tree induction
- Bayesian Classification
- Neural Networks
- Support Vector Machines (SVM)
- Classification Based on Associations.

1.5 Data Mining Issues

Q8. What are the issues in Data Mining ?

Ans :

Data mining algorithms embody techniques that have sometimes existed for many years, but have only lately been applied as reliable and scalable tools that time and again outperform older classical statistical methods. While data mining is still in its infancy, it is becoming a trend and ubiquitous. Before data mining develops into a conventional, mature and trusted discipline, many still pending issues have to be addressed. Some of these issues are addressed below. Note that these issues are not exclusive and are not ordered in any way.

- **Security and social issues**: Security is an important issue with any data collection that is shared and/or is intended to be used for strategic decision-making. In addition, when data is collected for customer profiling, user behaviour understanding, correlating personal data with other information, etc., large amounts of sensitive and private information about individuals or companies is gathered and stored. This becomes controversial given the confidential nature of some of this data and the potential illegal access to the information. Moreover, data mining could disclose new implicit knowledge about individuals or groups that could be against privacy policies, especially if there is potential dissemination of discovered information. Another issue that arises from this concern is the appropriate use of data mining. Due to the value of data, databases of all sorts of content are regularly sold, and because of the competitive advantage that can be attained from implicit knowledge discovered, some important information could be withheld, while other information could be widely distributed and used without control.

- **User interface issues**: The knowledge discovered by data mining tools is useful as long as it is interesting, and above all understandable by the user. Good data visualization eases the interpretation of data mining results, as well as helps users better understand their needs. Many data exploratory analysis tasks are significantly facilitated by the ability to see data in an appropriate visual presentation. There are many visualization ideas and proposals for effective data graphical presentation. However, there is still much research to accomplish in order to obtain good visualization tools for large datasets that could be used to display and manipulate mined knowledge. The major issues related to user interfaces and visualization are "screen real-estate", information rendering, and interaction. Interactivity with the data and data mining results is crucial since it provides means for the user to focus and refine the mining tasks, as well as to picture the discovered knowledge from different angles and at different conceptual levels.

- **Mining methodology issues**: These issues pertain to the data mining approaches applied and their limitations. Topics such as versatility of the mining approaches, the diversity of data available, the dimensionality of the domain, the broad analysis needs (when known), the assessment of the knowledge discovered, the exploitation of background knowledge and metadata, the control and handling of noise in data, etc. are all examples that can dictate mining methodology choices. For instance, it is often desirable to have different data mining methods available since different approaches may perform differently depending upon the data at hand. Moreover, different approaches may suit and solve user's needs differently.

  Most algorithms assume the data to be noise-free. This is of course a strong assumption. Most datasets contain exceptions, invalid or incomplete information, etc., which may complicate, if not obscure, the analysis process and in many cases compromise the accuracy of the results. As a consequence, data preprocessing (data cleaning and transformation) becomes vital. It is often seen as lost time, but data cleaning, as time-consuming and frustrating as it may be, is one of the most important phases in the knowledge discovery process. Data mining techniques should be able to handle noise in data or incomplete information.

  More than the size of data, the size of the search space is even more decisive for data mining techniques. The size of the search space is often depending upon the number of dimensions in the domain space. The search space usually grows exponentially when the number of dimensions increases. This is known as the *curse of dimensionality*. This "curse" affects so badly the performance of some data mining approaches that it is becoming one of the most urgent issues to solve.

- **Performance issues**: Many artificial intelligence and statistical methods exist for data analysis and interpretation. However, these methods were often not designed for the very large data sets data mining is dealing with today. Terabyte sizes are common. This raises the issues of scalability and efficiency of the data mining methods when processing considerably large data. Algorithms with exponential and even medium-order polynomial complexity cannot be of practical use for data mining. Linear algorithms are usually the norm. In same theme, sampling can be used for mining instead of the whole dataset. However, concerns such as completeness and choice of samples may arise. Other topics in the issue of performance are *incremental updating*, and parallel programming. There is no doubt that parallelism can help solve the size problem if the dataset can be subdivided and the results can be merged later. Incremental updating is important for merging results from parallel mining, or updating data mining results when new data becomes available without having to re-analyze the complete dataset.

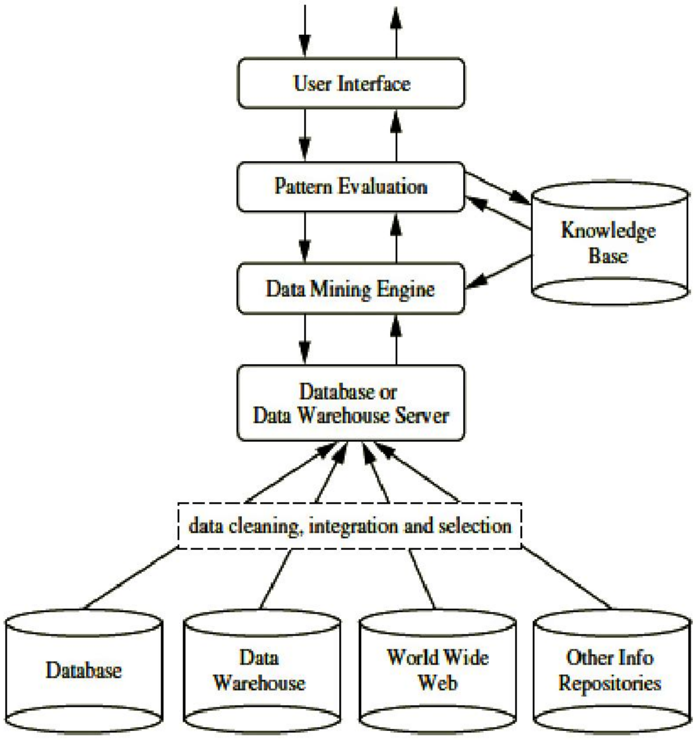
- **Data source issues**: There are many issues related to the data sources, some are practical such as the diversity of data types, while others are philosophical like the data glut problem. We certainly have an excess of data since we already have more data than we can handle and we are still collecting data at an even higher rate. If the spread of database management systems has helped increase the gathering of information, the advent of data mining is certainly encouraging more data harvesting. The current practice is to collect as much data as possible now and process it, or try to process it, later. The concern is whether we are collecting the right data at the appropriate amount, whether we know what we want to do with it, and whether we distinguish between what data is important and what data is insignificant. Regarding the practical issues related to data sources, there is the subject of heterogeneous databases and the focus on diverse complex data types. We are storing different types of data in a variety of repositories. It is difficult to expect a data mining system to effectively and efficiently achieve good mining results on all kinds of data and sources. Different kinds of data and sources may require distinct algorithms and methodologies. Currently, there is a focus on relational databases and data warehouses, but other approaches need to be pioneered for other specific complex data types. A versatile data mining tool, for all sorts of data, may not be realistic. Moreover, the proliferation of heterogeneous data sources, at structural and semantic levels, poses important challenges not only to the database community but also to the data mining community.

1.5.1 Architecture of Data Mining

Q9. Discuss in detail DATA MINIG COMPONENTS & ARCHITECTURE.

Ans :

A typical data mining system may have the following major components.



- **Knowledge Base:** This is the domain knowledge that is used to guide the search or evaluate the interestingness of resulting patterns. Such knowledge can include concept hierarchies, used to organize attributes or attribute values into different levels of abstraction. Knowledge such as user beliefs, which can be used to assess a pattern's interestingness based on its unexpectedness, may also be included. Other examples of domain knowledge are additional interestingness constraints or thresholds, and metadata (e.g., describing data from multiple heterogeneous sources).
- **Data Mining Engine:** This is essential to the data mining system and ideally consists of a set of functional modules for tasks such as characterization, association and correlation analysis, classification, prediction, cluster analysis, outlier analysis, and evolution analysis.
- **Pattern Evaluation Module:** This component typically employs interestingness measures interact with the data mining modules so as to focus the search toward interesting patterns. It may use interestingness thresholds to filter out discovered patterns. Alternatively, the pattern evaluation module may be integrated with the mining module, depending on the implementation of the data mining method used. For efficient data mining, it is highly recommended to push the evaluation of pattern interestingness as deep as possible into the mining process so as to confine the search to only the interesting patterns.
- **User interface:** This module communicates between users and the data mining system, allowing the user to interact with the system by specifying a data mining query or task, providing information to help focus the search, and performing exploratory data mining based on the intermediate data mining results. In addition, this component allows the user to browse database and data warehouse schemas or data structures, evaluate mined patterns, and visualize the patterns in different forms.

Five primitives for specifying a data mining task

- **Task-relevant data:** This primitive specifies the data upon which mining is to be performed. It involves specifying the database and tables or data warehouse containing the relevant data, conditions for selecting the relevant data, the relevant attributes or dimensions for exploration, and instructions regarding the ordering or grouping of the data retrieved.
- **Knowledge type to be mined:** This primitive specifies the specific data mining function to be performed, such as characterization, discrimination, association, classification, clustering, or evolution analysis. As well, the user can be more specific and provide pattern templates that all discovered patterns must match. These templates or meta patterns (also called meta rules or meta queries), can be used to guide the discovery process.
- **Background knowledge:** This primitive allows users to specify knowledge they have about the domain to be mined. Such knowledge can be used to guide the knowledge discovery process and evaluate the patterns that are found. Of the several kinds of background knowledge, this chapter focuses on concept hierarchies.
- **Pattern interestingness measure:** This primitive allows users to specify functions that are used to separate uninteresting patterns from knowledge and may be used to guide the mining process, as well as to evaluate the discovered patterns. This allows the user to confine the number of uninteresting patterns returned by the process, as a data mining process may generate a large number of patterns. Interestingness measures can be specified for such pattern characteristics as simplicity, certainty, utility and novelty.
- **Visualization of discovered patterns:** This primitive refers to the form in which discovered patterns are to be displayed. In order for data mining to be effective in conveying knowledge to users, data mining systems should be able to display the discovered patterns in multiple forms such as rules, tables, cross tabs (cross-tabulations), pie or bar charts, decision trees, cubes or other visual representations.

Integration of a Data Mining System with a Database or Data Warehouse System

The differences between the following architectures for the integration of a data mining system with a database or data warehouse system are as follows.

- **No coupling:** The data mining system uses sources such as flat files to obtain the initial data set to be mined since no database system or data warehouse system functions are implemented as part of the process. Thus, this architecture represents a poor design choice.
- **Loose coupling:** The data mining system is not integrated with the database or data warehouse system beyond their use as the source of the initial data set to be mined, and possible use in storage of the results. Thus, this architecture can take advantage of the flexibility, efficiency and features such as indexing that the database and data warehousing systems may provide. However, it is difficult for loose coupling to achieve high scalability and good performance with large data sets as many such systems are memory-based.
- **Semi tight coupling:** Some of the data mining primitives such as aggregation, sorting or pre computation of statistical functions are efficiently implemented in the database or data warehouse system, for use by the data mining system during mining-query processing. Also, some frequently used inter mediate mining results can be pre computed and stored in the database or data warehouse system, thereby enhancing the performance of the data mining system.
- **Tight coupling:** The database or data warehouse system is fully integrated as part of the data mining system and thereby provides optimized data mining query processing. Thus, the data mining sub system is treated as one functional component of an information system. This is a highly desirable architecture as it facilitates efficient implementations of data mining functions, high system performance, and an integrated information processing environment.

From the descriptions of the architectures provided above, it can be seen that tight coupling is the best alternative without respect to technical or implementation issues. However, as much of the technical infrastructure needed in a tightly coupled system is still evolving, implementation of such a system is non-trivial. Therefore, the most popular architecture is currently semi tight coupling as it provides a compromise between loose and tight coupling.

1.6 Getting to Know Your data

Introduction

It's tempting to jump straight into mining, but first, we need to get the data ready. This involves having a closer look at attributes and data values. Real-world data are typically noisy, enormous in volume (often several gigabytes or more), and may originate from a hodgepodge of heterogeneous sources.

We have to know the following :

- What are the types of *attributes* or fields that make up your data?
- What kind of values does each attribute have?
- Which attributes are discrete, and which are continuous-valued?
- What do the data *look like*? How are the values distributed?
- Are there ways we can visualize the data to get a better sense of it all?
- Can we spot any outliers?
- Can we measure the similarity of some data objects with respect to others?
- **"So what can we learn about our data that's helpful in data preprocessing?"**

These include nominal attributes, binary attributes, ordinal attributes, and numeric attributes. Basic *statistical descriptions* can be used to learn more about each attribute's values, we can determine its **mean** (average value), **median** (middle value), and **mode** (most common value). These are **measures of central tendency**, which give us an idea of the "middle" or center of distribution.

Knowing such basic statistics regarding each attribute makes it easier to fill in missing values, smooth noisy values, and spot outliers during data preprocessing. Knowledge of the attributes and attribute values can also help in fixing inconsistencies incurred during data integration. Plotting the measures of central tendency shows us if the data are symmetric or skewed. Quantile plots, histograms, and scatter plots are other graphic displays of basic statistical descriptions. These can all be useful during data preprocessing and can provide insight into areas for mining.

The field of data visualization provides many additional techniques for viewing data through graphical means. These can help identify relations, trends, and biases "hidden" in unstructured data sets. Techniques may be as simple as scatter-plot matrices.

We may want to find the similarity or dissimilarity between individual patients. Such information can allow us to find clusters of like patients within the data set. The similarity/ dissimilarity between objects may also be used to detect outliers in the data, or to perform nearest-neighbor classification.

Data Objects and Attribute Types

Data sets are made up of data objects. A **data object** represents an entity in a sales database, the objects may be customers, store items, and sales; in a medical database, the objects may be patients; in a university database, the objects may be students, professors, and courses. Data objects are typically described by attributes. Data objects can also be referred to as *samples, examples, instances, data points*, or *objects*. If the data objects are stored in a database, they are *data tuples*. That is, the rows of a database correspond to the data objects, and the columns correspond to the attributes. In this section, we define attributes and look at the various attribute types.

Q10. What is an Attribute ? and What are the different types of a attributes ?

Ans :

An **attribute** is a data field, representing a characteristic or feature of a data object. The nouns *attribute*, *dimension*, *feature*, and *variable* are often used interchangeably in the literature. The term *dimension* is commonly used in data warehousing. Machine learning literature tends to use the term *feature*, while statisticians prefer the term *variable*. Data mining and database professionals commonly use the term *attribute*, and we do here as well. Attributes describing a customer object can include, for example, *customer ID*, *name*, and *address*. Observed values for a given attribute are known as *observations*. A set of attributes used to describe a given object is called an *attribute vector* (or *feature vector*).

The distribution of data involving one attribute (or variable) is called *uni-variate*. A *bivariate* distribution involves two attributes, and so on. The **type** of an attribute is determined by the set of possible values—nominal, binary, ordinal, or numeric—the attribute can have. In the following subsections, we introduce each type.

- **Nominal Attributes**

Nominal means "relating to names." The values of a **nominal attribute** are symbols or *names of things*. Each value represents some kind of category, code, or state, and so nominal attributes are also referred to as **categorical**. The values do not have any meaningful order. In computer science, the values are also known as *enumerations*.

Although we said that the values of a nominal attribute are symbols or "names of things," it is possible to represent such symbols or "names" with numbers. Even though a nominal attribute may have integers as values, it is not considered a numeric attribute because the integers are not meant to be used quantitatively.

Nominal attribute values do not have any meaningful order about them and are not quantitative; it makes no sense to find the mean (average) value or median (middle) value for such an attribute, given a set of objects. One thing that is of interest, however, is the attribute's most commonly occurring value. This value, known as the *mode*, is one of the measures of central tendency.

- **Binary Attributes**

A **binary attribute** is a nominal attribute with only two categories or states: 0 or 1, where 0 typically means that the attribute is absent, and 1 means that it is present. Binary attributes are referred to as **Boolean** if the two states correspond to *true* and *false*.

A binary attribute is **symmetric** if both of its states are equally valuable and carry the same weight; that is, there is no preference on which outcome should be coded as 0 or 1. One such example could be the attribute *gender* having the states *male* and *female*.

A binary attribute is **asymmetric** if the outcomes of the states are not equally important, such as the *positive* and *negative* outcomes of a medical test for HIV. By convention, we code the most important outcome, which is usually the rarest one, by 1 (e.g., *HIV positive*) and the other by 0 (e.g., *HIV negative*).

Ordinal Attributes

An **ordinal attribute** is an attribute with possible values that have a meaningful order or *ranking* among them, but the magnitude between successive values is not known.

Ordinal attributes are useful for registering subjective assessments of qualities that cannot be measured objectively; thus ordinal attributes are often used in surveys for ratings. In one survey, participants were asked to rate how satisfied they were as customers.

Customer satisfaction had the following ordinal categories: 0: very dissatisfied,

1. somewhat dissatisfied
2. neutral
3. satisfied
4. very satisfied

Ordinal attributes may also be obtained from the discretization of numeric quantities by splitting the value range into a finite number of ordered categories. The central tendency of an ordinal attribute can be represented by its mode and its median (the middle value in an ordered sequence), but the mean cannot be defined.

Nominal, binary, and ordinal attributes are qualitative. That is, they describe a feature of an object without giving an actual size or quantity. The values of such qualitative attributes are typically words representing categories. If integers are used, they represent computer codes for the categories, as opposed to measurable quantities (e.g., 0 for small drink size, 1 for medium, and 2 for large). In the following subsection we look at numeric attributes, which provide quantitative measurements of an object.

Numeric Attributes

A **numeric attribute** is quantitative; that is, it is a measurable quantity, represented in integer or real values. Numeric attributes can be interval-scaled or ratio-scaled.

Interval-Scaled Attributes

Interval-scaled attributes are measured on a scale of equal-size units. The values of interval-scaled attributes have order and can be positive, 0, or negative. Thus, in addition to providing a ranking of values, such attributes allow us to compare and quantify the difference between values.

For example, **Interval-scaled attributes.** A temperature attribute is interval-scaled. Suppose that we have the outdoor temperature value for a number of different days, where each day is an object. By ordering the values, we obtain a ranking of the objects with respect to temperature. In addition, we can quantify the difference between values. For example, a temperature of 20_C is five degrees higher than a temperature of 15_C. Calendar dates are another example. For instance, the years 2002 and 2010 are eight years apart.

Temperatures in Celsius and Fahrenheit do not have a true zero-point, that is, neither 0_C nor 0_F indicates “no temperature.” (On the Celsius scale, for example, the unit of Measurement is 1/100 of the difference between the melting temperature and the boiling temperature of water in atmospheric pressure.) Although we can compute the difference between temperature values, we cannot talk of one temperature value as being a multiple of another. Without a true zero, we cannot say, for instance, that 10_C is twice as warm as 5_C. That is, we cannot speak of the values in terms of ratios. Similarly, there is no true zero-point for calendar dates. (The year 0 does not correspond to the beginning of time.) This brings us to ratio-scaled attributes, for which a true zero-point exits. Because interval-scaled attributes are numeric, we can compute their mean value, in addition to the median and mode measures of central tendency.

- **Ratio-Scaled Attributes**

  A **ratio-scaled attribute** is a numeric attribute with an inherent zero-point. That is, if a measurement is ratio-scaled, we can speak of a value as being a multiple (or ratio) of another value. In addition, the values are ordered, and we can also compute the difference between values, as well as the mean, median, and mode.

  For example **Ratio-scaled attributes.** Unlike temperatures in Celsius and Fahrenheit, the Kelvin (K) temperature scale has what is considered a true zero-point (0_K D =273.15_C): It is the point at which the particles that comprise matter have zero kinetic energy. Other examples of ratio-scaled attributes include count attributes such as years of experience (e.g., the objects are employees) and number of words (e.g., the objects are documents).

  Additional examples include attributes to measure weight, height, latitude and longitude coordinates (e.g., when clustering houses), and monetary quantities (e.g., you are 100 times richer with \$100 than with \$1).

▸ **Discrete versus Continuous Attributes**

Classification algorithms developed from the field of machine learning often talk of attributes as being either discrete or continuous. Each type may be processed differently. A **discrete attribute** has a finite or countable infinite set of values, which may or may not be represented as integers. If an attribute is not discrete, it is **continuous**. The terms numeric attribute and continuous attribute are often used interchangeably in the literature.

▸ **Basic Statistical Descriptions of Data**

For data preprocessing to be successful, it is essential to have an overall picture of your data. Basic statistical descriptions can be used to identify properties of the data and highlight which data values should be treated as noise or outliers.

We start with measures of central tendency, which measure the location of the middle or center of a data distribution. Intuitively speaking, given an attribute, where do most of its values fall? The most common data dispersion measures are the range, quartiles, and interquartile range; the five-number summary and boxplots; and the variance and standard deviation of the data these measures are useful for identifying outliers and are described. Finally, we can use many graphic displays of basic statistical descriptions to visually inspect our data.

1.7 Measuring the Central Tendency: Mean, Median, and Mode

Q11. Explain the process of measuring central tendency ?

Ans :

A measure of central tendency is a single value that attempts to describe a set of data by identifying the central position within that set of data. As such, measures of central tendency are sometimes called measures of central location.

In other words, in many real-life situations, it is helpful to describe data by a single number that is most representative of the entire collection of numbers. Such a number is called a measure of central tendency. The most commonly used measures are as follows. **Mean, Median, and Mode**

Mean: mean, or average, of numbers is the sum of the numbers divided by n. That is :

$$\bar{x} = \frac{(x_1 + x_2 + ... + x_n)}{n} \text{ i.e. Mean}$$

$$= \frac{\text{sum of all data values}}{\text{number of data values}}$$

shortly,

$$\bar{x} = \frac{\sum x}{n}$$

where $\bar{x}$ (read as ‘x bar’) is the mean of the set x values

$\sum x$ is the sum of all the x values, and

n is the number of x values

Example 1

The marks of seven students in a mathematics test with a maximum possible mark of 20 are given as

15 13 18 16 14 17 12.

Find the mean of this set of data values.

Solution :

$$\text{Mean} = \frac{\text{Sum of all data values}}{\text{Number of data values}}$$

$$= \frac{15+13+18+16+14+17+12}{7}$$

$$= \frac{105}{7} = 15$$

So, the mean mark is 15.

- **Midrange :** The midrange of a data set is the average of the minimum and maximum values.
- **Median :** median of numbers is the middle number when the numbers are written in order. If is even, the median is the average of the two middle numbers.

Example 2

The marks of nine students in a geography test that had a maximum possible mark of 50 are given as :

47 35 37 32 38 39 36 34 35.

Find the median of this set of data values.

Solution :

Arrange the data values in order from the lowest value to the highest value:

32 34 35 35 36 37 38 39 47

The fifth data value, 36, is the middle value in this arrangement.

Median = 36

The number of values, n in the data set = 9

$$\text{Mean} = \frac{1}{2}(9 + 1)\text{th value}$$

$$= \text{5th value}$$

$$= 36$$

Trimmed Mean

A trimming mean eliminates the extreme observations by removing observations from each end of the ordered sample. It is calculated by discarding a certain percentage of the lowest and the highest scores and then computing the mean of the remaining scores.

Mode of numbers is the number that occurs most frequently. If two numbers tie for most frequent occurrence, the collection has two modes and is called bimodal.

The mode has applications in printing. For example, it is important to print more of the most popular books; because printing different books in equal numbers would cause a shortage of some books and an oversupply of others.

Likewise, the mode has applications in manufacturing. For example, it is important to manufacture more of the most popular shoes; because manufacturing different shoes in equal numbers would cause a shortage of some shoes and an oversupply of others.

Example

Find the mode of the following data set :

48 44 48 45 42 49 48

Solution :

The mode is 48 since it occurs most often.

- It is possible for a set of data values to have more than one mode.
- If there are two data values that occur most frequently, we say that the set of data values is **bimodal**.
- If there is three data values that occur most frequently, we say that the set of data values is **trimodal**
- If two or more data values that occur most frequently, we say that the set of data values is **multimodal**
- If there is no data value or data values that occur most frequently, we say that the set of data values has no mode.

  The mean, median and mode of a data set are collectively known as measures of **central tendency** as these three measures focus on where the data is centered or clustered. To analyze data using the mean, median and mode, we need to use the most appropriate measure of central tendency. The following points should be remembered:

- The mean is useful for predicting future results when there are no extreme values in the data set. However, the impact of extreme values on the mean may be important and should be considered. E.g. the impact of a stock market crash on average investment returns.
- The median may be more useful than the mean when there are extreme values in the data set as it is not affected by the extreme values.
- The mode is useful when the most common item, characteristic or value of a data set is required.

1.7.1 Measuring the Dispersion of Data: Range, Quartiles, Variance, Standard Deviation, & Interquartile Range

Q12. Explain the following :

a) Range

b) Quartiles

c) Variance

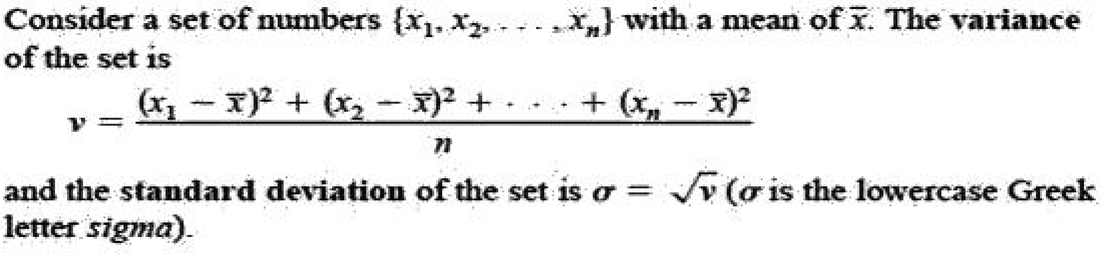
d) Standard Deviation, & Interquartile Range

Ans :

Measures of dispersion measure how spread out a set of data is. The two most commonly used measures of dispersion are the variance and the standard deviation. Rather than showing how data are similar, they show how data differs from its variation, spread, or dispersion. Other measures of dispersion that may be encountered include the Quartiles, Inter quartile range (IQR), five number summary, range and box plots.

Variance and Standard Deviation

Very different sets of numbers can have the same mean. You will now study two measures of dispersion, which give you an idea of how much the numbers in a set differ from the mean of the set. These two measures are called the variance of the set and the standard deviation of the set.



Consider a set of numbers $\{x_1, x_2, \ldots, x_n\}$ with a mean of $\bar{x}$. The variance of the set is

$$v = \frac{(x_1 - \bar{x})^2 + (x_2 - \bar{x})^2 + \cdots + (x_n - \bar{x})^2}{n}$$

and the standard deviation of the set is $\sigma = \sqrt{v}$ (σ is the lowercase Greek letter *sigma*).

The standard deviation of a set is a measure of how much a typical number in the set differs from the mean. The greater the standard deviation, the more the numbers in the set vary from the mean. For instance, each o the following sets ha a mean of 5.

{5, 5, 5, 5}, {4, 4, 6, 6} and {3, 3, 7, 7}

The standard deviation of the sets are 0, 1 and 2.

$$\sigma_1 = \sqrt{\frac{(5-5)^2 + (5-5)^2 + (5-5)^2 + (5-5)^2}{4}}$$

$$= 0$$

$$\sigma_2 = \sqrt{\frac{(4-5)^2 + (4-5)^2 + (6-5)^2 + (6-5)^2}{4}}$$

$$= 1$$

$$\sigma_3 = \sqrt{\frac{(3-5)^2 + (3-5)^2 + (7-5)^2 + (7-5)^2}{4}}$$

$$= 2$$

Percentile

Percentiles are values that divide a sample of data into one hundred groups containing (as far as possible) equal numbers of observations.

The pth percentile of a distribution is the value such that p percent of the observations fall at or below it. The most commonly used percentiles other than the median are the 25th percentile and the 75th percentile.

The 25th percentile demarcates the first quartile, the median or 50th percentile demarcates the second quartile, the 75th percentile demarcates the third quartile, and the 100th percentile demarcates the fourth quartile.

Quartiles

Quartiles are numbers that divide an ordered data set into four portions, each containing approximately one-fourth of the data. Twenty-five percent of the data values come before the first quartile (Q1). The median is the second quartile (Q2); 50% of the data values come before the median. Seventy-five percent of the data values come before the third quartile (Q3).

- Q1 = 25th percentile = (n*25/100), where n is total number of data in the given data set

- Q2 = median = 50th percentile = (n*50/100) Q3 = 75th percentile = (n*75/100)

Inter Quartile Range (IQR)

The inter quartile range is the length of the interval between the lower quartile (Q1) and the upper quartile (Q3). This interval indicates the central, or middle, 50% of a data set.

IQR = Q3 – Q1

Range

The range of a set of data is the difference between its largest (maximum) and smallest (minimum) values. In the statistical world, the range is reported as a single number, the difference between maximum and minimum. Sometimes, the range is often reported as "from (the minimum) to (the maximum)," i.e., two numbers.

Example

Given data set : 3, 4, 4, 5, 6, 8

The range of data set is 3–8. The range gives only minimal information about the spread of the data, by defining the two extremes. It says nothing about how the data are distributed between those two endpoints.

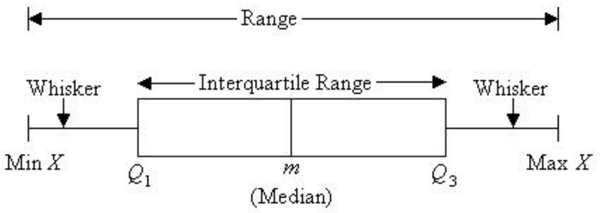
Box plots

A box plot is a graph used to represent the range, median, quartiles and inter quartile range of a set of data values. Constructing a Box plot: To construct a box plot:

I. Draw a box to represent the middle 50% of the observations of the data set.

II. Show the median by drawing a vertical line within the box.

III. Draw the lines (called **whiskers**) from the lower and upper ends of the box to the minimum and maximum values of the data set respectively, as shown in the following diagram.



- X is the set of data values.
- Min X is the minimum value in the data set.
- Max X is the maximum value in the data set.

Example : Draw a boxplot for the following data set of scores :

76 79 76 74 75 71 85 82 82 79 81

Solution :

Step 1: Arrange the score values in ascending order of magnitude :

71 74 75 76 76 79 79 81 82 82 85

There are 11 values in the data set.

Step 2: Q1=25th percentile value in the given data set

Q1 = 11*(25/100) th value

= 2.75 ⇒ 3rd value

= 75

Step 3: Q2 = median = 50th percentile value

= 11 * (50/100)th value

= 5.5th value ⇒ 6th value

= 79

Step 4: Q3 = 75th percentile value

= 11*(75/100)th value

= 8.25th value ⇒ 9th value

= 82

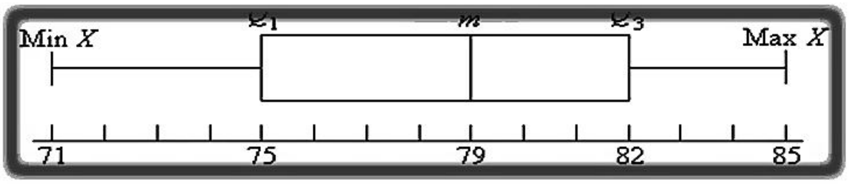
Step 5 : Min X= 71

Step 6 : Max X=85

Step 7 : Range= 85-71 = 14

Step 8 : IQR = height of the box = Q3–Q1

= 9 – 3 = 6th value = 79



Outliers

Outlier data is a data that falls outside the range. Outliers will be any points below Q1 – 1.5 × IQR or above Q3 + 1.5 × IQR.

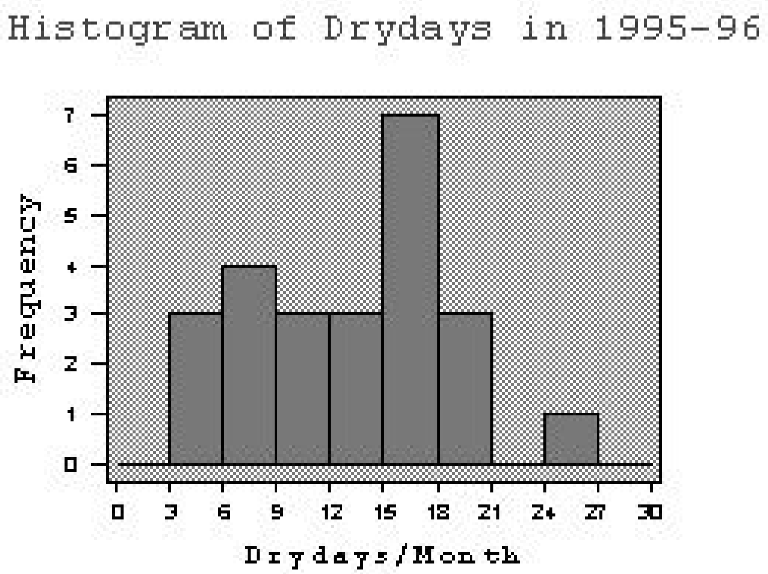
1.7.2 Graphic Displays of Basic Statistical Descriptions of Data

Q13. Explain different types of Graphical Representation of Data ?

Ans :

Histogram

A histogram is a way of summarizing data that are measured on an interval scale (either discrete or continuous). It is often used in exploratory data analysis to illustrate the major features of the distribution of the data in a convenient form. It divides up the range of possible values in a data set into classes or groups. For each group, a rectangle is constructed with a base length equal to the range of values in that specific group, and an area proportional to the number of observations falling into that group. This means that the rectangles might be drawn of non-uniform height.

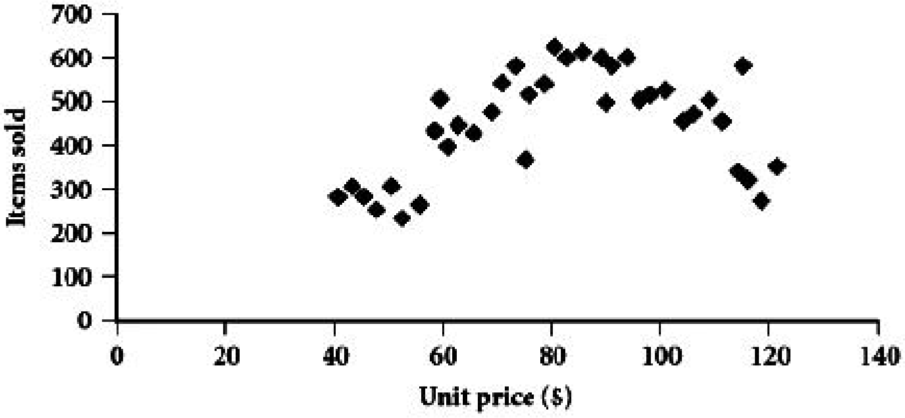


The histogram is only appropriate for variables whose values are numerical and measured on an interval scale. It is generally used when dealing with large data sets (>100 observations). A histogram can also help detect any unusual observations (outliers), or any gaps in the data set.

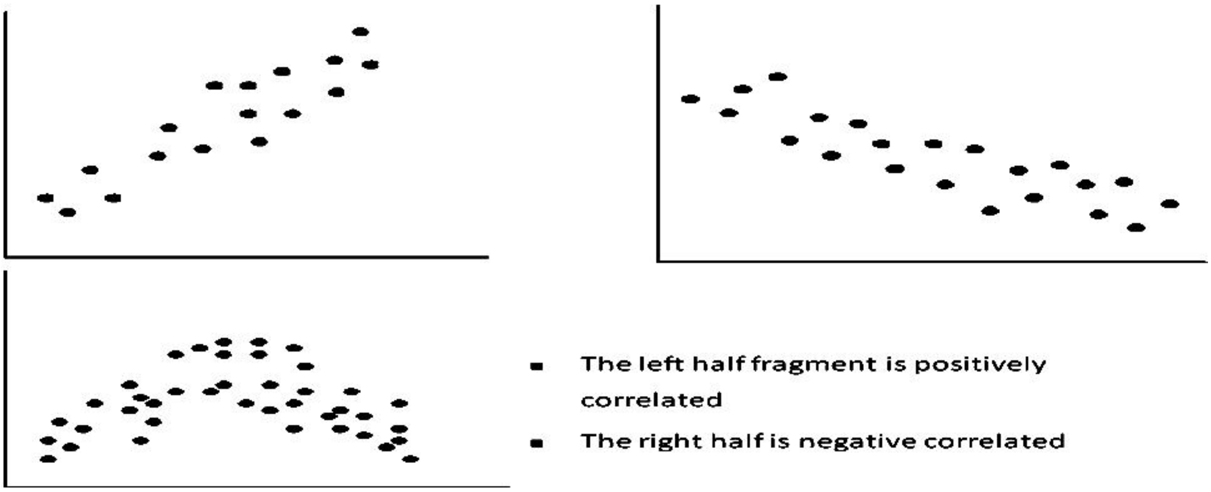
Scatter Plot

A scatter plot is a useful summary of a set of bivariate data (two variables), usually drawn before working out a linear correlation coefficient or fitting a regression line. It gives a good visual picture of the relationship between the two variables, and aids the interpretation of the correlation coefficient or regression model.

Each unit contributes one point to the scatter plot, on which points are plotted but not joined. The resulting pattern indicates the type and strength of the relationship between the two variables.



Positively and Negatively Correlated Data

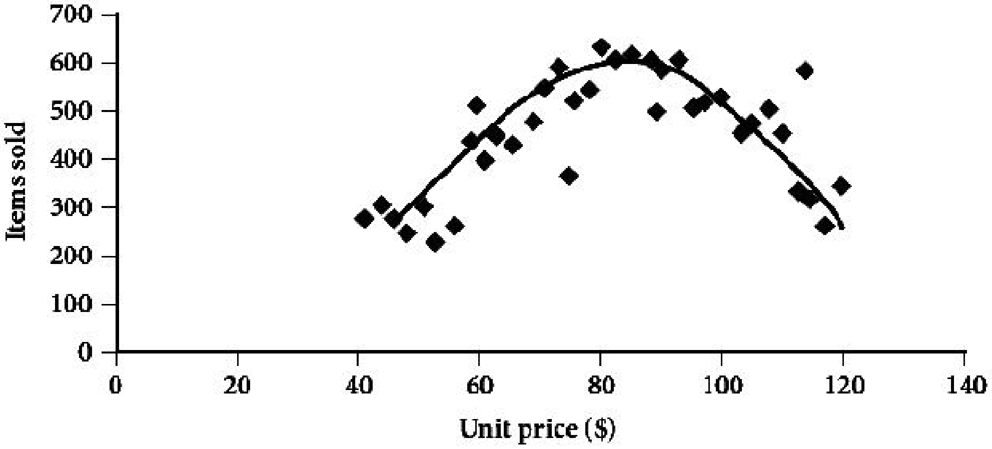


- The left half fragment is positively correlated
- The right half is negative correlated

A scatter plot will also show up a non-linear relationship between the two variables and whether or not there exist any outliers in the data.

Loose curve

It is another important exploratory graphic aid that adds a smooth curve to a scatter plot in order to provide better perception of the pattern of dependence. The word loose is short for “local regression.”

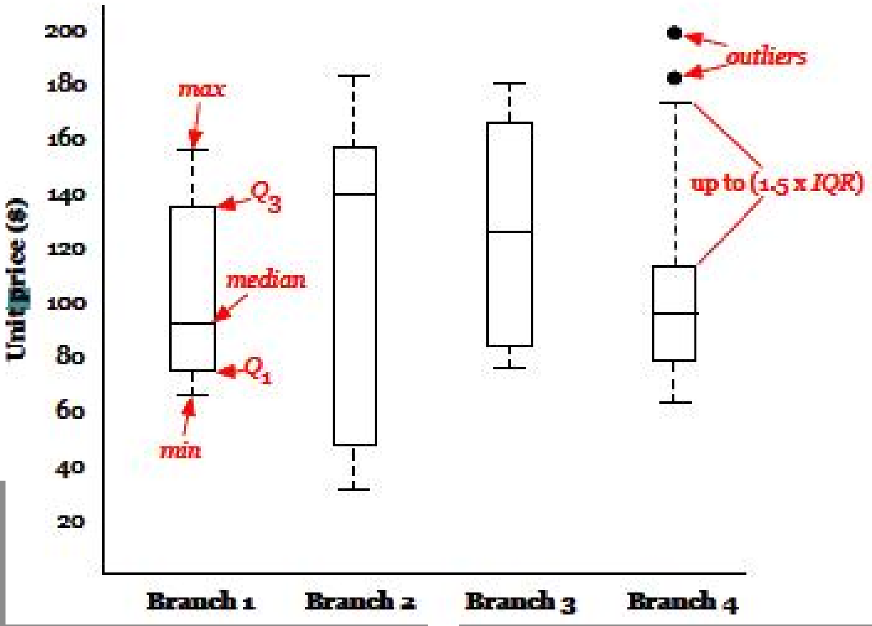


Box plot

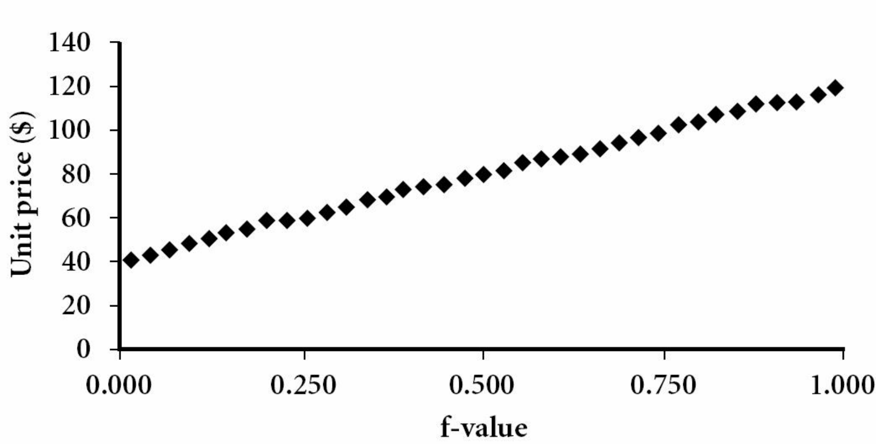
The picture produced consists of the most extreme values in the data set (maximum and minimum values), the lower and upper quartiles, and the median.

Quantile plot

- Displays all of the data (allowing the user to assess both the overall behavior and unusual occurrences)



- Plots quantile information
  - o For a data x_i data sorted in increasing order, f_i indicates that approximately 100 f_i% of the data are below or equal to the value x_i



The f quantile is the data value below which approximately a decimal fraction f of the data is found. That data value is denoted q(f). Each data point can be assigned an f-value. Let a time series x of length n be sorted from smallest to largest values, such that the sorted values have rank. The f-value for each observation is computed as . 1,2,..., n . The f-value for each observation is computed as,

$$f_1 = \frac{i-0.5}{n}$$

Quantile-Quantile plots (Q-Q plot)

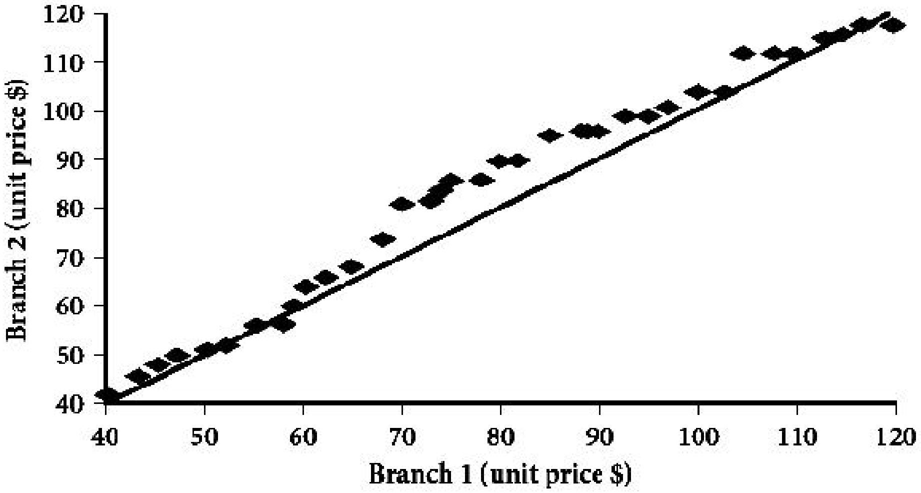
Quantile-quantile plots allow us to compare the quantiles of two sets of numbers. This kind of comparison is much more detailed than a simple comparison of means or medians.

A normal distribution is often a reasonable model for the data. Without inspecting the data, however, it is risky to assume a normal distribution. There are a number of graphs that can be used to check the deviations of the data from the normal distribution. The most useful tool for assessing normality is a quantile or QQ plot. This is a scatter plot with the quantiles of the scores on the horizontal axis and the expected normal scores on the vertical axis.

In other words, it is a graph that shows the quantiles of one univariate distribution against the corresponding quantiles of another. It is a powerful visualization tool in that it allows the user to view whether there is a shift in going from one distribution to another. The steps in constructing a QQ plot are as follows :

- First, we sort the data from smallest to largest. A plot of these scores against the expected normal scores should reveal a straight line.
- The expected normal scores are calculated by taking the z-scores of (I - ½)/n where I is the rank in increasing order.

Curvature of the points indicates departures of normality. This plot is also useful for detecting outliers. The outliers appear as points that are far away from the overall pattern op points



How is a quantile-quantile plot different from a quantile plot.

A quantile plot is a graphical method used to show the approximate percentage of values below or equal to the independent variable in a univariate distribution. Thus, it displays quantile information for all the data, where the values measured for the independent variable are plotted against their corresponding quantile.

A quantile-quantile plot however, graphs the quartiles of one univariate distribution against the corresponding quantiles of another univariate distribution. Both axes display the range of values measured for their corresponding distribution, and points are plotted that correspond to the quantile values of the two distributions. A line (y = x) can be added to the graph along with points representing where the first, second and third quantiles lie, in order to increase the graph’s informational value. Points that lie above such a line indicate a correspondingly higher value for the distribution plotted on the y-axis, than for the distribution plotted on the x-axis at the same quantile. The opposite effect is true for points lying below this line.

1.8 Data preprocessing

Q14. Explain Data Processing & Why ?

Ans :

Introduction

Today's real-world databases are highly susceptible to noisy, missing, and inconsistent data due to their typically huge size (often several gigabytes or more) and their likely origin from multiple, heterogenous sources. Low-quality data will lead to low-quality mining results. "How can the data be preprocessed in order to help improve the quality of the data and, consequently, of the mining results? How can the data be preprocessed so as to improve the efficiency and ease of the mining process?"

There are several data preprocessing techniques.

- *Data cleaning* can be applied to remove noise and correct inconsistencies in data.
- *Data integration* merges data from multiple sources into a coherent data store such as a data warehouse.
- *Data reduction* can reduce data size by, for instance, aggregating, eliminating redundant features, or clustering.
- *Data transformations* (e.g., normalization) may be applied, where data are scaled to fall within a smaller range like 0.0 to 1.0.

This can improve the accuracy and efficiency of mining algorithms involving distance measurements. These techniques are not mutually exclusive; they may work together. For example, data cleaning can involve transformations to correct wrong data, such as by transforming all entries for a date field to a common format.

Why preprocessing ?

1. Real world dare generally

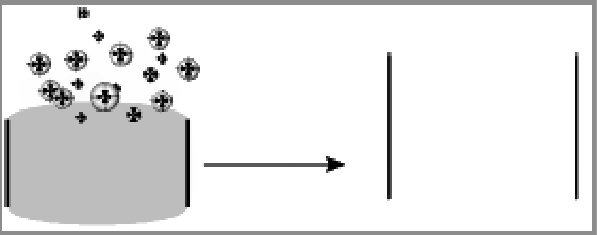
- Incomplete: lacking attribute values, lacking certain attributes of interest, or containing only aggregate data
- Noisy: containing errors or outliers
- Inconsistent: containing discrepancies in codes or names

2. Tasks in data preprocessing

- **Data cleaning :** fill in missing values, smooth noisy data, identify or remove outliers, and resolve inconsistencies.
- **Data integration :** using multiple databases, data cubes, or files.
- **Data transformation** : normalization and aggregation.
- **Data reduction** : reducing the volume but producing the same or similar analytical results.
- **Data discretization** : part of data reduction, replacing numerical attributes with nominal ones.

Data cleaning

1. Fill in missing values (attribute or class value) :



- Ignore the tuple: usually done when class label is missing.
- Use the attribute mean (or majority nominal value) to fill in the missing value.
- Use the attribute mean (or majority nominal value) for all samples belonging to the same class.
- Predict the missing value by using a learning algorithm: consider the attribute with the missing value as a dependent (class) variable and run a learning algorithm (usually Bayes or decision tree) to predict the missing value.

2. Identify outliers and smooth out noisy data :

- Binning
  - Sort the attribute values and partition them into bins (see "Unsupervised discretization" below);
  - Then smooth by bin means, bin median, or bin boundaries.

  - **Clustering:** group values in clusters and then detect and remove outliers (automatic or manual)
  - **Regression:** smooth by fitting the data into regression functions.

3. Correct inconsistent data: use domain knowledge or expert decision.

Data Transformation

1. *Normalization :*

- Scaling attribute values to fall within a specified range.
  - Example: to transform V in [min, max] to V' in [0,1], apply V'=(V-Min)/(Max-Min)
- Scaling by using mean and standard deviation (useful when min and max are unknown or when there are outliers): V'=(V-Mean)/StDev

2. *Aggregation :* moving up in the concept hierarchy on numeric attributes.
3. *Generalization* : moving up in the concept hierarchy on nominal attributes.
4. Attribute construction : replacing or adding new attributes inferred by existing attributes.

Data Teduction

1. Reducing the number of attributes
   - *Data cube aggregation*: applying roll-up, slice or dice operations.
   - *Removing irrelevant attributes*: attribute selection (filtering and wrapper methods), searching the attribute space (see Lecture 5: Attribute-oriented analysis).
   - *Principle component analysis (numeric attributes only)*: searching for a lower dimensional space that can best represent the data..
2. Reducing the number of attribute values
   - Binning (histograms): reducing the number of attributes by grouping them into intervals (bins).
   - Clustering : grouping values in clusters.
   - Aggregation or generalization
3. Reducing the number of tuples
   - Sampling

Discretization and Generating Concept Hierarchies

1. Unsupervised discretization - class variable is not used.
   - Equal-interval (equiwidth) binning: split the whole range of numbers in intervals with equal size.
   - Equal-frequency (equidepth) binning: use intervals containing equal number of values.
2. Supervised discretization - uses the values of the class variable.
   - Using class boundaries. Three steps :
     - Sort values.
     - Place breakpoints between values belonging to different classes.
     - If too many intervals, merge intervals with equal or similar class distributions.
   - Entropy (information)-based discretization. Example :
     - Information in a class distribution:
     - Denote a set of five values occurring in tuples belonging to two classes (+ and -) as [+,+,+,-,-]

- That is, the first 3 belong to “+” tuples and the last 2 - to “–” tuples
- Then, Info([+,+,+,-,-]) = -(3/5)*log(3/5)-(2/5)*log(2/5) (logs are base 2)
- 3/5 and 2/5 are relative frequencies (probabilities)
- Ignoring the order of the values, we can use the following notation: [3,2] meaning 3 values from one class and 2 - from the other.
- Then, Info([3,2]) = -(3/5)*log(3/5)-(2/5)*log(2/5)

- Information in a split (2/5 and 3/5 are weight coefficients):
  - Info([+,+],[+,-,-]) = (2/5)*Info([+,+]) + (3/5)*Info([+,-,-])
  - Or, Info([2,0],[1,2]) = (2/5)*Info([2,0]) + (3/5)*Info([1,2])
- Method :
  - Sort the values;
  - Calculate information in all possible splits;
  - Choose the split that minimizes information;
  - Do not include breakpoints between values belonging to the same class (this will increase information);
  - Apply the same to the resulting intervals until some stopping criterion is satisfied.

3. Generating concept hierarchies: recursively applying partitioning or discretization methods.

1.9 Data Cleaning

Q15. Explain the process of Data cleaning ?

Ans :

Real-world data tend to be incomplete, noisy, and inconsistent. Data cleaning (or data cleansing) routines attempt to fill in missing values, smooth out noise while identi- fying outliers, and correct inconsistencies in the data. In this section, you will study basic methods for data cleaning.

1. Missing Values

- **Ignore the tuple**: This is usually done when the class label is missing (assuming the mining task involves classification). This method is not very effective, unless the tuple contains several attributes with missing values. . It is especially poor when the percent- age of missing values per attribute varies considerably. By ignoring the tuple, we do not make use of the remaining attributes’ values in the tuple. Such data could have been useful to the task at hand.
- **Fill in the missing value manually**: In general, this approach is time consuming and may not be feasible given a large data set with many missing values.
- **Use a global constant to fill in the missing value**: Replace all missing attribute values by the same constant such as a label like “Unknown” or ”. If missing values are replaced by, say, “Unknown,” then the mining program may mistakenly think that they form an interesting concept, since they all have a value in common—that of “Unknown.”.
- **Use a measure of central tendency for the attribute to fill in the missing value**: For normal (symmetric) data dis- tributions, the mean can be used, while skewed data distribution should employ the median.

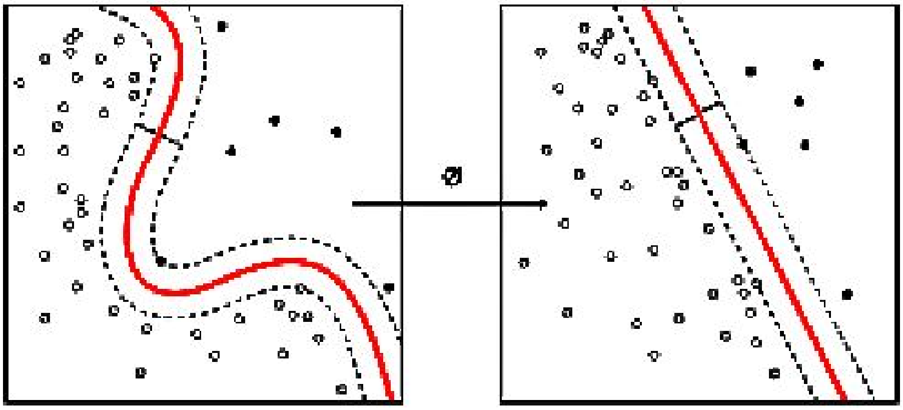
- **Use the attribute mean or median for all samples belonging to the same class as the given tuple**: If the data distribution for a given class is skewed, the median value is a better choice. For example, if classifying customers according to credit risk, we may replace the missing value with the mean income value for customers in the same credit risk category as that of the given tuple.
- **Use the most probable value to fill in the missing value**: This may be determined with regression, inference-based tools using a Bayesian formalism, or decision tree induction.

2. Noisy Data

"What is noise?" **Noise** is a random error or variance in a measured variable. Some basic statistical description techniques and methods of data visualization can be used to identify outliers, which may represent noise. We "smooth" out the data to remove the noise? Following are the data smoothing techniques.

- **Binning:** Binning methods smooth a sorted data value by consulting its "neighbor- hood," that is, the values around it. The sorted values are distributed into a number of "buckets," or bins. Because binning methods consult the neighborhood of values, they perform local smoothing. In **smoothing by bin means**, each value in a bin is replaced by the mean value of the bin. For example the data for price are first sorted and then partitioned into equal-frequency bins of size 3 (i.e., each bin contains three values).

  Similarly, **smoothing by bin medians** can be employed, in which each bin value is replaced by the bin median. In **smoothing by bin boundaries**, the minimum and maximum values in a given bin are identified as the bin boundaries. Each bin value is then replaced by the closest boundary value.
- **Regression:** Data smoothing can also be done by regression, a technique that con- forms data values to a function. Linear regression involves finding the "best" line to fit two attributes (or variables) so that one attribute can be used to predict the other. Multiple linear regression is an extension of linear regression, where more than two attributes are involved and the data are fit to a multidimensional surface.
- **Outlier analysis**: Outliers may be detected by clustering; similar values are organized into groups, or "clusters." Intuitively, values that fall outside of the set of clusters may be considered outliers. Many data smoothing methods are also used for data discretization (a form of data transformation) and data reduction. This acts as a form of data reduction for logic-based data mining methods, such as decision tree induction, which repeatedly makes value comparisons on sorted data. Concept hierarchies are a form of data discretization that can also be used for data smoothing.



Data Cleaning as a Process

Missing values, noise, and inconsistencies contribute to inaccurate data. So far, we have looked at techniques for handling missing data and for smoothing data. “But data clean- ing is a big job. What about data cleaning as a process? How exactly does one proceed in tackling this task? Are there any tools out there to help?”

The first step in data cleaning as a process is discrepancy detection. Discrepancies can be caused by several factors, including poorly designed data entry forms that have many optional fields, human error in data entry, deliberate errors (e.g., respondents not want- ing to divulge information about themselves), and data decay (e.g., outdated addresses). Discrepancies may also arise from inconsistent data representations and inconsistent use of codes.

Other sources of discrepancies include errors in instrumentation devices that record data and system errors. Errors can also occur when the data are (inadequately) used for purposes other than originally intended. There may also be inconsistencies due to data integration. “So, how can we proceed with discrepancy detection?” As a starting point, use any knowledge you may already have regarding properties of the data. Such knowledge or “data about data” is referred to as **metadata**. **Field overloading** is another error source that typically results when developers squeeze new attribute definitions into unused (bit) portions of already defined attributes.

The data should also be examined regarding unique rules, consecutive rules, and null rules. A **unique rule** says that each value of the given attribute must be different from all other values for that attribute. A **consecutive rule** says that there can be no miss- ing values between the lowest and highest values for the attribute, and that all values must also be unique (e.g., as in check numbers). A **null rule** specifies the use of blanks, question marks, special characters, or other strings that may indicate the null condition and how such values should be handled.

There are a number of different commercial tools that can aid in the discrepancy detection step. **Data scrubbing tools** use simple domain knowledge (e.g., knowledge of postal addresses and spell-checking) to detect errors and make corrections in the data. These tools rely on parsing and fuzzy matching techniques when cleaning data from multiple sources. **Data auditing tools** find discrepancies by analyzing the data to discover rules and relationships, and detecting data that violate such conditions. They are variants of data mining tools.

Some data inconsistencies may be corrected manually using external references. For example, errors made at data entry may be corrected by performing a papertrace. Most errors, however, will require data transformations. That is, once we find discrepancies, we typically need to define and apply (a series of) transformations to correct them.

Commercial tools can assist in the data transformation step. **Data migration tools** allow simple transformations to be specified such as to replace the string “gender” by “sex.” **ETL (extraction/transformation/loading) tools** allow users to specify transforms through a graphical user interface (GUI). These tools typically support only a restricted set of transforms so that, often, we may also choose to write custom scripts for this step of the data cleaning process.

The two-step process of discrepancy detection and data transformation (to correct discrepancies) iterates. This process, however, is error-prone and time consuming. Some transformations may introduce more discrepancies. Some nested discrepancies may only be detected after others have been fixed. Transformations are often done as a batch process while the user waits without feedback. Only after the transformation is complete can the user go back and check that no new anomalies have been mistakenly created. Typically, numerous iterations are required before the user is satisfied. Any tuples that cannot be automatically handled by a given transformation are typically written to a file without any explanation regarding the rea- soning behind their failure. As a result, the entire data cleaning process also suffers from a lack of interactivity.

1.10 Data Integration

Q16. Explain about Data Integration ?

Ans :

Data mining often requires data integration— the merging of data from multiple data stores. Careful integration can help reduce and avoid redundancies and inconsistencies in the resulting data set. This can help improve the accuracy and speed of the subsequent data mining process. The semantic heterogeneity and structure of data pose great challenges in data inte- gration. How can we match schema and objects from different sources? This is the essence of the entity identification problem,

- **Entity Identification Problem**

  It is likely that your data analysis task will involve data integration, which combines data from multiple sources into a coherent data store, as in data warehousing. These sources may include multiple databases, data cubes, or flat files.

  There are a number of issues to consider during data integration. Schema integration and object matching can be tricky. How can the data analyst or the computer be sure that customer_id in one database and cust_number in another refer to the same attribute? Examples of metadata for each attribute include the name, meaning, data type, and range of values permitted for the attribute, and null rules for handling blank, zero, or null values. When matching attributes from one database to another during integration, special attention must be paid to the structure of the data. This is to ensure that any attribute functional dependencies and referential constraints in the source system match those in the target system.

- **Redundancy and Correlation Analysis**

  Redundancy is another important issue in data integration. An attribute (such as annual revenue, for instance) may be redundant if it can be "derived" from another attribute or set of attributes. Inconsistencies in attribute or dimension naming can also cause redundancies in the resulting data set.

  Some redundancies can be detected by **correlation analysis**. Given two attributes, such analysis can measure how strongly one attribute implies the other, based on the available data. For nominal data, we use the x^2 (chi-square) test. For numeric attributes, we can use the correlation coefficient and covariance, both of which access how one attribute's values vary from those of another.

- **Correlation Coefficient for Numeric Data**

  For nominal data, a correlation relationship between two attributes, A and B, can be discovered by a χ^2 (**chi-square**) test. Suppose A has c distinct values, namely $a1$, $a2$, . . . ac . B has r distinct values, namely $b1$, $b2$, . . . br . The data tuples described by A and B can be shown as a **contingency table**, with the c values of A making up the columns and the r values of B making up the rows. Let (Ai , Bj) denote the joint event that attribute A takes on value ai and attribute B takes on value bj , that is, where (A = ai , B = bj). Each and every possible (Ai, Bj) joint event has its own cell (or slot) in the table. The χ^2 value (also known as the Pearson χ^2 statistic) is computed as

$$\frac{\sum(A-\bar{A})(B-B)}{(n-1)\sigma_A\,\sigma_B}$$

  where oij is the observed frequency (i.e., actual count) of the joint event (Ai , Bj) and eij is the expected frequency of (Ai, Bj), which can be computed as

$$\chi^2 = \sum\frac{(\text{Observed}-\text{Expected})^2}{\text{Expected}}$$

  Where n is the number of data tuples, count (A = ai) is the number of tuples having value ai for A, and count (B = bj) is the number of tuples having value bj for B. The sum is computed over all of the r × c cells. Note that the cells that contribute the most to the x^2 value are those for which the actual count is very different from that expected.

The x^2 statistic tests the hypothesis that A and B are independent, that is, there is no correlation between them. The test is based on a significance level, with (r – 1) × (c – 1) degrees of freedom.

- **Correlation Coefficient for Numeric Data**

For numeric attributes, we can evaluate the correlation between two attributes, A and B, by computing the **correlation coefficient** (also known as **Pearson's product moment coefficient**, named after its inventer, Karl Pearson). This is

$$\rho_{AB} = \frac{\sigma_{AB}}{\sigma_A \sigma_B} = \frac{\sigma_{AB}}{\sqrt{\sigma_A^2 B^2}}$$

where n is the number of tuples, a_i and b_i are the respective values of A and B in tuple i, $\bar{A}$ and $\bar{B}$ are the respective mean values of A and B, σ_A and σ_B are the respective standard deviations of A and B and ($a_i b_i$) is the sum of the AB cross-product (i.e., for each tuple, the value for A is multiplied by the value for B in that tuple). Note that $-1 \le r_{A,B} \le +1$. If $r_{A,B}$ is greater than 0, then A and B are positively correlated, meaning that the values of A increase as the values of B increase. The higher the value, the stronger the correlation (i.e., the more each attribute implies the other).

Hence, a higher value may indicate that A (or B) may be removed as a redundancy. If the resulting value is equal to 0, then A and B are independent and there is no correlation between them. If the resulting value is less than 0, then A and B are negatively correlated, where the values of one attribute increase as the values of the other attribute decrease.

- **Covariance of Numeric Data**

In probability theory and statistics, correlation and covariance are two similar measures for assessing how much two attributes change together. Consider two numeric attributes A and B, and a set of n observations $\{(a_1, b_1), \ldots, (a_n, b_n)\}$. The mean values of A and B, respectively, are also known as the **expected values** on A and B, that is,

$$\text{Cov}(A, B) \text{ or } \sigma_{AB} = E[(A - E(A))(B - E(B))]$$

The **covariance** between A and B is defined as

$$\text{Cov}(A, B) = E((A - \bar{A})(B - \bar{B})) =$$

$$\sigma_{AB} = E(AB) - E(A)\,E(B)$$

If we compare Eq. (3.3) for $r_{A,B}$ (correlation coefficient) with Eq. (3.4) for covariance, we see that,

where σ_A and σ_B are the standard deviations of A and B, respectively. It can also be shown that

$$\text{Cov}(A, B) = E(A \cdot B) - \bar{A}\bar{B}.$$

This equation may simplify calculations.

For two attributes A and B that tend to change together, if A is larger than $\bar{A}$ (the expected value of A), then B is likely to be larger than $\bar{B}$ (the expected value of B).

Therefore, the covariance between A and B is positive. On the other hand, if one of the attributes tends to be above its expected value when the other attribute is below its expected value, then the covariance of A and B is negative.

If A and B are independent (i.e., they do not have correlation), then E(A·B) = E(A)·E(B). Therefore, the covariance is Cov(A, B) = $E(A \cdot B) - \bar{A}\bar{B} = E(A) \cdot E(B) - \bar{A}\bar{B} = 0$. However, the converse is not true. Some pairs of random variables (attributes) may have a covariance of 0 but are not independent. Only under some additional assumptions

- **Tuple Duplication**

In addition to detecting redundancies between attributes, duplication should also be detected at the tuple level. The use of denormalized tables is another source of data redundancy. Inconsistencies often arise between various duplicates, due to inaccurate data entry or updating some but not all data occurrences. For example, if a purchase order database contains attributes for the purchaser's name and address instead of a key to this information in a purchaser database, discrepancies can occur, such as the same purchaser's name appearing with different addresses within the purchase order database.

- **Data Value Conflict Detection and Resolution**

Data integration also involves the detection and resolution of data value conflicts. For example, for the same real-world entity, attribute values from different sources may differ. This may be due to differences in representation, scaling, or encoding. For instance, a weight attribute may be stored in metric units in one system and British imperial units in another. For a hotel chain, the price of rooms in different cities may involve not only different currencies but also different services (e.g., free breakfast) and taxes. When exchanging information between schools, for example, each school may have its own curriculum and grading scheme. One university may adopt a quarter system, offer three courses on database systems, and assign grades from A+ to F, whereas another may adopt a semester system, offer two courses on databases, and assign grades from 1 to 10. It is difficult to work out precise course-to-grade transformation rules between the two universities, making information exchange difficult.

Attributes may also differ on the abstraction level, where an attribute in one sys- tem is recorded at, say, a lower abstraction level than the "same" attribute in another. For example, the total sales in one database may refer to one branch of All Electronics, while an attribute of the same name in another database may refer to the total sales for All Electronics stores in a given region. The topic of discrepancy detection is further described in Section 3.2.3 on data cleaning as a process.

Data Reduction

Imagine that we have selected data from the All Electronics devices to data warehouse for analysis. The data set will likely be huge! Complex data analysis and mining on huge amounts of data can take a long time, making such analysis impractical or infeasible.

Data reduction techniques can be applied to obtain a reduced representation of the data set that is much smaller in volume, yet closely maintains the integrity of the original data. Mining on the reduced data set should be more efficient yet produce the same analytical results.

- **Data Reduction Strategies**

Data reduction strategies include

- Dimensionality reduction
- Numerosity reduction
- Data compression.

- **Dimensionality reduction** is the process of reducing the number of random variables or attributes under consideration. Dimensionality reduction methods include wavelet transforms and principal components analysis, which transform or project the original data onto a smaller space. Attribute subset selection is a method of dimensionality reduction in which irrelevant, weakly relevant, or redundant attributes or dimensions are detected and removed.

- **Numerosity reduction** techniques replace the original data volume by alternative, smaller forms of data representation. These techniques may be parametric or non- parametric. For parametric methods, a model is used to estimate the data, so that typically only the data parameters need to be stored, instead of the actual data. Regression and log-linear models are examples. Nonparametric methods for storing reduced representations of the data include histograms, clustering, sampling, and data cube aggregation.

- In **data compression**, transformations are applied so as to obtain a reduced or "compressed" representation of the original data. If the original data can be reconstructed from the compressed data without any information loss, the data reduction is called **lossless**. If, instead, we can reconstruct only an approximation of the original data, then the data reduction is called **lossy**. There are several lossless algorithms for string compression; however, they typically allow only limited data manipulation. Dimensionality reduction and numerosity reduction techniques can also be considered forms of data compression. There are many other ways of organizing methods of data reduction. The computational time spent on data reduction should not outweigh or "erase" the time saved by mining on a reduced data set size.

1.11 Wavelet Transforms

Q17. Discuss about Wavelet Transformation of Data in Data Mining Process ?

Ans :

The **discrete wavelet transform (DWT)** is a linear signal processing technique that, when applied to a data vector **X**, transforms it to a numerically different vector, $\mathbf{X}^0$, of **wavelet coefficients**. The two vectors are of the same length. When applying this tech- nique to data reduction, we consider each tuple as an n-dimensional data vector, that is, **X** = (x_1 , x_2 , . . . , x_n), depicting n measurements made on the tuple from n database attributes.

"How can this technique be useful for data reduction if the wavelet transformed data are of the same length as the original data?" The usefulness lies in the fact that the wavelet transformed data can be truncated. A compressed approximation of the data can be retained by storing only a small fraction of the strongest of the wavelet coefficients.

The technique also works to remove noise without smoothing out the main features of the data, making it effective for data cleaning as well. Given a set of coefficients, an approximation of the original data can be constructed by applying the inverse of the DWT used.

The DWT is closely related to the discrete Fourier transform (DFT), a signal process- ing technique involving sines and cosines. In general, however, the DWT achieves better lossy compression. That is, if the same number of coefficients is retained for a DWT and a DFT of a given data vector, the DWT version will provide a more accurate approxima- tion of the original data. Hence, for an equivalent approximation, the DWT requires less space than the DFT. Unlike the DFT, wavelets are quite localized in space, contributing to the conservation of local detail.

There is only one DFT, yet there are several families of DWTs. Popular wavelet transforms include the Haar-2, Daubechies-4, and Daubechies-6. The general procedure for applying a discrete wavelet transform uses a hierarchical pyramid algorithm that halves the data at each iteration, resulting in fast computational speed. The method is as follows:

1. The length, L, of the input data vector must be an integer power of 2. This condition can be met by padding the data vector with zeros as necessary ($L \geq n$).
2. Each transform involves applying two functions. The first applies some data smoothing, such as a sum or weighted average. The second performs a weighted difference, which acts to bring out the detailed features of the data.
3. The two functions are applied to pairs of data points in X, that is, to all pairs of measurements (x_{2i}, x_{2i+1}). This results in two data sets of length L/2. In general, these represent a smoothed or low-frequency version of the input data and the high- frequency content of it, respectively.
4. The two functions are recursively applied to the data sets obtained in the previous loop, until the resulting data sets obtained are of length 2.
5. Selected values from the data sets obtained in the previous iterations are designated the wavelet coefficients of the transformed data.

Equivalently, a matrix multiplication can be applied to the input data in order to obtain the wavelet coefficients, where the matrix used depends on the given DWT. The matrix must be orthonormal, meaning that the columns are unit vectors and are mutually orthogonal, so that the matrix inverse is just its transpose. Although we do not have room to discuss it here, this property allows the reconstruction of the data from the smooth and smooth-difference data sets. By factoring the matrix used into a product of a few sparse matrices, the resulting "fast DWT" algorithm has a complexity of O(n) for an input vector of length n. Wavelet transforms have many real- world applications, including the compression of fingerprint images, computer vision, analysis of time-series data, and data cleaning.

Principal Components Analysis

In this subsection we provide an intuitive introduction to principal components analy- sis as a method of dimensionality reduction.

Suppose that the data to be reduced consist of tuples or data vectors described by n attributes or dimensions. **Principal components analysis** (**PCA**; also called the Karhunen-Loeve, or K-L, method) searches for k n-dimensional orthogonal vectors that can best be used to represent the data, where k d" n. The original data are thus projected onto a much smaller space, resulting in dimensionality reduction. Unlike attribute sub- set selection, which reduces the attribute set size by retaining a subset of the initial set of attributes, PCA "combines" the essence of attributes by creating an alternative, smaller set of variables. The initial data can then be projected onto this smaller set. PCA often reveals relationships that were not previously suspected and thereby allows interpretations that would not ordinarily result.

The basic procedure is as follows :

1. The input data are normalized, so that each attribute falls within the same range. This step helps ensure that attributes with large domains will not dominate attributes with smaller domains.
2. PCA computes k orthonormal vectors that provide a basis for the normalized input data. These are unit vectors that each point in a direction perpendicular to the others. These vectors are referred to as the principal components. The input data are a linear combination of the principal components.
3. The principal components are sorted in order of decreasing "significance" or strength. The principal components essentially serve as a new set of axes for the data providing important information about variance.
4. Because the components are sorted in decreasing order of "significance," the data size can be reduced by eliminating the weaker components, that is, those with low vari- ance. Using the strongest principal components, it should be possible to reconstruct a good approximation of the original data.

PCA can be applied to ordered and unordered attributes, and can handle sparse data and skewed data. Multidimensional data of more than two dimensions can be han- dled by reducing the problem to two dimensions. Principal components may be used as inputs to multiple regression and cluster analysis. In comparison with wavelet trans- forms, PCA tends to be better at handling sparse data, whereas wavelet transforms are more suitable for data of high dimensionality.

Attribute Subset Selection

Data sets for analysis may contain hundreds of attributes, many of which may be irrel- evant to the mining task or redundant. **Attribute subset selection** reduces the data set size by removing irrelevant or redundant attributes (or dimensions). The goal of attribute subset selection is to find a minimum set of attributes such that the resulting probability distribution of the data classes is as close as possible to the original distribution obtained using all attributes. Mining on a reduced set of attributes has an additional benefit: It reduces the number of attributes appearing in the discovered patterns, helping to make the patterns easier to understand.

"How can we find a 'good' subset of the original attributes?" For n attributes, there are 2^n possible subsets. An exhaustive search for the optimal subset of attributes can be pro- hibitively expensive, especially as n and the number of data classes increase. Therefore, heuristic methods that explore a reduced search space are commonly used for attribute subset selection. These methods are typically **greedy** in that, while searching through attribute space, they always make what looks to be the best choice at the time. Their strategy is to make a locally optimal choice in the hope that this will lead to a globally optimal solution. Such greedy methods are effective in practice and may come close to estimating an optimal solution.

The "best" (and "worst") attributes are typically determined using tests of statistical significance, which assume that the attributes are independent of one another. Many other attribute evaluation measures can be used such as the information gain measure used in building decision trees for classification.

Forward selection	Backward elimination	Decision tree induction

Initial attribute set:
{*A*1, *A*2, *A*3, *A*4, *A*5, *A*6}
Initial reduced set:
{}
=> {*A*1}
=> {*A*1, *A*4}
=> Reduced attribute set:
{*A*1, *A*4, *A*6}
	Initial attribute set:
	{*A*1, *A*2, *A*3, *A*4, *A*5, *A*6}
	=> {*A*1, *A*3, *A*4, *A*5, *A*6}
	=> {*A*1, *A*4, *A*5, *A*6}
	=> Reduced attribute set:
	{*A*1, *A*4, *A*6}
		Initial attribute set:
		{*A*1, *A*2, *A*3, *A*4, *A*5, *A*6}
		*A*4?
		Y N
		*A*1? *A*6?
		Y N
		Class 1 Class 2
		Class 1 Class 2
		=> Reduced attribute set:
		{*A*1, *A*4, *A*6}

1. **Stepwise forward selection**: The procedure starts with an empty set of attributes as the reduced set. The best of the original attributes is determined and added to the reduced set. At each subsequent iteration or step, the best of the remaining original attributes is added to the set.

2. **Stepwise backward elimination**: The procedure 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 stepwise for- ward selection and backward elimination methods can be combined so that, at each step, the procedure selects the best attribute and removes the worst from among the remaining attributes.

4. **Decision tree induction**: Decision tree algorithms (e.g., ID3, C4.5, and CART) were originally intended for classification. Decision tree induction constructs a flowchart- like structure where each internal (nonleaf) node denotes a test on an attribute, each branch corresponds to an outcome of the test, and each external (leaf) node denotes a class prediction. At each node, the algorithm chooses the "best" attribute to partition the data into individual classes.

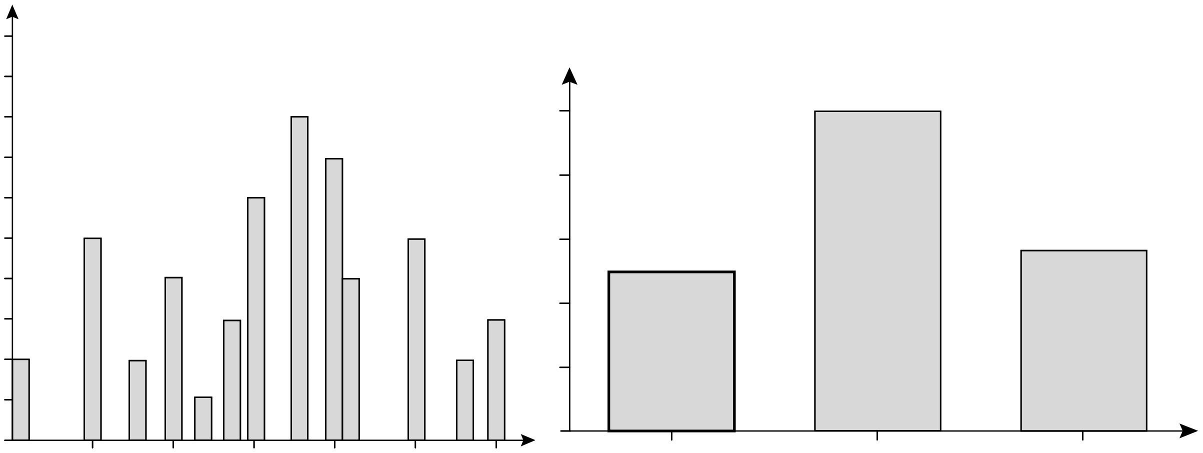
When decision tree induction is used for attribute subset selection, a tree is constructed from the given data. All attributes that do not appear in the tree are assumed to be irrelevant. The set of attributes appearing in the tree form the reduced subset of attributes.

The stopping criteria for the methods may vary. The procedure may employ a threshold on the measure used to determine when to stop the attribute selection process.

In some cases, we may want to create new attributes based on others. Such **attribute construction** can help improve accuracy and understanding of structure in high- dimensional data. For example, we may wish to add the attribute area based on the attributes height and width. By combining attributes, attribute construction can dis- cover missing information about the relationships between data attributes that can be useful for knowledge discovery.

Histograms

Histograms use binning to approximate data distributions and are a popular form of data reduction. A **histogram** for an attribute, A, partitions the data distribution of A into disjoint subsets, referred to as buckets or bins. If each bucket represents only a single attribute–value/frequency pair, the buckets are called singleton buckets. Often, buckets instead represent continuous ranges for the given attribute.



"How are the buckets determined and the attribute values partitioned?" There are several partitioning rules, including the following:

- **Equal-width**: In an equal-width histogram, the width of each bucket range is uniform.
- **Equal-frequency** (or equal-depth): In an equal-frequency histogram, the buckets are created so that, roughly, the frequency of each bucket is constant (i.e., each bucket contains roughly the same number of contiguous data samples).

Histograms are highly effective at approximating both sparse and dense data, as well as highly skewed and uniform data. The histograms described before for single attributes can be extended for multiple attributes. Multidimensional histograms can capture dependencies between attributes. These histograms have been found effective in approximating data with up to five attributes. More studies are needed regarding the effectiveness of multidimensional histograms for high dimensionalities. Singleton buckets are useful for storing high-frequency outliers.

Clustering

Clustering techniques consider data tuples as objects. They partition the objects into groups, or clusters, so that objects within a cluster are "similar" to one another and "dis- similar" to objects in other clusters. Similarity is commonly defined in terms of how "close" the objects are in space, based on a distance function. The "quality" of a cluster may be represented by its diameter, the maximum distance between any two objects in the cluster. **Centroid distance** is an alternative measure of cluster quality and is defined as the average distance of each cluster object from the cluster centroid (denoting the "average object," or average point in space for the cluster). Figure showed a 2-D plot of customer data with respect to customer locations in a city. Three data clusters are visible.

In data reduction, the cluster representations of the data are used to replace the actual data. The effectiveness of this technique depends on the data's nature. It is much more effective for data that can be organized into distinct clusters than for smeared data.

Sampling

Sampling can be used as a data reduction technique because it allows a large data set to be represented by a much smaller random data sample (or subset). Suppose that a large data set, D, contains N tuples. Let's look at the most common ways that we could sample D for data reduction.

- **Simple random sample without replacement (SRSWOR) of size** s: This is created by drawing s of the N tuples from D ($s < N$), where the probability of drawing any tuple in D is 1/N , that is, all tuples are equally likely to be sampled.

- **Simple random sample with replacement (SRSWR) of size**s : This is similar to SRSWOR, except that each time a tuple is drawn from D, it is recorded and then replaced. That is, after a tuple is drawn, it is placed back in D so that it may be drawn again.
- **Cluster sample**: If the tuples in D are grouped into M mutually disjoint "clusters," then an SRS of s clusters can be obtained, where s < M . For example, tuples in a database are usually retrieved a page at a time, so that each page can be considered a cluster. A reduced data representation can be obtained by applying, say, SRSWOR to the pages, resulting in a cluster sample of the tuples. Other clustering criteria conveying rich semantics can also be explored. For example, in a spatial database, we may choose to define clusters geographically based on how closely different areas are located.
- **Stratified sample**: If D is divided into mutually disjoint parts called strata, a stratified sample of D is generated by obtaining an SRS at each stratum. This helps ensure a representative sample, especially when the data are skewed. For example, a stratified sample may be obtained from customer data, where a stratum is created for each customer age group. In this way, the age group having the smallest number of customers will be sure to be represented.

1.11.1 Data Transformation and Data Discretization

Q18. Write about data Transformation and Data Discretization.

Ans :

In this preprocessing step, the data are transformed or consolidated so that the resulting mining process may be more efficient, and the patterns found may be easier to understand. Data discretization, a form of data transformation, is also discussed.

Data Transformation Strategies Overview

In data transformation, the data are transformed or consolidated into forms appropriate for mining. Strategies for data transformation include the following :

1. **Smoothing**, which works to remove noise from the data? Techniques include binning, regression, and clustering.
2. **Attribute construction** (or feature construction), where new attributes are constructured and added from the given set of attributes to help the mining process.
3. **Aggregation**, where summary or aggregation operations are applied to the data. For example, the daily sales data may be aggregated so as to compute monthly and annual total amounts. This step is typically used in constructing a data cube for data analysis at multiple abstraction levels.
4. **Normalization**, where the attribute data are scaled so as to fall within a smaller range, such as "1.0 to 1.0, or 0.0 to 1.0.
5. **Discretization**, where the raw values of a numeric attribute (e.g., age) are replaced by interval labels (e.g., 0–10, 11–20, etc.) or conceptual labels (e.g., youth, adult, senior). The labels, in turn, can be recursively organized into higher-level concepts, resulting in a concept hierarchy for the numeric attribute. Figure shows a concept hierarchy for the attribute price. More than one concept hierarchy can be defined for the same attribute to accommodate the needs of various users.
6. **Concept hierarchy generation for nominal data**, where attributes such as street can be generalized to higher-level concepts, like city or country. Many hierarchies for nominal attributes are implicit within the database schema and can be automatically defined at the schema definition level.

Smoothing is a form of data cleaning and was addressed on the data cleaning process also discussed ETL tools, where users specify transformations to correct data inconsistencies. Attribute construction and aggregation were discussed on data reduction.

Discretization techniques can be categorized based on how the discretization is per- formed, such as whether it uses class information or which direction it proceeds (i.e., top-down vs. bottom-up). If the discretization process uses class information, then we say it is supervised discretization. Otherwise, it is unsupervised. If the process starts by first finding one or a few points (called split points or cut points) to split the entire attribute range, and then repeats this recursively on the resulting intervals, it is called top-down discretization or splitting. This contrasts with bottom-up discretization or merging, which starts by considering all of the continuous values as potential split-points, removes some by merging neighborhood values to form intervals, and then recursively applies this process to the resulting intervals.

Data discretization and concept hierarchy generation are also forms of data reduction. The raw data are replaced by a smaller number of interval or concept labels. This simplifies the original data and makes the mining more efficient. The resulting patterns mined are typically easier to understand. Concept hierarchies are also useful for mining at multiple abstraction levels.

Data Transformation by Normalization

The measurement unit used can affect the data analysis. In general, expressing an attribute in smaller units will lead to a larger range for that attribute, and thus tend to give such an attribute greater effect or "weight." To help avoid dependence on the choice of measurement units, the data should be normalized or standardized. This involves transforming the data to fall within a smaller or common range such as [–1, 1] or [0.0, 1.0].

Normalizing the data attempts to give all attributes an equal weight. Normalization is particularly useful for classification algorithms involving neural networks or distance measurements such as nearest-neighbor classification and clustering. If using the neural network back propagation algorithm for classification mining, normalizing the input values for each attribute measured in the training tuples will help speed up the learning phase. For distance-based methods, normalization helps prevent attributes with initially large from outweighing attributes with initially smaller ranges (e.g., binary attributes). It is also useful when given no prior knowledge of the data.

There are many methods for data normalization. We study min-max normalization, z-score normalization, and normalization by decimal scaling. For our discussion, let A be a numeric attribute with n observed values, $v_1, v_2, \ldots, v_n$.

Min-max normalization performs a linear transformation on the original data. Suppose that min_A and max_A are the minimum and maximum values of an attribute, A. Min-max normalization maps a value, v_i, of A to v^0 in the range [new min_A, new max_A] by computing

$$\nu' = \frac{\nu - \text{Min}_A}{\text{Max}_A - \text{Min}_A}(\text{new\_max}_A - \text{new\_min}_A) + \text{new\_min}_A$$

(Or)

$$\nu' = (\nu'/v^0)$$

Min-max normalization preserves the relationships among the original data values. It will encounter an "out-of-bounds" error if a future input case for normalization falls outside of the original data range for A.

- **Discretization by Binning**

Binning is a top-down splitting technique based on a specified number of bins. These methods are also used as discretization methods for data reduction and concept hierarchy generation. For example, attribute values can be discretized by applying equal-width or equal-frequency binning, and then replacing each bin value by the bin mean or median, as in smoothing by bin means or smoothing by bin medians, respectively. These techniques can be applied recursively to the resulting partitions to generate concept hierarchies.

Binning does not use class information and is therefore an unsupervised discretization technique. It is sensitive to the user-specified number of bins, as well as the presence of outliers.

- **Discretization by Histogram Analysis**

Like binning, histogram analysis is an unsupervised discretization technique because it does not use class information. A histogram partitions the values of an attribute, A, into disjoint ranges called buckets or bins.

Various partitioning rules can be used to define histograms. In an equal-width histogram, for example, the values are partitioned into equal-size partitions or ranges. With an equal-frequency histogram, the values are partitioned so that, ideally, each partition contains the same number of data tuples. The histogram analysis algorithm can be applied recursively to each partition in order to automatically generate a multilevel concept hierarchy, with the procedure terminating once a pre-specified number of concept levels has been reached. A minimum interval size can also be used per level to control the recursive procedure. This specifies the minimum width of a partition, or the minimum number of values for each partition at each level. Histograms can also be partitioned based on cluster analysis of the data distribution, as described next.

- **Discretization by Cluster, Decision Tree, and Correlation Analyses**

Clustering, decision tree analysis, and correlation analysis can be used for data discartelization. We briefly study each of these approaches. Cluster analysis is a popular data discretization method. A clustering algorithm can be applied to discretize a numeric attribute, A, by partitioning the values of A into clusters or groups. Clustering takes the distribution of A into consideration, as well as the closeness of data points, and therefore is able to produce high-quality discretization results.

Clustering can be used to generate a concept hierarchy for A by following either a top-down splitting strategy or a bottom-up merging strategy, where each cluster forms a node of the concept hierarchy. In the former, each initial cluster or partition may be further decomposed into several sub-clusters, forming a lower level of the hierarchy. In the latter, clusters are formed by repeatedly grouping neighboring clusters in order to form higher-level concepts.

Techniques to generate decision trees for classification can be applied to discretization. Such techniques employ a top-down splitting approach. Unlike the other methods mentioned so far, decision tree approaches to discretization are supervised, that is, they make use of class label information. Intuitively, the main idea is to select split-points so that a given resulting partition contains as many tuples of the same class as possible. Entropy is the most commonly used measure for this purpose. To discretize a numeric attribute, A, the method selects the value of A that has the minimum entropy as a split-point, and recursively partitions the resulting intervals to arrive at a hierarchical discretization. Such discretization forms a concept hierarchy for A.

Because decision tree–based discretization uses class information, it is more likely that the interval boundaries (split-points) are defined to occur in places that may help improve classification accuracy.

Measures of correlation can be used for discretization. ChiMerge is a x^2 based discretization method. The discretization methods that we have studied up to this point have all employed a top-down, splitting strategy. This contrasts with ChiMerge, which employs a bottom-up approach by finding the best neighboring intervals and then merging them to form larger intervals, recursively. As with decision tree analysis, ChiMerge is supervised in that it uses class information. The basic notion is that for accurate discretization, the relative class frequencies should be fairly consistent within an interval. Therefore, if two adjacent intervals have a very similar distribution of classes, then the intervals can be merged. Otherwise, they should remain separate.

ChiMerge proceeds as follows. Initially, each distinct value of a numeric attribute A is considered to be one interval. x^2 tests are performed for every pair of adjacent intervals. Adjacent intervals with the least x^2 values are merged together, because low x^2 values for a pair indicate similar class distributions. This merging process proceeds recursively until a predefined stopping criterion is met.

- **Hierarchy Generation for Nominal Data**

Nominal attributes have a finite (but possibly large) number of distinct values, with no ordering among the values. Examples include geographic location, job category, and item type.

Manual definition of concept hierarchies can be a tedious and time-consuming task for a user or a domain expert. many hierarchies are implicit within the database schema and can be automatically defined at the schema definition level. The concept hierarchies can be used to transform the data into multiple levels of granularity. For example, data mining patterns regarding sales may be found relating to specific regions or countries, in addition to individual branch locations. The four methods for the generation of concept hierarchies for nominal data, as follows.

- o **Specification of a partial ordering of attributes explicitly at the schema level by users or experts:** Concept hierarchies for nominal attributes or dimensions typically involve a group of attributes. A user or expert can easily define a concept hierarchy by specifying a partial or total ordering of the attributes at the schema level.
- o **Specification of a portion of a hierarchy by explicit data grouping:** This is essen- tially the manual definition of a portion of a concept hierarchy. In a large database, it is unrealistic to define an entire concept hierarchy by explicit value enumera- tion. On the contrary, we can easily specify explicit groupings for a small portion of intermediate-level data.
- o **Specification of a set of attributes, but not of their partial ordering:** A user may specify a set of attributes forming a concept hierarchy, but omit to explicitly state their partial ordering. The system can then try to automatically generate the attribute ordering so as to construct a meaningful concept hierarchy.

- o “Without knowledge of data semantics, how can a hierarchical ordering for an arbitrary set of nominal attributes be found?” Consider the observation that since higher-level concepts generally cover several subordinate lower-level concepts, an attribute defining a high concept level (e.g., country) will usually contain a smaller number of distinct values than an attribute defining a lower concept level (e.g., street). Based on this observation, a concept hierarchy can be automatically gener- ated based on the number of distinct values per attribute in the given attribute set. The attribute with the most distinct values is placed at the lowest hierarchy level. The lower the number of distinct values an attribute has, the higher it is in the gener- ated concept hierarchy. This heuristic rule works well in many cases. Some local-level swapping or adjustments may be applied by users or experts, when necessary, after examination of the generated hierarchy.
- o **Specification of only a partial set of attributes:** Sometimes a user can be careless when defining a hierarchy, or have only a vague idea about what should be included in a hierarchy. Consequently, the user may have included only a small subset of the relevant attributes in the hierarchy specification.

Short Answers

1. Data Mining

Data mining refers to extracting or mining knowledge from large amounts of data. The term is actually a misnomer. Thus, data mining should have been more appropriately named as knowledge mining which emphasis on mining from large amounts of data.

It is the computational process of discovering patterns in large data sets involving methods at the intersection of artificial intelligence, machine learning, statistics, and database systems.

The overall goal of the data mining process is to extract information from a data set and transform it into an understandable structure for further use. The key properties of data mining are

- Automatic discovery of patterns
- Prediction of likely outcomes
- Creation of actionable information
- Focus on large datasets and databases

2. Scope of Data Mining

Data mining derives its name from the similarities between searching for valuable business information in a large database — for example, finding linked products in gigabytes of store Scanner data and mining a mountain for a vein of valuable ore. Both processes require either sifting through an immense amount of material, or intelligently probing it to find exactly where the value resides. Given databases of sufficient size and quality, data mining technology can generate new business opportunities by providing these capabilities :

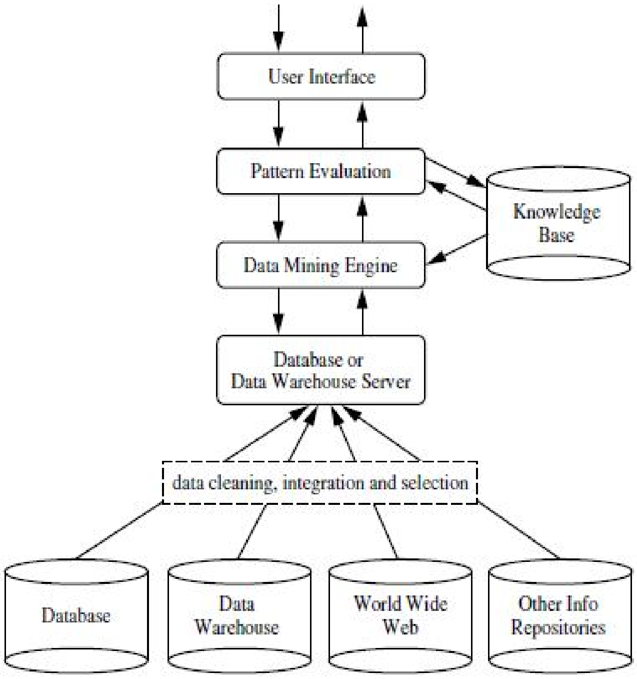
- **Automated prediction of trends and behaviors.** Data mining automates the process of finding predictive information in large databases. Questions that traditionally required extensive hands- on analysis can now be answered directly from the data — quickly. A typical example of a predictive problem is targeted marketing. Data mining uses data on past promotional mailings to identify the targets most likely to maximize return on investment in future mailings. Other predictive problems include forecasting bankruptcy and other forms of default, and identifying segments of a population likely to respond similarly to given events.
- **Automated discovery of previously unknown patterns.** Data mining tools sweep through databases and identify previously hidden patterns in one step. An example of pattern discovery is the analysis of retail sales data to identify seemingly unrelated products that are often purchased together. Other pattern discovery problems include detecting fraudulent credit card transactions and identifying anomalous data that could represent data entry keying errors.

3. Tasks of Data Mining

Data mining involves six common classes of tasks :

- **Anomaly detection (Outlier/change/deviation detection)** – The identification of unusual data records, that might be interesting or data errors that require further investigation.
- **Association rule learning (Dependency modelling)** – Searches for relationships between variables. For example a supermarket might gather data on customer purchasing habits. Using association rule learning, the supermarket can determine which products are frequently bought together and use this information for marketing purposes. This is sometimes referred to as market basket analysis.

- **Clustering –** is the task of discovering groups and structures in the data that are in some way or another "similar", without using known structures in the data.
- **Classification** – is the task of generalizing known structure to apply to new data. For example, an e-mail program might attempt to classify an e-mail as "legitimate" or as"spam".
- **Regression** – attempts to find a function which models the data with the least error.
- **Summarization** – providing a more compact representation of the data set, including visualization and report generation.



4. Architecture of Data Mining

A typical data mining system may have the following major components.

- **Knowledge Base:** This is the domain knowledge that is used to guide the search or evaluate the interestingness of resulting patterns. Such knowledge can include concept hierarchies, used to organize attributes or attribute values into different levels of abstraction. Knowledge such as user beliefs, which can be used to assess a pattern's interestingness based on its unexpectedness, may also be included. Other examples of domain knowledge are additional interestingness constraints or thresholds, and metadata (e.g., describing data from multiple heterogeneous sources).
- **Data Mining Engine:** This is essential to the data mining system and ideally consists of a set of functional modules for tasks such as characterization, association and correlation analysis, classification, prediction, cluster analysis, outlier analysis, and evolution analysis.
- **Pattern Evaluation Module:** This component typically employs interestingness measures interact with the data mining modules so as to focus the search toward interesting patterns. It may use interestingness thresholds to filter out discovered patterns. Alternatively, the pattern evaluation module may be integrated with the mining module, depending on the implementation of the data mining method used. For efficient data mining, it is highly recommended to push the evaluation of pattern interestingness as deep as possible into the mining process so as to confine the search to only the interesting patterns.

- **User interface:** This module communicates between users and the data mining system, allowing the user to interact with the system by specifying a data mining query or task, providing information to help focus the search, and performing exploratory data mining based on the intermediate data mining results. In addition, this component allows the user to browse database and data warehouse schemas or data structures, evaluate mined patterns, and visualize the patterns in different forms.

5. Process of Data Mining

Data Mining is a process of discovering various models, summaries, and derived values from a given collection of data. The general experimental procedure adapted to data-mining problems involves the following steps :

1. **State the problem and formulate the hypothesis:** Most data-based modeling studies are performed in a particular application domain. Hence, domain-specific knowledge and experience are usually necessary in order to come up with a meaningful problem statement. Unfortunately, many application studies tend to focus on the data-mining technique at the expense of a clear problem statement. In this step, a modeler usually specifies a set of variables for the unknown dependency and, if possible, a general form of this dependency as an initial hypothesis. There may be several hypotheses formulated for a single problem at this stage. The first step requires the combined expertise of an application domain and a data-mining model. In practice, it usually means a close interaction between the data-mining expert and the application expert. In successful data-mining applications, this cooperation does not stop in the initial phase; it continues during the entire data-mining process.

2. **Collect the data:** This step is concerned with how the data are generated and collected. In general, there are two distinct possibilities. The first is when the data-generation process is under the control of an expert (modeler): this approach is known as a designed experiment. The second possibility is when the expert cannot influence the data- generation process: this is known as the observational approach. An observational setting, namely, random data generation, is assumed in most data-mining applications.

3. Typically, the sampling distribution is completely unknown after data are collected, or it is partially and implicitly given in the data-collection procedure. It is very important, however, to understand how data collection affects its theoretical distribution, since such a priori knowledge can be very useful for modeling and, later, for the final interpretation of results. Also, it is important to make sure that the data used for estimating a model and the data used later for testing and applying a model come from the same, unknown, sampling distribution. If this is not the case, the estimated model cannot be successfully used in a final application of the results.

4. **Reprocessing the data:** In the observational setting, data are usually "collected" from the existing data base, data warehouses, and data marts. Data preprocessing usually includes at least two common tasks:

   I. **Detection (and removal)** – Outliers are unusual data values that are not consistent with most observations. Commonly, outliers result from measurement errors, coding and recording errors, and, sometimes, are natural, abnormal values. Such non-representative samples can seriously affect the model produced later. There are two strategies for dealing with outliers:

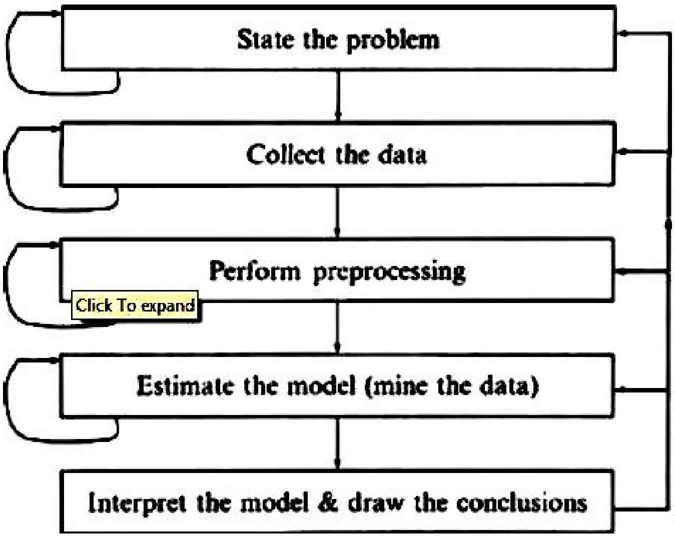
      - Detect and eventually remove outliers as a part of the preprocessing phase, or
      - Develop robust modeling methods that are insensitive to outliers.

II. **Scaling, encoding, and selecting features** – Data preprocessing includes several steps such as variable scaling and different types of encoding. For example, one feature with the range [0, 1] and the other with the range [“100, 1000] will not have the same weights in the applied technique; they will also influence the final data-mining results differently. Therefore, it is recommended to scale them and bring both features to the same weight for further analysis. Also, application-specific encoding methods usually achieve dimensionality reduction by providing a smaller number of informative features for subsequent data modeling.

These two classes of preprocessing tasks are only illustrative examples of a large spectrum of preprocessing activities in a data-mining process. Data-preprocessing steps should not be considered completely independent from other data-mining phases. In every iteration of the data-mining process, all activities, together, could define new and improved data sets for subsequent iterations. Generally, a good preprocessing method provides an optimal representation for a data-mining technique by incorporating a priori knowledge in the form of application-specific scaling and encoding.

III. **Estimate the model:** The selection and implementation of the appropriate data-mining technique is the main task in this phase. This process is not straightforward; usually, in practice, the implementation is based on several models, and selecting the best one is an additional task. The basic principles of learning and discovery from data are given in Chapter 4 of this book. Later, Chapter 5 through 13 explain and analyze specific techniques that are applied to perform a successful learning process from data and to develop an appropriate model.

IV. **Interpret the model and draw conclusions:** In most cases, data-mining models should help in decision making. Hence, such models need to be interpretable in order to be useful because humans are not likely to base their decisions on complex “black-box” models. Note that the goals of accuracy of the model and accuracy of its interpretation are somewhat contradictory. Usually, simple models are more interpretable, but they are also less accurate. Modern data-mining methods are expected to yield highly accurate results using high dimensional models. The problem of interpreting these models, also very important, is considered a separate task, with specific techniques to validate the results. A user does not want hundreds of pages of numeric results. He does not understand them; he cannot summarize, interpret, and use them for successful decision making.


State the problem
Collect the data
Perform preprocessing
Click To expand
Estimate the model (mine the data)
Interpret the model & draw the conclusions

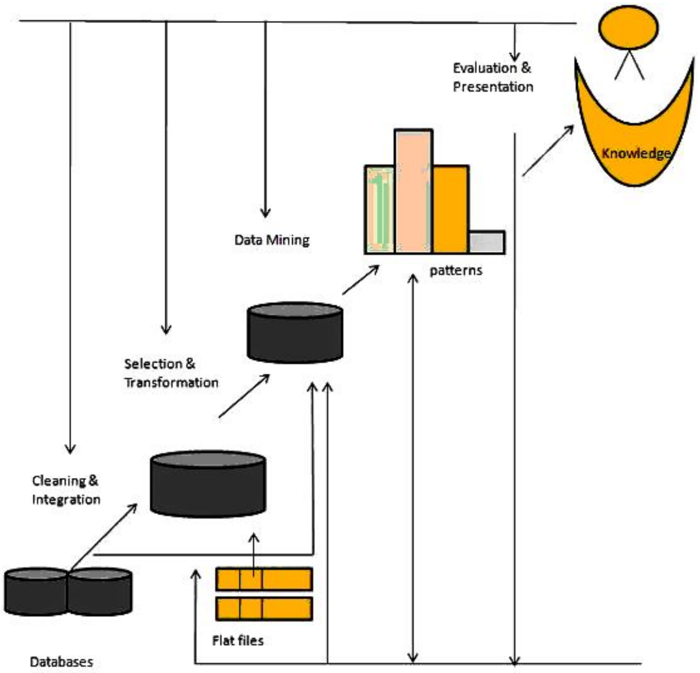
6. Issues related to Data Mining

- **Mining different kinds of knowledge in databases.** - The need of different users is not the same. And Different user may be in interested in different kind of knowledge. Therefore it is necessary for data mining to cover broad range of knowledge discovery task.
- **Interactive mining of knowledge at multiple levels of abstraction.** - The data mining process needs to be interactive because it allows users to focus the search for patterns, providing and refining data mining requests based on returned results.
- **Incorporation of background knowledge.** - To guide discovery process and to express the discovered patterns, the background knowledge can be used. Background knowledge may be used to express the discovered patterns not only in concise terms but at multiple level of abstraction.
- **Data mining query languages and ad hoc data mining.** - Data Mining Query language that allows the user to describe ad hoc mining tasks, should be integrated with a data warehouse query language and optimized for efficient and flexible data mining.
- **Presentation and visualization of data mining results.** - Once the patterns are discovered it needs to be expressed in high level languages, visual representations. This representations should be easily understandable by the users.
- **Handling noisy or incomplete data.** - The data cleaning methods are required that can handle the noise, incomplete objects while mining the data regularities. If data cleaning methods are not there then the accuracy of the discovered patterns will be poor.
- **Pattern evaluation.** - It refers to interestingness of the problem. The patterns discovered shouldbe interesting because either they represent common knowledge or lack novelty.
- **Efficiency and scalability of data mining algorithms.** - In order to effectively extract the information from huge amount of data in databases, data mining algorithm must be efficient and scalable.
- **Parallel, distributed, and incremental mining algorithms.** - The factors such as huge size of databases, wide distribution of data, and complexity of data mining methods motivate the development of parallel and distributed data mining algorithms. These algorithms divide the data into partitions which is further processed parallel. Then the results from the partitions are merged. The incremental algorithms, updates databases without having mine the data again from scratch.

7. Knowledge Discovery in Database

Knowledge Discovery in Databases (KDD): Some people treat data mining same as Knowledge discovery while some people view data mining essential step in process of knowledge discovery. Here is the list of steps involved in knowledge discovery process :

- **Data Cleaning** - In this step the noise and inconsistent data is removed.
- **Data Integration** - In this step multiple data sources are combined.
- **Data Selection** - In this step relevant to the analysis task are retrieved from the database.
- **Data Transformation** - In this step data are transformed or consolidated into forms appropriate for mining by performing summary or aggregation operations.
- **Data Mining** - In this step intelligent methods are applied in order to extract data patterns.
- **Pattern Evaluation** - In this step, data patterns are evaluated.
- **Knowledge Presentation** - In this step, knowledge is represented.



8. Data Warehouse

A data warehouse is a subject-oriented, integrated, time-variant and non-volatile collection of data in support of management's decision making process.

- **Subject-Oriented**: A data warehouse can be used to analyze a particular subject area. For example, "sales" can be a particular subject.
- **Integrated**: A data warehouse integrates data from multiple data sources. For example, source A and source B may have different ways of identifying a product, but in a data warehouse, there will be only a single way of identifying a product.
- **Time-Variant**: Historical data is kept in a data warehouse. For example, one can retrieve data from 3 months, 6 months, 12 months, or even older data from a data warehouse. This contrasts with a transactions system, where often only the most recent data is kept. For example, a transaction system may hold the most recent address of a customer, where a data warehouse can hold all addresses associated with a customer.
- **Non-volatile**: Once data is in the data warehouse, it will not change. So, historical data in a data warehouse should never be altered.

9. Issues in Data Integration

1. **Schema integration and object matching:** How can the data analyst or the computer be sure that customer id in one database and customer number in another reference to the same attribute.
2. **Redundancy:** An attribute (such as annual revenue, for instance) may be redundant if it can be derived from another attribute or set of attributes. Inconsistencies in attribute or dimension naming can also cause redundancies in the resulting data set.
3. **Detection and resolution of data value conflicts:** For the same real-world entity, attribute values from different sources may differ.

4. **Data Transformation:** In data transformation, the data are transformed or consolidated into forms appropriate for mining.

Data transformation can involve the following :

- **Smoothing**, which works to remove noise from the data? Such techniques include binning, regression, and clustering.
- **Aggregation**, where summary or aggregation operations are applied to the data. For example, the daily sales data may be aggregated so as to compute monthly and annual total amounts. This step is typically used in constructing a data cube for analysis of the data at multiple granularities.
- **Generalization of the data**, where low-level or ¯primitive (raw) data are replaced
- By higher-level concepts through the use of concept hierarchies. For example, categorical attributes, like street, can be generalized to higher-level concepts, like city or country.
- **Normalization**, where the attribute data are scaled so as to fall within a small specified range, such as 1:0 to 1:0, or 0:0 to 1:0.
- **Attribute construction** (or feature construction), where new attributes are constructed and added from the given set of attributes to help the mining process.

10. Strategies of Data Reduction

Data reduction techniques can be applied to obtain a reduced representation of the data set that is much smaller in volume, yet closely maintains the integrity of the original data. That is, mining on the reduced data set should be more efficient yet produce the same (or almost the same) analytical results. Strategies for data reduction include the following:

- **Data cube aggregation**, where aggregation operations are applied to the data in the construction of a data cube.
- **Attribute subset selection**, where irrelevant, weakly relevant, or redundant attributes or dimensions may be detected and removed.
- **Dimensionality reduction**, where encoding mechanisms are used to reduce the dataset size.
- **Numerosity reduction**, where the data are replaced or estimated by alternative, smaller data representations such as parametric models (which need store only the model parameters instead of the actual data) or nonparametric methods such as clustering, sampling, and the use of histograms.
- **Discretization and concept hierarchy generation**, where rawdata values for attributes are replaced by ranges or higher conceptual levels. Data discretization is a form of numerosity reduction that is very useful for the automatic generation of concept hierarchies. Discretization and concept hierarchy generation are powerful tools for data mining, in that they allow the mining of data at multiple levels of abstraction.

UNIT II

Data Warehousing and Online Analytical Processing (OLAP): Basic concepts of data warehouse, data warehouse modelling–data cube and OLAP, data warehouse design and usage, data warehouse implementation, data generalization by attribute–oriented.Data Cube Technology: data cube computation preliminary concepts, data cube computation methods, processing advanced kinds of queries by exploring cube technology, multidimensional data analysis in cube space. Mining Frequent Patterns, Associations, and Correlations: basic concepts, frequent itemset mining methods, mining various kinds of association rules, from association mining to correlation analysis, constraint-based association mining.

2.1 Data Warehouse Introduction

Q1. What is Data Warehouse ?

Ans :

A data warehouse is a collection of data marts representing historical data from different operations in the company. This data is stored in a structure optimized for querying and data analysis as a data warehouse. Table design, dimensions and organization should be consistent throughout a data warehouse so that reports or queries across the data warehouse are consistent. A data warehouse can also be viewed as a database for historical data from different functions within a company.

The term Data Warehouse was coined by Bill Inmon in 1990, which he defined 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. It is a single, complete and consistent store of data obtained from a variety of different sources made available to end users in what they can understand and use in a business context. It can be

- Used for decision Support
- Used to manage and control business
- Used by managers and end-users to understand the business and make judgments.
- Data Warehousing is an architectural construct of information systems that provides users with current and historical decision support information that is hard to access or present in traditional operational data stores.

Other Important Terminology

- **Enterprise Data warehouse :** It collects all information about subjects (customers, products, sales, assets, personnel) that span the entire organization.
- **Data Mart:** Departmental subsets that focus on selected subjects. A data mart is a segment of a data warehouse that can provide data for reporting and analysis on a section, unit, department or operation in the company, e.g. sales, payroll, production. Data marts are sometimes complete individual data warehouses which are usually smaller than the corporate data warehouse.
- **Decision Support System (DSS) :** Information technology to help the knowledge worker (executive, manager, and analyst) makes faster & better decisions.
- **Drill-down:** Traversing the summarization levels from highly summarized data to the underlying current or old detail.

- **Metadata:** Data about data. Containing location and description of warehouse system components: names, definition, structure, etc.

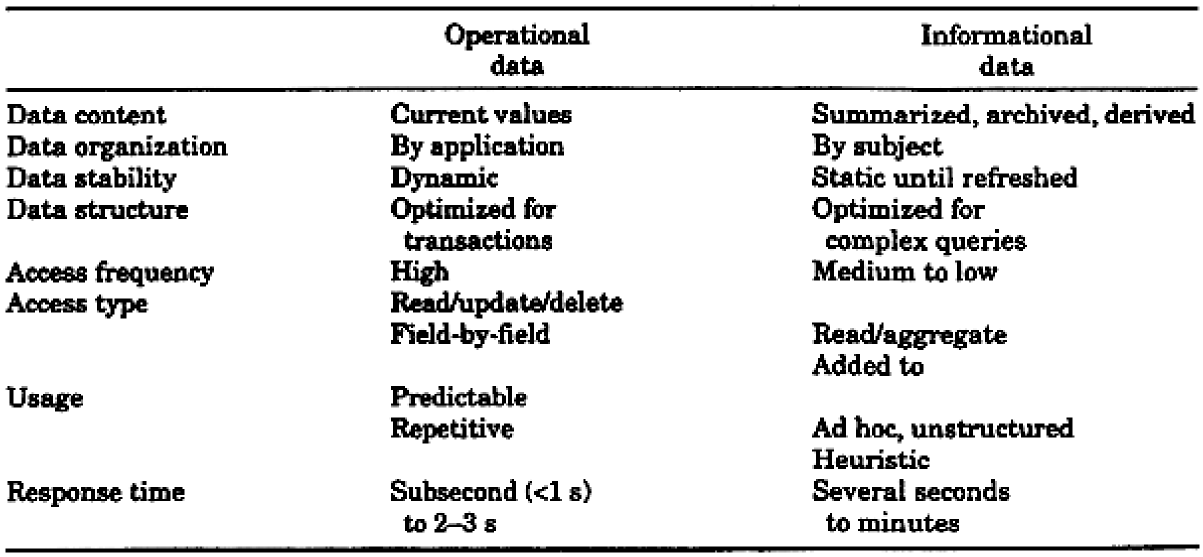
Benefits of Data Warehousing

- Data warehouses are designed to perform well with aggregate queries running on large amounts of data.
- The structure of data warehouses is easier for end users to navigate, understand and query against unlike the relational databases primarily designed to handle lots of transactions.
- Data warehouses enable queries that cut across different segments of a company's operation. E.g. production data could be compared against inventory data even if they were originally stored in different databases with different structures.
- Queries that would be complex in very normalized databases could be easier to build and maintain in data warehouses, decreasing the workload on transaction systems.
- Data warehousing is an efficient way to manage and report on data that is from a variety of sources, non-uniform and scattered throughout a company.
- Data warehousing is an efficient way to manage demand for lots of information from lots of users.
- Data warehousing provides the capability to analyze large amounts of historical data for nuggets of wisdom that can provide an organization with competitive advantage.

Operational and Informational Data

1. **Operational Data:**
   - Focusing on transactional function such as bank card withdrawals and deposits.
   - Detailed
   - Updateable
   - Reflects current data
2. **Informational Data:**
   - Focusing on providing answers to problems posed by decision makers
   - Summarized
   - Non updateable

These differences beween the informational and operational databases are summarized in the following table :



	Operational data	Informational data
Data content	Current values	Summarized, archived, derived
Data organization	By application	By subject
Data stability	Dynamic	Static until refreshed
Data structure	Optimized for transactions	Optimized for complex queries
Access frequency	High	Medium to low
Access type	Read/update/delete Field-by-field	Read/aggregate Added to
Usage	Predictable Repetitive	Ad hoc, unstructured Heuristic
Response time	Subsecond (<1 s) to 2–3 s	Several seconds to minutes

Data Warehouse Characteristics

- A data warehouse can be viewed as an information system with the following attributes: – It is a database designed for analytical tasks
  - It's content is periodically updated
  - It contains current and historical data to provide a historical perspective of information

Operational Data Store (ODS)

- ODS is an architecture concept to support day-to-day operational decision support and contains current value data propagated from operational applications
- DS is subject-oriented, similar to a classic definition of a Data warehouse
- ODS is integrated.

Volatile	Non volatile
Very current data	Current and historical data
Detailed data	Pre calculated summaries

2.1.1 Differences between Operational Database Systems and Data Warehouses

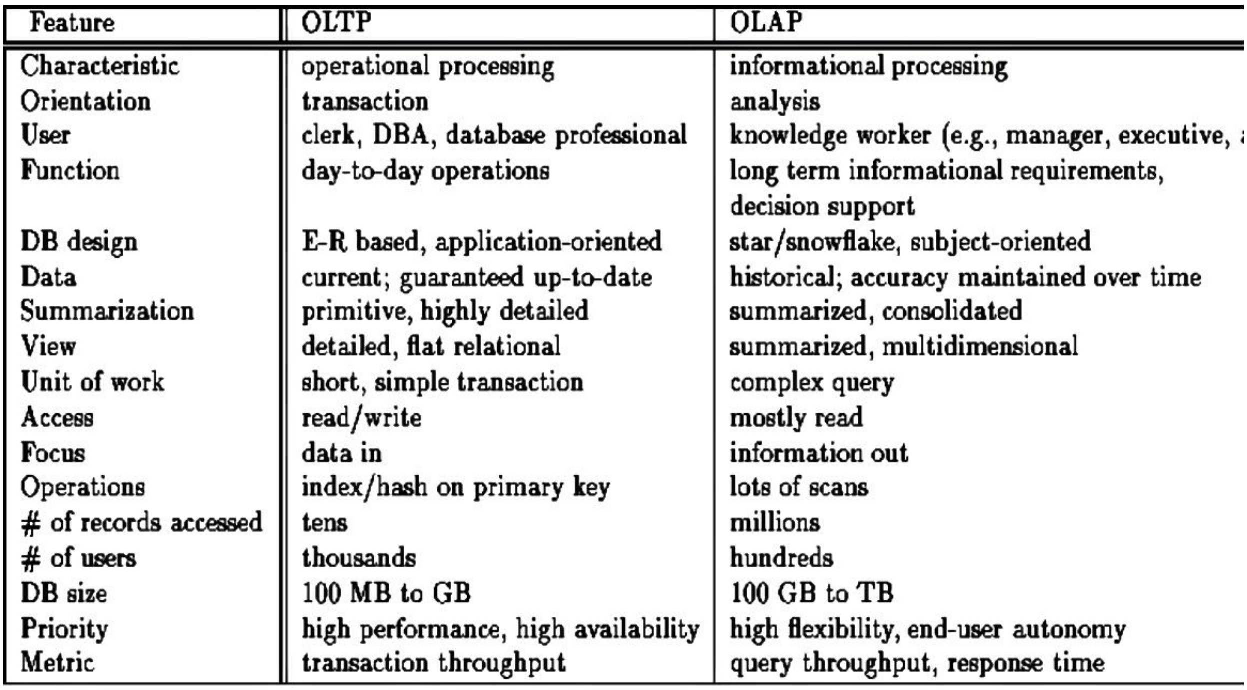
Q2. Explain Operational Database Systems vs. Data Warehouses.

Ans :

Features of OLTP and OLAP

The major distinguishing features between OLTP and OLAP are summarized as follows.

1. **Users and system orientation**: An OLTP system is customer-oriented and is used for transaction and query processing by clerks, clients, and information technology professionals. An OLAP system is market-oriented and is used for data analysis by knowledge workers, including managers, executives, and analysts.
2. **Data contents**: An OLTP system manages current data that, typically, are too detailed to be easily used for decision making. An OLAP system manages large amounts of historical data, provides facilities for summarization and aggregation, and stores and manages information at different levels of granularity. These features make the data easier for use in informed decision making.
3. **Database design**: An OLTP system usually adopts an entity-relationship (ER) data model and an application oriented database design. An OLAP system typically adopts either a star or snowflake model and a subject-oriented database design.
4. **View**: An OLTP system focuses mainly on the current data within an enterprise or department, without referring to historical data or data in different organizations. In contrast, an OLAP system often spans multiple versions of a database schema. OLAP systems also deal with information that originates from different organizations, integrating information from many data stores. Because of their huge volume, OLAP data are stored on multiple storage media.
5. **Access patterns**: The access patterns of an OLTP system consist mainly of short, atomic transactions. Such a system requires concurrency control and recovery mechanisms. However, accesses to OLAP systems are mostly read-only operations although many could be complex queries.



Feature	OLTP	OLAP
Characteristic	operational processing	informational processing
Orientation	transaction	analysis
User	clerk, DBA, database professional	knowledge worker (e.g., manager, executive,
Function	day-to-day operations	long term informational requirements, decision support
DB design	E-R based, application-oriented	star/snowflake, subject-oriented
Data	current; guaranteed up-to-date	historical; accuracy maintained over time
Summarization	primitive, highly detailed	summarized, consolidated
View	detailed, flat relational	summarized, multidimensional
Unit of work	short, simple transaction	complex query
Access	read/write	mostly read
Focus	data in	information out
Operations	index/hash on primary key	lots of scans
# of records accessed	tens	millions
# of users	thousands	hundreds
DB size	100 MB to GB	100 GB to TB
Priority	high performance, high availability	high flexibility, end-user autonomy
Metric	transaction throughput	query throughput, response time

Data Warehouse Models

There are three data warehouse models.

Enterprise warehouse

- An enterprise warehouse collects all of the information about subjects spanning the entire organization.
- It provides corporate-wide data integration, usually from one or more operational systems or external information providers, and is cross-functional in scope.
- It typically contains detailed data as well as summarized data, and can range in size from a few gigabytes to hundreds of gigabytes, terabytes, or beyond.
- An enterprise data warehouse may be implemented on traditional mainframes, computer super servers, or parallel architecture platforms. It requires extensive business modeling and may take years to design and build.

Data Mart

- A data mart contains a subset of corporate-wide data that is of value to a specific group of users. The scope is confined to specific selected subjects. For example, a marketing data mart may confine its subjects to customer, item, and sales. The data contained in datamarts tend to be summarized.
- Data marts are usually implemented on low-cost departmental servers that are UNIX/LINUX- or Windows-based. The implementation cycle of a data mart is more likely to be measured in weeks rather than months or years. However, it may involve complex integration in the long run if its design and planning were not enterprise-wide.
- Depending on the source of data, data marts can be categorized as independent or dependent. Independent data marts are sourced from data captured from one or more operational systems or external information providers, or from data generated locally within a particular department or geographic area. Dependent data marts are sourced directly from enterprise data warehouses.

Virtual Warehouse

A virtual warehouse is a set of views over operational databases. For efficient query processing, only some of the possible summary views may be materialized.

A virtual warehouse is easy to build but requires excess capacity on operational database servers.

Meta Data Repository

Metadata are data about data. When used in a data warehouse, metadata are the data that define warehouse objects. Metadata are created for the data names and definitions of the given warehouse. Additional metadata are created and captured for times tamping any extracted data, the source of the extracted data, and missing fields that have been added by data cleaning or integration processes. A metadata repository should contain the following :

- A description of the structure of the data warehouse, which includes the warehouse schema, view, dimensions, hierarchies, and derived data definitions, as well as data mart locations and contents.
- Operational metadata, which include data lineage (history of migrated data and the sequence of transformations applied to it), currency of data (active, archived, or purged), and monitoring information (warehouse usage statistics, error reports, and audit trails).
- The algorithms used for summarization, which include measure and dimension definition algorithms, data on granularity, partitions, subject areas, aggregation,
- Summarization, and predefined queries and reports.
- The mapping from the operational environment to the data warehouse, which includes source databases and their contents, gateway descriptions, data partitions, data extraction, cleaning, transformation rules and defaults, data refresh and purging rules, and security (user authorization and access control).
- Data related to system performance, which include indices and profiles that improve data access and retrieval performance, in addition to rules for the timing and scheduling of refresh, update, and replication cycles.
- Business metadata, which include business terms and definitions, data ownership information, and charging policies.

Data Warehouse Design Process

A data warehouse can be built using a top-down approach, a bottom-up approach, or a combination of both.

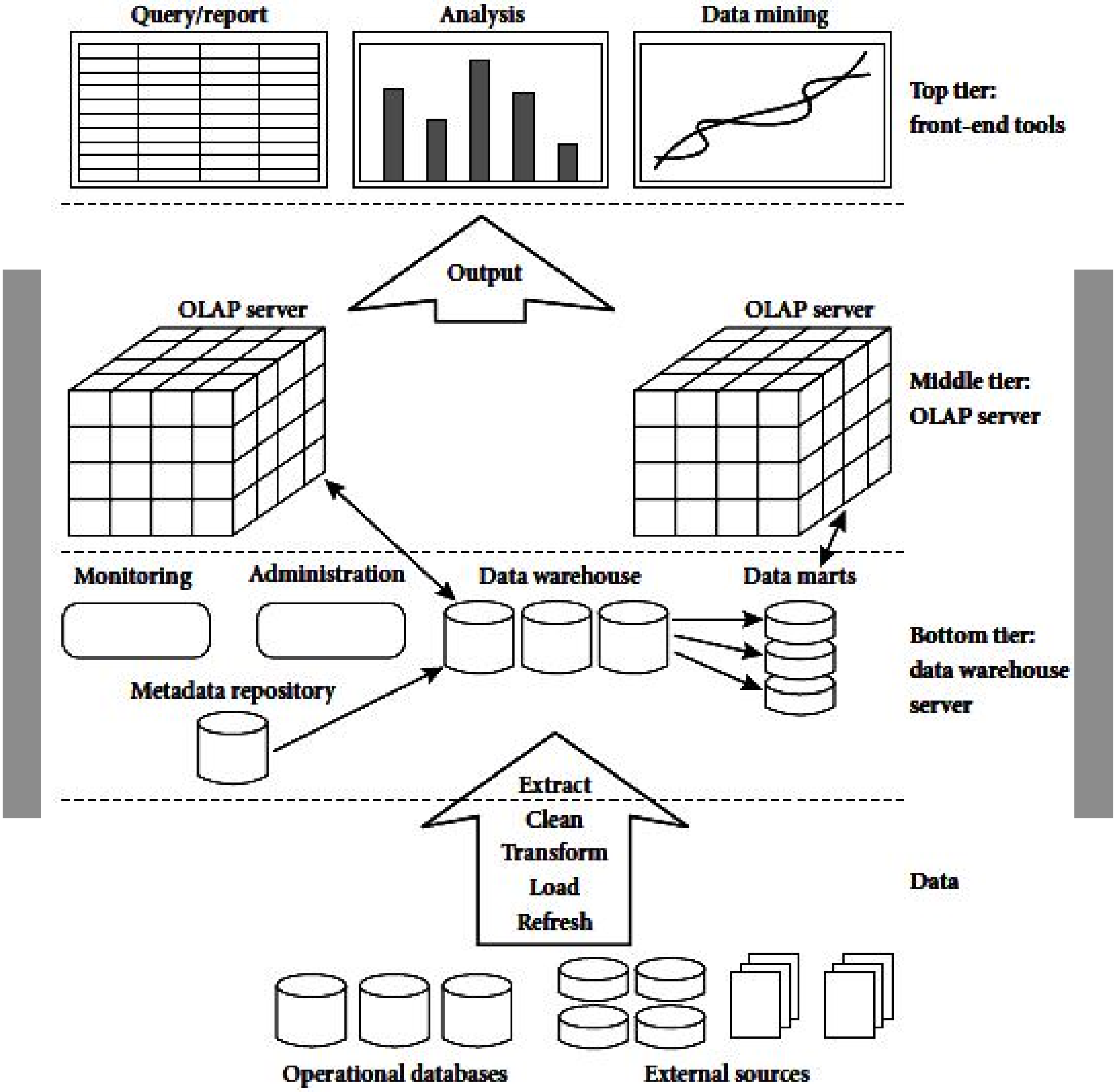
- The top-down approach starts with the overall design and planning. It is useful in cases where the technology is mature and well known, and where the business problems that must be solved are clear and well understood.
- The bottom-up approach starts with experiments and prototypes. This is useful in the early stage of business modeling and technology development. It allows an organization to move forward at considerably less expense and to evaluate the benefits of the technology before making significant commitments.
- In the combined approach, an organization can exploit the planned and strategic nature of the top-down approach while retaining the rapid implementation and opportunistic application of the bottom-up approach.

  The warehouse design process consists of the following steps :

  - Choose a business process to model, for example, orders, invoices, shipments, inventory, account administration, sales, or the general ledger. If the business process is organizational and involves multiple complex object collections, a data warehouse model should be followed. However, if the process is departmental and focuses on the analysis of one kind of business process, a data mart model should be chosen.
  - **Choose the grain of the business process.** The grain is the fundamental, atomic level of data to be represented in the fact table for this process, for example, individual transactions, individual daily snapshots, and so on.

- **Choose the dimensions that will apply to each fact table record.** Typical dimensions are time, item, customer, supplier, warehouse, transaction type, and status.
- **Choose the measures that will populate each fact table recor**d. Typical measures are numeric additive quantities like dollars sold and units sold.

A Three Tier Data Warehouse Architecture :



Tier-1

The bottom tier is a warehouse database server that is almost always a relational database system. Back-end tools and utilities are used to feed data into the bottom tier from operational databases or other external sources (such as customer profile information provided by external consultants). These tools and utilities perform data extraction, cleaning, and transformation (e.g., to merge similar data from different sources into a unified format), as well as load and refresh functions to update the data ware house. The data are extracted using application program interfaces known as gateways. A gateway is supported by the underlying DBMS and allows client programs to generate SQL code to be executed at a server.

Examples of gateways include ODBC (Open Database Connection) and OLEDB (Open Linking and Embedding for Databases) by Microsoft and JDBC (Java Database Connection).

This tier also contains a metadata repository, which stores information about the data warehouse and its contents.

Tier-2

The middle tier is an OLAP server that is typically implemented using either a relational OLAP (ROLAP) model or a multidimensional OLAP.

- OLAP model is an extended relational DBMS thatmaps operations on multidimensional data to standard relational operations.
- A multidimensional OLAP (MOLAP) model, that is, a special-purpose server that directly implements multidimensional data and operations.

Tier-3

The top tier is a front-end client layer, which contains query and reporting tools, analysis tools, and/ or data mining tools (e.g., trend analysis, prediction, and so on).

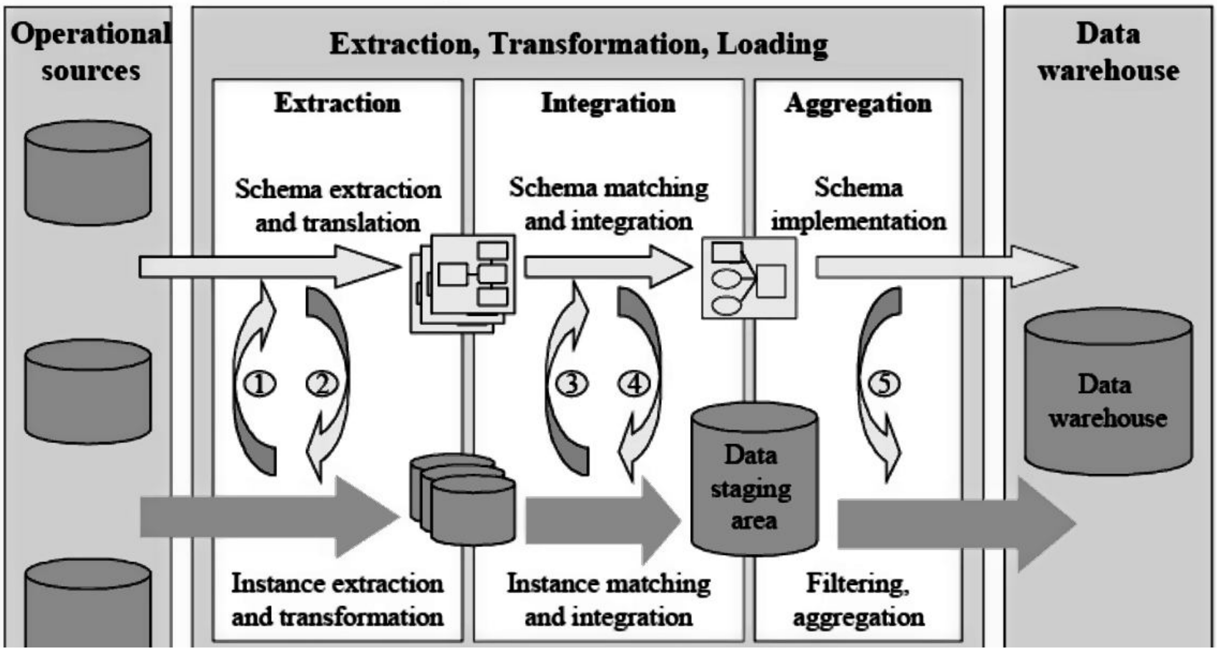
2.1.2 Data warehouse Back-End Tools and Utilities

Q3. Explain about Back-End Tools and Utilities in Data warehouse ?

Ans :

The ETL (Extract Transformation Load) Process

In this section we will discussed about the 4 major process of the data warehouse. They are extract (data from the operational systems and bring it to the data warehouse), transform (the data into internal format and structure of the data warehouse), cleanse (to make sure it is of sufficient quality to be used for decision making) and load (cleanse data is put into the data warehouse).


Operational sources
Extraction, Transformation, Loading
Data warehouse
Extraction
Integration
Aggregation
Schema extraction and translation
Schema matching and integration
Schema implementation
①
②
③
④
⑤
Data staging area
Data warehouse
Instance extraction and transformation
Instance matching and integration
Filtering, aggregation

The four processes from extraction through loading often referred collectively as **Data Staging**.

Extract

Some of the data elements in the operational database can be reasonably being expected to be useful in the decision making, but others are of less value for that purpose. For this reason, it is necessary to extract the relevant data from the operational database before bringing into the data warehouse. Many commercial tools are available to help with the extraction process.

Data Junction is one of the commercial products. The user of one of these tools typically has an easy-to-use windowed interface by which to specify the following :

- Which files and tables are to be accessed in the source database?
- Which fields are to be extracted from them? This is often done internally by SQL Select statement.
- What are those to be called in the resulting database?
- What is the target machine and database format of the output?
- On what schedule should the extraction process be repeated?

Transform

The operational databases developed can be based on any set of priorities, which keeps changing with the requirements. Therefore those who develop data warehouse based on these databases are typically faced with inconsistency among their data sources. Transformation process deals with rectifying any inconsistency (if any).

One of the most common transformation issues is 'Attribute Naming Inconsistency'. It is common for the given data element to be referred to by different data names in different databases. Employee Name may be EMP_NAME in one database, ENAME in the other. Thus one set of Data Names are picked and used consistently in the data warehouse. Once all the data elements have right names, they must be converted to common formats. The conversion may encompass the following:

I. Characters must be converted ASCII to EBCDIC or vise versa.

II. Mixed Text may be converted to all uppercase for consistency.

III. Numerical data must be converted in to a common format.

IV. Data Format has to be standardized.

V. Measurement may have to convert. (Rs/ $)

VI. Coded data (Male/Female, M/F) must be converted into a common format.

All these transformation activities are automated and many commercial products are Available to perform the tasks.

Data MAPPER from Applied Database Technologies is one such comprehensive tool.

Cleansing

Information quality is the key consideration in determining the value of the information. The developer of the data warehouse is not usually in a position to change the quality of its underlying historic data, though a data warehousing project can put spotlight on the data quality issues and lead to improvements for the future. It is, therefore, usually necessary to go through the data entered into the data warehouse and make it as error free as possible. This process is known as **Data Cleansing**.

Data Cleansing must deal with many types of possible errors. These include missing data and incorrect data at one source; inconsistent data and conflicting data when two or more source is involved. There are several algorithms followed to clean the data, which will be discussed in the coming lecture notes.

Loading

Loading often implies physical movement of the data from the computer(s) storing the source database(s) to that which will store the data warehouse database, assuming it is different. This takes place immediately after the extraction phase. The most common channel for data movement is a high-speed communication link. Ex: Oracle Warehouse Builder is the API from Oracle, which provides the features to perform the ETL task on Oracle Data Warehouse.

2.2 OLAP(Online Analytical Processing)

Q4. Explain in detail about OLAP ?

Ans :

- OLAP is an approach to answering multi-dimensional analytical (MDA) queries swiftly.
- OLAP is part of the broader category of business intelligence, which also encompasses
- relational database, report writing and data mining.
- OLAP tools enable users to analyze multidimensional data interactively from multiple perspectives.

OLAP consists of three basic analytical operations :

1. Consolidation (Roll-Up)
2. Drill-Down
3. Slicing And Dicing

- On solidation involves the aggregation of data that can be accumulated and computed in one or more dimensions. For example, all sales offices are rolled up to the sales department or sales division to anticipate sales trends.
- The drill-down is a technique that allows users to navigate through the details. For instance, users can view the sales by individual products that make up a region's sales.
- Slicing and dicing is a feature whereby users can take out (slicing) a specific set of data of the OLAP cube and view (dicing) the slices from different viewpoints.

Types of OLAP

1. Relational OLAP (ROLAP)

- ROLAP works directly with relational databases. The base data and the dimension tables are stored as relational tables and new tables are created to hold the aggregated information. It depends on a specialized schema design.
- 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.
- ROLAP tools do not use pre-calculated data cubes but instead pose the query to the standard relational database and its tables in order to bring back the data required to answer the question.
- ROLAP tools feature the ability to ask any question because the methodology does not limit to the contents of a cube. ROLAP also has the ability to drill down to the lowest level of detail in the database.

2. Multidimensional OLAP (MOLAP)

- MOLAP is the 'classic' form of OLAP and is sometimes referred to as just OLAP.
- MOLAP stores this data in optimized multi-dimensional array storage, rather than in a relational database. Therefore it requires the pre-computation and storage of information in the cube - the operation known as processing.
- MOLAP tools generally utilize a pre-calculated data set referred to as a data cube. The data cube contains all the possible answers to a given range of questions.
- MOLAP tools have a very fast response time and the ability to quickly write back data into the data set.

3. Hybrid OLAP (HOLAP)

- There is no clear agreement across the industry as to what constitutes Hybrid OLAP, except that a database will divide data between relational and specialized storage.

- For example, for some vendors, a HOLAP database will use relational tables to hold the larger quantities of detailed data, and use specialized storage for at least some aspects of the smaller quantities of more-aggregate or less-detailed data.
- HOLAP addresses the shortcomings of MOLAP and ROLAP by combining the capabilities of both approaches.
- HOLAP tools can utilize both pre-calculated cubes and relational data sources.

2.2.1 Data Warehouse Models- Enterprise Warehouse, Data Mart, and Virtual Warehouse

Q5. Explain about Data Warehouse Models ?

Ans :

There are three data warehouse models :

1. enterprise warehouse,
2. data mart, and
3. virtual warehouse.

1. Enterprise warehouse

An enterprise warehouse collects all of the information about subjects spanning the entire organization. It provides corporate-wide data integration, usually from one or more operational systems or external information providers, and is cross-functional in scope. It typically contains detailed data as well as summarized data, and can range in size from a few gigabytes to hundreds of gigabytes, terabytes, or beyond. An enterprise data warehouse may be implemented on traditional mainframes, computer superservers, or parallel architecture platforms. It requires extensive business modeling and may take years to design and build.

2. Data mart

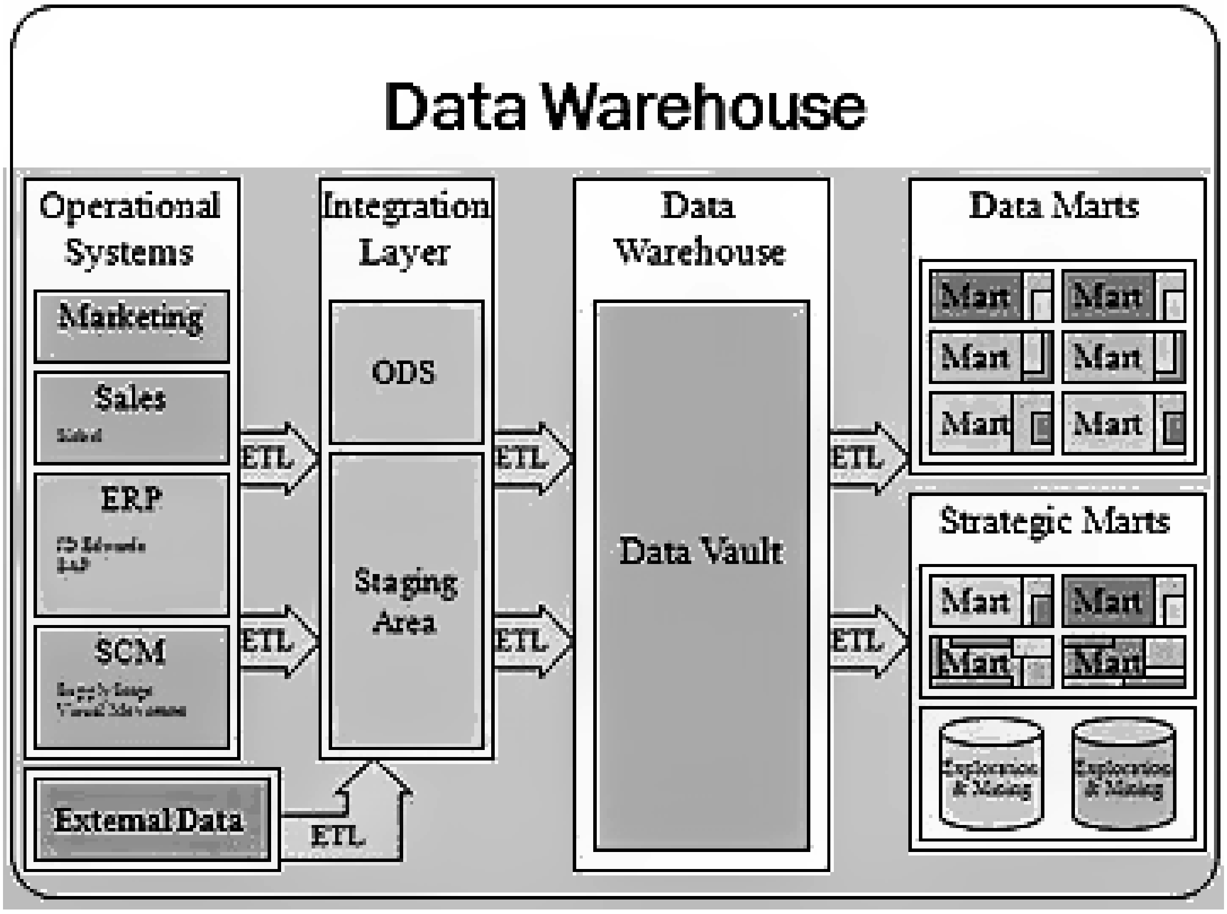
A data mart contains a subset of corporate-wide data that is of value to a specific group of users. The scope is confined to specific selected subjects. For example, a marketing data mart may confine its subjects to customer, item, and sales. The data contained in data marts tend to be summarized.

Data marts are usually implemented on low-cost departmental servers that are Unix/Linux or Windows based. The implementation cycle of a data mart is more likely to be measured in weeks rather than months or years. However, it may involve complex integration in the long run if its design and planning were not enterprise-wide.

Depending on the source of data, data marts can be categorized as independent or dependent. Independent data marts are sourced from data captured from one or more operational systems or external information providers, or from data generated locally within a particular department or geographic area. Dependent data marts are sourced directly from enterprise data warehouses.

3. Virtual warehouse

A virtual warehouse is a set of views over operational databases. For efficient query processing, only some of the possible summary views may be materialized. A virtual warehouse is easy to build but requires excess capacity on operational database servers.



"What are the pros and cons of the top-down and bottom-up approaches to data warehouse development ?"

The top-down development of an enterprise warehouse serves as a systematic solution and minimizes integration problems. However, it is expensive, takes a long time to develop, and lacks flexibility due to the difficulty in achieving consistency and consensus for a common data model for the entire organization. The bottom up approach to the design, development, and deployment of independent data marts provides flexibility, low cost, and rapid return of investment. It, however, can lead to problems when integrating various disparate data marts into a consistent enterprise data warehouse.

A recommended method for the development of data warehouse systems is to implement the warehouse in an incremental and evolutionary manner

I. A high-level corporate data model is defined within a reasonably short period (such as one or two months) that provides a corporate-wide, consistent, integrated view of data among different subjects and potential usages. This high-level model, although it will need to be refined in the further development of enterprise data warehouses and departmental data marts, will greatly reduce future integration problems.

II. Independent data marts can be implemented in parallel with the enterprise warehouse based on the same corporate data model set noted before. Third, distributed data marts can be constructed to integrate different data marts via hub servers. Finally, a **multitier data warehouse** is constructed where the enterprise warehouse is the sole custodian of all warehouse data, which is then distributed to the various dependent data marts.

Extraction, Transformation, and Loading

Data warehouse systems use back-end tools and utilities to populate and refresh their Data These tools and utilities include the following functions:

- **Data extraction**, which typically gathers data from multiple, heterogeneous, and external sources.
- **Data cleaning**, which detects errors in the data and rectifies them when possible.
- **Data transformation**, which converts data from legacy or host format to warehouse
- format.
- **Load**, which sorts, summarizes, consolidates, computes views, checks integrity, and builds indices and partitions.
- **Refresh**, which propagates the updates from the data sources to the warehouse.

Metadata Repository

Metadata are data about data. When used in a data warehouse, metadata are the data that define warehouse objects. Figure 4.1 showed a metadata repository within the bottom tier of the data warehousing architecture. Metadata are created for the data names and definitions of the given warehouse. Additional metadata are created and captured for timestamping any extracted data, the source of the extracted data, and missing fields that have been added by data cleaning or integration processes.

A metadata repository should contain the following :

A description of the data warehouse structure, which includes the warehouse schema, view, dimensions, hierarchies, and derived data definitions, as well as data mart locations and contents.

Operational metadata, which include data lineage (history of migrated data and the sequence of transformations applied to it), currency of data (active, archived, or purged), and monitoring information (warehouse usage statistics, error reports, and audit trails).

The algorithms used for summarization, which include measure and dimension definition algorithms, data on granularity, partitions, subject areas, aggregation, summarization, and predefined queries and reports.

Mapping from the operational environment to the data warehouse, which includes source databases and their contents, gateway descriptions, data partitions, data extraction, cleaning, transformation rules and defaults, data refresh and purging rules, and security (user authorization and access control).

Data related to system performance, which include indices and profiles that improve data access and retrieval performance, in addition to rules for the timing and scheduling of refresh, update, and replication cycles.

Business metadata, which include business terms and definitions, data ownership information, and charging policies.

A data warehouse contains different levels of summarization, of which metadata is one? Other types include current detailed data (which are almost always on disk), older detailed data (which are usually on tertiary storage), lightly summarized data, and highly summarized data (which may or may not be physically housed).

Metadata play a very different role than other data warehouse data and are important for many reasons. For example, metadata are used as a directory to help the decision support system analyst locate the contents of the data warehouse, and as a guide to the data mapping when data are transformed from the operational environment to the data warehouse environment. Metadata also serve as a guide to the algorithms used for summarization between the current detailed data and the lightly summarized data, and between the lightly summarized data and the highly summarized data.Metadata should be stored and managed persistently.

2.3 Stars, Snowflakes, and Fact Constellations: Schemas for Multidimensional Data Models

Q6. Discuss briefly about

1. Stars Schema

2. Snowflakes Schema

Ans :

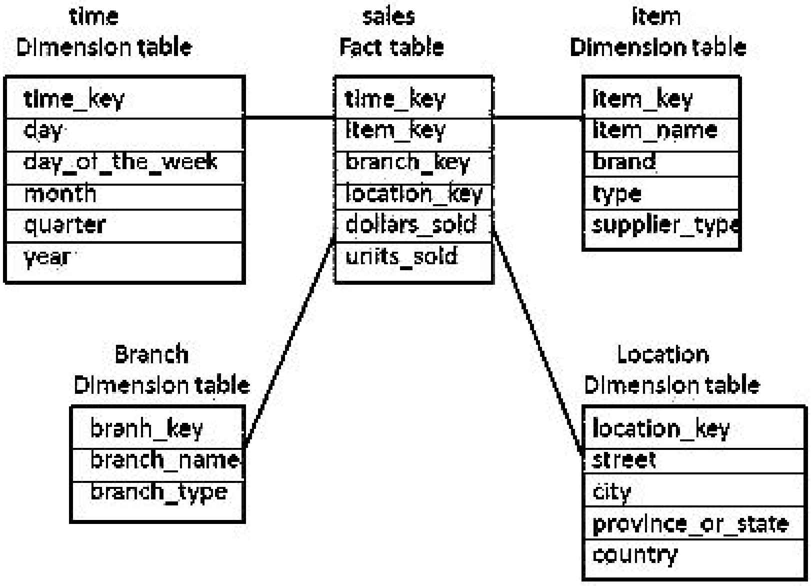
The entity-relationship data model is commonly used in the design of relational databases, where a database schema consists of a set of entities and the relationships between them. Such a data model is appropriate for online transaction processing.

A data warehouse, however, requires a concise, subject-oriented schema that facilitates online data analysis. Schema is a logical description of the entire database. It includes the name and description of records of all record types including all associated data-items and aggregates. Much like a database, a data warehouse also requires to maintain a schema. A database uses relational model, while a data warehouse uses Star, Snowflake, and Fact Constellation schema. In this chapter, we will discuss the schemas used in a data warehouse.

1. Star Schema

- Each dimension in a star schema is represented with only one-dimension table.
- This dimension table contains the set of attributes.
- The following diagram shows the sales data of a company with respect to the four dimensions, namely time, item, branch, and location.
- There is a fact table at the center. It contains the keys to each of four dimensions.
- The fact table also contains the attributes, namely dollars sold and units sold.

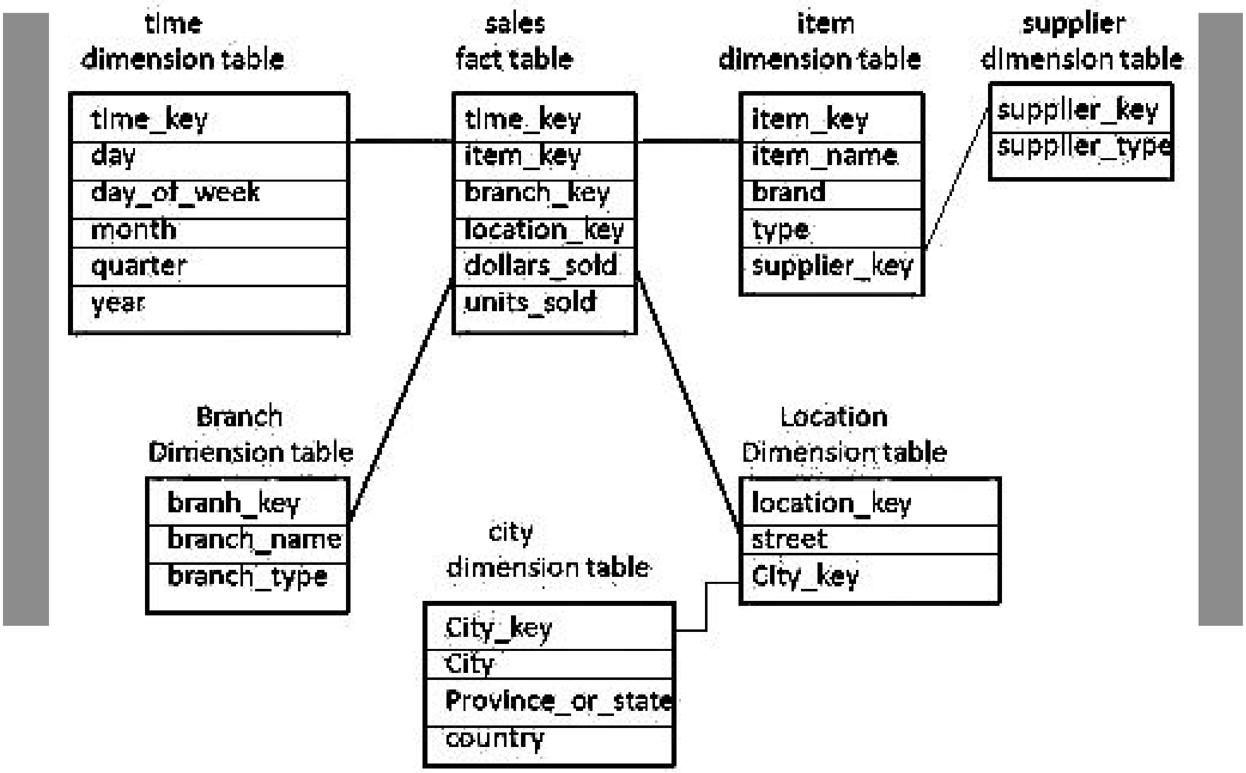
Note: Each dimension has only one dimension table and each table holds a set of attributes. For example, the location dimension table contains the attribute set {location_key, street, city, province_or_state,country}. This constraint may cause data redundancy. For example, "Vancouver" and "Victoria" both the cities are in the Canadian province of British Columbia. The entries for such cities may cause data redundancy along the attributes province_or_state and country.



2. Snowflake Schema

- Some dimension tables in the Snowflake schema are normalized.
- The normalization splits up the data into additional tables.
- Unlike Star schema, the dimensions table in a snowflake schema are normalized. For example, the item dimension table in star schema is normalized and split into two dimension tables, namely item and supplier table.
- Now the item dimension table contains the attributes item_key, item_name, type, brand, and supplier-key.
- The supplier key is linked to the supplier dimension table. The supplier dimension table contains the attributes supplier_key and supplier_type.

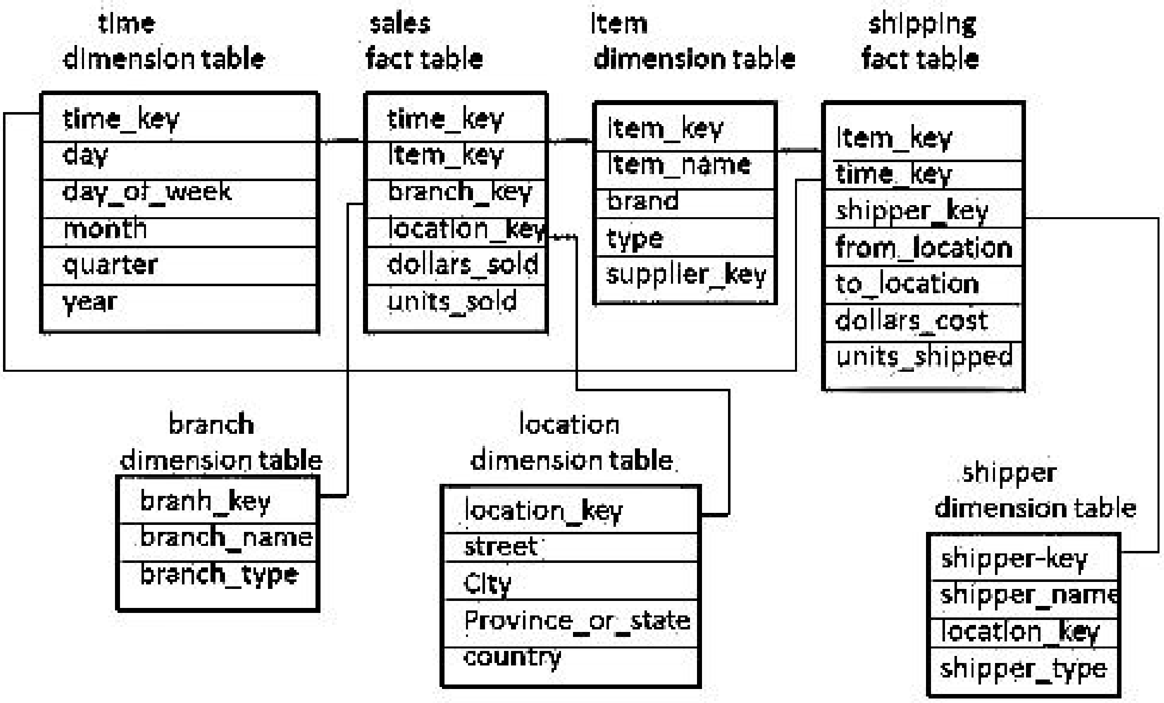
Note: Due to normalization in the Snowflake schema, the redundancy is reduced and therefore, it becomes easy to maintain and the save storage space.</b>



Fact Constellation Schema

- A fact constellation has multiple fact tables. It is also known as galaxy schema.
- The following diagram shows two fact tables, namely sales and shipping.
- he sales fact table is same as that in the star schema.
- The shipping fact table has the five dimensions, namely item_key, time_key, shipper_key, from_location, to_location.
- The shipping fact table also contains two measures, namely dollars sold and units sold.

It is also possible to share dimension tables between fact tables. For example, time, item, and location dimension tables are shared between the sales and shipping fact table.



Typical OLAP Operations

In the multidimensional model, data are organized into multiple dimensions, and each dimension contains multiple levels of abstraction defined by concept hierarchies. This organization provides users with the flexibility to view data from different perspectives. A number of OLAP data cube operations exist to materialize these different views, allowing interactive querying and analysis of the data at hand.

- **Roll-up:** The roll-up operation (also called the drill-up operation by some vendors) performs aggregation on a data cube, either by climbing up a concept hierarchy for a dimension or by dimension reduction. When roll-up is performed by dimension reduction, one or more dimensions are removed from the given cube. For example, consider a sales data cube containing only the location and time dimensions. Roll-up may be performed by removing, say,
- the time dimension, resulting in an aggregation of the total sales by location, rather than by location and by time.
- **Drill-down:** Drill-down is the reverse of roll-up. It navigates from less detailed data to more detailed data. Drill-down can be realized by either stepping down a concept hierarchy for a dimension or introducing additional dimensions. a drill-down adds more detail to the given data, it can also be performed by adding new dimensions to a cube.
- **Slice and dice:** The slice operation performs a selection on one dimension of the given cube,
- **Pivot (rotate):** Pivot (also called rotate) is a visualization operation that rotates the data axes in view to provide an alternative data presentation.

2.4 Data Warehouse Design Process

Q7. Explain the creation of Data Warehouse ?

Ans :

Data warehousing is a business analyst's dream—all the information about the organization's activities gathered in one place, open to a single set of analytical tools. But how do you make the dream a reality? First, you have to plan your data warehouse system. You must understand what questions users will ask it (e.g., how many registrations did the company receive in each quarter, or what industries are purchasing custom software development in the Northeast) because the purpose of a data warehouse system is to provide decision-makers the accurate, timely information they need to make the right choices.

We designed for a custom software development, consulting, staffing, and training company. The company's market is rapidly changing, and its leaders need to know what adjustments in their business model and sales practices will help the company continue to grow. To assist the company, we worked with the senior management staff to design a solution. First, we determined the business objectives for the system. Then we collected and analyzed information about the enterprise. We identified the core business processes that the company needed to track, and constructed a conceptual model of the data. Then we located the data sources and planned data transformations. Finally, we set the tracking duration.

Step 1: Determine Business Objectives

The company is in a phase of rapid growth and will need the proper mix of administrative, sales, production, and support personnel. Key decision-makers want to know whether increasing overhead staffing is returning value to the organization. As the company enhances the sales force and employs different sales modes, the leaders need to know whether these modes are effective. External market forces are changing the balance between a national and regional focus, and the leaders need to understand this change's effects on the business.

To answer the decision-makers' questions, we needed to understand what defines success for this business. The owner, the president, and four key managers oversee the company. These managers oversee profit centers and are responsible for making their areas successful. They also share resources, contacts, sales opportunities, and personnel. The managers examine different factors to measure the health and growth of their segments. Gross profit interests everyone in the group, but to make decisions about what generates that profit, the system must correlate more details. For instance, a small contract requires almost the same amount of administrative overhead as a large contract. Thus, many smaller contracts generate revenue at less profit than a few large contracts. Tracking contract size becomes important for identifying the factors that lead to larger contracts.

As we worked with the management team, we learned the quantitative measurements of business activity that decision-makers use to guide the organization. These measurements are the key performance indicators, a numeric measure of the company's activities, such as units sold, gross profit, net profit, hours spent, students taught, and repeat student registrations. We collected the key performance indicators into a table called a fact table.

Step 2: Collect and Analyze Information

The only way to gather this performance information is to ask questions. The leaders have sources of information they use to make decisions. Start with these data sources. Many are simple. You can get reports from the accounting package, the customer relationship management (CRM) application, the time reporting system, etc. You'll need copies of all these reports and you'll need to know where they come from.

Often, analysts, supervisors, administrative assistants, and others create analytical and summary reports. These reports can be simple correlations of existing reports, or they can include information that people overlook with the existing software or information stored in spreadsheets and memos. Such overlooked information can include logs of telephone calls someone keeps by hand, a small desktop database that tracks shipping dates, or a daily report a supervisor emails to a manager. A big challenge for data warehouse designers is finding ways to collect this information. People often write off this type of serendipitous information as unimportant or inaccurate. But remember that nothing develops without a reason. Before you disregard any source of information, you need to understand why it exists.

Another part of this collection and analysis phase is understanding how people gather and process the information. A data warehouse can automate many reporting tasks, but you can't automate what you haven't identified and don't understand. The process requires extensive interaction with the individuals involved. Listen carefully and repeat back what you think you heard. You need to clearly understand the process and its reason for existence. Then you're ready to begin designing the warehouse.

Step 3: Identify Core Business Processes

By this point, you must have a clear idea of what business processes you need to correlate. You've identified the key performance indicators, such as unit sales, units produced, and gross revenue. Now you need to identify the entities that interrelate to create the key performance indicators. For instance, at our example company, creating a training sale involves many people and business factors. The customer might not have a relationship with the company. The client might have to travel to attend classes or might need a trainer for an on-site class. New product releases such as Windows 2000 (Win2K) might be released often, prompting the need for training. The company might run a promotion or might hire a new salesperson.

The data warehouse is a collection of interrelated data structures. Each structure stores key performance indicators for a specific business process and correlates those indicators to the factors that generated them. To design a structure to track a business process, you need to identify the entities that work together to create the key performance indicator. Each key performance indicator is related to the entities that generated it. This relationship forms a dimensional model. If a salesperson sells 60 units, the dimensional structure relates that fact to the salesperson, the customer, the product, the sale date, etc.

Then you need to gather the key performance indicators into fact tables. You gather the entities that generate the facts into dimension tables. To include a set of facts, you must relate them to the dimensions (customers, salespeople, products, promotions, time, etc.) that created them.

For the fact table to work, the attributes in a row in the fact table must be different expressions of the same event or condition. You can express training sales by number of seats, gross revenue, and hours of instruction because these are different expressions of the same sale. An instructor taught one class in a certain room on a certain date. If you need to break the fact down into individual students and individual salespeople, however, you'd need to create another table because the detail level of the fact table in this example doesn't support individual students or salespeople.

A data warehouse consists of groups of fact tables, with each fact table concentrating on a specific subject. Fact tables can share dimension tables (e.g., the same customer can buy products, generate shipping costs, and return times). This sharing lets you relate the facts of one fact table to another fact table. After the data structures are processed as OLAP cubes, you can combine facts with related dimensions into virtual cubes.

Step 4: Construct a Conceptual Data Model

After identifying the business processes, you can create a conceptual model of the data. You determine the subjects that will be expressed as fact tables and the dimensions that will relate to the facts. Clearly identify the key performance indicators for each business process, and decide the format to store the facts in. Because the facts will ultimately be aggregated together to form OLAP cubes, the data needs to be in a consistent unit of measure. The process might seem simple, but it isn't. For example, if the organization is international and stores monetary sums, you need to choose a currency. Then you need to determine when you'll convert other currencies to the chosen currency and what rate of exchange you'll use. You might even need to track currency-exchange rates as a separate factor.

Now you need to relate the dimensions to the key performance indicators. Each row in the fact table is generated by the interaction of specific entities. To add a fact, you need to populate all the dimensions and correlate their activities. Many data systems, particularly older legacy data systems, have incomplete data. You need to correct this deficiency before you can use the facts in the warehouse. After making the corrections, you can construct the dimension and fact tables. The fact table's primary key is a composite key made from a foreign key of each of the dimension tables.

Data warehouse structures are difficult to populate and maintain, and they take a long time to construct. Careful planning in the beginning can save you hours or days of restructuring.

Step 5: Locate Data Sources and Plan Data Transformations

Now that you know what you need, you have to get it. You need to identify where the critical information is and how to move it into the data warehouse structure. For example, most of our example company's data comes from three sources. The company has a custom in-house application for tracking training sales. A CRM package tracks the sales-force activities, and a custom time-reporting system keeps track of time.

You need to move the data into a consolidated, consistent data structure. A difficult task is correlating information between the in-house CRM and time-reporting databases. The systems don't share information such as employee numbers, customer numbers, or project numbers. In this phase of the design, you need to plan how to reconcile data in the separate databases so that information can be correlated as it is copied into the data warehouse tables.

You'll also need to scrub the data. In online transaction processing (OLTP) systems, data-entry personnel often leave fields blank. The information missing from these fields, however, is often crucial for providing an accurate data analysis. Make sure the source data is complete before you use it. You can sometimes complete the information programmatically at the source. You can extract ZIP codes from city and state data, or get special pricing considerations from another data source. Sometimes, though, completion requires pulling files and entering missing data by hand. The cost of fixing bad data can make the system cost-prohibitive, so you need to determine the most cost-effective means of correcting the data and then forecast those costs as part of the system cost. Make corrections to the data at the source so that reports generated from the data warehouse agree with any corresponding reports generated at the source.

You'll need to transform the data as you move it from one data structure to another. Some transformations are simple mappings to database columns with different names. Some might involve converting the data storage type. Some transformations are unit-of-measure conversions (pounds to kilograms, centimeters to inches), and some are summarizations of data (e.g., how many total seats sold in a class per company, rather than each student's name). And some transformations require complex programs that apply sophisticated algorithms to determine the values. So you need to select the right tools (e.g., Data Transformation Services—DTS—running ActiveX scripts, or third-party tools) to perform these transformations. Base your decision mainly on cost, including the cost of training or hiring people to use the tools, and the cost of maintaining the tools.

You also need to plan when data movement will occur. While the system is accessing the data sources, the performance of those databases will decline precipitously. Schedule the data extraction to minimize its impact on system users (e.g., over a weekend).

Step 6: Set Tracking Duration

Data warehouse structures consume a large amount of storage space, so you need to determine how to archive the data as time goes on. But because data warehouses track performance over time, the data should be available virtually forever. So, how do you reconcile these goals?

The data warehouse is set to retain data at various levels of detail, or granularity. This granularity must be consistent throughout one data structure, but different data structures with different grains can be related through shared dimensions. As data ages, you can summarize and store it with less detail in another structure. You could store the data at the day grain for the first 2 years, then move it to another structure. The second structure might use a week grain to save space. Data might stay there for another 3 to 5 years, then move to a third structure where the grain is monthly. By planning these stages in advance, you can design analysis tools to work with the changing grains based on the age of the data. Then if older historical data is imported, it can be transformed directly into the proper format.

Step 7: Implement the Plan

After you've developed the plan, it provides a viable basis for estimating work and scheduling the project. The scope of data warehouse projects is large, so phased delivery schedules are important for keeping the project on track. We've found that an effective strategy is to plan the entire warehouse, then implement a part as a data mart to demonstrate what the system is capable of doing. As you complete the parts, they fit together like pieces of a jigsaw puzzle. Each new set of data structures adds to the capabilities of the previous structures, bringing value to the system.

Data warehouse systems provide decision-makers consolidated, consistent historical data about their organization's activities. With careful planning, the system can provide vital information on how factors interrelate to help or harm the organization. A solid plan can contain costs and make this powerful tool a reality.

2.5 Data Warehouse Usage for Information Processing

Q8. Explain applications of Information Processing ?

Ans :

Data warehouses and data marts are used in a wide range of applications. Business executives use the data in data warehouses and data marts to perform data analysis and make strategic decisions. In many firms, data warehouses are used as an integral part of a plan-execute-assess "closed-loop" feedback system for enterprise management. Data warehouses are used extensively in banking and financial services, consumer goods and retail distribution sectors, and controlled manufacturing such as demand-based production.

The longer a data warehouse has been in use, the more it will have evolved. This evolution takes place throughout a number of phases. Initially, the data warehouse is mainly used for generating reports and answering predefined queries. Progressively, it is used to analyze summarized and detailed data, where the results are presented in the form of reports and charts. Later, the data warehouse is used for strategic purposes, performing multidimensional analysis and sophisticated slice-and-dice operations.

The data warehouse may be employed for knowledge discovery and strategic decision making using data mining tools. In this context, the tools for data warehousing can be categorized into access and retrieval tools, database reporting tools, data analysis tools, and data mining tools.

Business users need to have the means to know what exists in the data warehouse (through metadata), how to access the contents of the data warehouse, how to examine the contents using analysis tools, and how to present the results of such analysis.

There are three kinds of data warehouse applications: information processing, analytical processing, and data mining.

- **Information processing** supports querying, basic statistical analysis, and reporting using crosstabs, tables, charts, or graphs. A current trend in data warehouse information processing is to construct low-cost web-based accessing tools that are then integrated with web browsers.
- **Analytical processing** supports basic OLAP operations, including slice-and-dice, drill-down, roll-up, and pivoting. It generally operates on historic data in both summarized and detailed forms. The major strength of online analytical processing over information processing is the multi dimensional data analysis of data warehouse data.
- **Data mining** supports knowledge discovery by finding hidden patterns and associations, constructing analytical models, performing classification and prediction, and presenting the mining results using visualization tools.

Online analytical processing comes a step closer to data mining because it can derive information summarized at multiple granularities from user-specified subsets of a data warehouse.

The functionalities of OLAP and data mining can be viewed as disjoint: OLAP is a data summarization/aggregation tool that helps simplify data analysis, while data mining allows the automated discovery of implicit patterns and interesting knowledge hidden in large amounts of data. OLAP tools are targeted toward simplifying and supporting interactive data analysis, whereas the goal of data mining tools is to automate as much of the process as possible, while still allowing users to guide the process. In this sense, data mining goes one step beyond traditional online analytical processing.

An alternative and broader view of data mining may be adopted in which data mining covers both data description and data modeling. Because OLAP systems can present general descriptions of data from data warehouses, OLAP functions are essentially for user-directed data summarization and comparison (by drilling, pivoting, slicing, dicing, and other operations). These are, though limited, data mining functionalities. Yet according to this view, data mining covers a much broader spectrum than simple OLAP operations, because it performs not only data summarization and comparison but also association, classification, prediction, clustering, time-series analysis, and other data analysis tasks.

Data mining is not confined to the analysis of data stored in data warehouses. It may analyze data existing at more detailed granularities than the summarized data provided in a data warehouse. It may also analyze transactional, spatial, textual, and multimedia data that are difficult to model with current multidimensional database technology. In this context, data mining covers a broader spectrum than OLAP with respect to data mining functionality and the complexity of the data handled.

2.6 Online Analytical Processing to Multidimensional Data Mining

Q9. Explain the Online Analytical Processing of Multidimensional Data Mining ?

Ans :

The data mining field has conducted substantial research regarding mining on various data types, including relational data, data from data warehouses, transaction data, time-series data, spatial data, text data, and flat files. **Multidimensional data mining** (also known as exploratory multidimensional data mining, **online analytical mining**, or **OLAM**) integrates OLAP with data mining to uncover knowledge in multidimensional databases. Among the many different paradigms and architectures of data mining systems, multidimensional data mining is particularly important for the following reasons:

- **High quality of data in data warehouses:** Most data mining tools need to work on integrated, consistent, and cleaned data, which requires costly data cleaning, data integration, and data transformation as preprocessing steps. A data warehouse constructed by such preprocessing serves as a valuable source of high-quality data for OLAP as well as for data mining. Notice that data mining may serve as a valuable tool for data cleaning and data integration as well.

- **Available information processing infrastructure surrounding data ware houses :** Comprehensive information processing and data analysis infrastructures have been or will be systematically constructed surrounding data warehouses, which include accessing, integration, consolidation, and transformation of multiple heterogeneous databases, ODBC/OLEDB connections, Web accessing and service facilities, and reporting and OLAP analysis tools. It is prudent to make the best use of the available infrastructures rather than constructing everything from scratch.

- **OLAP-based exploration of multi-dimensional data :** Effective data mining needs exploratory data analysis. A user will often want to traverse through a database, select portions of relevant data, and analyze them at different granularities, and present knowledge/results in different forms. Multidimensional data mining provides facilities for mining on different subsets of data and at varying levels of abstraction—by drilling, pivoting, filtering, dicing, and slicing on a data cube and/or intermediate data mining results. This, together with data/knowledge visualization tools, greatly enhances the power and flexibility of data mining.

- **Online selection of data mining functions:** Users may not always know the specific kinds of knowledge they want to mine. By integrating OLAP with various data mining functions, multidimensional data mining provides users with the flexibility to select desired data mining functions and swap data mining tasks dynamically.

2.7 Data Generalization

Q10. What is Data Generalization ?

Ans :

Data Generalization is the process of creating successive layers of summary data in an evaluational database. It is a process of zooming out to get a broader view of a problem, trend or situation. It is also known as rolling-up data.

There are millions and millions of data stored in the database and this number continues to increase everyday as a company heads for growth. In fact, a group of process of process called extract, transform, load (ETL) is periodically performed in order to manage data within the data warehouse.

A data warehouse is a rich repository of data, most of which are historical data from a company. But in modern data warehouses, data could come from other sources. Having data from several sources greatly helps in the overall business intelligence system of a company. With diverse data sources, the company can have a broader perspective not just about the trends and pattern within the organization but of the global industrial trends are well.

In order to get a view of trends and patterns based on the analytical outputs of the business intelligence system can be a daunting task. With those millions of data, most of which disparate (but of course ironed out by the ETL process), it may be difficult to generate reports.

Dealing alone with big volumes of data for consistent delivery of business critical applications can already affect the network management tools of a company. Many companies have found that existing network management tools could hardly cope up with the great bulk of data required by the organization to monitor network and applications usage.

The existing tools could hardly capture, store and report on traffic with speed and granularity which are requirements for real network improvements. In order to keep the volume down to speed up network performance for effective delivery, some network tools discard the details. What they would do is convert some detailed data into hourly, daily or weekly summaries. This is the process called data generalization or as some database professionals call it, rolling up data. Ensuring network manageability is just one of the benefits of data generalization.

Data generalization can provide a great help in Online Analytical Processing (OLAP) technology. OLAP is used for providing quick answers to analytical queries which are by nature multidimensional. They are commonly used as part of a broader category of business intelligence. Since OLAP is used mostly for business reporting such as those for sales, marketing, management reporting, business process management and other related areas, having a better view of trends and patterns greatly speeds up these reports.

Data generalization is also especially beneficial in the implementation of an Online transaction processing (OLTP). OLTP refers to a class systems designed for managing and facilitating transaction oriented applications especially those involved with data entry and retrieval transaction processing. OLAP was created later than OLTP and had slight modifications from OLTP.

Many companies who have been using the relatively older OLTP cannot abandon OLTP’s requirements and re-engineer for OLAP. In order to “upgrade” OLTP to some degree, the information system department needs to create, manage and support a dual database system. The two databases are the operational database and the evaluational database. The operational database supplies data to be used to support OLTP.

The evaluational database on the other hand will supply data to be used to support OLAP. By creating these two databases, the company can be able to maximize the effectiveness of both OLAP and OLTP. The two databases will differ in the characteristics of data contained within and how the data is used. For instance, in the “currentness” attribute of data, the operational data is current while the evaluational data is historic

Data Generalization by Attribute-Oriented Induction

In general, data generalization summarizes data by replacing relatively low-level values (e.g., numeric values for an attribute age) with higher-level concepts (e.g., young, middle-aged, and senior), or by reducing the number of dimensions to summarize data in concept space involving fewer dimensions (e.g., removing birth date and telephone number when summarizing the behavior of a group of students). Given the large amount of data stored in databases, it is useful to be able to describe concepts in concise and succinct terms at generalized (rather than low) levels of abstraction. Allowing data sets to be generalized at multiple levels of abstraction facilitates users in examining the general behavior of the data.

As a data mining task, concept description is not a simple enumeration of the data. Instead, **concept description** generates descriptions for data characterization and comparison. It is sometimes called **class description** when the concept to be described refers to a class of objects. **Characterization** provides a concise and succinct summarization of the given data collection, while concept or class **comparison** (also known as **discrimination**) provides descriptions comparing two or more data collections.

"Is data cube technology sufficient to accomplish all kinds of concept description tasks for large data sets?" Consider the following cases.

1. **Complex data types and aggregation** : Data warehouses and OLAP tools are based on a multidimensional data model that views data in the form of a data cube, consisting of dimensions (or attributes) and measures (aggregate functions). the aggregation of attributes in a database may include sophisticated data types such as the collection of non-numeric data, the merging of spatial regions, the composition of images, the integration of texts, and the grouping of object pointers.

   Therefore, OLAP, with its restrictions on the possible dimension and measure types, represents a simplified model for data analysis.

2. **User control versus automation** : Online analytical processing in data warehouses is a user-controlled process. The selection of dimensions and the application of OLAP operations (e.g., drill-down, roll-up, slicing, and dicing) are primarily directed and controlled by users. Although the control in most OLAP systems is quite userfriendly, users do require a good understanding of the role of each dimension.

Furthermore, in order to find a satisfactory description of the data, users may need to specify a long sequence of OLAP operations. It is often desirable to have a more automated process that helps users determine which dimensions (or attributes) should be included in the analysis, and the degree to which the given data set should be generalized in order to produce an interesting summarization of the data.

Data Characterization

The **attribute-oriented induction (AOI)** approach to concept description was first proposed in 1989, a few years before the introduction of the data cube approach. The data cube approach is essentially based on materialized views of the data, which typically have been precomputed in a data warehouse. In general, it performs offline aggregation before an OLAP or data mining query is submitted for processing. On the other hand, the attribute-oriented induction approach is basically a query-oriented, generalization-based, online data analysis technique. Note that there is no inherent barrier distinguishing the two approaches based on online aggregation versus offline precomputation. Some aggregations in the data cube can be computed online, while offline precomputation of multidimensional space can speed up attribute-oriented induction as well.

The generalization is performed by either attribute removal or attribute generalization. Aggregation is performed by merging identical generalized tuples and accumulating their respective counts. This reduces the size of the generalized data set.

Attribute-oriented induction. Here we show how attribute-oriented induction is performed on the initial working relation of Table 4.5. For each attribute of the relation, the generalization proceeds as follows :

- **name:** Since there are a large number of distinct values for name and there is no generalization operation defined on it, this attribute is removed.

- **gender**: Since there are only two distinct values for gender, this attribute is retained and no generalization is performed on it.
- **major:** Suppose that a concept hierarchy has been defined that allows the attribute major to be generalized to the values farts & sciences, engineering, business. Suppose also that the attribute generalization threshold is set to 5, and that there are more than 20 distinct values for major in the initial working relation. By attribute generalization and attribute generalization control, major is therefore generalized by climbing the given concept hierarchy.
- **birth place:** This attribute has a large number of distinct values; therefore, we would like to generalize it. Suppose that a concept hierarchy exists for birth place, defined as "city < province or state < country." If the number of distinct values for country in the initial working relation is greater than the attribute generalization threshold, then birth place should be removed, because even though a generalization operator exists for it, the generalization threshold would not be satisfied. If, instead, the number of distinct values for country is less than the attribute generalization threshold, then birth place should be generalized to birth country.
- **birth date:** Suppose that a hierarchy exists that can generalize birth date to age and age to age range, and that the number of age ranges (or intervals) is small with respect to the attribute generalization threshold. Generalization of birth date should therefore take place.
- **residence:** Suppose that residence is defined by the attributes number, street, residence city, residence province or state, and residence country. The number of distinct values for number and street will likely be very high, since these concepts are quite low level. The attributes number and street should therefore be removed so that residence is then generalized to residence city, which contains fewer distinct values.
- **phone#:** As with the name attribute, phone# contains too many distinct values and should therefore be removed in generalization.
- **gpa:** Suppose that a concept hierarchy exists for gpa that groups values for grade point average into numeric intervals like f3.75–4.0, 3.5–3.75, . . . g, which in turn are grouped into descriptive values such as f"excellent", "very good", . . . g. The attribute can therefore be generalized. gender major birth country age range residence city gpa count.

Attribute-Oriented Induction for Class Comparisons

In many applications, users may not be interested in having a single class (or concept) described or characterized, but prefer to mine a description that compares or distinguishes one class (or concept) from other comparable classes (or concepts). Class discrimination or comparison (hereafter referred to as **class comparison**) mines descriptions that distinguish a target class from its contrasting classes. Notice that the target and contrasting classes must be comparable in the sense that they share similar dimensions and attributes. For example, the three classes person, address, and item are not comparable. However, sales in the last three years are comparable classes, and so are, for example, computer science students versus physics students.

Our discussions on class characterization in the previous sections handle multilevel data summarization and characterization in a single class. The techniques developed can be extended to handle class comparison across several comparable classes. For example, the attribute generalization process described for class characterization can be modified so that the generalization is performed synchronously among all the classes compared. This allows the attributes in all of the classes to be generalized to the same abstraction levels. In general, the procedure is as follows:

1. **Data collection**: The set of relevant data in the database is collected by query processing and is partitioned respectively into a target class and one or a set of contrasting classes.
2. **Dimension relevance analysis**: If there are many dimensions, then dimension relevance analysis should be performed on these classes to select only the highly relevant dimensions for further analysis. Correlation or entropy-based measures can be used for this step

3. **Synchronous generalization**: Generalization is performed on the target class to the level controlled by a user- or expert-specified dimension threshold, which results in a **prime target class relation**. The concepts in the contrasting class(es) are generalized to the same level as those in the prime target class relation, forming the **prime contrasting class(es) relation**.

4. **Presentation of the derived comparison**: The resulting class comparison description can be visualized in the form of tables, graphs, and rules. This presentation usually includes a "contrasting" measure such as count% (percentage count) that reflects the comparison between the target and contrasting classes. The user can adjust the comparison description by applying drill-down, roll-up, and other OLAP operations to the target and contrasting classes, as desired.

2.8 Data Cube Implementations

Q11. Explain different types of Data Cube Technology and its implementation process ?

Ans :

1. **Pre-compute and store all:** This means that millions of aggregates will need to be computed and stored. Although this is the best solution as far as query response time is concerned, the solution is impractical since resources required to compute the aggregates and to store them will be prohibitively large for a large data cube. Indexing large amounts of data is also expensive.

2. **Pre-compute (and store) none:** This means that the aggregates are computed on- the-fly using the rawdata whenever a query is posed. This approach does not require additional space for storing the cube but the query response time is likely to be very poor for large data cubes.

3. **Pre-compute and store some:** This means that we pre-compute and store the most frequently queried aggregates and compute others as the need arises. We may also be able to derive some of the remaining aggregates using the aggregates that have already been computed. It may therefore be worthwhile also to pre-compute some aggregates that are not most frequently queried but help in deriving many other aggregates.

It will of course not be possible to derive all the aggregates from the pre-computed aggregates and it will be necessary to access the database (e.g the data warehouse) to compute the remaining aggregates. The more aggregates we are able to pre-compute the better the query performance.

Another related issue is where the data used by OLAP will reside. We assume that the data is stored in a data warehouse or in one or more data marts.

Data cube products use different techniques for pre-computing aggregates and storing them. They are generally based on one of two implementation models. The first model, supported by vendors of traditional relational model databases, is called the ROLAP model or the Relational OLAP model. The second model is called the MOLAP model for multidimensional OLAP. The MLOAP model provides a direct multidimensional view of the data whereas the RLOAP model provides a relational view of the multidimensional data in the form of a fact table.

- **ROLAP** uses a relational DBMS to implement an OLAP environment. It may be considered a bottom-up approach which is typically based on using a data warehouse that has been designed using a star schema. The data therefore is likely to be in a denormalized structure. A normalized database avoids redundancy but is usually not appropriate for high performance. The summary data will be held in aggregate tables. The data warehouse provides the multidimensional capabilities by representing data in fact table(s) and dimension tables. The fact table contains one column for each dimension and one column for each measure and every row of the table [rovides one fact. A fact then is represented as (BSc, India, 2001-01) with the last column as 30. An OLAP tool is then provided to manipulate the data in these data warehouse tables. This tool essentially groups the fact table to find aggregates and uses some of the aggregates already computed to find new aggregates.

The advantage of using ROLAP is that it is more easily used with existing relational DBMS and the data can be stored efficiently using tables since no zero facts need to be stored. The disadvantage of the ROLAP model is its poor query performance. Proponents of the MLOAP model have called the ROLAP model SLOWLAP. Some products in this category are Oracle OLAP mode, OLAP Discoverer, MicroStrategy and Microsoft Analysis Services.

- **MOLAP** is based on using a multidimensional DBMS rather than a data warehouse to store and access data. It may be considered as a top-down approach to OLAP. The multi-dimensional database systems do not have a standard approach to storing and maintaining their data. They often use special-purpose file systems or indexes that store pre-computation of all aggregations in the cube.

For example, in ROLAP a cell was represented as (BSc, India, 2001-01) with a value 30 stored in the last column. In MOLAP, the same information is stored as 30 and the storage location implicitly gives the values of the dimensions. The dimension values do not need to be stored since all the values of the cube could be stored in an array in a predefined way.

The differences between ROLAP and MOLAP are summarized as :

Property	MOLAP	ROLAP
Data structure	Multidimensional database using sparse arrays	Relational tables (each cell is a row)
Disk space	Separate database for data cube; large for large data cubes	May not require any space other than that available in the data warehouse
Retrieval	Fast(pre-computed)	Slow(computes on-the-fly)
Scalability	Limited (cubes can be very large)	Excellent
Best suited for	Inexperienced users, limited set of queries	Experienced users, queries change frequently
DBMS facilities	Usually weak	Usually very strong

2.9 Data Cube

Q12. Definition - What does Data Cube mean?

Ans :

A data cube refers is a three-dimensional (3D) (or higher) range of values that are generally used to explain the time sequence of an image's data. It is a data abstraction to evaluate aggregated data from a variety of viewpoints. It is also useful for imaging spectroscopy as a spectrally-resolved image is depicted as a 3-D volume.

A data cube can also be described as the multidimensional extensions of two-dimensional tables. It can be viewed as a collection of identical 2-D tables stacked upon one another. Data cubes are used to represent data that is too complex to be described by a table of columns and rows. As such, data cubes can go far beyond 3-D to include many more dimensions.

A data cube is generally used to easily interpret data. It is especially useful when representing data together with dimensions as certain measures of business requirements. A cube's every dimension represents certain characteristic of the database, for example, daily, monthly or yearly sales. The data included inside a data cube makes it possible analyze almost all the figures for virtually any or all customers, sales agents, products, and much more. Thus, a data cube can help to establish trends and analyze performance.

Data cubes are mainly categorized into two categories :

- **Multidimensional Data Cube:** Most OLAP products are developed based on a structure where the cube is patterned as a multi-dimensional array. These multi-dimensional OLAP (MOLAP) products usually offers improved performance when compared to other approaches mainly because they can be indexed directly into the structure of the data cube to gather subsets of data. When the number of dimensions is greater, the cube becomes sparser. That means that several cells that represent particular attribute combinations will not contain any aggregated data. This in turn boosts the storage requirements, which may reach undesirable levels at times, making the MOLAP solution untenable for huge data sets with many dimensions. Compression techniques might help; however, their use can damage the natural indexing of MOLAP.
- **Relational OLAP:** Relational OLAP make use of the relational database model. The ROLAP data cube is employed as a bunch of relational tables (approximately twice as many as the quantity of dimensions) compared to a multidimensional array. Each one of these tables, known as a cuboid, signifies a specific view.

2.9.1 Cube Materialization: Full Cube, Iceberg Cube,Closed Cube, and Cube Shell

Q13. Explain in detail about

1. **Full Cube**
2. **Iceberg Cube**
3. **Closed Cube**
4. **Cube Shell**

Ans :

A 3-D data cube for the dimensions A, B, and C, and an aggregate measure, M. Commonly used measures include count(), sum(), min(), max(), and total sales(). A data cube is a lattice of cuboids. Each cuboid represents a group-by. ABC is the base cuboid, containing all three of the dimensions. Here, the aggregate measure, M, is computed for each possible combination of the three dimensions. The base cuboid is the least generalized of all the cuboids in the data cube. The most generalized cuboid is the apex cuboid, commonly represented as all. It contains one value—it aggregates measure M for all the tuples stored in the base cuboid. To drill down in the data cube, we move from the apex cuboid downward in the lattice. To roll up, we move from the base cuboid upward.

A cell in the base cuboid is a **base cell**. A cell from a nonbase cuboid is an **aggregate cell**. An aggregate cell aggregates over one or more dimensions, where each aggregated dimension is indicated by a _ in the cell notation. Suppose we have an n-dimensional data cube. Let a D .a1, a2, : : : , an, measures/ be a cell from one of the cuboids making up the data cube. We say that a is an **m-dimensional cell** (i.e., from an m-dimensional cuboid) if exactly m (m _ n) values among fa1, a2, : : : , ang are not _. If m D n, then a is a base cell; otherwise, it is an aggregate cell (i.e., where m < n). Ensure fast OLAP, it is sometimes desirable to pre-compute the **full cube** (i.e., all the cells of all the cuboids for a given data cube). That is, a data cube of n dimensions contains 2n cuboids. There are even more cuboids if we consider concept hierarchies for each dimension.1 In addition, the size of each cuboid depends on the cardinality of its dimensions. Thus, pre-computation of the full cube can require huge and often excessive amounts of memory.

- full cube computation algorithms are important. **Individual** cuboids may be stored on secondary storage and accessed when necessary. Alternatively, we can use such algorithms to compute smaller cubes, consisting of a subset of the given set of dimensions, or a smaller range of possible values for some of the dimensions. In these cases, the smaller cube is a full cube for the given subset of dimensions and/or dimension values. A thorough understanding of full cube computation methods will help us develop efficient methods for computing partial cubes. Hence, it is important to explore scalable methods for computing all the cuboids making up a data cube, that is, for full materialization. These methods must take into consideration the limited amount of main memory available for cuboid computation, the total size of the computed data cube, as well as the time required for such computation.

- Partial materialization of data cubes offers an interesting trade-off between storage space and response time for OLAP. Instead of computing the full cube, we can compute only a subset of the data cube's cuboids, or sub cubes consisting of subsets of cells from the various cuboids. Many cells in a cuboid may actually be of little or no interest to the data analyst. Recall that each cell in a full cube records an aggregate value such as count or sum. For many cells in a cuboid, the measure value will be zero. When the product of the cardinalities for the dimensions in a cuboid is large relative to the number of nonzero-valued tuples that are stored in the cuboid, then we say that the cuboid is **sparse**. If a cube contains many sparse cuboids, we say that the cube is **sparse**. In many cases, a substantial amount of the cube's space could be taken up by a large number of cells with very low measure values. This is because the cube cells are often quite sparsely distributed within a multidimensional space.

2.10 General Strategies for Data Cube Computation

Q14. Explain about Data Cube Computation Strategies ?

Ans :

There are several methods for efficient data cube computation, based on the various kinds of cubes. In general, there are two basic data structures used for storing cuboids. The implementation of relational OLAP (ROLAP) uses relational tables, whereas multidimensional arrays are used in multidimensional OLAP (MOLAP). Although ROLAP and MOLAP may each explore different cube computation techniques, some optimization "tricks" can be shared among the different data representations. The following are general optimization techniques for efficient computation of data cubes.

- **Optimization Technique 1: Sorting, hashing, and grouping.** Sorting, hashing, and grouping operations should be applied to the dimension attributes to reorder and cluster related tuples.

  In cube computation, aggregation is performed on the tuples (or cells) that share the same set of dimension values. Thus, it is important to explore sorting, hashing, and grouping operations to access and group such data together to facilitate computation of such aggregates.

  To compute total sales by branch, day, and item, for example, it can be more efficient to sort tuples or cells by branch, and then by day, and then group them according to the item name. Efficient implementations of such operations in large data sets have been extensively studied in the database research community. Such implementations can be extended to data cube computation.

  This technique can also be further extended to perform **shared-sorts** (i.e., sharing sorting costs across multiple cuboids when sort-based methods are used), or to perform **shared-partitions** (i.e., sharing the partitioning cost across multiple cuboids when hash-based algorithms are used).

- **Optimization Technique 2: Simultaneous aggregation and caching of intermediate results.** In cube computation, it is efficient to compute higher-level aggregates from previously computed lower-level aggregates, rather than from the base fact table.

  Moreover, simultaneous aggregation from cached intermediate computation results may lead to the reduction of expensive disk input/output (I/O) operations. To compute sales by branch, for example, we can use the intermediate results derived fromthe computation of a lower-level cuboid such as sales by branch and day. This technique can be further extended to performamortized **scans** (i.e., computing as many cuboids as possible at the same time to amortize disk reads).

- **Optimization Technique 3: Aggregation from the smallest child when there exist multiplechild cuboids.** When there exist multiple child cuboids, it is usually more efficient to compute the desired parent (i.e., more generalized) cuboid from the smallest, previously computed child cuboid.

  To compute a sales cuboid, Cbranch, when there exist two previously computed cuboids, Cfbranch, yearg and Cfbranch, itemg, for example, it is obviously more efficient to compute Cbranch from the former than from the latter if there are many more distinct items than distinct years.

  Many other optimization techniques may further improve computational efficiency. For example, string dimension attributes can be mapped to integers with values ranging from zero to the cardinality of the attribute. In iceberg cube computation the following optimization technique plays a particularly important role.

- **Optimization Technique 4: The Apriori pruning method can be explored to compute iceberg cubes efficiently.** The **Apriori property**, in the context of data cubes, states as follows: If a given cell does not satisfy minimum support, then no descendant of the cell (i.e., more specialized cell) will satisfy minimum support either. This property can be used to substantially reduce the computation of iceberg cubes.

  Recall that the specification of iceberg cubes contains an iceberg condition, which is a constraint on the cells to be materialized. A common iceberg condition is that the cells must satisfy a minimum support threshold such as a minimum count or sum. In this situation, the Apriori property can be used to prune away the exploration of the cell's descendants. For example, if the count of a cell, c, in a cuboid is less than a minimum support threshold, v, then the count of any of c's descendant cells in the lower-level cuboids can never be greater than or equal to v, and thus can be pruned.

  In other words, if a condition (e.g., the iceberg condition specified in the having clause) is violated for some cell c, then every descendant of c will also violate that condition. Measures that obey this property are known as **antimonotonic**.This form of pruning was made popular in frequent pattern mining, yet also aids in data cube computation by cutting processing time and disk space requirements. It can lead to a more focused analysis because cells that cannot pass the threshold are unlikely to be of interest.

2.10.1 Data Cube Computation Methods

Q15. Discuss in detail about Data Cube Computation Methods.

Ans :

Data cube computation is an essential task in data warehouse implementation. The pre-computation of all or part of a data cube can greatly reduce the response time and enhance the performance of online analytical processing.

Multi- Dimensional aggregate computation extended basic sort based and hash based methods to compute multiple group-by by incorporating optimizations techniques like smallest-parent, cache-results, amortize-scans, share-sorts and share-partitions. Smallest-parent: This optimization aims at computing a group by from the smallest previously computed group-by. In this, each group-by can be computed from a number of other group bys. Cache-results: This optimization aims at caching (in memory) the results of a group-by from which other group by are computed to reduce disk I/O.

1. **Amortize-scans:** This optimization aims at amortizing disk reads by computing as many group-by as possible, together in memory. Share-sorts: This optimization is specific to the sort-based algorithms and aims at sharing sorting cost across multiple group bys. Share-partitions: This optimization is specific to the hash based algorithms. When the hash table is too large to fit in memory, data is partitioned and aggregation is done for each partition that fits in memory. We can save on partitioning cost by sharing this cost across multiple group-by.

2. **Top-Down Approach:** Multi-Way array aggregation [3] The computation starts from the larger group-bys and proceeds towards the smallest group-bys. In this, a partition-based loading algorithm designed and implemented to convert a relational table or external load file to a (possibly compressed) chunked array. There is no direct tuple comparisons.It perform Simultaneous aggregation on multiple dimensions.

   In MultiWay array aggregation Intermediate aggregate values are re-used for computing ancestor cuboids .It cannot do Apriori pruning means it cannot perform iceberg cube optimization . In Multi-Way array aggregation, It partition arrays into chunks (a small sub cube which fits in memory). It uses compressed sparse array addressing: (chunk_id, offset) and compute aggregates in ¯ "multiway" by visiting cube cells in the order which minimizes the # of times to visit each cell, and reduces memory access and storage cost. What is the best traversing order to do multi-way aggregation? - Method: the planes should be sorted and computed according to their size in ascending order - Idea: keep the smallest plane in the main memory, fetch and compute only one chunk at a time for the largest plane. Limitation of the method: computing well only for a small number of dimensions. If there are a large number of dimensions, top-down computation and iceberg cube computation methods can be explored.

3. **Bottom-Up Approach :** Bottom-Up Computation (BUC) BUC is an algorithm for sparse and iceberg cube computation. BUC uses the bottom-up approach that allows pruning unnecessary computation by recurring to A-priori pruning strategy. if a given cell does not satisfy minsup, then no descendant will satisfy minsup either. The Iceberg cube problem is to compute all group-bys that satisfy an iceberg condition.

4. **Mixed Approach:** Star cubing [8] Star Cubing integrate the top-down and bottom-up methods. It explore shared dimensions .E.g., dimension A is the shared dimension of ACD and AD. ABD/AB means cuboid ABD has shared dimensions AB. Star cubing allows for shared computations .e.g., cuboid AB is computed simultaneously as ABD . Star Cubing aggregate in a topdown manner but with the bottom-up sub-layer underneath which will allow Apriori pruning. It's shared dimensions grow in bottom-up fashion.

   Star-Cubing Algorithm—DFS on Lattice Tree Properties of Proposed Method Partitions the data vertically Reduces high-dimensional cube into a set of lower dimensional cubes Online re-construction of original high-dimensional space,

   - Lossless reduction
   - Offers tradeoffs between the amount of pre-processing and the speed of online computation Further Implementation Considerations Incremental Update:
   - Append more TIDs to inverted list
   - Add < tID_measure> table Incremental adding new dimensions
   - Form new inverted list and add new fragments
   - Bitmap indexing
   - May further improve space usage and speed
   - Inverted index compression
   - Store as d-gaps
   - Explore more IR compression methods

5. **High-Dimensional OLAP:** A Minimal Cubing Approach [9] In many applications, like bioinformatics, statistics and text processing, datasets are characterized by high dimensionality e.g. over 100 dimensions -> 2100 cuboids in a full cube. As huge cube there is infeasible computation time. Iceberg cube is not an ultimate solution as it cannot be incrementally updated. In this low minsup requires too space and high minsup gives no significant results.

   A minimal cubing approach, a new semi-online computational model is based on the computation of shell fragments. This method partitions 'vertically' a high dimensional dataset into a set of disjoint low dimensional datasets called fragments. Then, for each fragment, it computes local data cube. In shell fragment efficiency is obtained by using inverted index, i.e. a list of record-ids associated to each dimension value. Given the precomputed fragment cubes, intersection among fragments is performed online by re-assembling cuboids of the required data cube. It reduces high dimensionality of the data cube to lower dimensionality. Online operations of re-construction of original dimensional space. There is Tradeoffs between the pre-processing phase and the performance of online computation.

6. **Parallel Approaches:** Parallel Algorithms are introduced for cube computation over small PC clusters. Algorithm BPP (Breadth-first Writing, Partitioned, Parallel BUC), in which the dataset is not replicated, but is range partitioned on an attribute basis. The output of cuboids is done in a breadth-First fashion, as opposed to the depth-first writing that BUC do.

In Depth First writing, cells may belong to different cuboids. For example, the cell a1 belongs to cuboid A, the cell a1b1 to cuboid AB, and the cells a1b1c1 and a1b1c2 belong to ABC. The point is that cuboids is scattered. This clearly incurs a high I/O over-head. It is possible to use buffering to help the scattered writing to the disk. However, this may require a large amount of buffering space, thereby reducing the amount of memory available for the actual computation.

Furthermore, many cuboids may need to be maintained in the buffers at the same time, causing extra management overhead. In BPP, this problem is solved by breadth-first writing, implemented by first sorting the input dataset on the "prefix" attributes. Breadth-First I/O is a significant improvement over the scattering I/O used in BUC. Another Parallel algorithm PT (Partitioned Tree) works with tasks that are created by a recursive binary division of a tree into two sub trees having an equal number of nodes.

In PT, there is a parameter that controls when binary division stops.PT tries to exploit a affinity scheduling. During processor assignment, the manager tries to assign to a worker processor a task that can take advantage of prefix affinity based on the root of the subtree.PT is top-down. But interestingly, because each task is a sub tree, the nodes within the sub tree can be traversed/ computed in a bottomup fashion. In fact, PT calls BPP-BUC, which offers breadth-first writing, to complete the processing. Algorithm PT load-balances by using binary partitioning to divide the cube lattice as evenly as possible PT is the algorithm of choice for most situations.

Ranking Cubes: Efficient Computation of Top-k Queries

The data cube helps not only online analytical processing of multidimensional queries but also search and data mining. In this section, we introduce a new cube structure called Ranking Cube and examine how it contributes to the efficient processing of top-k queries. Instead of returning a large set of indiscriminative answers to a query, a **top-k query** (or **ranking query**) returns only the best k results according to a user-specified preference.

The results are returned in ranked order so that the best is at the top. The user specified preference generally consists of two components: a selection condition and a ranking function. Top-k queries are common in many applications like searching web databases, k-nearest-neighbor searches with approximate matches, and similarity queries in multimedia databases.

2.10.2 Processing Advanced Kinds of Queries by Exploring Cube Technology

Q16. Explain about Data Cube Technology ?

Ans :

The methods described in this section further develop data cube technology for effective processing of advanced kinds of queries. This extension of data cube technology can be used to answer queries on sample data, such as survey data, which represent a sample or subset of a target data population of interest. cars," according to some user-specified criteria.

The basic data cube structure has been further extended for various sophisticated data types and new applications. Here we list some examples, such as spatial data cubes for the design and implementation of geospatial data warehouses, and multimedia data cubes for the multidimensional analysis of multimedia data (those containing images and videos). RFID data cubes handle the compression and multidimensional analysis of RFID (i.e., radio-frequency identification) data. Text cubes and topic cubes were developed for the application of vector-space models and generative language models respectively, in the analysis of multidimensional text databases (which contain both structure attributes and narrative text attributes).

OLAP-Based Mining on Sampling Data

When collecting data, we often collect only a subset of the data we would ideally like to gather. In statistics, this is known as collecting a **sample** of the data population. The resulting data are called **sample data**. Data are often sampled to save on costs, manpower, time, and materials. In many applications, the collection of the entire data population of interest is unrealistic. it is impossible to gather the opinion of everyone in the population. Most published ratings or polls rely on a data sample for analysis. The results are extrapolated for the entire population, and associated with certain statistical measures such as a confidence interval. The confidence interval tells us how reliable a result is. Statistical surveys based on sampling are a common tool in many fields like politics, healthcare, market research, and social and natural sciences.

"How effective is OLAP on sample data?"

- **First**, sample data are often sparse in the multidimensional sense. When a user drills down on the data, it is easy to reach a point with very few or no samples even when the overall sample size is large. Traditional OLAP simply uses whatever data are available to compute a query answer. To extrapolate such an answer for a population based on a small sample could be misleading: A single outlier or a slight bias in the sampling can distort the answer significantly.
- **Second**, with sample data, statistical methods are used to provide a measure of reliability (e.g., a confidence interval) to indicate the quality of the query answer as it pertains to the population. Traditional OLAP is not equipped with such tools. A sampling cube framework was introduced to tackle each of the preceding challenges.

Sampling Cube Framework

The **sampling cube** is a data cube structure that stores the sample data and their multidimensional aggregates. It supports OLAP on sample data. It calculates confidence intervals as a quality measure for any multidimensional query.

In statistics, a confidence interval is used to indicate the reliability of an estimate. Suppose we want to estimate the mean age of all viewers of a given TV show. We have sample data (a subset) of this data population. say our sample mean is 35 years. This becomes our estimate for the entire population of viewers as well, but how confident can we be that 35 is also the mean of the true population? It is unlikely that the sample mean will be exactly equal to the true population mean because of sampling error.

This is typically done by computing a **confidence interval**, which is an estimated value range with a given high probability of covering the true population value. A confidence interval for our example could be "the actual mean will not vary by C/O two standard deviations 95% of the time." A confidence interval is always qualified by a particular confidence level. In our example, it is 95%.

Query Processing: Boosting Confidences for Small Samples

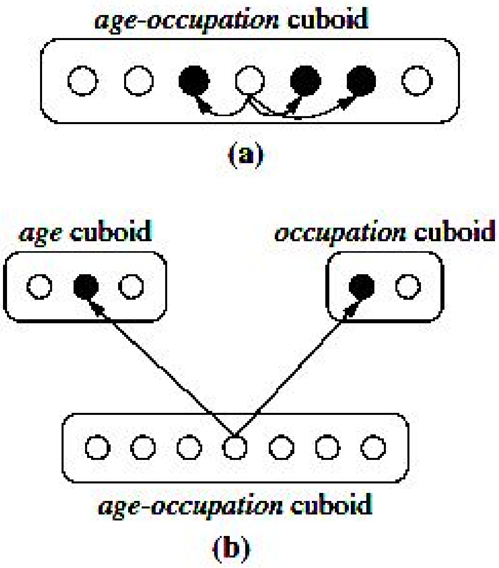
A query posed against a data cube can be either a point query or a range query. Without loss of generality, consider the case of a point query. Here, it corresponds to a cell in sampling cube. The goal is to provide an accurate point estimate for the samples in that cell. Because the cube also reports the confidence interval associated with the sample mean, there is some measure of "reliability" to the returned answer. If the confidence interval is small, the reliability is deemed good;

"What can we do to boost the reliability of query answers?" There are two main factors: the variance of the sample data and the sample size. First, a rather large variance in the cell may indicate that the chosen cube cell is poor for prediction. A better solution is probably to drill down on the query cell to a more specific one (i.e., asking more specific queries). Second, a small sample size can cause a large confidence interval. When there are very few samples, the corresponding %c is large because of the small degree of freedom. This in turn could cause a large confidence interval. Intuitively, this makes sense. Suppose one is trying to figure out the average income of people in the United States. Just asking two or three people does not give much confidence to the returned response.

The best way to solve this small sample size problem is to get more data. The data do not match the query cell exactly; however, we can consider data from cells that are "close by." There are two ways to incorporate such data to enhance the reliability of the query answer :

- **intracuboid query expansion** (where we consider nearby cells within the same cuboid)
- **intercuboid query expansion**

Method 1. Intracuboid query expansion. Here, we expand the sample size by including nearby cells in the same cuboid as the queried cell,. We just have to be careful that the new samples serve to increase the confidence in the answer without changing the query's semantics.



Method 2. Intercuboid query expansion. In this case, the expansion occurs by looking to a more general cell, the cell in the 2-D cuboid, can use its parent in either of the 1-D cuboids, age or occupation. Think of intercuboid expansion as just an extreme case of intracuboid expansion, where all the cells within a dimension are used in the expansion. This essentially sets the dimension to _ and thus generalizes to a higher-level cuboid.

A k-dimensional cell has k direct parents in the cuboid lattice, where each parent is (k>1)-dimensional. There are many more ancestor cells in the data cube. As with intracuboid query expansion, correlated dimensions are not allowed in intercuboid expansions. Within the uncorrelated dimensions, the two-sample t -test can be performed to confirm that the parent and the query cell share the same sample mean. If multiple parent cells pass the test, the test's confidence level can be adjusted progressively higher until only one passes. Alternatively, multiple parent cells can be used to boost the confidence simultaneously. The choice is application dependent.

2.10.3 Multidimensional Data Analysis in Cube Space

Q17. Discuss in detail about Data Analysis in Cube Space>

Ans :

Data cubes create a flexible and powerful means to group and aggregate data subsets. They allow data to be explored in multiple dimensional combinations and at varying aggregate granularities. This capability greatly increases the analysis bandwidth and helps effective discovery of interesting patterns and knowledge from data. The use of cube space makes the data space both meaningful and tractable. This section presents methods of multidimensional data analysis that make use of data cubes to organize data into intuitive regions of interest at varying granularities. presents prediction cubes, a technique for multidimensional data mining that facilitates predictive modeling in multidimensional space.

Prediction Cubes: Prediction

Mining in Cube Space

Researchers have turned their attention toward **multidimensional data mining** to uncover knowledge at varying dimensional combinations and granularities. Such mining is also known as exploratory multidimensional data mining and online analytical data mining (OLAM). Multidimensional data space is huge. This analyzes and mines the data by applying various data mining techniques systematically over these regions.

There are at least four ways in which OLAP-style analysis can be fused with data mining techniques :

1. Use cube space to define the data space for mining. Each region in cube space represents a subset of data over which we wish to find interesting patterns. Cube space is defined by a set of expert-designed, informative dimension hierarchies, not just arbitrary subsets of data. Therefore, the use of cube space makes the data space both meaningful and tractable.
2. Use OLAP queries to generate features and targets for mining. The features and even the targets (that we wish to learn to predict) can sometimes be naturally defined as OLAP aggregate queries over regions in cube space.
3. Use data mining models as building blocks in a multistep mining process. Multidimensional data mining in cube space may consist of multiple steps, where data mining models can be viewed as building blocks that are used to describe the behavior of interesting data sets, rather than the end results.
4. Use data cube computation techniques to speed up repeated model construction. Multi-dimensional data mining in cube space may require building a model for each candidate data space, which is usually too expensive to be feasible. However, by carefully sharing computation across model construction for different candidates based on data cube computation techniques, efficient mining is achievable.

Multi-feature Cubes: Complex Aggregation at Multiple Granularities

Data cubes facilitate the answering of analytical or mining-oriented queries as they allow the computation of aggregate data at multiple granularity levels. Traditional data cubes are typically constructed on commonly used dimensions using simple measures a newer way to define data cubes called **multi-feature cubes**. Multi-feature cubes enable more in-depth analysis. They can compute more complex queries of which the measures depend on groupings of multiple aggregates at varying granularity levels. The queries posed can be much more elaborate and task-specific than traditional queries,

The computation of a multi-feature cube depends on the types of aggregate functions used in the cube. We saw that aggregate functions can be categorized as either distributive, algebraic, or holistic. Multi-feature cubes can be organized into the same categories and computed efficiently by minor extension of the cube computation methods.

Exception-Based, Discovery-Driven Cube Space Exploration

A data cube may have a large number of cuboids, and each cuboid may contain a large number of (aggregate) cells. With such an overwhelmingly large space, it becomes a burden for users to even just browse a cube, let alone think of exploring it thoroughly. Tools need to be developed to assist users in intelligently exploring the huge aggregated space of a data cube. A **discovery-driven approach** to exploring cube space.

Pre-computed measures indicating data exceptions are used to guide the user in the data analysis process, at all aggregation levels. We hereafter refer to these measures as exception indicators. an **exception** is a data cube cell value that is significantly different from the value anticipated, based on a statistical model. The model considers variations and patterns in the measure value across all the dimensions to which a cell belongs. The model considers exceptions hidden at all aggregated group-by's of a data cube.

Visual cues, such as background color, are used to reflect each cell's degree of exception, based on the precomputed exception indicators. The computation of exception indicators can be overlapped with cube construction, so that the overall construction of data cubes for discovery-driven exploration is efficient. Three measures are used as exception indicators to help identify data anomalies. These measures indicate the degree of surprise that the quantity in a cell holds, with respect to its expected value. The measures are computed and associated with every cell, for all aggregation levels. They are as follows :

- **SelfExp** : This indicates the degree of surprise of the cell value, relative to other cells at the same aggregation level.
- **InExp** : This indicates the degree of surprise somewhere beneath the cell, if we were to drill down from it.
- **PathExp** : This indicates the degree of surprise for each drill-down path from the cell.

2.11 Mining Frequent Patterns, Associations, and Correlations: Basic Concepts and Methods

Q18. Explain different types of patterns used in Data Warehouse ?

Ans :

Frequent Pattern Mining

Frequent pattern mining can be classified in various ways, based on the following criteria :

1. **Based on the completeness of patterns to be mined:** We can mine the complete set of frequent itemsets, the closed frequent itemsets, and the maximal frequent itemsets, given a minimum support threshold. We can also mine constrained frequent itemsets, approximate frequent itemsets,near-match frequent itemsets, top-k frequent itemsets and so on.

2. **Based on the levels of abstraction involved in the rule set:** Some methods for associationrule mining can find rules at differing levels of abstraction. For example, supposethat a set of association rules mined includes the following rules where X is a variablerepresenting a customer :

buys(X, ¯computer)) = > buys(X, ¯HP printer) buys(X, ¯laptop computer))
= > buys(X, ¯HP printer)

The items bought are referenced at different levels ofabstraction (e.g., ¯computer is a higher-level abstraction of ¯laptop computer).

3. **Based on the number of data dimensions involved in the rule:** If the items or attributes in an association rule reference only one dimension, then it is a single-dimensional association rule.

buys(X, ¯computer)) = > buys(X, ¯antivirus software)

If a rule references two or more dimensions, such as the dimensions age, income, and buys, then it is amultidimensional association rule. The following rule is an exampleof a multidimensional rule:

age(X, ¯30,31…39) ^ income(X, ¯42K,…48K)) = > buys(X, ¯high resolution TV)

4. **Based on the types of values handled in the rule:**
   - If a rule involves associations between the presence or absence of items, it is a Boolean association rule.
   - If a rule describes associations between quantitative items or attributes, then it is a quantitative association rule.

5. **Based on the kinds of rules to be mined:** Frequent pattern analysis can generate various kinds of rules and other interesting relationships. Association rule mining cangenerate a large number of rules, many of which are redundant or do not indicatea correlation relationship among itemsets. The discovered associations can be further analyzed to uncover statistical correlations, leading to correlation rules.

6. **Based on the kinds of patterns to be mined:** Many kinds of frequent patterns can be mined from different kinds of data sets. Sequential pattern mining searches for frequent subsequences in a sequence data set, where a sequence records an ordering of events.

For example, with sequential pattern mining, we can study the order in which items are frequently purchased. For instance, customers may tend to first buy a PC, followed by a digital camera, and then a memory card.

- Structured pattern mining searches for frequent sub-structures in a structured data set.
- Single items are the simplest form of structure.
- Each element of an itemset may contain a subsequence, a subtree, and so on.

Therefore, structured pattern mining can be considered as the most general form of frequent pattern mining.

2.11.1 Association Rule - Market - Basket Analysis

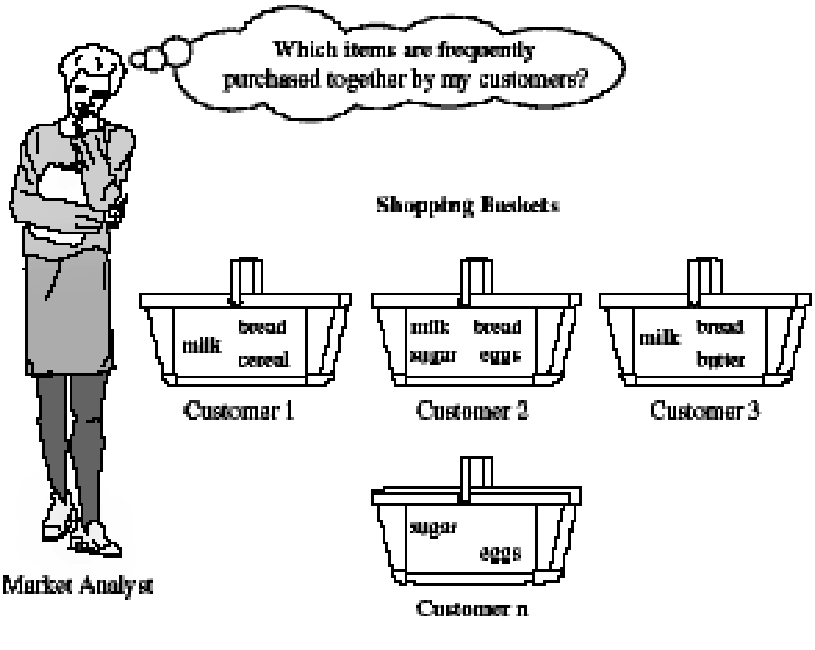
Q19. Define Association Rule using Basket Analysis ?

Ans :

A market basket is a collection of items purchased by a customer in a single transaction, which is a well-defined business activity. For example, a customer's visits to a grocery store or an online purchase from a virtual store on the Web are typical customer transactions. Retailers accumulate huge collections of transactions by recording business activities over time. One common analysis run against a transactions database is to find sets of items, or item sets, that appear together in many transactions. A business can use knowledge of these patterns to improve the Placement of these items in the store or the layout of mail- order catalog page and Web pages. An item set containing i items is called an i-item set. The percentage of transactions that contain an item set is called the item sets support. For an item set to be interesting, its support must be higher than a user-specified minimum. Such item sets are said to be frequent.

Computer→Financial_Management_Software [Support=2%, Confidence=60%]

Rule support and confidence are two measures of rule interestingness. They respectively reflect the usefulness and certainty of discovered rules. A support of 2% for association Rule means that 2% of all the transactions under analysis show that computer and financial management software are purchased together. A confidence of 60% means that 60% of the customers who purchased a computer also bought the software. Typically, association rules are considered interesting if they satisfy both a minimum support threshold and a minimum confidence threshold.



Frequent Item sets, Closed Item sets, and Association Rules

- **Association rule mining:**inding frequent patterns, associations, correlations, or causal structures among sets of items or objects in transaction databases, relational databases, and other information repositories.
- **Applications:**Basket data analysis, cross-marketing, catalog design, loss-leader analysis, clustering, classification, etc.

Examples :

- Rule form: “Body ® Head [support, confidence]”.
- buys(x, “diapers”) ® buys(x, “beers”) [0.5%, 60%]
- major(x, “CS”) ^ takes(x, “DB”) ® grade(x, “A”) [1%, 75%]

Frequent Itemsets, Closed Itemsets, and Association Rules

- A set of items is referred to as an **itemset.**
- An itemset that contains k items is a **k-itemset**.
- The set {computer, antivirus software} is a **2-itemset**.

The occurrence frequency of an itemset is the number of transactions that contain the item set. This is also known, simply, as the **frequency, support count,** or **count** of the item set.

$$\text{support } (A \Rightarrow B) = P(A \cup B)$$

$$\text{confidence } (A \Rightarrow B) = P(B/A)$$

$$\text{Confidence } (A \Rightarrow B) = P(B/A) = \frac{\text{support}(A \cup B)}{\text{support}(A)} = \frac{\text{support\_count}(A \cup B)}{\text{support\_count}(A)}$$

- Rules that satisfy both a minimum support threshold (min sup) and a minimum confidence threshold (min conf) are called **Strong Association Rules.**

2.11.2 Mining Various Kinds of Association Rules

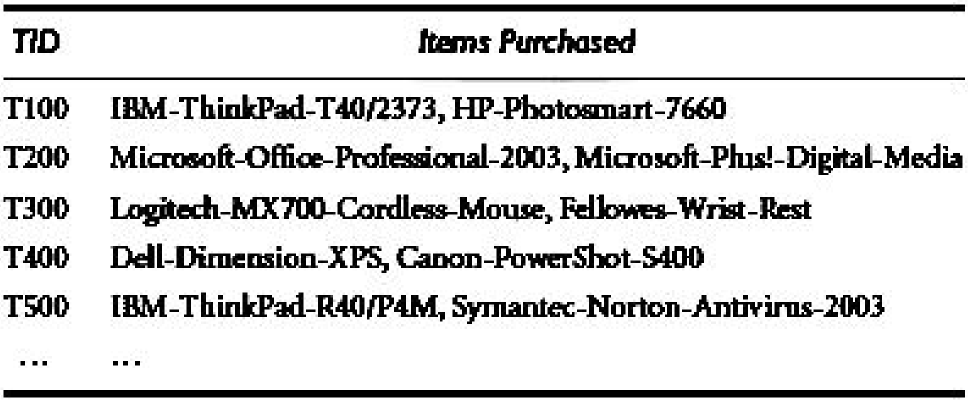
Q20. Mention Different types of Association Rules?

Ans :

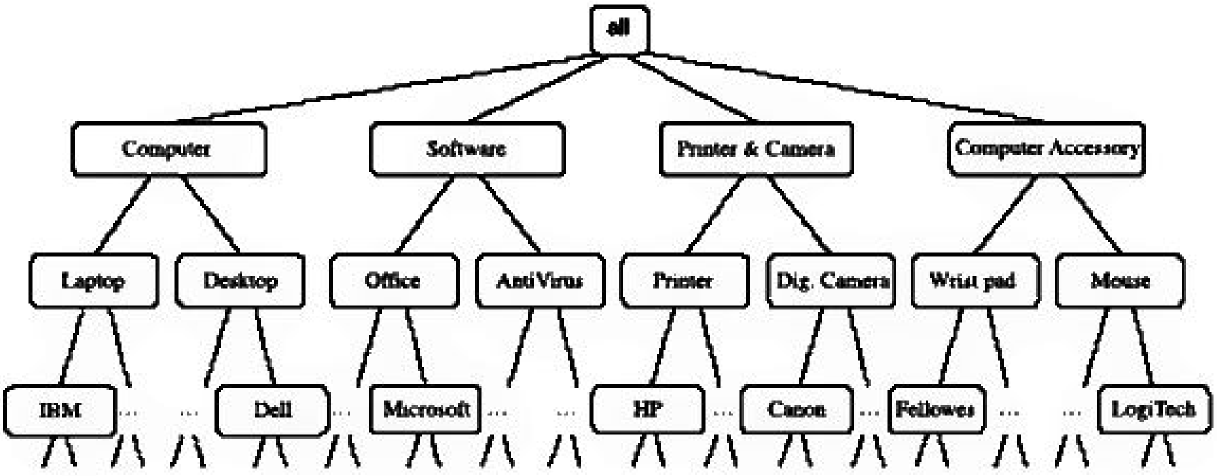
1. Mining Multilevel Association Rules

For many applications, it is difficult to find strong associations among data items at low or primitive levels of abstraction due to the sparsity of data at those levels. Strong associations discovered at high levels of abstraction may represent commonsense knowledge. Moreover, what may represent common sense to one user may seem novel to another. Therefore, data mining systems should provide capabilities for mining association rules at multiple levels of abstraction, with sufficient flexibility for easy traversal among different abstraction spaces.

Let’s examine the following example. Mining multi-level association rules. Suppose we are given the task-relevant set of transactional data in Table for sales in an AllElectronics store, showing the items purchased for each transaction. The concept hierarchy for the items. A concept hierarchy defines a sequence of mappings from a set of low-level concepts to higher level, more general concepts. Data can be generalized by replacing low-level concepts within the data by their higher-level concepts, or ancestors, from a concept hierarchy.



TID	Items Purchased
T100	IBM-ThinkPad-T40/2373, HP-Photosmart-7660
T200	Microsoft-Office-Professional-2003, Microsoft-Plus!-Digital-Media
T300	Logitech-MX700-Cordless-Mouse, Fellowes-Wrist-Rest
T400	Dell-Dimension-XPS, Canon-PowerShot-S400
T500	IBM-ThinkPad-R40/P4M, Symantec-Norton-Antivirus-2003
...	...

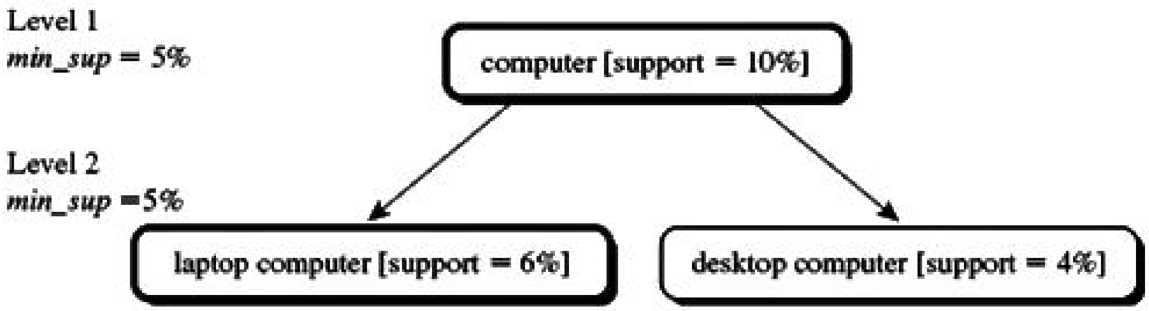


Association rules generated from mining data at multiple levels of abstraction are called multiple-level or multilevel association rules. Multilevel association rules can be mined efficiently using concept hierarchies under a support confidence framework. In general, a top-down strategy is employed, where counts are accumulated for the calculation of frequent itemsets at each concept level, starting at the concept level 1 and working downward in the hierarchy toward the more specific concept levels, until no more frequent item sets can be found. For each level, any algorithm for discovering frequent itemsets may be used, such as Apriori or its variations.

2. Using uniform minimum support for all levels (referred to as uniform support)

The same minimum support threshold is used when mining at each level of abstraction. A minimum support threshold of 5% is used throughout (e.g., for mining from “computer” down to “laptop computer”). Both “computer” and “laptop computer” are found to be frequent, while “desktop computer” is not.

When a uniform minimum support threshold is used, the search procedure is simplified. The method is also simple in that users are required to specify only one minimum support threshold. An Apriori-like optimization technique can be adopted, based on the knowledge that an ancestor is a superset of its descendants: The search avoids examining itemsets containing any item whose ancestors do not have minimum support.



Using reduced minimum support at lower levels (referred to as reduced support): Each level of abstraction has its own minimum support threshold. The deeper the level of abstraction, the smaller the corresponding threshold is. For example, in Figure, the minimum support thresholds for levels 1 and 2 are 5% and 3%, respectively. In this way, "computer," "laptop computer," and "desktop computer" are all considered frequent.

Using item or group-based minimum support (referred to as group-based support): Because users or experts often have insight as to which groups are more important than others, it is sometimes more desirable to set up user-specific, item, or group based minimal support thresholds when mining multilevel rules. For example, a user could set up the minimum support thresholds based on product price, or on items of interest, such as by setting particularly low support thresholds for laptop computers and flash drives in order to pay particular attention to the association patterns containing items in these categories.

1. **Mining Multidimensional Association Rules from Relational Databases and Data Warehouses.** We have studied association rules that imply a single predicate, that is, the predicate buys. For instance, in mining our All Electronics database, we may discover the Boolean association rule

buys(X, "digital camera") $\Rightarrow$ buys(X, "HP printer").

Following the terminology used in multidimensional databases, we refer to each distinct predicate in a rule as a dimension. Hence, we can refer to Rule above as a single dimensional or intra dimensional association rule because it contains a single distinct predicate (e.g., buys)with multiple occurrences (i.e., the predicate occurs more than once within the rule). As we have seen in the previous sections of this chapter, such rules are commonly mined from transactional data. Considering each database attribute or warehouse dimension as a predicate, we can therefore mine association rules containing multiple predicates, such as

age(X, "20....29") ^ occupation(X, "student") $\Rightarrow$ buys(X, "laptop").

Association rules that involve two or more dimensions or predicates can be referred to as multidimensional association rules. Rule above contains three predicates (age, occupation, and buys), each of which occurs only once in the rule. Hence, we say that it has no repeated predicates. Multidimensional association rules with no repeated predicates are called inter dimensional association rules. We can also mine multidimensional association rules with repeated predicates, which contain multiple occurrences of some predicates. These rules are called hybrid-dimensional association rules. An example of such a rule is the following, where the predicate buys is repeated :

age(X, "20...29") ^ buys(X, "laptop") $\Rightarrow$ buys(X, "HP printer")

Note that database attributes can be categorical or quantitative. Categorical attributes have a finite number of possible values, with no ordering among the values (e.g., occupation, brand, color). Categorical attributes are also called nominal attributes, because their values are¯names of things. Quantitative attributes are numeric and have an implicit ordering among values (e.g., age, income, price). Techniques for mining multidimensional association rules can be categorized into two basic approaches regarding the treatment of quantitative attributes.

2.11.3 Data Warehouse Implementation

Q21. Mention few approaches to Mining Multi -way Computation in Data Cubes ?

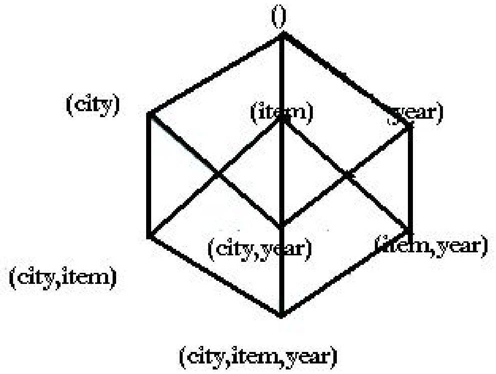
Ans :

Efficient Computation of Data Cubes

Data cube can be viewed as a lattice of cuboids, customer), (product, customer), (date), (product), (customer).

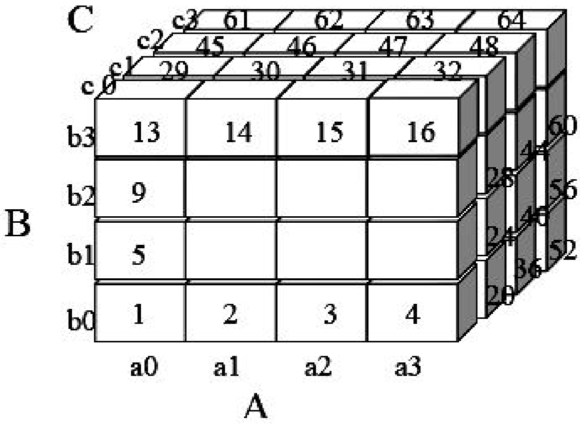
Cube Computation: ROLAP-Based Method

- Efficient cube computation methods
  - ROLAP-based cubing algorithms (Agarwal et al'96)
  - Array-based cubing algorithm (Zhao et al'97)
  - Bottom-up computation method (Bayer & Ramarkrishnan'99)
- ROLAP-based cubing algorithms
  - Sorting, hashing, and grouping operations are applied to the dimension attributes in order to reorder and cluster related tuples
- Grouping is performed on some sub aggregates as a "partial grouping step"
- Aggregates may be computed from previously computed aggregates, rather than from the base fact table.



Multi-way Array Aggregation for Cube

- Computation
- Partition arrays into chunks (a small sub cube which fits in memory).
- Compressed sparse array addressing: (chunk_id, offset)
- Compute aggregates in "multi-way" by visiting cube cells in the order which minimizes the # of times to visit each cell, and reduces memory access and storage cost.



The **base cuboid** contains all three dimensions, city, item, and year. It can return the total sales for any combination of the three dimensions. The **apex cuboid**, or 0-D cuboid, refers to the case where the group-by is empty. It contains the total sum of all sales. The base cuboid is the least generalized (most specific) of the cuboids.

The apex cuboid is the most generalized (least specific) of the cuboids, and is often denoted as all. If we start at the apex cuboid and explore downward in the lattice, this is equivalent to drilling down within the data cube. If we start at the base cuboid and explore upward.

Selected Computation of Cuboids

There are three choices for data cube materialization given a base cuboid :

1. **No materialization**: Do not precompute any of the "nonbase" cuboids. This leads to computing expensive multidimensional aggregates on-the-fly, which can be extremely slow.
2. **Full materialization**: Precompute all of the cuboids. The resulting lattice of computed cuboids is referred to as the full cube. This choice typically requires huge amounts of memory space in order to store all of the precomputed cuboids.
3. **Partial materialization**: Selectively compute a proper subset of the whole set of possible cuboids. Alternatively, we may compute a subset of the cube, which contains only those cells that satisfy some user-specified criterion, such as where the tuple count of each cell is above some threshold. We will use the term subcube to refer to the latter case, where only some of the cells may be precomputed for various cuboids. Partial materialization represents an interesting trade-off between storage space and response time.

The partial materialization of cuboids or subcubes should consider three factors :

- Identify the subset of cuboids or subcubes to materialize;
- Exploit the materialized cuboids or subcubes during query processing;
- Efficiently update the materialized cuboids or subcubes during load and refresh.

Efficient Processing of OLAP Queries

The purpose of materializing cuboids and constructing OLAP index structures is to speed up query processing in data cubes. Given materialized views, query processing should proceed as follows :

1. **Determine which operations should be performed on the available cuboids:** This involves transforming any selection, projection, roll-up (group-by), and drill-down operations specified in the query into corresponding SQL and/or OLAP operations. For example, slicing and dicing a data cube may correspond to selection and/or projection operations on a materialized cuboid.
2. **Determine to which materialized cuboid(s) the relevant operations should be applied:** This involves identifying all of the materialized cuboids that may potentially be used to answer the query, pruning the set using knowledge of "dominance" relationships among the cuboids, estimating the costs of using the remaining materialized cuboids, and selecting the cuboid with the least cost.

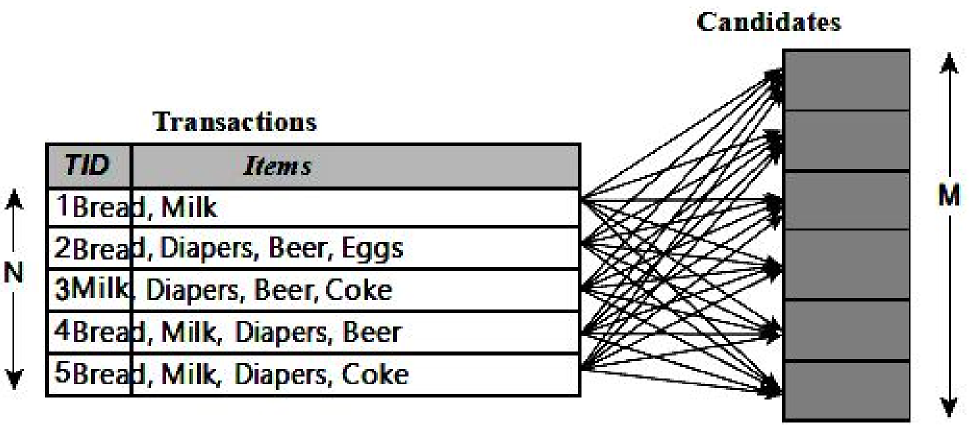
2.12 Frequent Item-set Generation

Q22. Explain FP Growth Algorithm ?

Ans :

A lattice structure can be used to enumerate the list of all possible item sets. an item set lattice for I = {a, b, c, d, e}. In general, a data set that contains k items can potentially generate up to 2k - 1 frequent item sets, excluding the null set. Because k can be very large in many practical applications, the search space of item sets that need to be explored is exponentially large.

A brute-force approach for finding frequent item sets is to determine the support count for every candidate item set in the lattice structure. To do this, we need to compare each candidate against every transaction, an operation that is shown in Figure. If the candidate is contained in a transaction, its support count will be incremented. For example, the support for {Bread, Milk} is incremented three times because the item set is contained in transactions 1, 4, and 5. Such an approach can be very expensive because it requires O(NMw) comparisons, where N is the number of transactions, M =2k - 1 is the number of candidate itemsets, and w is the maximum transaction width.



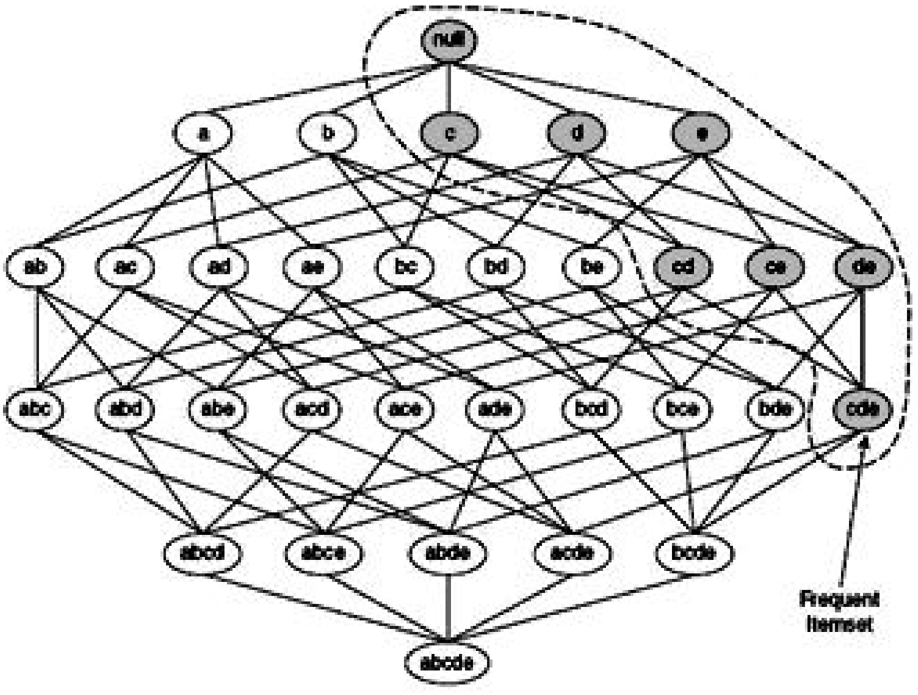
TID	Items
1	Bread, Milk
2	Bread, Diapers, Beer, Eggs
3	Milk, Diapers, Beer, Coke
4	Bread, Milk, Diapers, Beer
5	Bread, Milk, Diapers, Coke

There are several ways to reduce the computational complexity of frequent itemset generation.

1. Reduce the number of candidate itemsets (M). The Apriori principle, described in the next section, is an effective way to eliminate some of the candidate itemsets without counting their support values.
2. Reduce the number of comparisons. Instead of matching each candidate item-set against every transaction, we can reduce the number of comparisons by using more advanced data structures, either to store the candidate itemsets or to compress the data set.

The Apriori Principle

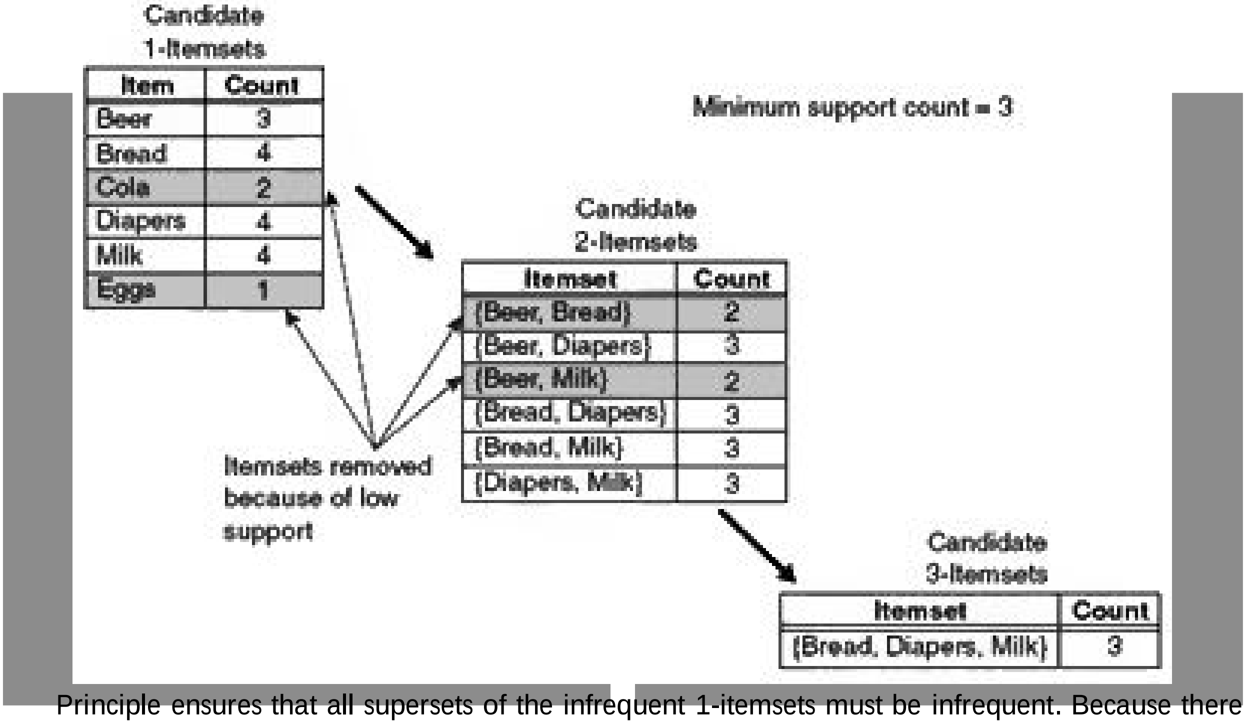
This section describes how the support measure helps to reduce the number of candidate itemsets explored during frequent itemset generation. The use of support for pruning candidate itemsets is guided by the following principle. If an itemset is frequent, then all of its subsets must also be frequent. To illustrate the idea behind the Apriori principle, consider the itemset lattice. Suppose {c, d, e} is a frequent itemset. Clearly, any transaction that contains {c, d, e} must also contain its subsets, {c, d},{c, e}, {d, e}, {c}, {d}, and {e}. As a result, if {c, d, e} is frequent, frequent.



If {c, d, e} is frequent, then all subsets of this item-set are frequent. Conversely, if an item-set such as {a, b} is infrequent, then all of its supersets must be infrequent too. The entire subgraph containing the supersets of {a, b} can be pruned immediately once {a, b} is found to be infrequent. This strategy of trimming the exponential search space based on the support measure is known as support-based pruning. Such a pruning strategy is made possible by a key property of the support measure, namely, that the support for an itemset never exceeds the support for its subsets. This property is also known as the anti-monotone property of the support measure.

Frequent Item-set Generation in the Apriori Algorithm

Apriori is the first association rule mining algorithm that pioneered the use of support-based pruning to systematically control the exponential growth of candidate itemsets.



Candidate 1-Itemsets

Item	Count
Beer	3
Bread	4
Cola	2
Diapers	4
Milk	4
Eggs	1

Minimum support count = 3

Candidate 2-Itemsets

Itemset	Count
{Beer, Bread}	2
{Beer, Diapers}	3
{Beer, Milk}	2
{Bread, Diapers}	3
{Bread, Milk}	3
{Diapers, Milk}	3

Itemsets removed because of low support

Candidate 3-Itemsets

Itemset	Count
{Bread, Diapers, Milk}	3

Principle ensures that all supersets of the infrequent 1-itemsets must be infrequent. Because there are only four frequent 1-itemsets, the number of candidate 2-itemsets generated by the algorithm is = 6.

Two of these six candidates, {Beer, Bread} and {Beer, Milk}, are subsequently found to be infrequent after computing their support values. The remaining four candidates are frequent, and thus will be used to generate candidate 3-itemsets. Without support- based pruning, there are = 20 candidate 3-itemsets that can be formed using the six items given in this example. With the Apriori principle, we only need to keep candidate 3-itemsets whose subsets are frequent. The only candidate that has this property is {Bread, Diapers,Milk}. The effectiveness of the Apriori pruning strategy can be shown by counting the number of candidate itemsets generated. A brute-force strategy of enumerating all itemsets (up to size 3) as candidates will produce.

$$\binom{6}{1}+\binom{6}{2}+\binom{6}{3} = 6 + 15 + 20 = 41$$

Candidates with the Apriori principle, this number decreases to

$$\binom{6}{1}+\binom{4}{2}+1=6+6+1=13$$

Let Ck denote the set of candidate k-itemsets and Fk denote the set of frequent k- itemsets :

- The algorithm initially makes a single pass over the data set to determine the support of each item. Upon completion of this step, the set of all frequent 1-itemsets, F1, will be known (steps 1 and 2).
- Next, the algorithm will iteratively generate new candidate k-itemsets using the frequent (k- 1)-item-sets found in the previous iteration (step 5). Candidate generation is implemented using a function called apriorigen

Frequent itemset generation o the Apriori algorithm

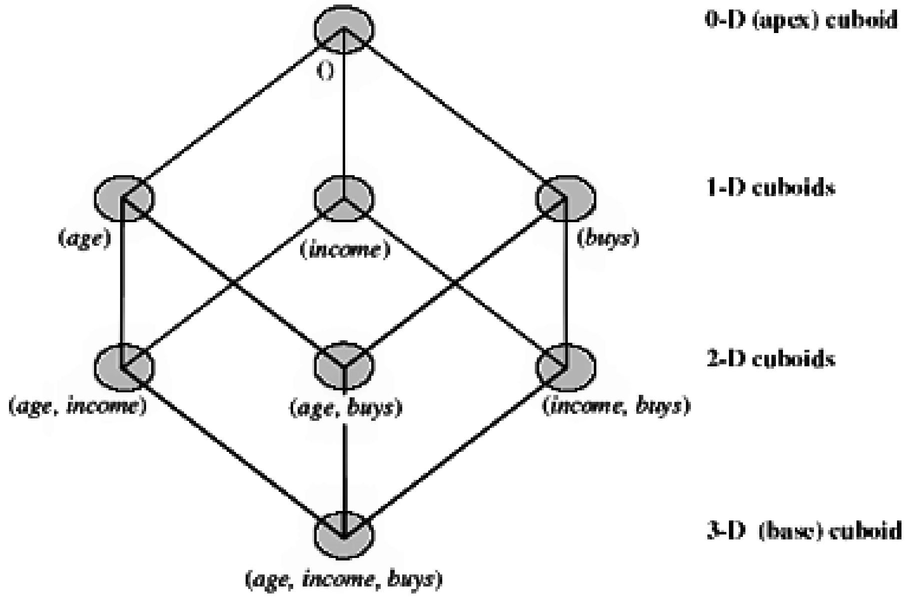
```
1.  k = 1
2.  $F_k = \{i \mid i \in I \wedge \sigma(\{i\}) \geq N \times \text{min sup}\}$.    {Find all frequent : 1-itemsets}
3.  repeat
4.     k = k+1
5.     $C_k$ = apriori-gen ($F_{k-1}$).   {Generate candidate itemsets}
6.     for each transaction $t \in T$ do
7.       $C_t$ = subset ($C_k$, t).   {Identify all candidates that belong to  t}
8.       for each candidate itemset $c \in C_t$ do
9.          $\sigma(c) = c(\sigma)+1$.       {Internet support count}
10.      end for
11. end for
12. $F_k = \{c \mid c \in C_k \wedge \sigma(c) \geq N \times \text{min sup}\}$   [Extract the frequent  k-itemsets}
13. until  Fk = $\phi$
14. Result = $\bigcup F_k$.
```

2.12.1 Mining Multidimensional Association Rules Using Static Discretization of Quantitative Attributes

Q23. Mention few approaches to mining Multi level Association Rules ?

Ans :

Quantitative attributes, in this case, are discretized before mining using predefined concept hierarchies or data discretization techniques, where numeric values are replaced by interval labels. Categorical attributes may also be generalized to higher conceptual levels if desired. If the resulting task-relevant data are stored in a relational table, then any of the frequent itemset mining algorithms we have discussed can be modified easily so as to find all frequent predicate sets rather than frequent itemsets. In particular, instead of searching on only one attribute like buys, we need to search through all of the relevant attributes, treating each attribute-value pair as an itemset.



Mining Quantitative Association Rules

Quantitative association rules are multidimensional association rules in which the numeric attributes are dynamically discretized during the mining process so as to satisfy some mining criteria, such as maximizing the confidence or compactness of the rules mined. In this section, we focus specifically on how to mine quantitative association rules having two quantitative attributes on the left-hand side of the rule and one categorical attribute on the right-hand side of the rule. That is,

$$A_{quan1} \wedge A_{quan2} \Rightarrow A_{cat}$$

where Aquan1 and Aquan2 are tests on quantitative attribute intervals (where the intervals are dynamically determined), and Acat tests a categorical attribute from the task-relevant data. Such rules have been referred to as two-dimensional quantitative association rules, because they contain two quantitative dimensions. For instance, suppose you are curious about the association relationship between pairs of quantitative attributes, like customer age and income, and the type of television (such as high-definition TV, i.e., HDTV) that customers like to buy. An example of such a 2-D quantitative association rule Is

age(X, “30...39”) $\wedge$ income(X, “42K....48K”) $\Rightarrow$ buys(X, “HDTV”)

Binning: Quantitative attributes can have a very wide range of values defining their domain. Just think about how big a 2-D grid would be if we plotted age and income as axes, where each possible value of age was assigned a unique position on one axis, and similarly, each possible value of income was assigned a unique position on the other axis! To keep grids down to a manageable size, we instead partition the ranges of quantitative attributes into intervals. These intervals are dynamic in that they may later be further combined during the mining process. The partitioning process is referred to as binning, that is, where the intervals are considered –bins. Three common binning strategies area as follows :

Three common binning strategies area as follows :

- Equal-width bining, where the interval size of each bin is the same
- Equal-frequency binning, where each bin has approximately the same number of tuples assigned to it.

- Clustering-based binning, where clustering is performed on the quantitative attribute to group neighbouring points(judged based on various distance measures) into the same bin.

Finding frequent predicate sets: Once the 2-D array containing the count distribution for each category is set up, it can be scanned to find the frequent predicate sets (those satisfying minimum support) that also satisfy minimum confidence. Strong association rules can then be generated from these predicate sets, using a rule generation algorithm.

Clustering the association rules : The strong associationrules obtained in the previous step are then mapped to a 2-D grid. Figures shows a 2-D grid for 2-D quantitative association rules predicting the condition buy (X, "HDTV") on the rule right-hand side, given the quantitative attributes age and income. The four Xs correspond to the rules

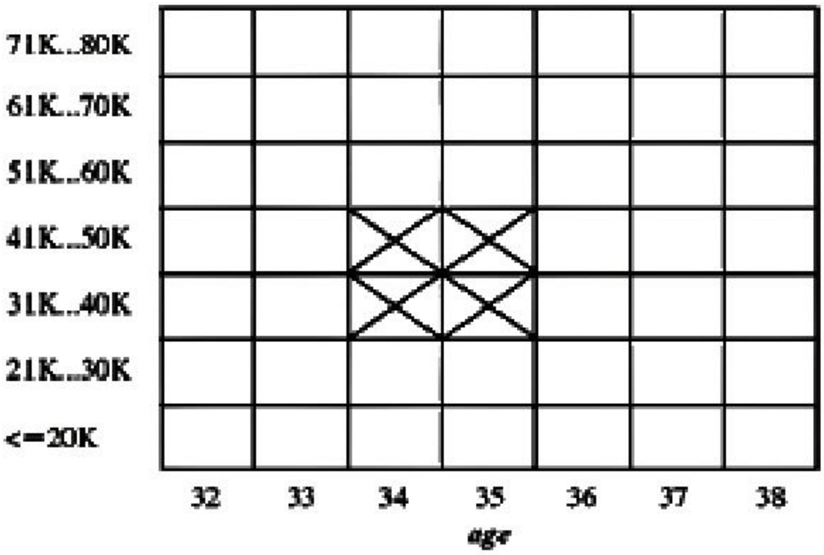
age(X, 34) $\wedge$ income(X, "31K...40K") $\Rightarrow$ buys(X, "HDTV") (5.16)

age(X, 35) $\wedge$ income(X, "31K...40K") $\Rightarrow$ buys(X, "HDTV") (5.17)

age(X, 34) $\wedge$ income(X, "41K...50K") $\Rightarrow$ buys(X, "HDTV") (5.18)

age(X, 35) $\wedge$ income(X, "41K...50K") $\Rightarrow$ buys(X, "HDTV") (5.19)

"Can we find a simpler rule to replace the above four rules?" Notice that these rules are quite "close" to one another, forming a rule cluster on the grid. Indeed, the four rules can be combined or "clustered" together to form the following simpler rule, which subsumes and replaces the above four rules :



2.12.2 Association Mining to Correlation Analysis

Q24. Explain Association Rule Mining on Correlation Analysis?

Ans :

Most association rule mining algorithms employ a support-confidence framework. Often, many interesting rules can be found using low support thresholds. Although minimum support and confidence thresholds help weed out or exclude the exploration of a good number of uninteresting rules, many rules so generated are still not interesting to the users.

Unfortunately, this is especially true when mining at low support thresholds or mining for long patterns. This has been one of the major bottlenecks for successful application of association rule mining.

Strong Rules Are Not Necessarily Interesting : An Example

Whether or not a rule is interesting can be assessed either subjectively or objectively. Ultimately, only the user can judge if a given rule is interesting, and this judgment, being subjective, may differ from one user to another. However, objective interestingness measures, based on the statistics ¯behind the data, can be used as one step toward the goal of weeding out uninteresting rules from presentation to the user.

The support and confidence measures are insufficient at filtering out uninteresting association rules. To tackle this weakness, a correlation measure can be used to augment the support-confidence framework for association rules. This leads to correlation rules of the form

$A \Rightarrow B$ [support, confidence. correlation].

That is, a correlation rule is measured not only by its support and confidence but also by the correlation between item-sets A and B. There are many different correlation measures from which to choose. In this section, we study various correlation measures to determine which would be good for mining large data sets.

Constraint-Based Association Mining

A data mining process may uncover thousands of rules from a given set of data, most of which end up being unrelated or uninteresting to the users. Often, users have a good sense of which ¯direction of mining may lead to interesting patterns and the ¯form of the patterns or rules they would like to find. Thus, a good heuristic is to have the users specify such intuition or expectations as constraints to confine the search space. This strategy is known as constraint-based mining.

The constraints can include the following :

- **Knowledge type constraints :** These specify the type of knowledge to be mined, such as azssociation or correlation.
- **Data constraints :** These specify the set of task-relevant data
- **Dimension/Level cosntraints :** These specify the desired dimensions (or attributes) of the data, or levels of the concept hierarchies, to be used in mining.
- **Interestingness constraints :** These specify thresholds on statistical measures of rule interestingness, such as support, confidence, and correlation.
- **Rule constraints :** These specify the form of rules to be mined. Such constraints may be expressed as metarules (rule templates), as the maximum or minimum number of predicates that can occur int he rule antecedent or consequent, or as relationships among attributes, attribute values, and/or aggregates.

Meta rule-Guided Mining of Association Rules

"How are meta rules useful?" Meta rules allow users to specify the syntactic form of rules that they are interested inmining. The rule forms can be used as constraints to help improve the efficiency of the mining process. Metarules may be based on the analyst's experience, expectations, or intuition regarding the data or may be automatically generated based on the database schema.

Metarule-guided mining:- Suppose that as a market analyst for All Electronics, you have access to the data describing customers (such as customer age, address, and credit rating) as well as the list of customer transactions. You are interested in finding associations between customer traits and the items that customers buy. However, rather than finding all of the association rules reflecting these relationships, you are particularly interested only in determining which pairs of customer traits promote the sale of office software. A meta rule can be used to specify this information describing the form of rules you are interested in finding. An example of such a meta rule is

$P_1(X, Y) \wedge P_2(X, W) \Rightarrow$ buys(X, "office software"),

where P1 and P2 are predicate variables that are instantiated to attributes from the given database during the mining process, X is a variable representing a customer, and Y and W take on values of the attributes assigned to P1 and P2, respectively. Typically, a user will specify a list of attributes to be considered for instantiation with P1 and P2. Otherwise, a default set may be used.

Constraint Pushing: Mining Guided by Rule Constraints

Rule constraints specify expected set/subset relationships of the variables in the mined rules, constant initiation of variables, and aggregate functions. Users typically employ their knowledge of the application or data to specify rule constraints for the mining task. These rule constraints may be used together with, or as an alternative to, metarule-guided mining. In this section, we examine rule constraints as to how they can be used to make the mining process more efficient.

Let's study an example where rule constraints are used to mine hybrid-dimensional association rules.

Our association mining query is to "Find the sales of which cheap items (where the sum of the prices is less than $100) may promote the sales of which expensive items (where the minimum price is $500) of the same group for Chicago customers in 2004." This can be expressed in the DMQL data mining query language as follows,

1) mine association as

2) lives_in(C, → "Chicago") $\wedge$ sales$^+$(C, ?{*I*},{S}) $\Rightarrow$ sales$^+$(C, ?{J},{T})

3) from sales

4) where S.year = 2004 and T.year = 2004 and I.group = J.group

5) group by C, I.group

6) having sum(I.price) $<$ 100 and min(J.price) $\geq$ 500

7) with support threshold = 1%

8) with confidence threshold = 50%

2.12.3 Alternative Methods for Generating Frequent Itemsets

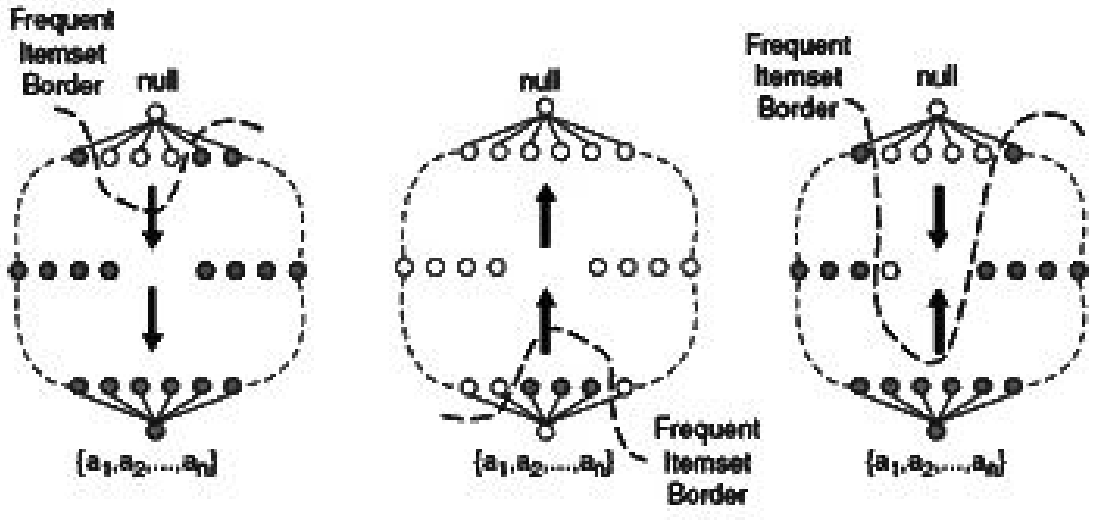
Q25. What are the alternative methods used in Frequent Item Sets ?

Ans :

Apriori is one of the earliest algorithms to have successfully addressed the combinatorial explosion of frequent itemset generation. It achieves this by applying the Apriori principle to prune the exponential search space. Despite its significant performance improvement, the algorithm still incurs considerable I/O overhead since it requires making several passes over the transaction data set.

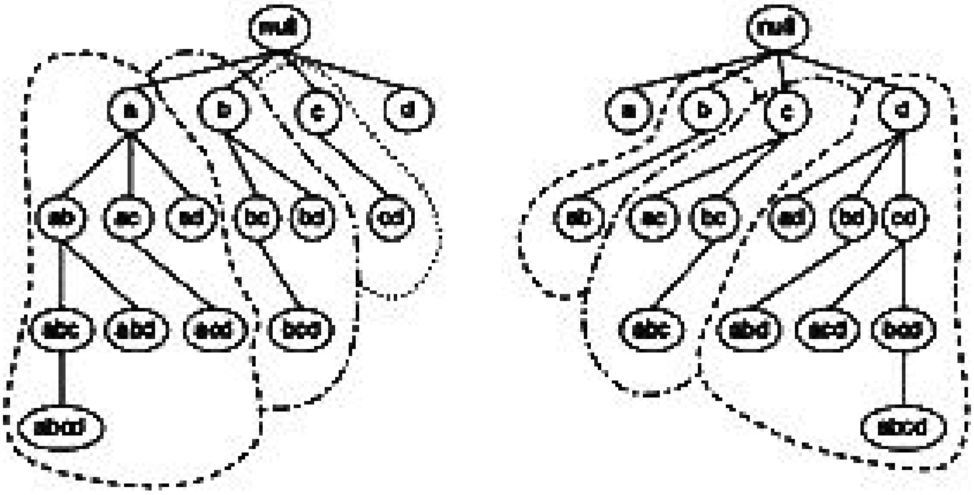
- **General-to-Specific versus Specific-to-General:** The Apriori algorithm uses a general-to-specific search strategy, where pairs of frequent (k-1)-itemsets are merged to obtain candidate k-itemsets. This general-to-specific search strategy is effective, provided the maximum length of a frequent itemset is not too long. The configuration of frequent item-sets that works best with this strategy where the darker nodes represent infrequent itemsets.

  Alternatively, a specific-to-general search strategy looks for more specific frequent itemsets first, before finding the more general frequent itemsets. This strategy is use-ful to discover maximal frequent itemsets in dense transactions, where the frequent itemset border is located near the bottom of the lattice, The Apriori principle can be applied to prune all subsets of maximal frequent itemsets. Specifically, if a candidate k-itemset is maximal frequent, we do not have to examine any of its subsets of size k - 1. However, if the candidate k-itemset is infrequent, we need to check all of its k - 1 subsets in the next iteration. Another approach is to combine both general-to-specific more space to and specific-to-general search strategies. This bidirectional approach requires

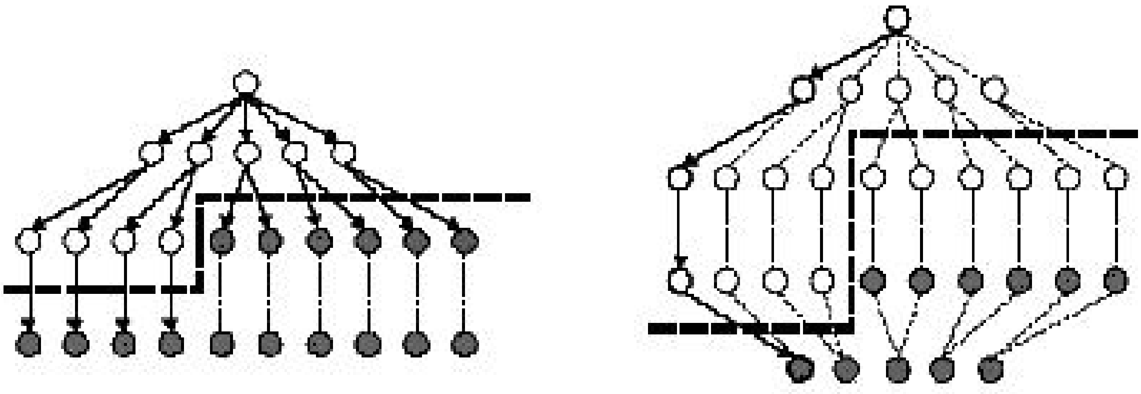


(a) General-to-specific (b) Specific-to-general (c) Bidirectional

- **Equivalence Classes:** Another way to envision the traversal is to first partition the lattice into disjoint groups of nodes (or equivalence classes). A frequent itemset generation algorithm searches for frequent itemsets within a particular equivalence class first before moving to another equivalence class. As an example, the level-wise strategy used in the Apriori algorithm can be considered to be partitioning the lattice on the basis of itemset sizes;
- **Breadth-First versus Depth-First:** The Apriori algorithm traverses the lattice in a breadth- first manner, It first discovers all the frequent 1-itemsets, followed by the frequent 2-itemsets, and so on, until no new frequent itemsets are generated.



(a) Prefix tree. (b) Suffix tree.



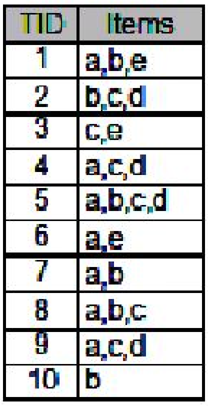
(a) Breadth first

(b) Depth First

Representation of Transaction Data Set

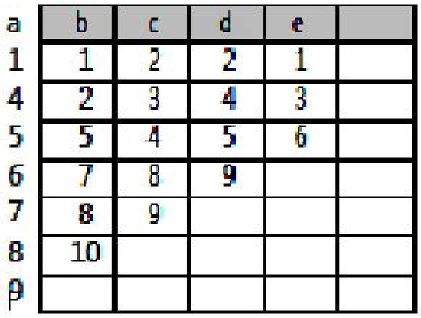
There are many ways to represent a transaction data set. The choice of representation can affect the I/O costs incurred when computing the support of candidate itemsets. two different ways of representing market basket transactions. The representation on the left is called a **horizontal** data layout, which is adopted by many association rule mining algorithms, including Apriori. Another possibility is to store the list of transaction identifiers (TID-list) associated with each item. Such a representation is known as the **vertical** data layout. The support for each candidate itemset is obtained by intersecting the TID-lists of its subset items. The length of the TID-lists shrinks as we progress to larger sized itemsets.

Horizontal Data Layout



TID	Items
1	a,b,e
2	b,c,d
3	c,e
4	a,c,d
5	a,b,c,d
6	a,e
7	a,b
8	a,b,c
9	a,c,d
10	b

Vertical Data Layout



a	b	c	d	e	
1	1	2	2	1	
4	2	3	4	3	
5	5	4	5	6	
6	7	8	9		
7	8	9			
8	10				
9					

However, one problem with this approach is that the initial set of TID-lists may be too large to fit into main memory, thus requiring more sophisticated techniques to compress the TID-lists.

Short Answers

1. Define Data Warehouse

A data warehouse is a collection of data marts representing historical data from different operations in the company. This data is stored in a structure optimized for querying and data analysis as a data warehouse. Table design, dimensions and organization should be consistent throughout a data warehouse so that reports or queries across the data warehouse are consistent. A data warehouse can also be viewed as a database for historical data from different functions within a company.

The term Data Warehouse was coined by Bill Inmon in 1990, which he defined 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. It is a single, complete and consistent store of data obtained from a variety of different sources made available to end users in what they can understand and use in a business context. It can be
- Used for decision Support
- Used to manage and control business
- Used by managers and end-users to understand the business and make judgments
- Data Warehousing is an architectural construct of information systems that provides users with current and historical decision support information that is hard to access or present in traditional operational data stores.

2. Benefits of Data Warehousing

- Data warehouses are designed to perform well with aggregate queries running on large amounts of data.
- The structure of data warehouses is easier for end users to navigate, understand and query against unlike the relational databases primarily designed to handle lots of transactions.
- Data warehouses enable queries that cut across different segments of a company's operation. E.g. production data could be compared against inventory data even if they were originally stored in different databases with different structures.
- Queries that would be complex in very normalized databases could be easier to build and maintain in data warehouses, decreasing the workload on transaction systems.
- Data warehousing is an efficient way to manage and report on data that is from a variety of sources, non-uniform and scattered throughout a company.
- Data warehousing is an efficient way to manage demand for lots of information from lots of users.
- Data warehousing provides the capability to analyze large amounts of historical data for nuggets of wisdom that can provide an organization with competitive advantage.

3. Data Mart and Virtual Warehouse

Data mart :

- A data mart contains a subset of corporate-wide data that is of value to a specific group of users. The scope is confined to specific selected subjects. For example, a marketing data mart may confine its subjects to customer, item, and sales. The data contained in datamarts tend to be summarized.
- Data marts are usually implemented on low-cost departmental servers that are UNIX/LINUX- or Windows-based. The implementation cycle of a data mart is more likely to be measured in weeks rather than months or years. However, it may involve complex integration in the long run if its design and planning were not enterprise-wide.
- Depending on the source of data, data marts can be categorized as independent or dependent. Independent data marts are sourced from data captured from one or more operational systems or external information providers, or from data generated locally within a particular department or geographic area. Dependent data marts are sourced directly from enterprise data warehouses.

Virtual warehouse:

- A virtual warehouse is a set of views over operational databases. For efficient query processing, only some of the possible summary views may be materialized.
- A virtual warehouse is easy to build but requires excess capacity on operational database servers.

4. Types of OLAP Tools

Relational OLAP (ROLAP):

- ROLAP works directly with relational databases. The base data and the dimension tables are stored as relational tables and new tables are created to hold the aggregated information. It depends on a specialized schema design.
- 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.
- ROLAP tools do not use pre-calculated data cubes but instead pose the query to the standard relational database and its tables in order to bring back the data required to answer the question.
- ROLAP tools feature the ability to ask any question because the methodology does not limit to the contents of a cube. ROLAP also has the ability to drill down to the lowest level of detail in the database.

Multidimensional OLAP (MOLAP) :

- MOLAP is the 'classic' form of OLAP and is sometimes referred to as just OLAP.
- MOLAP stores this data in optimized multi-dimensional array storage, rather than in a relational database. Therefore it requires the pre-computation and storage of information in the cube - the operation known as processing.
- MOLAP tools generally utilize a pre-calculated data set referred to as a data cube. The data cube contains all the possible answers to a given range of questions.
- MOLAP tools have a very fast response time and the ability to quickly write back data into the data set.

Hybrid OLAP (HOLAP):

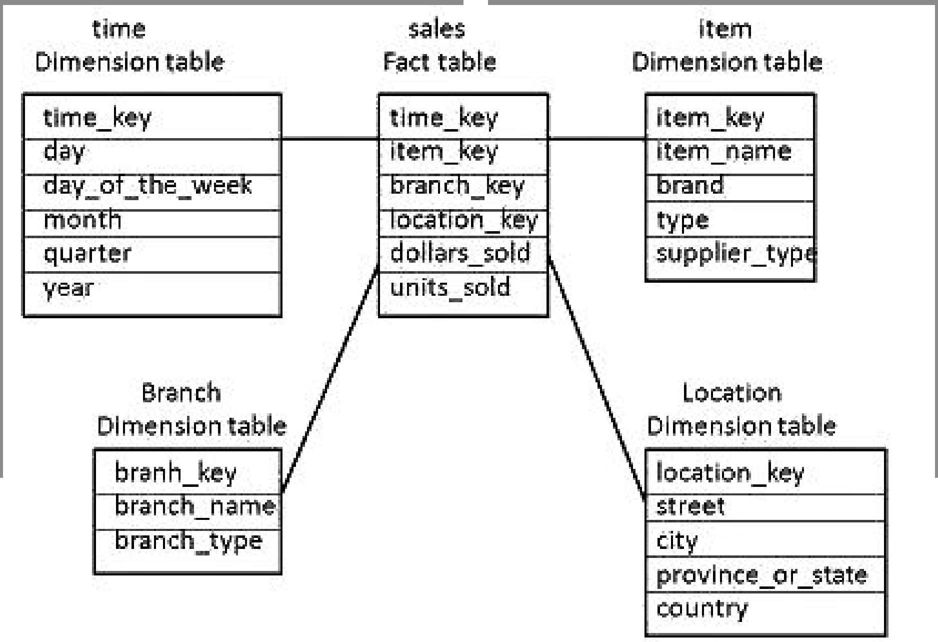
- There is no clear agreement across the industry as to what constitutes Hybrid OLAP, except that a database will divide data between relational and specialized storage.
- For example, for some vendors, a HOLAP database will use relational tables to hold the larger quantities of detailed data, and use specialized storage for at least some aspects of the smaller quantities of more-aggregate or less-detailed data.

- HOLAP addresses the shortcomings of MOLAP and ROLAP by combining the capabilities of both approaches.
- HOLAP tools can utilize both pre-calculated cubes and relational data sources.

5. Star Schema Data Warehouse

- Each dimension in a star schema is represented with only one-dimension table.
- This dimension table contains the set of attributes.
- The following diagram shows the sales data of a company with respect to the four dimensions, namely time, item, branch, and location.
- There is a fact table at the center. It contains the keys to each of four dimensions.
- The fact table also contains the attributes, namely dollars sold and units sold.

Note : Each dimension has only one dimension table and each table holds a set of attributes. For example, the location dimension table contains the attribute set {location_key, street, city, province_or_state,country}. This constraint may cause data redundancy. For example, "Vancouver" and "Victoria" both the cities are in the Canadian province of British Columbia. The entries for such cities may cause data redundancy along the attributes province_or_state and country.



6. Different OLAP Operations

In the multidimensional model, data areorganized into multiple dimensions, and each dimension contains multiple levels of abstraction defined by concept hierarchies. This organization provides users with the flexibility to view data from different perspectives. A number of OLAP data cube operations exist to materialize these different views, allowing interactive querying and analysis of the data at hand.

- **Roll-up:** The roll-up operation (also called the drill-up operation by some vendors) performs aggregation on a data cube, either by climbing up a concept hierarchy for a dimension or by dimension reduction. When roll-up is performed by dimension reduction, one or more dimensions are removed from the given cube. For example, consider a sales data cube containing only the location and time dimensions. Roll-up may be performed by removing, say, the time dimension, resulting in an aggregation of the total sales by location, rather than by location and by time.

- **Drill-down:** Drill-down is the reverse of roll-up. It navigates from less detailed data to more detailed data. Drill-down can be realized by either stepping down a concept hierarchy for a dimension or introducing additional dimensions. a drill-down adds more detail to the given data, it can also be performed by adding new dimensions to a cube.
- **Slice and dice:** The slice operation performs a selection on one dimension of the given cube,
- **Pivot (rotate):** Pivot (also called rotate) is a visualization operation that rotates the data axes in view to provide an alternative data presentation.

7. Data Generalization

Data Generalization is the process of creating successive layers of summary data in an evaluational database. It is a process of zooming out to get a broader view of a problem, trend or situation. It is also known as rolling-up data.

There are millions and millions of data stored in the database and this number continues to increase everyday as a company heads for growth. In fact, a group of process of process called extract, transform, load (ETL) is periodically performed in order to manage data within the data warehouse.

A data warehouse is a rich repository of data, most of which are historical data from a company. But in modern data warehouses, data could come from other sources. Having data from several sources greatly helps in the overall business intelligence system of a company. With diverse data sources, the company can have a broader perspective not just about the trends and pattern within the organization but of the global industrial trends are well.

In order to get a view of trends and patterns based on the analytical outputs of the business intelligence system can be a daunting task. With those millions of data, most of which disparate (but of course ironed out by the ETL process), it may be difficult to generate reports.

Dealing alone with big volumes of data for consistent delivery of business critical applications can already affect the network management tools of a company. Many companies have found that existing network management tools could hardly cope up with the great bulk of data required by the organization to monitor network and applications usage.

8. Data Characterization

The **attribute-oriented induction (AOI)** approach to concept description was first proposed in 1989, a few years before the introduction of the data cube approach. The data cube approach is essentially based on materialized views of the data, which typically have been precomputed in a data warehouse. In general, it performs offline aggregation before an OLAP or data mining query is submitted for processing. On the other hand, the attribute-oriented induction approach is basically a query-oriented, generalization-based, online data analysis technique. Note that there is no inherent barrier distinguishing the two approaches based on online aggregation versus offline precomputation. Some aggregations in the data cube can be computed online, while offline precomputation of multidimensional space can speed up attribute-oriented induction as well.

The generalization is performed by either attribute removal or attribute generalization. Aggregation is performed by merging identical generalized tuples and accumulating their respective counts. This reduces the size of the generalized data set.

Attribute-oriented induction. Here we show how attribute-oriented induction is performed on the initial working relation of Table 4.5. For each attribute of the relation, the generalization proceeds as follows :

1. **name:** Since there are a large number of distinct values for name and there is no generalization operation defined on it, this attribute is removed.

2. **gender:** Since there are only two distinct values for gender, this attribute is retained and no generalization is performed on it.

3. **major:** Suppose that a concept hierarchy has been defined that allows the attribute major to be generalized to the values farts&sciences, engineering, businessg. Suppose also that the attribute generalization threshold is set to 5, and that there are more than 20 distinct values for major in the initial working relation. By attribute generalization and attribute generalization control, major is therefore generalized by climbing the given concept hierarchy.

4. **birth place:** This attribute has a large number of distinct values; therefore, we would like to generalize it. Suppose that a concept hierarchy exists for birth place, defined as "city < province or state < country." If the number of distinct values for country in the initial working relation is greater than the attribute generalization threshold, then birth place should be removed, because even though a generalization operator exists for it, the generalization threshold would not be satisfied. If, instead, the number of distinct values for country is less than the attribute generalization threshold, then birth place should be generalized to birth country.

5. **birth date:** Suppose that a hierarchy exists that can generalize birth date to age and age to age range, and that the number of age ranges (or intervals) is small with respect to the attribute generalization threshold. Generalization of birth date should therefore take place.

6. **residence:** Suppose that residence is defined by the attributes number, street, residence city, residence province or state, and residence country. The number of distinct values for number and street will likely be very high, since these concepts are quite low level. The attributes number and street should therefore be removed so that residence is then generalized to residence city, which contains fewer distinct values.

7. **phone#:** As with the name attribute, phone# contains too many distinct values and should therefore be removed in generalization.

8. **gpa:** Suppose that a concept hierarchy exists for gpa that groups values for grade point average into numeric intervals like f3.75–4.0, 3.5–3.75, . . . g, which in turn are grouped into descriptive values such as f"excellent", "very good", . . . g. The attribute can therefore be generalized. gender major birth country age range residence city gpa count

9. How to Implement DATA CUBE Technology

1. **Pre-compute and store all:** This means that millions of aggregates will need to be computed and stored. Although this is the best solution as far as query response time is concerned, the solution is impractical since resources required to compute the aggregates and to store them will be prohibitively large for a large data cube. Indexing large amounts of data is also expensive.

2. **Pre-compute (and store) none:** This means that the aggregates are computed on- the-fly using the rawdata whenever a query is posed. This approach does not require additional space for storing the cube but the query response time is likely to be very poor for large data cubes.

3. **Pre-compute and store some:** This means that we pre-compute and store the most frequently queried aggregates and compute others as the need arises. We may also be able to derive some of the remaining aggregates using the aggregates that have already been computed. It may therefore be worthwhile also to pre-compute some aggregates that are not most frequently queried but help in deriving many other aggregates.

It will of course not be possible to derive all the aggregates from the pre-computed aggregates and it will be necessary to access the database (e.g the data warehouse) to compute the remaining aggregates. The more aggregates we are able to pre-compute the better the query performance.

Another related issue is where the data used by OLAP will reside. We assume that the data is stored in a data warehouse or in one or more data marts.

Data cube products use different techniques for pre-computing aggregates and storing them. They are generally based on one of two implementation models. The first model, supported by vendors of traditional relational model databases, is called the ROLAP model or the Relational OLAP model. The second model is called the MOLAP model for multidimensional OLAP. The MLOAP model provides a direct multidimensional view of the data whereas the RLOAP model provides a relational view of the multidimensional data in the form of a fact table.

- **ROLAP** uses a relational DBMS to implement an OLAP environment. It may be considered a bottom-up approach which is typically based on using a data warehouse that has been designed using a star schema.
- **MOLAP** is based on using a multi-dimensional DBMS rather than a data warehouse to store and access data. It may be considered as a top-down approach to OLAP. The multi-dimensional database systems do not have a standard approach to storing and maintaining their data. They often use special-purpose file systems or indexes that store pre-computation of all aggregations in the cube.

10. Data Cube

A data cube refers is a three-dimensional (3D) (or higher) range of values that are generally used to explain the time sequence of an image's data. It is a data abstraction to evaluate aggregated data from a variety of viewpoints. It is also useful for imaging spectroscopy as a spectrally-resolved image is depicted as a 3-D volume.

A data cube can also be described as the multidimensional extensions of two-dimensional tables. It can be viewed as a collection of identical 2-D tables stacked upon one another. Data cubes are used to represent data that is too complex to be described by a table of columns and rows. As such, data cubes can go far beyond 3-D to include many more dimensions.

A data cube is generally used to easily interpret data. It is especially useful when representing data together with dimensions as certain measures of business requirements. A cube's every dimension represents certain characteristic of the database, for example, daily, monthly or yearly sales. The data included inside a data cube makes it possible analyze almost all the figures for virtually any or all customers, sales agents, products, and much more. Thus, a data cube can help to establish trends and analyze performance.

11. OLAP Sampling Data Collection

When collecting data, we often collect only a subset of the data we would ideally like to gather. In statistics, this is known as collecting a **sample** of the data population. The resulting data are called **sample data**. Data are often sampled to save on costs, manpower, time, and materials. In many applications, the collection of the entire data population of interest is unrealistic. it is impossible to gather the opinion of everyone in the population. Most published ratings or polls rely on a data sample for analysis. The results are extrapolated for the entire population, and associated with certain statistical measures such as a confidence interval. The confidence interval tells us how reliable a result is. Statistical surveys based on sampling are a common tool in many fields like politics, healthcare, market research, and social and natural sciences.

"How effective is OLAP on sample data?"

- **First,** sample data are often sparse in the multidimensional sense. When a user drills down on the data, it is easy to reach a point with very few or no samples even when the overall sample size is large. Traditional OLAP simply uses whatever data are available to compute a query answer. To extrapolate such an answer for a population based on a small sample could be misleading: A single outlier or a slight bias in the sampling can distort the answer significantly.

- **Second,** with sample data, statistical methods are used to provide a measure of reliability (e.g., a confidence interval) to indicate the quality of the query answer as it pertains to the population. Traditional OLAP is not equipped with such tools. A sampling cube framework was introduced to tackle each of the preceding challenges.

12. Data Mining in Cube Space

Researchers have turned their attention toward **multidimensional data mining** to uncover knowledge at varying dimensional combinations and granularities. Such mining is also known as exploratory multidimensional data mining and online analytical data mining (OLAM). Multidimensional data space is huge. This analyzes and mines the data by applying various data mining techniques systematically over these regions.

There are at least four ways in which OLAP-style analysis can be fused with data mining techniques :

1. Use cube space to define the data space for mining. Each region in cube space represents a subset of data over which we wish to find interesting patterns. Cube space is defined by a set of expert-designed, informative dimension hierarchies, not just arbitrary subsets of data. Therefore, the use of cube space makes the data space both meaningful and tractable.
2. Use OLAP queries to generate features and targets for mining. The features and even the targets (that we wish to learn to predict) can sometimes be naturally defined as OLAP aggregate queries over regions in cube space.
3. Use data mining models as building blocks in a multistep mining process. Multidimensional data mining in cube space may consist of multiple steps, where data mining models can be viewed as building blocks that are used to describe the behavior of interesting data sets, rather than the end results.
4. Use data cube computation techniques to speed up repeated model construction. Multidimensional data mining in cube space may require building a model for each candidate data space, which is usually too expensive to be feasible. However, by carefully sharing computation across model construction for different candidates based on data cube computation techniques, efficient mining is achievable.

13. Association Rules (in Data Mining)

Association rules are if/then statements that help uncover relationships between seemingly unrelated data in a relational database or other information repository. An example of an association rule would be “If a customer buys a dozen eggs, he is 80% likely to also purchase milk.”

An association rule has two parts, an antecedent (if) and a consequent (then). An antecedent is an item found in the data. A consequent is an item that is found in combination with the antecedent.

Association rules are created by analyzing data for frequent if/then patterns and using the criteria support and confidence to identify the most important relationships. Support is an indication of how frequently the items appear in the database. Confidence indicates the number of times the if/then statements have been found to be true.

In data mining, association rules are useful for analyzing and predicting customer behavior. They play an important part in shopping basket data analysis, product clustering, catalog design and store layout.

Programmers use association rules to build programs capable of machine learning. Machine learning is a type of artificial intelligence (AI) that seeks to build programs with the ability to become more efficient without being explicitly programmed.

The main applications of association rule mining :

- **Basket data analysis -** is to analyze the association of purchased items in a single basket or single purchase as per the examples given above.

- **Cross marketing -** is to work with other businesses that complement your own, not competitors. For example, vehicle dealerships and manufacturers have cross marketing campaigns with oil and gas companies for obvious reasons.
- **Catalog design -** the selection of items in a business' catalog are often designed to complement each other so that buying one item will lead to buying of another. So these items are often complements or very related.

14. Frequent Itemset and Association Rule Mining

Frequent itemset mining is an interesting branch of data mining that focuses on looking at sequences of actions or events, for example the order in which we get dressed. Shirt first? Pants first? Socks second item or second shirt if wintertime? Sequence analysis is used in a lot of different areas, and is also highly useful in games for finding behavioral patterns that lead to particular behaviors, for example a player quitting a game. Here is how it works.

In frequent itemset mining, the base data takes the form of sets of instances (also called transactions) that each has a number of features (also called items). For example, a dataset of the items players bought in a social online game might contain 4 transactions as follows :

{Sword of Grungni, Shirt of Awesomeness, Pretty Pet}
{Shirt of Awesomeness, Pretty Pet, Healing Potion}
{Sword of Grungni, Healing Potion}
{Shirt of Awesomeness, Fancy Hat, Pretty Pet}

The task for the frequent itemset mining algorithm is then to find all common sets of items, defined as those itemsets that have at least a minimum support (exists at least a minimum amount of times). If the support is set to 3, the following 1-itemsets (sets of only one item) can be found in the dataset described above: {Sword of Grungni} and {Pretty Pet}.

It is also possible to find one 2-itemset: {Shirt of Awesomeness, Pretty Pet}, as three of the transactions contain both Shirt of Awesomeness and Pretty Pet. Other itemsets of the same lengths are considered non-frequent as they recur less than three times.

The original algorithm for mining frequent itemsets, which was published in 1993 by Agrawal and is still frequently used. This algorithm functions by first scanning the database to find all frequent 1-itemsets, then proceeding to find all frequent 2-itemsets, then 3-itemsets etc. At each iteration, candidate itemsets of length n are generated by joining frequent itemsets of length n– 1; the frequency of each candidate itemset is evaluated before being added to the set of frequent itemsets.

There exist several alternatives to this algorithm, e.g. the FP-growth algorithm, which finds frequent itemsets through building prefix trees.

Once a set of frequent itemsets has been found, association rules can be generated. Association rules are of the form A'!B, and could be read as "A implies B". Each association rule has support (how common the precondition is in the dataset), confidence (how often the precondition leads to the consequence in the dataset) and lift (how much more common the consequence is in instances covered by the rule compared to the whole dataset). From the dataset and frequent itemsets above, the association rule Shirt of Awesomeness '! Pretty Pet can be derived with support 3 and confidence 1, whereas the rule Shirt of Awesomeness '! Pretty Pet and Sword of Grungni only has a confidence of only 1 = 3 and so would most likely not be selected as a useful association rule.

Frequent itemset mining can be used in several different ways for understanding game data. One way is to find patterns among players. If a database is organized so that each instance describes a separate player and the (binary or ordinal) attributes of each instance describe the player's playing style (e.g.

{violent, speedrunner, cleared_level_3, dies_from_falling}), frequent itemset mining can be used to find playing style characteristics that frequently co-occur.

15. Apriori Algorithm

Apriori is an algorithm for frequent item set mining and association rule learning over transactional databases. It proceeds by identifying the frequent individual items in the database and extending them to larger and larger item sets as long as those item sets appear sufficiently often in the database. The frequent item sets determined by Apriori can be used to determine association rules which highlight general trends in the database: this has applications in domains such as market basket analysis.

The Apriori algorithm was proposed by Agrawal and Srikant in 1994. Apriori is designed to operate on databases containing transactions (for example, collections of items bought by customers, or details of a website frequentation). Other algorithms are designed for finding association rules in data having no transactions (Winepi and Minepi), or having no timestamps DNA sequencing). Each transaction is seen as a set of items (an itemset). Given a threshold C {\displaystyle C} , the Apriori algorithm identifies the item sets which are subsets of at least C {\displaystyle C} transactions in the database.

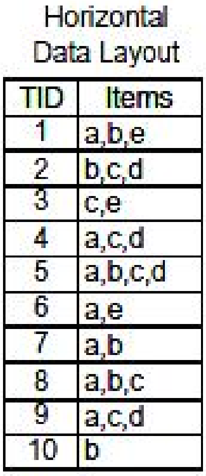
Apriori uses a “bottom up” approach, where frequent subsets are extended one item at a time (a step known as candidate generation), and groups of candidates are tested against the data. The algorithm terminates when no further successful extensions are found.

Apriori uses breadth-first search and a Hash tree structure to count candidate item sets efficiently. It generates candidate item sets of length k {\displaystyle k} from item sets of length k “ 1 {\displaystyle k-1}. Then it prunes the candidates which have an infrequent sub pattern. According to the downward closure lemma, the candidate set contains all frequent k {\displaystyle k} -length item sets. After that, it scans the transaction database to determine frequent item sets among the candidates.

The pseudo code for the algorithm is given below for a transaction database T {\displaystyle T} , and a support threshold of õ {\displaystyle \epsilon } . Usual set theoretic notation is employed, though note that T {\displaystyle T} is a multiset. C k {\displaystyle C_{k}} is the candidate set for level k {\displaystyle k} . At each step, the algorithm is assumed to generate the candidate sets from the item sets large of the preceding level, heeding the downward closure lemma. count [c] {\displaystyle count[c]} accesses a field of the data structure that represents candidate set c {\displaystyle c} , which is initially assumed to be zero. Many details are omitted below, usually the most important part of the implementation is the data structure used for storing the candidate sets, and counting their frequencies.

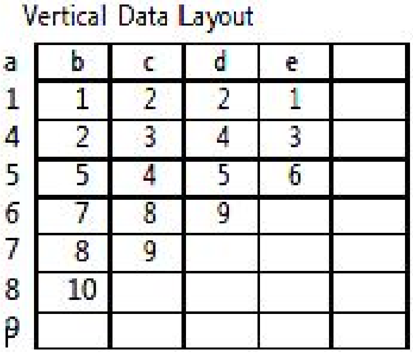
16. Representation of Transaction Data Set

There are many ways to represent a transaction data set. The choice of representation can affect the I/O costs incurred when computing the support of candidate itemsets. two different ways of representing market basket transactions. The representation on the left is called a **horizontal** data layout, which is adopted by many association rule mining algorithms, including Apriori. Another possibility is to store the list of transaction identifiers (TID-list) associated with each item. Such a representation is known as the **vertical** data layout. The support for each candidate itemset is obtained by intersecting the TID-lists of its subset items. The length of the TID-lists shrinks as we progress to larger sized itemsets.



Horizontal Data Layout

TID	Items
1	a,b,e
2	b,c,d
3	c,e
4	a,c,d
5	a,b,c,d
6	a,e
7	a,b
8	a,b,c
9	a,c,d
10	b



Vertical Data Layout

a	b	c	d	e	
1	1	2	2	1	
4	2	3	4	3	
5	5	4	5	6	
6	7	8	9		
7	8	9			
8	10				
9					

However, one problem with this approach is that the initial set of TID-lists may be too large to fit into main memory, thus requiring more sophisticated techniques to compress the TID-lists.

UNIT III

Classification : Basic Concepts, Decision Tree Induction, Bayes Classification Methods, Rule-Based Classification, classification by backpropagation, support vector machines, associative classification, lazy learners, other classification methods.Cluster Analysis: basic concepts of cluster analysis, partitioning methods, hierarchical methods, density-based methods, evaluation of clustering.

3.1 Classification and Prediction

Q1. What is classification and prediction? How Does Classification Works?

Ans :

There are two forms of data analysis that can be used for extracting models describing important classes or to predict future data trends. These two forms are as follows

- Classification
- Prediction

Classification

Classification models predict categorical class labels; and prediction models predict continuous valued functions. For example, we can build a classification model to categorize bank loan applications as either safe or risky, or a prediction model to predict the expenditures in dollars of potential customers on computer equipment given their income and occupation.

Following are the examples of cases where the data analysis task is Classification

- A bank loan officer wants to analyze the data in order to know which customer (loan applicant) is risky or which are safe.
- A marketing manager at a company needs to analyze a customer with a given profile, who will buy a new computer.

In both of the above examples, a model or classifier is constructed to predict the categorical labels. These labels are risky or safe for loan application data and yes or no for marketing data.

Prediction

Following are the examples of cases where the data analysis task is Prediction "Suppose the marketing manager needs to predict how much a given customer will spend during a sale at his company. In this example we are bothered to predict a numeric value. Therefore the data analysis task is an example of numeric prediction. In this case, a model or a predictor will be constructed that predicts a continuous-valued-function or ordered value.

Note

Regression analysis is a statistical methodology that is most often used for numeric prediction.

Classification Working Procedure

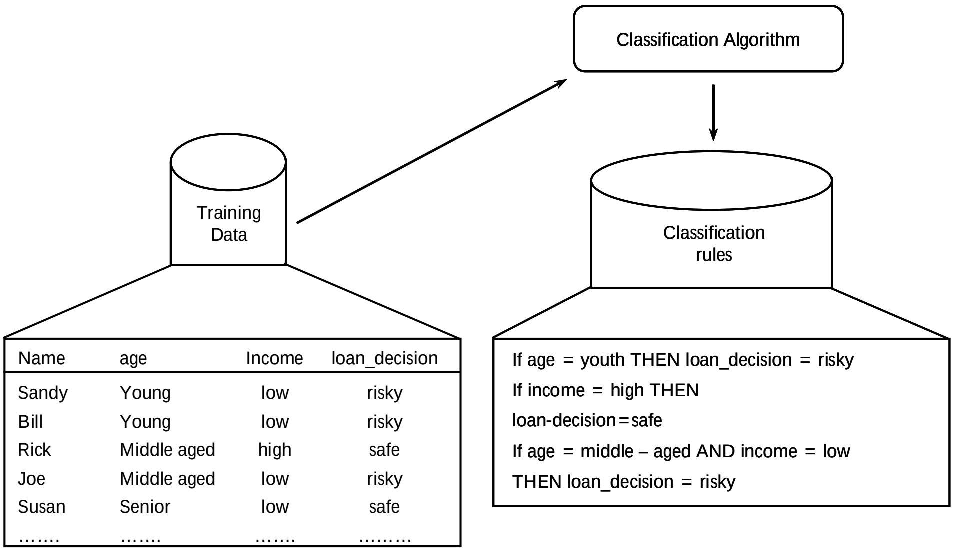
With the help of the bank loan application that we have discussed above, let us understand the working of classification. The Data Classification process includes two steps

- Building the Classifier or Model
- Using Classifier for Classification

Building the Classifier or Model

- This step is the learning step or the learning phase.
- In this step the classification algorithms build the classifier.
- The classifier is built from the training set made up of database tuples and their associated class labels.

- Each tuple that constitutes the training set is referred to as a category or class. These tuples can also be referred to as sample, object or data points.

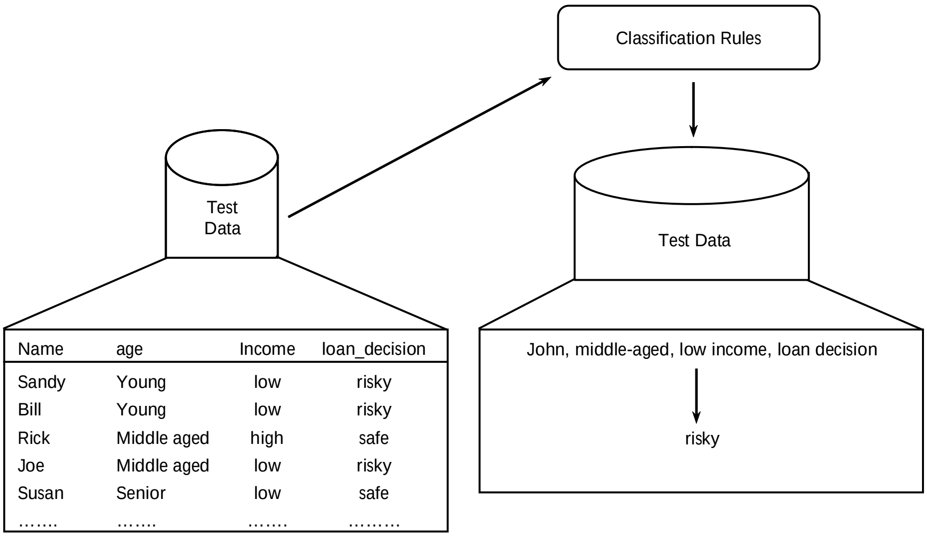


Name	age	Income	loan_decision
Sandy	Young	low	risky
Bill	Young	low	risky
Rick	Middle aged	high	safe
Joe	Middle aged	low	risky
Susan	Senior	low	safe
…….	…….	…….	………

If age = youth THEN loan_decision = risky
If income = high THEN
loan-decision=safe
If age = middle – aged AND income = low
THEN loan_decision = risky

Using Classifier for Classification

In this step, the classifier is used for classification. Here the test data is used to estimate the accuracy of classification rules. The classification rules can be applied to the new data tuples if the accuracy is considered acceptable.



Name	age	Income	loan_decision
Sandy	Young	low	risky
Bill	Young	low	risky
Rick	Middle aged	high	safe
Joe	Middle aged	low	risky
Susan	Senior	low	safe
…….	…….	…….	………

John, middle-aged, low income, loan decision
risky

Classification and Prediction Issues

The major issue is preparing the data for Classification and Prediction. Preparing the data involves the following activities :

- **Data Cleaning** : Data cleaning involves removing the noise and treatment of missing values. The noise is removed by applying smoothing techniques and the problem of missing values is solved by replacing a missing value with most commonly occurring value for that attribute.

- **Relevance Analysis** : Database may also have the irrelevant attributes. Correlation analysis is used to know whether any two given attributes are related.
- **Data Transformation and reduction** : The data can be transformed by any of the following methods.
  - **Normalization** : The data is transformed using normalization. Normalization involves scaling all values for given attribute in order to make them fall within a small specified range. Normalization is used when in the learning step, the neural networks or the methods involving measurements are used.
  - **Generalization :** The data can also be transformed by generalizing it to the higher concept. For this purpose we can use the concept hierarchies.

  Note: Data can also be reduced by some other methods such as wavelet transformation, binning, histogram analysis, and clustering.

Comparison of Classification and Prediction Methods

Here are the criteria for comparing the methods of Classification and Prediction

- **Accuracy** : Accuracy of classifier refers to the ability of classifier. It predict the class label correctly and the accuracy of the predictor refers to how well a given predictor can guess the value of predicted attribute for a new data.
- **Speed** : This refers to the computational cost in generating and using the classifier or predictor.
- **Robustness** : It refers to the ability of classifier or predictor to make correct predictions from given noisy data.
- **Scalability** : Scalability refers to the ability to construct the classifier or predictor efficiently; given large amount of data.
- **Interpretability** : It refers to what extent the classifier or predictor understands.

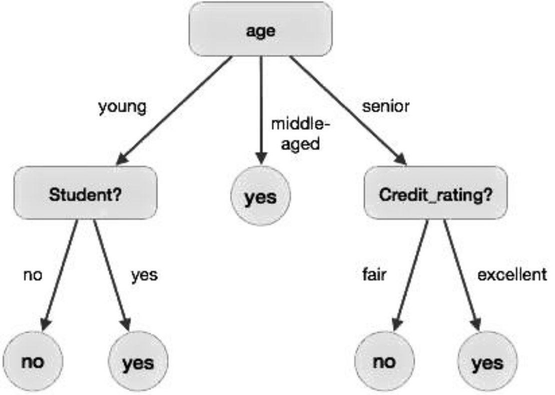
3.2 Data Mining - Decision Tree Induction

Q2. Explain Decision Tree Induction and Write algorithm?

Ans :

A decision tree is a structure that includes a root node, branches, and leaf nodes. Each internal node denotes a test on an attribute, each branch denotes the outcome of a test, and each leaf node holds a class label. The topmost node in the tree is the root node.

The following decision tree is for the concept buy_computer that indicates whether a customer at a company is likely to buy a computer or not. Each internal node represents a test on an attribute. Each leaf node represents a class.



The benefits of having a decision tree are as follows:

- It does not require any domain knowledge.
- It is easy to comprehend.
- The learning and classification steps of a decision tree are simple and fast.

Decision Tree Induction Algorithm

A machine researcher named J. Ross Quinlan in 1980 developed a decision tree algorithm known as ID3 (Iterative Dichotomiser). Later, he presented C4.5, which was the successor of ID3. ID3 and C4.5 adopt a greedy approach. In this algorithm, there is no backtracking; the trees are constructed in a top-down recursive divide-and-conquer manner. Generating a decision tree form training tuples of data partition D.

Algorithm : Generate_decision_tree

Input

Data partition, D, which is a set of training tuples and their associated class labels. attribute_list, the set of candidate attributes. Attribute selection method, a procedure to determine the splitting criterion that best partitions that the data tuples into individual classes. This criterion includes a splitting_attribute and either a splitting point or splitting subset.

Output : A Decision Tree

Method : create a node N;

```
if tuples in D are all of the same class, C then
      return N as leaf node labeled with class C;
if attribute_list is empty then
      return N as leaf node with labeled
      with majority class in D;|| majority voting
apply attribute_selection_method(D, attribute_list) to find the best splitting_criterion;
label node N with splitting_criterion;
if splitting_attribute is discrete-valued and multiway splits allowed then // no restricted to binary trees
attribute_list = splitting attribute; // remove splitting attribute for each outcome j of splitting criterion
      let Dj be the set of data tuples in D satisfying outcome j; // a partition
  if Dj is empty then
    attach a leaf labeled with the majority
    class in D to node N;
  else
    attach the node returned by Generate
    decision tree(Dj, attribute list) to node N;
  end for
return N;
```

Tree Pruning

Tree pruning is performed in order to remove anomalies in the training data due to noise or outliers. The pruned trees are smaller and less complex.

Tree Pruning Approaches

Here is the Tree Pruning Approaches listed below :

- **Pre-pruning**: The tree is pruned by halting its construction early.
- **Post-pruning**: This approach removes a sub-tree from a fully grown tree.

Cost Complexity

The cost complexity is measured by the following two parameters “

- Number of leaves in the tree, and
- Error rate of the tree.

3.3 Data Mining - Bayesian Classification

Q3. What are Bayesian Classification/ Belief Network?

Ans :

Bayesian classification is based on Bayes’ Theorem. Bayesian classifiers are the statistical classifiers. Bayesian classifiers can predict class membership probabilities such as the probability that a given tuple belongs to a particular class.

Baye’s Theorem

Bayes’ Theorem is named after Thomas Bayes. There are two types of probabilities :

- Posterior Probability [P(H/X)]
- Prior Probability [P(H)]

where X is data tuple and H is some hypothesis. According to Bayes’ Theorem,

$$P(H/X) = P(X/H)P(H) / P(X)$$

Bayesian Belief Network

Bayesian Belief Networks specify joint conditional probability distributions. They are also known as Belief Networks, Bayesian Networks, or Probabilistic Networks.

- A Belief Network allows class conditional independencies to be defined between subsets of variables.

- It provides a graphical model of causal relationship on which learning can be performed.
- We can use a trained Bayesian Network for classification.

There are two components that define a Bayesian Belief Network :

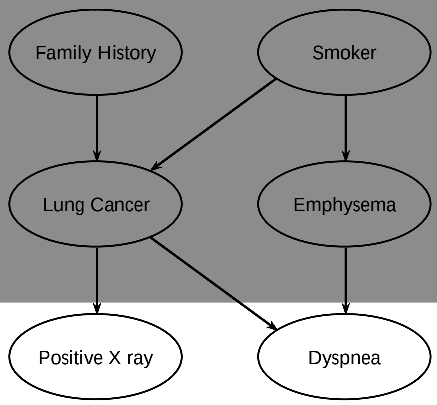
- Directed acyclic graph
- A set of conditional probability tables

Directed Acyclic Graph

- Each node in a directed acyclic graph represents a random variable.
- These variable may be discrete or continuous valued.
- These variables may correspond to the actual attribute given in the data.

Directed Acyclic Graph Representation

The following diagram shows a directed acyclic graph for six Boolean variables.



The arc in the diagram allows representation of causal knowledge. For example, lung cancer is influenced by a person's family history of lung cancer, as well as whether or not the person is a smoker. It is worth noting that the variable Positive X-ray is independent of whether the patient has a family history of lung cancer or that the patient is a smoker, given that we know the patient has lung cancer.

Conditional Probability Table

The arc in the diagram allows representation of causal knowledge. For example, lung cancer is influenced by a person's family history of lung cancer, as well as whether or not the person is a smoker. It is worth noting that the variable Positive X-ray is independent of whether the patient has a family history of lung cancer or that the patient is a smoker, given that we know the patient has lung cancer.

Conditional Probability Table

The arc in the diagram allows representation of causal knowledge. For example, lung cancer is influenced by a person's family history of lung cancer, as well as whether or not the person is a smoker. It is worth noting that the variable Positive X-ray is independent of whether the patient has a family history of lung cancer or that the patient is a smoker, given that we know the patient has lung cancer.

	FH, S	FH,–S	–FH,S	–FH,S
LC	0.8	0.5	0.7	0.1
–LC	0.2	0.5	0.3	0.9

3.4 Data Mining - Rule Based Classification

Q4. Explain Rule Based Classification ?

Ans :

IF-THEN Rules

Rule-based classifier makes use of a set of IF-THEN rules for classification. We can express a rule in the following from :

IF condition THEN conclusion

Let us consider a rule R1,

R1: IF age=youth AND student=yes

THEN buy_computer=yes

Points to Remember

- The IF part of the rule is called **rule antecedent** or **precondition**.
- The THEN part of the rule is called **rule consequent**.
- The antecedent part the condition consist of one or more attribute tests and these tests are logically ANDed.
- The consequent part consists of class prediction.

Note: We can also write rule R1 as follows:

R1: (age = youth) $\wedge$ (student = yes))(buys computer = yes)

If the condition holds true for a given tuple, then the antecedent is satisfied.

Rule Extraction

Here we will learn how to build a rule-based classifier by extracting IF-THEN rules from a decision tree.

Points to Remember

- One rule is created for each path from the root to the leaf node.
- To form a rule antecedent, each splitting criterion is logically AND ed.
- The leaf node holds the class prediction, forming the rule consequent.

Rule Induction Using Sequential Covering Algorithm

Sequential Covering Algorithm can be used to extract IF-THEN rules form the training data. We do not require to generate a decision tree first. In this algorithm, each rule for a given class covers many of the tuples of that class.

Some of the sequential Covering Algorithms are AQ, CN2, and RIPPER. As per the general strategy the rules are learned one at a time. For each time rules are learned, a tuple covered by the rule is removed and the process continues for the rest of the tuples. This is because the path to each leaf in a decision tree corresponds to a rule.

Note :

The Decision tree induction can be considered as learning a set of rules simultaneously.

The Following is the sequential learning Algorithm where rules are learned for one class at a time. When learning a rule from a class Ci, we want the rule to cover all the tuples from class C only and no tuple form any other class.

Algorithm: Sequential Covering

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

Output : A Set of IF-THEN rules.

Method

```
Rule_set={ }; // initial set of rules learned is empty
for each class c do
    repeat
    Rule = Learn_One_Rule(D, Att_valls, c);
    remove tuples covered by Rule form D;
    until termination condition;
  Rule_set=Rule_set+Rule; // add a new rule to rule-set
end for
return Rule_Set;
```

Rule Pruning

The rule is pruned is due to the following reason :

- The Assessment of quality is made on the original set of training data. The rule may perform well on training data but less well on subsequent data. That's why the rule pruning is required.
- The rule is pruned by removing conjunct. The rule R is pruned, if pruned version of R has greater quality than what was assessed on an independent set of tuples.

FOIL is one of the simple and effective method for rule pruning. For a given rule R, FOIL_Prune = pos - neg / pos + neg

where pos and neg is the number of positive tuples covered by R, respectively.

Note

This value will increase with the accuracy of R on the pruning set. Hence, if the FOIL_Prune value is higher for the pruned version of R, then we prune R.

3.4.1 Miscellaneous Classification Methods

Q5. Explain different Classification Methods?

Ans :

We will discuss other classification methods such as Genetic Algorithms, Rough Set Approach, and Fuzzy Set Approach.

Genetic Algorithms

The idea of genetic algorithm is derived from natural evolution. In genetic algorithm, first of all, the initial population is created. This initial population consists of randomly generated rules. We can represent each rule by a string of bits.

For example, in a given training set, the samples are described by two Boolean attributes such as A1 and A2. And this given training set contains two classes such as C1 and C2.

We can encode the rule **IF A1 AND NOT A2 THEN C2** into a bit string **100**. In this bit representation, the two leftmost bits represent the attribute A1 and A2, respectively.

Likewise, the rule **IF NOT A1 AND NOT A2 THEN C1** can be encoded as **001**.

Note “ If the attribute has K values where K>2, then we can use the K bits to encode the attribute values. The classes are also encoded in the same manner.

Points to Remember

- Based on the notion of the survival of the fittest, a new population is formed that consists of the fittest rules in the current population and offspring values of these rules as well.
- The fitness of a rule is assessed by its classification accuracy on a set of training samples.
- The genetic operators such as crossover and mutation are applied to create offspring.
- In crossover, the substring from pair of rules is swapped to form a new pair of rules.
- In mutation, randomly selected bits in a rule’s string are inverted.

Rough Set Approach

We can use the rough set approach to discover structural relationship within imprecise and noisy data.

Note

This approach can only be applied on discrete-valued attributes. Therefore, continuous-valued attributes must be discretized before its use.

The Rough Set Theory is based on the establishment of equivalence classes within the given training data. The tuples that forms the equivalence class are indiscernible. It means the samples are identical with respect to the attributes describing the data.

There are some classes in the given real world data, which cannot be distinguished in terms of available attributes. We can use the rough sets to **roughly** define such classes.

For a given class C, the rough set definition is approximated by two sets as follows :

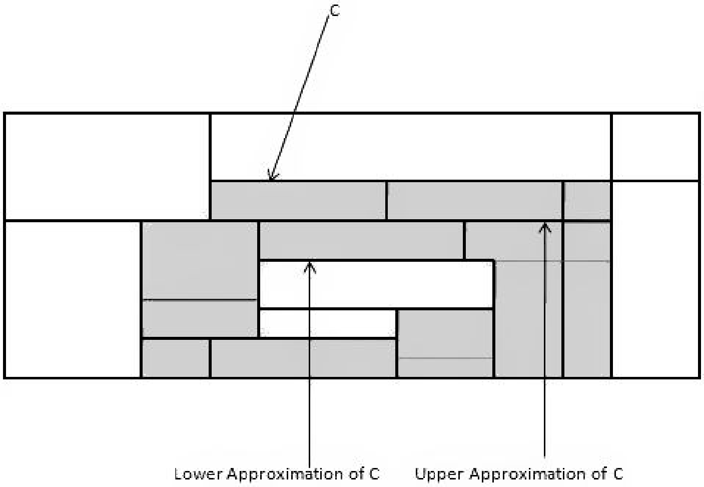
- **Lower Approximation of C**

  The lower approximation of C consists of all the data tuples, that based on the knowledge of the attribute, are certain to belong to class C.

- **Upper Approximation of C**

  The upper approximation of C consists of all the tuples, that based on the knowledge of attributes, cannot be described as not belonging to C.

The following diagram shows the Upper and Lower Approximation of class C:



Fuzzy Set Approaches

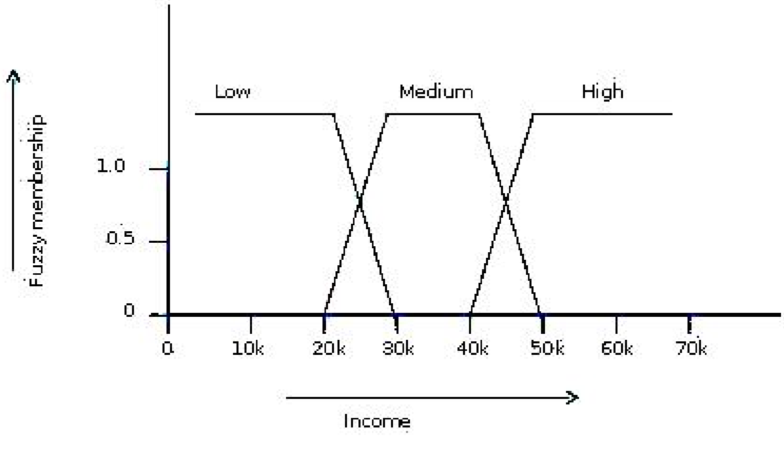
Fuzzy Set Theory is also called Possibility Theory. This theory was proposed by Lotfi Zadeh in 1965 as an alternative the **two-value logic** and **probability theory**. This theory allows us to work at a high level of abstraction. It also provides us the means for dealing with imprecise measurement of data.

The fuzzy set theory also allows us to deal with vague or inexact facts. For example, being a member of a set of high incomes is in exact (e.g. if $50,000 is high then what about $49,000 and $48,000). Unlike the traditional CRISP set where the element either belong to S or its complement but in fuzzy set theory the element can belong to more than one fuzzy set.

For example, the income value $49,000 belongs to both the medium and high fuzzy sets but to differing degrees. Fuzzy set notation for this income value is as follows –

$$\mathbf{m_{medium\_income}(\$49k)=0.15 \text{ and } m_{high\_income}(\$49k)=0.96}$$

where ‘m’ is the membership function that operates on the fuzzy sets of medium_income and high_income respectively. This notation can be shown diagrammatically as follows :



3.4.2 Associative Classification: Classification by Association Rule Analysis

Q6. What are the Methods in Association Classification ?

Ans :

Frequent patterns and their corresponding association or correlation rules characterize interesting relationships between attribute conditions and class labels, and thus have been recently used for effective classification. Association rules show strong associations between attribute-value pairs (or *items*) that occur frequently in a given data set. Association rules are commonly used to analyze the purchasing patterns of customers in a store. Such analysis is useful in many decision-making processes, such as product placement, catalog design, and cross-marketing.

The discovery of association rules is based on *frequent item set mining*. Many methods for frequent item set mining and the generation of association rules were described in Chapter 5. In this section, we look at associative classification, where association rules are generated and analyzed for use in classification. The general idea is that we can search for strong associations between frequent patterns (conjunctions of attribute value pairs) and class labels. Because association rules explore highly confident associations among multiple attributes, this approach may overcome some constraints introduced by decision tree induction, which considers only one attribute at a time. In many studies, associative classification has been found to be more accurate than some traditional classification methods, such as C4.5. In particular, we study three main methods: CBA, CMAR, and CPAR.

Association-Based Classification

Several methods for association-based classification :

- **ARCS: Quantitative association mining and clustering of association rules (Lent et al'97):** It beats C4.5 in (mainly) scalability and also accuracy
- **Associative classification: (Liu et al'98):** It mines high support and high confidence rules in the form of —cond_set $\Rightarrow$ y║ , where y is a class label
- **CAEP (Classification by aggregating emerging patterns) (Dong et al'99):** Emerging patterns (EPs): the itemsets whose support increases significantly from one class to another Mine Eps based on minimum support and growth rate.

3.5 Lazy Learners (or Learning from Your Neighbors)

Q7. Explain Lazy Learners ?

Ans :

The classification methods discussed so far in this chapter - decision tree induction, Bayesian Classification, rule-based classification, classification by back propagation, support vector machines, and classification based on association rule mining—are all examples of *eager learners.* Eager learners, when given a set of training tuples, will construct a generalization (i.e., classification) model before receiving new (e.g., test) tuples to classify. We can think of the learned model as being ready and eager to classify previously unseen tuples.

Other Classification Methods

- k-nearest neighbor classifier
- case-based reasoning
- Genetic algorithm
- Rough set approach
- Fuzzy set approaches

1. k-Nearest-Neighbor Classifiers

The k-nearest-neighbor method was first described in the early 1950s. The method is labor intensive when given large training sets, and did not gain popularity until the 1960s when increased computing power became available. It has since been widely used in the area of pattern recognition.

Nearest-neighbor classifiers are based on learning by analogy, that is, by comparing a given test tuple with training tuples that are similar to it. The training tuples are described by n attributes. Each tuple represents a point in an n-dimensional space. In this way, all of the training tuples are stored in an n-dimensional pattern space. When given an unknown tuple, a k-nearest-neighbor classifier searches the pattern space for the k training tuples that are closest to the unknown tuple. These k training tuples are the k ¯nearest neighbors of the unknown tuple. Closeness is defined in terms of a distance metric, such as Euclidean distance. The Euclidean distance between two points or tuples, say, X1 = (x11, x12,…,x1n) and X2 = (x21, x22,.., x2n), is

$$\mathrm{dist}(X_1, X_2) = \sqrt{\sum_{i=1}^{n}(x_{1i} - x_{2i})^2}$$

2. Case-Based Reasoning

Case-based reasoning (CBR) classifiers use a database of problem solutions to solve new problems. Unlike nearest-neighbor classifiers, which store training tuples as points in Euclidean space, CBR stores the tuples or ¯cases for problem solving as complex symbolic descriptions. Business applications of CBR include problem resolution for customer service help desks, where cases describe product-related diagnostic problems.

CBR has also been applied to areas such as engineering and law, where cases are either technical designs or legal rulings, respectively. Medical education is another area for CBR, where patient case histories and treatments are used to help diagnose and treat new patients. When given a new case to classify, a case-based reasoner will first check if an identical training case exists. If one is found, then the accompanying solution to that case is returned. If no identical case is found, then the case-based reasoner will search for training cases having components that are similar to those of the new case.

Conceptually, these training cases may be considered as neighbors of the new case. If cases are represented as graphs, this involves searching for subgraphs that are similar to subgraphs within the new case. The case based reasoner tries to combine the solutions of the neighboring training cases in order to propose a solution for the new case. If incompatibilities arise with the individual solutions, then backtracking to search for other solutions may be necessary. The case-based reasoner may employ background knowledge and problem-solving strategies in order to propose a feasible combined solution.

Instance-Based Methods

Instance-based learning:

Store training examples and delay the processing ("lazy evaluation") until a new instance must be classified.

Typical Approaches

- ***k*-nearest neighbor approach:** Instances represented as points in a Euclidean space.
- **Locally weighted regression:** Constructs local approximation
- **Case-based reasoning:** Uses symbolic representations and knowledge-based inference
- **Remarks on Lazy vs. Eager Learning**
  - Instance-based learning: lazy evaluation
  - Decision-tree and Bayesian classification: eager evaluation
- **Key differences**
  - Lazy method may consider query instance *xq* when deciding how to generalize beyond the training data *D*
  - Eager method cannot since they have already chosen global approximation when seeing the query
- **Efficiency:** Lazy - less time training but more time predicting
- **Accuracy**
  - Lazy method effectively uses a richer hypothesis space since it uses many local linear functions
  - to form its implicit global approximation to the target function
  - Eager: must commit to a single hypothesis that covers the entire instance space.

3.5.1 Isssues Regarding Classification and Prediction

Q8. What are the issues regarding classification and Predication ?

Ans :

1. **Preparing the Data for Classification and Prediction:** The following preprocessing steps may be applied to the data to help improve the accuracy, efficiency, and scalability of the classification or prediction process.

2. **Data cleaning:** This refers to the preprocessing of data in order to remove or reduce *noise* (by applying smoothing techniques) and the treatment of *missingvalues* (e.g., by replacing a missing value with the most commonly occurring value for that attribute, or with the most probable value based on statistics). Although most classification algorithms have some mechanisms for handling noisy or missing data, this step can help reduce confusion during learning.

3. **Relevance analysis:** Many of the attributes in the data may be *redundant*. Correlation analysis can be used to identify whether any two given attributes are statistically related. For example, a strong correlation between attributes *A*1 and *A*2 would suggest that one of the two could be removed from further analysis. A database may also contain *irrelevant* attributes. Attribute subset selection can be used in these cases to find a reduced set of attributes such that the resulting probability distribution of the data classes is as close as possible to the original distribution obtained using all attributes.

   Hence, relevance analysis, in the form of correlation analysis and attribute subset selection, can be used to detect attributes that do not contribute to the classification or prediction task. Such analysis can help improve classification efficiency and scalability.

4. **Data Transformation And Reduction:** The data may be transformed by normalization, particularly when neural networks or methods involving distance measurements are used in the learning step. Normalization involves scaling all values for a given attribute so that they fall within a small specified range, such as -1 to +1 or 0 to 1. The data can also be transformed by *generalizing* it to higher-level concepts. Concept hierarchies may be used for this purpose. This is particularly useful for continuous valued attributes.

   For example, numeric values for the attribute *income* can be generalized to discrete ranges, such as *low, medium*, and *high*. Similarly, categorical attributes, like *street*, can be generalized to higher-level concepts, like *city.*

   Data can also be reduced by applying many other methods, ranging from wavelet transformation and principle components analysis to discretization techniques, such as binning, histogram analysis, and clustering.

3.6 Classification by Back Propagation

Q9. What are the steps followed in Back Propagation ?

Ans :

Back propagation is a neural network learning algorithm. A neural network is a set of connected input/output units in which each connection has a weight associated with it. During the learning phase, the network learns by adjusting the weights so as to be able to predict the correct class label of the input tuples. Neural network learning is also referred to as connectionist learning due to the connections between units.

Neural networks involve long training times and are therefore more suitable for applications where this is feasible. Back propagation learns by iteratively processing a data set of training tuples, comparing the network's prediction for each tuple with the actual known target value.

The target value may be the known class label of the training tuple (for classification problems) or a continuous value (for prediction).

For each training tuple, the weights are modified so as to minimize the mean squared error between the network's prediction and the actual target value. These modifications are made in the ¯backwards direction, that is, from the output layer, through each hidden layer down to the first hidden layer hence the name is back propagation.

Although it is not guaranteed, in general the weights will eventually converge, and the learning process stops.

Advantages

- It include their high tolerance of noisy data as well as their ability to classify patterns on which they have not been trained.
- They can be used when you may have little knowledge of the relationships between attributes and classes.
- They are well-suited for continuous-valued inputs and outputs, unlike most decision tree algorithms.
- They have been successful on a wide array of real-world data, including handwritten character recognition, pathology and laboratory medicine, and training a computer to pronounce English text.
- Neural network algorithms are inherently parallel; parallelization techniques can be used to speed up the computation process.

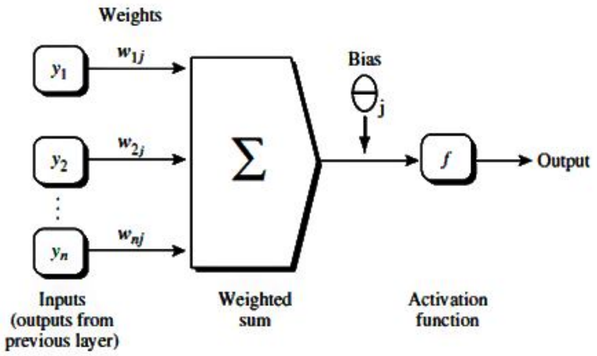
Process: Initialize the weights

The weights in the network are initialized to small random numbers ranging from-1.0 to 1.0, or -0.5 to 0.5. Each unit has a *bias* associated with it. The biases are similarly initialized to small random numbers.

Each training tuple, ***X***, is processed by the following steps. **Propagate the inputs forward:** First, the training tuple is fed to the input layer of the network. The inputs pass through the input units, unchanged. That is, for an input unit *j*, its output, *Oj*, is equal to its input value, *Ij*. Next, the net input and output of each unit in the hidden and output layers are computed. The net input to a unit in the hidden or output layers is computed as a linear combination of its inputs. Each such unit has a number of inputs to it that are, in fact, the outputs of the units connected to it in the previous layer. Each connection has a weight. To compute the net input to the unit, each input connected to the unit is multiplied by its corresponding weight, and this is summed.

$$I_j = \sum_i w_{ij} O_i + \theta_j,$$

Where wi, j is the weight of the connection from unit I in the previous layer to unit j; Oi is the output of unit I from the previous layer Ÿjis the bias of the unit & it acts as a threshold in that it serves to vary the activity of the unit. Each unit in the hidden and output layers takes its net input and then applies an activation function to it.



Back propagate the error

The error is propagated backward by updating the weights and biases to reflect the error of the network's prediction. For a unit *j* in the output layer, the error *Err j* is computed by

$$Err_j = O_j(1 - O_j)(T_j - O_j)$$

Where Ojis the actual output of unit j, and Tjis the known target value of the given training tuple. The error of a hidden layer unit *j* is

$$Err_j = O_j(1 - O_j)\sum_1 Err_k w_{jk}$$

Where wjk is the weight of the connection from unit j to a unit k in the next higher layer, and Errkis the error of unit k. Weights are updated by the following equations, where D*wi j* is the change in weight *wi j*:

$$\Delta w_{ij} = (l) Err_j O_i$$

$$w_{ij} = w_{ij} + \Delta w_{ij}$$

Biases are updated by the following equations below :

$$\Delta\theta_j = (l)\,Err_j$$

$$\theta_j = \theta_j + \Delta\theta_j$$

- D, a data set consisting of the training tuples and their associated target values;
- *l*, the learning rate;
- *network*, a multilayer feed-forward network.

Output : A trained neural network.

Method :

```
Initialize all weights and biases in network
while terminating condition is not satisfied {
    for each training tuple X in D {
        // Propagate the inputs forward:
        for each input layer unit j {
            $O_j = I_j$; // output of an input unit is its actual input value
        for each hidden or output layer unit j {
            $I_j = \Sigma_i w_{ij} O_i + \theta_j$; //compute the net input of unit j with respect to
                                        the previous layer, i
            $O_j = \frac{1}{1+e^{-I}1}$;} ff compute the output of each unit j
        // Backpropagate the errors:
        for each unit j in the output layer
            $Err_j = O_j(1 - O_j)(T_j - O_j)$; // compute the error
        for each unit j in the hidden layers, from the last to the first hidden layer
            $Err_j = O_j(1 - O_j) \Sigma_k Err_k w_{jk}$; // compute the error with respect tothe
                                        next higher layer, k
        for each weight $w_{ij}$ in network {
            $\Delta w_{ij} - (l) Err_j O_i$ ; weight increment
            $w_{ij} = w_{ij} + \Delta w_{ij}$; } // weight update
        for each bias $\theta_j$ in network {
            $\Delta\theta_j = (l) Err_j$ ; // bias increment
            $\theta_j = \theta_j + \Delta\theta_j$ ; } // bias update
        } }
```

3.7 Cluster Analysis

Q10. What is Cluster Analysis ? and What are its requirements ?

Ans :

The process of grouping a set of physical or abstract objects into classes of similar objects is called clustering. A cluster is a collection of data objects that are similar to one another within the same cluster and are dissimilar to the objects in other clusters.

A cluster of data objects can be treated collectively as one group and so may be considered as a form of data compression. Cluster analysis tools based on k-means, k-medoids, and several methods have also been built into many statistical analysis software packages or systems, such as S-Plus, SPSS, and SAS.

Applications

- Cluster analysis has been widely used in numerous applications, including market research, pattern recognition, data analysis, and image processing.
- In business, clustering can help marketers discover distinct groups in their customer bases and characterize customer groups based on purchasing patterns.
- In biology, it can be used to derive plant and animal taxonomies, categorize genes with similar functionality, and gain insight into structures inherent in populations.
- Clustering may also help in the identification of areas of similar land use in an earth observation database and in the identification of groups of houses in a city according to house type, value, and geographic location, as well as the identification of groups of automobile insurance policy holders with a high average claim cost.
- Clustering is also called data segmentation in some applications because clustering partitions large data sets into groups according to their *similarity*.
- Clustering can also be used for outlier detection, Applications of outlier detection include the detection of credit card fraud and the monitoring of criminal activities in electronic commerce.

Typical Requirements Of Clustering In Data Mining

- **Scalability**

  Many clustering algorithms work well on small data sets containing fewer than several hundred data objects; however, a large database may contain millions of objects. Clustering on a sample of a given large data set may lead to biased results. Highly scalable clustering algorithms are needed.

- **Ability to deal with different types of attributes**

  Many algorithms are designed to cluster interval-based (numerical) data. However, applications may require clustering other types of data, such as binary, categorical (nominal), and ordinal data, or mixtures of these data types.

- **Discovery of clusters with arbitrary shape**

  Many clustering algorithms determine clusters based on Euclidean or Manhattan distance measures. Algorithms based on such distance measures tend to find spherical clusters with similar size and density. However, a cluster could be of any shape. It is important to develop algorithms that can detect clusters of arbitrary shape.

- **Minimal requirements for domain knowledge to determine input parameters**

  Many clustering algorithms require users to input certain parameters in cluster analysis (such as the number of desired clusters). The clustering results can be quite sensitive to input parameters. Parameters are often difficult to determine, especially for data sets containing high-dimensional objects. This not only burdens users, but it also makes the quality of clustering difficult to control.

- **Ability to deal with noisy data:**

  Most real-world databases contain outliers or missing, unknown, or erroneous data. Some clustering algorithms are sensitive to such data and may lead to clusters of poor quality.

Incremental clustering and insensitivity to the order of input records

Some clustering algorithms cannot incorporate newly inserted data (i.e., database updates) into existing clustering structures and, instead, must determine a new clustering from scratch. Some clustering algorithms are sensitive to the order of input data. That is, given a set of data objects, such an algorithm may return dramatically different clusterings depending on the order of presentation of the input objects. It is important to develop incremental clustering algorithms and algorithms that are insensitive to the order of input.

- **High dimensionality**

  A database or a data warehouse can contain several dimensions or attributes. Many clustering algorithms are good at handling low-dimensional data, involving only two to three dimensions. Human eyes are good at judging the quality of clustering for up to three dimensions. Finding clusters of data objects in high dimensional space is challenging, especially considering that such data can be sparse and highly skewed.

- **Constraint-based clustering**

  Real-world applications may need to perform clustering under various kinds of constraints. Suppose that your job is to choose the locations for a given number of new automatic banking machines (ATMs) in a city. To decide upon this, you may cluster households while considering constraints such as the city's rivers and highway networks, and the type and number of customers per cluster. A challenging task is to find groups of data with good clustering behavior that satisfy specified constraints.

- **Interpretability and usability**

  Users expect clustering results to be interpretable, comprehensible, and usable. That is, clustering may need to be tied to specific semantic interpretations and applications. It is important to study how an application goal may influence the selection of clustering features and methods.

Major Clustering Methods

- Partitioning Methods
- Hierarchical Methods
- Density-Based Methods
- Grid-Based Methods
- Model-Based Methods

3.8 Partitioning Methods

Q11. What are different Clustering Methods?

Ans :

A partitioning method constructs *k* partitions of the data, where each partition represents a cluster and *k* <= *n*. That is, it classifies the data into *k* groups, which together satisfy the following requirements:

- Each group must contain at least one object, and
- Each object must belong to exactly one group.

A partitioning method creates an initial partitioning. It then uses an iterative relocation technique that attempts to improve the partitioning by moving objects from one group to another. The general criterion of a good partitioning is that objects in the same cluster are close or related to each other, whereas objects of different clusters are far apart or very different.

Hierarchical Methods

A hierarchical method creates a hierarchical decomposition of the given set of data objects. A hierarchical method can be classified as being either agglomerative or divisive, based on how the hierarchical decomposition is formed.

- The agglomerative approach, also called the bottom-up approach, starts with each object forming a separate group. It successively merges the objects or groups that are close to one another, until all of the groups are merged into one or until a termination condition holds.

- The divisive approach, also called the top-down approach, starts with all of the objects in the same cluster. In each successive iteration, a cluster is split up into smaller clusters, until eventually each object is in one cluster, or until a termination condition holds.
- Hierarchical methods suffer from the fact that once a step (merge or split) is done, it can never be undone. This rigidity is useful in that it leads to smaller computation costs by not having to worry about a combinatorial number of different choices. There are two approaches to improving the quality of hierarchical clustering :
  - Perform careful analysis of object –linkages‖ at each hierarchical partitioning, such as in Chameleon, or
  - Integrate hierarchical agglomeration and other approaches by first using a hierarchical agglomerative algorithm to group objects into microclusters, and then performing macroclustering on the microclusters using another clustering method such as iterative relocation.

Density-based methods

1. Most partitioning methods cluster objects based on the distance between objects. Such methods can find only spherical-shaped clusters and encounter difficulty at discovering clusters of arbitrary shapes.
2. Other clustering methods have been developed based on the notion of density. Their general idea is to continue growing the given cluster as long as the density in the neighborhood exceeds some threshold; that is, for each data point within a given cluster, the neighborhood of a given radius has to contain at least a minimum number of points. Such a method can be used to filter out noise (outliers)and discover clusters of arbitrary shape.
3. DBSCAN and its extension, OPTICS, are typical density-based methods that grow clusters according to a density-based connectivity analysis. DENCLUE is a method that clusters objects based on the analysis of the value distributions of density functions.

Grid-Based Methods

- Grid-based methods quantize the object space into a finite number of cells that form a grid structure.
- All of the clustering operations are performed on the grid structure i.e., on the quantized space. The main advantage of this approach is its fast processing time, which is typically independent of the number of data objects and dependent only on the number of cells in each dimension in the quantized space.
- STING is a typical example of a grid-based method. Wave Cluster applies wavelet transformation for clustering analysis and is both grid-based and density-based.

Model-Based Methods:

- Model-based methods hypothesize a model for each of the clusters and find the best fit of the data to the given model.
- A model-based algorithm may locate clusters by constructing a density function that reflects the spatial distribution of the data points.
- It also leads to a way of automatically determining the number of clusters based on standard statistics, taking –noise or outliers into account and thus yielding robust clustering methods.

3.9 Data Mining Operations

Q12. What are the Different Operations in Data Mining ?

Ans :

Tasks in Data Mining

- Clustering High-Dimensional Data
- Constraint-Based Clustering

1. Clustering High-Dimensional Data

It is a particularly important task in cluster analysis because many applications require the analysis of objects containing a large number of features or dimensions.

For example, text documents may contain thousands of terms or keywords as features, and DNA micro array data may provide information on the expression levels of thousands of genes under hundreds of conditions.

Clustering high-dimensional data is challenging due to the curse of dimensionality. Many dimensions may not be relevant. As the number of dimensions increases, the data become increasingly sparse so that the distance measurement between pairs offprints become meaningless and the average density of points anywhere in the data is likely to be low. Therefore, a different clustering methodology needs to be developed for high-dimensional data.

CLIQUE and PROCLUS are two influential subspace clustering methods, which search for clusters in subspaces of the data, rather than over the entire data space.

Frequent pattern–based clustering, another clustering methodology, and extracts distinct frequent patterns among subsets of dimensions that occur frequently. It uses such patterns to group objects and generate meaningful clusters.

2. Constraint-Based Clustering

It is a clustering approach that performs clustering by incorporation of user-specified or application-oriented constraints.

A constraint expresses a user's expectation or describes properties of the desired clustering results, and provides an effective means for communicating with the clustering process.

Various kinds of constraints can be specified, either by a user or as per application requirements.

Spatial clustering employs with the existence of obstacles and clustering under user-specified constraints. In addition, semi-supervised clustering employs for pair wise constraints in order to improve the quality of the resulting clustering.

Classical Partitioning Methods

The most well-known and commonly used partitioning methods are

- The *k*-Means Method
- k-Medoids Method

1. Centroid-Based Technique: The *K*-Means Method

The k-means algorithm takes the input parameter, k, and partitions a set of n objects into k clusters so that the resulting intra cluster similarity is high but the inter cluster similarity is low. Cluster similarity is measured in regard to the mean value of the objects in a cluster, which can be viewed as the cluster's centroid or center of gravity. The *k*-means algorithm proceeds as follows.

First, it randomly selects *k* of the objects, each of which initially represents a cluster mean or center.

For each of the remaining objects, an object is assigned to the cluster to which it is the most similar, based on the distance between the object and the cluster mean.

It then computes the new mean for each cluster.

This process iterates until the criterion function converges.

Typically, the square-error criterion is used, defined as

$$E = \sum_{i=1}^{k} \sum_{p \in C_i} |p - m_i|^2 ,$$

Where E is the sum of the square error for all objects in the data set p is the point in space representing a given object m is the mean of cluster Ci

The k-means partitioning algorithm: The *k*-means algorithm for partitioning, where each cluster's center is represented by the mean value of the objects in the cluster.

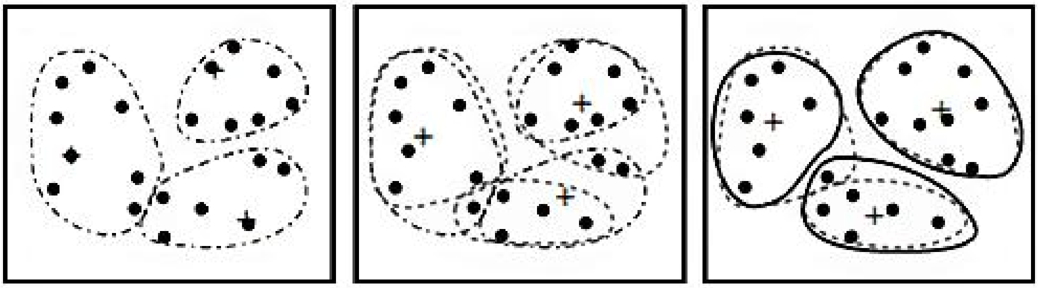
Input :

- k : the number of clusters,
- D : a data set containing n objects.

Output : A set of k clusters.

Method:

```
1) Arbitrarily choose k objects from D as the initial cluster centers;
2) Repeat
3) (Re)assign each object to the cluster to which the object is the most similar, based on the mean value of the objects in the cluster;
4) Update the cluster means, i.e., calculate the mean value of the objects for each cluster;
5) Until no change;
```



3.10 Constraint-Based Cluster Analysis

Q13. Explain in detail Constraint-Based Cluster Analysis ?

Ans :

Constraint-based clustering finds clusters that satisfy user-specified preferences or constraints. Depending on the nature of the constraints, constraint-based clustering may adopt rather different approaches. There are a few categories of constraints.

Constraints on individual objects

We can specify constraints on the objects to be clustered. In a real estate application, for example, one may like to spatially cluster only those luxury mansions worth over a million dollars. This constraint confines the set of objects to be clustered. It can easily be handled by preprocessing after which the problem reduces to an instance of unconstrained clustering.

Constraints on the selection of clustering parameters

A user may like to set a desired range for each clustering parameter. Clustering parameters are usually quite specific to the given clustering algorithm. Examples of parameters include k, the desired number of clusters in a k-means algorithm; or e the radius and the minimum number of points in the DBSCAN algorithm. Although such user-specified parameters may strongly influence the clustering results, they are usually confined to the algorithm itself. Thus, their fine tuning and processing are usually not considered a form of constraint-based clustering.

Constraints on distance or similarity functions

We can specify different distance or similarity functions for specific attributes of the objects to be clustered, or different distance measures for specific pairs of objects. When clustering sports men, for example, we may use different weighting schemes for height, body weight, age, and skill level. Although this will likely change the mining results, it may not alter the clustering process per se. However, in some cases, such changes may make the evaluation of the distance function nontrivial, especially when it is tightly intertwined with the clustering process.

User-specified constraints on the properties of individual clusters:

A user may like to specify desired characteristics of the resulting clusters, which may strongly influence the clustering process.

Semi-supervised clustering based on partial supervision:

The quality of unsupervised clustering can be significantly improved using some weak form of supervision. This may be in the form of pair wise constraints (i.e., pairs of objects labeled as belonging to the same or different cluster). Such a constrained clustering process is called semi-supervised clustering.

3.10.1 Outlier Analysis

Q14. Explain in detail about Outlier Analysis in Data Mining and Data Warehouse ?

Ans :

There exist data objects that do not comply with the general behavior or model of the data. Such data objects, which are grossly different from or inconsistent with the remaining set of data, are called outliers.

Many data mining algorithms try to minimize the influence of outliers or eliminate them all together. This, however, could result in the loss of important hidden information because one person's noise could be another person's signal. In other words, the outliers may be of particular interest, such as in the case of fraud detection, where outliers may indicate fraudulent activity. Thus, outlier detection and analysis is an interesting data mining task, referred to as outlier mining.

It can be used in fraud detection, for example, by detecting unusual usage of credit cards or telecommunication services. In addition, it is useful in customized marketing for identifying the spending behavior of customers with extremely low or extremely high incomes, or in medical analysis for finding unusual responses to various medical treatments.

Outlier mining can be described as follows: Given a set of *n* data points or objects and *k*, the expected number of outliers, find the top *k* objects that are considerably is similar, exceptional, or inconsistent with respect to the remaining data. The outlier mining problem can be viewed as two sub problems:

- Define what data can be considered as inconsistent in a given data set, and
- Find an efficient method to mine the outliers so defined.

Types of Outlier Detection

- Statistical Distribution-Based Outlier Detection
- Distance-Based Outlier Detection
- Density-Based Local Outlier Detection
- Deviation-Based Outlier Detection

Statistical Distribution-Based Outlier Detection

The statistical distribution-based approach to outlier detection assumes a distribution or probability model for the given data set (e.g., a normal or Poisson distribution) and then identifies outliers with respect to the model using a discordancy test. Application of the test requires knowledge of the data set parameters knowledge of distribution parameters such as the mean and variance and the expected number of outliers. A statistical discordancy test examines two hypotheses:

- A working hypothesis
- An alternative hypothesis

A working hypothesis, H, is a statement that the entire data set of n objects comes from an initial distribution model, F, that is, The hypothesis is retained if there is no statistically significant evidence supporting its rejection. A discordancy test verifies whether an object, O_i, is significantly large (or small) in relation to the distribution F. Different test statistics have been proposed for use as a discordancy test, depending on the available knowledge of the data. Assuming that some statistic, T, has been chosen for discordancy testing, and the value of the statistic for object oi is vi, then the distribution of T is constructed. Significance probability, SP(vi) = Prob(T>vi), is evaluated. If SP(vi) is sufficiently small, then oi is discordant and the working hypothesis is rejected.

An alternative hypothesis, H, which states that O_i comes from another distribution model, G, is adopted. The result is very much dependent on which model F is chosen because oi may be an outlier under one model and a perfectly valid value under another. The alternative distribution is very important in determining the power of the test, that is, the probability that the working hypothesis is rejected when O_i is really an outlier. There are different kinds of alternative distributions.

Inherent Alternative Distribution

In this case, the working hypothesis that all of the objects come from distribution F is rejected in favor of the alternative hypothesis that all of the objects arise from another distribution, G: H : O_i € G, where i = 1, 2,…, n F and G may be different distributions or differ only in parameters of the same distribution. There are constraints on the form of the *G* distribution in that it must have potential to produce outliers. For example, it may have a different mean or dispersion, or a longer tail.

Mixture Alternative Distribution

The mixture alternative states that discordant values are not outliers in the F population, but contaminants from some other population, G. In this case, the alternative hypothesis is **Slippage alternative distribution:**

This alternative states that all of the objects (apart from some prescribed small number) arise independently from the initial model, F, with its given parameters, whereas the remaining objects are independent observations from a modified version of F in which the parameters have been shifted. There are two basic types of procedures for detecting outliers : **Block procedures:** In this case, either all of the suspect objects are treated as outliers or all of them are accepted as consistent.

Consecutive procedures

An example of such a procedure is the *inside out* procedure. Its main idea is that the object that is least likely to be an outlier is tested first. If it is found to be an outlier, then all of the more extreme values are also considered outliers; otherwise, the next most extreme object is tested, and so on. This procedure tends to be more effective than block procedures.

Distance-Based Outlier Detection

The notion of distance-based outliers was introduced to counter the main limitations imposed by statistical methods. An object, o, in a data set, D, is a distance-based (DB) outlier with parameters pct and dmin, that is, a DB (pct;dmin)-outlier, if at least a fraction, pct, of the objects in D lie at a distance greater than dmin from o.

In other words, rather that relying on statistical tests, we can think of distance-based outliers as those objects that do not have enough neighbors, where neighbors are defined based on distance from the given object. In comparison with statistical-based methods, distance based outlier detection generalizes the ideas behind discordancy testing for various standard distributions. Distance-based outlier detection avoids the excessive computation that can be associated with fitting the observed distribution into some standard distribution and in selecting discordancy tests.

For many discordancy tests, it can be shown that if an object, o, is an outlier according to the given test, then o is also a DB(pct, dmin)-outlier for some suitably defined pct and dmin. For example, if objects that lie three or more standard deviations from the mean are considered to be outliers, assuming a normal distribution, then this definition can be generalized by a DB(0.9988, 0.13s) outlier. Several efficient algorithms for mining distance-based outliers have been developed.

Index-based algorithm

Given a data set, the index-based algorithm uses multidimensional indexing structures, such as R-trees or k-d trees, to search for neighbors of each object ***o*** within radius *dmin* around that object. Let *M* be the maximum number of objects within the *d min*-neighborhood of an outlier. Therefore, once *M*+1 neighbors of object ***o*** are found, it is clear that ***o*** is not an outlier. This algorithm has a worst-case complexity of *O*(*n*2*k*), where *n* is the number of objects in the data set and *k* is the dimensionality. The index-based algorithm scales well as *k* increases. However, this complexity evaluation takes only the search time into account, even though the task of building an index in itself can be computationally intensive.

Nested-loop algorithm

The nested-loop algorithm has the same computational complexity as the index-based algorithm but avoids index structure construction and tries to minimize the number of I/Os. It divides the memory buffer space into two halves and the data set into several logical blocks. By carefully choosing the order in which blocks are loaded into each half, I/O efficiency can be achieved.

Cell-based algorithm

To avoid O(n2) computational complexity, a cell-based algorithm was developed for memory-resident data sets. Its complexity is O(ck+n), where c is a constant depending on the number of cells and k is the dimensionality.

In this method, the data space is partitioned into cells with a side length equal to Each cell has two layers surrounding it. The first layer is one cell thick, while the second is cells thick, rounded up to the closest integer. The algorithm counts outliers on a cell-by-cell rather than an object-by-object basis. For a given cell, it accumulates three counts—the number of objects in the cell, in the cell and the first layer together, and in the cell and both layers together. Let's refer to these counts as cell count, cell + 1 layer count, and cell + 2 layers count, respectively. Let M be the maximum number of outliers that can exist in the dmin-neighborhood of an outlier.

An object, **o**, in the current cell is considered an outlier only if cell + 1 layer count is less than or equal to M. If this condition does not hold, then all of the objects in the cell can be removed from further investigation as they cannot be outliers.

If cell_+ 2_layers_count is less than or equal to M, then all of the objects in the cell are considered outliers. Otherwise, if this number is more than M, then it is possible that some of the objects in the cell may be outliers. To detect these outliers, object-by-object processing is used where, for each object, **o**, in the cell, objects in the second layer of **o** are examined. For objects in the cell, only those objects having no more than M points in their dmin-neighborhoods are outliers. The dmin-neighborhood of an object consists of the object's cell, all of its first layer, and some of its second layer.

A variation to the algorithm is linear with respect to n and guarantees that no more than three passes over the data set are required. It can be used for large disk-resident data sets, yet does not scale well for high dimensions.

Density-Based Local Outlier Detection

Statistical and distance-based outlier detection both depend on the overall or global distribution of the given set of data points, D. However, data are usually not uniformly distributed. These methods encounter difficulties when analyzing data with rather different density distributions. To define the local outlier factor of an object, we need to introduce the concepts of k-distance, k-distance neighborhood, reachability distance,13 and local reachability density. These are defined as follows:

The k-distance of an object p is the maximal distance that p gets from its k-nearest neighbors. This distance is denoted as k-distance(p). It is defined as the distance, d(p, o), between p and an object o 2 D, such that for at least k objects, 2 D, it holds that d(p, o')_d(p, o). That is, there are at least k objects in D that are as close as or closer to p than o, and for at most k-1 objects, 2D, it holds that d(p;o'') <d(p, o). That is, there are at most k-1 objects that are closer to p than o. You may be wondering at this point how k is determined.

The LOF method links to density-based clustering in that it sets k to the parameter rMinPts, which specifies the minimum number of points for use in identifying clusters based on density. Here, MinPts (as k) is used to define the local neighborhood of an object, p. The k-distance neighborhood of an object p is denoted Nk distance(p)(p), or Nk(p)for short. By setting k to MinPts, we get NMinPts(p). It contains the MinPts-nearest neighbors of p. That is, it contains every object whose distance is not greater than theMinPts-distance of p. The reachability distance of an object p with respect to object o (where o is within theMinPts-nearest neighbors of p), is defined as reach distMinPts(p, o) = max{MinPtsdistance(o), d(p, o)}.

Intuitively, if an object p is far away, then the reach ability distance between the two is simply their actual distance. However, if they are sufficiently close (i.e., where p is within the MinPts-distance neighborhood of o), then the actual distance is replaced by the MinPts-distance of o. This helps to significantly reduce the statistical fluctuations of d(p, o) for all of the p close to O. The higher the value of MinPts is, the more similar is the reachability distance for objects withinthe same neighborhood. Intuitively, the local reachability density of p is the inverse of the average reachability density based on the MinPts-nearest neighbors of p. It is defined as

$$lrd_{MinPts}(p) = \frac{|N_{min\,Pts}(p)|}{\sum_{0 \in N_{MinPts}(p)} reach\_dist_{MinPts}(p, o)}$$

The local outlier factor (LOF) of ***p*** captures the degree to which we call ***p*** an outlier. It is defined as

$$LOF_{MinPts}(P) = \frac{\sum_{0 \in N_{MinPts}(p)} \frac{lrd_{MinPts}(0)}{lrd_{MinPts}(p)}}{|N_{MinPts}(p)|}$$

is the average of the ratio of the local reachability density of ***p*** and those of ***p***'s *MinPts*-nearest neighbors. It is easy to see that the lower ***p***'s local reachability density is, and the higher the local reachability density of ***p***'s *MinPts*-nearest neighbors are, the higher *LOF*(***p***) is.

Deviation-Based Outlier Detection

Deviation-based outlier detection does not use statistical tests or distance-based measures to identify exceptional objects. Instead, it identifies outliers by examining the main characteristics of objects in a group. Objects that ¯deviate from this description are considered outliers. Hence, in this approach the term deviations is typically used to refer to outliers. In this section, we study two techniques for deviation-based outlier detection. The first sequentially compares objects in a set, while the second employs an OLAP data cube approach.

Sequential Exception Technique

The sequential exception technique simulates the way in which humans can distinguish unusual objects from among a series of supposedly like objects. It uses implicit redundancy of the data. Given a data set, D, of n objects, it builds a sequence of subsets,{D1, D2, …,Dm}, of these objects with 2<=m <= n such that

$$D_{j-1} \subset D_j, \text{ where } D_j \subseteq D.$$

Dissimilarities are assessed between subsets in the sequence. The technique introduces the following key terms.

Exception set

This is the set of deviations or outliers. It is defined as the smallest subset of objects whose removal results in the greatest reduction of dissimilarity in the residual set.

Dissimilarity function

This function does not require a metric distance between the objects. It is any function that, if given a set of objects, returns a low value if the objects are similar to one another. The greater the dissimilarity among the objects, the higher the value returned by the function. The dissimilarity of a subset is incrementally computed based on the subset prior to it in the sequence. Given a subset of *n* numbers, {*x*1, …,*xn*}, a possible dissimilarity function is the variance of the numbers in the set, that is,

$$\frac{1}{n}\sum_{i=1}^{n}(x_i - \overline{x})^2,$$

where x is the mean of the n numbers in the set. For character strings, the dissimilarity function may be in the form of a pattern string (e.g., containing wild card characters that is used to cover all of the patterns seen so far. The dissimilarity increases when the pattern covering all of the strings in Dj-1 does not cover any string in Dj that is not in Dj.

Cardinality function

This is typically the count of the number of objects in a given set. **Smoothing factor:** This function is computed for each subset in the sequence. It assesses how much the dissimilarity can be reduced by removing the subset from the original set of objects.

Short Answers

1. Classification and Prediction

There are two forms of data analysis that can be used for extracting models describing important classes or to predict future data trends. These two forms are as follows

- Classification
- Prediction

Classification

Classification models predict categorical class labels; and prediction models predict continuous valued functions. For example, we can build a classification model to categorize bank loan applications as either safe or risky, or a prediction model to predict the expenditures in dollars of potential customers on computer equipment given their income and occupation.

Following are the examples of cases where the data analysis task is Classification —

- A bank loan officer wants to analyze the data in order to know which customer (loan applicant) are risky or which are safe.
- A marketing manager at a company needs to analyze a customer with a given profile, who will buy a new computer.

In both of the above examples, a model or classifier is constructed to predict the categorical labels. These labels are risky or safe for loan application data and yes or no for marketing data.

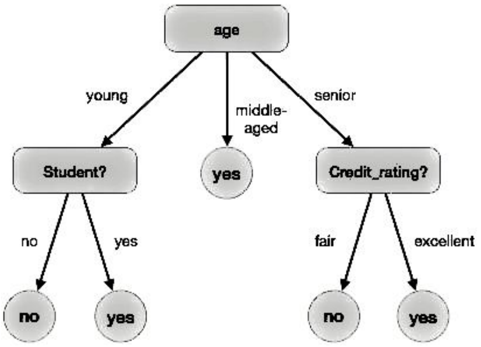
Prediction

Following are the examples of cases where the data analysis task is Prediction —Suppose the marketing manager needs to predict how much a given customer will spend during a sale at his company. In this example we are bothered to predict a numeric value. Therefore the data analysis task is an example of numeric prediction. In this case, a model or a predictor will be constructed that predicts a continuous-valued-function or ordered value.

2. Decision Tree Induction

A decision tree is a structure that includes a root node, branches, and leaf nodes. Each internal node denotes a test on an attribute, each branch denotes the outcome of a test, and each leaf node holds a class label. The topmost node in the tree is the root node.

The following decision tree is for the concept buy_computer that indicates whether a customer at a company is likely to buy a computer or not. Each internal node represents a test on an attribute. Each leaf node represents a class.



3. Algorithm- Generate _ decision _ tree

Input

Data partition, D, which is a set of training tuples and their associated class labels. attribute_list, the set of candidate attributes. Attribute selection method, a procedure to determine the splitting criterion that best partitions that the data tuples into individual classes. This criterion includes a splitting_attribute and either a splitting point or splitting subset.

Output : A Decision Tree

Method : create a node N;

```
if tuples in D are all of the same class, C then
        return N as leaf node labeled with class C;
if attribute_list is empty then
        return N as leaf node with labeled
        with majority class in D;|| majority voting

        apply attribute_selection_method(D, attribute_list) to find the best splitting_criterion;
label node N with splitting_criterion;
if splitting_attribute is discrete-valued and multiway splits allowed then  // no restricted to binary trees
attribute_list = splitting attribute; // remove splitting attribute for each outcome j of splitting criterion
        let Dj be the set of data tuples in D satisfying outcome j; // a partition
if Dj is empty then
        attach a leaf labeled with the majority
        class in D to node N;
else
        attach the node returned by Generate
        decision tree(Dj, attribute list) to node N;
end for
return N;
```

4. Rule Based Classification

IF-THEN Rules

Rule-based classifier makes use of a set of IF-THEN rules for classification. We can express a rule in the following from :

IF condition THEN conclusion

Let us consider a rule R1,

R1: IF age=youth AND student=yes

THEN buy_computer=yes

Points to Remember

- The IF part of the rule is called **rule antecedent** or **precondition**.
- The THEN part of the rule is called **rule consequent**.
- The antecedent part the condition consist of one or more attribute tests and these tests are logically AND ed.
- The consequent part consists of class prediction.

5. Associative Classification

Frequent patterns and their corresponding association or correlation rules characterize interesting relationships between attribute conditions and class labels, and thus have been recently used for effective classification. Association rules show strong associations between attribute-value pairs (or items) that occur frequently in a given data set. Association rules are commonly used to analyze the purchasing patterns of customers in a store. Such analysis is useful in many decision-making processes, such as product placement, catalog design, and cross-marketing.

The discovery of association rules is based on frequent item set mining. Many methods for frequent item set mining and the generation of association rules were described in Chapter 5. In this section, we look at associative classification, where association rules are generated and analyzed for use in classification. The general idea is that we can search for strong associations between frequent patterns (conjunctions of attribute value pairs) and class labels. Because association rules explore highly confident associations among multiple attributes, this approach may overcome some constraints introduced by decision tree induction, which considers only one attribute at a time. In many studies, associative classification has been found to be more accurate than some traditional classification methods, such as C4.5. In particular, we study three main methods: CBA, CMAR, and CPAR.

6. k-Nearest Classifiers

The k-nearest-neighbor method was first described in the early 1950s. The method is labor intensive when given large training sets, and did not gain popularity until the 1960s when increased computing power became available. It has since been widely used in the area of pattern recognition.

Nearest-neighbor classifiers are based on learning by analogy, that is, by comparing a given test tuple with training tuples that are similar to it. The training tuples are described by n attributes. Each tuple represents a point in an n-dimensional space. In this way, all of the training tuples are stored in an n-dimensional pattern space. When given an unknown tuple, a **k**-nearest-neighbor classifier searches the pattern space for the k training tuples that are closest to the unknown tuple. These k training tuples are the k ¯nearest neighbors of the unknown tuple.

– Closeness is defined in terms of a distance metric, such as Euclidean distance. The Euclidean distance between two points or tuples, say, X1 = (x11, x12, : : : , x1n) and X2 = (x21, x22, : : :, x2n), is

$$\text{dist}(X_1, X_2) = \sqrt{\sum_{i=1}^{n}(x_{1i} - x_{2i})^2}$$

7. Centroid-Based Classification

The *K*-Means Method: The k-means algorithm takes the input parameter, k, and partitions a set of n objects into k clusters so that the resulting intra cluster similarity is high but the inter cluster similarity is low. Cluster similarity is measured in regard to the mean value of the objects in a cluster, which can be viewed as the cluster's centroid or center of gravity. The *k*-means algorithm proceeds as follows.

First, it randomly selects *k* of the objects, each of which initially represents a cluster mean or center.

For each of the remaining objects, an object is assigned to the cluster to which it is the most similar, based on the distance between the object and the cluster mean.

- It then computes the new mean for each cluster.
- This process iterates until the criterion function converges.
- Typically, the square-error criterion is used, defined as

$$E = \sum_{i=1}^{k} \sum_{p \in C_i} |p - m_i|^2,$$

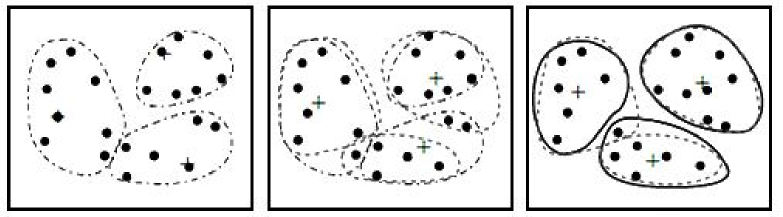
The k-means partitioning algorithm: The *k*-means algorithm for partitioning, where each cluster's center is represented by the mean value of the objects in the cluster.

- k: the number of clusters,
- D: a data set containing n objects.

Output : A set of k clusters.

Method :

```
1)  arbitrarily choose k objects from D as the initial cluster centers;
2)  repeat
3)  (re)assign each object to the cluster to which the object is the most similar,
        based on the mean value of the objects in the cluster;
4)  update the cluster means, i.e., calculate the mean value of the objects for
    each cluster;
5)  until no change;
```



8. Back propagation Classification

Back propagation is a neural network learning algorithm. A neural network is a set of connected input/output units in which each connection has a weight associated with it. During the learning phase, the network learns by adjusting the weights so as to be able to predict the correct class label of the input tuples. Neural network learning is also referred to as connectionist learning due to the connections between units.

Neural networks involve long training times and are therefore more suitable for applications where this is feasible. Back propagation learns by iteratively processing a data set of training tuples, comparing the network's prediction for each tuple with the actual known target value.

The target value may be the known class label of the training tuple (for classification problems) or a continuous value (for prediction).

For each training tuple, the weights are modified so as to minimize the mean squared error between the network's prediction and the actual target value. These modifications are made in the — backwards║ direction, that is, from the output layer, through each hidden layer down to the first hidden layer hence the name is back propagation.

Although it is not guaranteed, in general the weights will eventually converge, and the learning process stops.

9. Partitioning by Hierarchical Methods

A hierarchical method creates a hierarchical decomposition of the given set of data objects. A hierarchical method can be classified as being either agglomerative or divisive, based on how the hierarchical decomposition is formed.

- The agglomerative approach, also called the bottom-up approach, starts with each object forming a separate group. It successively merges the objects or groups that are close to one another, until all of the groups are merged into one or until a termination condition holds.
- The divisive approach, also called the top-down approach, starts with all of the objects in the same cluster. In each successive iteration, a cluster is split up into smaller clusters, until eventually each object is in one cluster, or until a termination condition holds.
- Hierarchical methods suffer from the fact that once a step (merge or split) is done, it can never be undone. This rigidity is useful in that it leads to smaller computation costs by not having to worry about a combinatorial number of different choices. There are two approaches to improving the quality of hierarchical clustering:
- Perform careful analysis of object ¯linkages at each hierarchical partitioning, such as in Chameleon, or Integrate hierarchical agglomeration and other approaches by first using a hierarchical agglomerative algorithm to group objects into micro clusters, and then performing macro clustering on the micro clusters using another clustering method such as iterative relocation.

UNIT IV

Outlier Detection: outliers and outlier analysis, outlier detection methods, statistical approaches, proximitybased approaches, clustering-based approaches,classification-based approaches.Data Mining Trends and

Research Frontiers: mining complex data types, other methodologies of data mining,data mining applications, data mining and society, data mining trends.

4.1 Outliers & Outlier Analysis

Q1. What are Outliers? Explain Outlier analysis.

Ans :

- A database may contain data objects that do not comply with the general behavior or model of the data. Such data objects, which are grossly different from or inconsistent with the remaining set of data, are called outliers.
- Most data mining methods discard outliers as noise or exceptions.
- However, in some applications such as fraud detection, the rare events can be more interesting than the more regularly occurring ones.
- The analysis of outlier data is referred to as outlier mining.
- Outliers may be detected using statistical tests that assume a distribution or probability model for the data, or using distance measures where objects that are a substantial distance from any other cluster are considered outliers.
- Rather than using statistical or distance measures, deviation-based methods identify outliers by examining differences in the main characteristics of objects in a group.
- Outliers can be caused by measurement or execution error.
- Outliers may be the result of inherent data variability.
- Many data mining algorithms try to minimize the influence of outliers or eliminate them all together.
- This, however, could result in the loss of important hidden information because one person's noise could be another person's signal.
- Thus, outlier detection and analysis is an interesting data mining task, referred to as outlier mining.

There are four approaches to computer-based methods for outlier detection.

- The statistical approach:
  - This approach assumes a distribution for the given data set and then identifies outliers with respect to the model using a discordancy test.
  - A statistical discordancy test examines two hypotheses: a working hypothesis and an alternative hypothesis.
  - A working hypothesis, H, is a statement that the entire data set of n objects comes from an initial distribution model, F, that is H:o_i ∊ F, where i = 1, 2, ..., n.
  - The hypothesis is retained if there is no statistically significant evidence supporting its rejection.
  - A discordancy test verifies whether an object, oi , is significantly large (or small) in relation to the distribution F.
  - Assuming that some statistic, T, has been chosen for discordancy testing, and the value of the statistic for object oiisvi , then the distribution of T is constructed.
  - Significance probability, SP(vi) = Prob (T > vi), is evaluated.

- If SP(vi) is sufficiently small, then oi is discordant and the working hypothesis is rejected.
- An alternative hypothesis, H, which states that oi comes from another distribution model (G), is adopted.
- The result is very much dependent on which model is chosen because oi may be an outlier under one model and a perfectly valid value under another.
- The distance-based approach:

➢ This approach generalizes the ideas behind discordancy testing for various standard distributions and its neighbors are defined based on their distance from the given object.

- Several efficient algorithms for mining distance-based outliers have been developed.

1. Index-based algorithm:

➢ Given a data set, the index-based algorithm uses multidimensional indexing structures, such as R-trees or k-d trees, to search for neighbors of each object 'o'within radius 'dmin' around that object.

➢ Let M be the maximum number of objects within the dmin-neighborhood of an outlier.

➢ Therefore, once $M+1$ neighbors of object o are found, it is clear that o is not an outlier.

➢ This algorithm has a worst-case complexity of O(n2k), where n is the number of objects in the data set and k is the dimensionality.

➢ The index-based algorithm scales well as k increases.

2. Nested-loop algorithm:

➢ The nested-loop algorithm has the same computational complexity as the index-based algorithm but avoids index structure construction and tries to minimize the number of I/Os.

➢ It divides the memory buffer space into two halves and the data set into several logical blocks.

➢ By carefully choosing the order in which blocks are loaded into each half, I/O efficiency can be achieved.

3. Cell-based algorithm:

➢ To avoidO(n2) computational complexity, a cell-based algorithm was developed for memory-resident data sets.

➢ Its complexity is O(ck + n), where c is a constant depending on the number of cells and k is the dimensionality.

➢ In this method, the data space is partitioned into cells with a side length equal to dmin/2$\sqrt{k}$.

➢ Each cell has two layers surrounding it.

➢ The first layer is one cell thick, while the second is $(d2/\sqrt{k-1})1e$ cells thick, rounded up to the closest integer.

➢ The algorithm counts outliers on a cell-by-cell rather than an object-by-object basis.

The density-based local outlier approach:

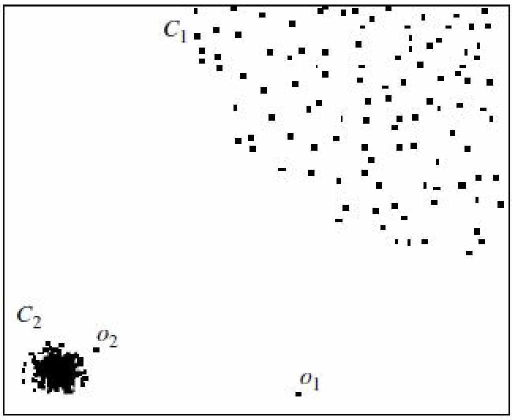
➢ Distance based outlier detection faces difficulty in identifying outliers if data is not uniformly distributed.

➢ Therefore this approach is used which depends on the overall distribution of the given set of data points

➢ Eg: Figure shows a simple 2-D data set containing 502 objects, with two obvious clusters.

➢ Cluster C1 contains 400 objects while Cluster C2 contains 100 objects.

➢ Two additional objects, o1 and o2 are clearly outliers.



However, by distance-based outlier detection only o1 is a reasonable outlier

- This brings us to the notion of local outliers.
- An object is a local outlier if it is outlyingrelative to its local neighborhood, particularly with respect to the density of the neighborhood.
- In this view, o2 is a local outlier relative to the density of C2.
- Object o1 is an outlier as well, and no objects in C1 are mislabeled as outliers.
- This forms the basis of density-based local outlier detection.
- Therefore unlike previous methods, it does not consider being an outlier as a binary property. Instead, it assesses the degree to which an object is an outlier.
- This degree of "outlierness" is computed as the local outlier factor (LOF) of an object.
- This degree of "outlierness" depends on how isolated the object is with respect to the surrounding neighborhood.
- This approach can detect both global and local outliers.

The deviation-based approach:

- This approach identifies outliers by examining the main characteristics of objects in a group.
- Objects that "deviate" from this description are considered outliers.
- Hence, in this approach the term deviation is typically used to refer to outliers.
- There are two techniques for deviation-based outlier detection.
- The first sequentially compares objects in a set, while the second employs an OLAP data cube approach.

4.2 Outlier Detection

Q2. Definition - What does Outlier Detection mean?

Ans :

Outlier detection is the process of detecting and subsequently excluding outliers from a given set of data. An outlier may be defined as a piece of data or observation that deviates drastically from the given norm or average of the data set. An outlier may be caused simply by chance, but it may also indicate measurement error or that the given data set has a heavy-tailed distribution.

Here is a simple scenario in outlier detection, a measurement process consistently produces readouts between 1 and 10, but in some rare cases we get measurements of greater than 20.

These rare measurements beyond the norm are called outliers since they "lie outside" the normal distribution curve.

There is really no standardized and rigid mathematical method for determining an outlier because it really varies depending on the set or data population, so its determination and detection ultimately becomes subjective. Through continuous sampling in a given data field, characteristics of an outlier may be established to make detection easier.

There are model-based methods for detecting outliers and they assume that the data are all taken from a normal distribution and will identify observations or points, which are deemed to be unlikely based on mean or standard deviation, as outliers. There are several methods for outlier detection:

- **Grubb's Test for Outliers :** This is based upon the assumption that the data are of a normal distribution and removes one outlier at a time with the test being iterated until no more outliers can be found.

- **Dixon's Q Test :** Also based upon normality of the data set, this method tests for bad data. It has been noted that this should be used sparingly and never more than once in a data set.
- **Chauvenet's Criterion :** This is used to analyze if the outlier is spurious or is still within the boundaries and be considered as part of the set. The mean and standard deviation are taken and the probability that the outlier occurs is calculated. The results will determine if it is should be included or not.
- **Pierce's Criterion :** An error limit is set for a series of observations, beyond which all observations will be discarded as they already involve such great error.

4.2.1 Types of Outliers

Q3. Explain different types of Outliers ?

Ans :

Outliers can be classified into three categories: point outliers, contextual outliers and collective outliers.

Point outliers

If an individual data point can be considered anomalous with respect to the rest of the data, then the datum is termed as a point outlier. This is the simplest type of outlier and it is the focus of the majority of research on outlier detection. For example, in Figure 2, points labeled O1 and O2 are typical point outliers since they are significantly different from the normal data points in regions G1 and G2.

Contextual outliers

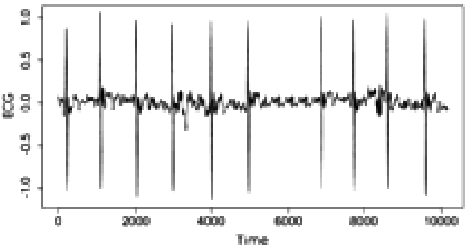
If an individual data instance is anomalous in a specific context (but not otherwise), then it is termed as a contextual (conditional) outlier. The notion of a context is induced by the structure of the data set and has to be specified as a part of the problem formulation. Each data point is defined with two sets of attributes:

- Contextual attributes, which are used to determine the context (or neighborhood); for example, time in time-series data, or longitude and latitude in spatial data sets.
- Behavioral attributes, which are used to define other characteristics of the data point, specific to the problem in hand; for example, number of sales at the specific location

Contextual outliers are detected using the values for the behavioral attributes in a specific context. Therefore, a data point might be an outlier in a given context, but could be considered normal in a different context.

These types of outliers have normally been explored in time-series data and spatial data.

Collective outliers



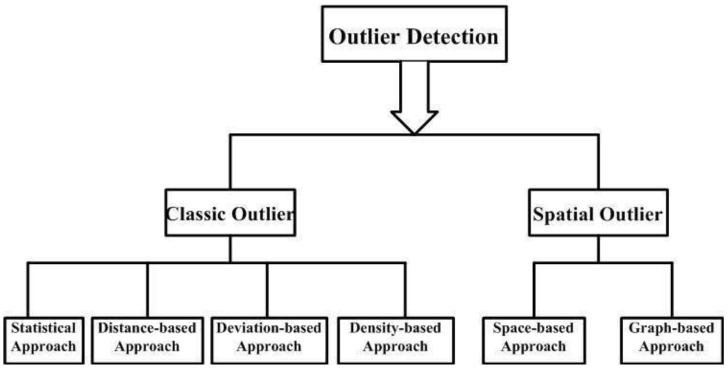
If a collection of data points is anomalous with respect to the entire data set, it is termed as a collective outlier. The individual data points inside the collective outlier may not be outliers by themselves alone, but their occurrence together as a collection is anomalous. Collective outliers can occur only in data sets in which data points are somehow related. For example, Figure 4 shows a human electrocardiogram output. Values in the range [5000, 7000] represent a collective outlier because the same low value exists for an abnormally long time. The low value itself is not an outlier, but its successive occurrence for long time is.

4.3 Outer Detection Approaches / Methods

Q4. Explain the Outlier Detection Approach.

Ans :

Outlier detection has been extensively studied in the past decennium and numerous methods hav e been created. Outlier detection approach is differentiating in two categories: classic outlier approach and spatial outlier approach. The classic outlier approach analyzes outlier based on transaction dataset, which can be grouped into statistical -based approach, distance-based approach, deviation-based approach, density based approach. The spatial outlier approaches analyze outlier based on spatial dataset, which can be grouped into space based approach and graph -based approach



Classic Outler

Classic outlier approach analyzes outlier based on transaction dataset, which consists of collections of items. A typical example is market basket data, where each transactions is the group of products purchased by a customer in a one transaction. Such data can also be augmented by additional "items" describing the customer or the context of the transaction. Commonly, transaction data is relative to other data to be simple for the outlier detection. Thus, most outlier approaches are researched on transaction data.

Statistical Approach

Statistical approaches were the oldest algorithms used for outlier identification, which cause a distribution model for the given data set and using a discordance test they detect outliers. In fact, many of the techniques described in both Barnett and Lewis [20] and Rousseeuw and Leroy are single dimensional. However, with the dimensions increasing, it becomes more difficult and inaccurate to make a model for dataset.

Distance-based Approach

The concept of distance-based outlier relies on the notion of the neighborhood of a point, typically, the k-nearest neighbors, and has been first introduced by Knorr and Ng. Distance-based outliers are those points for which there are less than k points within the distance in the input data set. Distance-based approach is not providing required knowledge about a ranking of outlier detection but it's used to define a preferable rank of the parameter.

Ramaswamy et al. Modified the definition of outlier introduced by Knorr and Ng and consider as outliers the top n point's p whose distance to their k-th nearest neighbor is greatest. Partition based technique are works as follow: Firstly they divide the input points using clustering technique and then prune that division that cannot contain outlier which is used to detect outliers.

The distanced-based approach is effective in rather low dimensions, because of the sparsity of high dimensional points, the approach is sensitive to the parameter λ and it is hard to figure out a-priori. As the dimensions increase, the method's effect and accuracy quickly decline.

Deviation-based Approach

Arning et a1. proposed a deviation-based method, which identify outliers by inspecting the main characteristics of objects in a dataset and objects that "deviate" from these features arc considered outliers.

Density-based Approach

The density-based approach estimates the density distribution of the data and identifies outliers as those lying in low-density regions. Breunig et al. assign a local outlier factor (LOF) to each point based on the local density of its neighborhood, which is determined by a user-given minimum number of points (MinPts). Papadimitriou et al. present LOCI (Local Correlation Integral) which uses statistical values based on the data itself to tackle the issue of choosing values for MinPts. Density-based techniques have the advantage that they can detect outliers that would be missed by techniques with a single, global criterion. However, data is usually sparse in high-dimensional spaces rendering density-based methods problematic.

4.4 Spatial Outlier

Q5. Explain about Spatial Outlier

Ans :

For spatial data, classic approaches have to be modified because of the qualitative difference between spatial and non- spatial attributes. Spatial dataset could be defined as a collection of spatially referenced objects. Attributes of spatial objects fall into two categories: spatial attributes and non s patial attributes. The spatial attributes include location, shape and other geometric or topological properties. Non spatial attributes include length, height, owner, building age and name. Comparisons between spatially referenced objects are based on non-spatial attributes.

Informally, a spatial outlier is a local instability, or an extreme observation with respect to its neighboring values, even though it may not be significantly different from the entire population. Detecting spatial outliers is useful in many applications of geographic information systems and spatial dataset.

The identification of spatial outliers can reveal hidden but valuable information in many applications, For example, it can help locate severe meteorological events, discover highway congestion segments, pinpoint military targets in satellite images, determine potential locations of oil reservoirs, and detect water pollution incidents.

Space-based Approach

Space-based outliers use Euclidean distances to define spatial neighborhoods. Kou et al. developed spatial weighted outlier detection algorithms which use properties such as center distance and common border length as weight when comparing non -spatial attributes. Adamet al. proposed an algorithm which considers both spatial relationship and semantic relationship among neighbors. Liu et al. proposed a method for detecting outliers in an irregularly-distributed spatial data set.

Graph-based Approach

Graph-based Approach uses graph connectivity to define spatial neighborhoods. Yufeng Kou et al. proposed a set of graph-based algorithms to identify spatial outliers, which first constructs a graph based on k - nearest neighbor relationship in spatial domain, assigns the non-spatial attribute differences as edge weights, and continuously cuts high- weight edges to identify isolated points or regions that are much dissimilar to their neighboring objects. The algorithms have two major advantages compared with the existing spatial outlier detection methods: accurate in detecting point outliers and capable of identifying region outliers .

RECENT ADVANCEMENTS IN OUTLIER DETECTION

SLOF Local Outlier Factor was proposed by Markus M. Breunig, Hans-Peter Kriegel, Ray-mond T. Ng and Jörg Sander. This method detects outlier by measuring the local deviation of a given data object with respect to its neighbors. Local outlier factor is based on the concept of local density. The object's neighbor is composed of the object's k - nearest neighbors. SLOF method is a density based out-lier detection method, the outliers detected by SLOF are local outliers. Based on the feature bagging approach, the SLOF method is robust and not quite sensitive to parameter k. The dimensions of the vector describe the features of the object. The objects' local density is calculated by the distances between objects. Finally, SLOF score of each object. If an object's SLOF score is approximate to 1, the object is a normal one, and if an object's SLOF score is significantly larger than 1, the object is an outlier.

Non-Parametric Composite Outlier Detection

Detection of the existence of data streams drawn from outlying distributions among data streams drawn from a typical distribution is investigated. It is assumed that the typical distribution is known and the outlying distribution is unknown. The generalized likelihood ratio test (GLRT) for this problem is constructed. With knowledge of the Kullback - Liebler divergence between the outlier and typical distributions, the GLRT is shown to be exponentially consistent (i.e, the error risk function decays exponentially fast). It is also shown that with knowledge of the Chernoff distance between the outlying and typical distributions, the same risk decay exponent as the parametric model can be achieved by using the GLRT. It is further shown that, without knowledge of the distance between the distributions, there does not exist an exponentially consistent test, although the GLRT with a diminishing threshold can still be consistent[25].

4.5 Statistical Methods for Data Mining

Q6. Explain the Statistical Approach in Data Mining?

Ans :

There are two problems in modern science: too many people using different terminology to solve the same problems and even more people using the same terminology to address completely different issues. This is particularly relevant to the relationship between traditional statistics and the new emerging field of knowledge data discovery (KDD) and data mining (DM). The explosive growth of interest and research in the domain of KDD and DM of recent years is not surprising given the proliferation of low-cost computers and the requisite software, low-cost database technology (for collecting and storing data) and the ample data that has been and continues to be collected and organized in databases and on the web.

Statistics is the traditional field that deals with the quantification, collection, analysis, interpretation, and drawing conclusions from data. Data mining is an interdisciplinary field that draws on computer sciences (data base, artificial intelligence, machine learning, graphical and visualization models), statistics and engineering (pattern recognition, neural networks). DM involves the analysis of large existing data bases in order to discover patterns and relationships in the data, and other findings (unexpected, surprising, and useful). Typically, it differs from traditional statistics on two issues: the size of the data set and the fact that the data were initially collected for purpose other than the that of the DM analysis. Thus, *experimental design*, a very important topic in traditional statistics, is usually irrelevant to DM. On the other hand

Asymptotic analysis, sometimes criticized in statistics as being irrelevant, becomes very relevant in DM.

While in traditional statistics a data set of 100 to 104 entries is considered large, in DM even 104 may be considered a small set fit to be used as an example, rather than a problem encountered in practice. Problem sizes of 107 to 1010 are more typical. It is important to emphasize, though, that data set sizes are not all created equal. One needs to distinguish between the number of cases (observations) in a large data set (n), and the number of features (variables) available for each case (m). In a large data set,n ,m or both can be large, and it does matter which, a point on which we will elaborate in the continuation. Moreover these definitions may change wh en the same data set is being used for two different purposes. We will address the following issues which are highly relevant to DM:

- Size
- Curse of Dimensionality
- Assessing uncertainty
- Automated analysis
- Algorithms for data analysis in Statistics
- Visualization
- Scalability
- Sampling
- Modelling relationships
- Model selection
- Statistical Issues in DM
- Size of the Data and Statistical Theory

Traditional statistics emphasizes the mathematical formulation and validation of a methodology, and views simulations and empirical or practical evidence as a less form of validation. The emphasis on rigor has required proof that a proposed method will work prior to its use. In contrast, computer science and machine learning use experimental validation methods. In many cases mathematical analysis of the performance of a statistical algorithm is not feasible in a specific setting, but becomes so when analyzed asymptotically. At the same time, when size becomes extremely large, studying performance by simulations is also not feasible. It is therefore in settings typical of DM problems that asymptotic analysis becomes both feasible and appropriate. Interest- ingly, in classical asymptotic analysis the number of cases n tends to infinity. In more contemporary literature there is a shift of emphasis to asymptotic analysis where the number of variables m tends to infinity.

The curse of dimensionality and approaches to address it The curse of dimensionality is a well documented and often cited fundamental problem. Not only do algorithms face more difficulties as the the data increases in dimension, but the structure of the data it- self changes This phenomenon becomes very evident when looking for the k-Nearest Neighbors of a point in high-dimensional space. The points are so far away from each other that the radius of the neighborhood becomes ex- tremely large.

The main remedy offered for the curse of dimensionality is to use only part of the available variables per case, or to combine variables in the data set in a way that will summarize the relevant information with fewer variables. This dimension reduction is the essence of what goes on in the data warehousing stage of the DM process, along with the cleansing of the data. It is an important and time-consuming stage of the DM operations, accounting for 80-90% of the time devoted to the analysis.

The dimension reduction comprises two types of activities: the first is quantifying and summarizing information into a number of variables, and the second is further reducing the variables thus constructed into a workable number of combined variables.

Assessing Uncertainty

Assessing the uncertainty surrounding knowledge derived from data is recognized as a the central theme in statistics. The concern about the uncertainty is down-weighted in KDD, often because of the myth that all relevant data is available in DM. Thus, standard errors of averages, for example, will be ridiculously low, as will prediction errors. On the other hand experienced users of DM tools are aware of the variability and uncertainty involved. They simply tend to rely on seemingly

"Non-statistical" technologies such as the use of a training sample and a test sample. Interestingly the latter is a methodology widely used in statistics, with origins going back to the 1950s. The use of such validation methods, in the form of cross-validation for smaller data sets, has been a common practice in exploratory data analysis when dealing with medium size data sets.

Some of the insights gained over the years in statistics regarding the use of these tools have not yet found their way into DM. Take, for example, data on food store baskets, available for the last four years, where the goal is to develop a prediction model. A typical analysis will involve taking a random training sample from the data, then testing the model on the training sample, with the results guiding us as to the choice of the most appropriate model. However, the model will be used next year, not last year. The main uncertainty surrounding its conclusions may not stem from the person to person variability captured by the differences between the values in the training sample, but rather follow from the year to year variability. If this is the case, we have all the data, but only four observations.

Automated Analysis

The inherent dangers of the necessity to rely on automatic strategies for analyzing the data, another main theme in DM, have been demonstrated again and again. There are many examples where trivial non- relevant variables, such as case number, turned out to be the best predictors in automated analysis. Similarly, variables displaying a major role in predicting a variable of interest in the past, may turn out to be useless because they reflect some strong phenomenon not expected to occur in the future. In spite of these warnings, it is clear that large parts of the analysis should be automated, especially at the warehousing stage of the DM.

This may raise new dangers. It is well known in statistics that having even a small proportion of outliers in the data can seriously distort its numerical summary. Such unreasonable values, deviating from the main structure of the data, can usually be identified by a careful human data analyst, and excluded from the analysis. But once we have to warehouse information about millions of customers, summarizing the information about each customer by a few numbers has to be automated and the analysis should rather deal automatically with the possible impact of a few outliers.

Statistical theory and methodology supply the framework and the tools for this endeavor. A numerical summary of the data that is not unboundedly influenced by a negligible proportion of the data is called a resistant summary. According to this definition the average is not re- sistant, for even one straying data value can have an unbounded effect on it. In contrast, the median is resistant. A resistant summary that retains its good properties under less than ideal situations is called a robust summary, the á-trimmed mean (rather than the median) being an example of such. The concepts of robustness and resistance, and the development of robust statistical tools for summarizing location, scale, and relationships, were developed during the 1970's and the 1980's.

Algorithms for Data Analysis in Statistics

Computing has always been a fundamental to statistic, and it remained so even in times when mathematical rigorousity was most highly valued quality of a data analytic tool. Some of the important computational tools for data analysis, rooted in classical statistics, can be found in the following list: efficient estimation by maximum likelihood, least squares and least absolute deviation estimation, and the EM algorithm; analysis of variance (ANOVA, MANOVA, ANCOVA), and the analysis of repeated measurements; nonparametric statistics; log-linear analysis of categorial data; linear regression analysis, generalized additive and linear models, logistic regression, survival analysis, and discriminant analysis; frequency domain (spectrum) and time domain (ARIMA) methods for the analysis of time series; multivariate analysis tools such as factor analysis, principal component and later independent component analyses, and cluster analysis; density estimation, smoothing and de- noising, and classification and regression trees (decision trees); Bayesian networks and the Monte Carlo Markov Chain (MCMC) algorithm for Bayesian inference.

Visualization

Visualization of the data and its structure, as well as visualization of the conclusions drawn from the data, are another central theme in DM. Visualization of quantitative data as a major activity flourished in the statistics of the 19th century, faded out of favor through most of the 20th century, and began to regain importance in the early 1980s. This importance in reflected in the development of the Journal of Computational and Graphical Statistics of the American Statistical Association. Both the theory of visualizing quantitative data and the practice have dramatically changed in recent years.

Much can be gained in DM by mining the knowledge about visualization available in statistics, though the visualization tools of statistics are usually not calibrated for the size of the data sets commonly dealt within DM. Take for example the extremely effective Boxplots display, used for the visual comparisons of batches of data. A well-known rule determines two fences for each batch, and points outside the fences are individually displayed. There is a traditional default value in most statistical software, even though the rule was developed with batches of very small size in mind (in DM terms). In order to adapt the visualization technique for routine use in DM, some other rule which will probably be adaptive to the size of the batch should be developed.

Scalability

In machine learning and data mining scalability relates to the ability of an algorithm to scale up with size, an essential condition being that the storage requirement and running time should not become infeasible as the size of the problem increases. Designing scalable algorithms for more complex tasks, such as decision tree modeling, optimization algorithms, and the mining of association rules, has been the most active research area in DM. Altogether, scalability is clearly a fundamental problem in DM mostly viewed with regard to its algorithmic aspects. We want to highlight the duality of the problem by suggesting that concepts should be scalable as well. In this respect, consider the general belief that hypothesis testing is a statistical concept that has nothing to offer in DM. The usual argument is that data sets are so large that every hypothesis tested will turn out to be statistically significant - even if differences or relationships are minuscule.

Sampling

Sampling is the ultimate scalable statistical tool: if the number of cases n is very large the conclusions drawn from the sample depend only on the size of the sample and not on the size of the data set. It is often used to get a first impression of the data, visualize its main features, and reach decisions as to the strategy of analysis. In spite of its scalability and usefulness sampling has been attacked in the KDD community for its inability to find very rare yet extremely interesting pieces of knowledge.

Sampling is a very well developed area of statistics (see for example Cochran, 1977), but is usually used in DM at the very basic level. Stratified sampling, where the probability of picking a case changes from one stratum to another, is hardly ever used. But the questions are relevant even in the simplest settings: should we sample from the few positive re-sponses at the same rate that we sample from the negative ones? When studying faulty loans, should we sample larger loans at a higher rate? A thorough investigations of such questions, phrased in the realm of particular DM applications may prove to be very beneficial.

4.5.1 Modeling Relationships using Regression Models

Q7. What is the importance of Modeling Relationship with Regression Model?

Ans :

Demonstrating that statistics, like data mining, is concerned with turning data into information and knowledge, even though the terminology may differ, in this section we present a major statistical approach being used in data mining, namely regression analysis. In the late 1990s, statistical methodologies such as regression analysis were not included in commercial data mining packages. Nowadays, most commercial data mining software includes many statistical tools and in particular regression analysis. Although regression analysis may seem simple and anachronistic, it is a very powerful tool in DM with large data sets, especially in the form of the generalized linear models (GLMs). We emphasize the assumptions of the models being used and how the underlying approach differs from that of machine learning.

Linear Regression Analysis

Regression analysis is the process of determining how a variable y is related to one, or more, other variables $x_1, ..., x_k$. The y is usually called the dependent variable and the x_i's are called the independent or explanatory variables. In a linear regression model we assume that

$$y_i = \beta_0 = \sum_{j=1}^{k} \beta_j x_{ji} + \in_i \quad i = 1, ..., M$$

and that the ε_i's are independent and are identically distributed as N (0, σ^2) and M is the number of data points. The expected value of yi is given by

$$E(y_i) = \beta_0 + \sum_{j=1}^{k} \beta_j x_{ji}$$

To estimate the coefficients of the linear regression model we use the least square estimation which gives results equivalent to the estimators obtained by the maximum likelihood method. Note that for the linear regression model there is an explicit formula of the β's

In Matrix Form

$$X = \begin{pmatrix} 1 & x_{11} & \cdots & x_{1k} \\ 1 & x_{21} & \cdots & x_{2k} \\ 1 & \vdots & \ddots & \vdots \\ 1 & x_{k1} & \cdots & x_{Mk} \end{pmatrix}$$

Generalized Linear Models

Although in many cases the set of linear function is good enough to model the relationship between the stochastic response y as a function of x it may not always suffice to represent the relationship. The generalized linear model increases the family of functions F that may represent the relationship between the response y and x. The tradeoff is between having a simple model and a more complex model representing the relationship between y and x. In the general linear model the distribution of y given x does not have to be normal, but can be any of the distributions in the exponential family (see McCullagh and Nelder, 1991). Instead of the expected value of y|x being a linear function,

$$g(E(y_i)) = \beta_0 + \sum_{j=1}^{k} \beta_j x_{ji}$$

In the generalized additive models, g(E(y_i)) need not to be a linear function of x but has the form:

$$g(E(y_i)) = \beta_0 + \sum_{j=1}^{k} \sigma_j(x_{ji})$$

4.6 Proximity-based Outlier Detection

Q8. Explain the process of Data mining using Proximity Detection ?

Ans :

Given a set of objects in feature space, a distance measure can be used to quantify the similarity between objects. Intuitively, objects that are far from others can be regarded as outliers. Proximity-based approaches assume that the proximity of an outlier object to its nearest neighbors significantly deviates from the proximity of the object to most of the other objects in the data set.

Proximity-based techniques define a data point as an outlier, if its locality (or proximity) is sparsely populated. The proximity of a data point may be defined in a variety of ways, which are subtly different from one another, The most common ways of defining proximity for outlier analysis are as follows:

- **Cluster-based:** The non-membership of a data point in any cluster, its distance from other clusters, and the size of the closest cluster, are used as criteria in order to compute the outlier score. The clustering problem has a complementary relationship to the outlier detection problem, in which points either belong to clustersor outliers.
- **Distance-based:** The distance of a data point to its k-nearest neighbor (or other variant) is used in order to define proximity. Data points with large k-nearest neighbor distances are defined as outliers. Distance-based algorithms typically perform the analysis at a much more detailed granularity than the other two methods. On the other hand, this greater granularity often comes at a significant computational cost.
- **Density-based:** The number of other points within a specified local region (grid region or distance-based region) of a data point, is used in order to define local density. These local density values may be converted into outlier scores. Other kernel-based methods or statistical methods for density estimation may also be used. The major difference between clustering and density-based methods is that clustering methods partition the data points, whereas density based methods partition the data space. Clearly, all these techniques are closely related, because they are based on some notion of proximity (or similarity). The major difference is at the detailed level of how this proximity is defined. These different ways of defining outliers may have different advantages and disadvantages, and this chapter will try to address these issues in a unified way. Furthermore, most of these methods generally work well when the data is highly clustered, and the outliers can be clearly distinguished from dense regions of the data. In many cases, the distinctions between these different classes of methods become blurred, when the definition of sparsity combines1 more than one of these concepts. One major difference between distance-based and the other two classes of methods is the level of granularity at which the analysis is performed.

In both clustering- and density-based methods, the data is pre-aggregated before outlier analysis by either partitioning the points or the space. The data points are compared to the distributions in this pre-aggregated data for analysis. On the other hand, in distance-based methods, the k-nearest neighbor distance to the original data points (or a similar variant) is computed as the outlier score. Thus, the analysis in nearest neighbor methods is performed at a much more detailed level of granularity. Correspondingly, these methods provide different tradeoffs between effectiveness and efficiency for data sets of different sizes. Nearest neighbor methods may require O(N2) time to compute all k-nearest neighbor distances for a data set with N records, unless indexing techniques are used to speed up the computations. Even in those cases, nearest neighbor methods can sometimes be slow, if the underlying data patterns do not support efficient pruning. On the other hand, nearest neighbors can often provide more detailed and accurate analysis, especially for smaller data sets, which may not support robust clustering or density analysis. Thus, the particular choice of the model should depend on the nature of the data and its size. Different methods may be more effective in different scenarios.

4.7 Cluster-based Outlier Detection

Q9. Explain the importance of Cluster based Data Mining ?

Ans :

Outlier detection has important applications in the field of data mining, such as fraud detection, customer behavior analysis, and intrusion detection. Outlier detection is the process of detecting the data objects which are grossly different from or inconsistent with the remaining set of data. Outliers are traditionally considered as single points; however, there is a key observation that many abnormal events have both temporal and spatial locality, which might form small clusters that also need to be deemed as outliers. In other words, not only a single point but also a small cluster can probably be an outlier. we present a new definition for outliers: cluster-based outlier, which is meaningful and provides importance to the local data behavior, and how to detect outliers by the clustering algorithm LDBSCAN which is capable of finding clusters and assigning LOF to single points.

The clustering algorithm is used to group objects into significant subclasses and the clustering data streams are a subarea of mining data streams. The clustering algorithms for data streams should be adaptive in the sense that up to date clusters are obtainable at any time, taking new data items into account as soon as they arrive. There are different types of clustering algorithms are fitting for different types of applications they are chased by Hierarchical clustering algorithm, Partition clustering algorithm, Density based clustering algorithm and Grid based clustering algorithm. Clustering is defined as an unsupervised problem. There are no predefined class label exists for the data points. Cluster analysis is used in a number of applications such as data analysis, image processing, Stock market analysis etc.

Outlier Detection

Outlier detection has a wide range of applications such as fraud detection, intrusion detection, and credit card analysis. It is further complicated by the fact that in many cases outliers have to be detected from a large volume of data growing at an unlimited rate. Traditional outlier detection algorithms cannot be functional to data stream efficiently, since the data stream is potentially infinite and evolving continuously. It has to be routed within an exact time constraint and limited space, thus outlier detection in data stream imposes great challenges are followed. The cluster based outlier detection is a best technique to supervise this problem.

K-Means

The K-means algorithm is the best known partitioned clustering algorithm. It is a simple method for estimating the mean (vector) of set K groups. The most widely used K-means among all clustering algorithms is due to its efficiency and simplicity. The K-means algorithm is as follows

Algorithm k-means (k, D)

```
1. chooses k data points as the initial cancroids (cluster centers)
2. repeat
3. for each data point x ∈ D do
4. compute the distance from x to each centered;
5. assign × to the closest centered
       // a centered represents a cluster
6. end for
7. re-compute the centered using the current cluster memberships
8. until the stopping criterion is met
```

Spatial Outler Detection

Spatial outliers are objects which have behavioral attribute values that are distinct from those of their surrounding spatial neighbors. Thus, spatial continuity plays an important role in the identification of anomalies.

This is an analogous principle to the concept of temporal continuity, which was discussed in the chapters on time series outlier detection. One of the most fundamental rules of spatial data is as follows:

"Everything is related to everything else, but nearby objects are more related than distant objects."

Spatial data does not contain only spatial attributes, just as temporal data does not necessarily contain only temporal attributes. Instead, spatial locations form the contextual points at which other behavioral attributes of interest are measured. Thus, two kinds of attributes may be available:

- **Behavioral Attributes:** This is the attribute of interest which is measured for each object. For example, this could correspond to sea surface temperatures, wind speeds, car speeds, disease outbreak numbers, the color of an image pixel, etc. It is possible to have more than one behavioral attribute at a spatial location in given application.
- **Contextual Attributes (Spatial Location):** This is the location of interest at which the behavioral attribute is measured. Typically, this would contain two or three dimensions, when the data is expressed in terms of coordinates. In some cases, the contextual attributes may be more complex, and may be expressed at the granularity of a region of interest, such as a county, zip-code etc. Alternatively, in an imaging application, the contextual attributes may correspond to individual pixels.

Spatial data shares a number of similarities with time-series data, in which one or more properties of interest (behavioral attributes) are measured at a given moment in time (contextual attribute). In fact, in spatiotemporal data, the contextual attributes may also contain a temporal component. This can be used to determine important spatiotemporal anomalies (or events) based on the underlying dynamics. For example, the dynamics of behavioral attributes such as humidity, wind speeds, sea surface temperatures and pressure can be used in order to identify and predict anomalous weather events. In such cases, both spatial and temporal continuity can play an important role in the prediction. It is also possible for the data to be purely spatiotemporal, in which no other behavioral attributes are present, and the trajectories of objects are measured over time. In such cases, no attribute needs to be treated as behavioral, since a joint analysis of both components provides the best insights in many applications. In some cases, it may be helpful to treat the temporal component as the contextual attribute, and the spatial components as the behaviorial attributes.

For example, in a two dimensional real-time trajectory mining application, this can be modeled as a bivariate time series, in which the evolving X-coordinate and Y - coordinate values are individual time series. In the offline trajectory shape analysis scenario, anomalies may correspond to unusual shapes, irrespective of their temporal provenance. The latter case is mostly a spatial analytics scenario, and the temporal aspects of the problem are limited. Therefore, trajectory-based applications can be modeled in multiple ways, depending upon the needs of the underlying application.

Spatial data is common in many real applications, such as the following:

- **Meteorological Data:** Numerous weather parameters are typically measured at different geographica locations, which may be used in order to predict anomalous weather patterns in the underlying data.
- **Traffic Data:** Moving objects may be associated with many parameters such as speed, direction etc. The location of an object is its contextual attribute. In many cases, such data is also spatiotemporal, since it has a temporal component. Finding anomalous behavior of moving objects [83] can provide numerous insights.

4.8 Modeling Classification Based Outlier Detection System

Q8. Explain the Classification of Models in Outlier Detection System?

Ans :

Problem Specification

Even though the training data set is heavily biased or unbalanced, we can model classification based outlier detection and any new or fresh data can be classified accordingly Support Vector Machines, Fuzzy Art Neural Networks can be used to solve such one class classification problem. Sometimes, the model can detect new outliers that may not appear close to any outlier objects in the training set. This occurs as long as such new outliers fall outside the decision boundary of the normal class. Most difficult task is to obtain high-quality training data. Moreover, the problem becomes much difficult if the data is a multi-dimensional one By assigning variables to different groups, the model can reduce the size of the data set and then can improve the efficiency of outlier detection. Second idea is to eliminate some variables by using the data reduction methods such as Principal components and factor analysis.

In this work, some of the popular classification algorithms for outlier detection in multi-dimensional cancer data set are evaluated and so, proposed dimensionality reduction and feature selection methods for measuring the training performance and accuracy testing issues in classification based outlier detection method.

Supervised, Semi-Supervised and Unsupervised Method

Supervised approach develops a predicative model for normal as well as outlier classes and new model instance is compared against it. But in semi-supervised outlier detection mode, training data set is available for normal data set. In a unsupervised outlier detection mode, the training data is not available, the data instances which are frequent are treated as normal and others are outliers.

Statistical Methods, Proximity-Based methods

Statistical methods can be used on the assumption of the data normality. Data objects that do not follow these methods are treated as outliers. In the proximity based method, the closeness of outlier object to its nearest objects significantly different from the closeness of the object to the most of other objects in the data set. So the varies tests like Grubb's test, Rosner's test and Dixon's test for outlier detection can be used on the normally distributed data.

Classification based Methods

If a training data set with class labels is available then outlier detection is treated as a classification problem. Once the classification model is constructed, the outlier detection process is very fast one. It only needs to compare the objects against the model learned from the training data.

Dimensionality Reduction Algorithm

The number of variables that are used to describe an object is the dimensionality of that object. The search for features in deep relation between variables is known as data exploration It is necessary to reduce the number of dimension using dimensionality reduction methods. The dimensionality reduction is the process of a search for a small set of features to describe a large set of observed dimensions. Since the small set is much faster than large one, it decreases the computational processing time

Reasons for Dimensionality Reduction

- Some features of cancer data may be irrelevant
- To visualize high dimensional cancer data on a low dimensional space data
- "Intrinsic" dimensionality of the cancer data may be smaller than the number of features
- In some cases, data analysis such as regression or classification can be more accurately in reduced space than in the original space.

Model of Classification Based Outlier Detection with Feature Selection

In this work, we have evaluated three feature selection based dimensionality reduction techniques, 1.Chi Square, 2.Information Gain and 3.Gini Index

Chi Square

The Chi-square measures the degree of dependence of feature of class The feature and the class are considered dependent if chi-square is greater than the critical value determined by degrees of freedom.. Such features are selected.

$$\chi^2(f) = \sum_{u\in V}\sum_{i=1}^{m}\frac{(A_i(f = u) - E_i(f = u)^2}{E_i(f = u)}$$

Information Gain

It is a measure of how an attribute is for predicting the class of each of the training data. Information gain is a measure of reduction in uncertainty once the value of an attribute is known. The information gain of a feature *f* is defined as

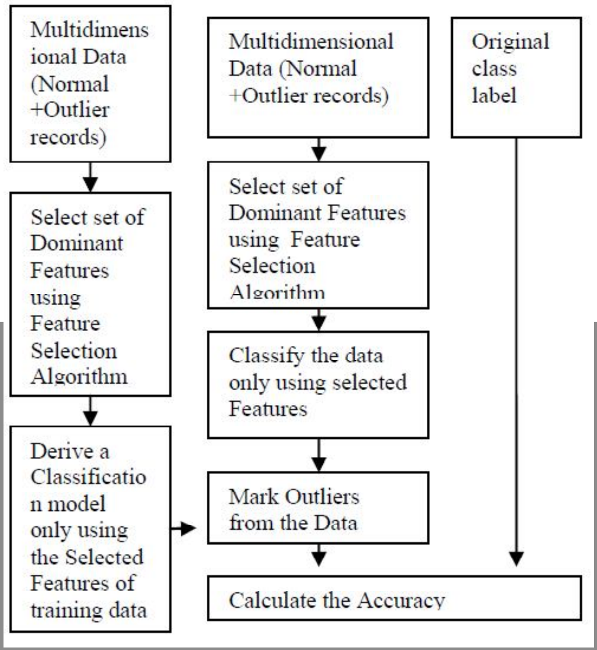
$$G(f) = \sum_{i=1}^{m} P(c_t) \log P(c_i) + \sum_{v\in V}\sum_{i=1}^{m} P(f = v)\, P(c_i \mid f = v) \log P(c_i \mid f = v)$$

Gini Index

The Gini coefficient or Index is a measure of resource inequality in a population developed by the Italian statistician Corrado Gini and published in his 1912 paper "Variabilità e mutabilità". It can be used to measure any form of uneven distribution. Index varies form 0 to 1, zero means no inequality (no uncertainty) and 1 means maximum possible inequality (maximum uncertainty) The Gini coefficient is often calculated with the more practical Brown Formula

$$G = |1 - \sum_{k=1}^{n} (X_k - X_{k-1})(Y_k + Y_{k-1})$$

Where, Gini coefficient and X_k: :cumulated portion of one variable for k= 0 to 1 with $X_0 = 0$, $X_n = 1$. Y_k : cumulated proportion of the target variable, for k = 0,...,n, with $Y_0 = 0$, $Y_n = 1$.


Multidimensional Data (Normal +Outlier records)
Select set of Dominant Features using Feature Selection Algorithm
Derive a Classification model only using the Selected Features of training data
Multidimensional Data (Normal +Outlier records)
Select set of Dominant Features using Feature Selection Algorithm
Classify the data only using selected Features
Mark Outliers from the Data
Original class label
Calculate the Accuracy

The Used Classification Algorithms

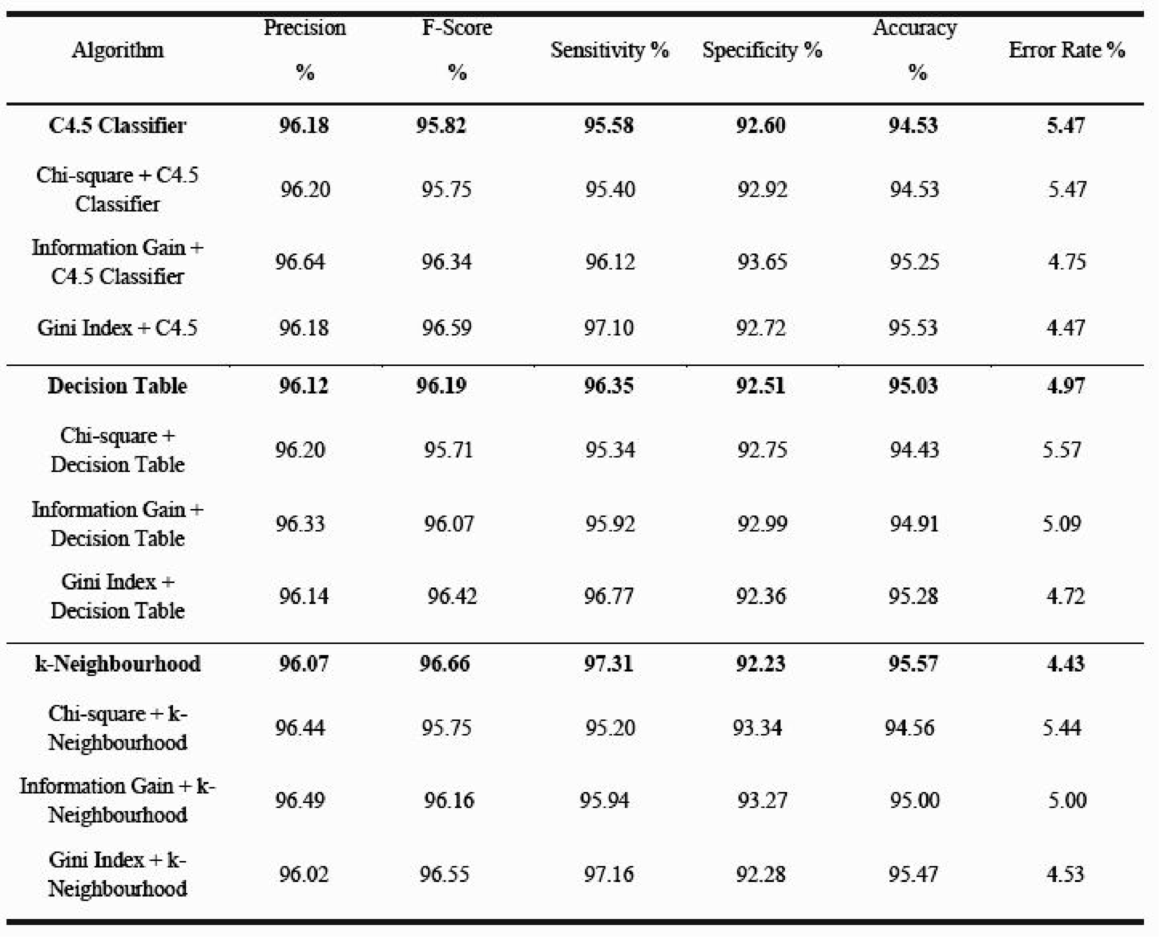
➢ **Classifier:** C4.5 is a tree pruning algorithm in Decision tree-based approach and creates a tree model by using values of only one attribute at a time.

➢ **Decision Table Classifier:** Decision table is a hierarchical breakdown of the data, with two attributes at each level of the hierarchy and the most important attributes for classifying the data.

➢ **K-Nearest Neighbors Classifier:** The "nearest" measurement refers to the Euclidean distance between two instances. For example, the Euclidean distance between ti and tj is

$$D(t_i, t_j) = \sqrt{\sum_{i=1}^{p} (x_{ik} - x_{jk})^2}$$ where p is number of attributes.

Comparison between Algorithms



Algorithm	Precision %	F-Score %	Sensitivity %	Specificity %	Accuracy %	Error Rate %
C4.5 Classifier	**96.18**	**95.82**	**95.58**	**92.60**	**94.53**	**5.47**
Chi-square + C4.5 Classifier	96.20	95.75	95.40	92.92	94.53	5.47
Information Gain + C4.5 Classifier	96.64	96.34	96.12	93.65	95.25	4.75
Gini Index + C4.5	96.18	96.59	97.10	92.72	95.53	4.47
Decision Table	**96.12**	**96.19**	**96.35**	**92.51**	**95.03**	**4.97**
Chi-square + Decision Table	96.20	95.71	95.34	92.75	94.43	5.57
Information Gain + Decision Table	96.33	96.07	95.92	92.99	94.91	5.09
Gini Index + Decision Table	96.14	96.42	96.77	92.36	95.28	4.72
k-Neighbourhood	**96.07**	**96.66**	**97.31**	**92.23**	**95.57**	**4.43**
Chi-square + k-Neighbourhood	96.44	95.75	95.20	93.34	94.56	5.44
Information Gain + k-Neighbourhood	96.49	96.16	95.94	93.27	95.00	5.00
Gini Index + k-Neighbourhood	96.02	96.55	97.16	92.28	95.47	4.53

4.9 Mining Complex Data Types

Q9. Explain the importance an types of Mining Complex Data Types?

Ans :

In this section, we outline the major developments and research efforts in mining complex data types. Complex data types are summarized covers mining sequence data such as time-series, symbolic sequences, and biological sequences. Addresses mining other kinds of data, including spatial data, spatiotemporal data, moving-object data, cyber-physical system data, multimedia data, text data, web data, and data streams.

Set-valued attribute

- Generalization of each value in the set into its corresponding higher-level concepts
- Derivation of the general behavior of the set, such as the number of elements in the set, the types or value ranges in the set, or the weighted average for numerical data
- hobby = {tennis, hockey, chess, violin, nintendo_games} generalizes to {sports, music, video_games}

List-valued or a sequence-valued attribute

Same as set-valued attributes except that the order of the elements in the sequence should be observed in the generalization

Generalizing Spatial and Multimedia Data

Spatial data:

- Generalize detailed geographic points into clustered regions, such as business, residential, industrial, or agricultural areas, according to land usage
- Require the merge of a set of geographic areas by spatial operations

Image data:

- Extracted by aggregation and/or approximation
- Size, color, shape, texture, orientation, and relative positions and structures of the contained objects or regions in the image

Music data:

- Summarize its melody: based on the approximate patterns that repeatedly occur in the segment
- Summarized its style: based on its tone, tempo, or the major musical instruments played

Generalizing Object Data

Object identifier : generalize to the lowest level of class in the class/subclass hierarchies

Class composition hierarchies

- generalize nested structured data
- generalize only objects closely related in semantics to the current one

Construction and mining of object cubes

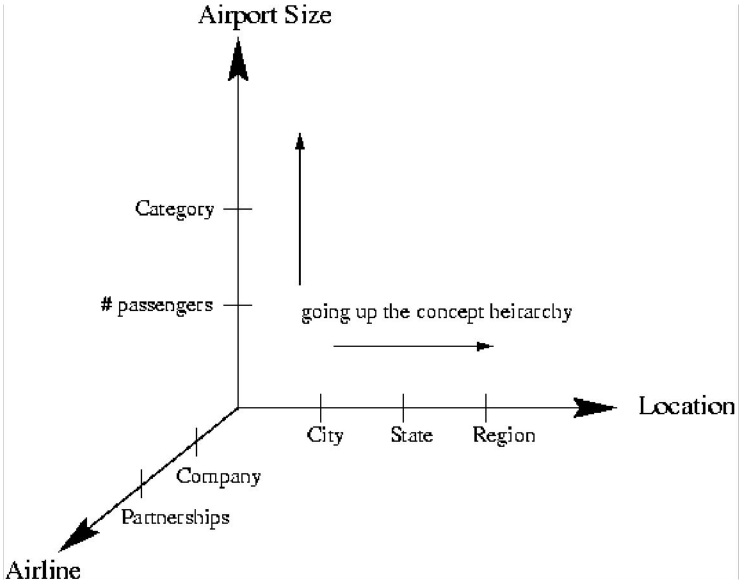
- Extend the attribute-oriented induction method

Apply a sequence of class-based generalization operators on different attributes Continue until getting a small number of generalized objects that can be summarized as a concise in high-level terms

- For efficient implementation

Examine each attribute, generalize it to simple-valued data Construct a multidimensional data cube (object cube)

Problem : It is not always desirable to generalize a set of values to single-valued data



Mining spatial databases

Spatial Data Warehousing

Spatial data warehouse: Integrated, subject-oriented, time-variant, and nonvolatile spatial data repository for data analysis and decision making.

Spatial data integration: a big issue

- Structure-specific formats (raster- vs. vector-based, OO vs. relational models, different storage and indexing)
- Vendor-specific formats (ESRI, MapInfo, Integraph)

Spatial data cube: multidimensional spatial database

Both dimensions and measures may contain spatial components Estimation of Trend Curve The freehand method

- Fit the curve by looking at the graph
- Costly and barely reliable for large-scaled data mining

The least-square method

- Find the curve minimizing the sum of the squares of the deviation of points on the curve from the corresponding data points

The moving-average method

- Eliminate cyclic, seasonal and irregular patterns
- Loss of end data
- Sensitive to outliers

Discovery of Trend in Time-Series

Estimation of seasonal variations

Seasonal Index

Set of numbers showing the relative values of a variable during the months of the year E.g., if the sales during October, November, and December are 80%, 120%, and 140% of the average monthly sales for the whole year, respectively, then 80, 120, and 140 are seasonal index numbers for these months.

Depersonalized Data

Data adjusted for seasonal variations E.g., divide the original monthly data by the seasonal index numbers for the corresponding months.

Estimation of cyclic variations

If (approximate) periodicity of cycles occurs, cyclic index can be constructed in much the same manner as seasonal indexes

Estimation of irregular variations

- By adjusting the data for trend, seasonal and cyclic variations With the systematic analysis of the trend, cyclic, seasonal, and irregular components, it is possible to make long- or short-term predictions with reasonable quality.

4.10 An Introduction to Sequential Pattern Mining

An introduction to sequential pattern mining, an important data mining task with a wide range of applications from text analysis to market basket analysis. This blog post is aimed to be a short introductino. If you want to read a more detailed introduction to sequential pattern mining,

Q10. What is sequential pattern mining ?

Ans :

Data mining consists of extracting information from data stored in databases to understand the data and/or take decisions. Some of the most fundamental data mining tasks are clustering, classification, outlier analysis, and pattern mining. **Pattern mining** consists of discovering interesting, useful, and unexpected patterns in databases Various types of patterns can be discovered in databases such as frequent item sets, associations, sub graphs, sequential rules, and periodic patterns.

The task of sequential pattern mining is a data mining task specialized for analyzing sequential data, to discover sequential patterns. More precisely, it consists of discovering interesting subsequences in a set of sequences, where the interestingness of a subsequence can be measured in terms of various criteria such as its occurrence frequency, length, and profit. Sequential pattern mining has numerous real-life applications due to the fact that data is naturally encoded as sequences of symbols in many fields such as bioinformatics, e-learning, market basket analysis, texts, and webpage click-stream analysis.

Consider the following sequence database, representing the purchases made by customers in a retail store.

SID	Sequence
1	⟨{a,b}, {c}, {f,g}, {g}, {e}⟩
2	⟨{a,d}, {c}, {b}, {a,b,e,f}⟩
3	⟨{a}, {b}, {f, g}, {e}⟩
4	⟨{b}, {f, g}⟩

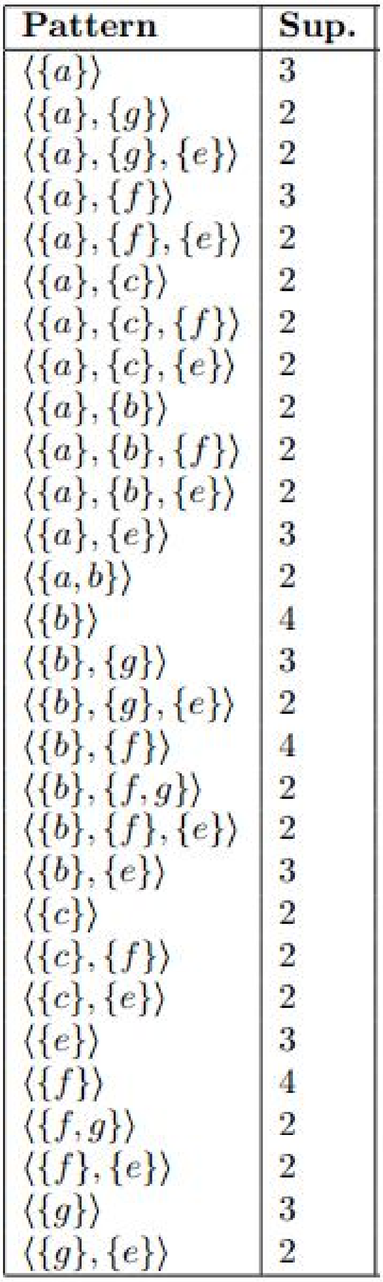
This database contains four sequences. Each sequence represents the items purchased by a customer at different times. A sequence is an ordered list of itemsets (sets of items bought together). For example, in this database, the first sequence (SID 1) indicates that a customer bought some items a and b together, then purchased an item c, then purchased items f and g together, then purchased an item g, and then finally purchased an item e.

Traditionally, sequential pattern mining is being used to find subsequences that appear often in a sequence database, i.e. that are common to several sequences. Those subsequences are called the frequent sequential patterns.

For example, in the context of our example, sequential pattern mining can be used to find the sequences of items frequently bought by customers. This can be useful to understand the behavior of customers to take marketing decisions.

To do sequential pattern mining, a user must provide a sequence database and specify a parameter called the minimum support threshold. This parameter indicates a minimum number of sequences in which a pattern must appear to be considered frequent, and be shown to the user. For example, if a user sets the minimum support threshold to 2 sequences, the task of sequential pattern mining consists of finding all subsequences appearing in at least 2 sequences of the input database. In the example database, 29 subsequences met this requirement. These sequential patterns are shown in the table below, where the number of sequences containing each pattern (called the support) is indicated in the right column of the table.

Sequential patterns found



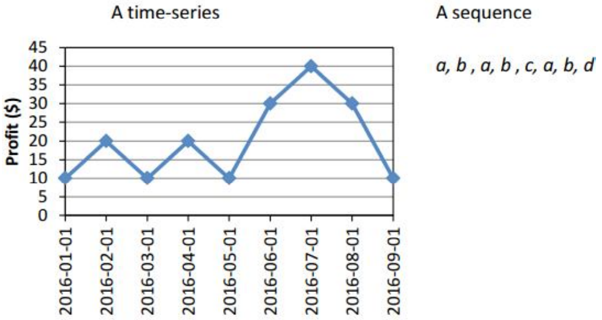
Pattern	Sup.
$\langle\{a\}\rangle$	3
$\langle\{a\},\{g\}\rangle$	2
$\langle\{a\},\{g\},\{e\}\rangle$	2
$\langle\{a\},\{f\}\rangle$	3
$\langle\{a\},\{f\},\{e\}\rangle$	2
$\langle\{a\},\{c\}\rangle$	2
$\langle\{a\},\{c\},\{f\}\rangle$	2
$\langle\{a\},\{c\},\{e\}\rangle$	2
$\langle\{a\},\{b\}\rangle$	2
$\langle\{a\},\{b\},\{f\}\rangle$	2
$\langle\{a\},\{b\},\{e\}\rangle$	2
$\langle\{a\},\{e\}\rangle$	3
$\langle\{a,b\}\rangle$	2
$\langle\{b\}\rangle$	4
$\langle\{b\},\{g\}\rangle$	3
$\langle\{b\},\{g\},\{e\}\rangle$	2
$\langle\{b\},\{f\}\rangle$	4
$\langle\{b\},\{f,g\}\rangle$	2
$\langle\{b\},\{f\},\{e\}\rangle$	2
$\langle\{b\},\{e\}\rangle$	3
$\langle\{c\}\rangle$	2
$\langle\{c\},\{f\}\rangle$	2
$\langle\{c\},\{e\}\rangle$	2
$\langle\{e\}\rangle$	3
$\langle\{f\}\rangle$	4
$\langle\{f,g\}\rangle$	2
$\langle\{f\},\{e\}\rangle$	2
$\langle\{g\}\rangle$	3
$\langle\{g\},\{e\}\rangle$	2

The patterns <{a}> and <{a}, {g}> are frequent and have a support of 3 and 2 sequences, respectively. In other words, these patterns appear in 3 and 2 sequences of the input database, respectively. The pattern <{a}> appears in the sequences 1, 2 and 3, while the pattern <{a}, {g}> appears in sequences 1 and 3. These patterns are interesting as they represent some behavior common to several customers. Of course, this is a toy example. Sequential pattern mining can actually be applied on database containing hundreds of thousands of sequences.

Another example of application of sequential pattern mining is text analysis. In this context, a set of sentences from a text can be viewed as sequence database, and the goal of sequential pattern mining is then to find subsequences of words frequently used in the text. If such sequences are contiguous, they are called “ngrams” in this context.

Can sequential pattern mining be applied to time series?

Besides sequences, sequential pattern mining can also be applied to time series (e.g. stock data), when discretization is performed as a pre-processing step. For example, the figure below shows a time series (an ordered list of numbers) on the left. On the right, a sequence (a sequence of symbols) is shown representing the same data, after applying a transformation. Various transformations can be done to transform a time series to a sequence such as the popular SAX transformation. After performing the transformation, any sequential pattern mining algorithm can be applied.



A time-series (left) and a sequence (right)

Sequential pattern mining implementations

To try sequential pattern mining with your datasets, you may try the open-source SPMF data mining software, which provides implementations of numerous sequential pattern mining algorithms:

It provides implementations of several algorithms for sequential pattern mining, as well as several variations of the problem such as discovering maximal sequential patterns, closed sequential patterns and sequential rules. Sequential rules are especially useful for the purpose of performing predictions, as they also include the concept of confidence.

4.11 Mining Sequence Patterns in Transactional Databases

Q11. What is sequential pattern mining in Transactional Data Base ?

Ans :

A sequence database consists of sequences of ordered elements or events, recorded with or without a concrete notion of time. There are many applications involving sequence data. Typical examples include customer shopping sequences, Web clickstreams, biological sequences, sequences of events in science and engineering, and in natural and social developments.

Sequential Pattern Mining: Concepts and Primitives

Sequential pattern mining is the mining of frequently occurring ordered events or subsequences as patterns. For retail data, sequential patterns are useful for shelf placement and promotions. This industry, as well as telecommunications and other businesses, may also use sequential patterns for targeted marketing, customer retention, and many other tasks. Other areas in which sequential patterns can be applied include Web access pattern analysis, weather prediction, production processes, and network intrusion detection.

Notice that most studies of sequential pattern mining concentrate on categorical (or symbolic) patterns, whereas numerical curve analysis usually belongs to the scope of trend analysis and forecasting in statistical time-series analysis,

"Given a set of sequences, where each sequence consists of a list of events (or elements) and each event consists of a set of items, and given a user-specified minimum support threshold of min sup, sequential pattern mining finds all frequent subsequences, that is, the subsequences whose occurrence frequency in the set of sequences is no less than min sup."

Let's establish some vocabulary for our discussion of sequential pattern mining. Let I = fI1, I2, .., Ipg be the set of all items. An itemset is a nonempty set of items.

A sequence is an ordered list of events. A sequence s is denoted he1e2e3 _ _ _eli, where event e1 occurs before e2, which occurs before e3, and so on. Event e j is also called an element of s. In the case of customer purchase data, an event refers to a shopping trip in which a customer bought items at a certain store. The event is thus an itemset, that is, an unordered list of items that the customer purchased during the trip. The itemset (or event) is denoted (x1 x2 _ _ _xq), where xk is an item. For brevity, the brackets are omitted if an element has only one item, that is, element (x) is written as x. Suppose that a customer made several shopping trips to the store. These ordered events form a sequence for the customer. That is, the customer first bought the items in s1, then later bought the items in s2, and so on. An item can occur at most once in an event of a sequence, but can occur multiple times in different events of a sequence. The number of instances of items in a sequence is called the length of the sequence. A sequence with length l is called an l-sequence. A sequence a = ha1a2 _ _ _ani is called a subsequence of another sequence b = hb1b2 _ _ _bmi, and b is a supersequence of a, denoted as a v b, if there exist integers 1 _ j1 < j2 < _ _ _ < jn _ m such that a1 _ bj1 , a2 _ bj2 , . . . , an _ bjn .

For example, if a = h(ab), di and b = h(abc), (de)i, where a, b, c, d, and e are items, then a is a subsequence of b and b is a super sequence of a. A sequence database, S, is a set of tuples, hSID, si, where SID is a sequence ID and s is a sequence. For our example, S contains sequences for all customers of the store. A tuple hSID, si is said to contain a sequence a, if a is a subsequence of s. The support of a sequence a in a sequence database S is the number of tuples in the database containing a, that is, supportS(a) = j fhSID, sij(hSID, si 2 S) ^ (a v s) g j. It can be denoted as support(a) if the sequence database is clear from the context. Given a positive integer min sup as the minimum support threshold, a sequence a is frequent in sequence database S if supportS(a)_min sup. That is, for sequence a to be frequent, itmust occur at least min sup times in S. A frequent sequence is called a sequential pattern. A sequential pattern with length l is called an l-pattern.

A sequence dstabase

Sequence – ID	Sequence
1	⟨a(abc)(ac)d(cf)⟩
2	⟨(ad)c(bc)(ae)⟩
3	⟨(ef)(ab)(df)cb⟩
4	⟨eg(af)abc⟩ |

Scalable Methods for Mining Sequential Patterns

Sequential pattern mining is computationally challenging because such mining may generate and/ or test a combinatorially explosive number of intermediate subsequences. "How can we develop efficient and scalable methods for sequential pattern mining?" Recent developments have made progress in two directions: (1) efficient methods for mining the full set of sequential patterns, and (2) efficient methods for mining only the set of closed sequential patterns, where a sequential pattern s is closed if there exists no sequential pattern s0 where s0 is a proper supersequence of s, and s0 has the same (frequency) support as s.6 Because all of the subsequences of a frequent sequence are also frequent, mining the set of closed sequential patterns may avoid the generation of unnecessary subsequences and thus lead to more compact results as well as more efficient methods than mining the full set. We will first examine methods for mining the full set and then study how they can be extended for mining the closed set. In addition, we discuss modifications for mining multilevel, multidimensional sequential patterns (i.e., with multiple levels of granularity). The major approaches for mining the full set of sequential patterns are similar to those introduced for frequent itemset mining in Chapter 5. Here, we discuss three such approaches for sequential pattern mining, represented by the algorithms GSP, SPADE, and PrefixSpan, respectively. GSP adopts a candidate generate-and-test approach using horizonal data format (where the data are represented as hsequence ID : sequence of itemsetsi, as usual, where each itemset is an event). SPADE adopts a candidate generateand - test approach using vertical data format (where the data are represented as **hitemset :** (sequence ID, event ID)i). The vertical data format can be obtained by transforming from a horizontally formatted sequence database in just one scan. PrefixSpan is a pattern growth method, which does not require candidate generation.

4.11.1 Web Usage Mining

Q12. Briefly Explain the Web Bases Mining?

Ans :

Web usage mining is the third category in web mining. This type of web mining allows for the collection of Web access information for Web pages. This usage data provides the paths leading to accessed Web pages. This information is often gathered automatically into access logs via the Web server. CGI scripts offer other useful information such as referrer logs, user subscription information and survey logs. This category is important to the overall use of data mining for companies and their internet/ intranet based applications and information access.

Usage mining allows companies to produce productive information pertaining to the future of their business function ability. Some of this information can be derived from the collective information of lifetime user value, product cross marketing strategies and promotional campaign effectiveness. The usage data that is gathered provides the companies with the ability to produce results more effective to their businesses and increasing of sales. Usage data can also be useful for developing marketing skills that will out-sell the competitors and promote the company's services or product on a higher level.

Usage mining is valuable not only to businesses using online marketing, but also to e-businesses whose business is based solely on the traffic provided through search engines. The use of this type of web mining helps to gather the important information from customers visiting the site. This enables an in-depth log to complete analysis of a company's productivity flow. E-businesses depend on this information to direct the company to the most effective Web server for promotion of their product or service.

This web mining also enables Web based businesses to provide the best access routes to services or other advertisements. When a company advertises for services provided by other companies, the usage mining data allows for the most effective access paths to these portals. In addition, there are typically three main uses for mining in this fashion.

The first is usage processing, used to complete pattern discovery. This first use is also the most difficult because only bits of information like IP addresses, user information, and site clicks are available. With this minimal amount of information available, it is harder to track the user through a site, being that it does not follow the user throughout the pages of the site.

The second use is content processing, consisting of the conversion of Web information like text, images, scripts and others into useful forms. This helps with the clustering and categorization of Web page information based on the titles, specific content and images available.

Finally, the third use is structure processing. This consists of analysis of the structure of each page contained in a Web site. This structure process can prove to be difficult if resulting in a new structure having to be performed for each page.

Analysis of this usage data will provide the companies with the information needed to provide an effective presence to their customers. This collection of information may include user registration, access logs and information leading to better Web site structure, proving to be most valuable to company online marketing. These present some of the benefits for external marketing of the company's products, services and overall management.

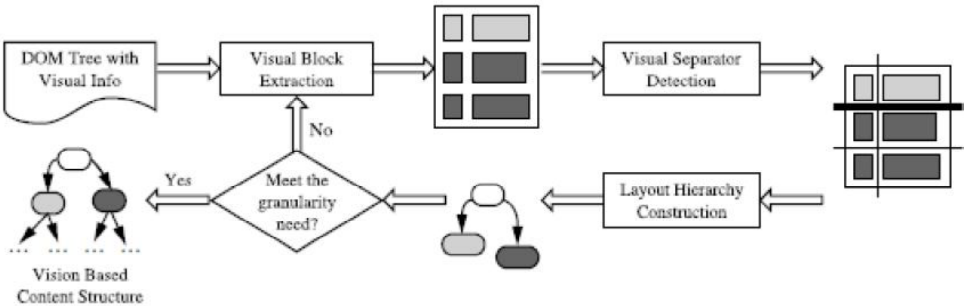
Internally, usage mining effectively provides information to improvement of communication through intranet communications. Developing strategies through this type of mining will allow for intranet based company databases to be more effective through the provision of easier access paths. The projection of these paths helps to log the user registration information giving commonly used paths the forefront to its access.

Therefore, it is easily determined that usage mining has valuable uses to the marketing of businesses and a direct impact to the success of their promotional strategies and internet traffic. This information is gathered on a daily basis and continues to be analyzed consistently. Analysis of this pertinent information will help companies to develop promotions that are more effective, internet accessibility, inter-company communication and structure, and productive marketing skills through web usage mining.

Mining the Web's Link Structures to Identify Authoritative Web Pages

But how can a search engine automatically identify authoritative Web pages for my topic?" Interestingly, the secrecy of authority is hiding in Web page linkages. The Web consists not only of pages, but also of hyperlinks pointing from one page to another. This idea has motivated some interesting studies on mining authoritative pages on the Web These properties of Web link structures have led researchers to consider another important category of Web pages called a hub. A hub is one or a set of Web pages that provides collections of links to authorities. Hub pages may not be prominent, or there may exist few links pointing to them.

An algorithm using hubs, called HITS (Hyperlink-Induced Topic Search), was developed as follows. First, HITS uses the query terms to collect a starting set of, say, 200 pages from an index-based search engine. These pages form the root set. Since many of these pages are presumably relevant to the search topic, some of them should contain links to most of the prominent authorities. Therefore, the root set can be expanded into a base set by including all of the pages that the root-set pages link to and all of the pages that link to a page in the root set, up to a designated size cutoff such as 1,000 to 5,000 pages.


DOM Tree with Visual Info
Visual Block Extraction
Visual Separator Detection
No
Yes
Meet the granularity need?
Layout Hierarchy Construction
Vision Based Content Structure

We first associate a non-negative authorityweight, ap, and a non-negative hubweight, hp, with each page p in the base set, and initialize all a and h values to a uniform constant

$$a_p = \sum (q \text{ such that } q \rightarrow p)\ h_q$$

$$h_p = \sum (q \text{ such that } q \rightarrow p)\ a_q$$

These equations can be written in matrix form as follows. Let us number the pages f1;2; : : : ; ng and define their adjacency matrix A to be an n_n matrix where A(i; j) is 1 if page i links to page j, or 0 otherwise. Similarly, we define the authority weight vector a = (a1; a2; : : : ; an), and the hub weight vector h = (h1; h2; : : : ;hn). Thus, we have

$$h = A \,.\, a$$

$$a = A^T \,.\, h,$$

where AT is the transposition of matrix A. Unfolding these two equations k times, we have

$$h = A \,.\, a = AA^T h = (AA^T)h = (AA^T)^2 h = ... = (AA^T)^k h$$

$$a = A^T.h = A^T Aa = (A^T A)\, a = (A^T A)^2 a = ... = (A^T A)^k a.$$

4.11.2 Effective Methods and Techniques in Text Mining

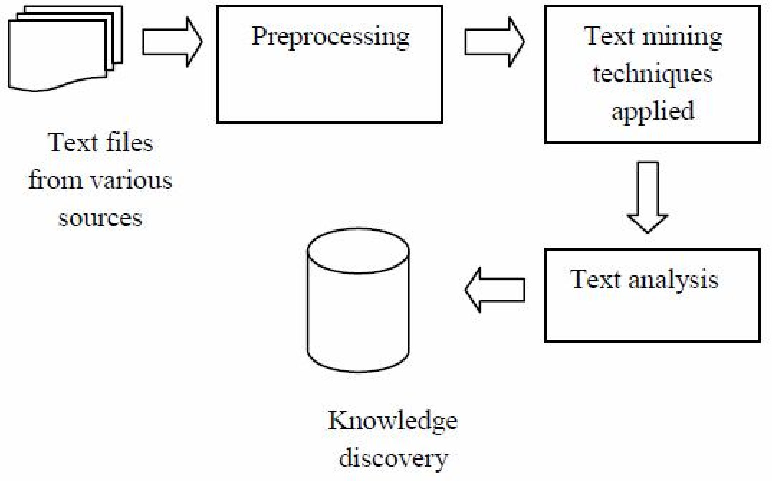
Q13. What are different and effective methods and approaches used in Data Mining ?

Ans :

To extract useful information and association from massive text data some text data mining approaches are available. Data mining isused where analyzing data to find rules and patterns depict the characteristic of the data. The term "Data Mining" also known as Knowledge Discovery in Databases (KDD) is formally defined as: "the relevant extraction of fixed, previously undiscovered, and useful information from massive chunk of data" The „mined information is represented as a well-formed structure of the dataset, where the structure perhaps used on new data for prediction or classification. Roughly, data mining works on structured data, while text works on special characteristics and is unstructured. An unstructured data is wholly different from databases, where mining techniques are usually applied to manage structured data. Text mining works with unstructured or semi-structured data sets.

Data mining techniques have progressively been studied, especially in the real-world databases. The goal of a data mining approach might be e.g. to allow a corporation either to improve marketing, sales, and customer support operations or to identify a fraudulent customer through better understanding of its customers. Data mining techniques are utilized in many fields such as marketing, manufacturing, process control, fraud detection, bioinformatics, information retrieval, adaptive hypermedia, electronic commerce and network management.

Text mining technique initially collect document from numerous resources. Text mining application retrieves a document, pre-processes it through checking format and character sets. Then the document insist to go through a next stage i.e. text analysis phase. Text analysis is semantic analysis to collect high quality information from text. There are many text analysis techniques available depending on objective of organization combinations of techniques could be used. Frequently text analysis techniques are repeated until information is extracted. The processed information can be placed in system called management information, yielding sufficient amount of knowledge for the user of that system.


Preprocessing
Text mining techniques applied
Text files from various sources
Text analysis
Knowledge discovery

Methods Used in Text Mining

There are so various techniques developed for solving problems of text mining those are nothing but relevant information retrieval according to user's requirement. Based on information retrieval techniques there are some methods explained below.

Term Based Method

Term in document is word having well-formed meaning. In term based method document is scrutinized on the basis of term and has benefits of productive computational performance as well as well understood theories for term weighting. These techniques are developed over few decades from the information retrieval and machine learning association. This method has disadvantages such as polysemy and synonymy. Polysemy means a word have multiple meanings and synonymy is multiple words having the same meaning. The allowable meaning of many discovered terms is ambiguous for answering what users want. Information retrieval approach provides many term based methods to solve raised challenge.

Phrase Based Method

Phrase gives more semantics like information and is uncertain. In this, document is estimated on phrase basis as phrases are less doubtful and more selective than individual terms. Some reasons which deter the performance:

- ➢ Due to secondary analytical properties to terms
- ➢ Less occurrence
- ➢ Massive duplicate and noisy phrases

Concept Based Method

In this method, terms are estimated on sentence and document level. Text Mining techniques are often based on analytical analysis of word or phrase. The term analytical analysis captures the importance of word without any document. Two terms might have same frequency in same document, but one term might contribute more appropriate meaning. A novel concept based mining is introduced to acquire the semantics of texts. This model contains three components. The first component evaluates semantic arrangement of sentences. The second component evaluates a conceptual ontological graph (COG) which describes semantic structures and the final component extracts top concepts based on the first two components to build feature vectors by using the standard vector space model. This model has ability to separate unnecessary terms and meaningful terms which describe a meaningful sentence. It is sometime depends upon natural language processing methods. A special aspect selection is enforced on the query concepts to strengthen the representation and remove noise and ambiguity.

Pattern Taxonomy Method

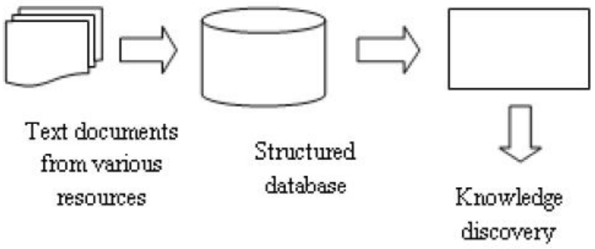
In pattern taxonomy, documents are evaluated on pattern basis. Patterns are constructed in taxonomy by applying is-a relation. From many years, pattern mining is been reviewed in data mining. Patterns can be detected by using data mining techniques like association rule; frequent item set mining, sequential and closed pattern mining. Use of detected knowledge in the field of text mining is very crucial and inefficient, because some useful long patterns with high selectivity lack in support. It is not always said that all short patterns are useful hence known as Mis constructions of patterns and it lead to the ineffective performance. An efficient pattern discovery procedure has been recommended to overcome low-frequency and misconstruction problems for text mining. The pattern related method uses two mechanism pattern deploying and pattern evolving. This technique refines the discovered patterns. The pattern based model performs better than any other pure data mining-based methods.

Techniques Used in Text Mining

To tell computers how to evaluate, understand and generate text, technologies are being produced by natural language processing. The technologies like information extraction, summarization, categorization, and clustering and information visualization are used in the text mining process. In the following sections each of these technologies and the role that they play in text mining are discussed. The types of situations where each technology may be useful in order to help users are also discussed.

Information extraction

Information extraction is the first step for computer to evaluate unstructured text by identifying key phrases and accordance within text. For this, pattern matching is used to look for predefined sequences in text. Information extraction includes tokenization, identification of named entities, sentence segmentation, and part-of-speech assignment. Initially phrases and sentences are parsed and semantically interpreted then required pieces of information are stored in database. This technology can be very useful when dealing with large amount of text. For many applications most challenging is electronic information which is in the form of free natural language documents rather than structured databases such as relational databases. Information extraction is a solution for this problem of reconstructing a collection of textual documents into a more structured database. For more mining of knowledge database which is built by an information extraction method can be then submitted to the KDD module.



In the accepted an draw itemsets, the transaction based algorithm discards individual items, from the individual item set depending on the system workload, i.e., if the system can allow 10 items from an itemset, whichisof12items,the transaction based would remove the last 2items from the itemset. These removed items from the item sets are then back to the buffer for next processing cycle.

Categorization

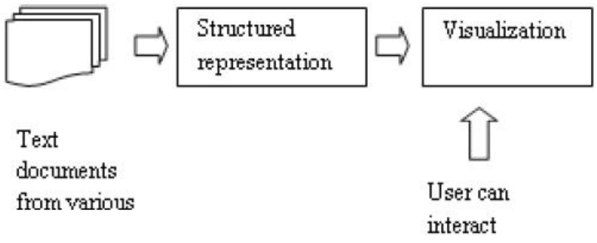
Categorization assigns one or more category to independent text document. Categorization is a supervised learning method because it is based on input output examples to segregate new documents. Predefined classes are being assigned to the text documents based on their content. This process consists of pre-processing, indexing, dimensionally reduction, and classification. The objective is to train classifier using known examples and then unknown examples are categorized automatically. Analytical classification such as Naïve Bayesian classifier, Nearest Neighbor classifier, Decision Tree, and Support Vector Machines are useful to categorize text.

Clustering

Clustering method is useful to find groups of documents with similar content. As a result of clustering a partition called clusters are generated and each cluster holdsa number of documents. The contents of the documents in single cluster are much similar and the contents of different clusters are dissimilar then the quality of clustering is considered better. Clustering technique is useful to group similar documents which it differs from categorization because in clustering documents are clustered on the fly instead of use of predefined topics. K-means is often used clustering algorithm in data mining; in text mining field also it obtains good results. A basic clustering algorithm maintains a track of topics for every document and calculates the weightage of how well the document fits into each cluster. The management information systems uses clustering technology as organizational database contain thousands of documents.

Visualization

Visualization can enhance and clarify the discovery of relevant information. For discriminating individual documents or chunks of documents text flags are used to show the category of document and to show density colors are used. Visual text mining collects huge textual sources in a visual hierarchy. The user can use the document by zooming and scaling. Information visualization is useful to government to classify terrorist networks or to find information about crimes. Following fig.3 shows steps involved in visualization process.



Summarization

Main objective of text summarization is to reduce the length and details of a document while retaining most important points and general meaning. Text summarization is helpful for resolving whether or not a lengthy document fulfills the user's needs and whether it is worth reading for further information hence summary can be replaced by the set of documents. When user reads the first paragraph, text summarization software processes and summarizes the large text document in minimal time as compared to user.

Even though computers are able to identify people, places, and time it is difficult to teach software to analyze semantics and to interpret meaning of text document. Humans first read entire text section to summarize it and then try to develop a full understanding. Then finally they Make highlights to show main points. Steps in summarization process are as follows:

1. Structured representation of the original text is a pre-processing step.
2. Algorithm is applied to translate summary structure from text structure in next processing step.
3. In the development step the final summary is retrieved from the summary structure.

4.11.3 Audio and video Data Mining

Q14. Write in brief about the Visual and Audio data mining Techniques ?

Ans :

Video databases are widespread and video data sets are extremely large. There are tools for managing and searching within such collections, but the need for tools to extract the hidden and useful knowledge embedded within the video data is becoming critical for many decision-making applications.

Video Processing

Though the acquisition and storage of video data is an easy task the retrieval of information from the video data is a challenging task. One of the most important steps involved is to transform the video data from non-structured data into a structured data set as the processing of the video data with image processing or computer vision techniques demands structured-format features. Before applying the data-mining techniques on the video key frame, the video, audio and text features are extracted using the image processing techniques, eliminating the digitalization noise and illumination changes to avoid false positive detection.

It is the most fundamental task in video processing to partition the long video sequences into a number of shots and find a key frame of each shot for further video information retrieval tasks.

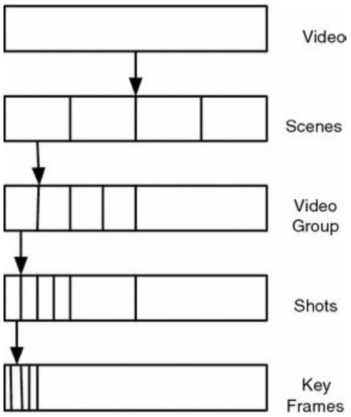
Video data model

Since, the relational or object oriented data model does not provide enough facilities for managing and retrieving the video contents efficiently, an appropriate data model is needed to do it. Three main reasons can be identified for this:

1. Lack of facilities for the management of spatiotemporal relations
2. Lack of knowledge-based methods for interpreting raw data into semantic contents and
3. Lack of query representations for complex structures.

Video segmentation

The first step in any video data management system is invariably, the segmentation of the video track into smaller units enabling the subsequent processing operations on video shots, such as video indexing, semantic representation or tracking of the selected video information and identifying the frames where a transition takes place from one shot to another. The visual-based segmentation identifies the shot boundaries and the motion-based segmentation identifies pans and zooms.



- **Video group:** It is an intermediate entity between the physical shots and semantic scenes serving as the bridge between the shot and scene.
- **Shot:** It is defined as a sequence of frames taken by a single camera with no major changes in the visual content.
- **Key frame:** The frame represents the salient visual contents of a shot. Since a large number of adjacent frames are likely to be similar, one or more key frames can be extracted from the shot depending on the complexity of the content of the shot.

Feature extraction

The video segmented and the key frames chosen, the low-level image features can be extracted from the key frames. The low-level visual features such as color, texture, edge and shapes can be extracted and represented as feature descriptors. A feature is defined as a descriptive parameter extracted from an image or a video stream]. There are two kinds of video features extracted from the video.

(i) The description based features that use metadata, such as keywords, caption, size and time of the creation.

(ii) The content based features are based on the content of the object itself. There are two categories of content based features: the global features extracted from a whole image and the local or regional features describing the chosen patches of a given image. Each region is then processed to extract a set of features characterizing the visual properties including the color, texture, motion and structure of the region. The shot-based features and the object-based features are the two approaches used to access the video sources in the database.

Video data mining

It is video data mining that deals with the extraction of implicit knowledge, video data relationships, or other patterns not explicitly stored in the video databases considered as an extension of still image mining by including mining of temporal image sequences. It is a process which not only automatically extracts content and structure of video, features of moving objects, spatial or temporal correlations of those features, but also discovers patterns of video structure, object activities, video events from vast amounts of video data with a little assumption of their contents.

Video information retrieval versus video data mining

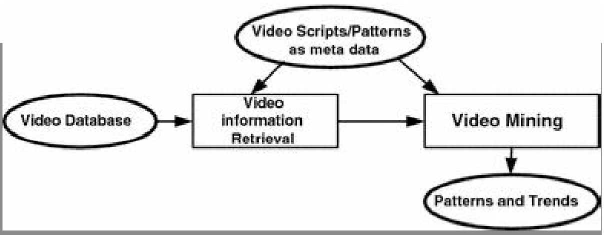
It is video information retrieval, not information retrieval that is considered as part of the video data mining of a certain level knowledge discovery such as feature selection, dimensionality reduction and concept discovery.

The dissimilarities of video data mining with related areas are as follows,

- **Video data mining versus computer vision or video processing:** The relationship between video processing and video mining is very subjective. The goal of video data mining is to extract patterns from the video sequences whereas video processing focuses on understanding and/or extracting features from the video database.

- **Video data mining versus pattern recognition:** Both areas share the feature extraction steps but differ in pattern specificity. The objective of pattern recognition is to recognize specific, classification patterns, pattern generation and analysis. Pattern recognition is indulging in a research on classifying special samples with an existing model while video mining is involved in discovering rules and patterns of samples with or without image processing. The objective of video data mining is to generate all significant patterns without prior knowledge of what they are.

- **Video information retrieval versus video data mining:** The difference is similar to the difference between database management and data mining.


Video Scripts/Patterns as meta data
Video Database
Video information Retrieval
Video Mining
Patterns and Trends

Key Problems in Video Data Mining

Video data mining is an emerging field that can be defined as the unsupervised discovery of patterns in audio visual contents. Mining video data is even more complicated than mining still image data requiring tools for discovering relationships between objects or segments within the video components, such as classifying video images based on their contents, extracting patterns in sound, categorizing speech and music, and recognizing and tracking objects in video streams. The existing data-mining tools pose various problems while applied to video database. They are:

- Database model problem in which video documents are generally unstructured in semantics and cannot be represented easily via the relational data model demanding a good video database model that is crucial to support more efficient video database management and mining. To adopt a good model, one needs to address three problems, namely
  - How many levels should be included in the model?
  - What kind of decision rules should be used at each node?

- o Do these nodes make sense to human beings?

- ➢ The retrieval results solely based on the low level feature extraction are mostly unsatisfactory and unpredictable. It is the semantic gap between the low level visual features and the high level user domain that happens to the one of the hurdles for the development of a video data-mining system. Modeling the high level features rather than the low level features is difficult as the former is depending on the semantics whereas the latter is based on the syntactic structure.

- ➢ Maintaining data integrity and security in video database management structure. These challenges have led to a lot of research and development in the area of video data mining. The main objective of video mining is to extract the significant objects, characters and scenes by determining their frequency of re-occurrence.

Audio data mining

Audio is what plays a significant role in the detection and recognition of events in video. Supplying speech, music and various special sounds and can be used to separate different speeches, detect various audio events, analyze for spoken text, emotions, high-light detection in sports videos and so on. In movies, the audio is often correlated with the scene. For instance, the shots of fighting and explosions are mostly accompanied by a sudden change in the audio level.

In addition, audio features can be used to characterize the media signals to discriminate between music and speech classes. In general, the audio features can be categorized into two groups, namely, time domain features which include zero-crossing rates, amplitudes, pitches and the frequency domain features consisting of spectrograms, cepstral coefficients and mel-frequency cepstral coefficients. There are two main approaches to audio data mining. Firstly, Text-based indexing approach converts speech to text and then identifies words in a dictionary having several hundred thousand entries. If a word or name is not in the dictionary, the Large Vocabulary Continuous Speech Recognizers system will choose the most similar word it can find. Secondly, phoneme-based indexing approach analyzes and identifies sounds in a piece of audio content to create a phonetic-based index. It then uses a dictionary of several dozen phonemes to convert a user's search term to the correct phoneme string. Finally, the system looks for the search terms in the index.

4.12 The Scope of Datamining

Q15. Explain the importance and Scope of Data Mining ?

Ans :

Data mining derives its name from the similarities between searching for valuable business information in a large database. Data mining technology can generate new business opportunities by providing these capabilities.

1. **Automated Prediction of Trends and Behaviors:** Data mining automates the process of finding predictive information in large databases. Other predictive problems include forecasting bankruptcy and other forms of default, and identifying segments of a population likely to respond similarly to given events.

2. **Automated Discovery of Previously unknown Patterns:** Data mining tools sweep through databases and identify previously hidden patterns in one step. An example of pattern discovery is the analysis of retail sales data to identify seemingly unrelated products that are often purchased together. Other pattern discovery problems include detecting fraudulent credit card transactions and identifying anomalous data that could represent data entry keying errors. The most commonly used techniques in data mining are artificial neural networks: Non-linear predictive models that learn through training and resemble biological neural networks in structure.

   - ➢ **Decision trees:** Tree-shaped structures that represent sets of decisions. These decisions generate rules for the classification of a dataset. Specific decision tree methods include

Classification and Regression Trees (CART) and Chi Square Automatic Interaction Detection (CHAID).

- **Genetic algorithms:** Optimization techniques that use process such as genetic combination, mutation, and natural selection in a design based on the concepts of evolution.
- **Nearest neighbour method:** A technique that classifies each record in a dataset based on a combination of the classes of the k record(s) most similar to it in a historical dataset (where k . Sometimes called the k-nearest neighbour technique. Rule induction: The extraction of useful if-then rules from data based on statistical significance.

Data Mining - Applications & Trends

Data mining is widely used in diverse areas. There are a number of commercial data mining system available today and yet there are many challenges in this field. In this tutorial, we will discuss the applications and the trend of data mining.

Data Mining Applications

Here is the list of areas where data mining is widely used

- Financial Data Analysis
- Retail Industry
- Telecommunication Industry
- Biological Data Analysis
- Other Scientific Applications
- Intrusion Detection

Financial Data Analysis

The financial data in banking and financial industry is generally reliable and of high quality which facilitates systematic data analysis and data mining. Some of the typical cases are as follows “

- Design and construction of data warehouses for multidimensional data analysis and data mining.
- Loan payment prediction and customer credit policy analysis.
- Classification and clustering of customers for targeted marketing.
- Detection of money laundering and other financial crimes.

Retail Industry

Data Mining has its great application in Retail Industry because it collects large amount of data from on sales, customer purchasing history, goods transportation, consumption and services. It is natural that the quantity of data collected will continue to expand rapidly because of the increasing ease, availability and popularity of the web.

Data mining in retail industry helps in identifying customer buying patterns and trends that lead to improved quality of customer service and good customer retention and satisfaction. Here is the list of examples of data mining in the retail industry -

- Design and Construction of data warehouses based on the benefits of data mining.
- Multidimensional analysis of sales, customers, products, time and region.
- Analysis of effectiveness of sales campaigns.
- Customer Retention.
- Product recommendation and cross-referencing of items.

Telecommunication Industry

Today the telecommunication industry is one of the most emerging industries providing various services such as fax, pager, cellular phone, internet messenger, images, e-mail, web data transmission, etc. Due to the development of new computer and communication technologies, the telecommunication industry is rapidly expanding. This is the reason why data mining is become very important to help and understand the business.

Data mining in telecommunication industry helps in identifying the telecommunication patterns, catch fraudulent activities, make better use of resource, and improve quality of service. Here is the list of examples for which data mining improves telecommunication services -

- Multidimensional Analysis of Telecommunication data.
- Fraudulent pattern analysis.
- Identification of unusual patterns.
- Multidimensional association and sequential patterns analysis.
- Mobile Telecommunication services.
- Use of visualization tools in telecommunication data analysis.

Biological Data Analysis

In recent times, we have seen a tremendous growth in the field of biology such as genomics, proteomics, functional Genomics and biomedical research. Biological data mining is a very important part of Bioinformatics. Following are the aspects in which data mining contributes for biological data analysis -

- Semantic integration of heterogeneous, distributed genomic and proteomic databases.
- Alignment, indexing, similarity search and comparative analysis multiple nucleotide sequences.
- Discovery of structural patterns and analysis of genetic networks and protein pathways.
- Association and path analysis.
- Visualization tools in genetic data analysis.

Other Scientific Applications

The applications discussed above tend to handle relatively small and homogeneous data sets for which the statistical techniques are appropriate. Huge amount of data have been collected from scientific domains such as geosciences, astronomy, etc. A large amount of data sets is being generated because of the fast numerical simulations in various fields such as climate and ecosystem modeling, chemical engineering, fluid dynamics, etc. Following are the applications of data mining in the field of Scientific Applications -

- Data Warehouses and data preprocessing.
- Graph-based mining.
- Visualization and domain specific knowledge.

Intrusion Detection

Intrusion refers to any kind of action that threatens integrity, confidentiality, or the availability of network resources. In this world of connectivity, security has become the major issue. With increased usage of internet and availability of the tools and tricks for intruding and attacking network prompted intrusion detection to become a critical component of network administration. Here is the list of areas in which data mining technology may be applied for intrusion detection -

- Development of data mining algorithm for intrusion detection.
- Association and correlation analysis, aggregation to help select and build discriminating attributes.
- Analysis of Stream data.
- Distributed data mining.
- Visualization and query tools.

Data Mining System Products

There are many data mining system products and domain specific data mining applications. The new data mining systems and applications are being added to the previous systems. Also, efforts are being made to standardize data mining languages.

Choosing a Data Mining System

The selection of a data mining system depends on the following features -

- **Data Types :** The data mining system may handle formatted text, record-based data, and relational data. The data could also be in ASCII text, relational database data or data warehouse data. Therefore, we should check what exact format the data mining system can handle.

- **System Issues :** We must consider the compatibility of a data mining system with different operating systems. One data mining system may run on only one operating system or on several. There are also data mining systems that provide web-based user interfaces and allow XML data as input.
- **Data Sources :** Data sources refer to the data formats in which data mining system will operate. Some data mining system may work only on ASCII text files while others on multiple relational sources. Data mining system should also support ODBC connections or OLE DB for ODBC connections.
- **Data Mining functions and methodologies :** There are some data mining systems that provide only one data mining function such as classification while some provides multiple data mining functions such as concept description, discovery-driven OLAP analysis, association mining, linkage analysis, statistical analysis, classification, prediction, clustering, outlier analysis, similarity search, etc.
- **Coupling data mining with databases or data warehouse systems :** Data mining systems need to be coupled with a database or a data warehouse system. The coupled components are integrated into a uniform information processing environment. Here are the types of coupling listed below "
  - No coupling
  - Loose Coupling
  - Semi tight Coupling
  - Tight Coupling
- **Scalability :** There are two scalability issues in data mining "
  - **Row (Database size) Scalability :** A data mining system is considered as row scalable when the number or rows are enlarged 10 times. It takes no more than 10 times to execute a query.
  - **Column (Dimension) Salability :** A data mining system is considered as column scalable if the mining query execution time increases linearly with the number of columns.
- **Visualization Tools** - Visualization in data mining can be categorized as follows :
  - Data Visualization
  - Mining Results Visualization
  - Mining process visualization
  - Visual data mining
- **Data Mining query language and graphical user interface :** An easy-to-use graphical user interface is important to promote user-guided, interactive data mining. Unlike relational database systems, data mining systems do not share underlying data mining query language.

Trends in Data Mining

Data mining concepts are still evolving and here are the latest trends that we get to see in this field -

- Application Exploration.
- Scalable and interactive data mining methods.
- Integration of data mining with database systems, data warehouse systems and web database systems.
- Standardization of data mining query language.
- Visual data mining.
- New methods for mining complex types of data.
- Biological data mining.
- Data mining and software engineering.
- Web mining.
- Distributed data mining.
- Real time data mining.
- Multi database data mining.
- Privacy protection and information security in data mining.

4.13 Trends In Data Mining

Q16. Explain the past, Present and Future Trend in Data Mining?

Ans :

- **Historical Trends:** Data mining application era was perceived in early 1980s principally focused on single tasks driven by research tools. Data mining is helpful in various disciplines like Data Base Management Systems (DBMS), Artificial Intelligence (AI), Machine Learning (ML) and Statistics. Historical trends of data mining are explained as follows [4]: Data mining algorithm work best with the numerical data especially collected from a single data base and various data mining techniques have developed for flat files, traditional and relational database where the data is mostly represented in the tabular form. Afterwards, with the convergence of Statistics and Machine Learning pave way to the evolution of various algorithms to mine the non-numerical data and relational data bases. Development in fourth generation programming language influenced much in the field of data mining and various related computing techniques. Initially, most of the algorithms engaged to work only on statistical techniques. Various computing techniques such as AI, ML and pattern reorganization evolved to do the data mining tasks in ease manner. Various data mining techniques like Induction, Compression, approximation and other algorithms developed to mine the large volume of heterogeneous data stored in the data warehouse.

- **Current Trends:** Advancement in data mining with various integrations and implications of the methods and techniques have formed the present data

- **Future Trends:** Data mining has been acquiring noteworthy amount of importance in recent years and it has a strong industrial impact. Future of data mining companies would be promising in the coming years based on this observation. A huge amount of data gets agitate in the research, medical, corporate and media industries as it becomes great for anybody involves in gathering useful information. Increasing technology and future application areas always creates new challenges and opportunities for data mining. Advance data mining techniques can be developed and used by R& D and other information rich companies to discover useful patterns that can help in research or business development to ensure the growth and development of the companies. Future data mining technologies involve standardization of data mining languages; predictive analysis, advanced text mining, Semantic and image mining are discussed as follows:

  o **Standardization of Data Mining Languages:** Different syntaxes are used in various data mining tools, hence standardized syntaxes needs to be developed in order to make convenient coding for the users. Standardization of interaction language and flexible user interaction has to be much concentrated by the data mining applications.

  o **Predictive Analysis:** In earlier days of data mining whereby assumptions about structure of data were unheard where as now a days, data is put through algorithms based on certain attributes such as trends, relations and patterns and predictions are thereby projected. This paves way for significant increase in decision making capabilities especially in business process. For instance, predicting customer behaviors with the help of mathematical modeling and statistical analysis, their spending habits on their credit cards can be determined and credit point allotted accordingly. This kind of predictive analysis can create huge impact in the near future and business can propagate in well manner based on such predictions.

  o **Advanced Text Mining:** In earlier times, text mining was only performed on structured data. But, majority of unstructured data are available in the form of memos, emails, surveys, notes, chats, whitepapers, forums, presentation, etc. It can be tapped and accessed using data mining services. Vast amount of information can be gathered using such text mining techniques and this can be used effectively for the business purpose. This is taking data mining a step further from earlier times.

- **Semantic and Image Mining:** Semantic and image mining will take a predominant stage in future as researchers will be able to find hidden meaning in data and document using artificial intelligence and structural analysis software. Images can be searched for identifying patterns and the information derived can be used for various scientific and business advancements. Plenty of opportunities will be opened through the data mining services offered by various professional data mining companies.

Comparison Fast, Current and Future Trends in Data mining	**Algorithms/ Techniques Employed**	**Data Formats**	**Computing Resources**
Past	Statistical, Machine Learning Techniques	Numerical data and structured data stored in traditional databases	Evolution of 4G PL and various related techniques
Current	Statistical, Machine Learning, Artificial Intelligence, Pattern Reorganization Techniques	Heterogeneous data formats includes structured, semi structured and unstructured data	High speed networks, High end storage devices and Parallel, Distributed computing etc…
Future	Soft Computing techniques like Fuzzy logic, Neural Networks and Genetic Programming	Complex data objects includes high dimensional, high speed data streams, sequence, noise in the time series, graph, Multi instance objects, Multi represented objects and temporal data etc…	Multi-agent technologies and Cloud

Challenges of Data Mining

There are many challenges faced by the data mining and these challenges of data mining are pointed as follows:

- Scalability
- Complex and Heterogeneous Data
- Network Setting
- Data Quality
- Data Ownership and Distribution
- Dimensionality
- Privacy preservation
- Streaming Data

Short Answers

1. Define Outlier Detection

Outlier detection is the process of detecting and subsequently excluding outliers from a given set of data. An outlier may be defined as a piece of data or observation that deviates drastically from the given norm or average of the data set. An outlier may be caused simply by chance, but it may also indicate measurement error or that the given data set has a heavy-tailed distribution.

Here is a simple scenario in outlier detection, a measurement process consistently produces readouts between 1 and 10, but in some rare cases we get measurements of greater than 20. These rare measurements beyond the norm are called outliers since they "lie outside" the normal distribution curve.

There is really no standardized and rigid mathematical method for determining an outlier because it really varies depending on the set or data population, so its determination and detection ultimately becomes subjective. Through continuous sampling in a given data field, characteristics of an outlier may be established to make detection easier.

2. Define the following

(a) Space-based Approach

(b) Graph-based Approach

(a) Space-based Approach

Space-based outliers use Euclidean distances to define spatial neighborhoods. Kou et al. developed spatial weighted outlier detection algorithms which use properties such as center distance and common border length as weight when comparing non -spatial attributes. Adamet al. proposed an algorithm which considers both spatial relationship and semantic relationship among neighbors. Liu et al. proposed a method for detecting outliers in an irregularly- distributed spatial data set.

(b) Graph-based Approach

Graph-based Approach uses graph connectivity to define spatial neighborhoods. Yufeng Kou et al. proposed a set of graph-based algorithms to identify spatial outliers, which first constructs a graph based on k - nearest neighbor relationship in spatial domain, assigns the non-spatial attribute differences as edge weights, and continuously cuts high- weight edges to identify isolated points or regions that are much dissimilar to their neighboring objects. The algorithms have two major advantages compared with the existing spatial outlier detection methods: accurate in detecting point outliers and capable of identifying region outliers.

3. Automated Analysis

The inherent dangers of the necessity to rely on automatic strategies for analyzing the data, another main theme in DM, have been demonstrated again and again. There are many examples where trivial non- relevant variables, such as case number, turned out to be the best predictors in automated analysis. Similarly, variables displaying a major role in predicting a variable of interest in the past, may turn out to be useless because they reflect some strong phenomenon not expected to occur in the future. In spite of these warnings, it is clear that large parts of the analysis should be automated, especially at the warehousing stage of the DM.

This may raise new dangers. It is well known in statistics that having even a small proportion of outliers in the data can seriously distort its numerical summary. Such unreasonable values, deviating from the main structure of the data, can usually be identified by a careful human data analyst, and excluded from the analysis. But once we have to warehouse information about millions of customers, summarizing the information about each customer by a few numbers has to be automated and the analysis should rather deal automatically with the possible impact of a few outliers.

Statistical theory and methodology supply the framework and the tools for this endeavor. A numerical summary of the data that is not unboundedly influenced by a negligible proportion of the data is called a resistant summary. According to this definition the average is not re- sistant, for even one straying data value can have an unbounded effect on it. In contrast, the median is resistant. A resistant summary that retains its good properties under less than ideal situations is called a robust summary, the á-trimmed mean (rather than the median) being an example of such. The concepts of robustness and resistance, and the development of robust statistical tools for summarizing location, scale, and relationships, were developed during the 1970's and the 1980's.

4. Define Sampling

Sampling is the ultimate scalable statistical tool: if the number of cases n is very large the conclusions drawn from the sample depend only on the size of the sample and not on the size of the data set. It is often used to get a first impression of the data, visualize its main features, and reach decisions as to the strategy of analysis. In spite of its scalability and usefulness sampling has been attacked in the KDD community for its inability to find very rare yet extremely interesting pieces of knowledge.

Sampling is a very well developed area of statistics (see for example Cochran, 1977), but is usually used in DM at the very basic level. Strat- ified sampling, where the probability of picking a case changes from one stratum to another, is hardly ever used. But the questions are relevant even in the simplest settings: should we sample from the few positive re- sponses at the same rate that we sample from the negative ones? When studying faulty loans, should we sample larger loans at a higher rate? A thorough investigations of such questions, phrased in the realm of particular DM applications may prove to be very beneficial.

5. Proximity Analysis

Given a set of objects in feature space, a distance measure can be used to quantify the similarity between objects. Intuitively, objects that are far from others can be regarded as outliers. Proximity-based approaches assume that the proximity of an outlier object to its nearest neighbors significantly deviates from the proximity of the object to most of the other objects in the data set.

Proximity-based techniques define a data point as an outlier, if its locality (or proximity) is sparsely populated. The proximity of a data point may be defined in a variety of ways, which are subtly different from one another, The most common ways of defining proximity for outlier analysis are as follows:

- **Cluster-based:** The non-membership of a data point in any cluster, its distance from other clusters, and the size of the closest cluster, are used as criteria in order to compute the outlier score. The clustering problem has a complementary relationship to the outlier detection problem, in which points either belong to clustersor outliers.

- **Distance-based:** The distance of a data point to its k-nearest neighbor (or other variant) is used in order to define proximity. Data points with large k-nearest neighbor distances are defined as outliers. Distance-based algorithms typically perform the analysis at a much more detailed granularity than the other two methods. On the other hand, this greater granularity often comes at a significant computational cost.

- **Density-based:** The number of other points within a specified local region (grid region or distance-based region) of a data point, is used in order to define local density. These local density values may be converted into outlier scores. Other kernel-based methods or statistical methods for density estimation may also be used. The major difference between clustering and density-based methods is that clustering methods partition the data points, whereas density based methods partition the data space. Clearly, all these techniques are closely related, because they are based on some notion of proximity (or similarity). The major difference is at the detailed level of how this proximity is defined. These different ways of defining outliers may have different advantages and disadvantages, and this chapter will try to address these issues in a unified way. Furthermore, most of these methods generally work well when the data is highly clustered, and the outliers can be clearly distinguished from dense regions of the data. In many cases, the distinctions between these different classes of methods become blurred, when the definition of sparsity combines1 more than one of these concepts. One major difference between distance-based and the other two classes of methods is the level of granularity at which the analysis is performed.

6. Gini Index

The Gini coefficient or Index is a measure of resource inequality in a population developed by the Italian statistician Corrado Gini and published in his 1912 paper “Variabilità e mutabilità”. It can be used to measure any form of uneven distribution. Index varies form 0 to 1, zero means no inequality (no uncertainty) and 1 means maximum possible inequality (maximum uncertainty) The Gini coefficient is often calculated with the more practical Brown Formula

$$G = |1 - \sum_{k=1}^{n} (X_k - X_{k-1})(Y_k - Y_{k-1})|$$

Where, Gini coefficient and Xk: :cumulated portion of one variable for k= 0 to 1 with X0 = 0, Xn = 1.Yk: cumulated proportion of the target variable, for k = 0,...,n, with Y0 = 0, Yn = 1

7. Scalable Methods for Mining Sequential Patterns

Sequential pattern mining is computationally challenging because such mining may generate and/or test a combinatorially explosive number of intermediate subsequences. “How can we develop efficient and scalable methods for sequential pattern mining?” Recent developments have made progress in two directions: (1) efficient methods for mining the full set of sequential patterns, and (2) efficient methods for mining only the set of closed sequential patterns, where a sequential pattern s is closed if there exists no sequential pattern s0 where s0 is a proper super sequence of s, and s0 has the same (frequency) support as s.6 Because all of the subsequences of a frequent sequence are also frequent, mining the set of closed sequential patterns may avoid the generation of unnecessary subsequences and thus lead to more compact results as well as more efficient methods than mining the full set. We will first examine methods for mining the full set and then study how they can be extended for mining the closed set. In addition, we discuss modifications for mining multilevel, multidimensional sequential patterns (i.e., with multiple levels of granularity). The major approaches for mining the full set of sequential patterns are similar to those introduced for frequent itemset mining in Chapter 5. Here, we discuss three such approaches for sequential pattern mining, represented by the algorithms GSP, SPADE, and PrefixSpan, respectively. GSP adopts a candidate generate-and-test approach using horizonal data format (where the data are represented as hsequence ID : sequence of itemsetsi, as usual, where each itemset is an event). SPADE adopts a candidate generateand - test approach using vertical data format (where the data are represented as hitemset : (sequence ID, event ID)i). The vertical data format can be obtained by transforming from a horizontally formatted sequence database in just one scan. PrefixSpan is a pattern growth method, which does not require candidate generation.

8. Explain the Following

(a) Term based method

(b) Phrase Based Method

(c) Concept Based Method

(a) Term based method

Term in document is word having well-formed meaning. In term based method document is scrutinized on the basis of term and has benefits of productive computational performance as well as well understood theories for term weighting. These techniques are developed over few decades from the information retrieval and machine learning association. This method has disadvantages such as polysemy and synonymy. Polysemy means a word have multiple meanings and synonymy is multiple words having the same meaning. The allowable meaning of many discovered terms is ambiguous for answering what users want. Information retrieval approach provides many term based methods to solve raised challenge.

(b) Phrase Based Method

Phrase gives more semantics like information and is uncertain. In this, document is estimated on phrase basis as phrases are less doubtful and more selective than individual terms. Some reasons which deter the performance:

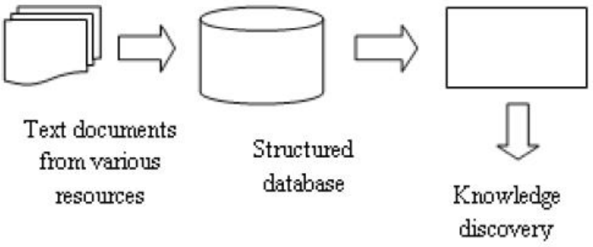
- Due to secondary analytical properties to terms
- Less occurrence
- Massive duplicate and noisy phrases

(c) Concept Based Method

In this method, terms are estimated on sentence and document level. Text Mining techniques are often based on analytical analysis of word or phrase. The term analytical analysis captures the importance of word without any document. Two terms might have same frequency in same document, but one term might contribute more appropriate meaning. A novel concept based mining is introduced to acquire the semantics of texts. This model contains three components. The first component evaluates semantic arrangement of sentences. The second component evaluates a conceptual ontological graph (COG) which describes semantic structures and the final component extracts top concepts based on the first two components to build feature vectors by using the standard vector space model. This model has ability to separate unnecessary terms and meaningful terms which describe a meaningful sentence. It is sometime depends upon natural language processing methods. A special aspect selection is enforced on the query concepts to strengthen the representation and remove noise and ambiguity.

9. Information Extraction

Information extraction is the first step for computer to evaluate unstructured text by identifying key phrases and accordance within text. For this, pattern matching is used to look for predefined sequences in text. Information extraction includes tokenization, identification of named entities, sentence segmentation, and part-of-speech assignment. Initially phrases and sentences are parsed and semantically interpreted then required pieces of information are stored in database. This technology can be very useful when dealing with large amount of text. For many applications most challenging is electronic information which is in the form of free natural language documents rather than structured databases such as relational databases. Information extraction is a solution for this problem of reconstructing a collection of textual documents into a more structured database. For more mining of knowledge database which is built by an information extraction method can be then submitted to the KDD module.



In the accepted an draw itemsets, the transaction based algorithm discards individual items, from the individual item set depending on the system workload, i.e., if the system can allow 10 items from an itemset, whichisof12items,the transaction based would remove the last 2items from the itemset. These removed items from the item sets are then back to the buffer for next processing cycle.

10. Visual Mining

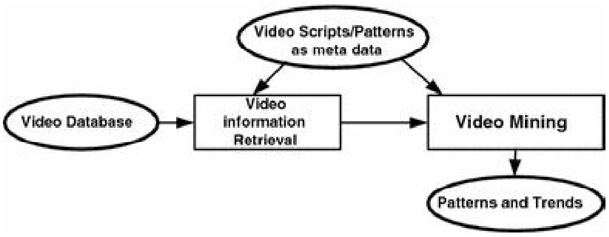
It is video data mining that deals with the extraction of implicit knowledge, video data relationships, or other patterns not explicitly stored in the video databases considered as an extension of still image mining by including mining of temporal image sequences. It is a process which not only automatically extracts content and structure of video, features of moving objects, spatial or temporal correlations of those features, but also discovers patterns of video structure, object activities, video events from vast amounts of video data with a little assumption of their contents.

Video information retrieval versus video data mining

It is video information retrieval, not information retrieval that is considered as part of the video data mining of a certain level knowledge discovery such as feature selection, dimensionality reduction and concept discovery.

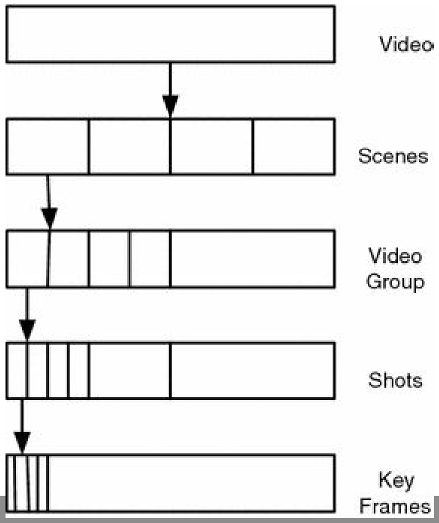
The dissimilarities of video data mining with related areas are as follows,

- **Video data mining versus computer vision or video processing:** The relationship between video processing and video mining is very subjective. The goal of video data mining is to extract patterns from the video sequences whereas video processing focuses on understanding and/or extracting features from the video database.
- **Video data mining versus pattern recognition:** Both areas share the feature extraction steps but differ in pattern specificity. The objective of pattern recognition is to recognize specific, classification patterns, pattern generation and analysis. Pattern recognition is indulging in a research on classifying special samples with an existing model while video mining is involved in discovering rules and patterns of samples with or without image processing. The objective of video data mining is to generate all significant patterns without prior knowledge of what they are.
- **Video information retrieval versus video data mining:** The difference is similar to the difference between database management and data mining.


Video Scripts/Patterns as meta data
Video Database
Video Information Retrieval
Video Mining
Patterns and Trends

11. Video segmentation

The first step in any video data management system is invariably, the segmentation of the video track into smaller units enabling the subsequent processing operations on video shots, such as video indexing, semantic representation or tracking of the selected video information and identifying the frames where a transition takes place from one shot to another. The visual-based segmentation identifies the shot boundaries and the motion-based segmentation identifies pans and zooms.



- **Video group:** It is an intermediate entity between the physical shots and semantic scenes serving as the bridge between the shot and scene.
- **Shot:** It is defined as a sequence of frames taken by a single camera with no major changes in the visual content.
- **Key frame:** The frame represents the salient visual contents of a shot. Since a large number of adjacent frames are likely to be similar, one or more key frames can be extracted from the shot depending on the complexity of the content of the shot.

12. Audio Mining

Audio is what plays a significant role in the detection and recognition of events in video. Supplying speech, music and various special sounds and can be used to separate different speeches, detect various audio events, analyze for spoken text, emotions, high-light detection in sports videos and so on. In movies, the audio is often correlated with the scene. For instance, the shots of fighting and explosions are mostly accompanied by a sudden change in the audio level.

In addition, audio features can be used to characterize the media signals to discriminate between music and speech classes. In general, the audio features can be categorized into two groups, namely, time domain features which include zero-crossing rates, amplitudes, pitches and the frequency domain features consisting of spectrograms, cepstral coefficients and mel-frequency cepstral coefficients. There are two main approaches to audio data mining. Firstly, Text-based indexing approach converts speech to text and then identifies words in a dictionary having several hundred thousand entries. If a word or name is not in the dictionary, the Large Vocabulary Continuous Speech Recognizers system will choose the most similar word it can find. Secondly, phoneme-based indexing approach analyzes and identifies sounds in a piece of audio content to create a phonetic-based index. It then uses a dictionary of several dozen phonemes to convert a user's search term to the correct phoneme string. Finally, the system looks for the search terms in the index.

13. Types of Mining Systems

Data Mining System : The selection of a data mining system depends on the following features "

- **Data Types :** The data mining system may handle formatted text, record-based data, and relational data. The data could also be in ASCII text, relational database data or data warehouse data. Therefore, we should check what exact format the data mining system can handle.
- **System Issues :** We must consider the compatibility of a data mining system with different operating systems. One data mining system may run on only one operating system or on several. There are also data mining systems that provide web-based user interfaces and allow XML data as input.
- **Data Sources :** Data sources refer to the data formats in which data mining system will operate. Some data mining system may work only on ASCII text files while others on multiple relational sources. Data mining system should also support ODBC connections or OLE DB for ODBC connections.

- **Data Mining functions and methodologies :** There are some data mining systems that provide only one data mining function such as classification while some provides multiple data mining functions such as concept description, discovery-driven OLAP analysis, association mining, linkage analysis, statistical analysis, classification, prediction, clustering, outlier analysis, similarity search, etc.

- **Coupling data mining with databases or data warehouse systems :** Data mining systems need to be coupled with a database or a data warehouse system. The coupled components are integrated into a uniform information processing environment. Here are the types of coupling listed below “

  - No coupling
  - Loose Coupling
  - Semi tight Coupling
  - Tight Coupling

- **Scalability :** There are two scalability issues in data mining -

  - **Row (Database size) Scalability :** A data mining system is considered as row scalable when the number or rows are enlarged 10 times. It takes no more than 10 times to execute a query.
  - **Column (Dimension) Salability :** A data mining system is considered as column scalable if the mining query execution time increases linearly with the number of columns.

- **Visualization Tools** - Visualization in data mining can be categorized as follows-

  - Data Visualization
  - Mining Results Visualization
  - Mining process visualization
  - Visual data mining

- **Data Mining query language and graphical user interface :** An easy-to-use graphical user interface is important to promote user-guided, interactive data mining. Unlike relational database systems, data mining systems do not share underlying data mining query language.

Important Questions of Data Mining and Data Warehouse VIVA Questions

1. What is data warehouse?

A data warehouse is a electronic storage of an Organization's historical data for the purpose of reporting, analysis and data mining or knowledge discovery.

2. What is the benefits of data warehouse?

A data warehouse helps to integrate data and store them historically so that we can analyze different aspects of business including, performance analysis, trend, prediction etc. over a given time frame and use the result of our analysis to improve the efficiency of business processes.

3. What are the different types of data warehosuing?

Following are the different types of Data warehousing:

- Enterprise Data warehousing
- Operational Data Store
- Data Mart

4. What is the difference between OLTP and OLAP?

OLTP is the transaction system that collects business data. Whereas OLAP is the reporting and analysis system on that data.

- OLTP systems are optimized for INSERT, UPDATE operations and therefore highly normalized.
- On the other hand, OLAP systems are deliberately denormalized for fast data retrieval through SELECT operations.

5. What is data mart?

Data marts are generally designed for a single subject area. An organization may have data pertaining to different departments like Finance, HR, Marketting etc. stored in data warehouse and each department may have separate data marts. These data marts can be built on top of the data warehouse.

6. What is dimension?

A dimension is something that qualifies a quantity (measure). For an example, consider this: If I just say... "20kg", it does not mean anything. But if I say, "20kg of Rice (Product) is sold to Ramesh (customer) on 5th April (date)", then that gives a meaningful sense. These product, customer and dates are some dimension that qualified the measure - 20kg.

Dimensions are mutually independent. Technically speaking, a dimension is a data element that categorizes each item in a data set into non-overlapping regions.

7. What is Fact?

A fact is something that is quantifiable (Or measurable). Facts are typically (but not always) numerical values that can be aggregated.

8. Briefly state different between data ware house & data mart?

Dataware house is made up of many datamarts. DWH contain many subject areas. but data mart focuses on one subject area generally. e.g. If there will be DHW of bank then there can be one data mart for accounts, one for Loans etc. This is high level definitions. Metadata is data about data. e.g. if in data mart we are receving any file. then metadata will contain information like how many columns, file is fix width/elimted, ordering of fileds, dataypes of field etc...

9. What is the difference between dependent data warehouse and independent data warehouse?

There is a third type of Datamart called Hybrid. The Hybrid datamart having source data from Operational systems or external files and central Datawarehouse as well. I will definitely check for Dependent and Independent Datawarehouses and update.

10. What are the storage models of OLAP?

ROLAP, MOLAP and HOLAP

11. What are CUBES?

A data cube stores data in a summarized version which helps in a faster analysis of data. The data is stored in such a way that it allows reporting easily.

E.g. using a data cube A user may want to analyze weekly, monthly performance of an employee. Here, month and week could be considered as the dimensions of the cube.

12. What is MODEL in Data mining world?

Models in Data mining help the different algorithms in decision making or pattern matching. The second stage of data mining involves considering various models and choosing the best one based on their predictive performance.

13. Explain how to mine an OLAP cube.

A data mining extension can be used to slice the data the source cube in the order as discovered by data mining. When a cube is mined the case table is a dimension.

14. Explain how to use DMX-the data mining query language.

Data mining extension is based on the syntax of SQL. It is based on relational concepts and mainly used to create and manage the data mining models. DMX comprises of two types of statements: Data definition and Data manipulation. Data definition is used to define or create new models, structures.

15. What is called data cleaning?

Name itself implies that it is a self explanatory term. Cleaning of Orphan records, Data breaching business rules, Inconsistent data and missing information in a database.

16. Define Rollup and cube.

Custom rollup operators provide a simple way of controlling the process of rolling up a member to its parents values.The rollup uses the contents of the column as custom rollup operator for each member and is used to evaluate the value of the member’s parents.

If a cube has multiple custom rollup formulas and custom rollup members, then the formulas are resolved in the order in which the dimensions have been added to the cube.

17. Differentiate between Data Mining and Data warehousing.

Data warehousing is merely extracting data from different sources, cleaning the data and storing it in the warehouse. Where as data mining aims to examine or explore the data using queries. These queries can be fired on the data warehouse. Explore the data in data mining helps in reporting, planning strategies, finding meaningful patterns etc.

E.g. a data warehouse of a company stores all the relevant information of projects and employees. Using Data mining, one can use this data to generate different reports like profits generated etc.

18. What is Star Schema?

Star schema is nothing but a type of organizing the tables in such a way that result can be retrieved from the database quickly in the data warehouse environment.

19. What is Snowflake Schema?

Snowflake schema which has primary dimension table to which one or more dimensions can be joined. The primary dimension table is the only table that can be joined with the fact table.

20. What is Discrete and Continuous data in Data mining world?

Discreet data can be considered as defined or finite data. E.g. Mobile numbers, gender. Continuous data can be considered as data which changes continuously and in an ordered fashion. E.g. age

21. What is a Decision Tree Algorithm?

A decision tree is a tree in which every node is either a leaf node or a decision node. This tree takes an input an object and outputs some decision. All Paths from root node to the leaf node are reached by either using AND or OR or BOTH. The tree is constructed using the regularities of the data. The decision tree is not affected by Automatic Data Preparation.

22. What is Naïve Bayes Algorithm?

Naïve Bayes Algorithm is used to generate mining models. These models help to identify relationships between input columns and the predictable columns. This algorithm can be used in the initial stage of exploration. The algorithm calculates the probability of every state of each input column given predictable columns possible states. After the model is made, the results can be used for exploration and making predictions.

23. Explain clustering algorithm.

Clustering algorithm is used to group sets of data with similar characteristics also called as clusters. These clusters help in making faster decisions, and exploring data. The algorithm first identifies relationships in a dataset following which it generates a series of clusters based on the relationships. The process of creating clusters is iterative. The algorithm redefines the groupings to create clusters that better represent the data.

24. Explain Association algorithm in Data mining?

Association algorithm is used for recommendation engine that is based on a market based analysis. This engine suggests products to customers based on what they bought earlier. The model is built on a dataset containing identifiers. These identifiers are both for individual cases and for the items that cases contain. These groups of items in a data set are called as an item set. The algorithm traverses a data set to find items that appear in a case. MINIMUM_SUPPORT parameter is used any associated items that appear into an item set.

25. What are the goals of data mining?

Prediction, identification, classification and optimization

26. Is data mining independent subject?

No, it is interdisciplinary subject. includes, database technology, visualization, machine learning, pattern recognition, algorithm etc.

27. What are different types of database?

Relational database, data warehouse and transactional database.

28. What are data mining functionality?

Mining frequent pattern, association rules, classification and prediction, clustering, evolution analysis and outlier Analise

29. What are issues in data mining?

Issues in mining methodology, performance issues, user interactive issues, different source of data types issues etc.

30. List some applications of data mining.

Agriculture, biological data analysis, call record analysis, DSS, Business intelligence system etc

31. What do you mean by interesting pattern?

A pattern is said to be interesting if it is 1. easily understood by human 2. valid 3. potentially useful 4. novel

32. Why do we pre-process the data?

To ensure the data quality. [accuracy, completeness, consistency, timeliness, believability, interpret-ability]

33. What are the steps involved in data pre-processing?

Data cleaning, data integration, data reduction, data transformation.

34. What is distributed data warehouse?

Distributed data warehouse shares data across multiple data repositories for the purpose of OLAP operation.

35. Define virtual data warehouse.

A virtual data warehouse provides a compact view of the data inventory. It contains meta data and uses middle-ware to establish connection between different data sources.

36. What is are different data warehouse model?

- Enterprise data ware houst
- Data marts
- Virtual Data warehouse

37. What is the definition of Cube in Datawarehousing?

Cubes are logical representation of multidimensional data. The edge of the cube has the dimension members,and the body of the cube contains the data values.

38. List few roles of data warehouse manager.

Creation of data marts, handling users, concurrency control, updation etc,

39. What are different types of cuboids?

- 0-D cuboids are called as apex cuboids
- n-D cuboids are called base cuboids
- Middle cuboids

40. What are the forms of multidimensional model?

- Star schema
- Snow flake schema
- Fact constellation Schema

41. What are frequent pattern?

A set of items that appear frequently together in a transaction data set.eg milk, bread, sugar

42. What are the issues regarding classification and prediction?

- Preparing data for classification and prediction
- Comparing classification and prediction

43. Define model over fitting.

A model that fits training data well can have generalization errors. Such situation is called as model over fitting.

44. What are the methods to remove model over fitting?

- Pruning [Pre-pruning and post pruning)
- Constraint in the size of decision tree
- Making stopping criteria more flexible

45. What is regression?

- Regression can be used to model the relationship between one or more independent and dependent variables.
- Linear regression and non-linear regression

46. Compare K-mean and K-mediods algorithm.

K-mediods is more robust than k-mean in presence of noise and outliers. K-Mediods can be computationally costly.

47. What is K-nearest neighbor algorithm?

It is one of the lazy learner algorithm used in classification. It finds the k-nearest neighbor of the point of interest.

48. What is Baye's Theorem?

P(H/X) = P(X/H)* P(H)/P(X)

49. What is concept Hierarchy?

It defines a sequence of mapping from a set of low level concepts to higher -level, more general concepts.

50. What are the causes of model over fitting?

- ➢ Due to presence of noise
- ➢ Due to lack of representative samples
- ➢ Due to multiple comparison procedure

51. What is decision tree classifier?

A decision tree is an hierarchically based classifier which compares data with a range of properly selected features.

52. If there are n dimensions, how many cuboids are there?

There would be 2 ^ n cuboids.

53. What is spatial data mining?

- ➢ Spatial data mining is the process of discovering interesting, useful, non-trivial patterns from large spatial datasets.
- ➢ Spatial Data Mining = Mining Spatial Data Sets (i.e. Data Mining + Geographic Information Systems)

54. What is multimedia data mining?

Multimedia Data Mining is a subfield of data mining that deals with an extraction of implicit knowledge, multimedia data relationships, or other patterns not explicitly stored in multimedia databases

55. What are different types of multimedia data?

image, video, audio

56. What is text mining?

Text mining is the procedure of synthesizing information, by analyzing relations, patterns, and rules among textual data. These procedures contains text summarization, text categorization, and text clustering.

57. List some application of text mining.

- ➢ Customer profile analysis
- ➢ patent analysis
- ➢ Information dissemination
- ➢ Company resource planning

58. What do you mean by web content mining?

Web content mining refers to the discovery of useful information from Web contents, including text, images, audio, video, etc.

59. Define web structure mining and web usage mining.

Web structure mining studies the model underlying the link structures of the Web. It has been used for search engine result ranking and other Web applications.

Web usage mining focuses on using data mining techniques to analyze search logs to find interesting patterns. One of the main applications of Web usage mining is its use to learn user profiles.

60. What are frequent patterns?

These are the patterns that appear frequently in a data set.

item-set, sub sequence, etc

61. What is data warehouse?

A data warehouse is a electronic storage of an Organization's historical data for the purpose of reporting, analysis and data mining or knowledge discovery.

62. What is data characterization?

Data Characterization is s summarization of the general features of a target class of data. Example, analyzing software product with sales increased by 10%

63. What is data discrimination?

Data discrimination is the comparison of the general features of the target class objects against one or more contrasting objects.

64. What can business analysts gain from having a data warehouse?

- ➢ First, having a data warehouse may provide a competitive advantage by presenting relevant information from which to measure performance and make critical adjustments in order to help win over competitors.

- Second, a data warehouse can enhance business productivity because it is able to quickly and efficiently gather information that accurately describes the organization.
- Third, a data warehouse facilitates customer relationship management because it provides a consistent view of customers and item across all lines of business, all departments and all markets.
- Finally, a data warehouse may bring about cost reduction by tracking trends, patterns, and exceptions over long periods in a consistent and reliable manner.

65. Why is association rule necessary?

In data mining, association rule learning is a popular and well researched method for discovering interesting relations between variables in large databases.

It is intended to identify strong rules discovered in database using different measures of interesting.

66. What are two types of data mining tasks?

- Descriptive task
- Predictive task

67. Define classification.

Classification is the process of finding a model (or function) that describes and distinguishes data classes or concepts.

68. What are outliers?

A database may contain data objects that do not comply with the general behavior or model of the data. These data objects are called outliers.

69. What do you mean by evolution analysis?

- Data evolution analysis describes and models regularities or trends for objects whose behavior change over time.
- Although this may include characterization, discrimination, association and correlation analysis, classification, prediction, or clustering of time related data.
- Distinct features of such as analysis include time-series data analysis, sequence or periodicity pattern matching, and similarity-based data analysis.

70. Define KDD.

The process of finding useful information and patterns in data.

71. What are the components of data mining?

Database, Data Warehouse, World Wide Web, or other information repository

(a) Database or Data Warehouse Server

(b) Knowledge Based

(c) Data Mining Engine

(d) Pattern Evaluation Module

(e) User Interface

72. Define metadata.

A database that describes various aspects of data in the warehouse is called metadata.

73. What are the usage of metadata?

- Map source system data to data warehouse tables
- Generate data extract, transform, and load procedures for import jobs
- Help users discover what data are in the data warehouse
- Help users structure queries to access data they need

74. List the demerits of distributed data warehouse.

- There is no metadata, no summary data or no individual DSS (Decision Support System) integration or history. All queries must be repeated, causing additional burden on the system.

- Since compete with production data transactions, performance can be degraded.
- There is no refreshing process, causing the queries to be very complex.

75. Define HOLAP.

The hybrid OLAP approach combines ROLAP and MOLAP technology.

76. What are data mining techniques?

- Association rules
- Classification and prediction
- Clustering
- Deviation detection
- Similarity search
- Sequence Mining

77. List different data mining tools.

- Traditional data mining tools
- Dashboards
- Text mining tools

78. Define sub sequence.

A subsequence, such as buying first a PC, the a digital camera, and then a memory card, if it occurs frequently in a shopping history database, is a (frequent) sequential pattern.

79. What is data warehouse?

A data warehouse is a electronic storage of an Organization's historical data for the purpose of reporting, analysis and data mining or knowledge discovery.

80. What is the main goal of data mining?

Prediction

81. List the typical OLAP operations.

- Roll UP
- DRILL DOWN
- ROTATE
- SLICE AND DICE
- DRILL trough and drill across

82. If there are 3 dimensions, how many cuboids are there in cube?

2 ^ 3 = 8 cuboids

83. Differentiate between star schema and snowflake schema.

Star Schema is a multi-dimension model where each of its disjoint dimension is represented in single table.

- Snow-flake is normalized multi-dimension schema when each of disjoint dimension is represent in multiple tables.
- Star schema can become a snow-flake
- Both star and snowflake schemas are dimensional models; the difference is in their physical implementations.
- Snowflake schemas support ease of dimension maintenance because they are more normalized.
- Star schemas are easier for direct user access and often support simpler and more efficient queries.
- It may be better to create a star version of the snowflaked dimension for presentation to the users

84. List the advantages of star schema.

- Star Schema is very easy to understand, even for non technical business manager.
- Star Schema provides better performance and smaller query times
- Star Schema is easily extensible and will handle future changes easily

85. What are the characteristics of data warehouse?

- Integrated
- Non-volatile
- Subject oriented
- Time varient

86. Define support and confidence.

The support for a rule R is the ratio of the number of occurrences of R, given all occurrences of all rules.

The confidence of a rule X->Y, is the ratio of the number of occurrences of Y given X, among all other occurrences given X

87. What are the criteria on the basic of which classification and prediction can be compared?

speed, accuracy, robustness, scalability, goodness of rules, interpret-ability

88. What is Data purging?

The process of cleaning junk data is termed as data purging. Purging data would mean getting rid of unnecessary NULL values of columns. This usually happens when the size of the database gets too large.

89. What is the difference between metadata and data dictionary?

Metadata is defined as data about the data. But, Data dictionary contain the information about the project information, graphs, abinito commands and server information.