Paper Code: BCA 302 L T C **Paper ID: 20302 3 1 4**
Paper: Data warehouse and data mining Pre-requisites: • Information System Concepts

Objectives: This course is an attempt to provide you with the basic information about data ware house and their development. This course also provides the basic conceptual background necessary to design and develop data ware house applications.

INSTRUCTIONS TO PAPER SETTERS: Maximum Marks: 75

1. Question No. 1 should be compulsory and cover the entire syllabus. This question should have objective or short answer type questions. It should be of 25 marks.
2. Apart from Question No. 1, rest of the paper shall consist of four units as per the syllabus. Every unit should have two questions. However, student may be asked to attempt only 1 question from each unit. Each question should be 12.5 marks.

UNIT – I

Data mining: Introduction, Data mining – on what kind of data, data mining functionalities – what kind of patterns to be mined, Classification of data mining systems, data mining task primitives, integration of a data mining systems with a database or data warehouse systems, major issues in data mining.

Data preprocessing: Descriptive data summarization, data cleaning, data integration and transformation, data reduction, data descretization and concept hierarchy generation.

[No. of Hrs: 11]

UNIT – II

Data warehouse and OLAP technology: What is data warehouse, A multidimensional data model, data warehouse architecture, data warehouse implementation, data warehouse usage, OLAP, OLAM

Mining frequent patterns, association and correlation, efficient and scalable frequent itemset mining methods, From association mining to correlation analysis.

[No. of Hrs: 11]

UNIT – III

Classification and prediction: Introduction, issues, classification by decision tree induction, rule based classification, classification by back propagation, lazy learners, other classification methods, Prediction: accuracy and error measures, evaluating the accuracy of a classifier or predictor.

Cluster Analysis: Types of data in cluster analysis, a categorization of major clustering methods, partitioning methods.

[No. of Hrs: 11]

UNIT – IV

Mining complex types of data: Multidimensional analysis and descriptive mining of complex data objects, mining spatial database, multimedia database, mining world wide web.

Applications and trends in data mining: Data mining applications, data mining system products and research prototypes, social impact of data mining, trends in data mining.

[No. of Hrs: 11]

UNIT – I

Data mining:
It is a process of extracting hidden predictive information from large databases. It is a powerful new technology to help companies focus on the most important information in their data warehouses. Data mining tools predict future trends and behaviors, allowing businesses to make proactive, knowledge-driven decisions. For a commercial business, the discovery of previously unknown statistical patterns or trends can provide valuable insight into the function and environment of their organization. Data-mining techniques can generally be grouped into two categories: predictive method and descriptive method. Descriptive method: It a method of finding human interpretable patterns that describe the data. Data mining in this case is useful to group together similar documents returned by search engine according to their context. Predictive method: In this method, we can use some variables to predict unknown or future values of other variable. It is used to predict whether a newly arrived customer will spend more than 100$ at a department store. **Data-mining techniques**: The following list describes many data-mining techniques in use today. Each of these techniques exists in several variations and can be applied to one or more of the categories above.
 Regression modeling—This technique applies standard statistics to data to prove or disprove a hypothesis. One example of this is linear regression, in which variables are measured against a standard or target variable path over time. A second example is logistic regression, where the probability of an event is predicted based on known values in correlation with the occurrence of prior similar events.

- Visualization—This technique builds multidimensional graphs to allow a data analyst to decipher trends, patterns, or relationships.
- Correlation—This technique identifies relationships between two or more variables in a data group.
- Variance analysis—This is a statistical technique to identify differences in mean values between a target or known variable and nondependent variables or variable groups.
- Discriminate analysis—This is a classification technique used to identify or "discriminate" the factors leading to membership within a grouping.
- Forecasting—Forecasting techniques predict variable outcomes based on the known outcomes of past events.
- Cluster analysis—This technique reduces data instances to cluster groupings and then analyzes the attributes displayed by each group.
- Decision trees—Decision trees separate data based on sets of rules that can be described in "if-then-else" language.
- Neural networks—Neural networks are data models that are meant to simulate cognitive functions. These techniques "learn" with each iteration through the data, allowing for greater flexibility in the discovery of patterns and trends.

DATA MINING FUNCTIONALITIES – WHAT KIND OF PATTERNS TO BE MINED, DATA MINING AS A PART OF KNOWLEDGE DISCOVERY IN DATABASE:

Data mining addresses inductive knowledge which discovers new rules and patterns from the supplied data. It comprises six phases such as data selection, data cleansing, enrichment, data transformation or encoding, data mining and the reporting and display of the discovered information.

KDD process: Consider a transaction database maintained by a specialty consumer goods retailer. Client data includes customer name, zip code, phone number, data of purchase, item code, price, quantity and total amount.KDD process can be applied to this database to discover variety of knowledge.

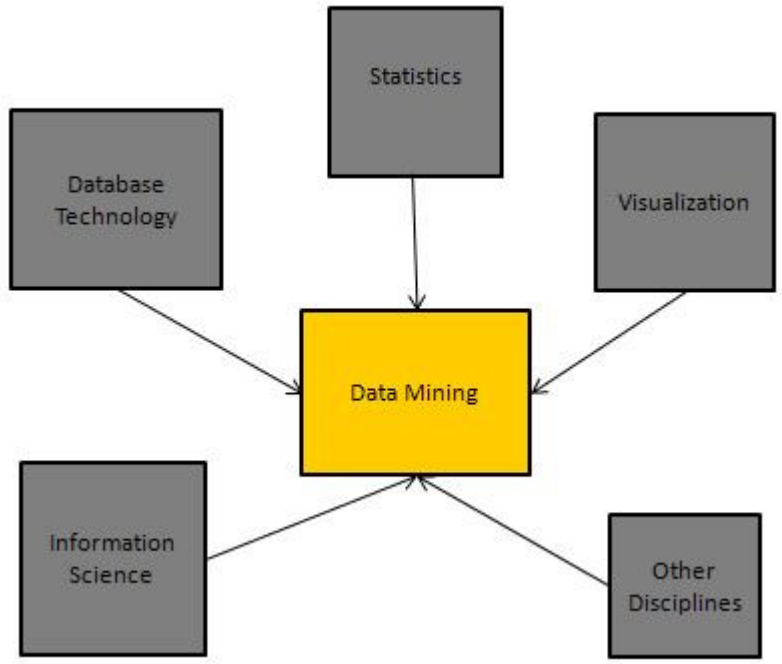
- Data selection: Selecting a data set, or focusing on a subset of variables or data samples, on which discovery is to be performed. Data about specific item or categories of items, or from stores in a specific region or area of the country, may be selected.
- Data cleansing: It checks and resolves data conflicts, outliers, noisy, erroneous, missing data and ambiguity. In this they may correct invalid zip codes or eliminate records with incorrect phone prefixes.
- Enrichment: Enhances the data with additional sources of information. For example, with client name and phone numbers, new information about the client such as age, income and credit rating can be appended to each record.
- Data transformation and encoding: Data is transformed or consolidated into forms appropriate for mining, by performing summary, or aggregation, operations. It is done to reduce the amount of data. For example, item codes may be grouped in terms of product categories into audio, video, supplies, accessories and so on.
- Data mining: It is a process where intelligent methods are applied to extracts meaningful new patterns. It searches for patterns of interest in a particular representational form or a set of such representations, including classification rules or trees, regression, clustering, sequence modeling and so on.
- Reporting: The results of data mining may be reported in a variety of formats, such as listings, graphic outputs, summary tables or visualizations

DATA MINING SYSTEM CLASSIFICATION

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:

- Characterization
- Discrimination
- Association and Correlation Analysis
- Classification
- Prediction
- Clustering
- Outlier Analysis
- 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:

- Finance
- Telecommunications
- DNA
- Stock Markets
- E-mail

INTEGRATING DATA MINING SYSTEM WITH A DATABASE OR DATA WAREHOUSE SYSTEM

The data mining system needs to be integrated with database or the data warehouse system. If the data mining system is not integrated with any database or data warehouse system then there will be no system to communicate with. This scheme is known as non-coupling scheme. In this scheme the main focus is put on data mining design and for developing efficient and effective algorithms for mining the available data sets.

Here is the list of Integration Schemes:

- No Coupling - In this scheme the Data Mining system does not utilize any of the database or data warehouse functions. It then fetches the data from a particular source and process that data using some data mining algorithms. The data mining result is stored in other file.

- Loose Coupling - In this scheme the data mining system may use some of the functions of database and data warehouse system. It then fetches the data from data respiratory managed by these systems and perform data mining on that data. It then stores the mining result either in a file or in a designated place in a database or data warehouse.

- Semi-tight Coupling - In this scheme the data mining system is along with the kinking the efficient implementation of data mining primitives can be provided in database or data warehouse systems.

- Tight coupling - In this coupling scheme data mining system is smoothly integrated into database or data warehouse system. The data mining subsystem is treated as one functional component of an information system.

Data Mining Task Primitives

- We can specify the data mining task in form of data mining query.

- This query is input to the system.

- The data mining query is defined in terms of data mining task primitives.

Note: Using these primitives allow us to communicate in interactive manner with the data mining system. Here is the list of Data Mining Task Primitives:

- Set of task relevant data to be mined

- Kind of knowledge to be mined

- Background knowledge to be used in discovery process

- Interestingness measures and thresholds for pattern evaluation

- Representation for visualizing the discovered patterns

Set of task relevant data to be mined

This is the portion of database in which the user is interested. This portion includes the following:

- Database Attributes
- Data Warehouse dimensions of interest

Kind of knowledge to be mined

It refers to the kind of functions to be performed. These functions are:

- Characterization
- Discrimination
- Association and Correlation Analysis
- Classification
- Prediction
- Clustering
- Outlier Analysis
- Evolution Analysis

Background knowledge to be used in discovery process

The background knowledge allow data to be mined at multiple level of abstraction. For example the Concept hierarchies are one of the background knowledge that allow data to be mined at multiple level of abstraction.

Interestingness measures and thresholds for pattern evaluation

This is used to evaluate the patterns that are discovers by the process of knowledge discovery. There are different interestingness measures for different kind of knowledge.

Representation for visualizing the discovered patterns

This refers to the form in which discovered patterns are to be displayed. These representations may include the following:

- Rules
- Tables
- Charts
- Graphs
- Decision Trees
- Cubes

MAJOR ISSUES IN DATA MINING

1. Mining Methodology
   - Mining various and new kinds of knowledge
   - Mining knowledge in multi-dimensional space
   - Data mining: An interdisciplinary effort
   - Boosting the power of discovery in a networked environment
   - Handling noise, uncertainty, and incompleteness of data
   - Pattern evaluation and pattern- or constraint-guided mining
2. User Interaction
   - Interactive mining
   - Incorporation of background knowledge
   - Presentation and visualization of data mining results
3. Efficiency and Scalability
   - Efficiency and scalability of data mining algorithms
   - Parallel, distributed, stream, and incremental mining methods
4. Diversity of data types
   - Handling complex types of data
   - Mining dynamic, networked, and global data repositories
5. Data mining and society
   - Social impacts of data mining
   - Privacy-preserving data mining
   - Invisible data mining

DATA PREPROCESSING: DESCRIPTIVE DATA SUMMARIZATION, DATA CLEANING, DATA INTEGRATION AND TRANSFORMATION, DATA REDUCTION, DATA DESCRETIZATION AND CONCEPT HIERARCHY GENERATION

Data preprocessing is a data mining technique that involves transforming raw data into an understandable format. Real-world data is often incomplete, inconsistent, and/or lacking in certain behaviors or trends, and is likely to contain many errors. Data preprocessing is a proven

method of resolving such issues. Data preprocessing prepares raw data for further processing.

Data preprocessing is used database-driven applications such as customer relationship management and rule-based applications (like neural networks).
Data goes through a series of steps during preprocessing:

- Data Cleaning: Data is cleansed through processes such as filling in missing values, smoothing the noisy data, or resolving the inconsistencies in the data.
- Data Integration: Data with different representations are put together and conflicts within the data are resolved.
- Data Transformation: Data is normalized, aggregated and generalized.
- Data Reduction: This step aims to present a reduced representation of the data in a data warehouse.
- Data Discretization: Involves the reduction of a number of values of a continuous attribute by dividing the range of attribute intervals.

UNIT II

DATA WAREHOUSE INTRODUCTION

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 In mon 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

Types of Data Warehouse

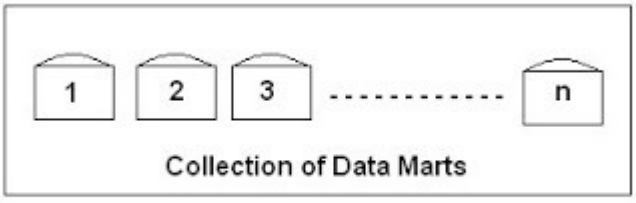
There are mainly three type of Data Warehouse.....

1). Enterprise Data Warehouse.
2). Operational data store.
3). Data Mart.

Enterprise Data Warehouse provide a control Data Base for decision support through out the enterprise.

Operational data store has a broad enterprise under scope but unlike a real enterprise DW. Data is refreshed in rare real time and used for routine business activity.

Data Mart is a sub part of Data Warehouse. It support a particular reason or it is design for particular lines of business such as sells, marketing or finance, or in any organization documents of a particular department will be data mart



Data Warehouse

DBMS schemas for decision support

The basic concepts of dimensional modeling are: facts, dimensions and measures. A fact is a collection of related data items, consisting of measures and context data. It typically represents business items or business transactions. A dimension is a collection of data that describe one business dimension. Dimensions determine the contextual background for the facts; they are the parameters over which we want to perform OLAP. A measure is a numeric attribute of a fact, representing the performance or behavior of the business relative to the dimensions. Considering Relational context, there are three basic schemas that are used in dimensional modeling:

1. Star schema

2. Snowflake schema

3. Fact constellation schema

Star schema :The multidimensional view of data that is expressed using relational data base semantics is provided by the data base schema design called star schema. The basic of stat schema is that Information can be classified into two groups:

• Facts

• Dimension

Star schema has one large central table (fact table) and a set of smaller tables (dimensions) arranged in a radial pattern around the central table. Facts are core data element being analyzed while dimensions are attributes about the facts. The determination of which schema model should be used for a data warehouse should be based upon the analysis of project requirements, accessible tools and project team preferences. What is star schema? The star schema architecture is the simplest data warehouse schema. It is called a star schema because the diagram resembles a star, with points radiating from a center. The center of the star consists of fact table and the points of the star are the dimension tables. Usually the fact tables in a star schema are in third normal form(3NF) whereas dimensional tables are de-normalized. Despite the fact that the star schema is the simplest architecture, it is most commonly used nowadays and is recommended by Oracle.

Fact Tables A fact table is a table that contains summarized numerical and historical data (facts) and a multipart index composed of foreign keys from the primary keys of related dimension tables. A fact table typically has two types of columns: foreign keys to dimension tables and measures those that contain numeric facts. A fact table can contain fact's data on detail or aggregated level.

Dimension Tables Dimensions are categories by which summarized data can be viewed. E.g. a profit summary in a fact table can be viewed by a Time dimension (profit by month, quarter, year), Region dimension (profit by country, state, city), Product dimension (profit for product1, product2).A dimension is a structure usually composed of one or more hierarchies that categorizes data. If a dimension hasn't got a hierarchies and levels it is called flat dimension or list. The primary keys of each of the dimension tables are part of the composite primary key of the fact table. Dimensional attributes help to describe the dimensional value. They are normally descriptive, textual values. Dimension tables are generally small in size then fact table. Typical fact tables store data about sales while dimension tables data about geographic region (markets, cities), clients, products, times, channels.

Measures: Measures are numeric data based on columns in a fact table. They are the primary data which end users are interested in. E.g. a sales fact table may contain a profit measure which represents profit on each sale. Aggregations are pre calculated numeric data. By calculating and storing the answers to a query before users ask for it, the query processing time can be reduced. This is key in providing fast query performance in OLAP .Cubes are data processing units composed of fact tables and dimensions from the data warehouse. They provide multidimensional views of data, querying and analytical capabilities to clients.

The main characteristics of star schema:

• Simple structure -> easy to understand schema

• Great query effectives -> small number of tables to join • Relatively long time of loading data into dimension tables -> de-normalization, redundancy data caused that size of the table could be large.

• The most commonly used in the data warehouse implementations -> widely supported by a large number of business intelligence tools snowflake schema: is the result of decomposing one or more of the dimensions. The many-to-one relationships among sets of attributes of a dimension can separate new dimension tables, forming a hierarchy. The decomposed snowflake structure visualizes the hierarchical structure of dimensions very well. Fact constellation schema: For each star schema it is possible to construct fact constellation schema (for example by splitting the original star schema into more star schemes each of them describes facts on another level of dimension hierarchies). The fact constellation architecture contains multiple fact tables that share many dimension tables.

The main shortcoming of the fact constellation schema is a more complicated design because many variants for particular kinds of aggregation must be considered and selected. Moreover, dimension tables are still large.

OLAP, OLAM

OLAP : OLAP stands for Online Analytical Processing. It uses database tables (fact and dimension tables) to enable multidimensional viewing, analysis and querying of large amounts of data. E.g. OLAP technology could provide management with fast answers to complex queries on their operational data or enable them to analyze their company's historical data for trends and patterns. Online Analytical Processing (OLAP) applications and tools are those that are designed to ask "complex queries of large multidimensional collections of data." Due to that OLAP is accompanied with data warehousing.

Need The key driver of OLAP is the multidimensional nature of the business problem. These problems are characterized by retrieving a very large number of records that can reach gigabytes and terabytes and summarizing this data into a form information that can by used by business analysts. One of the limitations that SQL has, it cannot represent these complex problems. A query will be translated in to several SQL statements. These SQL statements will involve multiple joins, intermediate tables, sorting, aggregations and a huge temporary memory to store these tables. These procedures required a lot of computation which will require a long time in computing. The second limitation of SQL is its inability to use mathematical models in these SQL statements. If an analyst, could create these complex statements using SQL statements, still there will be a large number of computation and huge memory needed. Therefore the use of OLAP is preferable to solve this kind of problem. Categories of OLAP Tools

MOLAP : This is the more traditional way of OLAP analysis. In MOLAP, data is stored in a multidimensional cube. The storage is not in the relational database, but in proprietary formats. That is, data stored in array-based structures .Advantages: • Excellent performance: MOLAP cubes are built for fast data retrieval, and are optimal for slicing and dicing operations. • Can perform complex calculations: All calculations have been pre-generated when the cube is created. Hence, complex calculations are not only doable, but they return quickly. Disadvantages: • Limited in the amount of data it can handle: Because all calculations are

performed when the cube is built, it is not possible to include a large amount of data in the cube itself. This is not to say that the data in the cube cannot be derived from a large amount of data. Indeed, this is possible. But in this case, only summary-level information will be included in the cube itself.• Requires additional investment: Cube technology are often proprietary and do not already exist in the organization. Therefore, to adopt MOLAP technology, chances are additional investments in human and capital resources are needed. Examples: Hyperion Essbase, Fusion (Information Builders)

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

• Can handle large amounts of data: The data size limitation of ROLAP technology is the limitation on data size of the underlying relational database. In other words, ROLAP itself places no limitation on data amount. • Can leverage functionalities inherent in the relational database: Often, relational database already comes with a host of functionalities. ROLAP technologies, since they sit on top of the relational database, can therefore leverage these functionalities. **Disadvantages:** • Performance can be slow: Because each ROLAP report is essentially a SQL query (or multiple SQL queries) in the relational database, the query time can be long if the underlying data size is large. • Limited by SQL functionalities: Because ROLAP technology mainly relies on generating SQL statements to query the relational database, and SQL statements do not fit all needs (for example, it is difficult to perform complex calculations using SQL), ROLAP technologies are therefore traditionally limited by what SQL can do. ROLAP vendors have mitigated this risk by building into the tool out-of-the-box complex functions as well as the ability to allow users to define their own functions. Examples: Microstrategy Intelligence Server, MetaCube (Informix/IBM)

HOLAP (MQE: Managed Query Environment)HOLAP technologies attempt to combine the advantages of MOLAP and ROLAP. For summary-type information, HOLAP leverages cube technology for faster performance. It stores only the indexes and aggregations in the multidimensional form while the rest of the data is stored in the relational database.Examples: PowerPlay (Cognos), Brio, Microsoft Analysis Services, Oracle Advanced Analytic Services

OLAP GuidelinesDr. E.F. Codd the “father” of the relational model, created a list of rules to deal with the OLAP systems. Users should priorities these rules according to their needs to match their business requirements.

These rules are:

1)Multidimensional conceptual view: The OLAP should provide an appropriate multidimensional Business model that suits the Business problems and Requirements.

2) Transparency: The OLAP tool should provide transparency to the input data for the users.

3) Accessibility: The OLAP tool should only access the data required only to the analysis needed.

4) Consistent reporting performance: The Size of the database should not affect in any way the performance.

5) Client/server architecture: The OLAP tool should use the client server architecture to ensure better performance and flexibility.

6) Generic dimensionality: Data entered should be equivalent to the structure and operation requirements.

7) Dynamic sparse matrix handling: The OLAP too should be able to manage the sparse matrix and so maintain the level of performance.

8) Multi-user support: The OLAP should allow several users working concurrently to work together.

9) Unrestricted cross-dimensional operations: The OLAP tool should be able to perform operations across the dimensions of the cube.

10)Intuitive data manipulation. "Consolidation path re-orientation, drilling down across columns or rows, zooming out, and other manipulation inherent in the consolidation path outlines should be accomplished via direct action upon the cells of the analytical model, and should neither require the use of a menu nor multiple trips across the user interface.

11) Flexible reporting: It is the ability of the tool to present the rows and column in a manner suitable to be analyzed.

12)Unlimited dimensions and aggregation levels: This depends on the kind of Business, where multiple dimensions and defining hierarchies can be made. In addition to these guidelines an OLAP system should also support:• Comprehensive database management tools: This gives the database management to control distributed Businesses• The ability to drill down to detail source record level: Which requires that The OLAP tool should allow smooth transitions in the multidimensional database.• Incremental database refresh: The OLAP tool should provide partial refresh.• Structured Query Language (SQL interface): the OLAP system should be able to integrate effectively in the surrounding enterprise environment.

OLTP vs OLAP

OLTP stands for On Line Transaction Processing and is a data modeling approach typically used to facilitate and manage usual business applications. Most of applications you see and use are OLTP based. OLTP technology used to perform updates on operational or transactional systems (e.g., point of sale systems)

OLAP stands for On Line Analytic Processing and is an approach to answer multi-dimensional queries. OLAP was conceived for Management Information Systems and Decision Support Systems. OLAP technology used to perform complex analysis of the data in a data warehouse.

Online Analytical Processing Server (OLAP) is based on multidimensional data model. It allows the managers , analysts to get insight the information through fast, consistent, interactive access to information. In this chapter we will discuss about types of OLAP, operations on OLAP, Difference between OLAP and Statistical Databases and OLTP.

Types of OLAP Servers

We have four types of OLAP servers that are listed below.

- Relational OLAP(ROLAP)
- Multidimensional OLAP (MOLAP)
- Hybrid OLAP (HOLAP)
- Specialized SQL Servers

Relational OLAP(ROLAP)

The Relational OLAP servers are placed between relational back-end server and client front-end tools. To store and manage warehouse data the Relational OLAP use relational or extended-relational DBMS.

ROLAP includes the following.

- implementation of aggregation navigation logic.
- optimization for each DBMS back end.
- additional tools and services.

Multidimensional OLAP (MOLAP)

Multidimensional OLAP (MOLAP) uses the array-based multidimensional storage engines for multidimensional views of data.With multidimensional data stores, the storage utilization may be low if the data set is sparse. Therefore many MOLAP Server uses the two level of data storage representation to handle dense and sparse data sets.

Hybrid OLAP (HOLAP)

The hybrid OLAP technique combination of ROLAP and MOLAP both. It has both the higher scalability of ROLAP and faster computation of MOLAP. HOLAP server allows to store the large data volumes of detail data. the aggregations are stored separated in MOLAP store.

Specialized SQL Servers

specialized SQL servers provides advanced query language and query processing support for SQL queries over star and snowflake schemas in a read-only environment.

OLAP Operations

As we know that the OLAP server is based on the multidimensional view of data hence we will discuss the OLAP operations in multidimensional data.

Here is the list of OLAP operations.

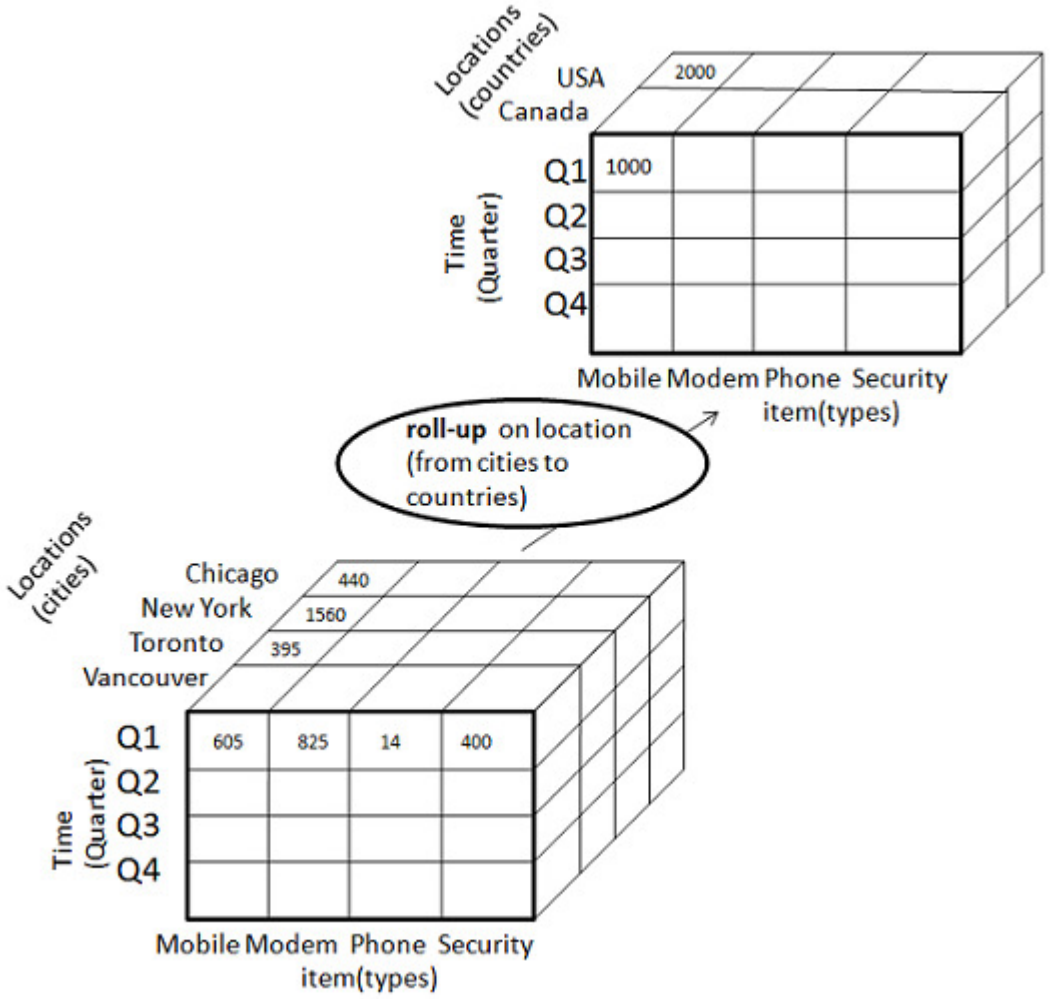
- Roll-up
- Drill-down
- Slice and dice
- Pivot (rotate)

Roll-up

This operation performs aggregation on a data cube in any of the following way:

- By climbing up a concept hierarchy for a dimension
- By dimension reduction.

Consider the following diagram showing the roll-up operation.



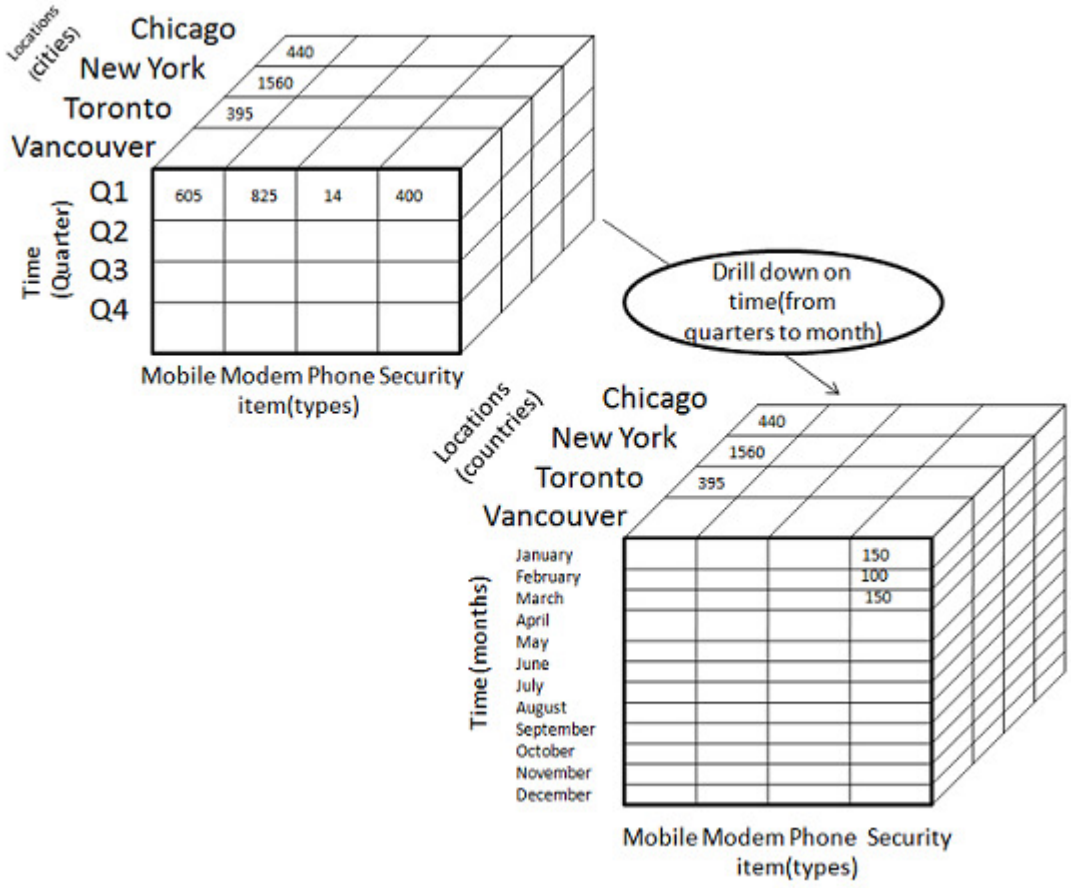
- The roll-up operation is performed by climbing up a concept hierarchy for the dimension location.
- Initially the concept hierarchy was "street < city < province < country".
- On rolling up the data is aggregated by ascending the location hierarchy from the level of city to level of country.
- The data is grouped into cities rather than countries.
- When roll-up operation is performed then one or more dimensions from the data cube are removed.

Drill-down

Drill-down operation is reverse of the roll-up. This operation is performed by either of the following way:

- By stepping down a concept hierarchy for a dimension.
- By introducing new dimension.

Consider the following diagram showing the drill-down operation:

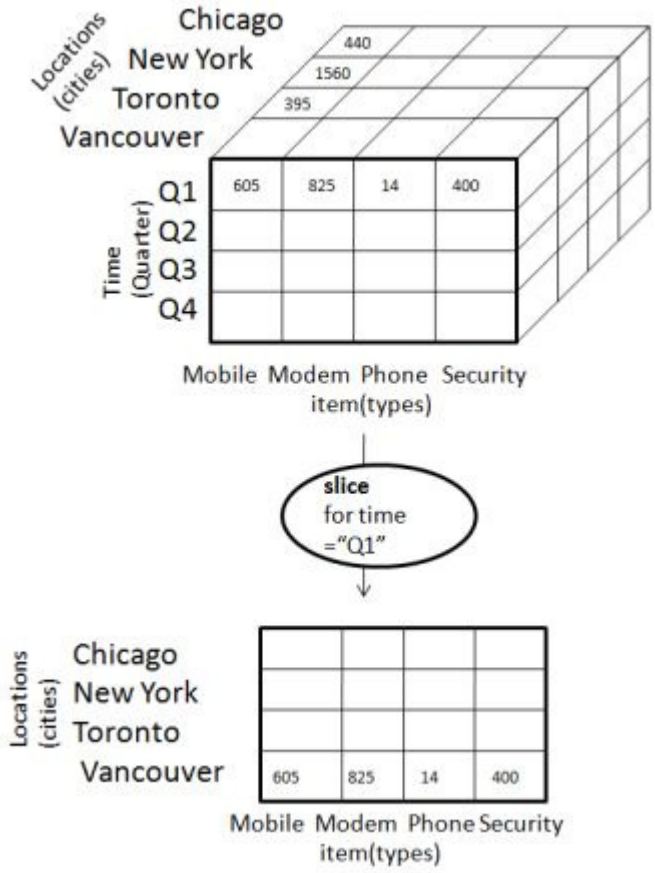


- The drill-down operation is performed by stepping down a concept hierarchy for the dimension time.
- Initially the concept hierarchy was "day < month < quarter < year."
- On drill-up the time dimension is descended from the level quarter to the level of month.

- When drill-down operation is performed then one or more dimensions from the data cube are added.
- It navigates the data from less detailed data to highly detailed data.

Slice

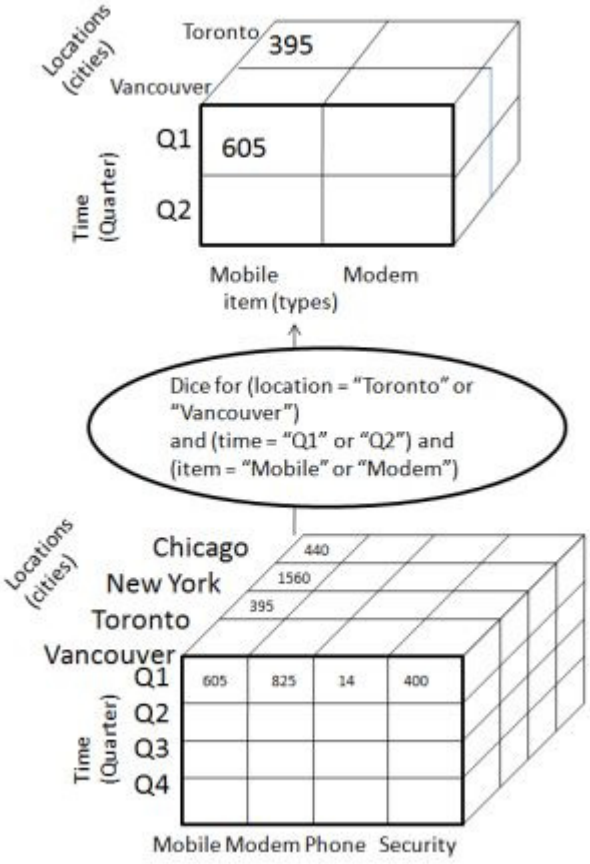
The slice operation performs selection of one dimension on a given cube and give us a new sub cube. Consider the following diagram showing the slice operation.



- The Slice operation is performed for the dimension time using the criterion time ="Q1".
- It will form a new sub cube by selecting one or more dimensions.

Dice

The Dice operation performs selection of two or more dimension on a given cube and give us a new subcube. Consider the following diagram showing the dice operation:

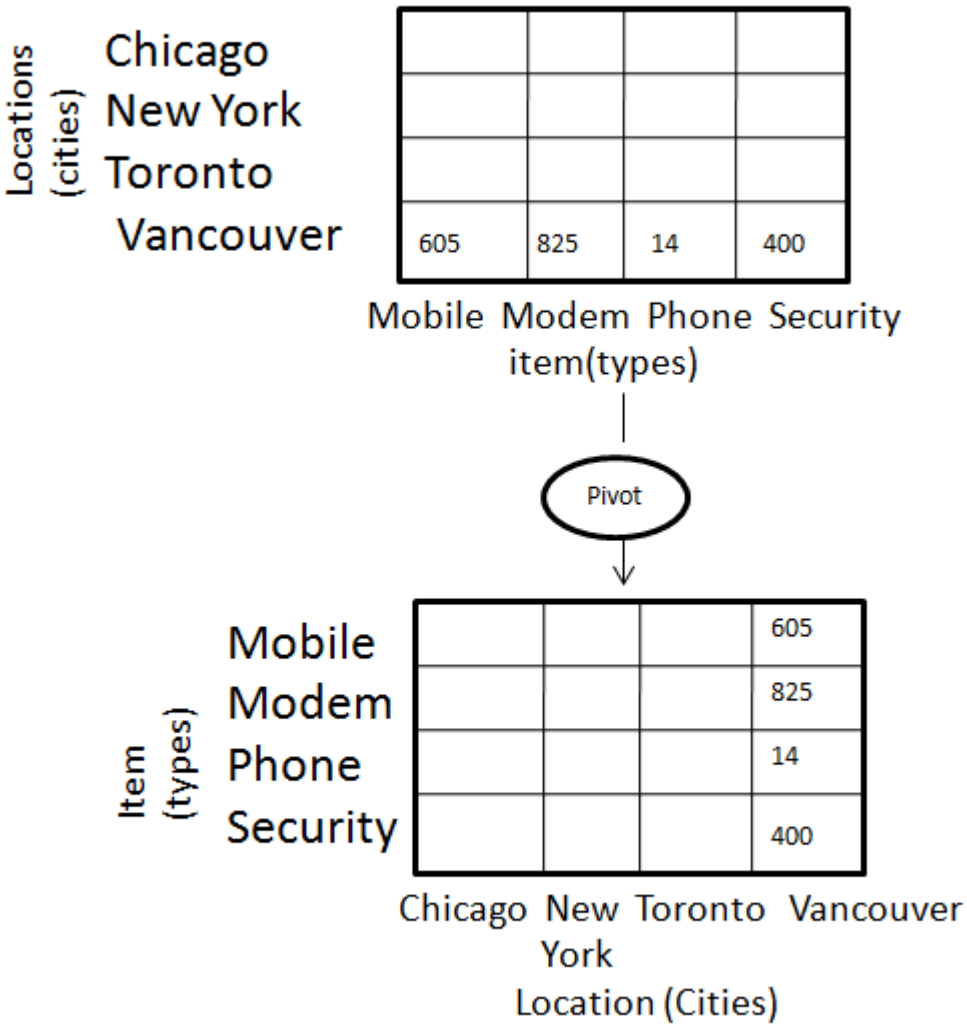


The dice operation on the cube based on the following selection criteria that involve three dimensions.

- (location = "Toronto" or "Vancouver")
- (time = "Q1" or "Q2")
- (item =" Mobile" or "Modem").

Pivot

The pivot operation is also known as rotation.It rotates the data axes in view in order to provide an alternative presentation of data.Consider the following diagram showing the pivot operation.



In this the item and location axes in 2-D slice are rotated.

OLAP vs OLTP

SN	Data Warehouse (OLAP)	Operational Database(OLTP)
1	This involves historical processing of information.	This involves day to day processing.

2	OLAP systems are used by knowledge workers such as executive, manager and analyst.	OLTP system are used by clerk, DBA, or database professionals.
3	This is used to analysis the business.	This is used to run the business.
4	It focuses on Information out.	It focuses on Data in.
5	This is based on Star Schema, Snowflake Schema and Fact Constellation Schema.	This is based on Entity Relationship Model.
6	It focuses on Information out.	This is application oriented.
7	This contains historical data.	This contains current data.
8	This provides summarized and consolidated data.	This provide primitive and highly detailed data.
9	This provide summarized and multidimensional view of data.	This provides detailed and flat relational view of data.
10	The number or users are in Hundreds.	The number of users are in thousands.
11	The number of records accessed are in millions.	The number of records accessed are in tens.
12	The database size is from 100GB to TB	The database size is from 100 MB to GB.
13	This are highly flexible.	This provide high performance.

A MULTIDIMENSIONAL DATA MODEL

A multidimensional database (MDB) is a type of database that is optimized for data warehouse and online analytical processing (OLAP) applications. Multidimensional databases are frequently created using input from existing relational databases. Whereas a relational database is typically accessed using a Structured Query Language (SQL) query, a multidimensional database allows a user to ask questions like "How many Aptiva's have been sold in Nebraska so far this year?" and similar questions related to summarizing business operations and trends. An OLAP application that accesses data from a multidimensional database is known as a MOLAP(multidimensional OLAP) application.

A multidimensional database - or a multidimensional database management system (MDDBMS) -implies the ability to rapidly process the data in the database so that answers can be generated quickly. A number of vendors provide products that use multidimensional databases. Approaches to how data is stored and the user interface vary.

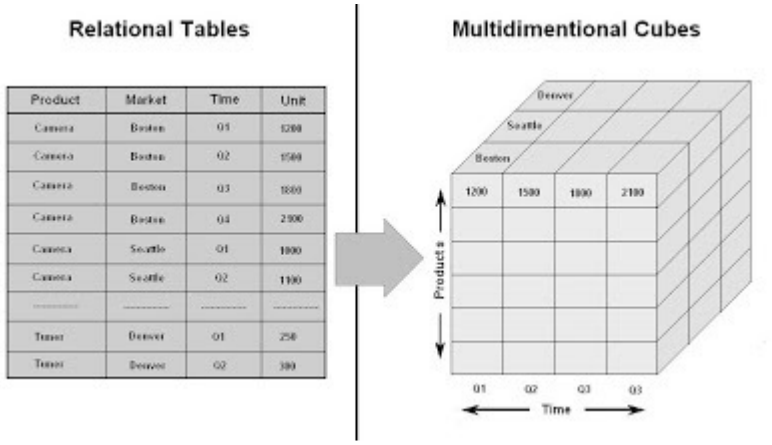
Conceptually, a multidimensional database uses the idea of a data cube to represent the dimensions of data available to a user. For example, "sales" could be viewed in the dimensions of product model, geography, time, or some additional dimension. In this case, "sales" is known as the *measure attribute* of the data cube and the other dimensions are seen as *feature attribute*s. Additionally, a database creator can define hierarchies and levels within a dimension (for example, state and city levels within a regional hierarchy).

Multidimensional data model is to view it as a cube. The cable at the left contains detailed sales data by product, market and time. The cube on the right associates sales number (unit sold) with dimensions-product type, market and time with the unit variables organized as cell in an array.

This cube can be expended to include another array-price-which can be associates with all or only some dimensions.

As number of dimensions increases number of cubes cell increase exponentially.

Dimensions are hierarchical in nature i.e. time dimension may contain hierarchies for years, quarters, months, weak and day. GEOGRAPHY may contain country, state, city etc.



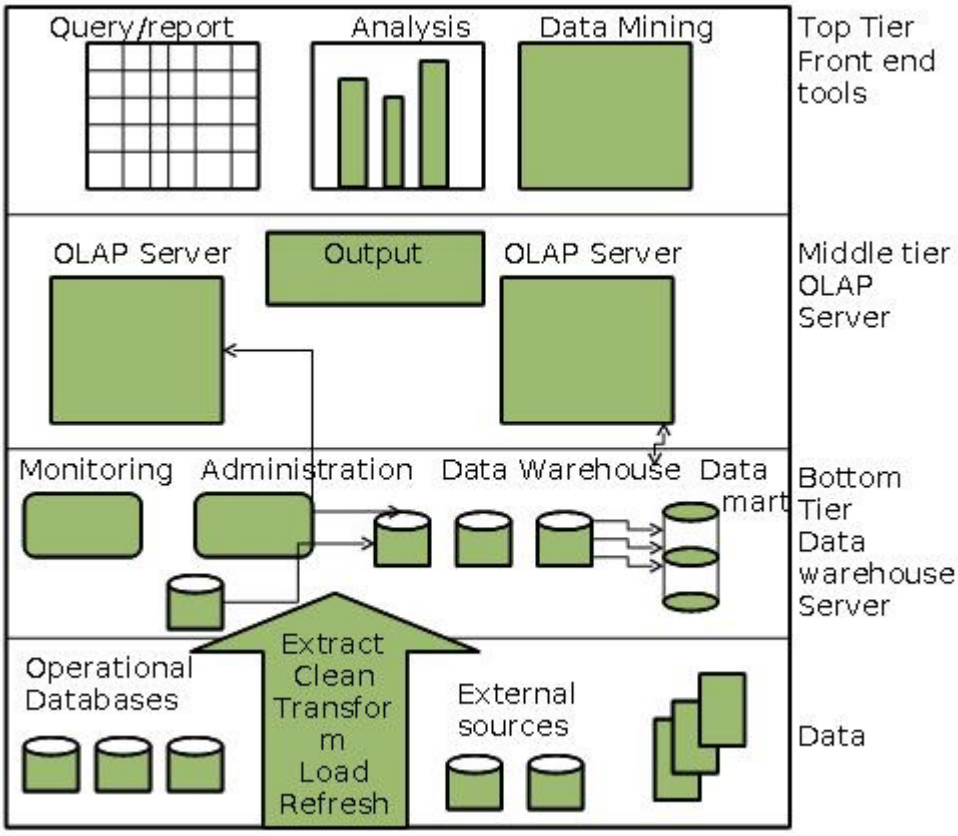
DATA WAREHOUSE ARCHITECTURE

Three-Tier Data Warehouse Architecture

Generally the data warehouses adopt the three-tier architecture. Following are the three tiers of data warehouse architecture.

- **Bottom Tier** - The bottom tier of the architecture is the data warehouse database server.It is the relational database system.We use the back end tools and utilities to feed data into bottom tier.these back end tools and utilities performs the Extract, Clean, Load, and refresh functions.
- **Middle Tier** - In the middle tier we have OLAp Server. the OLAP Server can be implemented in either of the following ways.
    - By relational OLAP (ROLAP), which is an extended relational database management system. The ROLAP maps the operations on multidimensional data to standard relational operations.
    - By Multidimensional OLAP (MOLAP) model, which directly implements multidimensional data and operations.
- **Top-Tier** - This tier is the front-end client layer. This layer hold the query tools and reporting tool, analysis tools and data mining tools.

Following diagram explains the Three-tier Architecture of Data warehouse:



DATA WAREHOUSE MODELS

From the perspective of data warehouse architecture we have the following data warehouse models:

- Virtual Warehouse
- Data mart
- Enterprise Warehouse

Virtual Warehouse

- The view over a operational data warehouse is known as virtual warehouse. It is easy to built the virtual warehouse.
- Building the virtual warehouse requires excess capacity on operational database servers.

Data Mart

- Data mart contains the subset of organisation-wide data.

- This subset of data is valuable to specific group of an organisation

Note: in other words we can say that data mart contains only that data which is specific to a particular group. For example the marketing data mart may contain only data related to item, customers and sales. The data mart are confined to subjects.

Points to remember about data marts

- window based or Unix/Linux based servers are used to implement data marts. They are implemented on low cost server.
- The implementation cycle of data mart is measured in short period of time i.e. in weeks rather than months or years.
- The life cycle of a data mart may be complex in long run if it's planning and design are not organization-wide.
- Data mart are small in size.
- Data mart are customized by department.
- The source of data mart is departmentally structured data warehouse.
- Data mart are flexible.

Enterprise Warehouse

- The enterprise warehouse collects all the information all the subjects spanning the entire organization
- This provide us the enterprise-wide data integration.
- This provide us the enterprise-wide data integration.
- The data is integrated from operational systems and external information providers.
- This information can vary from a few gigabytes to hundreds of gigabytes, terabytes or beyond.

DATA WARE HOUSE USAGE

A data warehouse maintains a copy of information from the source transaction systems. This architectural complexity provides the opportunity to :

- Congregate data from multiple sources into a single database so a single query engine can be used to present data.
- Mitigate the problem of database isolation level lock contention in transaction processing systems caused by attempts to run large, long running, analysis queries in transaction processing databases.
- Maintain data history, even if the source transaction systems do not.
- Integrate data from multiple source systems, enabling a central view across the enterprise. This benefit is always valuable, but particularly so when the organization has grown by merger.
- Improve data quality, by providing consistent codes and descriptions, flagging or even fixing bad data.
- Present the organization's information consistently.
- Provide a single common data model for all data of interest regardless of the data's source.
- Restructure the data so that it makes sense to the business users.
- Restructure the data so that it delivers excellent query performance, even for complex analytic queries, without impacting the operational systems.
- Add value to operational business applications, notably customer relationship management (CRM) systems.
- Making decision–support queries easier to write.

MINING FREQUENT PATTERNS, ASSOCIATION AND CORRELATION,
Data mining algorithms: Association rules

Motivation and terminology

1. Data mining perspective
    - Market basket analysis: looking for associations between items in the shopping cart.
    - Rule form: Body => Head [support, confidence]
    - Example: buys(x, "diapers") => buys(x, "beers") [0.5%, 60%]
2. Machine Learning approach: treat every possible combination of attribute values as a separate class, learn rules using the rest of attributes as input and then evaluate them for support and confidence. Problem: computationally intractable (too many classes and consequently, too many rules).
3. Basic terminology:
    1. Tuples are *transactions*, attribute-value pairs are *items.*
    2. *Association rule*: {A,B,C,D,...} => {E,F,G,...}, where A,B,C,D,E,F,G,... are items.
    3. *Confidence* (accuracy) of A => B : P(B|A) = (# of transactions containing both A and B) / (# of transactions containing A).
    4. *Support* (coverage) of A => B : P(A,B) = (# of transactions containing both A and B) / (total # of transactions)
    5. We looking for rules that exceed pre-defined support (*minimum support*) and have high confidence.

Example

1. Load the weather data in Weka (click on **Preprocess** and then on **Open file...** weather.nominal.arff). The data are shown below in tabular form.

outlook	temperature	humidity	windy	play
sunny	hot	high	false	no
sunny	hot	high	true	no
overcast	hot	high	false	yes
rainy	mild	high	false	yes
rainy	cool	normal	false	yes
rainy	cool	normal	true	no
overcast	cool	normal	true	yes
sunny	mild	high	false	no
sunny	cool	normal	false	yes
rainy	mild	normal	false	yes
sunny	mild	normal	true	yes
overcast	mild	high	true	yes
overcast	hot	normal	false	yes
rainy	mild	high	true	no

Click on **Associate** and then on **Start.** You get the following 10 association rules in **Associator output** window:

1. humidity=normal windy=FALSE 4 ==> play=yes 4 conf:(1)
2. temperature=cool 4 ==> humidity=normal 4 conf:(1)
3. outlook=overcast 4 ==> play=yes 4 conf:(1)
4. temperature=cool play=yes 3 ==> humidity=normal 3 conf:(1)

5. outlook=rainy windy=FALSE 3 ==> play=yes 3 conf:(1)
6. outlook=rainy play=yes 3 ==> windy=FALSE 3 conf:(1)
7. outlook=sunny humidity=high 3 ==> play=no 3 conf:(1)
8. outlook=sunny play=no 3 ==> humidity=high 3 conf:(1)
9. temperature=cool windy=FALSE 2 ==> humidity=normal play=yes 2 conf:(1)
10. temperature=cool humidity=normal windy=FALSE 2 ==> play=yes 2 conf:(1)

Basic idea: item sets

1. Item set: sets of all items in a rule (in both LHS and RHS).
2. Item sets for weather data: 12 one-item sets (3 values for outlook + 3 for temperature + 2 for humidity + 2 for windy + 2 for play), 47 two-item sets, 39 three-item sets, 6 four-item sets and 0 five-item sets (with minimum support of two).

One-item sets	Two-item sets	Three-item sets	Four-item sets
Outlook = Sunny (5) Temperature = Cool (4) ...	Outlook = Sunny Temperature = Mild (2) Outlook = Sunny Humidity = High (3) ...	Outlook = Sunny Temperature = Hot Humidity = High (2) Outlook = Sunny Humidity = High Windy = False (2) ...	Outlook = Sunny Temperature = Hot Humidity = High Play = No (2) Outlook = Rainy Temperature = Mild Windy = False Play = Yes (2) ...

3. Generating rules from item sets. Once all item sets with minimum support have been generated, we can turn them into rules.

- Item set: {Humidity = Normal, Windy = False, Play = Yes} (support 4).
- Rules: for a **n-item set there are (2^n -1) possible rules**, chose the ones with highest confidence. For example:

If Humidity = Normal and Windy = False then Play = Yes (4/4)
If Humidity = Normal and Play = Yes then Windy = False (4/6)
If Windy = False and Play = Yes then Humidity = Normal (4/6)
If Humidity = Normal then Windy = False and Play = Yes (4/7)
If Windy = False then Humidity = Normal and Play = Yes (4/8)
If Play = Yes then Humidity = Normal and Windy = False (4/9)
If True then Humidity = Normal and Windy = False and Play = Yes (4/12)

Generating item sets efficiently

1. Frequent item sets: item sets with the desired minimal support.
2. Observation: if {A,B} is a frequent item set, then both A and B are frequent item sets too. The inverse, however is not true (find a counter-example).
3. Basic idea (Apriori algorithm):
    - Find *all* n-item sets. Example (n=2): L2 = { {A,B}, {A,D}, {C,D}, {B,D} }
    - Generate (n+1)-item sets by merging n-item sets. L3 = { {A,B,C}, {A,C,D}, {A,B,D}, {B,C,D} }.
    - Test the newly generated (n+1)-items sets for minimum support.
        - Eliminate {A,B,C}, {A,C,D} and {B,C,D} because they contain non-frequent 2-item sets (which ones?).
        - Test the remaining item sets for minimal support by counting their occurrences in data.
    - Increment n and continue until no more frequent item sets can be generated.
    - Test step uses a *hash table* with all n-item sets: remove an item from the (n+1)-item set and check if it is in the hash table.

Generating rules efficiently

1. Brute-force method (for small item sets):
    - Generate all possible subsets of an item sets, excluding the empty set (2^n - 1) and use them as rule consequents (the remaining items form the antecedents).
    - Compute the confidence: divide the support of the item set by the support of the antecedent (get it from the hash table).
    - Select rules with high confidence (using a threshold).
2. Better way: iterative rule generation within minimal accuracy.
    - Observation: if an n-consequent rule holds then all corresponding (n-1)-consequent rules hold as well.
    - Algorithm: generate n-consequent candidate rules from (n-1)-consequent rules (similar to the algorithm for the item sets).
3. Weka's approach (default settings for Apriori): generate best 10 rules. Begin with a minimum support 100% and decrease this in steps of 5%. Stop when generate 10 rules or the support falls below 10%. The minimum confidence is 90%.

Advanced association rules

1. *Multi-level* association rules: using concept hierarchies.
    - Example: no frequent item sets.

A	B	C	D	...
1	0	1	0	...

0	1	0	1	...
...	...	...	...	...

- Assume now that A and B are children of A&B, and C and D are children of C&D in concept hierarchies. Assume also that A&B and C&D aggregate the values for their children. Then {A&B, C&D} will be a frequent item set with support 2.

2. Approaches to mining multi-level association rules:

- Using uniform support: same minimum support for all levels in the hierarchies: top-down strategy.
- Using reduced minimum support at lower levels: various approaches to define the minimum support at lower levels.

3. Interpretation of association rules:

- *Single-dimensional* rules: single predicate. Example: buys(x, "diapers") => buys(x, "beers"). Create a table with as many columns as possible values for the predicate. Consider these values as binary attributes (0,1) and when creating the item sets ignore the 0's.

diapers	beers	milk	bread	...
1	1	0	1	...
1	1	1	0	...
...	...	...	...	...

- *Multidimensional* association rules: multiple predicates. Example: age(x, 20) and buys(x, computer) => buys(x, computer_games). Mixed-type attributes. Problem: the algorithms discussed so far cannot handle numeric attributes.

age	computer	computer_games	...
20	1	1	...
35	1	0	...

...	...	...	...

4. Handling numeric attributes.

   - Static discretization: discretization based on predefined ranges.
   - Discretization based on the distribution of data: binning. Problem: grouping together very distant values.
   - Distance-based association rules:
     - cluster values by distance to generate clusters (intervals or groups of nominal values).
     - search for frequent cluster sets.
   - Approximate Association Rule Mining. Read the paper by Nayak and Cook.

Correlation analysis

1. High support and high confidence rules are not necessarily interesting. Example:
   - Assume that A occurs in 60% of the transactions, B - in 75% and both A and B - in 40%.
   - Then the association A => B has support 40% and confidence 66%.
   - However, P(B)=75%, higher than P(B|A)=66%.
   - In fact, A and B are negatively correlated, corr(A,B)=0.4/(0.6*0.75)=0.89<1
2. Support-confidence framework: an estimate of the *conditional probability* of B given A. We need a measure for the certainty of the implication A => B, that is, whether A implies B and to what extend.
3. Correlation between occurrences of A and B:
   - corr(A,B) = P(A,B)/(P(A)P(B))
   - corr(A,B)<1 => A and B are negatively correlated.
   - corr(A,B)>1 => A and B are positively correlated.
   - corr(A,B)=1 => A and B are independent.
4. Contingency table:

	outlook=sunny	outlook<>sunny	Row total
play=yes	2	7	9
play=no	3	2	5
Column total	5	9	14

   - if outlook=sunny then play=yes [support=14%, confidence=40%].

- corr(outlook=sunny,play=yes) = (2/14)/[(5/14)*(9/14)] = 0.62 < 1 => negaive correlation.
- if outlook=sunny then play=no [support=21%, confidence=60%].
- corr(outlook=sunny,play=no) = (3/14)/[(5/14)*(5/14)] = 1.68 > 1 => positive correlation.

EFFICIENT AND SCALABLE FREQUENT ITEM SET MINING METHODS

when we know what a frequent item set is, let's list down 2 major properties that will help us later on in defining algorithms to find the frequent item sets:

- Every subset of a frequent item set is also frequent. Also known as Apriori Property or Downward Closure Property, this rule essentially says that we don't need to find the count of an item set, if all its subsets are not frequent. This is made possible because of the anti-monotone property of support measure – the support for an item set never exceeds the support for its subsets. Stay tuned for this.
- If we divide the entire database in several partitions, then an item set can be frequent only if it is frequent in at least one partition. Bear in mind that the support of an item set is actually a percentage and if this minimum percentage requirement is not met for at least one individual partitions, it will not be met for the whole database. This property enables us to apply divide and conquer type of algorithms. Again, stay tuned for this too.

when we know what a frequent item set is, let's list down 2 major properties that will help us later on in defining algorithms to find the frequent item sets:

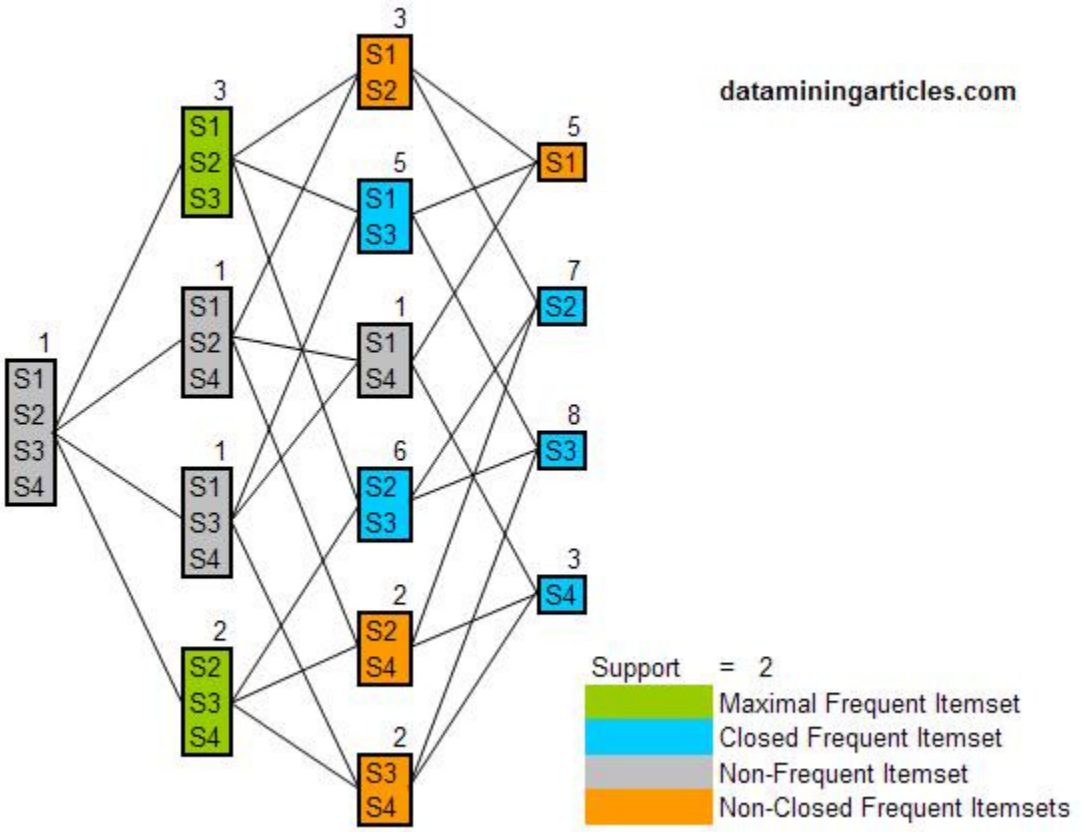
- Every subset of a frequent item set is also frequent. Also known as Apriori Property or Downward Closure Property, this rule essentially says that we don't need to find the count of an item set, if all its subsets are not frequent. This is made possible because of the anti-monotone property of support measure – the support for an item set never exceeds the support for its subsets. Stay tuned for this.
- If we divide the entire database in several partitions, then an item set can be frequent only if it is frequent in at least one partition. Bear in mind that the support of an item set is actually a percentage and if this minimum percentage requirement is not met for at least one individual partitions, it will not be met for the whole database. This property enables us to apply divide and conquer type of algorithms. Again, stay tuned for this too.

Maximal Frequent Item sets

Well, setting up a support percentage for an itermset, solved only a part of the problem. Now at least we know what we want. We know how frequent an item set should be to become worth considering it. But the toughest part is still unsolved. In order to find a frequent item set we have to go through all the sub-item sets which themselves are frequent due to the Downward Closure Property mentioned above. So we unavoidably generate an exponential number of sub patterns that we might not really need. Let's go back to our example and say that we want to find all the frequent item sets that have a support of 30%. We take advantage of our very small dataset and observe that (S1, S2, S3) is present in 3 contracts: C7, C9, C10 – that means the item set is present in 30% of contracts and hence it is frequent. And the only one superset is (S1, S2, S3, S4) which is not frequent. But what about all the subsets? There are 6 subsets that are obviously present in at least 30% of the contracts so they are frequent too, and don't need them (well, we don't need them now, we'll see that all the subsets are very important to determine the associations). But we have to go through all of them to get to the maximum number of items that form a frequent item set. This procedure is very time consuming because the search space is huge. The presence of a frequent item set of length k implies the presence of 2`k -2 additional frequent item sets. So, wouldn't we be better off if we could consider only the frequent item set that has the maximum number of items bypassing all the sub-item sets? This is how the Maximal Frequent Item set was invented. The definition says that an item set is maximal frequent if none of its immediate supersets is frequent. In our example (S1, S2, S3) is maximal frequent because the only one superset is not frequent.

Closed Frequent Item sets

The only one downside of a maximal frequent item set is that, even though we know that all the sub-item sets are frequent, we don't know the actual support of those sub-item sets. And we'll see how important this is when we'll try to find the association rules within the item sets. Keep that in mind for now and let's think of how to get all the frequent item sets that have the same support with all their subsets. This is how the Closed Frequent Item sets came into picture: an item set is closed if none of its immediate supersets has the same support as the item set. Finding these closed frequent item sets can be of a great



help to purge a lot of item sets that are not needed and to find, as I said above, the right associations rules. That's why a lot of researchers banged their heads against the walls to find a solution for that. But it happened only in 1999 when 4 French researchers came up with a very acclaimed article (Discovering Frequent Closed Item sets for Association Rules by Nicolas Pasquier, Yves Bastide, Rafik Taouil, and Lotfi Lakhal) where they proposed to mine only closed frequent item sets using an algorithm called A-CLOSE. In the following years a lot of other algorithms have been invented (CHARM, CLOSET, etc) that improved the performances of the initial algorithm. We'll talk about all these algorithms some other time. For now, let see some examples of Closed and Maximal Frequent Item sets. As you see in the graph, all individual items S1, S2, S3, S4 are frequent item sets because their support in greater than 2. But only 3 of them are closed because S1 has the superset (S1, S3) having the same support. So it contradicts the definition. The item sets (S1, S2, S3) and (S2, S3, S4) are frequent because they are present in at least 2 of the contracts, and they are maximal as well because their frequent superset – (S1, S2, S3, S4) is the only one superset and it's not frequent (so, obviously, it can't have the same support).

To conclude, we have to keep in mind the following important concepts:

- A frequent item set is one that occurs in at least a user-specific percentage of the database. That percentage is called support.

- An item set is closed if none of its immediate supersets has the same support as the item set.
- An item set is maximal frequent if none of its immediate supersets is frequent.

UNIT – III

CLASSIFICATION AND PREDICTION

Introduction and issues

Databases are rich with hidden information that can be used for making intelligent business decisions. Classification and prediction are two forms of data analysis that can be used to extract models describing important data classes or to predict future data trends. Whereas classification predicts categorical labels, prediction models continuous-valued functions. For example ,a classification model may be built to predict the expenditures of potential customers on computer equipment given their income and occupation. Many classification and prediction methods have been proposed by researches in machine learning, expert systems, statistics, and neurobiology. Most algorithms are memory resident, typically assuming a small data size. Recent database mining research has built on such work, developing scalable classification and prediction techniques capable of handling large disk-resident data. These techniques often consider parallel and distributed processing. The basic techniques for data classification such as decision tree induction, Bayesian classification and Bayesian belief networks, and neural networks. The integration of data warehousing technology with classification is also discussed ,as well as association-based classification. other approaches to classification, such as k-nearest neighbor classifiers, case-based reasoning, genetic algorithms, rough sets, and fuzzy logic techniques are introduced. Methods for prediction, including linear, non-linear and generalized linear regression models, are briefly discussed. Where applicable ,you will learn of modifications, extensions, and optimizations to these techniques for their application to data classification and prediction for large databases, Data classification is a two-step process. In the first step ,a model is built describing a predetermined set of data classes or concepts. The data classification process:

(a) Learning :Training data are analyzed by a classification algorithm. Here, the class label attribute is credit_ rating ,and the learned model or classifier is represented in the form of classification rules.

(b) Classification: Test data are used to estimate the accuracy of the classification rules-if the accuracy is considered acceptable, the rules can be applied to the classification of new data tuples constructed by analyzing database tuples described by attributes. Each tuple is assumed to belong to a predefined class, as determined by one of the attributes, called the class label attribute. In the context of classification, data tuples are also referred to as samples, examples, or objects. The data tuples analyzed to build the model collectively form the training data set. The individual tuples making up the training set are referred to as training samples and are randomly selected from the sample population. Since the class label of each training sample is provided, this step is also known as supervised learning(i.e., the learning of the model is" supervised " in that it is told to which class each training sample belongs). It contrasts with unsupervised learning(or clustering), in which the class label of each training sample is not known, and the number or set of classes to be learned may not be known in advance. Typically, the learned

model is represented in the form of classification rules, decision trees, or mathematical formulae. For example, given a database of customer credit information, classification rules can be learned to identify customers as having either excellent or fair credit ratings The rules can be used to categorize future data samples ,as well as provide a better understanding of the database contents..

In the second step, the model is used for classification. First, the predictive accuracy of the model [or classifier] is estimated. The hold method is a simple technique that uses a test set of class-labeled samples. These samples are randomly selected and are independent of the training samples. The accuracy of a model on a given test set is the percentage of test set samples that are correctly classified by the model .For each test set sample, the know class label is compared with the learned model's class prediction for that sample .Note that if the accuracy of the model were estimate based on the training data set, this estimate could be optimistic since the learned model tends to over fit the data (that is, it may have incorporated some particular anomalies of the training data that are not present in the in the overall sample population) therefore, a test set is used.

If the accuracy of the model is considered acceptable, the model can be used to classify future data tuples or objects for which the class label is not known. (Such data are also referred to in the machine learning literature as "unknown" or "previously unseen" data.) For example, the classification rules learned from the analysis of data from existing customers can be used to predict the credit rating of new or future (i.e., previously unseen) customers.

Prediction can be viewed as the construction and use of a model to assess the class of an unlabeled sample, or to assess the value or value ranges of an attribute that a given sample is likely to have. In this view, classification and regression are the two major types of prediction problems, where classification is used to predict discrete or nominal values, while regression is used to predict continuous or ordered values. In our view, however, we refer to the use of prediction to predict class labels as classification, accurate use of prediction to predict continuous values (e.g., using regression techniques) as prediction. This view is commonly accepted in data mining. Classification and prediction have numerous applications including credit approval, medical diagnosis, performance prediction, and selective marketing.

Example: Suppose that we have a database of customers on the ABCompany mailing list.

The mailing list is used to send out promotional literature describing new products and upcoming price discounts. The database describes attributes of the customers, such as their name, age, income, occupation, and credit rating. The customers can be classified as to whether or not they have purchased a computer at ABC company. Suppose that new customers are added to the database and that you would like to notify these customers of an upcoming computer sale. To send out promotional literature to every new customer in the database can be quite costly. A more cost efficient method would be to target only those new customers who are likely to Purchase a new computer. A classification model can be constructed and used for this Purpose.

Suppose instead that you would like to predict the number of major purchases that a customer will make at ABCompany during a fiscal year. Since the predicted value here is ordered, a prediction model can be constructed for this purpose.

PREPARING THE DATA FOR CLASSIFICATION AND PREDICTION

The following preprocessing steps may be applied to the data in order to help improve the accuracy, efficiency, and scalability of the classification or prediction process.

Data cleaning: This refers to the preprocessing of data in order to remove or reduce noise (by applying smoothing techniques, for example) and the treatment of missing values [e.g by replacing a missing value with the most commonly occurring value for that attribute, or with 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.

Relevance analysis: Many of the attributes in the data may be irrelevant to the classification or prediction task. For example, data recording the day of the week on which a bank loan application was filed is unlikely to be relevant to the success of the application. Furthermore, other attributes may be redundant. Hence, relevance analysis May be performed on the data with the aim of removing any irrelevant or redundant attributes from the learning process. In machine learning, this step is known as feature Selection. Including such attributes may otherwise slow down, and possibly mislead, the Learning step. Ideally, the time spent on relevance analysis, when added to the time spent on learning from the resulting "reduced" feature subset, should be less than the time that would have been sent on learning from the original set of features. Hence, such Analysis can help improve classification efficiency and scalability.

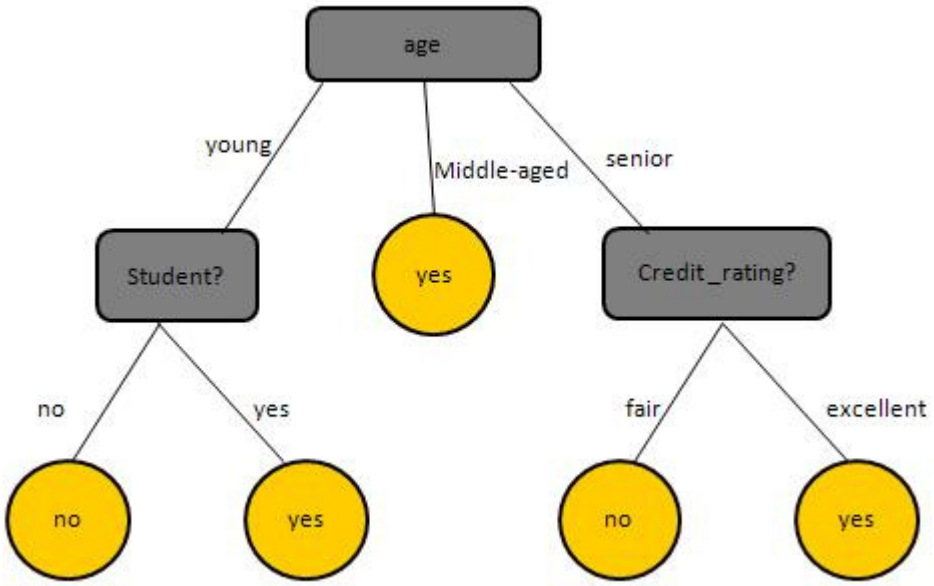
Data transformation: The data can be generalized to higher-level concepts. Concept Hierarchies may be used for this purpose. This is particularly useful for continuous valued attributes. For example, numeric values for the attribute income may be generalized to discrete ranges such as low, medium, and high. Similarly, nominal-valued attributes, like street, can be generalized to higher-level concepts, like city, Since Generalization compresses the original training data, fewer input/output operations may Be involved during learning. The data may also be normalized, 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 – 1.0 to 1.0, or 0.0 to 1.0. In methods that use distance measurements, for Example, this would prevent attributes with initially large ranges (like ,say, income) From outweighing attributes with initially smaller ranges (such as binary attributes).

CLASSIFICATION BY DECISION TREE INDUCTION

Introduction

The decision tree is a structure that includes root node, branch and leaf node. Each internal node denotes a test on attribute, each branch denotes the outcome of test and each leaf node holds the class label. The topmost node in the tree is the root node.

The following decision tree is for concept buy_computer, that indicates whether a customer at a company is likely to buy a computer or not. Each internal node represents the test on the attribute. Each leaf node represents a class.



Advantages of Decision Tree

- It does not require any domain knowledge.
- It is easy to assimilate by human.
- Learning and classification steps of decision tree are simple and fast.

Decision Tree Induction Algorithm

A machine researcher named J. Ross Quinlan in 1980 developed a decision tree algorithm. This Decision Tree Algorithm is known as ID3(Iterative Dichotomiser). Later he gave C4.5 which was 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
  // partition the tuples and grow subtrees for each partition
  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 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:

- **Prepruning** - The tree is pruned by halting its construction early.
- **Postpruning** - This approach removes subtree form fully grown tree.

Cost Complexity

The cost complexity is measured by following two parameters:

- Number of leaves in the tree
- Error rate of the tree

BAYESIAN CLASSIFICATION (CLASSIFICATION BY BACK PROPAGATION,)
INTRODUCTION

Bayesian classification is based on Baye's Theorem. Bayesian classifiers are the statistical classifiers. Bayesian classifier are able to predict class membership probabilities such as the probability that a given tuple belongs to a particular class.

Baye's Theorem

Baye's Theorem is named after Thomas Bayes. There are two types of probability as follows:

- Posterior Probability [P(H/X)]

- Prior Probability [P(H)]

Where, X is data tuple and H is some hypothesis.

According to Baye's Theorem

P(H/X)= P(X/H)P(H) / P(X)

Bayesian Belief Network

- Bayesian Belief Network specify joint conditional probability distributions
- Bayesian Networks and Probabilistic Network are known as belief network.
- Bayesian Belief Network allows class conditional independencies to be defined between subsets of variables.
- Bayesian Belief Network provide a graphical model of causal relationship on which learning can be performed.

We can use the trained Bayesian Network for classification. Following are the names with which the Bayesian Belief are also known:

- Belief networks
- Bayesian networks
- Probabilistic networks

There are two components to define Bayesian Belief Network:

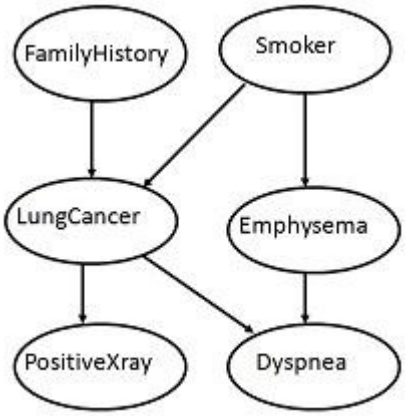
- Directed acyclic graph
- A set of conditional probability tables

Directed Acyclic Graph

- Each node in directed acyclic graph is represents a random variable.
- These variable may be discrete or continuous valued.
- These variable may corresponds to actual attribute given in 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 woth noting that the variable PositiveXRay is independent of whether the patient has a family history of lung cancer or is a smoker, given that we know the patient has lung cancer.

Set of Conditional probability table representation:

The conditional probability table for the values of the variable LungCancer (LC) showing each possible combination of the values of its parent nodes, FamilyHistory (FH) and Smoker (S).

	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

RULE BASED CLASSIFICATION

IF-THEN Rules

Rule-based classifier make use of set of IF-THEN rules for classification. We can express the 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.
- In the antecedent part the condition consist of one or more attribute tests and these tests are logically ANDed.
- The consequent part consist class prediction.

Note:

We can also write rule R1 as follows:

R1: (age = youth) ^ (student = yes))(buys computer = yes)

If the condition holds the 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 decision tree. Points to remember to extract rule from a decision tree:

- One rule is created for each path from the root to the leaf node.
- To from the rule antecedent each splitting criterion is logically ANDed.
- 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 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 are 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 pruning set. Hence, if the FOIL_Prune value is higher for the pruned version of R, then we prune R.

LAZY LEARNERS

lazy learning is a learning method in which generalization beyond the training data is delayed until a query is made to the system, as opposed to in eager learning, where the system tries to generalize the training data before receiving queries.

The main advantage gained in employing a lazy learning method, such as Case based reasoning, is that the target function will be approximated locally, such as in the k-nearest neighbor algorithm. Because the target function is approximated locally for each query to the system, lazy learning systems can simultaneously solve multiple problems and deal successfully with changes in the problem domain.

The disadvantages with lazy learning include the large space requirement to store the entire training dataset. Particularly noisy training data increases the case base unnecessarily, because no abstraction is made during the training phase. Another disadvantage is that lazy learning methods are usually slower to evaluate, though this is coupled with a faster training phase.

PREDICTION: ACCURACY AND ERROR MEASURES, EVALUATING THE ACCURACY OF A CLASSIFIER OR PREDICTOR.

Numerical Models (Regressions)

Mean Squared Error (MSE) is by far the most common measure of numerical model performance. It is simply the average of the squares of the differences between the predicted and actual values. It is a reasonably good measure of performance, though it could be argued that it overemphasizes the importance of larger errors. Many modeling procedures directly minimize the MSE.

Mean Absolute Error (MAE) is similar to the Mean Squared Error, but it uses absolute values instead of squaring. This measure is not as popular as MSE, though its meaning is more intuitive (the "average error").

Bias is the average of the differences between the predicted and actual values. With this measure, positive errors cancel out negative ones. Bias is intended to assess how much higher or lower predictions are, on average, than actual values.

Mean Absolute Percent Error (MAPE) is the average of the absolute errors, as a percentage of the actual values. This is a relative measure of error, which is useful when larger errors are more acceptable on larger actual values.

Classifiers

Classifiers come in two basic varieties: those which produce class outputs, and those which produce probabilities of classes.

Classifiers: Class Output

Accuracy is the proportion of the time that the predicted class equals the actual class, usually expressed as a percentage. It's meaning is straightforward, but may obscure important differences in costs associated with different errors. The classic example of such costs is the medical diagnostic situation, in which one can err be either: 1. keeping a healthy patient in the hospital (low cost), or 2. sending home a sick patient (very high cost).

Classifiers: Probability Output

These classifiers need to be checked for both the accuracy of their probabilities (Do cases predicted to have a 5% (30%, 80%, etc.) probability really belong to the target class 5% (30%, 80%, etc.) of the time?) and their ability to separate the classes in question.

Accuracy can be measured using many of the same metrics used to evaluate numerical models (MSE, MAE, etc.). One interesting alternative which is specific to classification, the *informational loss*, is based on information theory and is described in *Data Mining* by Witten and Frank (ISBN 1-55860-552-5).

Some applications (as in marketing) are focused on how many items from the target class can be identified in the best so-many percent of the population. If for example, one only has the resources to mail marketing literature to 10% of the customer file, the ideal would be to pack as many actual respondents as possible into that best 10%. The mirror situation is typified by lenders who wish to cram as many bad loans as possible into the worst 10% of their file. Probably the most popular measure of class separation at present in the literature is the **Area Under the ROC Curve** (AUC or AUROC), which is like measuring separation across the whole spectrum.

CLUSTER ANALYSIS: TYPES OF DATA IN CLUSTER ANALYSIS, A CATEGORIZATION OF MAJOR CLUSTERING METHODS, PARTITIONING METHODS.

What is Cluster?

Cluster is a group of objects that belong to the same class. In other words the similar object are grouped in one cluster and dissimilar are grouped in other cluster.

What is Clustering?

Clustering is the process of making group of abstract objects into classes of similar objects.

Points to Remember

- A cluster of data objects can be treated as a one group.
- While doing the cluster analysis, we first partition the set of data into groups based on data similarity and then assign the label to the groups.
- The main advantage of Clustering over classification is that, It is adaptable to changes and help single out useful features that distinguished different groups.

Applications of Cluster Analysis

- Clustering Analysis is broadly used in many applications such as market research, pattern recognition, data analysis, and image processing.
- Clustering can also help marketers discover distinct groups in their customer basis. And they can characterize their customer groups based on purchasing patterns.
- In field of 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 also helps in identification of areas of similar land use in an earth observation database. It also helps in the identification of groups of houses in a city according house type, value, geographic location.
- Clustering also helps in classifying documents on the web for information discovery.
- Clustering is also used in outlier detection applications such as detection of credit card fraud.
- As a data mining function Cluster Analysis serve as a tool to gain insight into the distribution of data to observe characteristics of each cluster.

Requirements of Clustering in Data Mining

Here is the typical requirements of clustering in data mining:

- **Scalability** - We need highly scalable clustering algorithms to deal with large databases.
- **Ability to deal with different kind of attributes** - Algorithms should be capable to be applied on any kind of data such as interval based (numerical) data, categorical, binary data.
- **Discovery of clusters with attribute shape** - The clustering algorithm should be capable of detect cluster of arbitrary shape. The should not be bounded to only distance measures that tend to find spherical cluster of small size.
- **High dimensionality** - The clustering algorithm should not only be able to handle low-dimensional data but also the high dimensional space.
- **Ability to deal with noisy data** - Databases contain noisy, missing or erroneous data. Some algorithms are sensitive to such data and may lead to poor quality clusters.
- **Interpretability** - The clustering results should be interpretable, comprehensible and usable.

Clustering Methods

The clustering methods can be classified into following categories:

- Partitioning Method
- Hierarchical Method
- Density-based Method
- Grid-Based Method
- Model-Based Method
- Constraint-based Method

Partitioning Method

Suppose we are given a database of n objects, the partitioning method construct k partition of data. Each partition will represents a cluster and k≤n. It means that it will classify the data into k groups, which satisfy the following requirements:

- Each group contain at least one object.
- Each object must belong to exactly one group.

Points to remember:

- For a given number of partitions (say k), the partitioning method will create an initial partitioning.
- Then it uses the iterative relocation technique to improve the partitioning by moving objects from one group to other.

Hierarchical Methods

This method create the hierarchical decomposition of the given set of data objects. We can classify Hierarchical method on basis of how the hierarchical decomposition is formed as follows:

- Agglomerative Approach
- Divisive Approach

Agglomerative Approach

This approach is also known as bottom-up approach. In this we start with each object forming a separate group. It keeps on merging the objects or groups that are close to one another. It keep on doing so until all of the groups are merged into one or until the termination condition holds.

Divisive Approach

This approach is also known as top-down approach. In this we start with all of the objects in the same cluster. In the continuous iteration, a cluster is split up into smaller clusters. It is down until each object in one cluster or the termination condition holds.

Disadvantage

This method is rigid i.e. once merge or split is done, It can never be undone.

Approaches to improve quality of Hierarchical clustering

Here is the two approaches that are used to improve quality of hierarchical clustering:

- Perform careful analysis of object linkages at each hierarchical partitioning.
- Integrate hierarchical agglomeration by first using a hierarchical agglomerative algorithm to group objects into micro clusters, and then performing macro clustering on the micro clusters.

Density-based Method

This method is based on the notion of density. The basic idea is to continue growing the given cluster as long as the density in the neighbourhood exceeds some threshold i.e. for each data point within a given cluster, the radius of a given cluster has to contain at least a minimum number of points.

Grid-based Method

In this the objects together from a grid. The object space is quantized into finite number of cells that form a grid structure.

Advantage

- The major advantage of this method is fast processing time.
- It is dependent only on the number of cells in each dimension in the quantized space.

Model-based methods

In this method a model is hypothesize for each cluster and find the best fit of data to the given model. This method locate the clusters by clustering the density function. This reflects spatial distribution of the data points.

This method also serve a way of automatically determining number of clusters based on standard statistics, taking outlier or noise into account. It therefore yield robust clustering methods.

Constraint-based Method

In this method the clustering is performed by incorporation of user or application oriented constraints. The constraint refers to the user expectation or the properties of desired clustering results. The constraint give us the interactive way of communication with the clustering process. The constraint can be specified by the user or the application requirement.

UNIT IV

MULTIDIMENSIONAL ANALYSIS AND DESCRIPTIVE MINING OF COMPLEX DATA OBJECTS

A major limitation of many commercial data warehouse and OLAP tools for multidimensional database analysis is their restriction on the allowable data types for dimensions and measures. Most data cube implementations confine dimensions to nonnumeric data and measures to simple aggregated values. To introduce data mining and multidimensional data analysis for complex objects, this section examines how to perform generalization on complex structured objects and construct object cubes for OLAP and mining in object databases. The storage and access of complex structured data have been studied in object-relational and object-oriented database systems. These systems organize a large set of complex data objects into classes, which are in turn organized into class/subclass hierarchies. Each object in a class is associated with

1) An object-identifier
2) A set of attributes that may contain sophisticated data structures, set- or list-valued data, class composition and hierarchies, multimedia data and so on &
3) A set of methods that specify the computational routines or rules associated with the object class.

An important feature of object-relational and object-oriented databases is their capability of storing, accessing and modeling complex structure-valued data, such as set-valued and list-valued data end data with nested structures.

Let's start by having a look at the generalization of set-valued and list-valued attributes.

A set-valued attribute may be homogenous or heterogeneous type. Typically, set-valued data can be generalized by

(1) Generalization of each value in the set into Us corresponding higher-level concepts or
(2) Derivation of general behavior of the set, such as the number of elements in the set, or the weighted average for numerical data. Moreover, applying different generalization operators to explore alternative generalization paths can perform generalization. In this case the result of generalization is a heterogeneous set.

A set-valued attribute may be generalized into a set-valued or single-valued attribute; a single valued attribute may be generalized into a set valued attribute if the values form a lattice or "hierarchy" or the generalization follows different paths. Further generalizations on such a generalized set-valued attribute should follow the generalization path of each value in the set.

A list-valued or sequence valued attribute can be generalized in a manner similar to that for set valued attributes except that the order of the elements in the sequence should be observed in the generalization. Each value in the list can be generalized into its corresponding higher level concept. Alternatively a list can be generalized according to its general behavior, such as the length of the list, the type of list elements, the value range, the weighted average value for numerical data, or by dropping
unimportant elements in the list. A list may be generalized into a list, set, or a single value.

A complex structure-valued attribute may contain sets, tuples, lists, trees, records and so on, and their combinations where one structure may be nested in another at any level. In general, a structure valued attribute can be generalized in several ways, such as
(1) Generalizing each attribute in the structure while maintaining the shape of the structure,
(2) Flattening the structure and generalizing the flattened structure,
(3) Summarizing the low-level structure by high-level concepts or aggregation &
(4) Returning the type or an overview of the structure.

MINING SPATIAL DATABASE

A spatial database stores a large amount of space-related data, such as maps, preprocessed remote sensing or medical imaging data, and VLSI chip layout data. Spatial databases have many features distinguishing them from relational databases. They carry topological and/or distance information, usually organized by sophisticated, multidimensional spatial indexing structures that are accessed by spatial data access methods and often require spatial reasoning, geometric computation, and spatial knowledge representation techniques. Spatial data mining refers to the extraction of knowledge, spatial relationships, or other interesting patterns not explicitly stored in spatial databases. Such mining demands an integration of data mining with spatial database technologies. It can be used for understanding spatial data, discovering spatial relationships and relationships between spatial and non spatial data, constructing spatial knowledge bases, reorganizing spatial databases, and optimizing spatial queries. It is expected to have wide applications in geographic information systems, remarketing, remote sensing, image database exploration, medical imaging, navigation, traffic control, environmental studies, and many other areas where spatial data are used. A crucial challenge to spatial data mining is the exploration of efficient spatial data mining techniques due to the huge amount of spatial data and the complexity of spatial data types and spatial access methods.
Statistical spatial data analysis has been a popular approach to analyzing spatial data. The approach handles numerical data well and usually proposes realistic models of spatial phenomena.
However, it typically assumes statistical independence among the spatially distributed data, although in reality, spatial objects are often inter-related. Moreover, experts having a fair amount of domain knowledge and statistical expertise can only perform most statistical modeling. Furthermore, statistical methods do not work well with symbolic values, or incomplete or inconclusive data, and are computationally expensive in large databases. Spatial data allows the extension of traditional spatial analysis methods by placing minimum emphasis on efficiency, scalability, cooperation spatial Data Cube Construction and Spatial OLAP
As with relational data, we can integrate spatial data to construct a data warehouse that facilitates spatial data mining. A spatial data warehouse is a subject-oriented, integrated, time-variant, and nonvolatile collection of both spatial and non-spatial data in support of spatial data mining and spatial-data-related decision-making processes. Let's have a look at the following example,
Example There are about 3000 weather probes distributed in British Columbia (BC), each recording daily temperature and precipitation for a designated small area and transmitting signals to a provincial weather station. With a spatial data warehouse that supports spatial OLAP, a user can view weather patterns on a map by month by region, and by different combinations of

temperature and precipitation, and can dynamically drill down or roll up along any dimension to explore desire patterns, such as "wet and hot regions in the Fraser Valley in Summer 1999".
There are several challenging issues regarding the construction and utilization of spatial data Warehouses. The first challenge is the integration of spatial data from heterogeneous sources and systems.
Spatial data are usually stored in different industry firms and government agencies using various data formats. Data formats are not only structure-specific (e.g., raster- vs. vector-based spatial data object oriented vs. relational models, different spatial storage and indexing structures, etc.), but also vendor specific (e.g., ESRI, MapInfo, Intergraph, etc.). There has been a great deal of work on the integration and exchange of heterogeneous spatial data, which has paved the way for spatial data integration and spatial data warehouse construction.
The second challenge is the realization of fast and flexible on-line analytic processing in spatial data warehouses. The star schema model is a good choice for modeling spatial data warehouses since it provided a concise and organized warehouse structure and facilitates OLAP operation However, in a spatial warehouse, both dimensions and measures may contain spatial components.
There are three types of dimensions in a spatial data cube:
• A non spatial dimension contains on l y non spatial data .Non spatial dimensions temperature and
precipitation can be constructed for the warehouse since each contains non spatial data whose generalizations are non spatial (such as "hot" for temperature. and "wet" for precipitation).
• A spatial-to-non spatial dimension is a dimension whose primitive-level data are spatial but whose generalization, starting at a certain high level, become non spatial. For example, the spatial dimension City relays geographic data fit the U.S. map. Suppose that the dimension's spatial representation of, say, Seattle is generalized to the string "pacific-northwest". Although "Pacific-northwest" a spatial concept, its representation is not spatial (since, in our example, it if string).It therefore plays the role of a non spatial dimension.
• A spatial-to-spatial dimension is a dimension whose primitive level and all its high-level generalized data are spatial. For example, the dimension equ_temperature_region contains spatial data, as do all of its generalizations, such as with regions covering 0-5 degrees (Celsius), 5-10 degrees, and so on.

We distinguish two types of measures in a spatial data cube.
A nonspatial data cube contains only non spatial dimensions and numerical measures. If a spatial data cube contains spatial dimensions but no spatial measures, its OLAP operations, such as drilling or pivoting, can be implemented in a manner similar to that for non spatial data cubes.
Of the three measures, area and count are numerical measures that can be computed similarly to that for non spatial data cubes; regions-map is a spatial measure that represents a collection of spatial pointers to the corresponding regions. Since different spatial OLAP operations result in different collections of spatial objects in region map, it is a major challenge to compute the merges of a large number of regions flexibly and dynamically. There are at least three possible choices in regard to the computation of spatial measures in spatial data cube construction.
Collect and store the corresponding spatial object pointers but do not perform pre computation of

spatial measures in the spatial data cube. This can be implemented by storing, in the corresponding cube cell,, a pointer to a collection of spatial object pointers, and invoking and performing the spatial merge (or other computation) of the corresponding spatial objects, when necessary, on- the-fly. This method is a good choice if only spatial display is required (i.e., no real spatial merge has to be performed), or if there are not many regions to be merged in any pointer collection (so that the on-line merge is not very costly), or if on-line spatial merge computation is fast (recently, some efficient spatial merge methods have been developed for fast spatial OLAP). Since OLAP results are often used for on-line spatial analysis and mining, it is still recommended to pre compute some of the spatially connected regions to speed up such
Analysis. Pre compute and store a rough approximation of the spatial measures in the spatial data cube. These choices good for a rough view or coarse estimation of spatial merge results under the assumption that it requires little storage space. For example, a minimum bounding rectangle (MBR), represented by two points, can be taken as a rough estimate of a merged region. Such a pre computed result is small and can be presented quickly to users. If higher precision is needed for specific cells, the application can either fetch pre computed high-quality results, if available, or compute them on-the-fly. Selectively pre compute some spatial measures in the spatial data cube. This can be a smart choice. The selection can be performed at the cuboids level, that is, either pre compute or store each set of merge able spatial regions for each cell of a selected cuboids, or pre compute none if the cuboids is not selected. Since a cuboids usually consists of a large number of spatial objects, it may involve Pre computation and storage of a large number of merge able spatial objects, some of which may be rarely used. Therefore, it is recommended to perform selection at a finer granularity level examining each group of merge able spatial objects in cuboids to determine whether such a merger should be pre computed.
Decision should be based on the utility (such as access frequency or access priority), sharability of merged regions, and the balanced overall cost of space and on-line computation.
With efficient implementation of spatial data cubes and spatial OLAP, generalization-based Descriptive spatial mining, such as spatial characterization and discrimination, can be performed Efficiently.

MULTIMEDIA DATABASE

Multimedia database is a kind of database like any other databases containing multimedia collections. Multimedia is defined as the combination of more than one media, they may be of two types --static and dynamic media. Text, graphics, and images are categorized as static media; on the other hand, objects like- animation, music, audio, speech, video are categorized as dynamic media. Graphic images may consist of clipart, photographs, logos, and custom drawings. Sound consists of voice narration, speech, music etc. Video data encompasses sound as well as photos. To manage these data multimedia database management system is essential. Multimedia database management system can be defined as a software system that manages a collection of multimedia data and provides access to users to query and retrieve multimedia objects. Generally, multimedia database contains text, image, animation, video, audio, movie sound etc. But, all data are stored in the database in binary form.

Why multimedia databases?

Following arguments will try to justify the requirements of multimedia database as explained below:

- Multimedia Database is capable of hand ling huge volume of multimedia objects which a general database fails to do effectively;
- Multimedia Database will help to create virtual museum;
- It will surely help to develop multimedia applications in various fields like teaching, medical sciences and libraries;
- Preserving decaying photographs, maps, films having got historical evidence or national importance;
- Using multimedia database, we can develop the excellent teaching packages;
- Helps multi-user operations.

Multimedia database: types

There are generally two types of multimedia databases:

- Linked Multimedia
- Databases and Embedded Multimedia Databases.

Linked multimedia databases

Multimedia database can be organized as a database of metadata. This metadata links to the actual data such as graphic, image, animation, audio, sound etc. These data may store on Hard Disc, CD-ROM, DVD or Online. In this database, multimedia elements are organized as image, audio/ MP3, video etc. In this multimedia database system, all data may be stored either on off-line i.e. CD-ROM, Hard Disc, DVD etc. or on Online. One great advantage of this type of atabase is that the size of database will be small due to the reason that multimedia elements are not embedded in the database, but only linked to it.

Embedded multimedia database

Embedded Multimedia Database implies that the database itself contains the multimedia objects as in the binary form in the database. The main advantage of such kind of database is that retrieval of data will be faster because of the reduced data access time. However, the size of the database will be very large.

Characteristics of MDBMS

A MDBMS (Multimedia Database Management System) can be characterized based on its objectives at the time of handling multimedia objects:

- Corresponding storage media
- Comprehensive search methods
- Device and format Independence Interface
- Simultaneous data access
- Management of large amount of data

- Relational consistency of Data Management
- Long Transaction

Multimedia database content

Multimedia Database generally holds the following multimedia components like--
text, graphics, animation, sounds, video etc.

Text

In multimedia applications, text is being used. Reason is that a longer text reading is difficult by the smaller screen resolution. At the same time, when a piece of information fails to communicate to others using other multimedia elements, text is mandatory. Text should be used only such cases where it able to eliminate potential information ambiguity.

Speech

Speech is continuous concept. Speech can introduce, give survey, stimulate and tell. Speech is ideals as an additional explanation of text.

Graphics

It is a very powerful multimedia component. The real strength of graphics is to maintain context. Graphics are discrete concepts. The user himself determines viewing moments and duration. In this way, graphics are very suitable for individual studying and analyzing of connections. The combination with text is good because both are discrete representations. Graphics approve more interpretation than the image and can be used better for the support of mental model.

Image

The image is very much related by its photorealistic representation to the concrete contents. User's mood can be influenced by images. In this case, the combination of image with sound will be very much effective.

Animation

Animation is also a component in multimedia database. It can be defined as the change in the characteristics of an object over a period of time. Animation files require more storage space than graphic files involving single image.

Sound

Sound as music or speech has a power to invoke emotions. Music can stimulate moods positively in reviving or relaxation of mind and body; whereas sound as noise helps to irritate people. The combination of sound with animation will really have a realistic effect on users.

Video

Video is the most powerful of all the multimedia components. It helps to portray the real world events. It will also help to grasp the more delicate and complicated situation/ ideas into minds.

MINING WORLD WIDE WEB

The World Wide Web contains the huge information such as hyperlink information, web page access info, education etc that provide rich source for data mining.

Challenges in Web Mining

The web poses great challenges for resource and knowledge discovery based on the following observations:

- The web is too huge. - The size of the web is very huge and rapidly increasing. This seems that the web is too huge for data warehousing and data mining.

- Complexity of Web pages. - The web pages do not have unifying structure. They are very complex as compared to traditional text document. There are huge amount of documents in digital library of web. These libraries are not arranged according in any particular sorted order.

- Web is dynamic information source. - The information on the web is rapidly updated. The data such as news, stock markets, weather, sports, shopping etc are regularly updated.

- Diversity of user communities. - The user community on the web is rapidly expanding. These users have different backgrounds, interests, and usage purposes. There are more than 100 million workstations that are connected to the Internet and still rapidly increasing.

- Relevancy of Information. - It is considered that a particular person is generally interested in only small portion of the web, while the rest of the portion of the web contains the information that is not relevant to the user and may swamp desired results.

Mining Web page layout structure

The basic structure of the web page is based on Document Object Model (DOM). The DOM structure refers to a tree like structure. In this structure the HTML tag in the page corresponds to a node in the DOM tree.We can segment the web page by using predefined tags in HTML. The HTML syntax is flexible therefore, the web pages do not follow the W3C specifications. Not following the specifications of W3C may cause error in DOM tree structure.

The DOM structure was initially introduced for presentation in the browser not for description of semantic structure of the web page. The DOM structure cannot correctly identify the semantic relationship between different parts of a web page.

Vision-based page segmentation (VIPS)

- The purpose of VIPS is to extract the semantic structure of a web page based on its visual presentation.

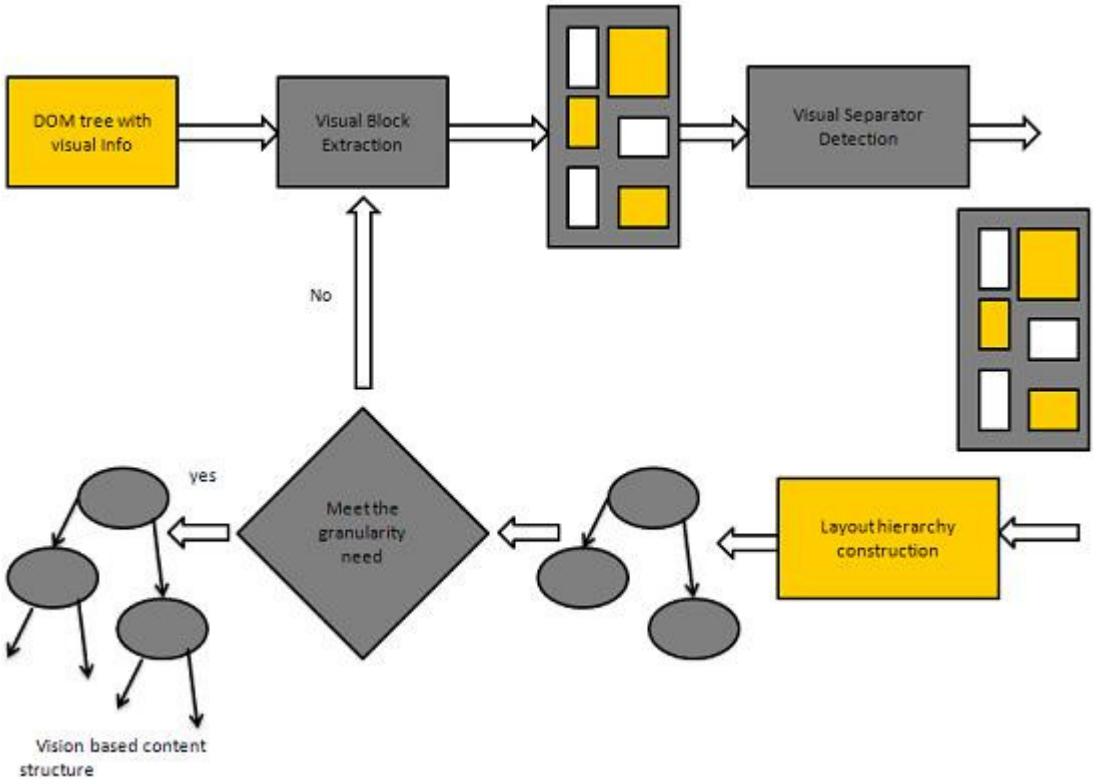
- Such a semantic structure corresponds to tree structure.In this tree each node corresponds to a block.

- A value is assigned to each node. This value is called Degree of Coherence. This value is assigned to indicate how coherent is the content in the block based on visual perception.

- The VIPS algorithm first extracts all the suitable blocks from the HTML DOM tree. After that it finds the separators between these blocks.

- The separators refer to the horizontal or vertical lines in a web page that visually cross with no blocks.
- The semantic of the web page is constructed on the basis of these blocks.

The following figure shows the procedure of VIPS algorithm:



APPLICATIONS AND TRENDS IN DATA MINING:
DATA MINING APPLICATIONS, DATA MINING SYSTEM PRODUCTS AND RESEARCH PROTOTYPES, SOCIAL IMPACT OF DATA MINING, TRENDS IN DATA MINING.

Data Mining is widely used in diverse areas. There are number of commercial data mining system available today yet there are many challenges in this field. In this tutorial we will applications and 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 the systematic data analysis and data mining. Here are the few typical cases:

- 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 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 increasing ease, availability and popularity of web.

The Data Mining in Retail Industry helps in identifying customer buying patterns and trends. That leads to improved quality of customer service and good customer retention and satisfaction. Here is the list of examples of data mining in retail industry:

- Design and Construction of data warehouses based on 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 examples for which data mining improve 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

Now a days we see that there is vast growth in field of biology such as genomics, proteomics, functional Genomics and biomedical research. Biological data mining is very important part of Bioinformatics. Following are the aspects in which Data mining contribute 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. There is large amount of data sets 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 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 availability of network resources. In this world of connectivity security has become the major issue. With increased usage of internet and availability of 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 are available. The new data mining systems and applications are being added to the previous systems. Also the efforts are being made towards standardization of data mining languages.

Choosing Data Mining System

Which data mining system to choose will depend on following features of Data Mining System:

- **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 Data Mining system with different operating systems. One data mining system may run on only on 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 refers to the data formats in which data mining system will operate. Some data mining system may work only on ASCII text files while other 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 system need to be coupled with database or the data warehouse systems. 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 as follows:

    - Row (Database size) Scalability - Data mining System is considered as row scalable when the number or rows are enlarged 10 times, It takes no more than the 10 times to execute the query.
    - Column (Dimension) Salability - Data mining system is considered as column scalable if the mining query execution time increases linearly with 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** - The graphical user interface which is easy to use and is required 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

Here is the list of trends in data mining that reflects pursuit of the challenges such as construction of integrated and interactive data mining environments, design of data mining languages:

- 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