SCSA3001	DATA MINING AND DATA WAREHOUSING	L	T	P	Credits	Total Marks
		3	0	0	3	100

COURSE OBJECTIVES

- Identify the scope and necessity of Data Mining & Warehousing for the society.
- Describe various Data Models and Design Methodologies of Data Warehousing destined to solve the root problems.
- To understand various Tools of Data Mining and their Techniques to solve the real time problems.
- To learn how to analyze the data, identify the problems, and choose the relevant algorithms to apply.
- To assess the Pros and Cons of various algorithms and analyze their behaviour on real datasets.

UNIT 1 DATA MINING **9 Hrs.**

Introduction - Steps in KDD - System Architecture – Types of data -Data mining functionalities - Classification of data mining systems - Integration of a data mining system with a data warehouse - Issues - Data Preprocessing - Data Mining Application

UNIT 2 DATA WAREHOUSING **9 Hrs.**

Data warehousing components - Building a data warehouse - Multi Dimensional Data Model - OLAP Operation in the Multi-Dimensional Model - Three Tier Data Warehouse Architecture - Schemas for Multi-dimensional data Model - Online Analytical Processing (OLAP) - OLAP Vs OLTP Integrated OLAM and OLAP Architecture

UNIT 3 ASSOCIATION RULE MINING **9 Hrs.**

Mining frequent patterns - Associations and correlations - Mining methods - Finding Frequent itemset using Candidate Generation - Generating Association Rules from Frequent Itemsets - Mining Frequent itemset without Candidate Generation - Mining various kinds of association rules - Mining Multi-Level Association Rule-Mining MultiDimensional Association Rule-Mining Correlation analysis - Constraint based association mining.

UNIT 4 CLASSIFICATION AND PREDICTION **9 Hrs.**

Classification and prediction - Issues Regarding Classification and Prediction - Classification by Decision Tree Induction - Bayesian classification - Baye's Theorem - Naïve Bayesian Classification - Bayesian Belief Network - Rule based classification - Classification by Backpropagation - Support vector machines - Prediction - Linear Regression

UNIT 5 CLUSTERING, APPLICATIONS AND TRENDS IN DATA MINING **9 Hrs.**

Cluster analysis - Types of data in Cluster Analysis - Categorization of major clustering methods -Partitioning methods - Hierarchical methods - Density-based methods - Grid-based methods - Model based clustering methods -Constraint Based cluster analysis - Outlier analysis - Social Impacts of Data Mining- Case Studies: Mining WWW- Mining Text Database-Mining Spatial Databases

Max.45 Hrs.

COURSE OUTCOMES

On completion of the course the student will be able to

CO1: Assess Raw Input Data and process it to provide suitable input for a range of data mining algorithm.

CO2: Design and Modelling of Data Warehouse.

CO3: Discover interesting pattern from large amount of data

CO4: Design and Deploy appropriate Classification Techniques

CO5: Able to cluster high dimensional Data

CO6: Apply suitable data mining techniques for various real time applications

SATHYABAMA
INSTITUTE OF SCIENCE AND TECHNOLOGY
(DEEMED TO BE UNIVERSITY)
Accredited "A" Grade by NAAC | 12B Status by UGC | Approved by AICTE
www.sathyabama.ac.in

SCHOOL OF COMPUTING

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

UNIT – I - DATA MINING - SCSA3001

DATA MINING

Introduction - Steps in KDD - System Architecture – Types of data -Data mining functionalities - Classification of data mining systems - Integration of a data mining system with a data warehouse - Issues - Data Preprocessing - Data Mining Application.

INTRODUCTION

What is Data?

• Collection of data objects and their attributes

• An attribute is a property or characteristic of an object – Examples: eye color of a person, temperature, etc. – Attribute is also known as variable, field, characteristic, or feature

• A collection of attributes describe an object – Object is also known as record, point, case, sample, entity, or instance Attributes

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

Attribute:

It can be seen as a data field that represents characteristics or features of a data object. For a customer object attributes can be customer Id, address etc.

We can say that a **set of attributes used to describe a given object are known as attribute vector or feature vector.**

Type of attributes:

This is the First step of Data Data-preprocessing. We differentiate between different types of attributes and then pre process the data. So here is description of attribute types.

1. Qualitative (Nominal (N), Ordinal (O), Binary (B)).

2. Quantitative (Discrete, Continuous)

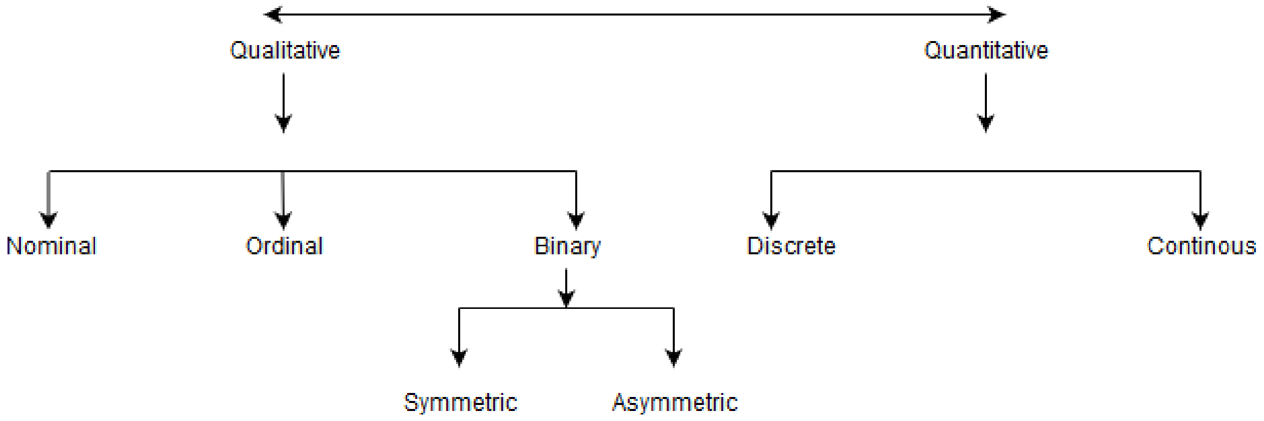
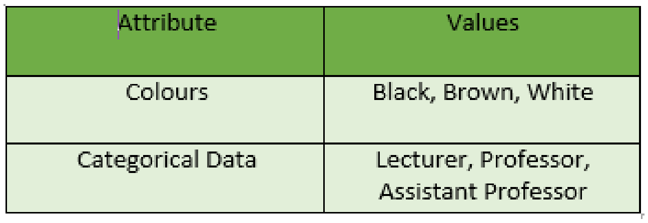


Figure 1.1 Type of attributes

Qualitative Attributes

1. Nominal Attributes – related to names:

The values of a Nominal attribute are name of things, some kind of symbols. Values of Nominal attributes represents some category or state and that's why nominal attribute also referred as **categorical attributes** and there is no order among values of nominal attribute. Example



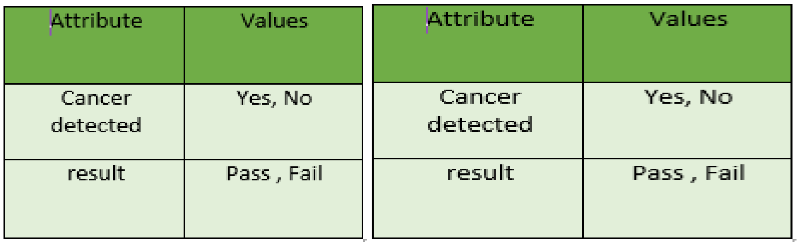
Attribute	Values
Colours	Black, Brown, White
Categorical Data	Lecturer, Professor, Assistant Professor

Table 1.1 Nominal Attributes

2. ***Binary Attributes:*** Binary data has only 2 values/states. For Example yes or no, affected or unaffected, true or false.

3. i) ***Symmetric:*** Both values are equally important (Gender).

   ii) ***Asymmetric:*** Both values are not equally important (Result).

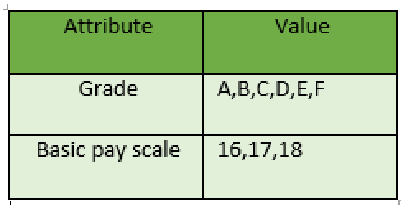


Attribute	Values
Cancer detected	Yes, No
result	Pass , Fail

Attribute	Values
Cancer detected	Yes, No
result	Pass , Fail

Table 1.2 binary Attributes

Ordinal Attributes **:** The Ordinal Attributes contains values that have a meaningful sequence or ranking(order) between them, but the magnitude between values is not actually known, the order of values that shows what is important but don't indicate how important it is.



Attribute	Value
Grade	A,B,C,D,E,F
Basic pay scale	16,17,18

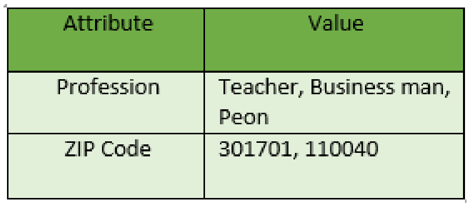
Table 1.3 Ordinal Attributes

Quantitative Attributes

1. **Numeric:** A numeric attribute is quantitative because, it is a measurable quantity, represented in integer or real values. Numerical attributes are of 2 types, **interval** and **ratio**. i) An **interval-scaled** attribute has values, whose differences are interpretable, but the numerical attributes do not have the correct reference point or we can call zero point. Data can be added and subtracted at interval scale but cannot be multiplied or divided. Consider an example of temperature in degrees Centigrade. If a day's temperature of one day is twice than the other day we cannot say that one day is twice as hot as another day.

ii) A **ratio-scaled** attribute is a numeric attribute with an fix zero-point. If a measurement is ratio-scaled, we can say of a value as being a multiple (or ratio) of another value. The values are ordered, and we can also compute the difference between values, and the mean, median, mode, Quantile-range and five number summaries can be given.

2. ***Discrete:*** Discrete data have finite values it can be numerical and can also be in categorical form. These attributes has finite or countable infinite set of values. Example

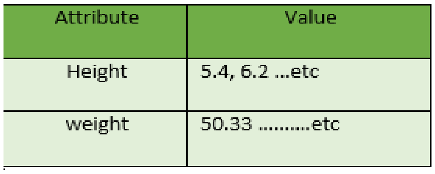


Attribute	Value
Profession	Teacher, Business man, Peon
ZIP Code	301701, 110040

Table 1.4 Discrete Attributes

3. ***Continuous:*** Continuous data have infinite no of states. Continuous data is of float type. There can be many values between 2 and 3.

   Example:

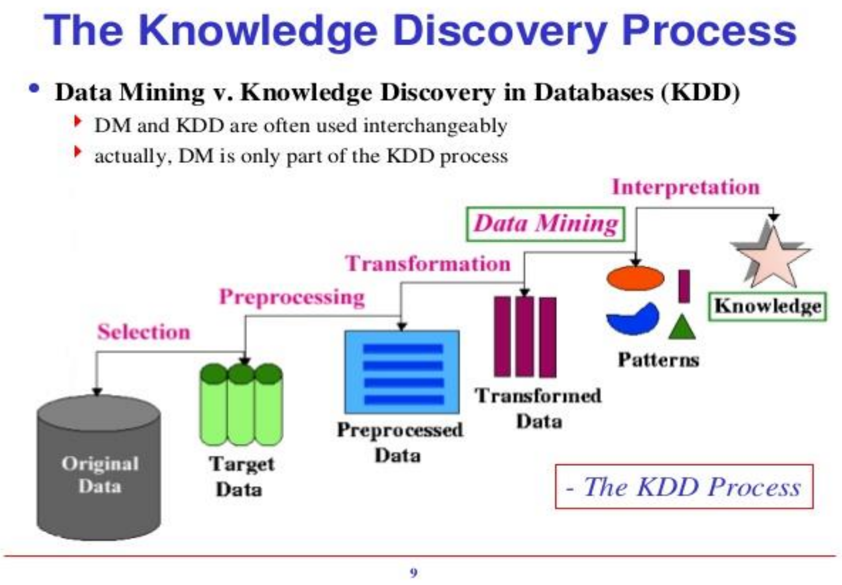


Attribute	Value
Height	5.4, 6.2 ...etc
weight	50.33 ....etc

Table 1.5 Continuous Attributes

STEPS INVOLVED IN KDD PROCESS

Data Mining also known as Knowledge Discovery in Databases refers to the nontrivial extraction of implicit, previously unknown and potentially useful information from data stored in databases.



The Knowledge Discovery Process

- Data Mining v. Knowledge Discovery in Databases (KDD)
  - DM and KDD are often used interchangeably
  - actually, DM is only part of the KDD process

Figure 1.2 KDD Process

1. ***Data Cleaning***: Data cleaning is defined as removal of noisy and irrelevant data from collection.
   - Cleaning in case of ***Missing values***.
   - Cleaning ***noisy*** data, where noise is a random or variance error.
   - Cleaning with ***Data discrepancy detection*** and ***Data transformation tools***.
2. ***Data Integration***: Data integration is defined as heterogeneous data from multiple sources combined in a common source (Data Warehouse).
   - Data integration using ***Data Migration tools***.
   - Data integration using ***Data Synchronization tools***.
   - Data integration using ***ETL*** (Extract-Load-Transformation) process.

3. ***Data Selection***: Data selection is defined as the process where data relevant to the analysis is decided and retrieved from the data collection.
    - Data selection using ***Neural network***.
    - Data selection using ***Decision Trees***.
    - Data selection using ***Naive bayes***.
    - Data selection using ***Clustering***, ***Regression***, etc.
4. ***Data Transformation***: Data Transformation is defined as the process of transforming data into appropriate form required by mining procedure.

    Data Transformation is a two-step process:

    - ***Data Mapping***: Assigning elements from source base to destination to capture transformations.
    - ***Code generation***: Creation of the actual transformation program.
5. ***Data Mining***: Data mining is defined as clever techniques that are applied to extract patterns potentially useful.
    - Transforms task relevant data into ***patterns***.
    - Decides purpose of model using ***classification*** or ***characterization***.
6. ***Pattern Evaluation***: Pattern Evaluation is defined as as identifying strictly increasing patterns representing knowledge based on given measures.
    - Find ***interestingness score*** of each pattern.
    - Uses ***summarization*** and ***Visualization*** to make data understandable by user.
7. ***Knowledge representation***: Knowledge representation is defined as technique which utilizes visualization tools to represent data mining results.
    - Generate ***reports***.
    - Generate ***tables***.
    - Generate ***discriminant rules***, ***classification rules***, ***characterization rules***, etc.

Note:

- KDD is an ***iterative* process** where evaluation measures can be enhanced, mining can be refined, new data can be integrated and transformed in order to get different and more appropriate results.
- ***Preprocessing* of databases** consists of **Data cleaning** and **Data Integration**.

SYSTEM ARCHITECTURE

Data mining is a very important process where potentially useful and previously unknown information is extracted from large volumes of data. There are a number of components involved in the data mining process. These components constitute the architecture of a data mining system.

Data Mining Architecture

The major components of any data mining system are data source, data warehouse server, data mining engine, pattern evaluation module, graphical user interface and knowledge base.

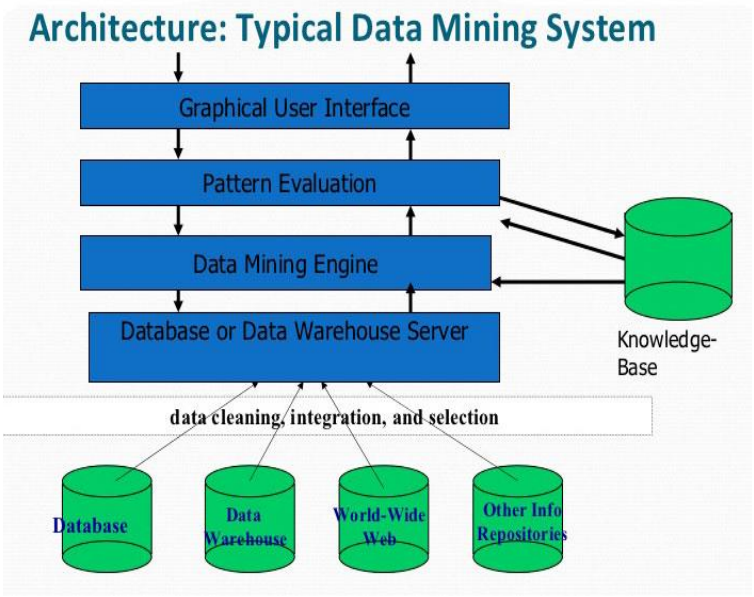


Figure 1.3 system Architecture

a) Data Sources

Database, data warehouse, World Wide Web (WWW), text files and other documents are the actual sources of data. You need large volumes of historical data for data mining to be successful. Organizations usually store data in databases or data warehouses. Data warehouses may contain one or more databases, text files, spreadsheets or other kinds of information repositories. Sometimes, data may reside even in plain text files or spreadsheets. World Wide Web or the Internet is another big source of data.

Different Processes

The data needs to be cleaned, integrated and selected before passing it to the database or data warehouse server. As the data is from different sources and in different formats, it cannot be used directly for the data mining process because the data might not be complete and reliable. So, first data needs to be cleaned and integrated. Again, more data than required will be collected from different data sources and only the data of interest needs to be selected and passed to the server. These processes are not as simple as we think. A number of techniques may be performed on the data as part of cleaning, integration and selection.

b) Database or Data Warehouse Server

The database or data warehouse server contains the actual data that is ready to be processed. Hence, the server is responsible for retrieving the relevant data based on the data mining request of the user.

c) Data Mining Engine

The data mining engine is the core component of any data mining system. It consists of a number of modules for performing data mining tasks including association, classification, characterization, clustering, prediction, time-series analysis etc.

d) Pattern Evaluation Modules

The pattern evaluation module is mainly responsible for the measure of interestingness of the pattern by using a threshold value. It interacts with the data mining engine to focus the search towards interesting patterns.

e) Graphical User Interface

The graphical user interface module communicates between the user and the data mining system. This module helps the user use the system easily and efficiently without knowing the real complexity behind the process. When the user specifies a query or a task, this module interacts with the data mining system and displays the result in an easily understandable manner.

f) Knowledge Base

The knowledge base is helpful in the whole data mining process. It might be useful for guiding the search or evaluating the interestingness of the result patterns. The knowledge base might even contain user beliefs and data from user experiences that can be useful in the process of data mining. The data mining engine might get inputs from the knowledge base to make the result more accurate and reliable. The pattern evaluation module interacts with the knowledge base on a regular basis to get inputs and also to update it.

Summary

Each and every component of data mining system has its own role and importance in completing data mining efficiently.

DATA MINING FUNCTIONALITIES

Data mining functionalities are used to specify the kind of patterns to be found in data mining tasks. Data mining tasks can be classified into two categories: descriptive and predictive.

Descriptive mining tasks characterize the general properties of the data in the database.

Predictive mining tasks perform inference on the current data in order to make predictions.

Concept/Class Description: Characterization and Discrimination

Data can be associated with classes or concepts. For example, in the Electronics store, classes of items for sale include computers and printers, and concepts of customers include big Spenders and budget Spenders.

Data characterization

Data characterization is a summarization of the general characteristics or features of a target class of data.

Data discrimination

Data discrimination is a comparison of the general features of target class data objects with the general features of objects from one or a set of contrasting classes.

Mining Frequent Patterns, Associations, and Correlations

Frequent patterns, are patterns that occur frequently in data. There are many kinds of frequent patterns, including itemsets, subsequences, and substructures.

Association analysis

Suppose, as a marketing manager, you would like to determine which items are frequently purchased together within the same transactions.

buys(X,“computer”)=buys(X,“software”) [support=1%,confidence=50%]

Where X is a variable representing a customer. Confidence=50% means that if a customer buys a computer, there is a 50% chance that she will buy software as well.

Support=1% means that 1% of all of the transactions under analysis showed that computer and software were purchased together.

Classification:

There is a large variety of data mining systems available. Data mining systems may integrate techniques from the following −

- Spatial Data Analysis
- Information Retrieval

- Pattern Recognition
- Image Analysis
- Signal Processing
- Computer Graphics
- Web Technology
- Business
- Bioinformatics

DATA MINING SYSTEM CLASSIFICATION

A data mining system can be classified according to the following criteria −

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

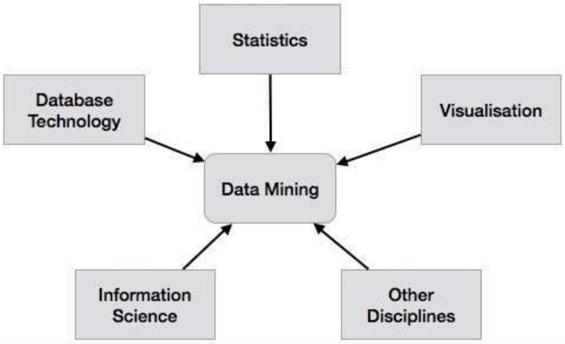


Figure 1.4 system Architecture

Apart from these, a data mining system can also be classified based on the kind of (a) databases mined, (b) knowledge mined, (c) techniques utilized, and (d) applications adapted.

Classification Based on the Databases Mined

We can classify a data mining system according to the 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 a database according to the data model, then we may have a relational, transactional, object-relational, or data warehouse mining system.

Classification Based on the kind of Knowledge Mined

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

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

Classification Based on the Techniques Utilized

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

Classification Based on the Applications Adapted

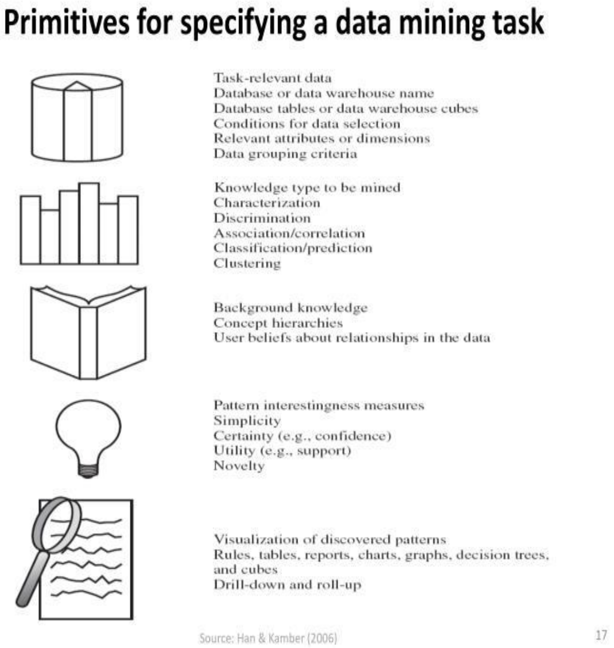
We can classify a data mining system according to the applications adapted. These applications are as follows −

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

Data Mining Task Primitives

Each user will have a data mining task in mind, that is, some form of data analysis that he or she would like to have performed. A data mining task can be specified in the form of a data mining query, which is input to the data mining system. A data mining query is defined in terms of data mining task primitives. These primitives allow the user to interactively communicate with the data mining system during discovery in order to direct the mining process, or examine the findings from different angles or depths. set of task-relevant data to be mined: This specifies the portions of the database or the set of data in which the user is interested. This includes the database attributes or data warehouse dimensions of interest (referred to as the relevant attributes or dimensions). The kind of knowledge to be mined: This specifies the data mining functions to be performed, such as characterization, discrimination, association or correlation analysis, classification, prediction, clustering, outlier analysis, or evolution analysis.

The background knowledge to be used in the discovery process: This knowledge about the domain to be mined is useful for guiding the knowledge discovery process and for evaluating the patterns found. Concept hierarchies are a popular form of background knowledge, which allow data to be mined at multiple levels of abstraction. User beliefs regarding relationships in the data are another form of background knowledge. The interestingness measures and thresholds for pattern evaluation: They may be used to guide the mining process or, after discovery, to evaluate the discovered patterns. Different kinds of knowledge may have different interestingness measures. For example, interestingness measures for association rules include support and confidence. Rules whose support and confidence values are below user-specified thresholds are considered uninteresting. The expected representation for visualizing the discovered patterns: This refers to the form in which discovered patterns are to be displayed, which may include rules, tables, charts, graphs, decision trees, and cubes. A data mining query language can be designed to incorporate these primitives, allowing users to flexibly interact with data mining systems. Having a data mining query language provides a foundation on which user-friendly graphical interfaces can be built.



Primitives for specifying a data mining task

Task-relevant data
Database or data warehouse name
Database tables or data warehouse cubes
Conditions for data selection
Relevant attributes or dimensions
Data grouping criteria

Knowledge type to be mined
Characterization
Discrimination
Association/correlation
Classification/prediction
Clustering

Background knowledge
Concept hierarchies
User beliefs about relationships in the data

Pattern interestingness measures
Simplicity
Certainty (e.g., confidence)
Utility (e.g., support)
Novelty

Visualization of discovered patterns
Rules, tables, reports, charts, graphs, decision trees, and cubes
Drill-down and roll-up

Source: Han & Kamber (2006)

17

Figure 1.5 Data mining tasks

INTEGRATING A DATA MINING SYSTEM WITH A DB/DW SYSTEM

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

The list of Integration Schemes is as follows −

- ***No Coupling*** − In this scheme, the data mining system does not utilize any of the database or data warehouse functions. It fetches the data from a particular source and processes that data using some data mining algorithms. The data mining result is stored in another file.
- ***Loose Coupling*** − In this scheme, the data mining system may use some of the functions of database and data warehouse system. It fetches the data from the data respiratory managed by these systems and performs data mining on that data. It then stores the mining result either in a file or in a designated place in a database or in a data warehouse.
- ***Semi−tight Coupling*** − In this scheme, the data mining system is linked with a database or a data warehouse system and in addition to that, efficient implementations of a few data mining primitives can be provided in the database.
- ***Tight coupling*** − In this coupling scheme, the data mining system is smoothly integrated into the database or data warehouse system. The data mining subsystem is treated as one functional component of an information system.

MAJOR ISSUES IN DATA WAREHOUSING AND MINING

• **Mining methodology and user interaction**

– Mining different kinds of knowledge in databases

– Interactive mining of knowledge at multiple levels of abstraction – Incorporation of background knowledge

– Data mining query languages and ad-hoc data mining

– Expression and visualization of data mining results

– Handling noise and incomplete data

– Pattern evaluation: the interestingness problem

• Performance and scalability

– Efficiency and scalability of data mining algorithms

– Parallel, distributed and incremental mining methods

• Issues relating to the diversity of data types

– Handling relational and complex types of data

– Mining information from heterogeneous databases and global information systems (WWW)

• Issues related to applications and social impacts

– Application of discovered knowledge

• Domain-specific data mining tools

Issues:

Data mining is not an easy task, as the algorithms used can get very complex and data is not always available at one place. It needs to be integrated from various heterogeneous data sources. These factors also create some issues. Here in this tutorial, we will discuss the major issues regarding −

- Mining Methodology and User Interaction
- Performance Issues
- Diverse Data Types Issues

The following diagram describes the major issues.

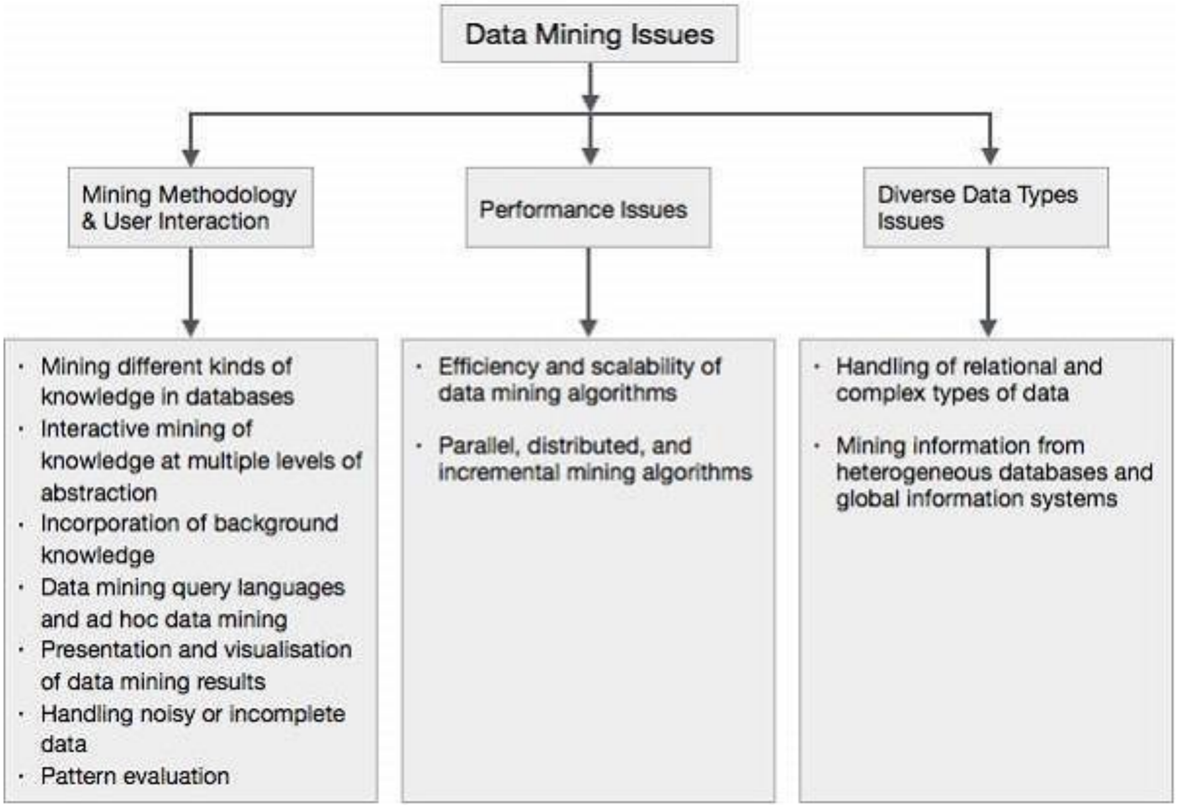

Data Mining Issues
Mining Methodology & User Interaction
Performance Issues
Diverse Data Types Issues
· Mining different kinds of knowledge in databases
· Interactive mining of knowledge at multiple levels of abstraction
· Incorporation of background knowledge
· Data mining query languages and ad hoc data mining
· Presentation and visualisation of data mining results
· Handling noisy or incomplete data
· Pattern evaluation
· Efficiency and scalability of data mining algorithms
· Parallel, distributed, and incremental mining algorithms
· Handling of relational and complex types of data
· Mining information from heterogeneous databases and global information systems

Figure 1.6 Data Mining Issues

Mining Methodology and User Interaction Issues:

It refers to the following kinds of issues −

- ***Mining different kinds of knowledge in databases*** − Different users may be interested in different kinds of knowledge. Therefore it is necessary for data mining to cover a broad range of knowledge discovery task.
- ***Interactive mining of knowledge at multiple levels of abstraction*** − The data mining process needs to be interactive because it allows users to focus the search for patterns, providing and refining data mining requests based on the returned results.
- ***Incorporation of background knowledge*** − To guide discovery process and to express the discovered patterns, the background knowledge can be used. Background knowledge may be used to express the discovered patterns not only in concise terms but at multiple levels of abstraction.
- ***Data mining query languages and ad hoc data mining*** − Data Mining Query language that allows the user to describe ad hoc mining tasks, should be integrated with a data warehouse query language and optimized for efficient and flexible data mining.
- ***Presentation and visualization of data mining results*** − Once the patterns are discovered it needs to be expressed in high level languages, and visual representations. These representations should be easily understandable.
- ***Handling noisy or incomplete data*** − The data cleaning methods are required to handle the noise and incomplete objects while mining the data regularities. If the data cleaning methods are not there then the accuracy of the discovered patterns will be poor.
- ***Pattern evaluation*** − The patterns discovered should be interesting because either they represent common knowledge or lack novelty.

Performance Issues:

There can be performance-related issues such as follows −

- ***Efficiency and scalability of data mining algorithms***− In order to effectively extract the information from huge amount of data in databases; data mining algorithm must be efficient and scalable.
- ***Parallel, distributed, and incremental mining algorithms*** − The factors such as huge size of databases, wide distribution of data, and complexity of data mining methods motivate the development of parallel and distributed data mining algorithms. These algorithms divide the

data into partitions which is further processed in a parallel fashion. Then the results from the partitions are merged. The incremental algorithms, update databases without mining the data again from scratch.

Diverse Data Types Issues:

- ***Handling of relational and complex types of data*** − The database may contain complex data objects, multimedia data objects, spatial data, temporal data etc. It is not possible for one system to mine all these kind of data.
- ***Mining information from heterogeneous databases and global information systems*** − The data is available at different data sources on LAN or WAN. These data source may be structured, semi structured or unstructured. Therefore mining the knowledge from them adds challenges to data mining.

DATA PREPROCESSING

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 pre processing:

- **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.

Integration of a data mining system with a data warehouse:

DB and DW systems, possible integration schemes include *no coupling*, *loose coupling*, *semi-tight coupling*, and *tight coupling*. We examine each of these schemes, as follows:

1. No coupling: *No coupling* means that a DM system will not utilize any function of a DB or DW system. It may fetch data from a particular source (such as a file system), process data using some data mining algorithms, and then store the mining results in another file.

2. Loose coupling: *Loose coupling* means that a DM system will use some facilities of a DB or DW system, fetching data from a data repository managed by these systems, performing data mining, and then storing the mining results either in a file or in a designated place in a database or data Warehouse. Loose coupling is better than no coupling because it can fetch any portion of data stored in databases or data warehouses by using query processing, indexing, and other system facilities.

However, many loosely coupled mining systems are main memory-based. Because mining does not explore data structures and query optimization methods provided by DB or DW systems, it is difficult for loose coupling to achieve high scalability and good performance with large data sets.

3. Semi-tight coupling: *Semi-tight coupling* means that besides linking a DM system to a DB/DW system, efficient implementations of a few essential data mining primitives (identified by the analysis of frequently encountered data mining functions) can be provided in the DB/DW system. These primitives can include sorting, indexing, aggregation, histogram analysis, multi way join, and pre computation of some essential statistical measures, such as sum, count, max, min ,standard deviation,

4. Tight coupling: *Tight coupling* means that a DM system is smoothly integrated into the DB/DW system. The data mining subsystem is treated as one functional component of information system. Data mining queries and functions are optimized based on mining query analysis, data structures, indexing schemes, and query processing methods of a DB or DW system.

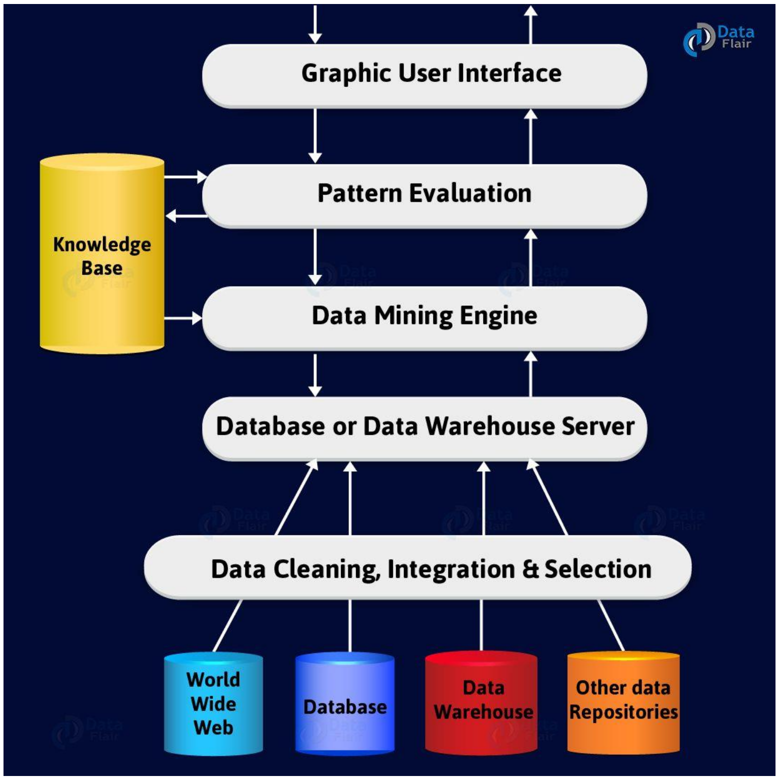


Figure 1.7 Integration of a data mining system with a data warehouse:

DATA MINING APPLICATIONS

Here is the list of areas where data mining is widely used −

- Financial Data Analysis
- Retail Industry
- Telecommunication Industry
- Biological Data Analysis
- Other Scientific Applications
- Intrusion Detection

Financial Data Analysis

The financial data in banking and financial industry is generally reliable and of high quality which facilitates systematic data analysis and data mining. Some of the typical cases are as follows −

- Design and construction of data warehouses for multidimensional data analysis and data mining.
- Loan payment prediction and customer credit policy analysis.
- Classification and clustering of customers for targeted marketing.
- Detection of money laundering and other financial crimes.

Retail Industry

Data Mining has its great application in Retail Industry because it collects large amount of data from on sales, customer purchasing history, goods transportation, consumption and services. It is natural that the quantity of data collected will continue to expand rapidly because of the increasing ease, availability and popularity of the web.

Data mining in retail industry helps in identifying customer buying patterns and trends that lead to improved quality of customer service and good customer retention and satisfaction. Here is the list of examples of data mining in the retail industry −

- Design and Construction of data warehouses based on the benefits of data mining.
- Multidimensional analysis of sales, customers, products, time and region.
- Analysis of effectiveness of sales campaigns.
- Customer Retention.
- Product recommendation and cross-referencing of items.

Telecommunication Industry

Today the telecommunication industry is one of the most emerging industries providing various services such as fax, pager, cellular phone, internet messenger, images, e-mail, web data transmission, etc. Due to the development of new computer and communication technologies, the telecommunication industry is rapidly expanding. This is the reason why data mining is become very important to help and understand the business.

Data mining in telecommunication industry helps in identifying the telecommunication patterns, catch fraudulent activities, make better use of resource, and improve quality of service. Here is the list of examples for which data mining improves telecommunication services −

- Multidimensional Analysis of Telecommunication data.
- Fraudulent pattern analysis.
- Identification of unusual patterns.
- Multidimensional association and sequential patterns analysis.
- Mobile Telecommunication services.
- Use of visualization tools in telecommunication data analysis.

Biological Data Analysis

In recent times, we have seen a tremendous growth in the field of biology such as genomics, proteomics, functional Genomics and biomedical research. Biological data mining is a very important part of Bioinformatics. Following are the aspects in which data mining contributes for biological data analysis −

- Semantic integration of heterogeneous, distributed genomic and proteomic databases.
- Alignment, indexing, similarity search and comparative analysis multiple nucleotide sequences.
- Discovery of structural patterns and analysis of genetic networks and protein pathways.
- Association and path analysis.
- Visualization tools in genetic data analysis.

Other Scientific Applications

The applications discussed above tend to handle relatively small and homogeneous data sets for which the statistical techniques are appropriate. Huge amount of data have been collected from scientific domains such as geosciences, astronomy, etc. A large amount of data sets is being generated because of the fast numerical simulations in various fields such as climate and ecosystem modelling, chemical engineering, fluid dynamics, etc. Following are the applications of data mining in the field of Scientific Applications −

- Data Warehouses and data preprocessing.
- Graph-based mining.
- Visualization and domain specific knowledge.

Intrusion Detection

Intrusion refers to any kind of action that threatens integrity, confidentiality, or the availability of network resources. In this world of connectivity, security has become the major issue. With increased usage of internet and availability of the tools and tricks for intruding and attacking network prompted intrusion detection to become a critical component of network administration. Here is the list of areas in which data mining technology may be applied for intrusion detection −

- Development of data mining algorithm for intrusion detection.
- Association and correlation analysis, aggregation to help select and build discriminating attributes.
- Analysis of Stream data.
- Distributed data mining.
- Visualization and query tools.

Data Mining System Products

There are many data mining system products and domain specific data mining applications. The new data mining systems and applications are being added to the previous systems. Also, efforts are being made to standardize data mining languages.

Choosing a Data Mining System

The selection of a data mining system depends on the following features −

- ***Data Types*** − The data mining system may handle formatted text, record-based data, and relational data. The data could also be in ASCII text, relational database data or data warehouse data. Therefore, we should check what exact format the data mining system can handle.
- ***System Issues*** − We must consider the compatibility of a data mining system with different operating systems. One data mining system may run on only one operating system or on several. There are also data mining systems that provide web-based user interfaces and allow XML data as input.
- ***Data Sources*** − Data sources refer to the data formats in which data mining system will operate. Some data mining system may work only on ASCII text files while others on multiple relational sources. Data mining system should also support ODBC connections or OLE DB for ODBC connections.
- ***Data Mining functions and methodologies*** − There are some data mining systems that provide only one data mining function such as classification while some provides multiple data mining functions such as concept description, discovery-driven OLAP analysis, association mining, linkage analysis, statistical analysis, classification, prediction, clustering, outlier analysis, similarity search, etc.
- ***Coupling data mining with databases or data warehouse systems*** − Data mining systems need to be coupled with a database or a data warehouse system. The coupled components are integrated into a uniform information processing environment. Here are the types of coupling listed below −
    - No coupling
    - Loose Coupling
    - Semi tight Coupling
    - Tight Coupling
- ***Scalability*** − There are two scalability issues in data mining −

- ***Row (Database size) Scalability*** − A data mining system is considered as row scalable when the number or rows are enlarged 10 times. It takes no more than 10 times to execute a query.
- ***Column (Dimension) Scalability*** − A data mining system is considered as column scalable if the mining query execution time increases linearly with the number of columns.

- ***Visualization Tools*** − Visualization in data mining can be categorized as follows −
  - Data Visualization
  - Mining Results Visualization
  - Mining process visualization
  - Visual data mining
- ***Data Mining query language and graphical user interface*** − An easy-to-use graphical user interface is important to promote user-guided, interactive data mining. Unlike relational database systems, data mining systems do not share underlying data mining query language.

Trends in Data Mining

Data mining concepts are still evolving and here are the latest trends that we get to see in this field

- Application Exploration.
- Scalable and interactive data mining methods.
- Integration of data mining with database systems, data warehouse systems and web database systems.
- Standardization of data mining query language.
- Visual data mining.
- New methods for mining complex types of data.
- Biological data mining.
- Data mining and software engineering.
- Web mining.
- Distributed data mining.
- Real time data mining.
- Multi database data mining.
- Privacy protection and information security in data mining

PART-A			
Q. No	**Questions**	**Competence**	**BT Level**
1.	Define Data mining. List out the steps in data mining.	Remember	BTL-1
2.	Compare Discrete versus Continuous Attributes.	Analyze	BTL-4
3.	Give the applications of Data Mining.	Understand	BTL-2
4.	Analyze the issues in Data Mining Techniques.	Apply	BTL-3
5.	Generalize in detail about Numeric Attributes.	Create	BTL-6
6.	Evaluate the major tasks of data preprocessing.	Evaluate	BTL-5
7.	Define an efficient procedure for cleaning the noisy data.	Remember	BTL-1
8.	Distinguish between data similarity and dissimilarity.	Understand	BTL-2
9.	Show the Displays of Basic Statistical Descriptions of Data.	Analyze	BTL-4
10.	Formulate what is data discretization.	Create	BTL-6
PART-B			
Q. No	**Questions**	**Competence**	**BT Level**
1.	i) Describe the issues of data mining. (7) ii) Describe in detail about the applications of data mining (6)	Remember	BTL-1
2.	i) State and explain the various classifications of data mining systems with example. (7) ii) Explain the various data mining functionalities in detail. (6)	Analyze	BTL-4
3.	i) Describe the steps involved in Knowledge discovery in databases (KDD). (7) ii) Draw the diagram and Describe the architecture of data mining system. (6)	Remember	BTL-1

4.	Suppose that the data for analysis include the attributed age. The age values for the data tuples are 13,15,16,19,20,20,21,22,22,25,25,25,25,30,33,33,35,35, 35,35,36,40,45,46,52,70. i)Use smoothing by bin depth of 3.Illustrate your steps (6) ii) Classify the various methods for data smoothing. (7)	Create	BTL-6
5.	(i) Discuss whether or not each of the following activities is a data mining task.(5) 1. Credit card fraud detection using transaction records. 2. Dividing the customers of a company according to their gender. 3. Computing the total sales of a company 4. Predicting the future stock price of a company using historical records. 5. Monitoring seismic waves for earthquake activities. (ii) Discuss on descriptive and predictive data mining tasks with illustrations. (8)	Understand	BTL-2
6.	i) Generalize why do we need data preprocessing step in data mining (8) ii) Explain the various methods of data cleaning and data reduction techniques (7)	Evaluate	BTL-5
7.	i) Compose in detail the various data transformation techniques (7) ii) Develop a short note on discretization techniques (6)	Create	BTL-6

TEXT / REFERENCE BOOKS

1. Jiawei Han and Micheline Kamber, “Data Mining Concepts and Techniques”, 2nd Edition, Elsevier, 2007

2. Alex Berson and Stephen J. Smith, “ Data Warehousing, Data Mining & OLAP”, Tata McGraw Hill, 2007.

3. Pang-Ning Tan, Michael Steinbach and Vipin Kumar, “Introduction To Data Mining”, Person Education, 2007.

4. K.P. Soman, Shyam Diwakar and V. Ajay, “Insight into Data mining Theory and Practice”, Easter Economy Edition,
Prentice Hall of India, 2006.

5. G. K. Gupta, “Introduction to Data Mining with Case Studies”, Easter Economy Edition, Prentice Hall of India, 2006.

6. Daniel T.Larose, “Data Mining Methods and Models”, Wile-Interscience, 2006

UNIT – II - DATA WAREHOUSING- SCSA3001

DATA WAREHOUSING

Data warehousing components - Building a data warehouse - Multi Dimensional Data Model - OLAP Operation in the Multi-Dimensional Model - Three Tier Data Warehouse Architecture - Schemas for Multi-dimensional data Model - Online Analytical Processing (OLAP) - OLAP Vs OLTP Integrated OLAM and OLAP Architecture

DATA WAREHOUSING COMPONENTS

What is Data warehouse?

Data warehouse is an information system that contains historical and commutative data from single or multiple sources. It simplifies reporting and analysis process of the organization. It is also a single version of truth for any company for decision making and forecasting.

Characteristics of Data warehouse

- Subject-Oriented
- Integrated
- Time-variant
- Non-volatile

Subject-Oriented

A data warehouse is subject oriented as it offers information regarding a theme instead of companies' on-going operations. These subjects can be sales, marketing, distributions, etc.

A data warehouse never focuses on the on-going operations. Instead, it put emphasis on modelling and analysis of data for **decision making**. It also provides a simple and concise view around the specific subject by excluding data which not helpful to support the decision process.

Integrated

In Data Warehouse, integration means the establishment of a common unit of measure for all similar data from the dissimilar database. The data also needs to be stored in the Data warehouse in common and universally acceptable manner.

A data warehouse is developed by integrating data from varied sources like a mainframe, relational databases, flat files, etc. Moreover, it must keep consistent naming conventions, format, and coding.

This integration helps in effective analysis of data. Consistency in naming conventions, attribute measures, encoding structure etc. has to be ensured.

Time-Variant

The time horizon for data warehouse is quite extensive compared with operational systems. The data collected in a data warehouse is recognized with a particular period and offers information from the historical point of view. It contains an element of time, explicitly or implicitly. One such place where Data warehouse data display time variance is in in the structure of the record key. Every primary key contained with the DW should have either implicitly or explicitly an element of time. Like the day, week month, etc. Another aspect of time variance is that once data is inserted in the warehouse, it can't be updated or changed.

Non-volatile

Data warehouse is also non-volatile means the previous data is not erased when new data is entered in it. Data is read-only and periodically refreshed. This also helps to analyze historical data and understand what & when happened. It does not require transaction process, recovery and concurrency control mechanisms.

Activities like delete, update, and insert which are performed in an operational application environment are omitted in Data warehouse environment. Only two types of data operations performed in the Data Warehousing are

1. Data loading
2. Data access

Data Warehouse Architectures

Single-tier architecture

The objective of a single layer is to minimize the amount of data stored. This goal is to remove data redundancy. This architecture is not frequently used in practice.

Two-tier architecture

Two-layer architecture separates physically available sources and data warehouse. This architecture is not expandable and also not supporting a large number of end-users. It also has connectivity problems because of network limitations.

Three-tier architecture

This is the most widely used architecture.

It consists of the Top, Middle and Bottom Tier.

1. ***Bottom Tier:*** The database of the Data warehouse servers as the bottom tier. It is usually a relational database system. Data is cleansed, transformed, and loaded into this layer using back-end tools.
2. ***Middle-Tier:*** The middle tier in Data warehouse is an OLAP server which is implemented using either ROLAP or MOLAP model. For a user, this application tier presents an abstracted view of the database. This layer also acts as a mediator between the end-user and the database.
3. ***Top-Tier:*** The top tier is a front-end client layer. Top tier is the tools and API that you connect and get data out from the data warehouse. It could be Query tools, reporting tools, managed query tools, Analysis tools and Data mining tools.

DATA WAREHOUSE COMPONENTS

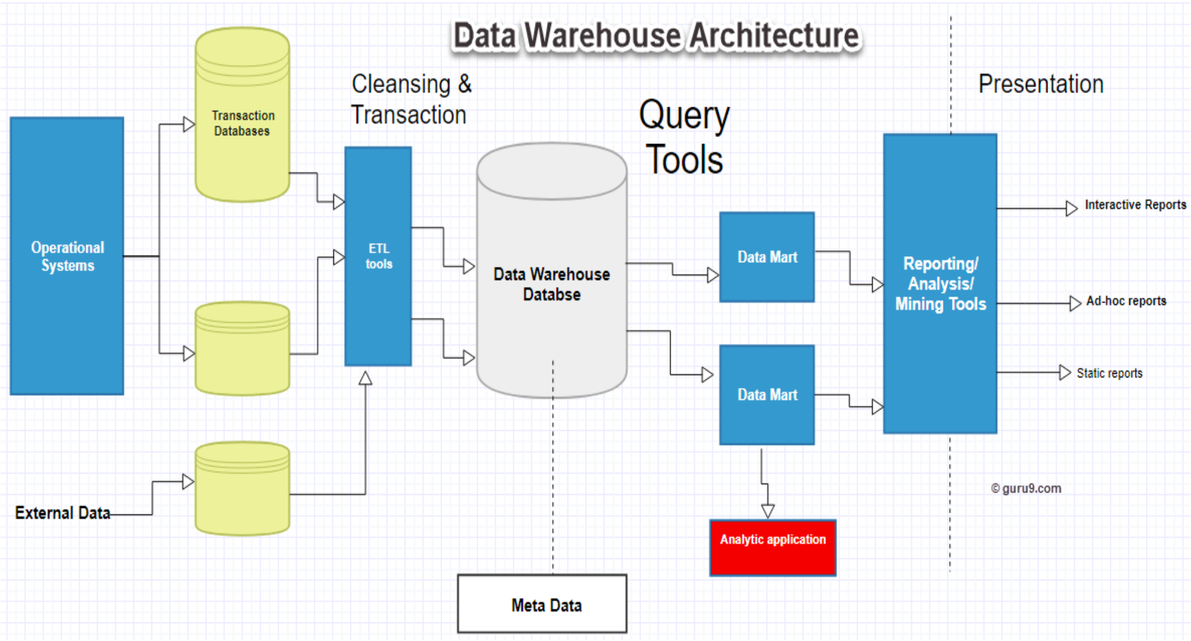


Figure 2.1 Data warehouse Components

The data warehouse is based on an RDBMS server which is a central information repository that is surrounded by some key components to make the entire environment functional, manageable and accessible

There are mainly five components of Data Warehouse:

Data Warehouse Database:

The central database is the foundation of the data warehousing environment. This database is implemented on the RDBMS technology. Although, this kind of implementation is constrained by the fact that traditional RDBMS system is optimized for transactional database processing and not for data warehousing. For instance, ad-hoc query, multi-table joins, aggregates are resource intensive and slow down performance.

Hence, alternative approaches to Database are used as listed below-

- In a data warehouse, relational databases are deployed in parallel to allow for scalability. Parallel relational databases also allow shared memory or shared nothing model on various multiprocessor configurations or massively parallel processors.
- New index structures are used to bypass relational table scan and improve speed.
- Use of multidimensional database (MDDBs) to overcome any limitations which are placed because of the relational data model. Example: Essbase from Oracle.

Sourcing, Acquisition, Clean-up and Transformation Tools (ETL)

The data sourcing, transformation, and migration tools are used for performing all the conversions, summarizations, and all the changes needed to transform data into a unified format in the data warehouse. They are also called Extract, Transform and Load (ETL) Tools.

Their functionality includes:

- Anonymize data as per regulatory stipulations.
- Eliminating unwanted data in operational databases from loading into Data warehouse.
- Search and replace common names and definitions for data arriving from different sources.
- Calculating summaries and derived data
- In case of missing data, populate them with defaults.
- De-duplicated repeated data arriving from multiple data sources.

These Extract, Transform, and Load tools may generate cron jobs, background jobs, Cobol programs, shell scripts, etc. that regularly update data in data warehouse. These tools are also helpful to maintain the Metadata.

These ETL Tools have to deal with challenges of Database & Data heterogeneity.

Metadata

The name Meta Data suggests some high- level technological concept. However, it is quite simple. Metadata is data about data which defines the data warehouse. It is used for building, maintaining and managing the data warehouse.

In the Data Warehouse Architecture, meta-data plays an important role as it specifies the source, usage, values, and features of data warehouse data. It also defines how data can be changed and processed. It is closely connected to the data warehouse.

Metadata helps to answer the following questions

- What tables, attributes, and keys does the Data Warehouse contain?
- Where did the data come from?
- How many times do data get reloaded?
- What transformations were applied with cleansing?

Metadata can be classified into following categories:

1. **Technical Meta Data**: This kind of Metadata contains information about warehouse which is used by Data warehouse designers and administrators.
2. **Business Meta Data:** This kind of Metadata contains detail that gives end-users a way easy to understand information stored in the data warehouse.

Query Tools

One of the primary objects of data warehousing is to provide information to businesses to make strategic decisions. Query tools allow users to interact with the data warehouse system.

These tools fall into four different categories:

1. Query and reporting tools
2. Application Development tools
3. Data mining tools
4. OLAP tools

1. Query and reporting tools:

Query and reporting tools can be further divided into

- Reporting tools
- Managed query tools

Reporting tools: Reporting tools can be further divided into production reporting tools and desktop report writer.

1. Report writers: This kind of reporting tool is tools designed for end-users for their analysis.
2. Production reporting: This kind of tools allows organizations to generate regular operational reports. It also supports high volume batch jobs like printing and calculating. Some popular reporting tools are Brio, Business Objects, Oracle, Power Soft, SAS Institute.

Managed query tools:

This kind of access tools helps end users to resolve snags in database and SQL and database structure by inserting meta-layer between users and database.

2. Application development tools:

Sometimes built-in graphical and analytical tools do not satisfy the analytical needs of an organization. In such cases, custom reports are developed using Application development tools.

3. Data mining tools:

Data mining is a process of discovering meaningful new correlation, pattens, and trends by mining large amount data. Data mining tools are used to make this process automatic.

4. OLAP tools:

These tools are based on concepts of a multidimensional database. It allows users to analyse the data using elaborate and complex multidimensional views.

Data warehouse Bus Architecture

Data warehouse Bus determines the flow of data in your warehouse. The data flow in a data warehouse can be categorized as Inflow, Upflow, Downflow, Outflow and Meta flow.

While designing a Data Bus, one needs to consider the shared dimensions, facts across data marts.

Data Marts

A data mart is an access layer which is used to get data out to the users. It is presented as an option for large size data warehouse as it takes less time and money to build. However, there is no standard definition of a data mart is differing from person to person.

In a simple word Data mart is a subsidiary of a data warehouse. The data mart is used for partition of data which is created for the specific group of users.

Data marts could be created in the same database as the Data warehouse or a physically separate Database.

Data warehouse Architecture Best Practices

To design Data Warehouse Architecture, you need to follow below given best practices:

- Use a data model which is optimized for information retrieval which can be the dimensional mode, denormalized or hybrid approach.
- Need to assure that Data is processed quickly and accurately. At the same time, you should take an approach which consolidates data into a single version of the truth.
- Carefully design the data acquisition and cleansing process for Data warehouse.
- Design a Meta Data architecture which allows sharing of metadata between components of Data Warehouse

- Consider implementing an ODS model when information retrieval need is near the bottom of the data abstraction pyramid or when there are multiple operational sources required to be accessed.
- One should make sure that the data model is integrated and not just consolidated. In that case, you should consider 3NF data model. It is also ideal for acquiring ETL and Data cleansing tools

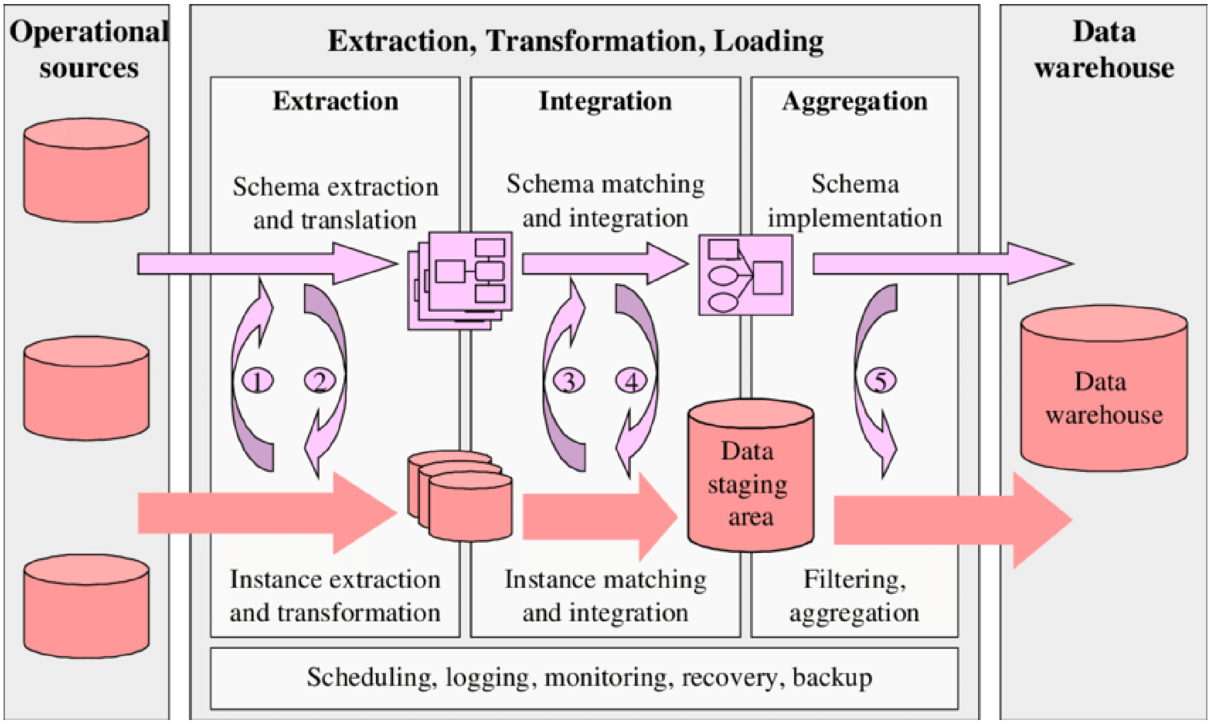
Summary:

- Data warehouse is an information system that contains historical and commutative data from single or multiple sources.
- A data warehouse is subject oriented as it offers information regarding subject instead of organization's ongoing operations.
- In Data Warehouse, integration means the establishment of a common unit of measure for all similar data from the different databases
- Data warehouse is also non-volatile means the previous data is not erased when new data is entered in it.
- A Data warehouse is Time-variant as the data in a DW has high shelf life.
- There are 5 main components of a Data warehouse. 1) Database 2) ETL Tools 3) Meta Data 4) Query Tools 5) Data Marts
- These are four main categories of query tools 1. Query and reporting, tools 2. Application Development tools, 3. Data mining tools 4. OLAP tools
- The data sourcing, transformation, and migration tools are used for performing all the conversions and summarizations.
- In the Data Warehouse Architecture, meta-data plays an important role as it specifies the source, usage, values, and features of data warehouse data.

BUILDING A DATA WAREHOUSE

In general, building any data warehouse consists of the following steps:

1. Extracting the transactional data from the data sources into a staging area
2. Transforming the transactional data
3. Loading the transformed data into a dimensional database
4. Building pre-calculated summary values to speed up report generation
5. Building (or purchasing) a front-end reporting tool


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

2.2 Diagram for building a data warehouse

Extracting Transactional Data:

A large part of building a DW is **pulling data from various data sources and placing it in a central storage area**. In fact, this can be the most difficult step to accomplish due to the reasons mentioned earlier: Most people who worked on the systems in place have moved on to other jobs. Even if they haven't left the company, you still have a lot of work to do: You need to figure out which database system to use for your staging area and how to pull data from various sources into that area.

Fortunately for many small to mid-size companies, Microsoft has come up with an excellent tool for data extraction. Data Transformation Services (DTS), which is part of Microsoft SQL Server 7.0 and 2000, allows you to import and export data from any OLE DB or ODBC-compliant database as long as you have an appropriate provider. This tool is available at no extra cost when you purchase Microsoft SQL Server. The sad reality is that you won't always have an OLE DB or ODBC-compliant data source to work with, however. If not, you're bound to make a considerable investment of time and effort in writing a custom program that transfers data from the original source into the staging database.

Transforming Transactional Data:

An equally important and challenging step after extracting is **transforming and relating the data** extracted from multiple sources. As I said earlier, your source systems were most likely built by many different IT professionals. Let's face it. Each person sees the world through their own eyes, so each solution is at least a bit different from the others. The data model of your mainframe system might be very different from the model of the client-server system.

Most companies have their data spread out in a number of various database management systems: MS Access, MS SQL Server, Oracle, Sybase, and so on. Many companies will also have much of their data in flat files, spread sheets, mail systems and other types of data stores. When building a data warehouse, you need to relate data from all of these sources and build some type of a staging area that can handle data extracted from any of these source systems. After all the data is in the staging area, you have to massage it and give it a common shape. Prior to massaging data, you need to figure out a way to relate tables and columns of one system to the tables and columns coming from the other systems.

Creating a Dimensional Model:

The third step in building a data warehouse is **coming up with a dimensional model**. Most modern transactional systems are built using the relational model. The relational database is highly normalized; when designing such a system, you try to get rid of repeating columns and make all columns dependent on the primary key of each table. The relational systems perform well in the On-Line Transaction Processing (OLTP) environment. On the other hand, they perform rather poorly in the reporting (and especially DW) environment, in which joining multiple huge tables just is *not* the best idea.

The relational format is not very efficient when it comes to building reports with summary and aggregate values. The dimensional approach, on the other hand, provides a way to improve query performance without affecting data integrity. However, the query performance improvement comes with a storage space penalty; a dimensional database will generally take up much more space than its relational counterpart. These days, storage space is fairly inexpensive, and most companies can afford large hard disks with a minimal effort.

The dimensional model consists of the fact and dimension tables. The fact tables consist of foreign keys to each dimension table, as well as measures. The *measures* are a factual representation of how well (or how poorly) your business is doing (for instance, the number of parts produced per hour or the number of cars rented per day). *Dimensions*, on the other hand, are what your business users expect in the reports—the details about the measures. For example, the time dimension tells the user that 2000 parts were produced between 7 a.m. and 7 p.m. on the specific day; the plant dimension specifies that these parts were produced by the Northern plant.

Just like any modeling exercise the dimensional modeling is not to be taken lightly. Figuring out the needed dimensions is a matter of discussing the business requirements with your users over and over again. When you first talk to the users they have very minimal requirements: "Just give me those reports that show me how each portion of the company performs." Figuring out what "each portion of the company" means is your job as a DW architect. The company may consist of regions, each of which report to a different vice president of operations. Each region, on the other hand, might consist of areas, which in turn might consist of individual stores. Each store could have several departments. When the DW is complete, splitting the revenue among the regions won't be enough. That's when your users will demand more features and additional drill-down capabilities. Instead of waiting for that to happen, an architect should take proactive measures to get all the necessary requirements ahead of time.

It's also important to realize that not every field you import from each data source may fit into the dimensional model. Indeed, if you have a sequential key on a mainframe system, it won't have much meaning to your business users. Other columns might have had significance eons ago when the system was built. Since then, the management might have changed its mind about the relevance of such columns. So don't worry if all of the columns you imported are not part of your dimensional model.

Loading the Data:

After you've built a dimensional model, it's time to **populate it with the data** in the staging database. This step only sounds trivial. It might involve combining several columns together or splitting one field into several columns. You might have to perform several lookups before calculating certain values for your dimensional model.

Keep in mind that such data transformations can be performed at either of the two stages: while extracting the data from their origins or while loading data into the dimensional model. I wouldn't recommend one way over the other—make a decision depending on the project. If your users need to be sure that they can extract all the data first, wait until all data is extracted prior to transforming it. If the dimensions are known prior to extraction, go on and transform the data while extracting it.

Generating Precalculated Summary Values:

The next step is **generating the precalculated summary values** which are commonly referred to as *aggregations*. This step has been tremendously simplified by SQL Server Analysis Services (or OLAP Services, as it is referred to in SQL Server 7.0). After you have populated your dimensional database, SQL Server Analysis Services does all the aggregate generation work for you. However, remember that depending on the number of dimensions you have in your DW, building aggregations can take a long time. As a rule of thumb, the more dimensions you have, the more time it'll take to build aggregations. However, the size of each dimension also plays a significant role.

Prior to generating aggregations, you need to make an important choice about which dimensional model to use: ROLAP (Relational OLAP), MOLAP (Multidimensional OLAP), or HOLAP (Hybrid OLAP). The ROLAP model builds additional tables for storing the aggregates, but this takes much more storage space than a dimensional database, so be careful! The MOLAP model stores the aggregations as well as the data in multidimensional format, which is far more efficient than ROLAP. The HOLAP approach keeps the data in the relational format, but builds aggregations in multidimensional format, so it's a combination of ROLAP and MOLAP.

Regardless of which dimensional model you choose, ensure that SQL Server has as much memory as possible. Building aggregations is a memory-intensive operation, and the more memory you provide, the less time it will take to build aggregate values.

Building (or Purchasing) a Front-End Reporting Tool

After you've built the dimensional database and the aggregations you can decide how sophisticated your **reporting tools** need to be. If you just need the drill-down capabilities, and your users have Microsoft Office 2000 on their desktops, the Pivot Table Service of Microsoft Excel 2000 will do the job. If the reporting needs are more than what Excel can offer, you'll have to investigate the alternative of building or purchasing a reporting tool. The cost of building a custom reporting (and OLAP) tool will usually outweigh the purchase price of a third-party tool. That is not to say that OLAP tools are cheap.

There are several major vendors on the market that have top-notch analytical tools. In addition to the third-party tools, Microsoft has just released its own tool, Data Analyzer, which can be a cost-effective alternative. Consider purchasing one of these suites before delving into the process of developing your own software because reinventing the wheel is not always beneficial or affordable. Building OLAP tools is not a trivial exercise by any means.

MULTIDIMENSIONAL DATA MODEL

Multidimensional data model stores data in the form of data cube. Mostly, data warehousing supports two or three-dimensional cubes.

A data cube allows data to be viewed in multiple dimensions. Dimensions are entities with respect to which an organization wants to keep records. For example in store sales record, dimensions allow the store to keep track of things like monthly sales of items and the branches and locations. A multidimensional database helps to provide data-related answers to complex business queries quickly and accurately. Data warehouses and Online Analytical Processing (OLAP) tools are based on a multidimensional data model. OLAP in data warehousing enables users to view data from different angles and dimensions

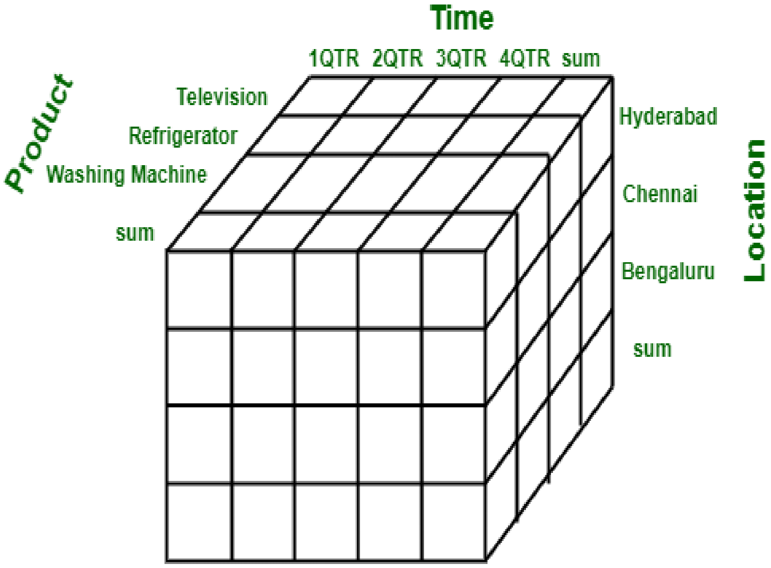


Figure 2.3 Multidimensional Data Representation

The multi-Dimensional Data Model is a method which is used for ordering data in the database along with good arrangement and assembling of the contents in the database.

The Multi-Dimensional Data Model allows customers to interrogate analytical questions associated with market or business trends, unlike relational databases which allow customers to access data in the form of queries. They allow users to rapidly receive answers to the requests which they made by creating and examining the data comparatively fast.

OLAP (online analytical processing) and data warehousing uses multi-dimensional databases. It is used to show multiple dimensions of the data to users.

Working on a Multidimensional Data Model

The following stages should be followed by every project for building a Multi-Dimensional Data Model:

Stage 1: Assembling data from the client: In first stage, a Multi-Dimensional Data Model collects correct data from the client. Mostly, software professionals provide simplicity to the client about the range of data which can be gained with the selected technology and collect the complete data in detail.

Stage 2: Grouping different segments of the system: In the second stage, the Multi-Dimensional Data Model recognizes and classifies all the data to the respective section they belong to and also builds it problem-free to apply step by step.

Stage 3: Noticing the different proportions: In the third stage, it is the basis on which the design of the system is based. In this stage, the main factors are recognized according to the user's point of view. These factors are also known as "Dimensions".

Stage 4: Preparing the actual-time factors and their respective qualities: In the fourth stage, the factors which are recognized in the previous step are used further for identifying the related qualities. These qualities are also known as "attributes" in the database.

Stage 5: Finding the actuality of factors which are listed previously and their qualities: In the fifth stage, A Multi-Dimensional Data Model separates and differentiates the actuality from the factors which are collected by it. These actually play a significant role in the arrangement of a Multi-Dimensional Data Model.

Stage 6: Building the Schema to place the data, with respect to the information collected from the steps above: In the sixth stage, on the basis of the data which was collected previously, a Schema is built.

For Example:

1. Let us take the example of a firm. The revenue cost of a firm can be recognized on the basis of different factors such as geographical location of firm's workplace, products of the firm, advertisements done, time utilized to flourish a product, etc.

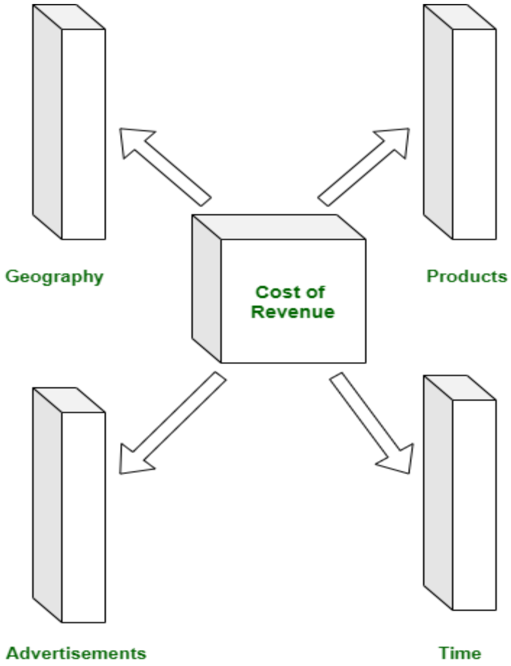
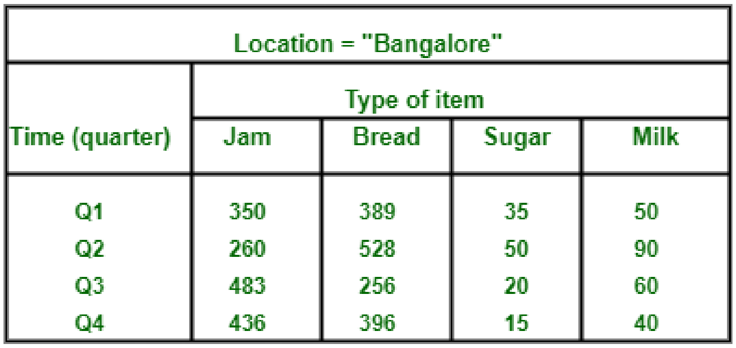


Figure 2.4 Multidimensional Data Model

2. Let us take the example of the data of a factory which sells products per quarter in Bangalore. The data is represented in the table given below:

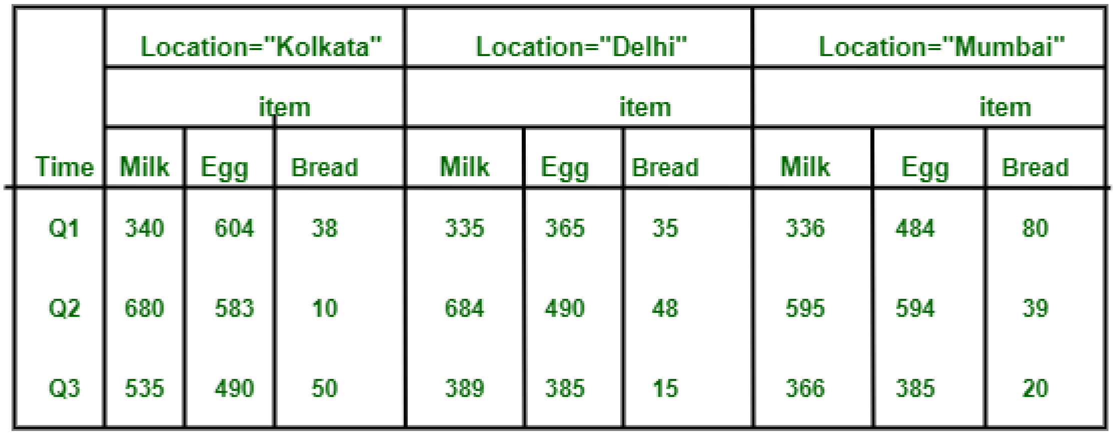


Location = "Bangalore"				
Time (quarter)	Type of item			
	Jam	Bread	Sugar	Milk
Q1	350	389	35	50
Q2	260	528	50	90
Q3	483	256	20	60
Q4	436	396	15	40

Table 2.1 2D Factory Data

In the above given presentation, the factory's sales for Bangalore are, for the time dimension, which is organized into quarters and the dimension of items, which is sorted according to the kind of item which is sold. The facts here are represented in rupees (in thousands).

Now, if we desire to view the data of the sales in a three-dimensional table, then it is represented in the diagram given below. Here the data of the sales is represented as a two dimensional table. Let us consider the data according to item, time and location (like Kolkata, Delhi, and Mumbai). Here is the table:



	Location="Kolkata"			Location="Delhi"			Location="Mumbai"		
	item			item			item		
Time	Milk	Egg	Bread	Milk	Egg	Bread	Milk	Egg	Bread
Q1	340	604	38	335	365	35	336	484	80
Q2	680	583	10	684	490	48	595	594	39
Q3	535	490	50	389	385	15	366	385	20

Figure 2.2 3D Data Representation as 2D

This data can be represented in the form of three dimensions conceptually, which is shown in the image below:

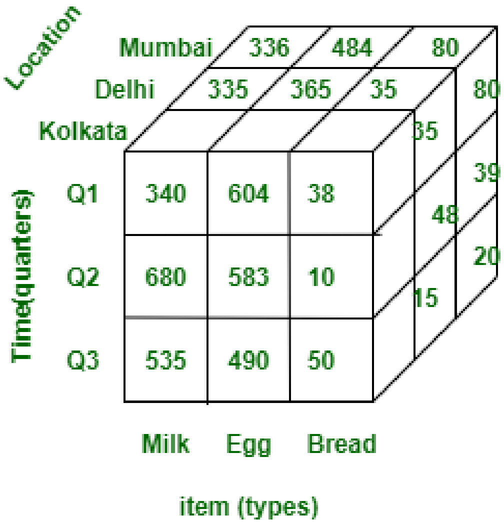


Figure 2.5 Figure 3D data representation

Advantages of Multi-Dimensional Data Model

The following are the advantages of a multi-dimensional data model:

- A multi-dimensional data model is easy to handle.
- It is easy to maintain.
- Its performance is better than that of normal databases (e.g. relational databases).
- The representation of data is better than traditional databases. That is because the multi-dimensional databases are multi-viewed and carry different types of factors.
- It is workable on complex systems and applications, contrary to the simple one-dimensional database systems.

Disadvantages of Multi-Dimensional Data Model

The following are the disadvantages of a Multi-Dimensional Data Model:

- The multi-dimensional Data Model is slightly complicated in nature and it requires professionals to recognize and examine the data in the database.
- During the work of a Multi-Dimensional Data Model, when the system caches, there is a great effect on the working of the system.
- It is complicated in nature due to which the databases are generally dynamic in design.

OLAP OPERATIONS

Online Analytical Processing Server (OLAP) is based on the multidimensional data model. It allows managers, and analysts to get an insight of the information through fast, consistent, and interactive access to information. This chapter cover the types of OLAP, operations on OLAP, difference between OLAP, and statistical databases and OLTP.

Since OLAP servers are based on multidimensional view of data, we will discuss OLAP operations in multidimensional data.

Here is the list of OLAP operations −

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

Roll-up

Roll-up performs aggregation on a data cube in any of the following ways −

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

The following diagram illustrates how roll-up works.

- Roll-up 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 the level of country.
- The data is grouped into cities rather than countries.
- When roll-up is performed, one or more dimensions from the data cube are removed.

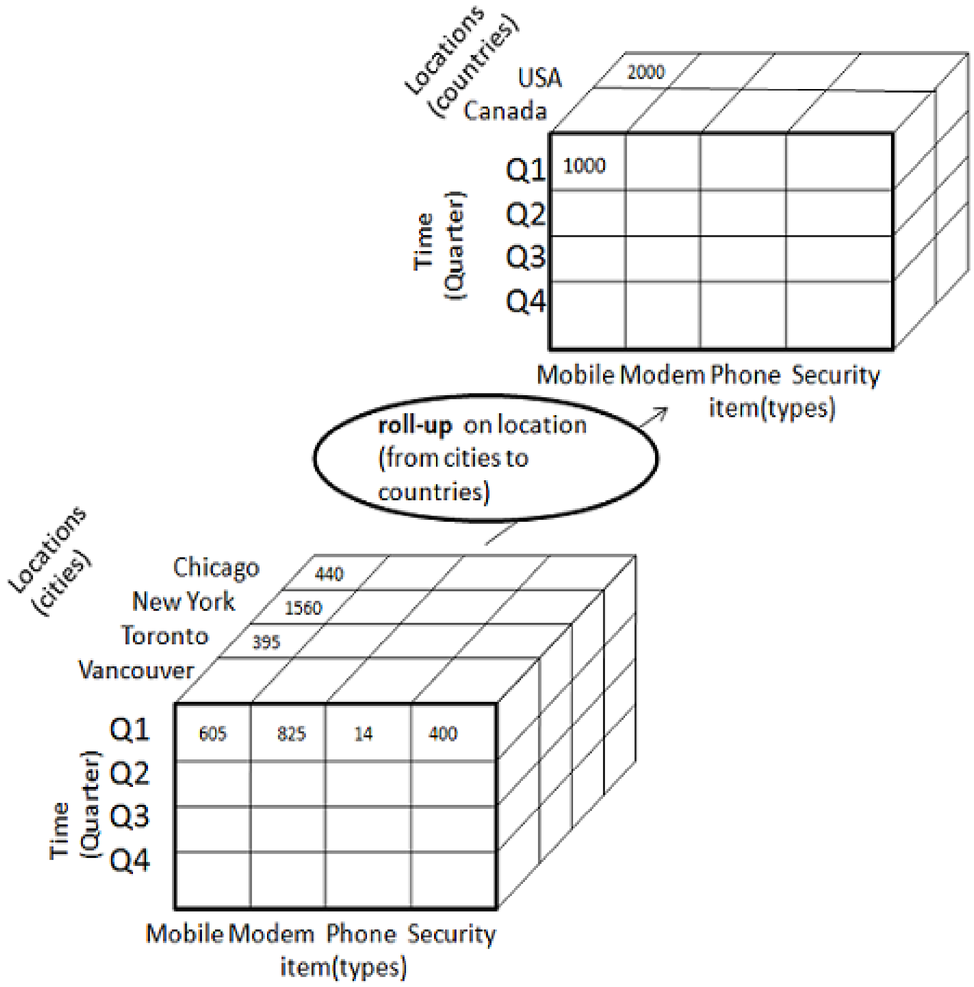


Figure 2.6 Roll up Operation

Drill-down

Drill-down is the reverse operation of roll-up. It is performed by either of the following ways −

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

The following diagram illustrates how drill-down works –

- Drill-down is performed by stepping down a concept hierarchy for the dimension time.
- Initially the concept hierarchy was "day < month < quarter < year."
- On drilling down, the time dimension is descended from the level of quarter to the level of month.
- When drill-down is performed, one or more dimensions from the data cube are added.
- It navigates the data from less detailed data to highly detailed data.

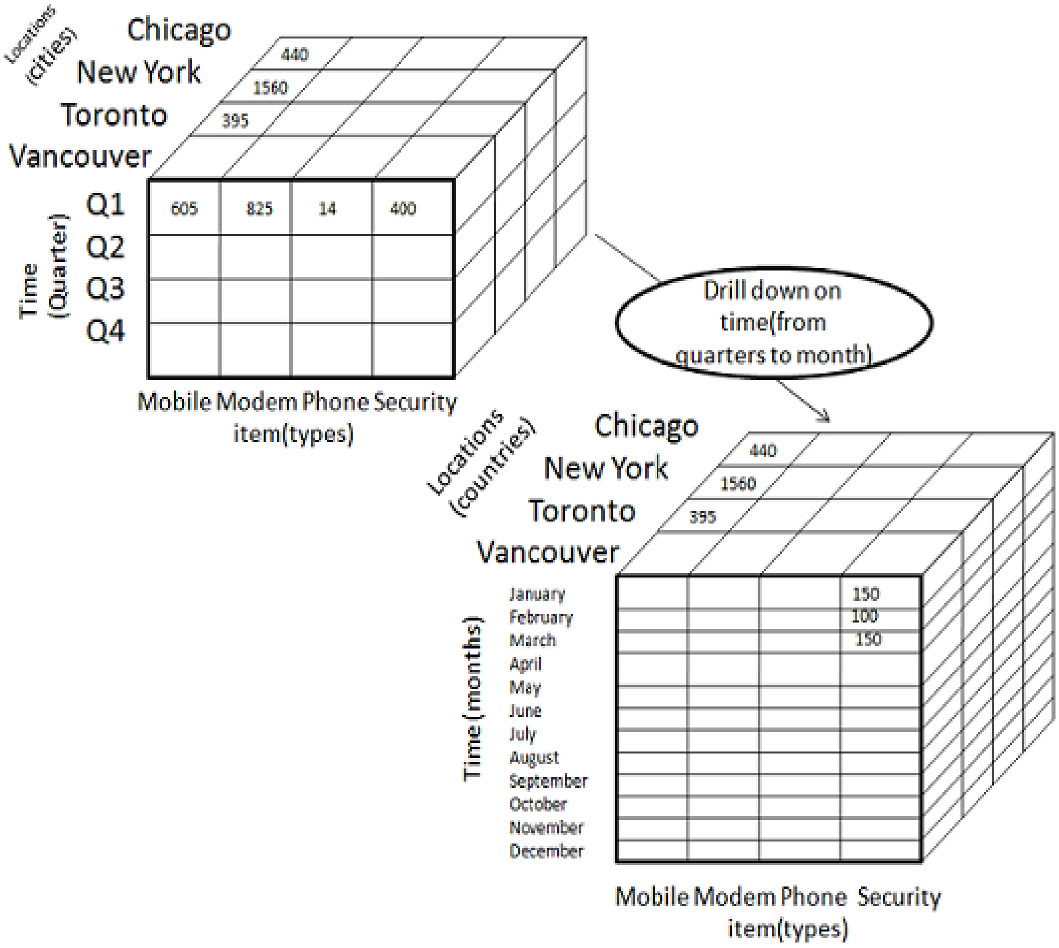


Figure 2.7 Drill-down Operation

Slice

The slice operation selects one particular dimension from a given cube and provides a new sub-cube. Consider the following diagram that shows how slice works.

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

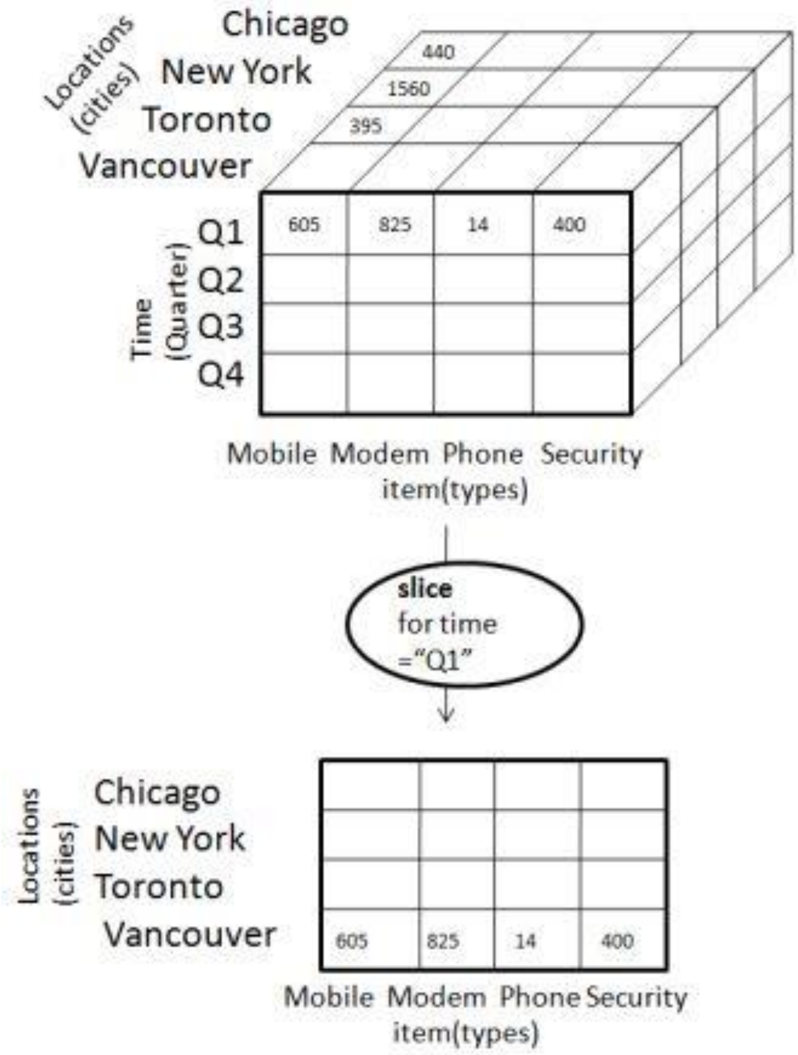


Figure 2.8 Slice Operation

Dice

Dice selects two or more dimensions from a given cube and provides a new sub-cube. Consider the following diagram that shows the dice operation.

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

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

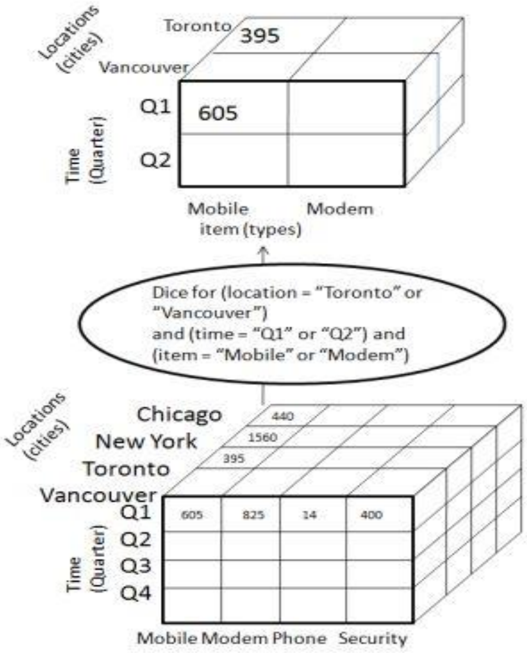


Figure 2.9 Dice Operation

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 that shows the pivot operation.

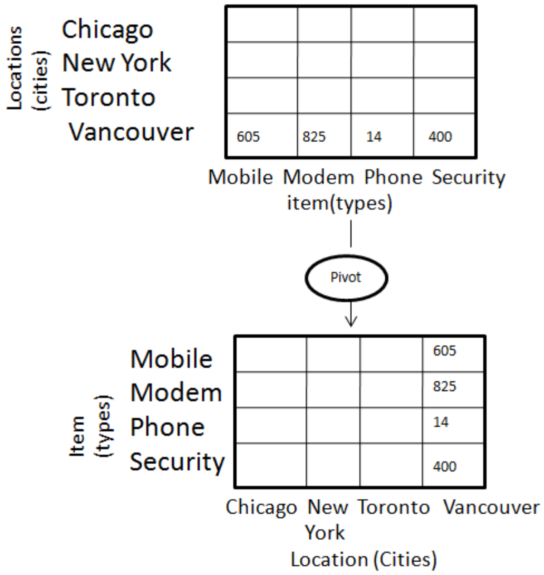


Figure 2.10 Pivot Operation

THREE-TIER DATA WAREHOUSE ARCHITECTURE

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

These 3 tiers are:

1. Bottom Tier (Data warehouse server)
2. Middle Tier (OLAP server)
3. Top Tier (Front end tools)

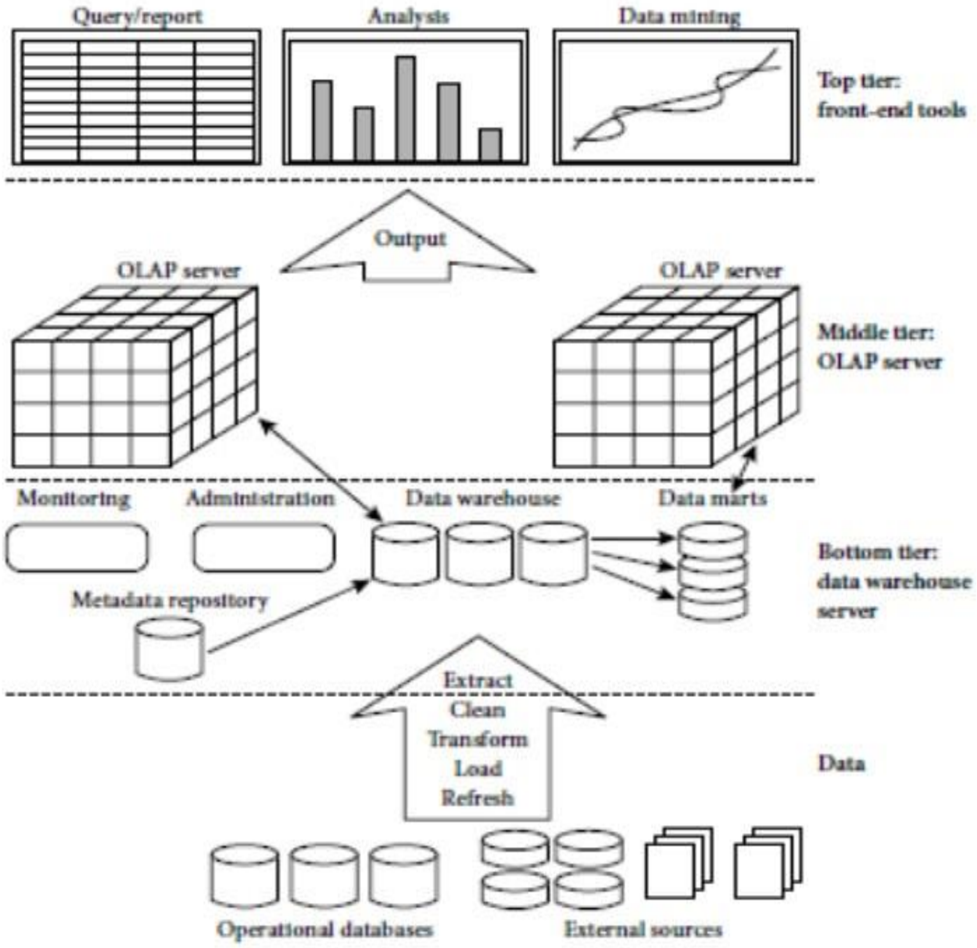


Figure 2.11 Three Tier Data Warehouse Architecture

- ***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 the bottom tier. These back end tools and utilities perform the Extract, Clean, Load, and refresh functions.

- ***Middle Tier*** − In the middle tier, we have the OLAP Server that 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 the multidimensional data and operations?

- ***Top-Tier*** − This tier is the front-end client layer. This layer holds the query tools and reporting tools, analysis tools and data mining tools.

The following diagram depicts 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 an operational data warehouse is known as a virtual warehouse. It is easy to build a virtual warehouse. Building a virtual warehouse requires excess capacity on operational database servers.

Data Mart

Data mart contains a subset of organization-wide data. This subset of data is valuable to specific groups of an organization.

In other words, we can claim that data marts contain data specific to a particular group. For example, the marketing data mart may contain data related to items, customers, and sales. Data marts 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 servers.
- The implementation data mart cycles is measured in short periods 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 its planning and design are not organization-wide.
- Data marts are small in size.
- Data marts are customized by department.
- The source of a data mart is departmentally structured data warehouse.
- Data marts are flexible.

Enterprise Warehouse

- An enterprise warehouse collects all the information and the subjects spanning an entire organization
- It provides us 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.

SCHEMAS FOR MULTI-DIMENSIONAL DATA MODEL

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

Star Schema

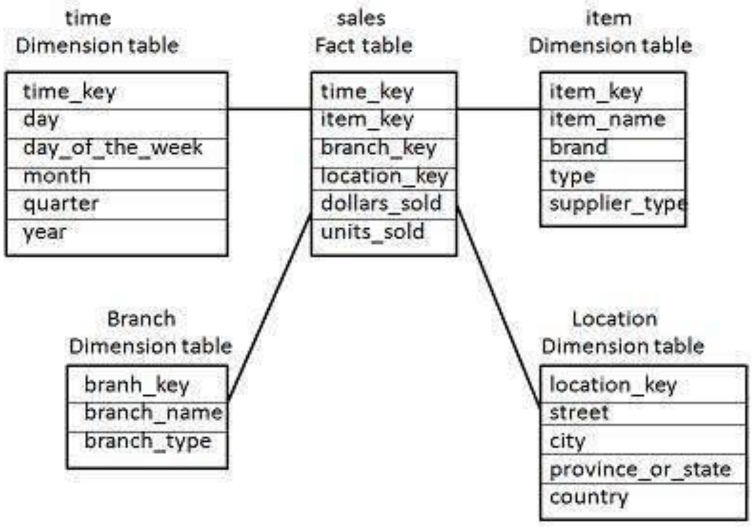


Figure 2.12 Star Schema

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

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

Snowflake Schema

- Some dimension tables in the Snowflake schema are normalized.
- The normalization splits up the data into additional tables.
- Unlike Star schema, the dimensions table in a snowflake schema are normalized. For example, the item dimension table in star schema is normalized and split into two dimension tables, namely item and supplier table.

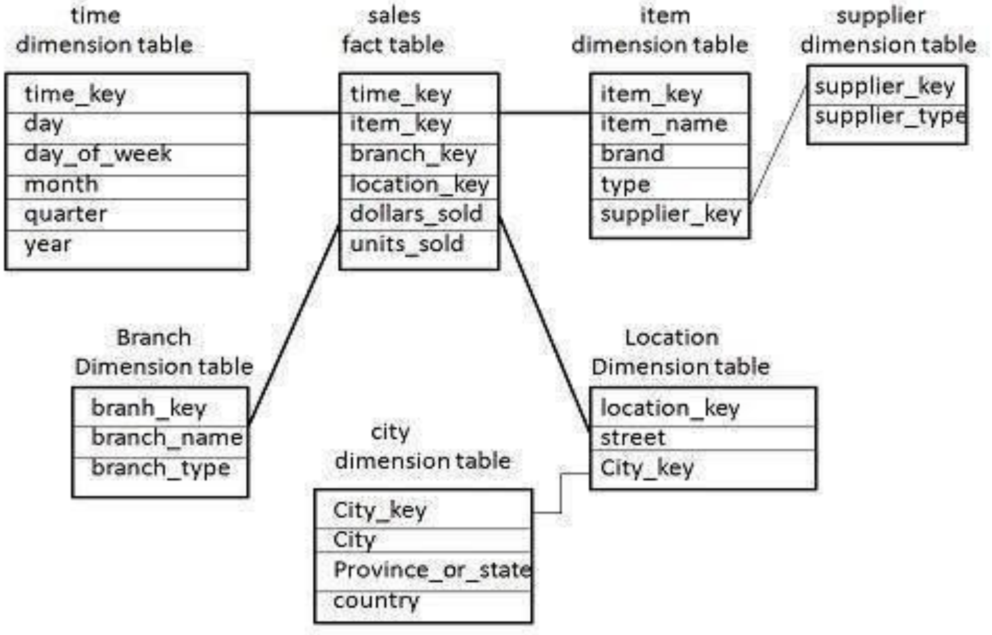


Figure 2.13 Snowflake Schema

- Now the item dimension table contains the attributes item_key, item_name, type, brand, and supplier-key.
- The supplier key is linked to the supplier dimension table. The supplier dimension table contains the attributes supplier_key and supplier_type.

Note − Due to normalization in the Snowflake schema, the redundancy is reduced and therefore, it becomes easy to maintain and the save storage space.

Fact Constellation Schema

- A fact constellation has multiple fact tables. It is also known as galaxy schema.
- The following diagram shows two fact tables, namely sales and shipping.

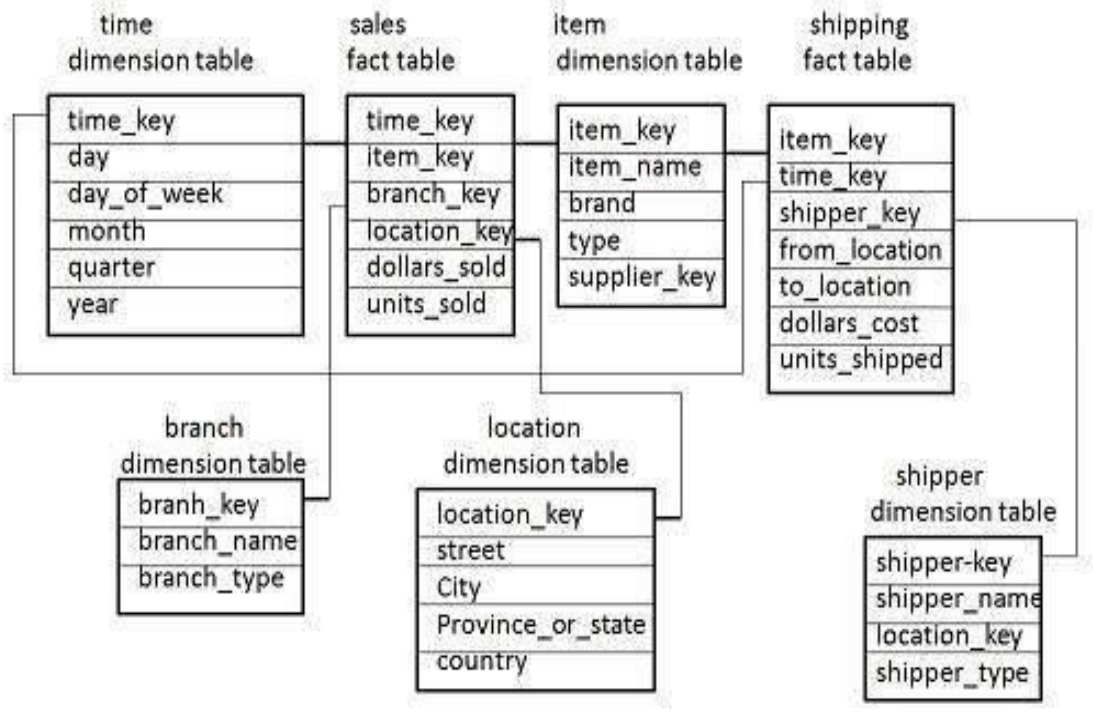


Figure 2.14 Fact Constellation Schema

- The sales fact table is same as that in the star schema.
- The shipping fact table has the five dimensions, namely item_key, time_key, shipper_key, from_location, to_location.
- The shipping fact table also contains two measures, namely dollars sold and units sold.
- It is also possible to share dimension tables between fact tables. For example, time, item, and location dimension tables are shared between the sales and shipping fact table.

OLAP (ONLINE ANALYTICAL PROCESSING)

The most popular data model for data warehouses is a multidimensional model. This model can exist in the form of a star schema, a snowflake schema, or a fact constellation schema. Let's have a look at each of these schema types

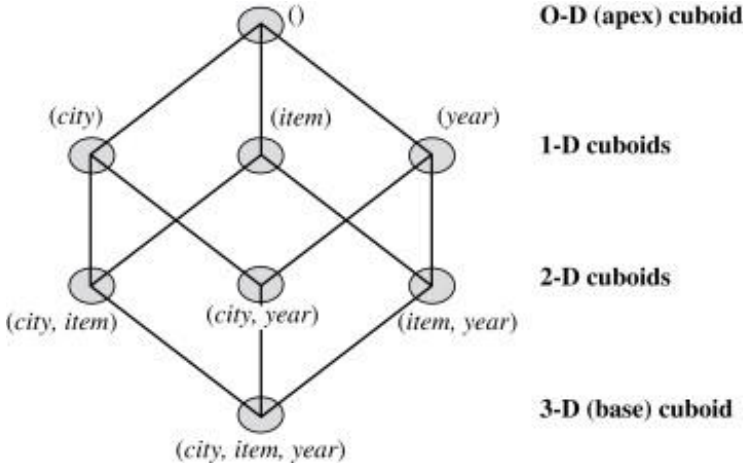


Figure 2.15 Multidimensional Data

1. Roll-up: The roll-up operation performs aggregation on a data cube, either by climbing-up a concept hierarchy for a dimension or by dimension reduction. Figure shows the result of a roll-up operation performed on the central cube by climbing up the concept hierarchy for location. This hierarchy was defined as the total order street < city < province or state <country.

2. Drill-down: Drill-down is the reverse of roll-up. It navigates from less detailed data to more detailed data. Drill-down can be realized by either stepping-down a concept hierarchy for a dimension or introducing additional dimensions. Figure shows the result of a drill-down operation performed on the central cube by stepping down a concept hierarchy for time defined as day < month < quarter < year. Drill-down occurs by descending the time hierarchy from the level of quarter to the more detailed level of month.

3. Slice and dice: The slice operation performs a selection on one dimension of the given cube, resulting in a subcube. Figure shows a slice operation where the sales data are selected from the central cube for the dimension time using the criteria time=”Q2". The dice operation defines a subcube by performing a selection on two or more dimensions.

4. Pivot (rotate): Pivot is a visualization operation which rotates the data axes in view in order to provide an alternative presentation of the data. Figure shows a pivot operation where the item and location axes in a 2-D slice are rotated.

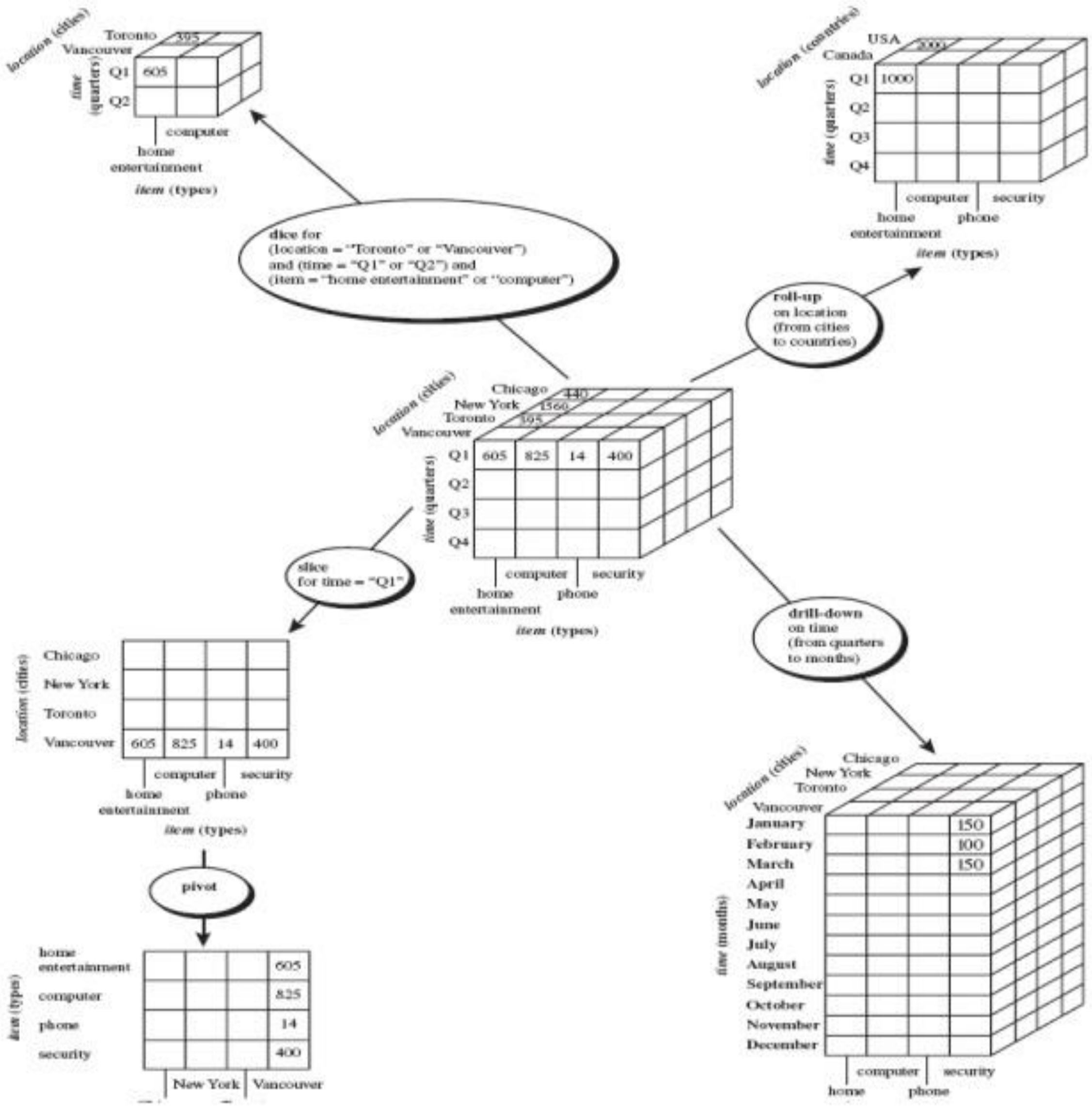


Figure 2.16 Examples of typical OLAP operations on multidimensional data

Types of OLAP Servers

We have four types of OLAP servers −

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

Relational OLAP

ROLAP servers are placed between relational back-end server and client front-end tools. To store and manage warehouse data, ROLAP uses 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 uses 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 use two levels of data storage representation to handle dense and sparse data sets.

Hybrid OLAP

Hybrid OLAP is a combination of both ROLAP and MOLAP. It offers higher scalability of ROLAP and faster computation of MOLAP. HOLAP servers allow to store the large data volumes of detailed information. The aggregations are stored separately in MOLAP store.

Specialized SQL Servers

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

INTEGRATED OLAP AND OLAM ARCHITECTURE

Online Analytical Mining integrates with Online Analytical Processing with data mining and mining knowledge in multidimensional databases. Here is the diagram that shows the integration of both OLAP and OLAM

OLAM is important for the following reasons −

High quality of data in data warehouses − The data mining tools are required to work on integrated, consistent, and cleaned data. These steps are very costly in the preprocessing of data. The data warehouses constructed by such preprocessing are valuable sources of high quality data for OLAP and data mining as well.

Available information processing infrastructure surrounding data warehouses − Information processing infrastructure refers to accessing, integration, consolidation, and transformation of multiple heterogeneous databases, web-accessing and service facilities, reporting and OLAP analysis tools

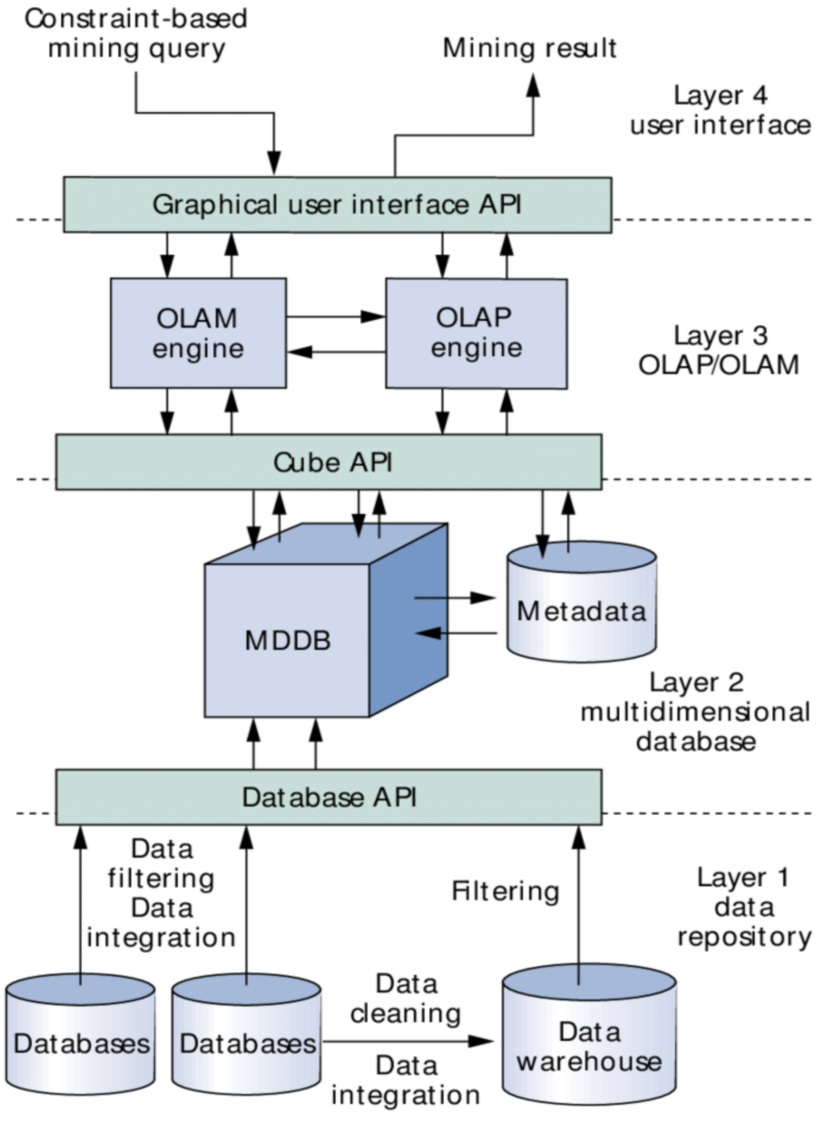


Figure 2.17 OLAM Architecture

Available information processing infrastructure surrounding data warehouses − Information processing infrastructure refers to accessing, integration, consolidation, and transformation of multiple heterogeneous databases, web-accessing and service facilities, reporting and OLAP analysis tools

OLAP−based exploratory data analysis − Exploratory data analysis is required for effective data mining. OLAM provides facility for data mining on various subsets of data and at different levels of abstraction.

Online selection of data mining functions − Integrating OLAP with multiple data mining functions and online analytical mining provide users with the flexibility to select desired data mining functions and swap data mining tasks dynamically

Features of OLTP and OLAP:

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

1. Users and system orientation: An OLTP system is customer-oriented and is used for transaction and query processing by clerks, clients, and information technology professionals. An OLAP system is market-oriented and is used for data analysis by knowledge workers, including managers, executives, and analysts.

2. Data contents: An OLTP system manages current data that, typically, are too detailed to be easily used for decision making. An OLAP system manages large amounts of historical data, provides facilities for summarization and aggregation, and stores and manages information at different levels of granularity. These features make the data easier for use in informed decision making.

3. Database design: An OLTP system usually adopts an entity-relationship (ER) data model and an application oriented database design. An OLAP system typically adopts either a star or snowflake model and a subject-oriented database design.

4. View: An OLTP system focuses mainly on the current data within an enterprise or department, without referring to historical data or data in different organizations. In contrast, an OLAP system often spans multiple versions of a database schema. OLAP systems also deal with information that originates from different organizations, integrating information from many data stores. Because of their huge volume, OLAP data are stored on multiple storage media.

5. Access patterns: The access patterns of an OLTP system consist mainly of short, atomic transactions. Such a system requires concurrency control and recovery mechanisms. However, accesses to OLAP systems are mostly read-only operations although many could be complex queries.

PART-A			
Q. No	**Questions**	**Competence**	**BT Level**
1.	How is data ware house different from a database? Also Identify the similarity.	Remember	BTL-1
2.	Compare OLTP and OLAP system.	Analyze	BTL-4
3.	Differentiate metadata and data mart.	Understand	BTL-2
4.	How would you show your understanding of Multi-dimensional data model?	Apply	BTL-3
5.	Generalize the function of OLAP tools in the internet.	Create	BTL-6
6.	How would you evaluate the goals of data mining?	Evaluate	BTL-5
7.	Can you list the categories of tools in business analysis?	Remember	BTL-1
8.	Give the need for OLAP.	Understand	BTL-2
9.	Compare drill down with roll up approach.	Analyze	BTL-4
10.	Design the data warehouse architecture.	Create	BTL-6

PART-B			
Q. No	**Questions**	**Competence**	**BT Level**
1.	i) Diagrammatically illustrate and describe the architecture of MOLAP and ROLAP. (7) ii) Identify the major differences between MOLAP and ROLAP. (6)	Remember	BTL-1
2.	(i) Draw the data warehouse architecture and explain its components. (7) (ii) Explain the different types of OLAP tools. (6)	Analyze	BTL-4
3.	(i) Discuss in detail about components of data warehousing. (7) (ii) Describe the overall architecture of data warehouse? (6)	Understand	BTL-2

4.	Examine the relevant examples discuss multidimensional online analytical processing and multi relational online analytical processing.	Apply	BTL-3
5.	What is data warehouse? Give the steps for design and construction of Data Warehouses and explain with three tier architecture diagram.	Understand	BTL-2
6.	i) Compare the similarities and differences between the database and data warehouse. (8) ii) Explain what data visualization is. How it helps in data warehousing. (7)	Evaluate	BTL-5
7.	i) Depict the 3 tier data warehousing architecture and explain its features in Detail. (8) ii).Explain the different types of OLAP servers (7)	Create	BTL-6

TEXT / REFERENCE BOOKS

1. Jiawei Han and Micheline Kamber, “Data Mining Concepts and Techniques”, 2nd Edition, Elsevier, 2007

2. Alex Berson and Stephen J. Smith, “ Data Warehousing, Data Mining & OLAP”, Tata McGraw Hill, 2007.

3. Pang-Ning Tan, Michael Steinbach and Vipin Kumar, “Introduction To Data Mining”, Person Education, 2007.

4. K.P. Soman, Shyam Diwakar and V. Ajay, “Insight into Data mining Theory and Practice”, Easter Economy Edition,
Prentice Hall of India, 2006.

5. G. K. Gupta, “Introduction to Data Mining with Case Studies”, Easter Economy Edition, Prentice Hall of India, 2006.

6. Daniel T.Larose, “Data Mining Methods and Models”, Wile-Interscience, 2006

UNIT – III - ASSOCIATION RULE MINING- SCSA3001

ASSOCIATION RULE MINING

Mining frequent patterns - Associations and correlations - Mining methods - Finding Frequent itemset using Candidate Generation - Generating Association Rules from Frequent Itemsets - Mining Frequent itemset without Candidate Generation-Mining various kinds of association rules - Mining Multi-Level Association Rule-Mining Multi-Dimensional Association Rule Mining Correlation analysis - Constraint based association mining.

MINING FREQUENT PATTERNS

- Frequent patterns are patterns (e.g., itemsets, subsequences, or substructures) that appear frequently in a data set. For example, a set of items, such as milk and bread that appear frequently together in a transaction data set is a frequent itemset.
- A subsequence, such as buying first a PC, then a digital camera, and then a memory card, if it occurs frequently in a shopping history database, is a (frequent) sequential pattern.
- A substructure can refer to different structural forms, such as subgraphs, subtrees, or sublattices, which may be combined with itemsets or subsequences. If a substructure occurs frequently, it is called a (frequent) structured pattern

Applications

Market Basket Analysis: given a database of customer transactions, where each transaction is a set of items the goal is to find groups of items which are frequently purchased together.

Telecommunication (each customer is a transaction containing the set of phone calls)

Credit Cards/ Banking Services (each card/account is a transaction containing the set of customer's payments)

Medical Treatments (each patient is represented as a transaction containing the ordered set of diseases)

Basketball-Game Analysis (each game is represented as a transaction containing the ordered set of ball passes)

Association Rule Definitions

✓ $I=\{i1, i2, ..., in\}$: a set of all the items

✓ Transaction T: a set of items such that $T \subseteq I$

- ✓ Transaction Database *D*: a set of transactions
- ✓ A transaction $T \subseteq I$ contains a set $A \subseteq I$ of some items, if $A \subseteq T$
- ✓ <u>*An Association Rule*</u>: is an implication of the form $A \Rightarrow B$, where $A, B \subseteq I$

It has two measures

1. ***Support***
2. ***Confidence***

- The rule $A \Rightarrow B$ holds in the transaction set *D* with **support** ***s*****,** where *s* is the percentage of transactions in *D* that contain *A*[*B* (i.e., the *union* of sets *A* and *B*, or say, both *A* and *B*). This is taken to be the probability, *P*(*AUB*)
- The rule $A \Rightarrow B$ has **confidence** ***c*** in the transaction set *D*, where *c* is the percentage of transactions in *D* containing *A* that also contain *B*. This is taken to be the conditional probability, *P*(*B*|*A*).

$$\text{Support (A)} = \frac{\text{Number of transaction in which A appears}}{\text{Total number of transactions}}$$

$$\text{Confidence } (A \longrightarrow B) = \frac{\text{Support}(AUB)}{\text{Support}(A)}$$

Examples

Rule form: —Body ® Head [support, confidence]‖.

buys(x, —diapers‖) ® buys(x, —beers‖) [0.5%, 60%]

major(x, —CS‖) ^ takes(x, —DB‖) ® grade(x, —A‖) [1%, 75%]

ASSOCIATIONS AND CORRELATIONS

Association Rule: Basic Concepts

Given: (1) database of transactions, (2) each transaction is a list of items (purchased by a customer in a visit)

Find: all rules that correlate the presence of one set of items with that of another set of items

E.g., 98% of people who purchase tires and auto accessories also get automotive services done

Applications

* ⇒Maintenance Agreement (What the store should do to boost Maintenance

Agreement sales)

– Home Electronics ⇒* (What other products should the store stocks up?)

– Attached mailing in direct marketing

– Detecting - ping-pongling of patients, faulty – collisions

Rule Measures: Support and Confidence

Find all the rules X & Y⇒Z with minimum confidence and support

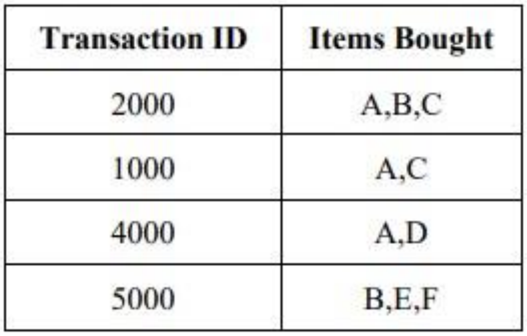
– Support, s, probability that a transaction contains {X 4 Y 4 Z}

– Confidence, c, conditional probability that a transaction having {X 4 Y} also contains Z

Let minimum support 50%, and minimum confidence 50%, we have

A ⇒ C (50%, 66.6%)

C ⇒A (50%, 100%)



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

Table 3.1

Association Rule Mining: A Road Map

• Boolean vs. quantitative associations (Based on the types of values handled)

– buys(x, ―SQLServer‖) ^ buys(x, ―DMBook‖) ® buys(x, ―DBMiner‖) [0.2%, 60%]

– age(x, ―30..39‖) ^ income(x, ―42..48K‖) ® buys(x, ―PC‖) [1%, 75%]

• Single dimension vs. multiple dimensional associations (see ex. above)

• Single level vs. multiple-level analysis

– What brands of beers are associated with what brands of diapers?

• Various extensions

– Correlation, causality analysis

• Association does not necessarily imply correlation or causality

– Maxpatterns and closed itemsets

– Constraints enforced

• E.g., small sales (sum < 100) trigger big buys (sum > 1,000)?

Market – Basket analysis

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

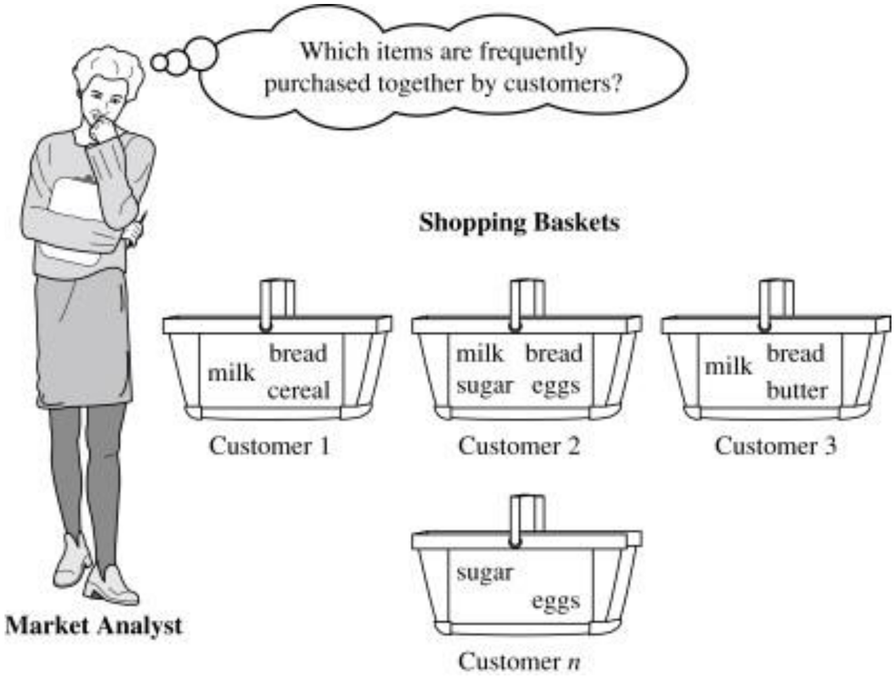


Figure 3.1 Market Basket Analysis

Computer ⇒ financial_management_ software [support = 2%, confidence = 60%]

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

MINING METHODS

- Mining Frequent Pattern with candidate generation
- Mining Frequent Pattern without candidate generation

MINING FREQUENT PATTERNS WITH CANDIDATE GENERATION

The method that mines the complete set of frequent item sets with candidate generation.

Apriori property & The Apriori Algorithm. Apriori property

- All nonempty subsets of a frequent item set most also be frequent.
  - – An item set I does not satisfy the minimum support threshold, min-sup, then I is not frequent, i.e., support (I) < min-sup
  - – If an item A is added to the item set I then the resulting item set (I U A) cannot occur more frequently than I.
- Monotonic functions are functions that move in only one direction.
- This property is called anti-monotonic.
- If a set cannot pass a test, all its supersets will fail the same test as well.
- This property is monotonic in failing the test.

The Apriori Algorithm

- ***Join Step:*** Ck is generated by joining Lk-1with itself
- ***Prune Step:*** Any (k-1)-itemset that is not frequent cannot be a subset of a frequent kitemset

Method

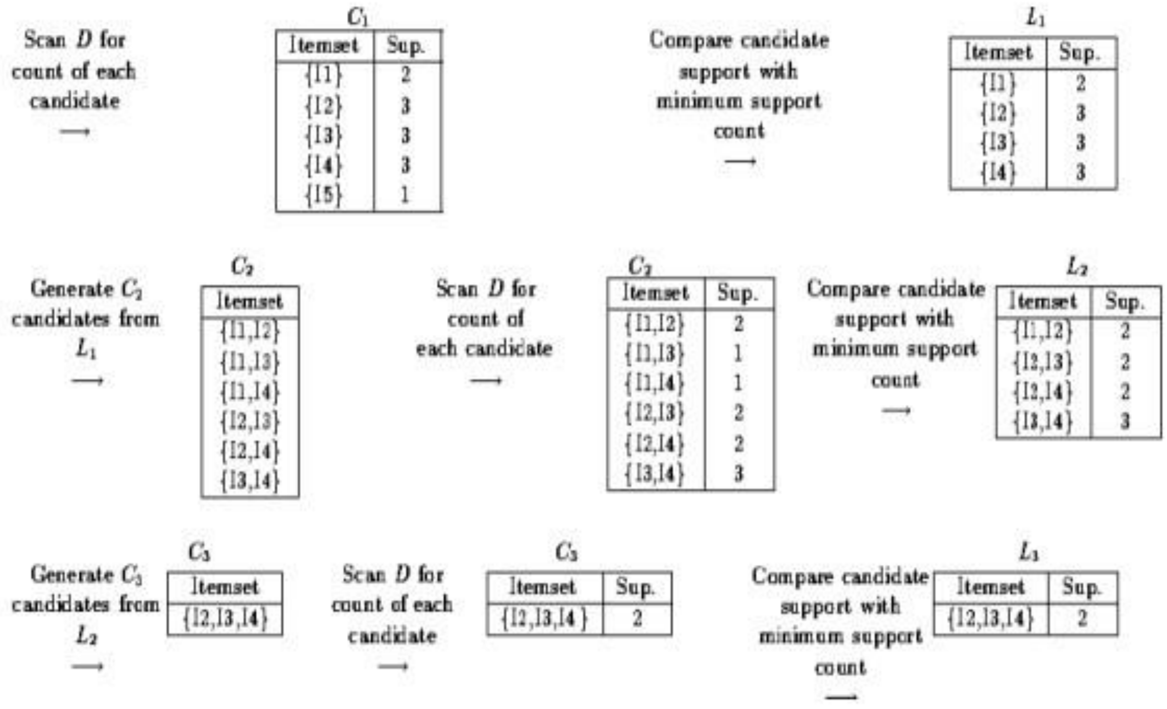
```
L1 = find_frequent_1 itemsets(D);
        for (k = 2; $L_{k-1}\phi\neq$; k++) {
        $C_k$ = apriori_gen ($L_{k-1}$, min_sup);
                For each transaction t$\in$ D { // scan D for counts
                Ct = subset ($C_k$, t); // get the subsets of t that are candidates
        for each candidate c $\in$ Ct
        c.count++;
        }
$L_k$ = {c $\in$ Ck|c.count $\geq$ min_sup}
}
return L = $U_k L_L$;
```

Procedure a priori_gen (Lk-1: frequent (k=t) itemsets; min_sup; minimum support)

```
for each itemset $l_1 \in L_{k-1}$
        for each itemset $l_2 \in L_{k-1}$
If $(l_1[1] = l_2[1]) \wedge (l_1[2] = l_2[2]) \wedge \ldots \wedge (l_1[k-2] = l_2[k-2]) \wedge (l_1[k-1] < l_2[k-1])$ that {
c = $l_1 \times l_2$; // join step: generate candidates
        if has_infrequent_subset (c, $L_{k-1}$) then
                Delete c, // prune step: remove unfruitful candidate
                else add c to $C_k$;
        }
Return Ck;
```

Procedure has_infrequent_subset (c: candidate k itemsetm; L_{k-1}: frequent (k-1) itemsets); // use prior knowledge

```
for each (k-1) subset s of c
        if $s \notin L_{k-1}$ then
return TRUE;
        return FALSE;
```



Scan D for count of each candidate →

C_1

Itemset	Sup.
{I1}	2
{I2}	3
{I3}	3
{I4}	3
{I5}	1

Compare candidate support with minimum support count →

L_1

Itemset	Sup.
{I1}	2
{I2}	3
{I3}	3
{I4}	3

Generate C_2 candidates from L_1 →

C_2

Itemset
{I1,I2}
{I1,I3}
{I1,I4}
{I2,I3}
{I2,I4}
{I3,I4}

Scan D for count of each candidate →

C_2

Itemset	Sup.
{I1,I2}	2
{I1,I3}	1
{I1,I4}	1
{I2,I3}	2
{I2,I4}	2
{I3,I4}	3

Compare candidate support with minimum support count →

L_2

Itemset	Sup.
{I1,I2}	2
{I2,I3}	2
{I2,I4}	2
{I3,I4}	3

Generate C_3 candidates from L_2 →

C_3

Itemset
{I2,I3,I4}

Scan D for count of each candidate →

C_3

Itemset	Sup.
{I2,I3,I4}	2

Compare candidate support with minimum support count →

L_3

Itemset	Sup.
{I2,I3,I4}	2

Table 3.1 Mining Frequent Patterns with candidate Generation

MINING FREQUENT ITEM SET WITHOUT CANDIDATE GENERATION

Frequent Pattern Growth Tree Algorithm

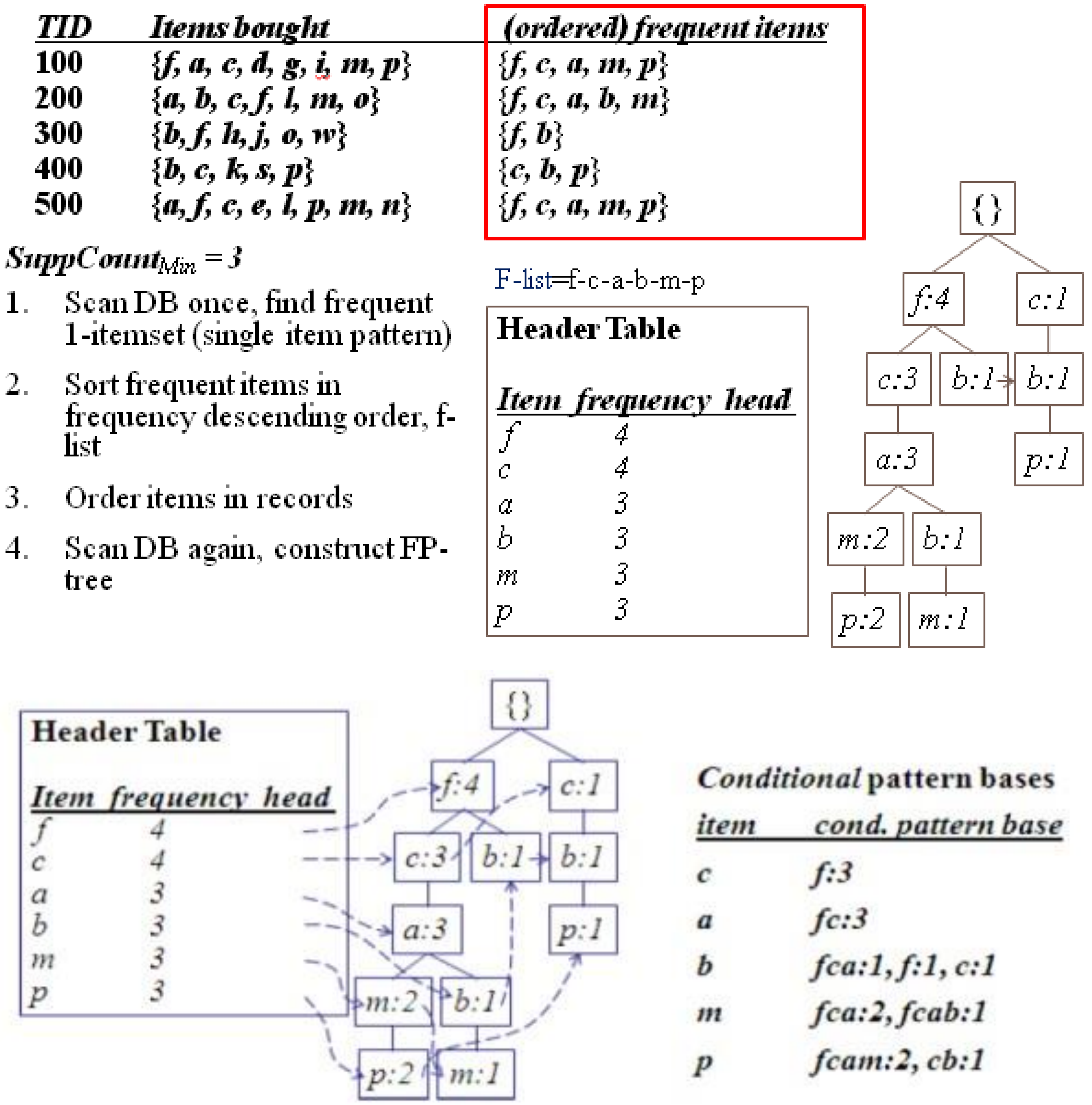
It grows long patterns from short ones using local frequent items

• "abc" is a frequent pattern

• Get all transactions having "abc": DB|abc

• "d" is a local frequent item in DB | abc € abcd is a frequent pattern

Construct FP-tree from a Transaction Database



TID	Items bought	(ordered) frequent items
100	{f, a, c, d, g, i, m, p}	{f, c, a, m, p}
200	{a, b, c, f, l, m, o}	{f, c, a, b, m}
300	{b, f, h, j, o, w}	{f, b}
400	{b, c, k, s, p}	{c, b, p}
500	{a, f, c, e, l, p, m, n}	{f, c, a, m, p}

SuppCount$_{Min}$ = 3

1. Scan DB once, find frequent 1-itemset (single item pattern)
2. Sort frequent items in frequency descending order, f-list
3. Order items in records
4. Scan DB again, construct FP-tree

F-list=f-c-a-b-m-p

Header Table

Item	frequency	head
f	4	
c	4	
a	3	
b	3	
m	3	
p	3	

Header Table

Item	frequency	head
f	4	
c	4	
a	3	
b	3	
m	3	
p	3	

Conditional **pattern bases**

item	cond. pattern base
c	f:3
a	fc:3
b	fca:1, f:1, c:1
m	fca:2, fcab:1
p	fcam:2, cb:1

Table 3.2 Construct FP-tree from a Transaction Database

Find Patterns Having P from P-conditional Database

• Starting at the frequent item header table in the FP-tree

• Traverse the FP-tree by following the link of each frequent item P

• Accumulate all of transformed prefix paths of items p to form P's conditional pattern base

Benefits of the FP-tree Structure

• Completeness:

– Never breaks a long pattern of any transaction

– Preserves complete information for frequent pattern mining

• Compactness

– Reduce irrelevant information—infrequent items are gone

– Frequency descending ordering: more frequent items are more likely to be shared

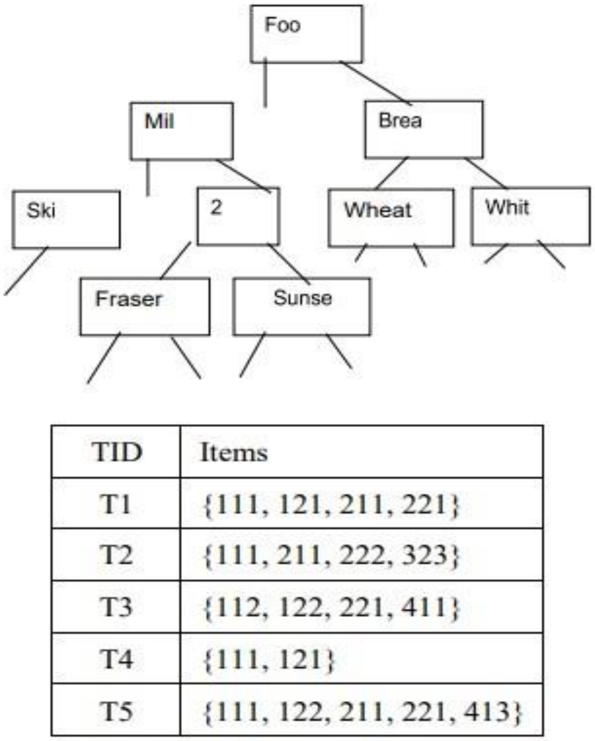
– Never be larger than the original database (if not count node-links and counts)

MINING VARIOUS KINDS OF ASSOCIATION RULES

• Mining Multi-level association rule

• Mining Multi dimensional Association Rule

Mining multilevel association rules from transactional databases.



TID	Items
T1	{111, 121, 211, 221}
T2	{111, 211, 222, 323}
T3	{112, 122, 221, 411}
T4	{111, 121}
T5	{111, 122, 211, 221, 413}

Figure 3.1 Mining multilevel association rules from transactional databases.

• Items often form hierarchy.

• Items at the lower level are expected to have lower support.

• Rules regarding itemsets at appropriate levels could be quite useful.

• Transaction database can be encoded based on dimensions and levels

• We can explore shared multi-level mining

MINING MULTI-LEVEL ASSOCIATIONS

• A top_down, progressive deepening approach:

– First find high-level strong rules:

milk ® bread [20%, 60%].

– Then find their lower-level ―weaker‖ rules:

2% milk ® wheat bread [6%, 50%].

• Variations at mining multiple-level association rules.

– Level-crossed association rules:

2% milk ® Wonder wheat bread

– Association rules with multiple, alternative hierarchies:

2% milk ® Wonder bread

Multi-level Association: Uniform Support vs. Reduced Support

• ***Uniform Support***: the same minimum support for all levels

– + One minimum support threshold. No need to examine itemsets containing any item whose ancestors do not have minimum support.

– Lower level items do not occur as frequently. If support threshold

• too high ⇒miss low level associations

• too low ⇒ generate too many high level associations

• ***Reduced Support:*** reduced minimum support at lower levels

– There are 4 search strategies:

• Level-by-level independent

• Level-cross filtering by k-itemset

• Level-cross filtering by single item

• Controlled level-cross filtering by single item

Multi-level Association: Redundancy Filtering

• Some rules may be redundant due to ―ancestor‖ relationships between items.

• Example

– milk ⇒ wheat bread [support = 8%, confidence = 70%]

– 2% milk ⇒wheat bread [support = 2%, confidence = 72%]

• We say the first rule is an ancestor of the second rule.

• A rule is redundant if its support is close to the ―expected‖ value, based on the rule's ancestor

Multi-Level Mining: Progressive Deepening

• A top-down, progressive deepening approach:

✓ First mine high-level frequent items: milk (15%), bread (10%)

✓ Then mine their lower-level ―weaker‖ frequent itemsets: 2% milk (5%), wheat bread (4%)

• Different min_support threshold across multi-levels lead to different algorithms:

✓ If adopting the same min_support across multi-levels then toss t if any of t's ancestors is infrequent.

✓ If adopting reduced min_support at lower levels then examine only those descendents whose ancestor's support is frequent/non-negligible

Mining Multidimensional Association mining

Mining our *AllElectronics* database, we may discover the Boolean association rule

buys (X, "digital camera") ⇒buys(X, "HP printer").

Following the terminology used in multidimensional databases,

Single- dimensional or intradimensional associations rule because it contains a single distinct predicate (e.g., buys) with multiple occurrences (i.e., the predicate occurs more than once within the rule). Such rules are commonly mined from transactional data.

Considering each database attribute or warehouse dimension as a predicate, we can therefore mine association rules containing multiple predicates such as

age(X, "20 . . . 29") occupation(X, "student") ⇒buys(X, "laptop").

Association rules that involve two or more dimensions or predicates can be referred to as multidimensional association rules. Rule contains three predicates (age, occupation, and buys), each of which occurs only once in the rule. Hence, we say that it has no repeated predicates.

Multidimensional association rules with no repeated predicates are called inter dimensional association rules. We can also mine multidimensional association rules with repeated predicates, which contain multiple occurrences of some predicates. These rules are called hybrid-dimensional association rules.

An example of such a rule is the following, where the predicate buys is repeated:

age(X, "20 . . . 29") ⇒buys(X, "laptop") ⇒buys(X, "HP printer").

Database attributes can be nominal or quantitative. The values of nominal (or categorical) attributes are "names of things." Nominal attributes have a finite number of possible values, with no ordering among the values (e.g., occupation, brand, color)

Quantitative attributes are numeric and have an implicit ordering among values (e.g., age, income, price). Techniques for mining multidimensional association rules can be categorized into two basic approaches regarding the treatment of quantitative attributes. In the first approach, quantitative attributes are discretized using predefined concept hierarchies. This discretization occurs before mining. For instance, a concept hierarchy for income may be used to replace the original numeric values of this attribute by interval labels such as "0..20K," "21K..30K," "31K..40K," and so on.

Here, discretization is static and predetermined. Chapter 3 on data preprocessing gave several techniques for discretizing numeric attributes. The discretized numeric attributes, with their interval labels, can then be treated as nominal attributes (where each interval is considered a category).

Mining Quantitative Association Rules

- Determine the number of partitions for each quantitative attribute
- Map values/ranges to consecutive integer values such that the order is preserved
- Find the support of each value of the attributes, and combine when support is less than MaxSup. Find frequent itemsets, whose support is larger than MinSup
- Use frequent set to generate association rules
- Pruning out uninteresting rules

Partial Completeness

- R : rules obtained before partition
- R': rules obtained after partition
- Partial Completeness measures the maximum distance between a rule in R and its closest generalization in R'
- $\grave{X}$ is a generalization of itemset X: if

∈ x ∀attributes (X) [∈ x ∀< x, l, u X ^ x, l', u' ∈> l' ≤ l ≤ u ≤ u'] ⇒X∈>

- The distance is defined by the ratio of support

K-Complete

- C: the set of frequent itemsets
- For any K ≥ 1, P is K-complete w.r.t C if:

1. P C
2. For any itemset X (or its subset) in C, there exists a generalization whose support is no more than K times that of X (or its subset)

- The smaller K is, the less the information lost

CORRELATION ANALYSIS

- Interest (correlation, lift)

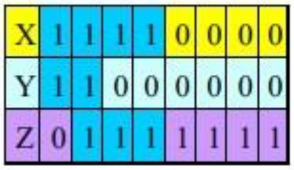
– taking both P(A) and P(B) in consideration

– P(A^B)=P(B)*P(A), if A and B are independent events

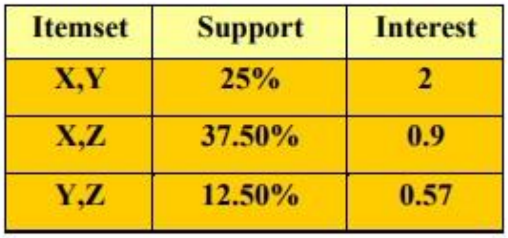
– A and B negatively correlated, if the value is less than 1; otherwise A and B positively correlated

X2 Correlation

- X2 measures correlation between categorical attributes

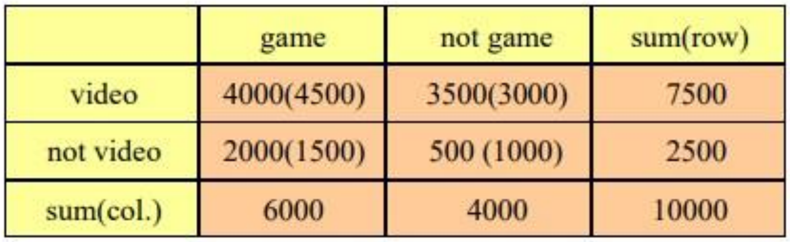


X	1	1	1	1	0	0	0	0
Y	1	1	0	0	0	0	0	0
Z	0	1	1	1	1	1	1	1



Itemset	Support	Interest
X,Y	25%	2
X,Z	37.50%	0.9
Y,Z	12.50%	0.57

$$X =^{2} = \sum \frac{(observe\_expe\ cted)^{2}}{exp\ ected}$$



	game	not game	sum(row)
video	4000(4500)	3500(3000)	7500
not video	2000(1500)	500 (1000)	2500
sum(col.)	6000	4000	10000

Table 3.2 Correlation

- Expected (i,j) = count(row i) * count(column j) / N
- X2 = (4000 - 4500)2 / 4500 - (3500 - 3000)2 / 3000 - (2000 - 1500)2 / 1500 - (500 -1000)2 / 1000 = 555.6
- X2 > 1 and observed value of (game, video) < expected value, there is a negative correlation

Numeric correlation

• ***Correlation concept in statistics***

– Used to study the relationship existing between 2 or more numeric variables

– A correlation is a measure of the linear relationship between variables Ex: number of hours spent studying in a class with grade received

– Outcomes:

• → positively related

• → Not related

• → negatively related

– Statistical relationships

• Covariance

• Correlation coefficient

CONSTRAINT-BASED ASSOCIATION MINING

• ***Interactive, exploratory mining giga-bytes of data?***

– Could it be real? — Making good use of constraints!

• ***What kinds of constraints can be used in mining?***

– Knowledge type constraint: classification, association, etc.

• ***Find product pairs sold together in Vancouver in Dec.'98.***

– Dimension/level constraints:

• ***in relevance to region, price, brand, customer category.***

– Rule constraints

• ***small sales (price < $10) triggers big sales (sum > $200).***

– Interestingness constraints:

• ***Strong rules (min_support ≥ 3%, min_confidence ≥ 60%).***

Rule Constraints in Association Mining

• Two kind of rule constraints:

– Rule form constraints: meta-rule guided mining.

• P(x, y) ^ Q(x, w) ® takes(x, ―database systems‖).

– Rule (content) constraint: constraint-based query optimization (Ng, et al., SIGMOD'98).

• sum (LHS) < 100 ^ min(LHS) > 20 ^ count(LHS) > 3 ^ sum(RHS) > 1000

• 1-variable vs. 2-variable constraints (Lakshmanan, et al. SIGMOD'99):

– 1-var: A constraint confining only one side (L/R) of the rule, e.g., as shown above.

– 2-var: A constraint confining both sides (L and R).

• sum (LHS) < min(RHS) ^ max(RHS) < 5* sum(LHS)

Constrain-Based Association Query

• Database: (1) trans (TID, Itemset), (2) itemInfo (Item, Type, Price)

• A constrained asso. Query (CAQ) is in the form of {(S1, S2) |C },

– Where C is a set of constraints on S1, S2 including frequency constraint

• A classification of (single-variable) constraints:

– Class constraint: : S ⊂ A. e.g. S ⊂ Item

Constrained Association Query Optimization Problem

• Given a CAQ = { (S1, S2) | C }, the algorithm should be :

– Sound: It only finds frequent sets that satisfy the given constraints C

– Complete: All frequent sets satisfy the given constraints C are found

• A naïve solution:

– Apply Apriori for finding all frequent sets, and then to test them for constraint satisfaction one by one.

• Our approach:

– Comprehensive analysis of the properties of constraints and try to push them as deeply as possible inside the frequent set computation.

Categories of Constraints

1. Anti-monotone and Monotone Constraints

• Constraint Ca is anti-monotone iff. for any pattern S not satisfying Ca, none of the super patterns of S can satisfy Ca

• A constraint Cm is monotone iff. for any pattern S satisfying Cm, every super-pattern of S also satisfies it

2. Succinct Constraint

• A subset of item Is is a succinct set, if it can be expressed as σp (I) for some selection predicate p, where σ is a selection operator

• SP⊆2I is a succinct power set, if there is a fixed number of succinct set I1, …, Ik ⊆I,s.t. SP can be expressed in terms of the strict power sets of I1, …, Ik using union and minus

• A constraint Cs is succinct provided SATCs (I) is a succinct power set

3. Convertible Constraint

• Suppose all items in patterns are listed in a total order R

• A constraint C is convertible anti-monotone iff a pattern S satisfying the constraint implies that each suffix of S w.r.t. R also satisfies C

• A constraint C is convertible monotone iff a pattern S satisfying the constraint implies that each pattern of which S is a suffix w.r.t. R also satisfies C

Property of Constraints: Anti-Monotone

• Anti-monotonicity: If a set S violates the constraint, any superset of S violates the constraint.

• Examples:

– sum(S.Price) ≤v is anti-monotone

– sum(S.Price) ≥v is not anti-monotone

– sum(S.Price) = v is partly anti-monotone

• Application:

– Push ―sum(S.price) ≤ 1000‖ deeply into iterative frequent set computation.

Property of Constraints: Succinctness

• Succinctness:

– For any set S1 and S2 satisfying C, S1 S2 satisfies C

– Given A1 is the sets of size 1 satisfying C, then any set S satisfying C are based on A1 , i.e., it contains a subset belongs to A1 ,

- Example :
  - – sum(S.Price) ≥v is not succinct
  - – min(S.Price) ≤v is succinct

Optimization:

– If C is succinct, then C is pre-counting prunable. The satisfaction of the constraint alone is not affected by the iterative support counting.

- ed based on the training set
- Unsupervised learning (clustering)
- The class labels of training data is unknown
- Given a set of measurements, observations, etc. with the aim of establishing the existence of classes or clusters in the data.

PART-A			
Q. No	**Questions**	**Competence**	**BT Level**
1.	Define association and correlations.	Remember	BTL-1
2.	List the ways in which interesting patterns should be mined.	Remember	BTL-1
3.	Are all patterns generated are interesting and useful? Give reasons to justify	Understand	BTL-2
4.	Compare the advantages of FP growth algorithm over Apriori algorithm	Analyze	BTL-4
5.	How will you apply FP growth algorithm in Data mining?	Apply	BTL-3
6.	How will you Apply pattern mining in Multilevel space?	Apply	BTL-3
7.	Analyze the constraint based frequent pattern mining.	Analyze	BTL-4
8.	Evaluate the classification using Frequent patterns	Evaluate	BTL-5
9.	Generalize on Mining Closed and Max Patterns.	Create	BTL-6
10.	Define correlation and market basket analysis.	Remember	BTL-1

PART-B			
Q. No	**Questions**	**Competence**	**BT Level**
1.	Define Market Basket Analysis. Describe about Frequent Itemsets, Closed Itemset and Association Rules.	Remember	BTL-1
2.	Discuss about constraint based association rule mining with examples and state how association mining to correlation analysis is dealt with.	Apply	BTL-3

3.	Find all frequent item sets for the given training set using Apriori and FP growth respectively. Compare the efficiency of the two mining processes (13)
	TID ITEMS BROUGHT
	T100 {M , O, N , K , E , Y }
	T200 {D , O, N, K , E, Y }
	T300 {M , A K, E }
	T400 {M ,U , C , K ,Y }
	T500 {C , O , O ,K , I , E }
		Apply	BTL-3

4.	i) How would you summarize in detail about mining methods? (6) ii) Summarize in detail about various kinds of association rules. (7)	Understand	BTL-2
5.	i) What is interestingness of a pattern? (5) ii) Summarize the various classification methods using frequent patterns. (10)	Create	BTL-6
6.	Analyze the various Frequent Itemset mining method with examples.	Analyze	BTL-4
7.	Generalize how pattern mining is done in multilevel and multidimensional space with necessary examples.	Create	BTL-6

TEXT / REFERENCE BOOKS

1. Jiawei Han and Micheline Kamber, “Data Mining Concepts and Techniques”, 2nd Edition, Elsevier, 2007

2. Alex Berson and Stephen J. Smith, “ Data Warehousing, Data Mining & OLAP”, Tata McGraw Hill, 2007.

3. Pang-Ning Tan, Michael Steinbach and Vipin Kumar, “Introduction To Data Mining”, Person Education, 2007.

4. K.P. Soman, Shyam Diwakar and V. Ajay, “Insight into Data mining Theory and Practice”, Easter Economy Edition,
Prentice Hall of India, 2006.

5. G. K. Gupta, “Introduction to Data Mining with Case Studies”, Easter Economy Edition, Prentice Hall of India, 2006.

6. Daniel T.Larose, “Data Mining Methods and Models”, Wile-Interscience, 2006

UNIT – IV - CLASSIFICATION AND PREDICTION- SCSA3001

CLASSIFICATION AND PREDICTION

Classification and prediction - Issues Regarding Classification and Prediction - Classification by Decision Tree Induction -Bayesian classification - Baye's Theorem - Naïve Bayesian Classification - Bayesian Belief Network - Rule based classification - Classification by Back propagation - Support vector machines - Prediction - Linear Regression.

CLASSIFICATION AND PREDICTION

* Used for prediction (future analysis) to know the unknown attributes with their values by using classifier algorithms and decision tree. (In data mining)

* Which constructs some models (like decision trees) then which classifies the attributes.

* Already we know the types of attributes are 1.categorical attribute and 2.numerical attribute

* These classification can work on both the above mentioned attributes.

Prediction: prediction also used for to know the unknown or missing values.

1. Which also uses some models in order to predict the attributes
2. Models like neural networks, if else rules and other mechanisms

Classification and prediction are used in the Applications like

*Credit approval

*Target marketing

*Medical diagnosis

Classification—A Two-Step Process

- Model construction: describing a set of predetermined classes

– Each tuple/sample is assumed to belong to a predefined class, as determined by the class label attribute

– The set of tuples used for model construction: training set

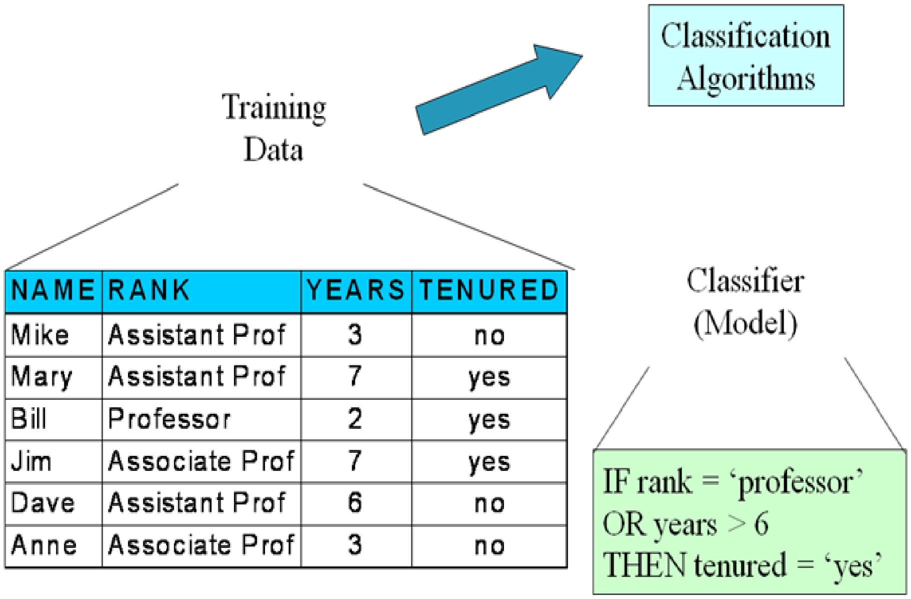
– The model is represented as classification rules, decision trees, or mathematical formulae

- Model usage: for classifying future or unknown objects

– Estimate accuracy of the model

- The known label of test sample is compared with the classified result from the model
- Accuracy rate is the percentage of test set samples that are correctly classified by the model
- Test set is independent of training set, otherwise over-fitting will occur

Process (1): Model Construction

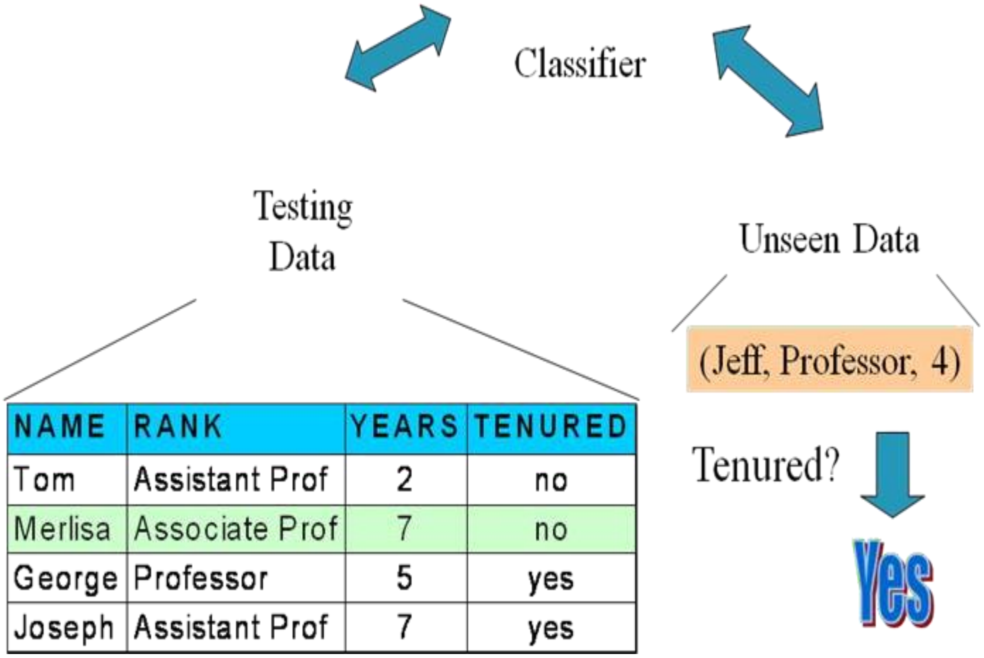


NAME	RANK	YEARS	TENURED
Mike	Assistant Prof	3	no
Mary	Assistant Prof	7	yes
Bill	Professor	2	yes
Jim	Associate Prof	7	yes
Dave	Assistant Prof	6	no
Anne	Associate Prof	3	no

IF rank = 'professor'
OR years > 6
THEN tenured = 'yes'

Figure 4.1 Model Construction

Process (2): Using the Model in Prediction



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

Figure 4.2 Model in Prediction

Supervised vs. Unsupervised Learning

- ***Supervised learning (classification)***
  - Supervision: The training data (observations, measurements, etc.) are accompanied by labels indicating the class of the observations
  - New data is classified based on the training set
- ***Unsupervised learning (clustering)***
  - The class labels of training data is unknown
  - Given a set of measurements, observations, etc. with the aim of establishing the existence of classes or clusters in the data

ISSUES REGARDING CLASSIFICATION AND PREDICTION

There are two issues regarding classification and prediction they are

Issues (1): Data Preparation

Issues (2): Evaluating Classification Methods

Issues (1): Data Preparation: Issues of data preparation includes the following

1) Data cleaning

Preprocess data in order to reduce noise and handle missing values (refer preprocessing techniques i.e. data cleaning notes)

2) Relevance analysis (feature selection)

Remove the irrelevant or redundant attributes (refer unit-iv AOI Relevance analysis)

3) Data transformation (refer preprocessing techniques i.e data cleaning notes) Generalize and/or normalize data

Issues (2): Evaluating Classification Methods: considering classification methods should satisfy the following properties

1. Predictive accuracy

2. Speed and scalability

   ✓ Time to construct the model

   ✓ Time to use the model

3. Robustness

   ✓ Handling noise and missing values

4. Scalability

   ✓ Efficiency in disk-resident databases

5. Interpretability:

- ✓ Understanding and insight provided by the model

6. Goodness of rules

- ✓ Decision tree size
- ✓ Compactness of classification rules

Comparing Classification Methods

Classification and prediction methods can be compared and evaluated according to the following criteria:

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

Speed: This refers to the computation costs involved in generating and using the model.

Robustness: This is the ability of the model to make correct predictions given noisy data or data with missing values.

Scalability: This refers to the ability to construct the model efficiently given large amount of data.

Interpretability: This refers to the level of understanding and insight that is provided by the model

CLASSIFICATION BY DECISION TREE INDUCTION

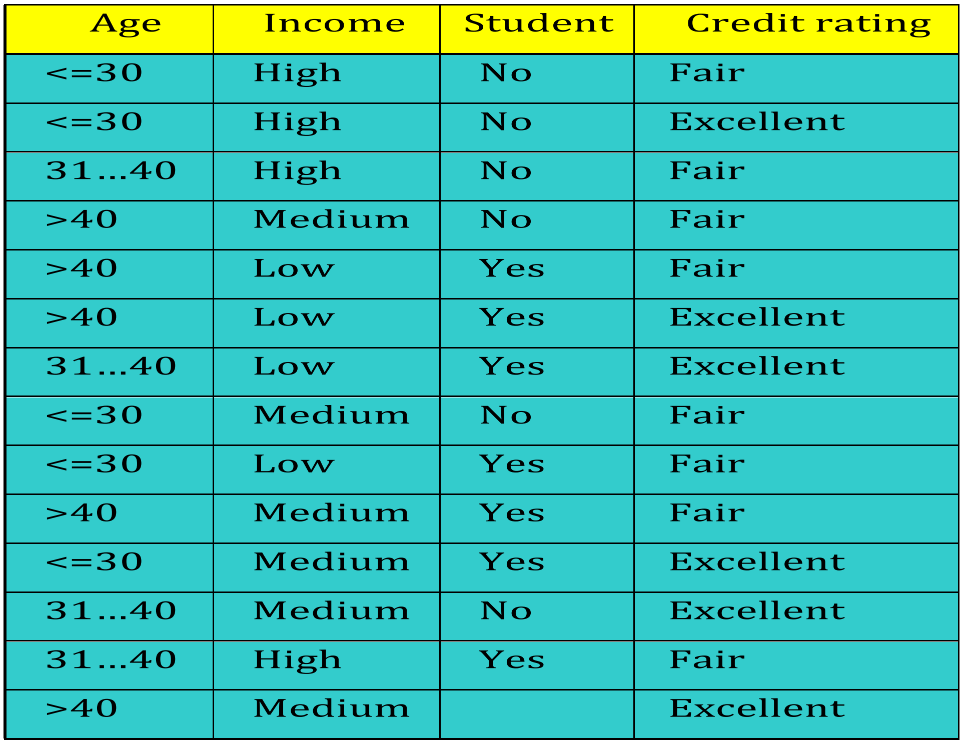
Decision tree

- – A flow-chart-like tree structure
- – Internal node denotes a test on an attribute
- – Branch represents an outcome of the test
- – Leaf nodes represent class labels or class distribution

- ***Decision tree generation consists of two phases***
    - – ***Tree construction***
        - At start, all the training examples are at the root
        - Partition examples recursively based on selected attributes
    - – ***Tree pruning***
        - Identify and remove branches that reflect noise or outliers
- ***Use of decision tree: Classifying an unknown sample***
    - – Test the attribute values of the sample against the decision tree

Training Dataset

This follows an example from Quinlan's ID3



Age	Income	Student	Credit rating
<=30	High	No	Fair
<=30	High	No	Excellent
31…40	High	No	Fair
>40	Medium	No	Fair
>40	Low	Yes	Fair
>40	Low	Yes	Excellent
31…40	Low	Yes	Excellent
<=30	Medium	No	Fair
<=30	Low	Yes	Fair
>40	Medium	Yes	Fair
<=30	Medium	Yes	Excellent
31…40	Medium	No	Excellent
31…40	High	Yes	Fair
>40	Medium		Excellent

Table 4.1 Training Dataset

Algorithm for decision tree induction

- ***Basic algorithm (a greedy algorithm)***
  - Tree is constructed in a top-down recursive divide-and-conquer manner
  - At start, all the training examples are at the root
  - Attributes are categorical (if continuous-valued, they are discretized in advance)
  - Examples are partitioned recursively based on selected attributes
  - Test attributes are selected on the basis of a heuristic or statistical measure (e.g., information gain)
- ***Conditions for stopping partitioning***
  - All samples for a given node belong to the same class
  - There are no remaining attributes for further partitioning – majority voting is employed for classifying the leaf
  - There are no samples left

Extracting Classification Rules from Trees

- Represent the knowledge in the form of IF-THEN rules
- One rule is created for each path from the root to a leaf

- Each attribute-value pair along a path forms a conjunction
- The leaf node holds the class prediction
- Rules are easier for humans to understand

Example

IF *age* = “<=30” AND *student* = “*no*” THEN *buys_computer* = “*no*”

IF *age* = “<=30” AND *student* = “*yes*” THEN *buys_computer* = “*yes*”

IF *age* = “31…40” THEN *buys_computer* = “*yes*”

IF *age* = “>40” AND *credit_rating* = “*excellent*” THEN *buys_computer* = “*yes*”

IF *age* = “>40” AND *credit_rating* = “*fair*” THEN *buys_computer* = “*no*”

Avoid Overfitting in Classification

- The generated tree may overfit the training data
  - Too many branches, some may reflect anomalies due to noise or outliers
  - Result is in poor accuracy for unseen samples
- Two approaches to avoid over fitting

Prepruning:

✓ Halt tree construction early—do not split a node if this would result in the goodness measure falling below a threshold

✓ Difficult to choose an appropriate threshold

Post pruning:

✓ Remove branches from a “fully grown” tree—get a sequence of progressively pruned trees

✓ Use a set of data different from the training data to decide which the “best pruned tree”

Tree Mining in Weka and Tree Mining in Clementine.

Tree Mining in Weka

- Example:
  - Weather problem: build a decision tree to guide the decision about whether or not to play tennis.
  - Dataset (weather.nominal.arff)
- Validation:
  - Using training set as a test set will provide optimal classification accuracy.

- – Expected accuracy on a different test set will always be less.
- – 10-fold cross validation is more robust than using the training set as a test set.

- Divide data into 10 sets with about same proportion of class label values as in original set.
- Run classification 10 times independently with the remaining 9/10 of the set as the training set.
- Average accuracy.
  - – Ratio validation: 67% training set / 33% test set.
  - – Best: having a separate training set and test set.
- Results:
  - – Classification accuracy (correctly classified instances).
  - – Errors (absolute mean, root squared mean …)
  - – Kappa statistic (measures agreement between predicted and observed classification; -100%-100% is the proportion of agreements after chance agreement has been excluded; 0% means complete agreement by chance)
- Results:
  - – TP (True Positive) rate per class label
  - – FP (False Positive) rate
  - – Precision = TP rate = TP / (TP + FN)) * 100%
  - – Recall = TP / (TP + FP)) * 100%
  - – F-measure = 2* recall * precision / recall + precision
- **ID3 characteristics:**

– Requires nominal values

– Improved into C4.5

- Dealing with numeric attributes
- Dealing with missing values
- Dealing with noisy data
- Generating rules from trees

Tree Mining in Clementine

• Methods:

– C5.0: target field must be categorical, predictor fields may be numeric or categorical, provides multiple splits on the field that provides the maximum information gain at each level

– QUEST: target field must be categorical, predictor fields may be numeric ranges or categorical, statistical binary split

– C&RT: target and predictor fields may be numeric ranges or categorical, statistical binary split based on regression

– CHAID: target and predictor fields may be numeric ranges or categorical, statistical binary split based on chi-square

Attribute Selection Measures

- Information Gain
- Gain ratio
- Gini Index

Pruning of decision trees

Discarding one or more sub trees and replacing them with leaves simplify a decision tree, and that is the main task in decision-tree pruning. In replacing the sub tree with a leaf, the algorithm expects to lower the *predicted error rate and* increase the quality of a classification model. But computation of error rate is not simple. An error rate based only on a training data set does not provide a suitable estimate. One possibility to estimate the predicted error rate is to use a new, additional set of test samples if they are available, or to use the cross-validation techniques. This technique divides initially available samples into equal sized blocks and, for each block; the tree is constructed from all samples except this block and tested with a given block of samples. With the available training and testing samples, the basic idea of decision tree-pruning is to remove parts of the tree (sub trees) that do not contribute to the classification accuracy of unseen testing samples, producing a less complex and thus more comprehensible tree. There are two ways in which therecursive-partitioning method can be modified:

1. Deciding not to divide a set of samples any further under some conditions. The stopping criterion is usually based on some statistical tests, such as the $\chi 2$ test: If there are no significant differences in classification accuracy before and after division, then represent a current node as a leaf. The decision is made in advance, before splitting, and therefore this approach is called pre pruning.
2. Removing retrospectively some of the tree structure using selected accuracy criteria. The decision in this process of post pruning is made after the tree has been built.

C4.5 follows the *post pruning* approach, but it uses a specific technique to estimate the predicted error rate. This method is called *pessimistic pruning*. For every node in a tree, the estimation of the upper confidence limit u_{cf} is computed using the statistical tables for binomial distribution (given in most textbooks on statistics). Parameter U_{cf} is a function of $|T_i|$ and E for a given node. C4.5 uses the default confidence level of 25%, and compares $U_{25\%}$ ($|T_i|$ /E) for a given node T_i with a weighted confidence of its leaves. Weights are the total number of cases for every leaf. If the predicted error for a root node in a sub tree is less than weighted sum of $U_{25\%}$ for the leaves (predicted error for the sub tree), then a sub tree will be replaced with its root node, which becomes a new leaf in a pruned tree.

Let us illustrate this procedure with one simple example. A sub tree of a decision tree is given in Figure, where the root node is the test x_1 on three possible values {1, 2, 3} of the attribute A. The children of the root node are leaves denoted with corresponding classes and ($|T_i|$ /E) parameters. The question is to estimate the possibility of pruning the sub tree and replacing it with its root node as a new, generalized leaf node.

To analyze the possibility of replacing the sub tree with a leaf node it is necessary to compute a predicted error PE for the initial tree and for a replaced node. Using default confidence of 25%, the upper confidence limits for all nodes are collected from statistical tables: $U_{25\%}$ (6, 0) = 0.206, $U_{25\%}$(9, 0) = 0.143, $U_{25\%}$(1, 0) = 0.750, and $U_{25\%}$(16, 1) = 0.157. Using these values, the predicted errors for the initial tree and the replaced node are

PE_{tree} = 6.0.206 + 9.0.143 + 1.0.750 = 3.257

PE_{node} = 16.0.157 = 2.512

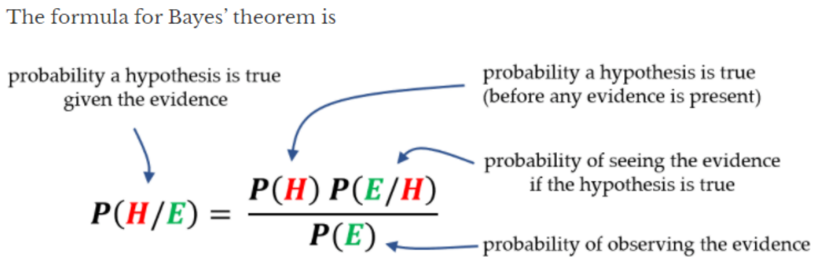
Since the existing subtree has a higher value of predicted error than the replaced node, it is recommended that the decision tree be pruned and the subtree replaced with the new leaf node.

BAYESIAN CLASSIFICATION

- ***Probabilistic learning:*** Calculate explicit probabilities for hypothesis, among the most practical approaches to certain types of learning problems
- ***Incremental:*** Each training example can incrementally increase/decrease the probability that a hypothesis is correct. Prior knowledge can be combined with observed data.
- ***Probabilistic prediction:*** Predict multiple hypotheses, weighted by their probabilities
- ***Standard:*** Even when Bayesian methods are computationally intractable, they can provide a standard of optimal decision making against which other methods can be measured

BAYESIAN THEOREM

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


The formula for Bayes' theorem is
probability a hypothesis is true given the evidence
probability a hypothesis is true (before any evidence is present)
probability of seeing the evidence if the hypothesis is true
probability of observing the evidence
$P(H/E) = \frac{P(H)\,P(E/H)}{P(E)}$

- MAP (maximum posteriori) hypothesis

$$\begin{aligned} h_{map} &= \underset{h \in H}{\arg\max}(P(h \mid D)) \\ &= \underset{h \in H}{\arg\max}\left(\frac{P(D \mid h)P(h)}{P(D)}\right) \\ &= \underset{h \in H}{\arg\max}(P(D \mid h)P(h)) \end{aligned}$$

- Practical difficulty: require initial knowledge of many probabilities, significant computational cost

Naïve Bayes Classifier (I)

- A simplified assumption: attributes are conditionally independent:

$$P(C_j \mid V) \infty P(C_j)\prod_{i=1}^{n} P(v_i \mid C_j)$$

- Greatly reduces the computation cost, only count the class distribution.

Naive Bayesian Classifier (II)

Given a training set, we can compute the probabilities

Outlook	P	N	Humidity	P	N
sunny	2/9	3/5	high	3/9	4/5
overcast	4/9	0	normal	6/9	1/5
rain	3/9	2/5			
Temperature			Windy		
hot	2/9	2/5	true	3/9	3/5
mild	4/9	2/5	false	6/9	2/5
cool	3/9	1/5			

Table 4.1 Training Dataset

BAYESIAN CLASSIFICATION

- The classification problem may be formalized using a-posteriori probabilities:
- P(C|X) = prob. that the sample tuple
- X=<x1,…,xk> is of class C.
- E.g. P(class=N | outlook=sunny, windy=true,…)
- Idea: assign to sample X the class label C such that P(C|X) is maximal

Estimating a-posteriori probabilities

- Bayes theorem:

  P(C|X) = P(X|C)·P(C) / P(X)
- P(X) is constant for all classes
- P(C) = relative freq of class C samples
- C such that P(C|X) is maximum = C such that P(X|C)·P(C) is maximum
- Problem: computing P(X|C) is unfeasible!

NAÏVE BAYESIAN CLASSIFICATION

- Naïve assumption: attribute independence

  P(x1,…,xk|C) =

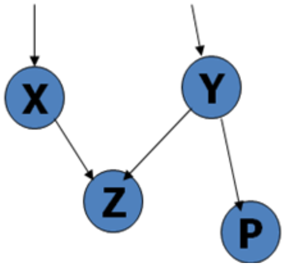
  P(x1|C)·…·P(xk|C)
- If i-th attribute is categorical:

  P(xi|C) is estimated as the relative freq of samples having value xi as i-th attribute in class C
- If i-th attribute is continuous:

  P(xi|C) is estimated thru a Gaussian density function
- Computationally easy in both cases

BAYESIAN BELIEF NETWORKS

- ■ Bayesian belief network allows a subset of the variables conditionally independent
- ■ A graphical model of causal relationships
  - ■ Represents dependency among the variables
  - ■ Gives a specification of joint probability distribution



❑ Nodes: random variables

❑ Links: dependency

❑ X and Y are the parents of Z, and Y is the parent of P

❑ No dependency between Z and P

❑ Has no loops or cycles

Figure 4.3 Bayesian Belief Networks

Bayesian belief network: an example

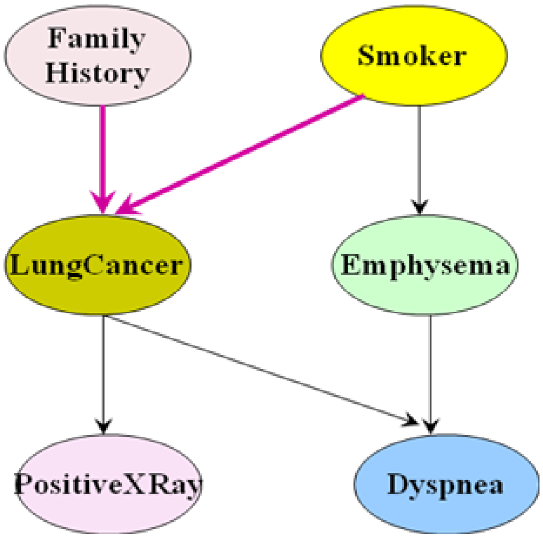
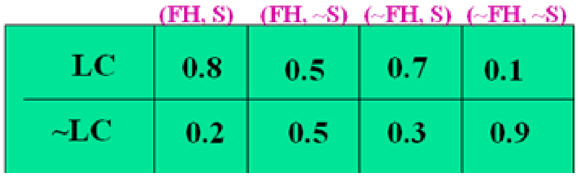


Figure 4.4 Bayesian Belief Networks

The conditional probability table (CPT) for variable Lung Cancer:



	(FH, S)	(FH, ~S)	(~FH, S)	(~FH, ~S)
LC	0.8	0.5	0.7	0.1
~LC	0.2	0.5	0.3	0.9

Table 4.2 conditional probability table

CPT shows the conditional probability for each possible combination of its parents

Derivation of the probability of a particular combination of values of X, from CPT:

$$P(x_1, \ldots, x_n) = \prod_{i=1}^{n} P(x_i \mid Parents(Y_i))$$

Association-Based Classification

- Several methods for association-based classification
  - ARCS: Quantitative association mining and clustering of association rules (Lent et al'97)
- It beats C4.5 in (mainly) scalability and also accuracy
  - Associative classification: (Liu et al'98)
- It mines high support and high confidence rules in the form of "cond_set => y", where y is a class label
  - CAEP (Classification by aggregating emerging patterns) (Dong et al'99)
    - Emerging patterns (EPs): the item sets whose support increases significantly from one class to another
    - Mine Eps based on minimum support and growth rate

RULE BASED CLASSIFICATION

Using IF-THEN Rules for Classification

- Represent the knowledge in the form of IF-THEN rules

R: IF *age* = youth AND *student* = yes THEN *buys_computer* = yes

- Rule antecedent/precondition vs. rule consequent
- Assessment of a rule: *coverage* and *accuracy*
  - n_{covers} = # of tuples covered by R
  - $n_{correct}$ = # of tuples correctly classified by R

coverage(R) = n_{covers} /|D| /* D: training data set */ accuracy(R) = $n_{correct}$ / n_{covers}

- If more than one rule is triggered, need conflict resolution
  - Size ordering: assign the highest priority to the triggering rules thathas the "toughest" requirement (i.e., with the *most attribute test*)
  - Class-based ordering: decreasing order of prevalence or misclassification cost per class
  - Rule-based ordering (decision list): rules are organized into one long priority list, according to some measure of rule quality or by experts

Rule Extraction from a Decision Tree

- Rules are easier to understand than large trees

- One rule is created for each path from the root to a leaf
- Each attribute-value pair along a path forms a conjunction: the leaf holds the class prediction
- Rules are mutually exclusive and exhaustive

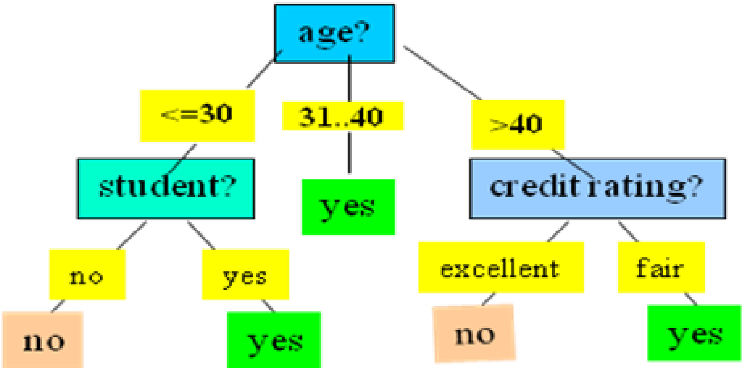


Figure 4.5 Decision Tree

- Example: Rule extraction from our buys_computer decision-tree

IF *age* = young AND *student* = *no* THEN *buys_computer* = *no*

IF *age* = young AND *student* = *yes* THEN *buys_computer* = *yes*

IF *age* = mid-age THEN *buys_computer* = *yes*

IF *age* = old AND *credit_rating* = *excellent* THEN *buys_computer* = *yes*

IF *age* = young AND *credit_rating* = *fair* THEN *buys_computer* = *no*

Rule Extraction from the Training Data

- Sequential covering algorithm: Extracts rules directly from training data
- Typical sequential covering algorithms: FOIL, AQ, CN2, RIPPER
- Rules are learned sequentially, each for a given class Ci will cover many tuples of Ci but none (or few) of the tuples of other classes
- Steps:
  - Rules are learned one at a time
  - Each time a rule is learned, the tuples covered by the rules are removed
  - The process repeats on the remaining tuples unless termination condition, e.g., when no more training examples or when the quality of a rule returned is below a user-specified threshold
- Comp. w. decision-tree induction: learning a set of rules simultaneously

CLASSIFICATION BY BACKPROPAGATION

- ***Back propagation:*** A neural network learning algorithm
- Started by psychologists and neurobiologists to develop and test computational analogues of neurons
- ***A neural network:*** A set of connected input/output units where each connection has a weight associated with it
- During the learning phase, the network learns by adjusting the weights so as to be able to predict the correct class label of the input tuples
- Also referred to as connectionist learning due to the connections between units

Neural network as a classifier

- ***Weakness***
  - Long training time
  - Require a number of parameters typically best determined empirically, e.g., the network topology or ``structure."
  - Poor interpretability: Difficult to interpret the symbolic meaning behind the learned weights and of ``hidden units" in the network
- ***Strength***
  - High tolerance to noisy data
  - Ability to classify untrained patterns
  - Well-suited for continuous-valued inputs and outputs
  - Algorithms are inherently parallel
  - Techniques have recently been developed for the extraction of rules from trained neural networks

A Neuron (= a perceptron)

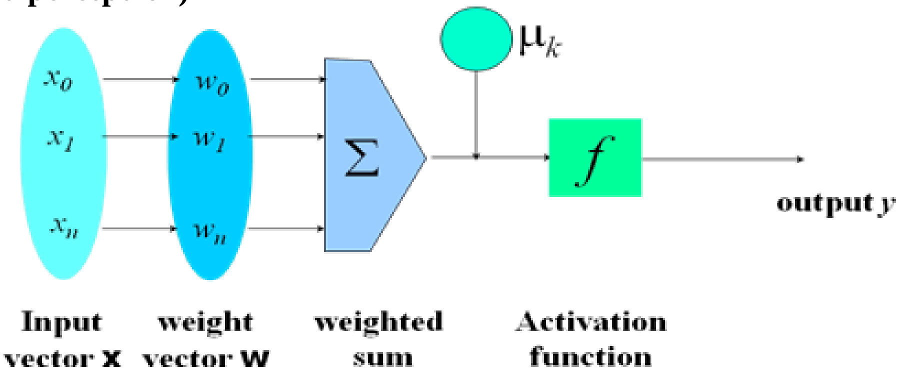


Figure 4.6 Neuron

- The *n*-dimensional input vector x is mapped into variable y by means of the scalar product and a nonlinear function mapping

A multi-layer feed-forward neural network

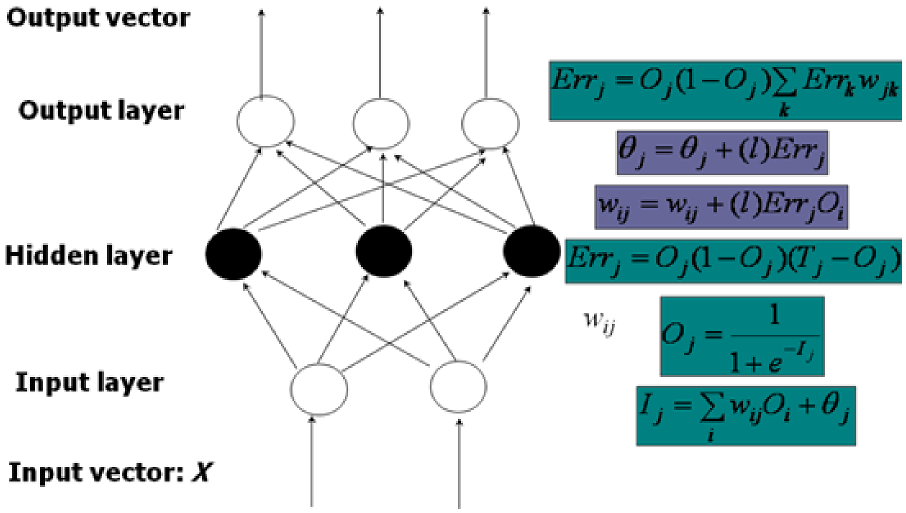

Output vector
Output layer
Hidden layer
Input layer
Input vector: *X*
$Err_j = O_j(1-O_j)\sum_k Err_k w_{jk}$
$\theta_j = \theta_j + (l)Err_j$
$w_{ij} = w_{ij} + (l)Err_j O_i$
$Err_j = O_j(1-O_j)(T_j - O_j)$
w_{ij}
$O_j = \frac{1}{1+e^{-I_j}}$
$I_j = \sum_i w_{ij} O_i + \theta_j$

Figure 4.7 A multi-layer feed-forward neural network

The inputs to the network correspond to the attributes measured for each training tuple

- Inputs are fed simultaneously into the units making up the input layer
- They are then weighted and fed simultaneously to a hidden layer
- The number of hidden layers is arbitrary, although usually only one
- The weighted outputs of the last hidden layer are input to units making up the output layer, which emits the network's prediction
- The network is feed-forward in that none of the weights cycles back to an input unit or to an output unit of a previous layer
- From a statistical point of view, networks perform nonlinear regression: Given enough hidden units and enough training samples, they can closely approximate any function

Back propagation

- Iteratively process a set of training tuples & compare the network's prediction with the actual known target value
- For each training tuple, the weights are modified to minimize the mean squared error between the network's prediction and the actual target value
- Modifications are made in the “backwards” direction: from the output layer, through each hidden layer down to the first hidden layer, hence “backpropagation”
- Steps

    - Initialize weights (to small random #s) and biases in the network
    - Propagate the inputs forward (by applying activation function)
    - Back propagate the error (by updating weights and biases)
    - Terminating condition (when error is very small, etc.)
  - Efficiency of backpropagation: Each epoch (one interaction through the training set) takes O(|D| * *w*), with |D| tuples and *w* weights, but # of epochs can be exponential to n, the number of inputs, in the worst case
  - Rule extraction from networks: network pruning
    - Simplify the network structure by removing weighted links that have the least effect on the trained network
    - Then perform link, unit, or activation value clustering
    - The set of input and activation values are studied to derive rules describing the relationship between the input and hidden unit layers
  - Sensitivity analysis: assess the impact that a given input variable has on a network output. The knowledge gained from this analysis can be represented in rules

SVM—SUPPORT VECTOR MACHINES

- A new classification method for both linear and nonlinear data
- It uses a nonlinear mapping to transform the original training data into a higher dimension
- With the new dimension, it searches for the linear optimal separating hyper plane (i.e., “decision boundary”)
- With an appropriate nonlinear mapping to a sufficiently high dimension, data from two classes can always be separated by a hyper plane
- SVM finds this hyper plane using support vectors (“essential” training tuples) and margins (defined by the support vectors)
- Features: training can be slow but accuracy is high owing to their ability to model complex nonlinear decision boundaries (margin maximization)
- Used both for classification and prediction
  - ***Applications***
    - Handwritten digit recognition,
    - Object recognition

- Speaker identification,
- Benchmarking time-series prediction tests

SVM—General Philosophy

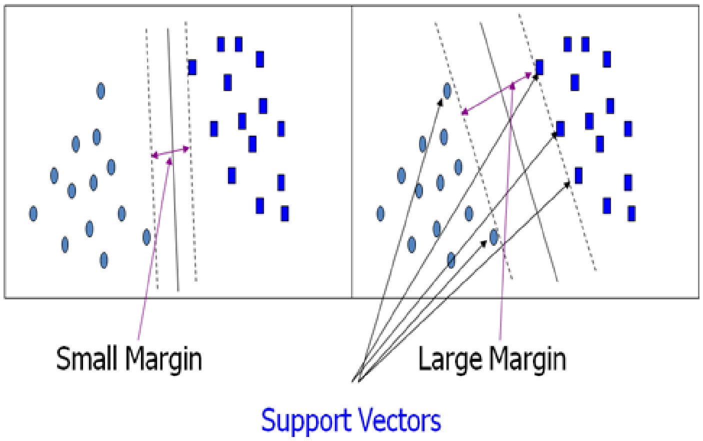


Figure 4.8 SVM—General Philosophy

SVM—Margins and Support Vectors

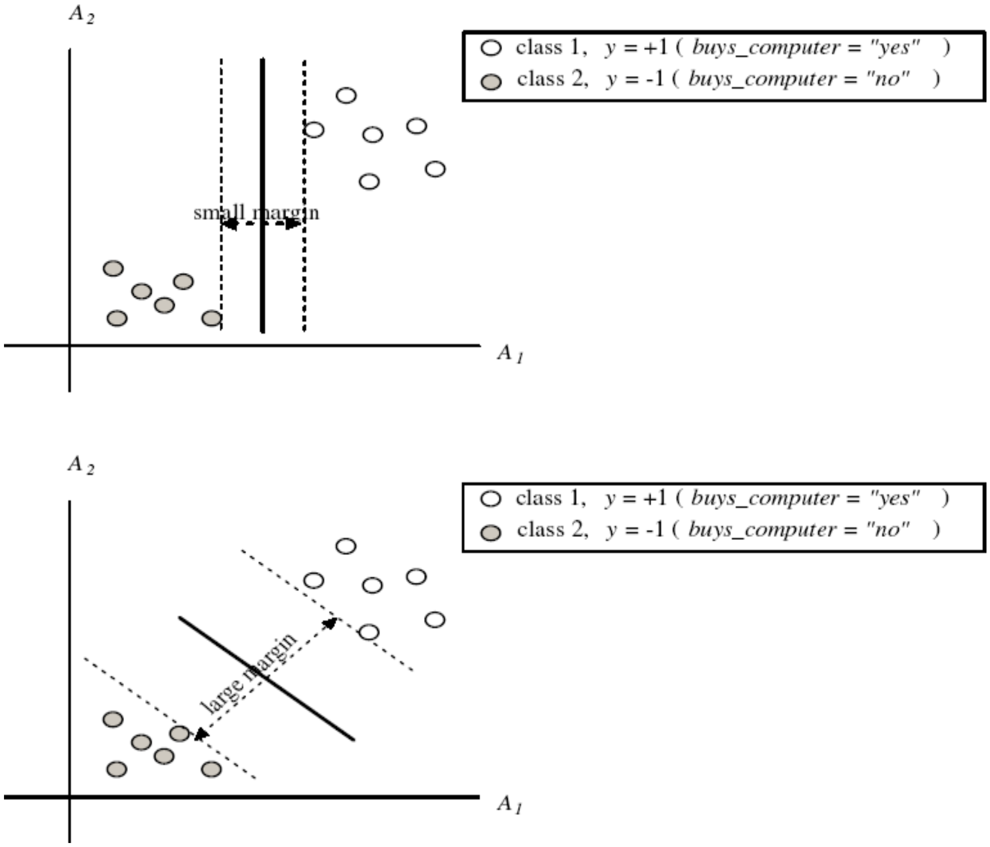
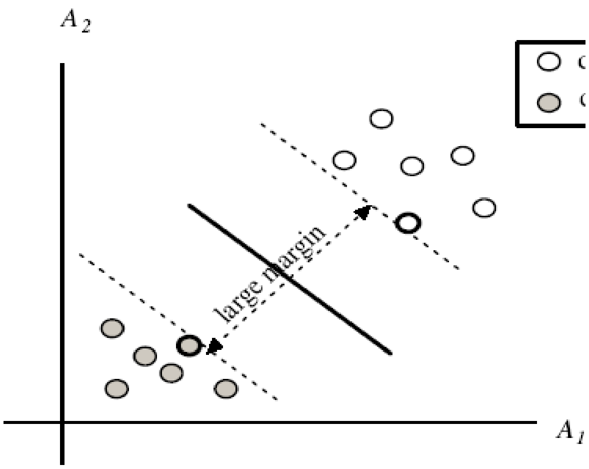

A 2
class 1, y = +1 (buys_computer = "yes")
class 2, y = -1 (buys_computer = "no")
small margin
A 1
A 2
class 1, y = +1 (buys_computer = "yes")
class 2, y = -1 (buys_computer = "no")
large margin
A 1

Figure 4.9 SVM—Margins and Support Vectors

SVM—Linearly Separable



- A separating hyper plane can be written as

$$W \bullet X + b = 0$$

Where W={w_1, w_2, …, w_n} is a weight vector and b a scalar (bias)

- For 2-D it can be written as

$$w_0 + w_1\, x_1 + w_2\, x_2 = 0$$

- The hyper plane defining the sides of the margin:

$$H_1: w_0 + w_1\, x_1 + w_2\, x_2 \geq 1 \quad \text{for } y_i = +1, \text{ and}$$

$$H_2: w_0 + w_1\, x_1 + w_2\, x_2 \leq -1 \text{ for } y_i = -1$$

- Any training tuples that fall on hyper planes H1 or H2 (i.e., the sides defining the margin) are support vectors
- This becomes a constrained (convex) quadratic optimization problem: Quadratic objective function and linear constraints → Quadratic Programming (QP) → Lagrangian multipliers

Why Is SVM Effective on High Dimensional Data?

- The complexity of trained classifier is characterized by the # of support vectors rather than the dimensionality of the data
- The support vectors are the essential or critical training examples —they lie closest to the decision boundary (MMH)
- If all other training examples are removed and the training is repeated, the same separating hyper plane would be found

- The number of support vectors found can be used to compute an (upper) bound on the expected error rate of the SVM classifier, which is independent of the data dimensionality
- Thus, an SVM with a small number of support vectors can have good generalization, even when the dimensionality of the data is high

PREDICTION

- (Numerical) prediction is similar to classification
  - construct a model
  - use model to predict continuous or ordered value for a given input
- Prediction is different from classification
  - Classification refers to predict categorical class label
  - Prediction models continuous-valued functions
- Major method for prediction: regression
  - model the relationship between one or more *independent* or predictorvariables and a *dependent* or response variable
- Regression analysis
  - Linear and multiple regression
  - Non-linear regression
  - Other regression methods: generalized linear model, Poisson regression, log-linear models, regression trees

LINEAR REGRESSION

- ***Linear regression:*** involves a response variable y and a single predictorvariable x

  $$y = w_0 + w_1 x$$

  Where w_0 (y-intercept) and w_1 (slope) are regression coefficients
- ***Method of least squares:*** estimates the best-fitting straight line
  - ***Multiple linear regression:*** involves more than one predictor variable
  - Training data is of the form $(\mathbf{X_1}, y_1), (\mathbf{X_2}, y_2),\ldots, (\mathbf{X_{|D|}}, y_{|D|})$
  - Ex. For 2-D data, we may have: $y = w_0 + w_1 x_1 + w_2 x_2$
  - Solvable by extension of least square method or using SAS, S-Plus
  - Many nonlinear functions can be transformed into the above

Nonlinear Regression

- Some nonlinear models can be modeled by a polynomial function

- A polynomial regression model can be transformed into linear regression model. For example,

$$y = w_0 + w_1\ x + w_2\ x_2 + w_3\ x_3$$

Convertible to linear with new variables: $x_2 = x^2$, $x_3 = x^3$

$$y = w_0 + w_1\ x + w_2\ x_2 + w_3\ x_3$$

- Other functions, such as power function, can also be transformed to linear model
- Some models are intractable nonlinear (e.g., sum of exponential terms)

  Possible to obtain least square estimates through extensive calculation on more complex formulae

PART-A			
Q. No	**Questions**	**Competence**	**BT Level**
1.	Define decision tree induction	Remember	BTL-1
2.	Define Data pruning. State the need for pruning phase in decision tree construction	Remember	BTL-1
3.	Name the features of Decision tree induction.	Understand	BTL-2
4.	Give why pruning is needed in decision tree	Understand	BTL-2
5.	Demonstrate the Bayes classification methods.	Apply	BTL-3
6.	How would you show your understanding about pessimistic pruning?	Apply	BTL-3
7.	What is Naïve Bayesian classification? How is it differing from Bayesian classification?	Analyze	BTL-4
8.	How would you evaluate accuracy of a classifier?	Evaluate	BTL-5
9.	What inference can you formulate with Bayes theorem?	Create	BTL-6
10.	Define Lazy learners and eager learners with an example.	Remember	BTL-1

PART-B			
Q. No	**Questions**	**Competence**	**BT Level**
1.	i) Develop an algorithm for classification using decision trees. Illustrate the algorithm with a relevant example. (7) ii) What approach would you use to apply decision tree induction? (6)	Apply	BTL-3

2.	i)What is Classification? What are the features of Bayesian classification? Explain in detail with an example. (8) ii) Explain how the Bayesian Belief Networks are trained to perform classification. (5)	Evaluate	BTL-5
3.	i) Generalize the Bayes theorem of posterior probability and explain the working of a Bayesian classifier with an example. (9) ii) Formulate rule based classification techniques. (4)	Create	BTL-6
4.	i) Define classification. With an example explain how support vector machines can be used for classification. (7) ii) What are the prediction techniques supported by a data mining systems? (6)	Remember	BTL-1
5.	(i) Explain algorithm for constructing a decision tree from training samples. (9) (ii)Write Bayes theorem. (4)	Analyze	BTL-4
6.	i) Describe in detail about the following Classification methods. (6) (a) Bayesian classification (b) Fuzzy set approach (c) Genetic algorithms. ii) Describe in detail Classification by Back propagation.	Remember	BTL-1
7.	i) Examine in detail about Lazy learners with examples. (4) ii) Describe about the process of multi-layer feed-forward neural network classification using back propagation learning.	Remember	BTL-1

TEXT / REFERENCE BOOKS

1. Jiawei Han and Micheline Kamber, “Data Mining Concepts and Techniques”, 2nd Edition, Elsevier, 2007

2. Alex Berson and Stephen J. Smith, “ Data Warehousing, Data Mining & OLAP”, Tata McGraw Hill, 2007.

3. Pang-Ning Tan, Michael Steinbach and Vipin Kumar, “Introduction To Data Mining”, Person Education, 2007.

4. K.P. Soman, Shyam Diwakar and V. Ajay, “Insight into Data mining Theory and Practice”, Easter Economy Edition, Prentice Hall of India, 2006.

5. G. K. Gupta, “Introduction to Data Mining with Case Studies”, Easter Economy Edition, Prentice Hall of India, 2006.

6. Daniel T.Larose, “Data Mining Methods and Models”, Wile-Interscience, 2006

UNIT – V - CLUSTERING, APPLICATIONS AND TRENDS IN DATA MINING- SCSA3001

CLUSTERING, APPLICATIONS AND TRENDS IN DATA MINING

Cluster analysis - Types of data in Cluster Analysis - Categorization of major clustering methods -Partitioning methods - Hierarchical methods - Density-based methods - Grid-based methods - Model based clustering methods -Constraint Based cluster analysis - Outlier analysis - Social Impacts of Data Mining- Case Studies: Mining WWW- Mining Text Database Mining Spatial Databases.

WHAT IS CLUSTER ANALYSIS?

The process of grouping a set of physical objects into classes of similar objects is called clustering.

Cluster – collection of data objects

– Objects within a cluster are similar and objects in different clusters are dissimilar.

Cluster applications – pattern recognition, image processing and market research.

- helps marketers to discover the characterization of customer groups based on purchasing patterns
- Categorize genes in plant and animal taxonomies
- Identify groups of house in a city according to house type, value and geographical location
- Classify documents on WWW for information discovery

Clustering is a preprocessing step for other data mining steps like classification, characterization.

Clustering – Unsupervised learning – does not rely on predefined classes with class labels.

Typical requirements of clustering in data mining

1. Scalability – Clustering algorithms should work for huge databases

2. Ability to deal with different types of attributes – Clustering algorithms should work not only for numeric data, but also for other data types.

3. Discovery of clusters with arbitrary shape – Clustering algorithms (based on distance measures) should work for clusters of any shape.

4. Minimal requirements for domain knowledge to determine input parameters – Clustering results are sensitive to input parameters to a clustering algorithm (example – number of desired clusters). Determining the value of these parameters is difficult and requires some domain knowledge.

5. Ability to deal with noisy data – Outlier, missing, unknown and erroneous data detected by a clustering algorithm may lead to clusters of poor quality.

6. Insensitivity in the order of input records – Clustering algorithms should produce same results even if the order of input records is changed.

7. High dimensionality – Data in high dimensional space can be sparse and highly skewed, hence it is challenging for a clustering algorithm to cluster data objects in high dimensional space.

8. Constraint-based clustering – In Real world scenario, clusters are performed based on various constraints. It is a challenging task to find groups of data with good clustering behavior and satisfying various constraints.

9. Interpretability and usability – Clustering results should be interpretable, comprehensible and usable. So we should study how an application goal may influence the selection of clustering methods.

TYPES OF DATA IN CLUSTERING ANALYSIS

1. Data Matrix: (object-by-variable structure)

Represents n objects, (such as persons) with p variables (or attributes) (such as age, height, weight, gender, race and so on. The structure is in the form of relational table or n x p matrix as shown below:

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

→ called as "two mode" matrix

2. Dissimilarity Matrix: (object-by-object structure)

This stores a collection of proximities (closeness or distance) that are available for all pairs of n objects. It is represented by an n-by-n table as shown below.

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

→ called as "one mode" matrix

Where d (i, j) is the dissimilarity between the objects i and j; d (i, j) = d (j, i) and d (i, i) = 0

Many clustering algorithms use Dissimilarity Matrix. So data represented using Data matrixes are converted into Dissimilarity Matrix before applying such clustering algorithms.

Clustering of objects done based on their similarities or dissimilarities. Similarity coefficients or dissimilarity coefficients are derived from correlation coefficients

CATEGORIZATION OF MAJOR CLUSTERING METHODS

The choice of many available clustering algorithms depends on type of data available and the application used.

Major Categories are:

1. Partitioning Methods:

- Construct k-partitions of the n data objects, where each partition is a cluster and k <= n.

- Each partition should contain at least one object & each object should belong to exactly one partition.

- Iterative Relocation Technique – attempts to improve partitioning by moving objects from one group to another.

- Good Partitioning – Objects in the same cluster are “close” / related and objects in the different clusters are “far apart” / very different.

Uses the Algorithms

- ✓ ***K-means Algorithm***: - Each cluster is represented by the mean value of the objects in the cluster.
- ✓ ***K-mediods Algorithm***: - Each cluster is represented by one of the objects located near the center of the cluster.
- ✓ These work well in small to medium sized database.

2. Hierarchical Methods:

- Creates hierarchical decomposition of the given set of data objects.

- Two types – Agglomerative and Divisive

- Agglomerative Approach: (Bottom-Up Approach):

- ✓ Each object forms a separate group
- ✓ Successively merges groups close to one another (based on distance between clusters)
- ✓ Done until all the groups are merged to one or until a termination condition holds. (Termination condition can be desired number of clusters)

- Divisive Approach: (Top-Down Approach):

- ✓ Starts with all the objects in the same cluster
- ✓ Successively clusters are split into smaller clusters
- ✓ Done until each object is in one cluster or until a termination condition holds (Termination condition can be desired number of clusters)

- Disadvantage – Once a merge or split is done it cannot be undone.

- Advantage – Less computational cost

- If both these approaches are combined it gives more advantage.

- Clustering algorithms with this integrated approach are BIRCH and CURE.

3. Density Based Methods:

- Above methods produce Spherical shaped clusters.

- To discover clusters of arbitrary shape, clustering done based on the notion of density.

- Used to filter out noise or outliers

Continue growing a cluster so long as the density in the neighborhood exceeds some threshold.

- Density = number of objects or data points

- That is for each data point within a given cluster; the neighborhood of a given radius has to contain at least a minimum number of points.

- Uses the algorithms: DBSCAN and OPTICS

4. Grid-Based Methods:

- Divides the object space into finite number of cells to forma grid structure.

- Performs clustering operations on the grid structure.

- Advantage – Fast processing time – independent on the number of data objects & dependent on the number of cells in the data grid.

- STING – typical grid based method

- CLIQUE and Wave-Cluster – grid based and density based clustering algorithms.

5. Model-Based Methods:

- Hypothesizes a model for each of the clusters and finds a best fit of the data to the model.

- Forms clusters by constructing a density function that reflects the spatial distribution of the data points.

- Robust clustering methods

- Detects noise / outliers.

PARTITIONING METHODS

Database has n objects and k partitions where k<=n; each partition is a cluster.

Partitioning criterion = Similarity function:

Objects within a cluster are similar; objects of different clusters are dissimilar.

Classical Partitioning Methods: k-means and k-mediods:

(A) Centroid-based technique: The k-means method:

- Cluster similarity is measured using mean value of objects in the cluster (or clusters center of gravity)

- Randomly select k objects. Each object is a cluster mean or center.

- Each of the remaining objects is assigned to the most similar cluster – based on the distance between the object and the cluster mean.

- Compute new mean for each cluster.

- This process iterates until all the objects are assigned to a cluster and the partitioning criterion is met.

- This algorithm determines k partitions that minimize the squared error function.

- Square Error Function is defined as:

$$E = \sum_{i=1}^{k} \sum_{x \in C_i} \left| x - m_i \right|^2$$

Where x is the point representing an object, mi is the mean of the cluster Ci.

Algorithm

K-Means Algorithm

1. Given k, the k-means algorithm is implemented in 4 steps:
2. Partition objects into k nonempty subsets
3. Compute seed points as the centroids of the clusters of the current partition. The centroid is the center (mean point) of the cluster.
4. Assign each object to the cluster with the nearest seed point.

Here, E is the sum of the square error for all objects in the data set.x is the point in space representing a given object, and mi is the mean of cluster Ci (both x and mi are multidimensional).

In other words, for each object in each cluster, the distance from the object to its cluster center is squared, and the distances are summed.

This criterion tries to make the resulting k clusters as compact and as separate as possible.

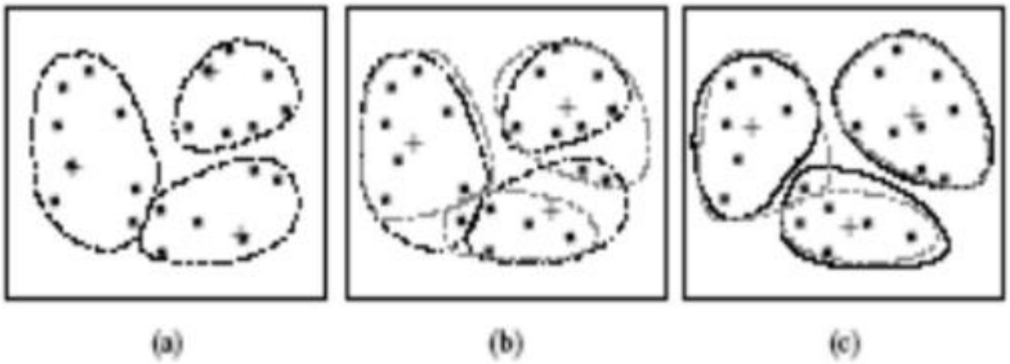


Figure 5.1 K-Means Clustering

Suppose that there is a set of objects located in space as depicted in the rectangle.

Let k =3; i.e. the user would like to cluster the object into three clusters.

According to the algorithm, we arbitrarily choose three objects as the three initial cluster centers, were cluster centers are marked by a "+".

Each object is distributed to a cluster based on the cluster center to which it is the nearest.

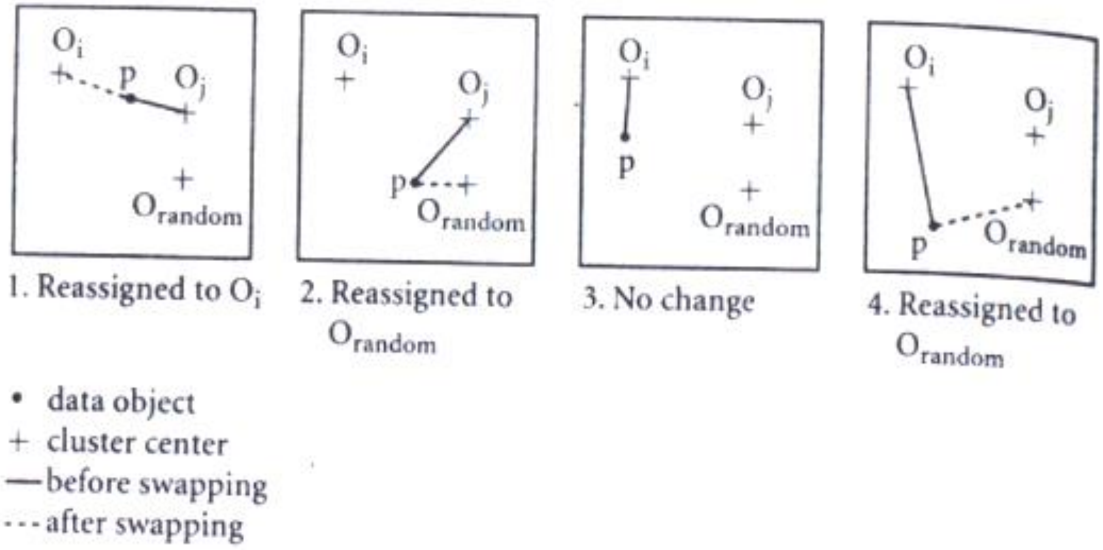
Such distribution forms circled by dotted curves.

Advantages: Scalable; efficient in large databases

K-Medoids Clustering

- A medoid can be defined as that object of a cluster, whose average dissimilarity to all the objects in the cluster is minimal i.e. it is a most centrally located point in the given dataset.
- K-Medoids: Instead of taking the mean value of the object in a cluster as a reference point, medoids can be used, which is the most centrally located object in a cluster.
- The basic strategy of k-medoids clustering algorithms is to find k clusters in n objects by first arbitrarily finding a representative object (the medoid) for each cluster.
- Each remaining object is clustered with the medoid to which it is most similar.
- The strategy then iteratively replaces one of the medoids by one of the non-medoids as long as the quality of the resulting clustering is improved.
- This quality is estimated by using a cost function that measures the average dissimilarity between an object and the medoid of its cluster.

- To determine whether a non-medoid object is "Oi" random is a good replacement for a current medoid "Oj", the following four cases are examined for each of the non-medoid objects "P".



1. Reassigned to O_i 2. Reassigned to O_{random} 3. No change 4. Reassigned to O_{random}

Figure 5.2 K-Mediods Clustering

Case 1: "P" currently belongs to medoid "Oj", If "Oj" is replaced by "Orandom", as a medoid and "P" is closest to one of "Oi", it do not belong "j", then "P" is assigned to "Oi".

Case 2: "P" currently belongs to medoid "Oj". If "Oj" is replaced by "Orandom" as medoid and "P" is closest to "Orandom", then "P" is reassigned to "Orandom".

Case 3: "P" currently belongs to medoid "Oi", it does not belong "j". If "Oj" is replaced by "Orandom" as a medoid and "P" is still closest to "Oi", then the assignment does not change.

Case 4: "P" currently belongs to medoid "Oi", it does not belong to "j". If "Oj" is replaced by "Orandom" as a medoid and "P" is closest to "Orandom", then "P" is reassigned to "Orandom".

Computational Complexity of this algorithm:

✓ O(nkt); n = number of objects, k number of partitions, t = number of iterations

✓ k << n and t << n

HIERARCHICAL METHODS

This method creates the hierarchical decomposition of the given set of data objects.

✓ 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 keeps 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.

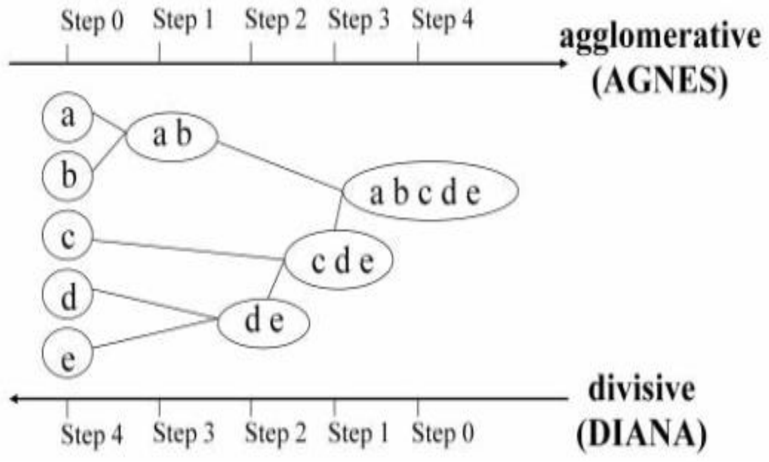


Figure 5.3 Divisive Approach

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 are 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.

Typical methods: Diana, Agnes, BIRCH, ROCK, CAMELEON

BIRCH (1996): uses CF-tree and incrementally adjusts the quality of sub-clusters

Incrementally construct a CF (Clustering Feature) tree, a hierarchical data structure for multiphase clustering

Phase1: scan DB to build an initial in-memory CF tree (a multi-level compression of the data that tries to preserve the inherent clustering structure of the data)

Phase2: use an arbitrary clustering algorithm to cluster the leaf nodes of the CF-tree

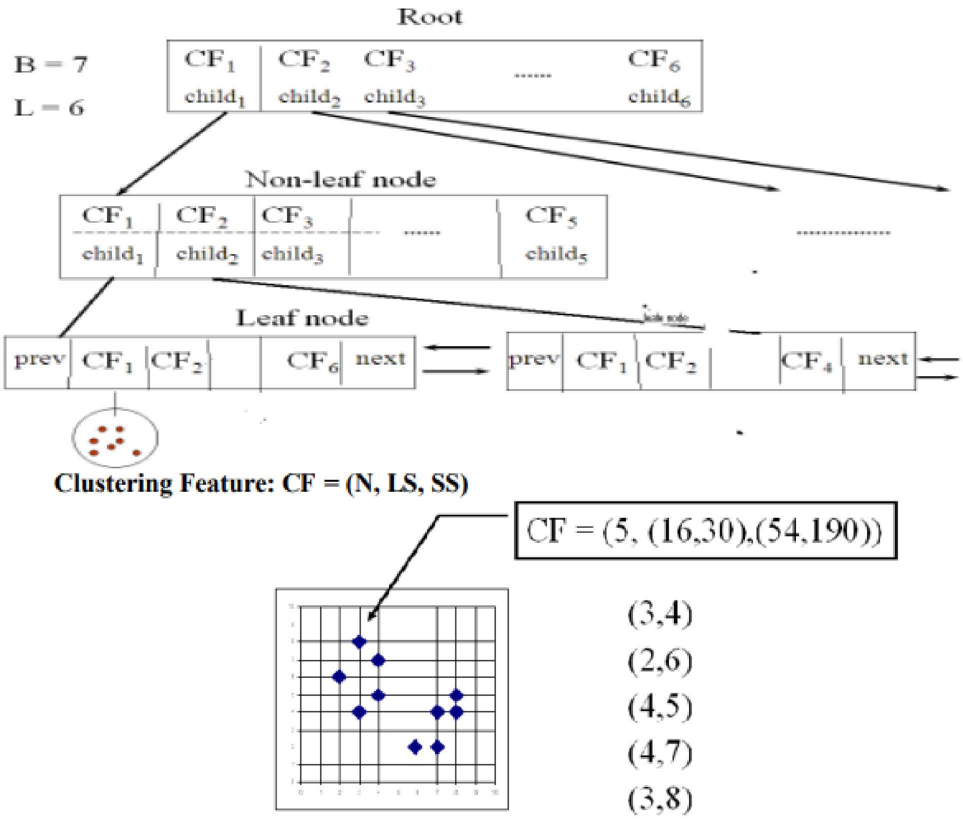


Figure 5.4 BIRCH

Major ideas

- ✓ Use links to measure similarity/proximity
- ✓ Not distance-based
- ✓ Computational complexity:

Algorithm: sampling-based clustering

- ✓ Draw random sample
- ✓ Cluster with links
- ✓ Label data in disk
- ✓ CHAMELEON (1999): hierarchical clustering using dynamic modeling

➢ ***Measures the similarity based on a dynamic model***

- Two clusters are merged only if the interconnectivity and closeness (proximity) between two clusters are high relative to the internal inter connectivity of the clusters and closeness of items within the clusters
- Cure ignores information about inter connectivity of the objects, Rock ignores information about the closeness of two clusters

➢ ***A two-phase algorithm***

- Use a graph partitioning algorithm: cluster objects into a large number of relatively small sub-clusters
- Use an agglomerative hierarchical clustering algorithm: find the genuine clusters by repeatedly combining these sub-clusters

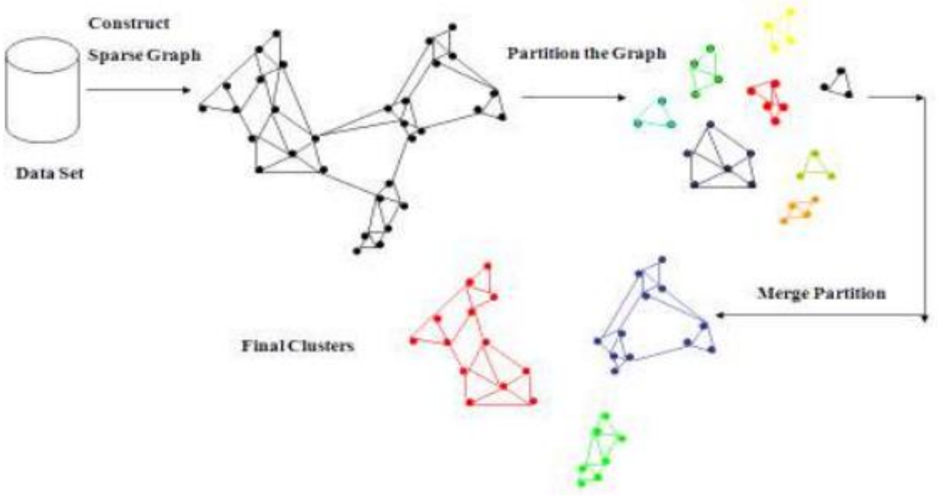


Figure 5.4 Two-phase algorithm

DENSITY-BASED METHOD

Clustering based on density (local cluster criterion), such as density-connected points

Major features:

✓ Discover clusters of arbitrary shape

✓ Handle noise

✓ One scan

✓ Need density parameters as termination condition

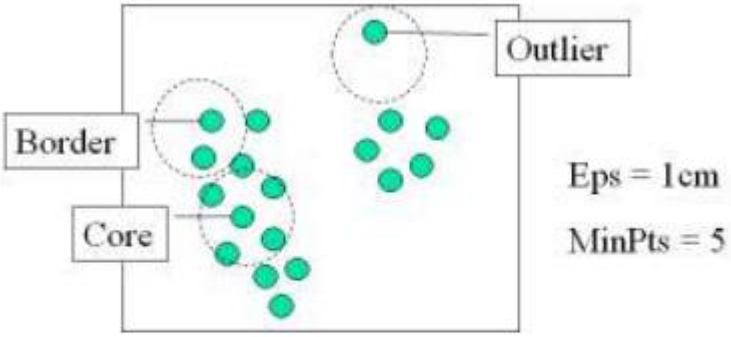
Two parameters:

✓ Eps : Maximum radius of the neighbor hood

✓ Min Pts: Minimum number of points in an Eps- neighborhood of that point

Typical methods: DBSACN, OPTICS, Den Clue

- DBSCAN: Density Based Spatial Clustering of Applications with Noise
- Relies on a density-based notion of cluster: A cluster is defined as a maximal set of density connected points
- Discovers clusters of arbitrary shape in spatial databases with noise
- DBSCAN: The Algorithm
- Arbitrary select a point p
- Retrieve all points density-reachable from p w.r.t .Eps and Min Pts.
- If p is a core point, a cluster is formed.
- If p is a border point, no points are density-reachable from p and DB SCAN visits the next point of the database.
- Continue the process until all of the points have been processed.



OPTICS: Ordering Points To Identify the Clustering Structure

- Produces a special order of the database with its density- based clustering structure
- This cluster-ordering contains info equiv to the density-based clustering’s
- Corresponding to a broad range of parameter settings
- Good for both automatic and interactive cluster analysis, including finding intrinsic
- clustering structure
- Can be represented graphically or using visualization techniques

DENCLUE: DEN sity -based CLU st Ering

- Major features
- Solid mathematical foundation

- Good for data sets with large amounts of noise
- Allows a compact mathematical description of arbitrarily shaped clusters in high Dimensional datasets
- Significant faster than existing algorithm (e.g., DBSCAN)
- But needs a large number of parameters

GRID-BASED METHOD

Using multi-resolution grid data 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.
- Typical methods: STING, Wave Cluster, CLIQUE
- STING: a Statistical INformation Grid approach
- The spatial area is divided into rectangular cells
- There are several levels of cells corresponding to different levels of resolution

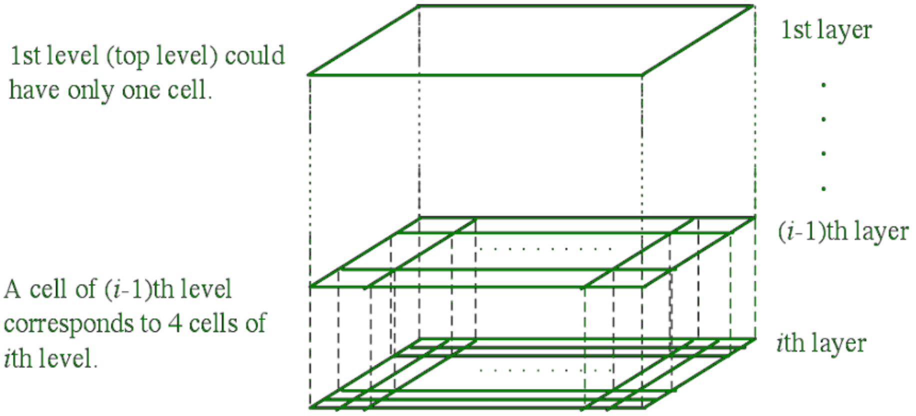

1st level (top level) could have only one cell.
1st layer
(i-1)th layer
A cell of (i-1)th level corresponds to 4 cells of ith level.
ith layer

Figure 5.6 Grid Based

- ➢ Each cell at a high level is partitioned into a number of smaller cells in the next lower level
- ➢ Statistical info of each cell is calculated and stored before hand and is used to answer queries
- ➢ Parameters of higher level cells can be easily calculated from parameters of lower level cell
  - ✓ Count, mean s, min, and max
  - ✓ Type of distribution—normal, uniform, etc.
- ➢ Use a top-down approach to answer spatial data queries
- ➢ Start from a pre-selected layer—typically with a small number of cells
- ➢ For each cell in the current level compute the confidence interval

- Wave Cluster: Clustering by Wavelet Analysis
- A multi-resolution clustering approach which applies wavelet transform to the feature space
- How to apply wavelet transform to find clusters
  - ✓ Summarizes the data by imposing a multidimensional grid structure onto data space
  - ✓ These multidimensional spatial data objects are represented in a n-dimensional feature space
  - ✓ Apply wavelet transform on feature space to find the dense regions in the feature space
  - ✓ Apply wavelet transform multiple times which result in clusters at different scales from fine to coarse
- Wavelet transform: A signal processing technique that decomposes a signal into different frequency sub-band (can be applied to n-dimensional signals)
- Data are transformed to preserve relative distance between objects at different levels of resolution
- Allows natural clusters to become more distinguishable

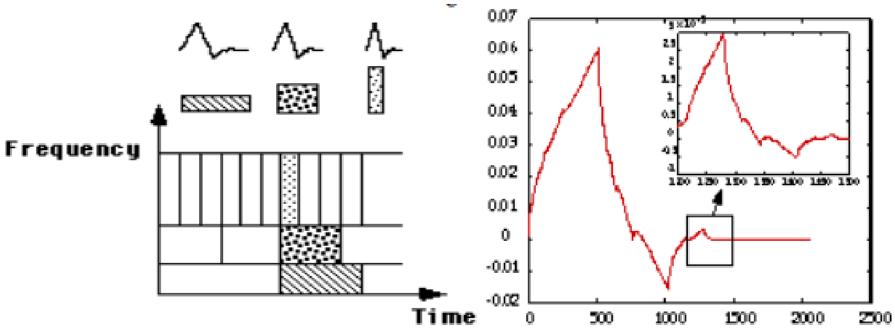


Figure 5.7 Wavelet transform

MODEL-BASED METHODS

- Attempt to optimize the fit between the given data and some mathematical model
- Based on the assumption: Data are generated by a mixture of underlying probability distribution

In this method a model is hypothesize for each cluster and find the best fit of data to the given model

This method also serves away of automatically determining number of clusters based on standard statistics, taking outlier or noise into account. It therefore yields robust clustering methods.

- Typical methods: EM, SOM, COBWEB
- EM — A popular iterative refinement algorithm

An extension to k-means

- ✓ Assign each object to a cluster according to a weight (prob. distribution)
- ✓ New means are computed based on weighted measures

General idea

- ✓ Starts with an initial estimate of the parameter vector
- ✓ Iteratively rescores the patterns against the mixture density produced by the parameter vector
- ✓ The rescored patterns are used to update the parameter updates
- ✓ Patterns belonging to the same cluster, if they are placed by their scores in a particular component

- ➢ Algorithm converges fast but may not be in global optima
- ➢ COBWEB (Fisher'87)
  - ✓ A popular a simple method of incremental conceptual learning
  - ✓ Creates a hierarchical clustering in the form of a classification tree
  - ✓ Each node refers to a concept and contains a probabilistic description of that concept

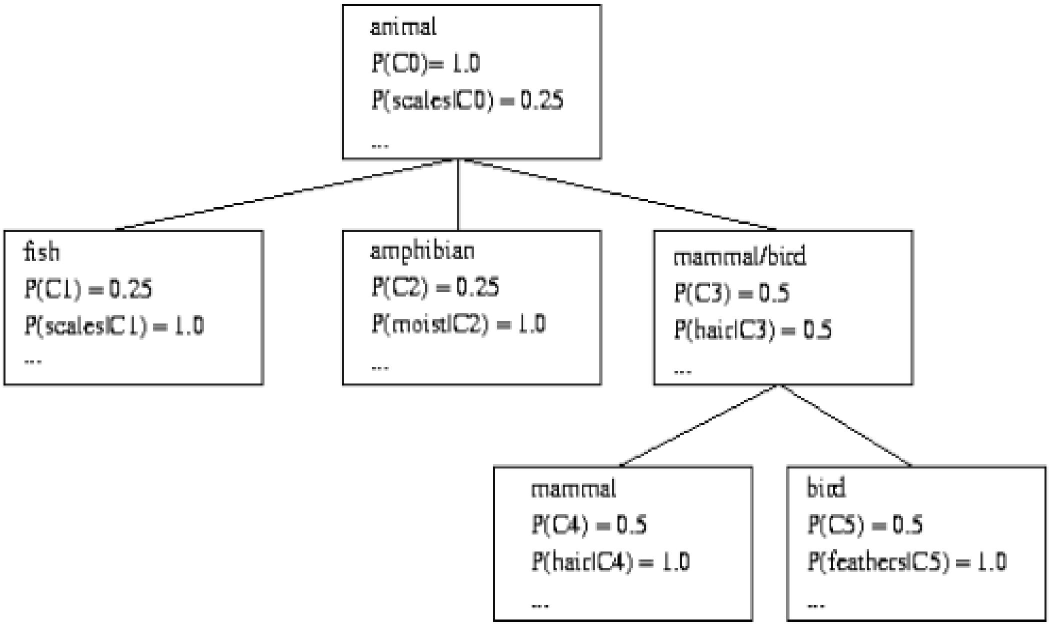


Figure 5.8 COBWEB

SOM (Soft-Organizing feature Map)

Competitive learning

Involves a hierarchical architecture of several units (neurons)

Neurons compete in a— winner- takes- all‖ fashion for the object currently being presented

SOMs, also called topological ordered maps, or Kohonen Self-Organizing Feature Map (KSOMs)

It maps all the points in a high- dimensional source space into a 2 to 3- d target space, s.t the distance and proximity relationship (i.e., topology) are preserved as much as possible

Similarity ok-means: cluster centers tend to lie in a low- dimensional fold in the feature space

Clustering is performed by having several units competing for the current object

- The unit whose weight vector is closest to the current object wins
- The winner and its neighbors learn by having their weights adjusted

SOMs are believed to resemble processing that can occur in the brain

Useful for visualizing high-dimensional data in 2-or3-D space

CONSTRAINT-BASED METHOD

- Clustering by considering user- specified or application-specific constraints
- Typical methods: COD(obstacles), constrained clustering
- Need user feedback: Users know their applications the best
- Less parameters but more user-desired constraints, e.g., an ATM allocation problem: Obstacle & desired clusters
- Clustering in applications: desirable to have user-guided (i.e., constrained) cluster analysis
- Different constraints in cluster analysis:
  - Constraints on individual objects (do selection first)
    - ✓ Cluster on houses worth over$300K
  - Constraints on distance or similarity functions
    - ✓ Weighted functions, obstacles (e.g., rivers, lakes)
  - Constraints on the selection of clustering parameters
    - ✓ # of clusters, Min Pts, etc.
  - User-specified constraints
    - ✓ Contain at least 500 valued customers and 5000 ordinary ones
  - Semi- supervised: giving small training sets as —constraints‖ or hints

Example: Locating k delivery centers, each serving at least m valued customers and n ordinary ones

Proposed approach

- Find an initial—solution ‖by partitioning the data set into k groups and satisfying user-constraints
- Iteratively refine the solution by micro-clustering relocation (e.g., moving δ μ clusters from cluster Ci to Cj) and— deadlock ‖handling (break the micro clusters when necessary)
- Efficiency is improved by micro-clustering
- How to handle more complicated constraints?
- E.g., having approximately same number of valued customers in each cluster?!— Can you solve it?

WHAT IS OUTLIER DISCOVERY

What are outliers?

The set of objects are considerably dissimilar from the remainder of the data

Example: Sports: Michael Jordon, Wayne Gretzky

Problem: Define and find outliers in large data sets

Applications:

- ✓ Credit card fraud detection
- ✓ Telecom fraud detection
- ✓ Customer segmentation
- ✓ Medical analysis

Outlier Discovery: Statistical Approaches

Assume a model underlying distribution that generates data set (e.g. normal distribution)

Use discordancy tests depending on

Data distribution

Distribution parameter (e.g., mean, variance)

Number of expected outliers

Drawbacks

Most tests are for single attribute

In many cases, data distribution may not be known

Outlier Discovery: Distance-Based Approach:

Introduced to counter the main limitations imposed by statistical methods

We need multi-dimensional analysis without knowing data distribution

Distance-based outlier: A DB (p, D)-outlier is an object O in a dataset T such that at least a fraction p of the objects in T lies at a distance greater than D from O

Algorithms for mining distance-based outliers

- Index-based algorithm
- Nested-loop algorithm
- Cell-based algorithm

Density-Based Local Outlier Detection:

- ✓ Distance-based outlier detection is based on global distance distribution
- ✓ It encounters difficulties to identify outliers if data is not uniformly distributed
- ✓ Ex. C1 contains 400 loosely distributed points; C2 has 100 tightly condensed points, 2 outlier points' o1, and o2
- ✓ Distance-based method cannot identify o2 as an outlier
- ✓ Need the concept of local outlier

Outlier Discovery: Deviation-Based Approach:

Identifies outliers by examining the main characteristics of objects in a group

Objects that "deviate" from this description are considered outliers

Sequential exception technique - simulates the way in which humans can distinguish unusual objects from among a series of supposedly like objects

OLAP data cube technique

Uses data cubes to identify regions of anomalies in large multidimensional data

Summary:

- ✓ Cluster analysis groups an object based on their similarity and has wide applications
- ✓ Measure of similarity can be computed for various types of data
- ✓ Clustering algorithms can be categorized into partitioning methods, hierarchical methods, density-based methods, grid-based methods, and model-based methods
- ✓ Outlier detection and analysis are very useful for fraud detection, etc. and can be performed by statistical, distance-based or deviation-based approaches
- ✓ There are still lots of research issues on cluster analysis

Problems and Challenges:

- ✓ Considerable progress has been made in scalable clustering methods
- ✓ Partitioning: k-means, k-medoids, CLARANS
- ✓ Hierarchical: BIRCH, ROCK, CHAMELEON
- ✓ Density-based: DBSCAN, OPTICS, DenClue
- ✓ Grid-based: STING, Wave Cluster, CLIQUE
- ✓ Model-based: EM, Cobweb, SOM
- ✓ Frequent pattern-based: pCluster
- ✓ Constraint-based: COD, constrained-clustering
- ✓ Current clustering techniques do not address all the requirements adequately, still an active area of research

SOCIAL IMPACTS OF DATA MINING

1. Is Data Mining Hype or Will It Being Persistent?

- Data mining is a technology
- Technological life cycle
- Innovators
- Early Adopters
- Early Adopters
- Chasm
- Early Majority
- Late Majority

2. Data Mining: Managers' Business or Everyone's?

- Data mining will surely be an important tool for managers' decision making
- Bill Gates: "Business @ the speed of thought"
- The amount of the available data is increasing, and data mining systems will be more affordable
- Multiple personal uses
- Mine your family's medical history to identify genetically-related medical conditions
- Mine the records of the companies you deal with
- Mine data on stocks and company performance, etc.
- Invisible data mining
- Build data mining functions into many intelligent tools

3. Social Impacts: Threat to Privacy and Data Security?

- Is data mining a threat to privacy and data security?
- "Big Brother", "Big Banker", and "Big Business" are carefully watching you
- Profiling information is collected every time
- Credit card, debit card, supermarket loyalty card, or frequent flyer card, or apply for any of the above
- You surf the Web, rent a video, and fill out a contest entry form,
- You pay for prescription drugs, or present you medical care number when visiting the doctor
- Collection of personal data may be beneficial for companies and consumers, there is also potential for misuse
- Medical Records, Employee Evaluations, etc.

4. Protect Privacy and Data Security

1. Fair information practices
    - International guidelines for data privacy protection
    - Cover aspects relating to data collection, purpose, use, quality, openness, individual participation, and accountability
    - Purpose specification and use limitation
    - Openness: Individuals have the right to know what information is collected about them, who has access to the data, and how the data are being used
2. Develop and use data security-enhancing techniques
    - Blind signatures
    - Biometric encryption
    - Anonymous databases

MINING WWW (WORLD WIDE WEB)

The World Wide Web contains huge amounts of information that provides a 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 to 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 the Document Object Model (DOM). The DOM structure refers to a tree like structure where 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 does 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 and not for description of semantic structure of the web page. The DOM structure cannot correctly identify the semantic relationship between the 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 a tree structure. In this tree each node corresponds to a block.
- A value is assigned to each node. This value is called the Degree of Coherence. This value is assigned to indicate the coherent 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 semantics of the web page is constructed on the basis of these blocks.

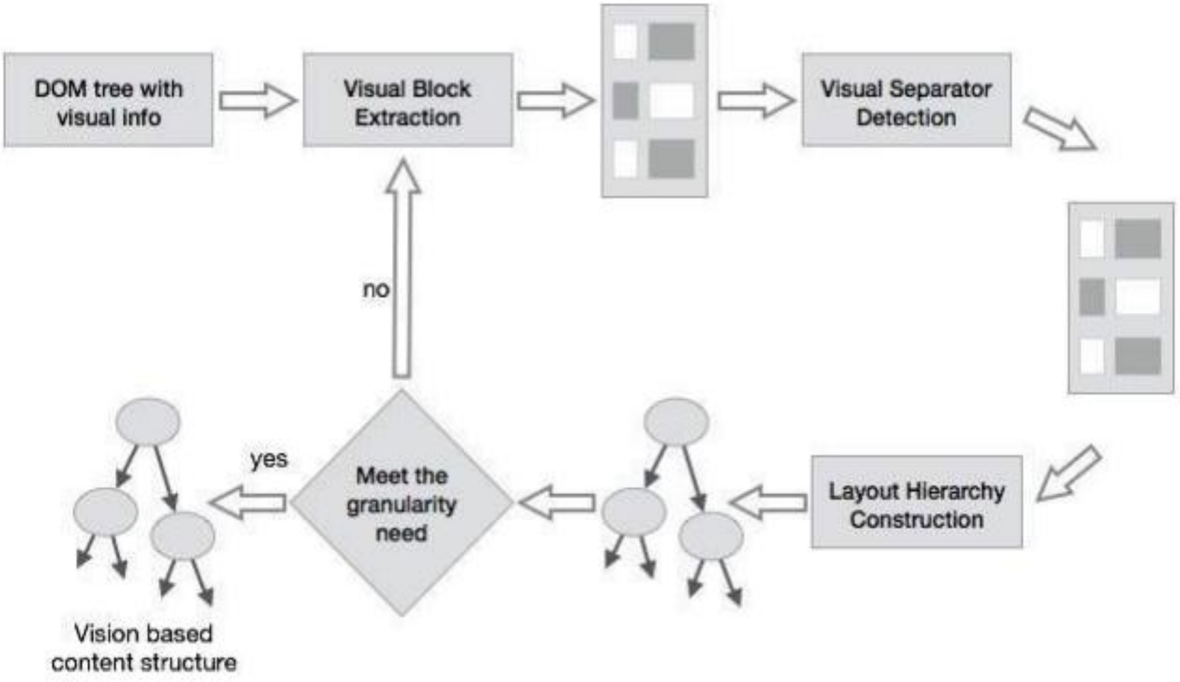

DOM tree with visual info
Visual Block Extraction
Visual Separator Detection
no
Layout Hierarchy Construction
yes
Meet the granularity need
Vision based content structure

Figure 5.9 Vision-based page segmentation

MINING TEXT DATABASE

Text databases consist of huge collection of documents. They collect this information from several sources such as news articles, books, digital libraries, e-mail messages, web pages, etc.

Due to increase in the amount of information, the text databases are growing rapidly. In many of the text databases, the data is semi-structured.

For example, a document may contain a few structured fields, such as title, author, publishing date, etc. But along with the structure data, the document also contains unstructured text components, such as abstract and contents. Without knowing what could be in the documents, it is difficult to formulate effective queries for analyzing and extracting useful information from the data. Users require tools to compare the documents and rank their importance and relevance. Therefore, text mining has become popular and an essential theme in data mining.

Information Retrieval

Information retrieval deals with the retrieval of information from a large number of text-based documents. Some of the database systems are not usually present in information retrieval systems because both handle different kinds of data. Examples of information retrieval system include −

- Online Library catalogue system
- Online Document Management Systems
- Web Search Systems etc.

Basic Measures for Text Retrieval

We need to check the accuracy of a system when it retrieves a number of documents on the basis of user's input. Let the set of documents relevant to a query be denoted as {Relevant} and the set of retrieved document as {Retrieved}. The set of documents that are relevant and retrieved can be denoted as {Relevant} ∩ {Retrieved}. This can be shown in the form of a Venn diagram as follows –

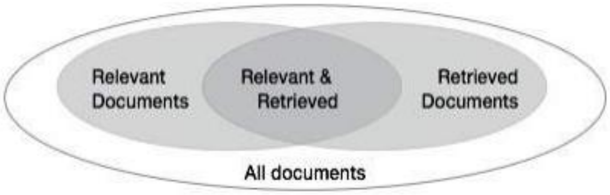


Figure 5.10 Information Retrieval

There are three fundamental measures for assessing the quality of text retrieval −

- Precision
- Recall
- F-score

Precision

Precision is the percentage of retrieved documents that are in fact relevant to the query.

Precision can be defined as −

Precision= |{Relevant} ∩ {Retrieved}| / |{Retrieved}|

Recall

Recall is the percentage of documents that are relevant to the query and were in fact retrieved.

Recall is defined as −

Recall = |{Relevant} ∩ {Retrieved}| / |{Relevant}|

F-score

F-score is the commonly used trade-off. The information retrieval system often needs to trade-off for precision or vice versa.

F-score is defined as harmonic mean of recall or precision as follows

F-score = recall x precision / (recall + precision) / 2

MINING SPATIAL DATABASES

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, geomarketing, 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.

“What about using statistical techniques for spatial data mining?” Statistical spatial data analysis has been a popular approach to analyzing spatial data and exploring geographic information. The term geostatistics is often associated with continuous geographic space. Whereas the term spatial statistics is often associated with discrete space. In a statistical model that handles non-spatial data, one usually assumes statistical independence among different portions of data. However, different from traditional data sets, there is no such independence among spatially distributed data because in reality, spatial objects are often interrelated, or more exactly spatially co-located, in the sense that the closer the two objects are located, the more likely they share similar properties. For example, nature resource, climate, temperature, and economic situations are likely to be similar in geographically closely located regions. People even consider this as the first law of geography: “Everything is related to everything else, but nearby things are more related than distant things.” Such a property of close interdependency across nearby space leads to the notion of spatial autocorrelation. Based on this notion, spatial statistical modeling methods have been developed with good success. Spatial data mining will further develop spatial statistical analysis methods and extend them for huge amounts of spatial data, with more emphasis on efficiency, scalability, cooperation with database and data warehouse systems, improved user interaction, and the discovery of new types of knowledge.

There are three types of dimensions in a spatial data cube:

A non-spatial dimension contains only nonspatial data. Non-spatial dimensions temperature and precipitation

A spatial-to-nonspatial dimension is a dimension whose primitive-level data are spatial but whose generalization, starting at a certain high level, becomes nonspatial

A spatial-to-spatial dimension is a dimension whose primitive level and all of its high level generalized data are spatial.

We distinguish two types of measures in a spatial data cube:

A numerical measure contains only numerical data. For example, one measure in a spatial data warehouse could be the monthly revenue of a region, so that a roll-up may compute the total revenue by year, by county, and so on. Numerical measures can be further classified into distributive, algebraic, and holistic, as discussed in

A spatial measure contains a collection of pointers to spatial objects. For example, in a generalization (or roll-up) in the spatial data cube of Example 10.5, the regions with the same range of temperature and precipitation will be grouped into the same cell, and the measure so formed contains a collection of pointers to those regions.

PART-A			
Q. No	**Questions**	**Competence**	**BT Level**
1.	Identify what changes you make to solve the problem in cluster analysis.	Remember	BTL-1
2.	Formulate the role of application and challenges in clustering	Create	BTL-6
3.	List the challenges of outlier detection	Remember	BTL-1
4.	Classify the hierarchical clustering methods.	Understand	BTL-2
5.	Distinguish between Classification and clustering	Understand	BTL-2

6.	Show the intrinsic methods in cluster analysis	Apply	BTL-3
7.	Evaluate the different types of data used for cluster analysis?	Create	BTL-6
8.	State hierarchical method?	Analyze	BTL-4
9.	Evaluate agglomerative and divisive hierarchical clustering?	Create	BTL-6
10.	Define Outlier Detection?	Remember	BTL-1
PART-B			
Q. No	**Questions**	**Competence**	**BT Level**
1.	Discuss the various types of data in cluster analysis?	Understand	BTL-2
2.	Explain the categories of major clustering methods?	Understand	BTL-2
3.	State algorithms for k-means and k-medoids? Explain?	Analyze	BTL-4
4.	Define the distance-based outlier? Illustrate the efficient algorithms forming distance-based algorithm?	Apply	BTL-3
5.	i) Explain the hierarchical based method for cluster analysis. (7) ii) Explain in detail about density based methods.	Analyze	BTL-4
6.	(i) With an example explain how support vector machine scan be used for classification. (7) (ii) What are the prediction techniques supported by a data mining systems? (6)	Remember	BTL-1
7.	i) Develop a clustering high dimensional data. (8) ii) Consider five points { X1, X2,X3, X4, X5} with the following coordinates as a two dimensional sample for clustering: X1 = (0,2.5); X2 = (0,0); X3= (1.5,0); X4 = (5,0); X5 = (5,2)Compose the K-means partitioning algorithm using the above data set. (7)	Create	BTL-6
8.	Explain the process of mining the World Wide Web	Understand	BTL-2
9.	Describe in detail about spatial mining and time series mining	Remember	BTL-1

TEXT / REFERENCE BOOKS

1. Jiawei Han and Micheline Kamber, “Data Mining Concepts and Techniques”, 2nd Edition, Elsevier, 2007

2. Alex Berson and Stephen J. Smith, “ Data Warehousing, Data Mining & OLAP”, Tata McGraw Hill, 2007.

3. Pang-Ning Tan, Michael Steinbach and Vipin Kumar, “Introduction To Data Mining”, Person Education, 2007.

4. K.P. Soman, Shyam Diwakar and V. Ajay, “Insight into Data mining Theory and Practice”, Easter Economy Edition, Prentice Hall of India, 2006.

5. G. K. Gupta, “Introduction to Data Mining with Case Studies”, Easter Economy Edition, Prentice Hall of India, 2006.

6. Daniel T.Larose, “Data Mining Methods and Models”, Wile-Interscience, 2006

*************************ALL THE BEST*****************************