SRI INDU COLLEGE OF ENGINEERING & TECHNOLOGY

(An Autonomous Institution under UGC, New Delhi)

B.Tech. - IV Year – I Semester	L	T	P	C
	2	0	0	2

(R18CSE4102) Data Mining

UNIT I

Data Warehousing, Business Analysis and On-Line Analytical Processing (OLAP) : Basic Concepts – Data Warehousing Components – Building a Data Warehouse – Database Architectures for Parallel Processing – Parallel DBMS Vendors – Multidimensional Data Model – Data Warehouse Schemas for Decision Support, Concept Hierarchies -Characteristics of OLAP Systems – Typical OLAP Operations, OLAP and OLTP.

UNIT II

Data Mining – Introduction : Introduction to Data Mining Systems – Knowledge Discovery Process – Data Mining Techniques – Issues – applications- Data Objects and attribute types, Statistical description of data, Data Preprocessing – Cleaning, Integration, Reduction, Transformation and discretization, Data Visualization, Data similarity and dissimilarity measures.

UNIT III

Data Mining – Frequent Pattern Analysis : Mining Frequent Patterns, Associations and Correlations – Mining Methods- Pattern Evaluation Method – Pattern Mining in Multilevel, Multi Dimensional Space – Constraint Based Frequent Pattern Mining, Classification using Frequent Patterns

UNIT IV

Classification and Clustering : Decision Tree Induction – Bayesian Classification – Rule Based Classification – Classification by Back Propagation – Support Vector Machines — Lazy Learners – Model Evaluation and Selection-Techniques to improve Classification Accuracy. Clustering Techniques – Cluster analysis-Partitioning Methods – Hierarchical Methods – Density Based Methods – Grid Based Methods – Evaluation of clustering – Clustering high dimensional data- Clustering with constraints, Outlier analysis-outlier detection methods.

UNITV:

Applications and Trends In Data Mining : Data mining applications, Data Mining Products and Research Prototypes, Additional Themes on Data Mining and Social Impacts Of Data Mining.

TEXT BOOK:

1. Jiawei Han and Micheline Kamber, ―Data Mining Concepts and Techniques, Third Edition, Elsevier, 2012.
2. Alex Berson and Stephen J.Smith, ―Data Warehousing, Data Mining & OLAP‖, Tata McGraw – Hill Edition, 35th Reprint 2016.

REFERENCES:

1. K.P. Soman, Shyam Diwakar and V. Ajay, ―Insight into Data Mining Theory and Practice, Eastern Economy Edition, Prentice Hall of India, 2006.

2. Ian H.Witten and Eibe Frank, ―Data Mining: Practical Machine Learning Tools and Techniques, Elsevier, Second Edition.

OUTCOMES

After successful completion of the course, the learners would be able to

- Understand the fundamental concepts, benefits and problem areas associated with datawarehousing
- Evaluate the different models of OLAP and data preprocessing
- Remember the concept, structure and major issues of data mining
- Analyze and compare various data mining techniques based on different parameters.
- Applying Association and classification knowledge to different data sets
- Create the clusters for different data set

UNIT 1

Data Warehousing, Business Analysis and On-Line Analytical Processing (OLAP) : Basic Concepts – Data Warehousing Components – Building a Data Warehouse – Database Architectures for Parallel Processing – Parallel DBMS Vendors – Multidimensional Data Model – Data Warehouse Schemas for Decision Support, Concept Hierarchies -Characteristics of OLAP Systems – Typical OLAP Operations, OLAP and OLTP

Data Warehouse

Data Warehouse is a relational database management system (RDBMS) construct to meet the requirement of transaction processing systems. It can be loosely described as any centralized data repository which can be queried for business benefits. It is a database that stores information oriented to satisfy decision-making requests. It is a group of decision support technologies, targets to enabling the knowledge worker (executive, manager, and analyst) to make superior and higher decisions. So, Data Warehousing support architectures and tool for business executives to systematically organize, understand and use their information to make strategic decisions.

Data Warehouse environment contains an extraction, transportation, and loading (ETL) solution, an online analytical processing (OLAP) engine, customer analysis tools, and other applications that handle the process of gathering information and delivering it to business users.

What is a Data Warehouse?

A Data Warehouse (DW) is a relational database that is designed for query and analysis rather than transaction processing. It includes historical data derived from transaction data from single and multiple sources.

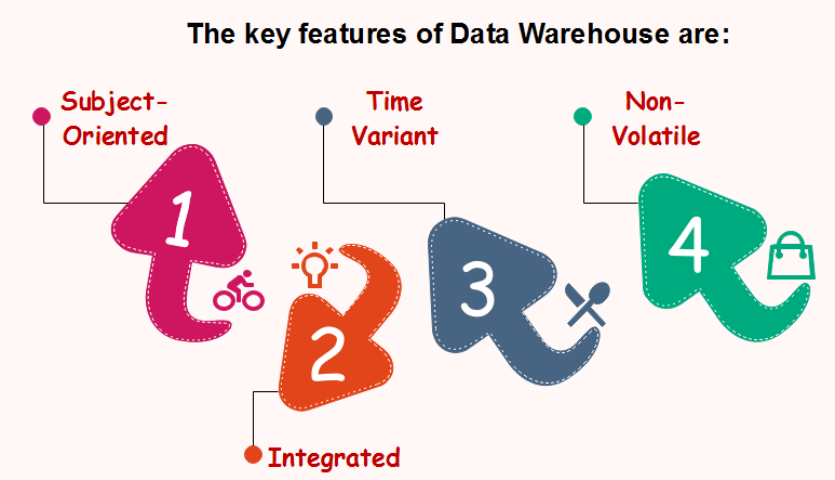
A Data Warehouse provides integrated, enterprise-wide, historical data and focuses on providing support for decision-makers for data modeling and analysis.

A Data Warehouse is a group of data specific to the entire organization, not only to a particular group of users.

It is not used for daily operations and transaction processing but used for making decisions.

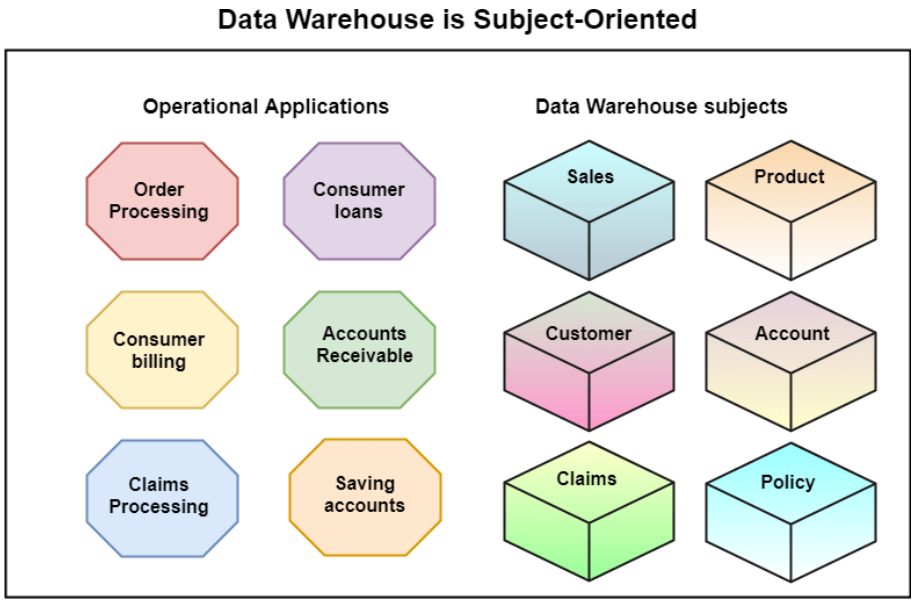
A Data Warehouse can be viewed as a data system with the following attributes:

It is a database designed for investigative tasks, using data from various applications.
It supports a relatively small number of clients with relatively long interactions.
It includes current and historical data to provide a historical perspective of information.
Its usage is read-intensive.
It contains a few large tables.
"Data Warehouse is a subject-oriented, integrated, and time-variant store of information in support of management's decisions."
Characteristics of Data Warehouse



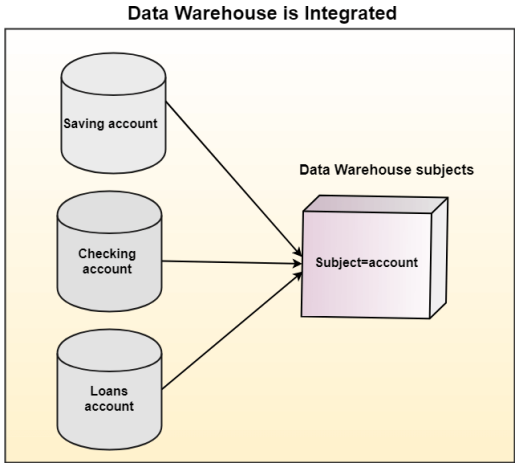
Subject-Oriented

A data warehouse target on the modeling and analysis of data for decision-makers. Therefore, data warehouses typically provide a concise and straightforward view around a particular subject, such as customer, product, or sales, instead of the global organization's ongoing operations. This is done by excluding data that are not useful concerning the subject and including all data needed by the users to understand the subject.



Integrated

A data warehouse integrates various heterogeneous data sources like RDBMS, flat files, and online transaction records. It requires performing data cleaning and integration during data warehousing to ensure consistency in naming conventions, attributes types, etc., among different data sources.

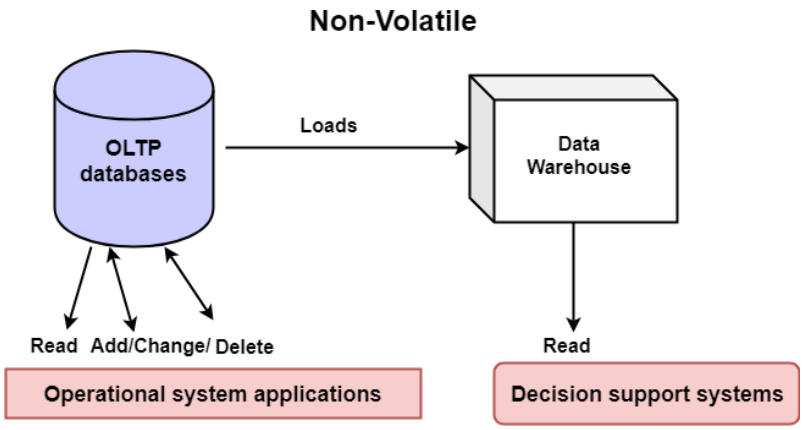


Time-Variant

Historical information is kept in a data warehouse. For example, one can retrieve files from 3 months, 6 months, 12 months, or even previous data from a data warehouse. These variations with a transactions system, where often only the most current file is kept.

Non-Volatile

The data warehouse is a physically separate data storage, which is transformed from the source operational RDBMS. The operational updates of data do not occur in the data warehouse, i.e., update, insert, and delete operations are not performed. It usually requires only two procedures in data accessing: Initial loading of data and access to data. Therefore, the DW does not require transaction processing, recovery, and concurrency capabilities, which allows for substantial speedup of data retrieval. Non-Volatile defines that once entered into the warehouse, and data should not change.



Goals of Data Warehousing

To help reporting as well as analysis
Maintain the organization's historical information
Be the foundation for decision making.
Benefits of Data Warehouse

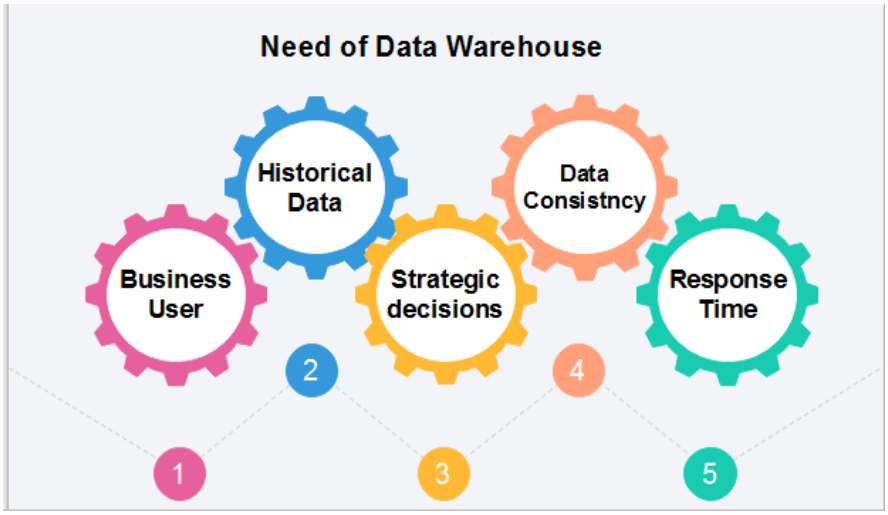
Understand business trends and make better forecasting decisions.
Data Warehouses are designed to perform well enormous amounts of data.
The structure of data warehouses is more accessible for end-users to navigate, understand, and query.
Queries that would be complex in many normalized databases could be easier to build and maintain in data warehouses.

Data warehousing is an efficient method to manage demand for lots of information from lots of users.

Data warehousing provide the capabilities to analyze a large amount of historical data.

Need for Data Warehouse

Data Warehouse is needed for the following reasons:



1) Business User: Business users require a data warehouse to view summarized data from the past. Since these people are non-technical, the data may be presented to them in an elementary form.

2) Store historical data: Data Warehouse is required to store the time variable data from the past. This input is made to be used for various purposes.

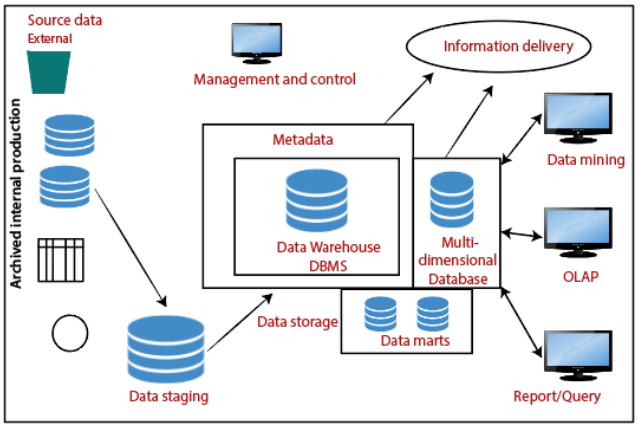
3) Make strategic decisions: Some strategies may be depending upon the data in the data warehouse. So, data warehouse contributes to making strategic decisions.

4) For data consistency and quality: Bringing the data from different sources at a commonplace, the user can effectively undertake to bring the uniformity and consistency in data.

5) High response time: Data warehouse has to be ready for somewhat unexpected loads and types of queries, which demands a significant degree of flexibility and quick response time.

Components or Building Blocks of Data Warehouse

Architecture is the proper arrangement of the elements. We build a data warehouse with software and hardware components. To suit the requirements of our organizations, we arrange these building we may want to boost up another part with extra tools and services. All of these depends on our circumstances.



Components or Building Blocks of Data Warehouse

The figure shows the essential elements of a typical warehouse. We see the Source Data component shows on the left. The Data staging element serves as the next building block. In the middle, we see the Data Storage component that handles the data warehouses data. This element not only stores and manages the data; it also keeps track of data using the metadata repository. The Information Delivery component shows on the right consists of all the different ways of making the information from the data warehouses available to the users.

Source Data Component

Source data coming into the data warehouses may be grouped into four broad categories: Production Data: This type of data comes from the different operating systems of the enterprise. Based on the data requirements in the data warehouse, we choose segments of the data from the various operational modes.

Internal Data: In each organization, the client keeps their "private" spreadsheets, reports, customer profiles, and sometimes even department databases. This is the internal data, part of which could be useful in a data warehouse.

Archived Data: Operational systems are mainly intended to run the current business. In every operational system, we periodically take the old data and store it in achieved files. External Data: Most executives depend on information from external sources for a large percentage of the information they use. They use statistics associating to their industry produced by the external department.

Data Staging Component

After we have been extracted data from various operational systems and external sources, we have to prepare the files for storing in the data warehouse. The extracted data coming from several different sources need to be changed, converted, and made ready in a format that is relevant to be saved for querying and analysis.

We will now discuss the three primary functions that take place in the staging area.

Data Extraction: This method has to deal with numerous data sources. We have to employ the appropriate techniques for each data source.

Data Transformation: As we know, data for a data warehouse comes from many different sources. If data extraction for a data warehouse posture big challenges, data transformation present even significant challenges. We perform several individual tasks as part of data transformation.

First, we clean the data extracted from each source. Cleaning may be the correction of misspellings or may deal with providing default values for missing data elements, or elimination of duplicates when we bring in the same data from various source systems.

Standardization of data components forms a large part of data transformation. Data transformation contains many forms of combining pieces of data from different sources. We combine data from single source record or related data parts from many source records.

On the other hand, data transformation also contains purging source data that is not useful and separating outsource records into new combinations. Sorting and merging of

data take place on a large scale in the data staging area. When the data transformation function ends, we have a collection of integrated data that is cleaned, standardized, and summarized.

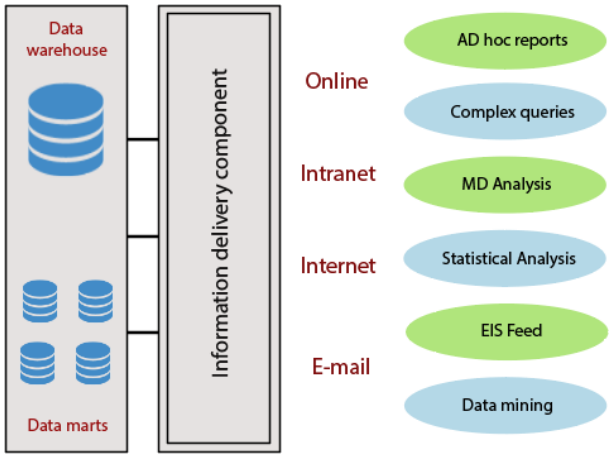
Data Loading: Two distinct categories of tasks form data loading functions. When we complete the structure and construction of the data warehouse and go live for the first time, we do the initial loading of the information into the data warehouse storage. The initial load moves high volumes of data using up a substantial amount of time.

Data Storage Components

Data storage for the data warehousing is a split repository. The data repositories for the operational systems generally include only the current data. Also, these data repositories include the data structured in highly normalized for fast and efficient processing.

Information Delivery Component

The information delivery element is used to enable the process of subscribing for data warehouse files and having it transferred to one or more destinations according to some customer-specified scheduling algorithm.



Information delivery component

Metadata Component

Metadata in a data warehouse is equal to the data dictionary or the data catalog in a database management system. In the data dictionary, we keep the data about the logical data structures, the data about the records and addresses, the information about the indexes, and so on.

Data Marts

It includes a subset of corporate-wide data that is of value to a specific group of users. The scope is confined to particular selected subjects. Data in a data warehouse should be a fairly current, but not mainly up to the minute, although development in the data warehouse industry has made standard and incremental data dumps more achievable. Data marts are lower than data warehouses and usually contain organization. The current trends in data warehousing are to developed a data warehouse with several smaller related data marts for particular kinds of queries and reports.

Management and Control Component

The management and control elements coordinate the services and functions within the data warehouse. These components control the data transformation and the data transfer into the data warehouse storage. On the other hand, it moderates the data delivery to the clients. Its work with the database management systems and authorizes data to be correctly saved in the repositories. It monitors the movement of information into the staging method and from there into the data warehouses storage itself.

Building a Data Warehouse

A Data warehouse is a heterogeneous collection of different data sources organized under unified schema. Builders should take a broad view of the anticipated use of the warehouse while constructing a data warehouse. During the design phase, there is no way to anticipate all possible queries or analyses. Some characteristic of Data warehouse are:

Subject oriented

Integrated

Time Variant

Non-volatile

Building a Data Warehouse –

Some steps that are needed for building any data warehouse are as following below:

To extract the data (transnational) from different data sources: For building a data warehouse, a data is extracted from various data sources and that data is stored in central storage area. For extraction of the data Microsoft has come up with an

excellent tool. When you purchase Microsoft SQL Server, then this tool will be available at free of cost.

To transform the transnational data:

There are various DBMS where many of the companies stores their data. Some of them are: MS Access, MS SQL Server, Oracle, Sybase etc. Also these companies saves the data in spreadsheets, flat files, mail systems etc. Relating a data from all these sources is done while building a data warehouse.

To load the data (transformed) into the dimensional database:

After building a dimensional model, the data is loaded in the dimensional database. This process combines the several columns together or it may split one field into the several columns. There are two stages at which transformation of the data can be performed and they are: while loading the data into the dimensional model or while data extraction from their origins.

To purchase a front-end reporting tool:

There are top notch analytical tools are available in the market. These tools are provided by the several major vendors. A cost effective tool and Data Analyzer is released by the Microsoft on its own.

For the warehouse there is an acquisition of the data. There must be a use of multiple and heterogeneous sources for the data extraction, example databases. There is a need for the consistency for which formation of data must be done within the warehouse. Reconciliation of names, meanings and domains of data must be done from unrelated sources. There is also a need for the installation of the data from various sources in the data model of the warehouse.

Conversion of the data might be done from object oriented, relational or legacy databases to a multidimensional model. One of the largest labor demanding component of data warehouse construction is data cleaning, which is one of the complex process. Before loading of the data in the warehouse, there should be cleaning of the data. All the work of loading must be done in warehouse for better performance. The only feasible and better approach for it is incremental updating.

Data storage in the data warehouse:

Refresh the data

To provide the time variant data

To store the data as per the data model of the warehouse

Purging the data

To support the updating of the warehouse data

Some of the important designs for the data warehouse are:

Modular component design

Consideration of the parallel architecture

Consideration of the distributed architecture

Usage protection

Characteristics of available sources

Design of the metadata component

The fit of the data model

The major determining characteristics for the design of the warehouse is the architecture of the organizations distributed computing environment. The distributed warehouse and the federated warehouse are the two basic distributed architecture.There are some benefits from the distributed warehouse, some of them are:

Improved load balancing

Scalability of performance

Higher availability

Federated warehouse is a decentralized confederation of autonomous data warehouses. Each of them has its own metadata repository. Now a days large organizations start choosing a federated data marts instead of building a huge data warehouse.

Data Warehouse Architecture

A data warehouse architecture is a method of defining the overall architecture of data communication processing and presentation that exist for end-clients computing within the enterprise. Each data warehouse is different, but all are characterized by standard vital components.

Production applications such as payroll accounts payable product purchasing and inventory control are designed for online transaction processing (OLTP). Such applications gather detailed data from day to day operations.

Data Warehouse applications are designed to support the user ad-hoc data requirements, an activity recently dubbed online analytical processing (OLAP). These include applications such as forecasting, profiling, summary reporting, and trend analysis.

Production databases are updated continuously by either by hand or via OLTP applications. In contrast, a warehouse database is updated from operational systems periodically, usually during off-hours. As OLTP data accumulates in production databases, it is regularly extracted, filtered, and then loaded into a dedicated warehouse server that

is accessible to users. As the warehouse is populated, it must be restructured tables de-normalized, data cleansed of errors and redundancies and new fields and keys added to reflect the needs to the user for sorting, combining, and summarizing data.

Data warehouses and their architectures very depending upon the elements of an organization's situation.

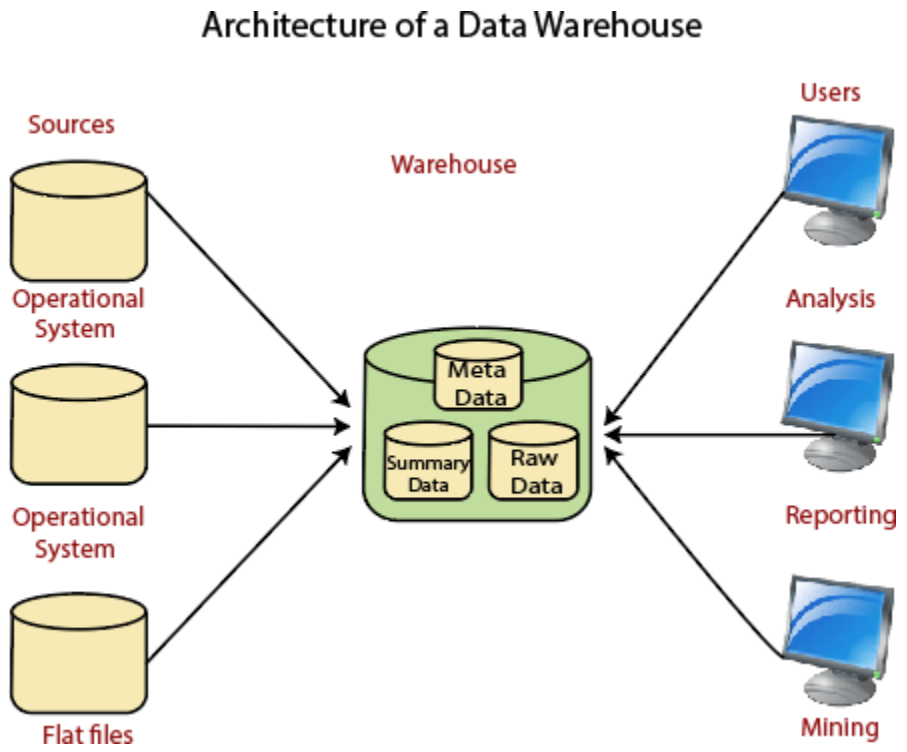
Three common architectures are:

Data Warehouse Architecture: Basic

Data Warehouse Architecture: With Staging Area

Data Warehouse Architecture: With Staging Area and Data Marts

Data Warehouse Architecture: Basic



Operational System

An operational system is a method used in data warehousing to refer to a system that is used to process the day-to-day transactions of an organization.

Flat Files

A Flat file system is a system of files in which transactional data is stored, and every file in the system must have a different name.

Meta Data

A set of data that defines and gives information about other data. Meta Data used in Data Warehouse for a variety of purpose, including:

Meta Data summarizes necessary information about data, which can make finding and work with particular instances of data more accessible. For example, author, data build, and data changed, and file size are examples of very basic document metadata.

Metadata is used to direct a query to the most appropriate data source.

Lightly and highly summarized data

The area of the data warehouse saves all the predefined lightly and highly summarized (aggregated) data generated by the warehouse manager.

The goals of the summarized information are to speed up query performance. The summarized record is updated continuously as new information is loaded into the warehouse.

End-User access Tools

The principal purpose of a data warehouse is to provide information to the business managers for strategic decision-making. These customers interact with the warehouse using end-client access tools.

The examples of some of the end-user access tools can be:

Reporting and Query Tools

Application Development Tools

Executive Information Systems Tools

Online Analytical Processing Tools

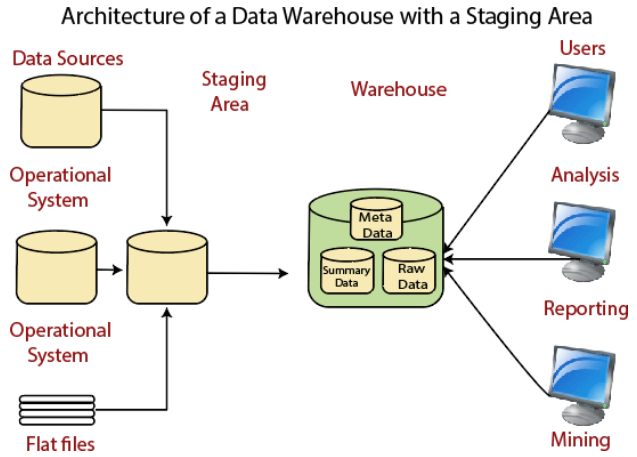
Data Mining Tools

Data Warehouse Architecture: With Staging Area

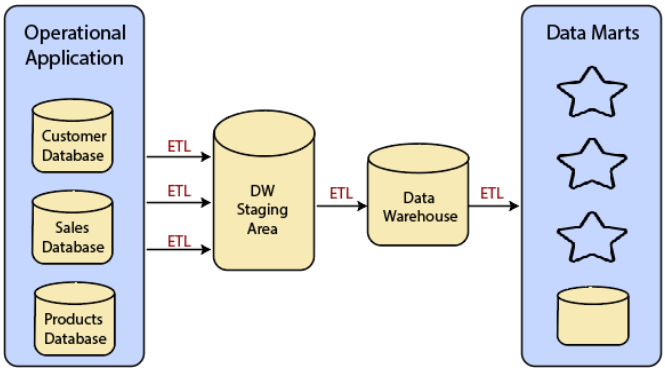
We must clean and process your operational information before put it into the warehouse.

We can do this programmatically, although data warehouses uses a staging area (A place where data is processed before entering the warehouse).

A staging area simplifies data cleansing and consolidation for operational method coming from multiple source systems, especially for enterprise data warehouses where all relevant data of an enterprise is consolidated.



Data Warehouse Staging Area is a temporary location where a record from source systems is copied.

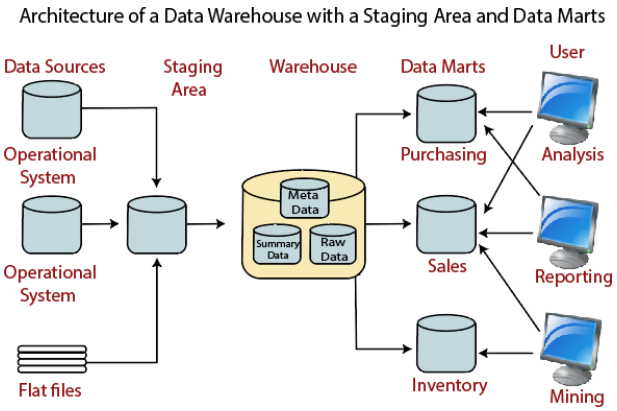


Data Warehouse Architecture: With Staging Area and Data Marts

We may want to customize our warehouse's architecture for multiple groups within our organization.

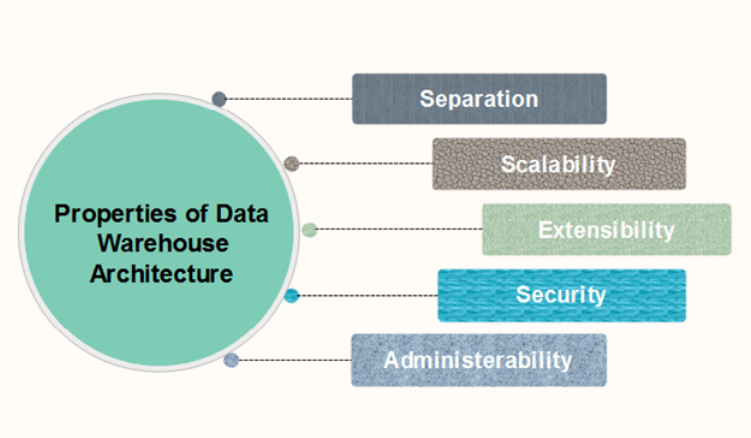
We can do this by adding data marts. A data mart is a segment of a data warehouses that can provided information for reporting and analysis on a section, unit, department or operation in the company, e.g., sales, payroll, production, etc.

The figure illustrates an example where purchasing, sales, and stocks are separated. In this example, a financial analyst wants to analyze historical data for purchases and sales or mine historical information to make predictions about customer behavior.



Properties of Data Warehouse Architectures

The following architecture properties are necessary for a data warehouse system:



Separation: Analytical and transactional processing should be keep apart as much as possible.

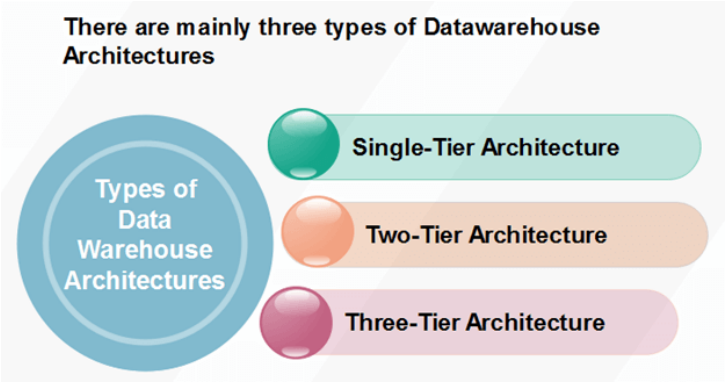
Scalability: Hardware and software architectures should be simple to upgrade the data volume, which has to be managed and processed, and the number of user's requirements, which have to be met, progressively increase.

Extensibility: The architecture should be able to perform new operations and technologies without redesigning the whole system.

Security: Monitoring accesses are necessary because of the strategic data stored in the data warehouses.

Administerability: Data Warehouse management should not be complicated.

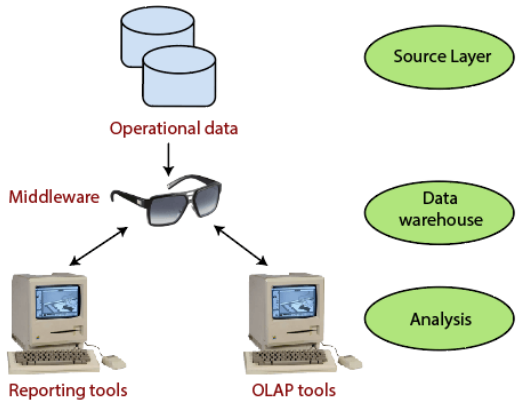
Types of Data Warehouse Architectures



Single-Tier Architecture

Single-Tier architecture is not periodically used in practice. Its purpose is to minimize the amount of data stored to reach this goal; it removes data redundancies.

The figure shows the only layer physically available is the source layer. In this method, data warehouses are virtual. This means that the data warehouse is implemented as a multidimensional view of operational data created by specific middleware, or an intermediate processing layer.

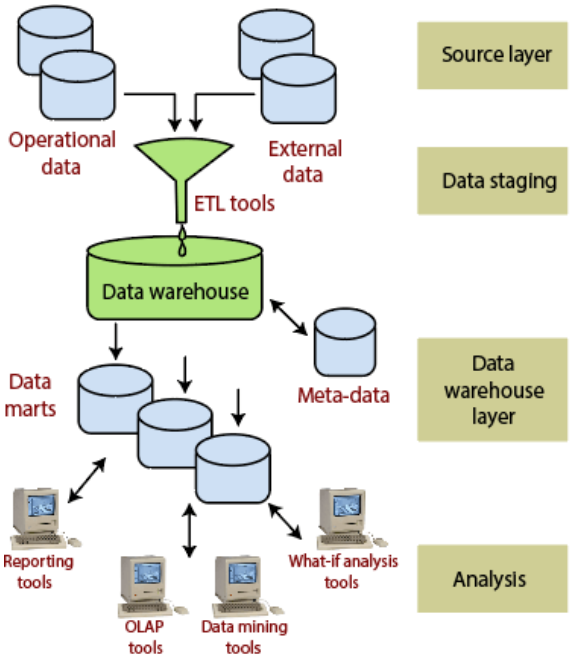


Single-Tier Data Warehouse Architecture

The vulnerability of this architecture lies in its failure to meet the requirement for separation between analytical and transactional processing. Analysis queries are agreed to operational data after the middleware interprets them. In this way, queries affect transactional workloads.

Two-Tier Architecture

The requirement for separation plays an essential role in defining the two-tier architecture for a data warehouse system, as shown in fig:



Two-Tier Data Warehouse Architecture

Although it is typically called two-layer architecture to highlight a separation between physically available sources and data warehouses, in fact, consists of four subsequent data flow stages:

Source layer: A data warehouse system uses a heterogeneous source of data. That data is stored initially to corporate relational databases or legacy databases, or it may come from an information system outside the corporate walls.

Data Staging: The data stored to the source should be extracted, cleansed to remove inconsistencies and fill gaps, and integrated to merge heterogeneous sources into one standard schema. The so-named Extraction, Transformation, and Loading Tools (ETL) can combine heterogeneous schemata, extract, transform, cleanse, validate, filter, and load source data into a data warehouse.

Data Warehouse layer: Information is saved to one logically centralized individual repository: a data warehouse. The data warehouses can be directly accessed, but it can also be used as a source for creating data marts, which partially replicate data warehouse contents and are designed for specific enterprise departments. Meta-data repositories store information on sources, access procedures, data staging, users, data mart schema, and so on.

Analysis: In this layer, integrated data is efficiently, and flexible accessed to issue reports, dynamically analyze information, and simulate hypothetical business scenarios. It should feature aggregate information navigators, complex query optimizers, and customer- friendly GUIs.

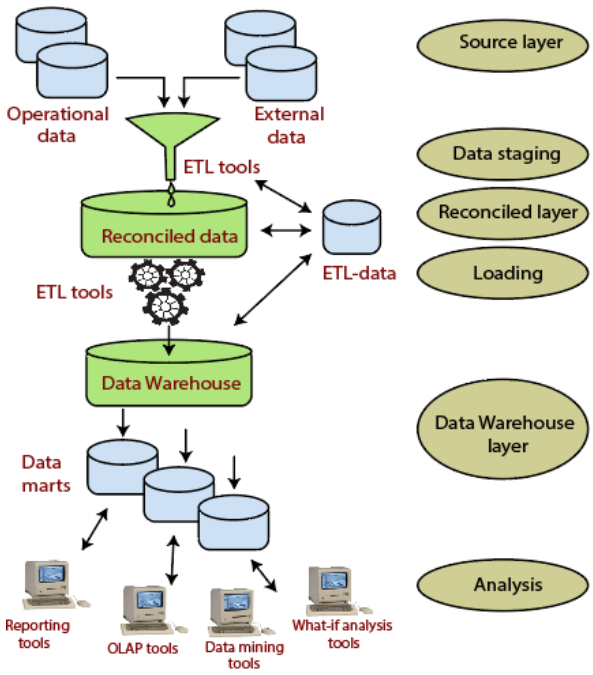
DataBase Architectures for Parallel Processing

Three-Tier Architecture

The three-tier architecture consists of the source layer (containing multiple source system), the reconciled layer and the data warehouse layer (containing both data warehouses and data marts). The reconciled layer sits between the source data and data warehouse.

The main advantage of the reconciled layer is that it creates a standard reference data model for whole enterprise. At the same time, it separates the problems of source data extraction and integration from those of data warehouse population. In some cases, the reconciled layer is also directly used to accomplish better some operational tasks, such as producing daily reports that cannot be satisfactorily prepared using the corporate

applications or generating data flows to feed external processes periodically to benefit from cleaning and integration. This architecture is especially useful for the extensive, enterprise-wide systems. A disadvantage of this structure is the extra file storage space used through the extra redundant reconciled layer. It also makes the analytical tools a little further away from being real-time.



Three-Tier Architecture for a data warehouse system

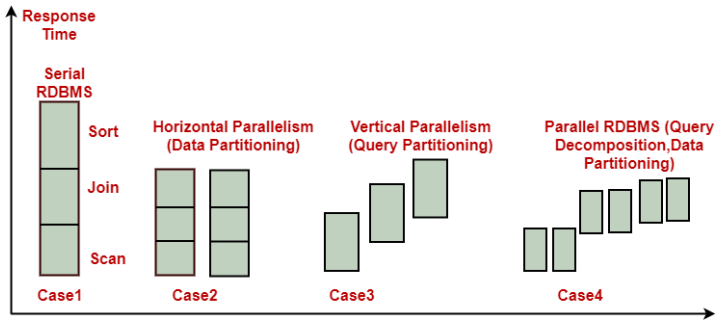
Parallel DBMS Vendors

Parallelism is used to support speedup, where queries are executed faster because more resources, such as processors and disks, are provided. Parallelism is also used to provide scale-up, where increasing workloads are managed without increase response-time, via an increase in the degree of parallelism.

Different architectures for parallel database systems are shared-memory, shared-disk, shared-nothing, and hierarchical structures.

Horizontal Parallelism: It means that the database is partitioned across multiple disks, and parallel processing occurs within a specific task (i.e., table scan) that is performed concurrently on different processors against different sets of data.

Vertical Parallelism: It occurs among various tasks. All component query operations (i.e., scan, join, and sort) are executed in parallel in a pipelined fashion. In other words, an output from one function (e.g., join) as soon as records become available.



Intraquery Parallelism:

Intraquery parallelism defines the execution of a single query in parallel on multiple processors and disks. Using intraquery parallelism is essential for speeding up long- running queries. Interquery parallelism does not help in this function since each query is run sequentially. To improve the situation, many DBMS vendors developed versions of their products that utilized intraquery parallelism.This application of parallelism decomposes the serial SQL, query into lower-level operations such as scan, join, sort, and aggregation. These lower-level operations are executed concurrently, in parallel. Interquery Parallelism In interquery parallelism, different queries or transaction execute in parallel with one another. This form of parallelism can increase transactions throughput. The response times of individual transactions are not faster than they would be if the transactions were run in isolation. Thus, the primary use of interquery parallelism is to scale up a transaction processing system to support a more significant number of transactions per second.Database vendors started to take advantage of parallel hardware architectures by implementing multiserver and multithreaded systems designed to handle a large number of client requests efficiently. This approach naturally resulted in interquery parallelism, in which different server threads (or processes) handle multiple requests at the same time.

Interquery parallelism has been successfully implemented on SMP systems, where it increased the throughput and allowed the support of more concurrent users.

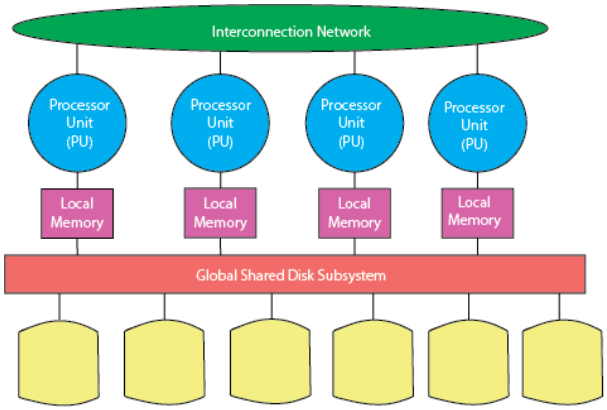
Shared Disk Architecture Shared-disk architecture implements a concept of shared ownership of the entire database between RDBMS servers, each of which is running on a node of a distributed memory system.

Each RDBMS server can read, write, update, and delete information from the same shared database, which would need the system to implement a form of a distributed lock manager (DLM).

DLM components can be found in hardware, the operating system, and separate software layer, all depending on the system vendor.

On the positive side, shared-disk architectures can reduce performance bottlenecks resulting from data skew (uneven distribution of data), and can significantly increase system availability.

The shared-disk distributed memory design eliminates the memory access bottleneck typically of large SMP systems and helps reduce DBMS dependency on data partitioning.



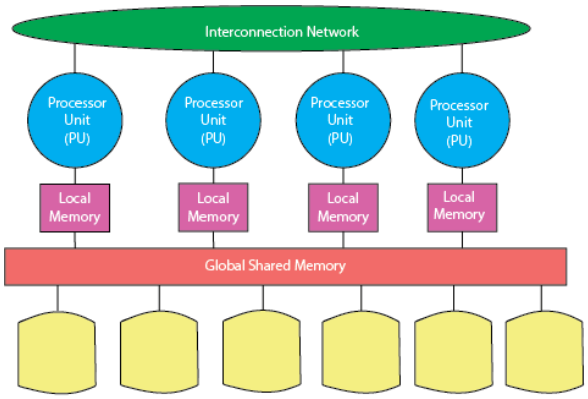
Distributed memory shared-disk architecture

Shared-Memory Architecture

Shared-memory or shared-everything style is the traditional approach of implementing an RDBMS on SMP hardware.

It is relatively simple to implement and has been very successful up to the point where it runs into the scalability limitations of the shared-everything architecture.

The key point of this technique is that a single RDBMS server can probably apply all processors, access all memory, and access the entire database, thus providing the client with a consistent single system image.



Shared-Memory Architecture

In shared-memory SMP systems, the DBMS considers that the multiple database components executing SQL statements communicate with each other by exchanging messages and information via the shared memory.

All processors have access to all data, which is partitioned across local disks.

Shared-Nothing Architecture

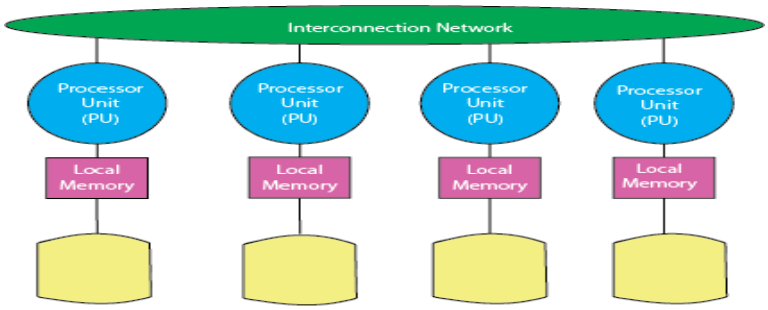
In a shared-nothing distributed memory environment, the data is partitioned across all disks, and the DBMS is "partitioned" across multiple co-servers, each of which resides on individual nodes of the parallel system and has an ownership of its disk and thus its database partition.

A shared-nothing RDBMS parallelizes the execution of a SQL query across multiple processing nodes.

Each processor has its memory and disk and communicates with other processors by exchanging messages and data over the interconnection network.

This architecture is optimized specifically for the MPP and cluster systems.

The shared-nothing architectures offer near-linear scalability. The number of processor nodes is limited only by the hardware platform limitations (and budgetary constraints), and each node itself can be a powerful SMP system.



Shared-Nothing Architecture

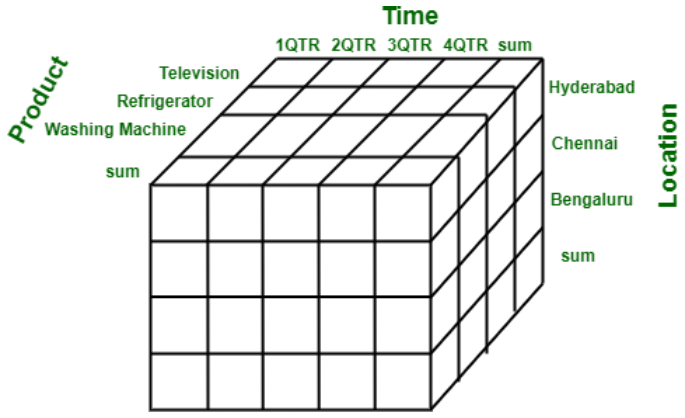
Multidimensional Data Model

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.

It represents data in the form of data cubes. Data cubes allow to model and view the data from many dimensions and perspectives. It is defined by dimensions and facts and is represented by a fact table. Facts are numerical measures and fact tables contain measures of the related dimensional tables or names of the facts.



Multidimensional Data Representation

Working on a Multidimensional Data Model

On the basis of the pre-decided steps, the Multidimensional Data Model works.

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".

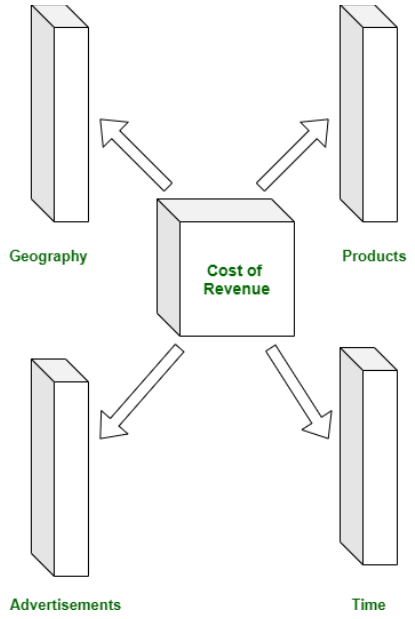
Stage 4 : Preparing the actual-time factors and their respective qualities : In the fourth stage, the factors which are recognized in the previous step are used further for identifying the related qualities. These qualities are also known as "attributes" in the database.

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 :

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.



Example 1

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

2D factory data

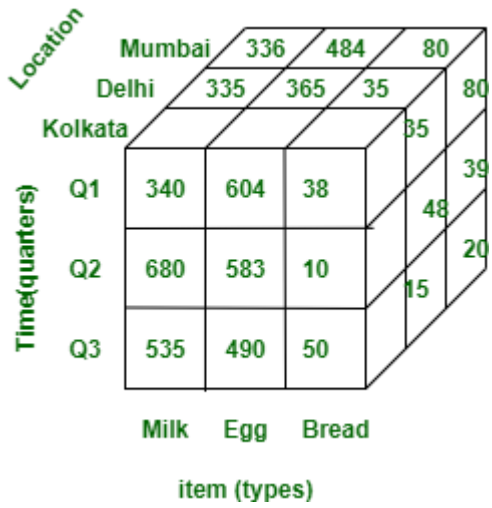
In the above given presentation, the factory's sales for Bangalore are, for the time dimension, which is organized into quarters and the dimension of items, which is sorted according to the kind of item which is sold. The facts here are represented in rupees (in thousands).

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, 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

3D data representation as 2D

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



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.

The compatibility in this type of database is an upliftment for projects having lower bandwidth for maintenance staff.

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.

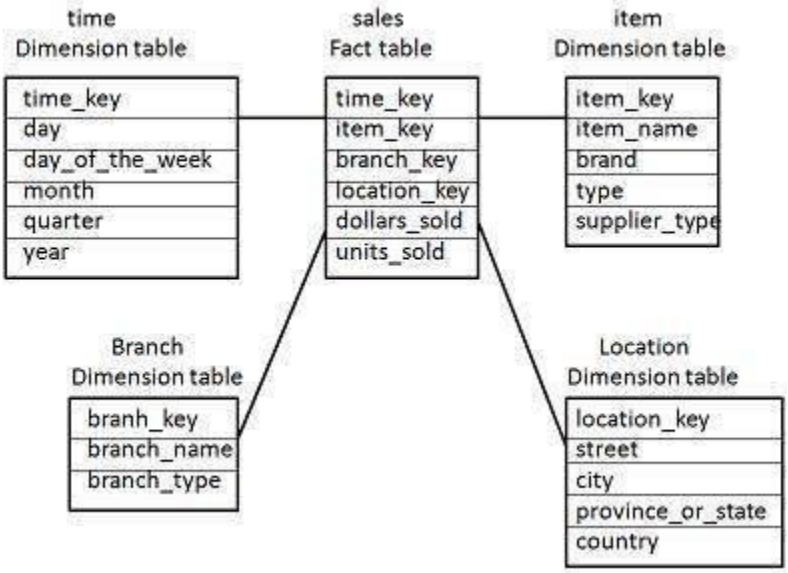
It is complicated in nature due to which the databases are generally dynamic in design.
The path to achieving the end product is complicated most of the time.
As the Multi Dimensional Data Model has complicated systems, databases have a large number of databases due to which the system is very insecure when there is a security break.

Data Warehouse Schemas for Decision Support

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

Each dimension in a star schema is represented with only one-dimension table.
This dimension table contains the set of attributes.
The following diagram shows the sales data of a company with respect to the four dimensions, namely time, item, branch, and location.



There is a fact table at the center. It contains the keys to each of four dimensions.
The fact table also contains the attributes, namely dollars sold and units sold.

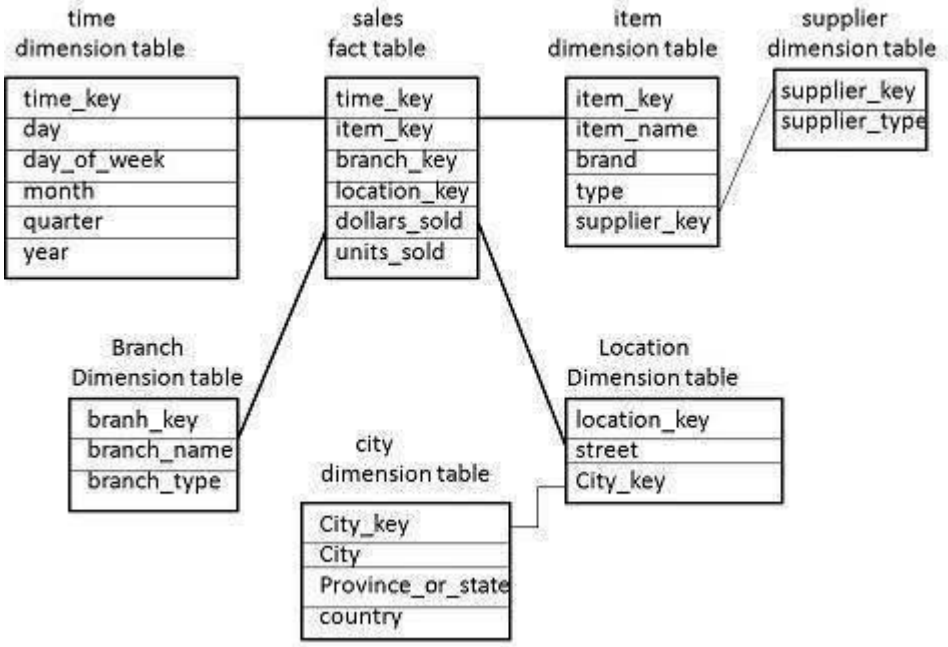
Note − Each dimension has only one dimension table and each table holds a set of attributes. For example, the location dimension table contains the attribute set {location_key, street, city, province_or_state,country}. This constraint may cause data redundancy. For example, "Vancouver" and "Victoria"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

Snowflake Schema

Some dimension tables in the Snowflake schema are normalized.

The normalization splits up the data into additional tables.

Unlike Star schema, the dimensions table in a snowflake schema are normalized. For example, the item dimension table in star schema is normalized and split into two dimension tables, namely item and supplier table.



Now the item dimension table contains the attributes item_key, item_name, type, brand, and supplier-key.

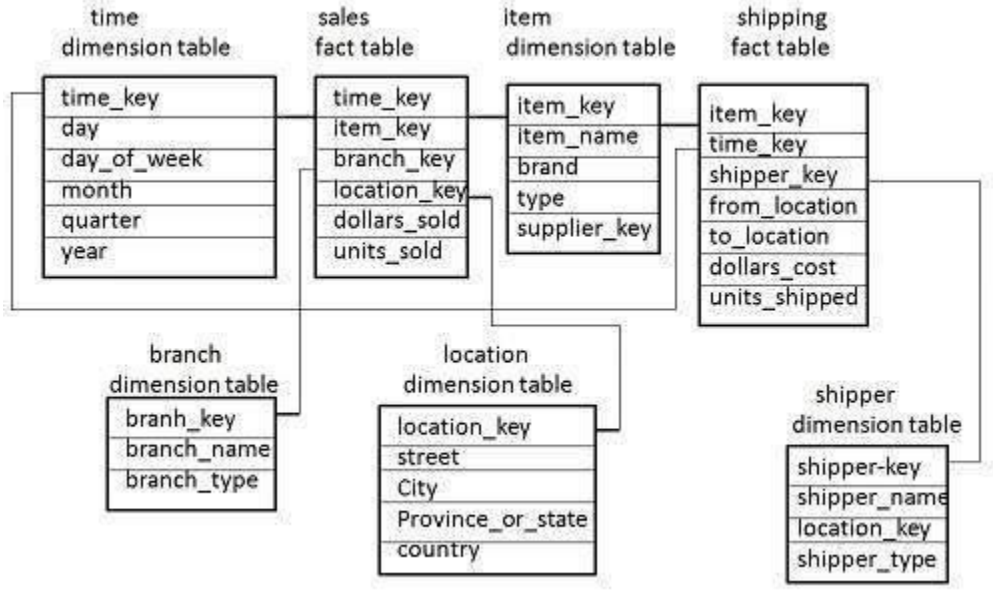
The supplier key is linked to the supplier dimension table. The supplier dimension table contains the attributes supplier_key and supplier_type.

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

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.



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.

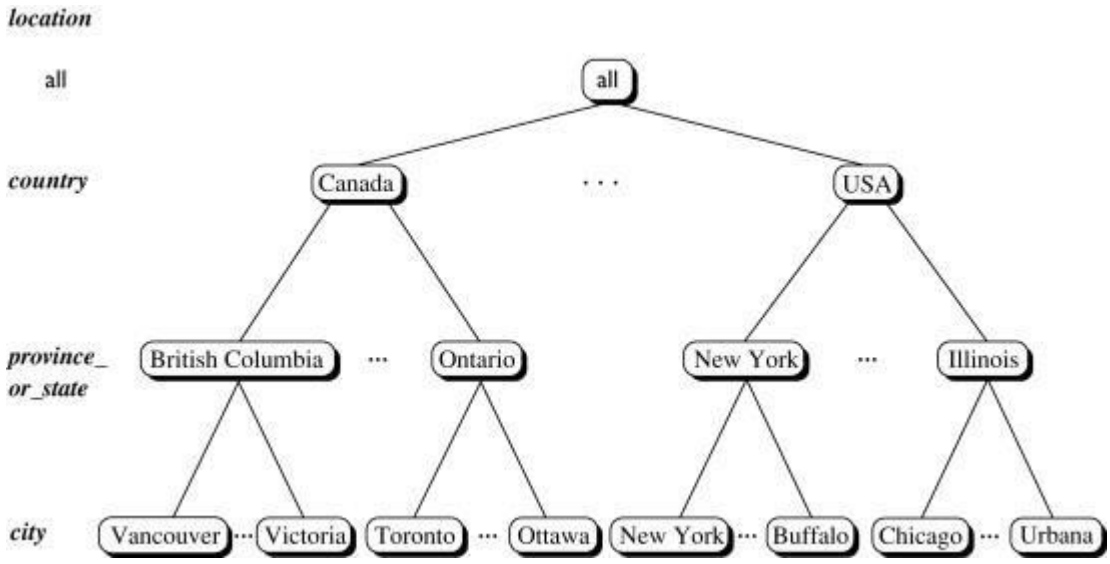
The shipping fact table also contains two measures, namely dollars sold and units sold.

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.

Concept Hierarchies

A concept hierarchy defines a sequence of mappings from a set of low-level concepts to higher-level, more general concepts. Consider a concept hierarchy for

the dimension location. City values for location include Vancouver, Toronto, New York, and Chicago. Each city, however, can be mapped to the province or state to which it belongs. For example, Vancouver can be mapped to British Columbia, and Chicago to Illinois. The provinces and states can in turn be mapped to the country (e.g., Canada or the United States) to which they belong. These mappings form a concept hierarchy for the dimension location, mapping a set of low-level concepts (i.e., cities) to higher-level, more general concepts (i.e., countries). This concept hierarchy is illustrated in Figure



A concept hierarchy represents a series of mappings from a set of low-level concepts to larger-level, more general concepts. Concept hierarchy organizes information or concepts in a hierarchical structure or a specific partial order, which are used for defining knowledge in brief, high-level methods, and creating possible mining knowledge at several levels of abstraction.

A conceptual hierarchy includes a set of nodes organized in a tree, where the nodes define values of an attribute known as concepts. A specific node, "ANY", is constrained for the root of the tree. A number is created to the level of each node in a conceptual hierarchy. The level of the root node is one. The level of a non-root node is one more the level of its parent level number.

Because values are defined by nodes, the levels of nodes can also be used to describe the levels of values. Concept hierarchy enables raw information to be managed at a higher and more generalized level of abstraction. There are several types of concept hierarchies which are as follows −

Schema Hierarchy − Schema hierarchy represents the total or partial order between attributes in the database. It can define existing semantic relationships between attributes. In a database, more than one schema hierarchy can be generated by using multiple sequences and grouping of attributes.

Set-Grouping Hierarchy − A set-grouping hierarchy constructs values for a given attribute or dimension into groups or constant range values. It is also known as instance hierarchy because the partial series of the hierarchy is represented on the set of instances or values of an attribute. These hierarchies have more functional sense and are so approved than other hierarchies.

Operation-Derived Hierarchy − Operation-derived hierarchy is represented by a set of operations on the data. These operations are defined by users, professionals, or the data mining system. These hierarchies are usually represented for mathematical attributes.

Such operations can be as easy as range value comparison, as difficult as a data clustering and data distribution analysis algorithm.

Rule-based Hierarchy − In a rule-based hierarchy either a whole concept hierarchy or an allocation of it is represented by a set of rules and is computed dynamically based on the current information and rule definition. A lattice-like architecture is used for graphically defining this type of hierarchy, in which each child-parent route is connected with a generalization rule.

The static and dynamic generation of concept hierarchy is based on data sets. In this context, the generation of a concept hierarchy depends on a static or dynamic data set is known as the static or dynamic generation of concept hierarchy.

OLAP

OLAP stands for On-Line Analytical Processing. OLAP is a classification of software technology which authorizes analysts, managers, and executives to gain insight into information through fast, consistent, interactive access in a wide variety of possible views of data that has been transformed from raw information to reflect the real dimensionality of the enterprise as understood by the clients.

OLAP implement the multidimensional analysis of business information and support the capability for complex estimations, trend analysis, and sophisticated data modeling. It is rapidly enhancing the essential foundation for Intelligent Solutions containing Business Performance Management, Planning, Budgeting, Forecasting, Financial Documenting, Analysis, Simulation-Models, Knowledge Discovery, and Data Warehouses Reporting. OLAP enables end-clients to perform ad hoc analysis of record in multiple dimensions, providing the insight and understanding they require for better decision making.

OLAP applications are used by a variety of the functions of an organization.

Finance and accounting:

Budgeting

Activity-based costing

Financial performance analysis

And financial modeling

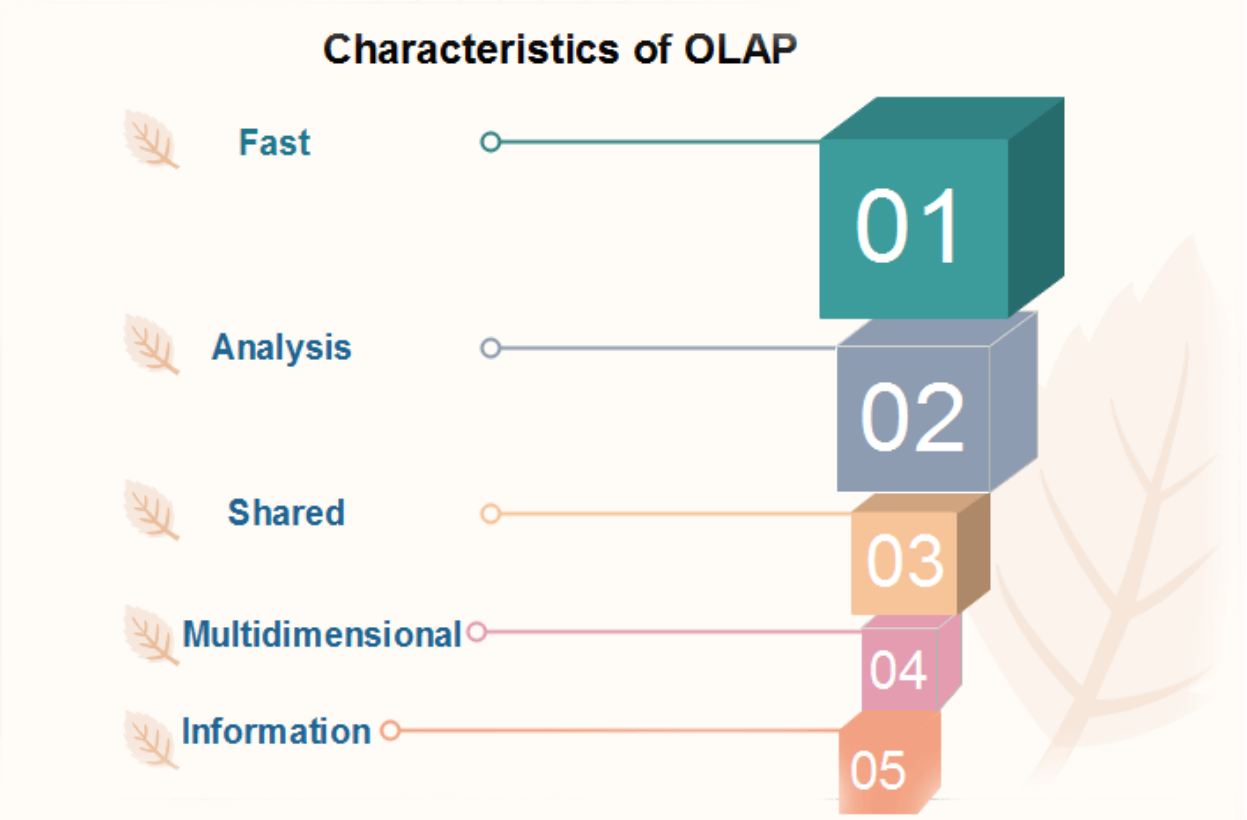
Sales and Marketing

Sales analysis and forecasting

Market research analysis

Promotion analysis
Customer analysis
Market and customer segmentation
Production

Production planning
Defect analysis
OLAP cubes have two main purposes. The first is to provide business users with a data model more intuitive to them than a tabular model. This model is called a Dimensional Model.
Characteristics of OLAP
In the FASMI characteristics of OLAP methods, the term derived from the first letters of the characteristics are:



Fast

It defines which the system targeted to deliver the most feedback to the client within about five seconds, with the elementary analysis taking no more than one second and very few taking more than 20 seconds.

Analysis

It defines which the method can cope with any business logic and statistical analysis that is relevant for the function and the user, keep it easy enough for the target client. Although some preprogramming may be needed we do not think it acceptable if all application definitions have to be allow the user to define new Adhoc calculations as part of the analysis and to document on the data in any desired method, without having to program so we excludes products (like Oracle Discoverer) that do not allow the user to define new Adhoc calculation as part of the analysis and to document on the data in any desired product that do not allow adequate end user-oriented calculation flexibility.

Share

It defines which the system tools all the security requirements for understanding and, if multiple write connection is needed, concurrent update location at an appropriated level, not all functions need customer to write data back, but for the increasing number which does, the system should be able to manage multiple updates in a timely, secure manner.

Multidimensional

This is the basic requirement. OLAP system must provide a multidimensional conceptual view of the data, including full support for hierarchies, as this is certainly the most logical method to analyze business and organizations.

Information

The system should be able to hold all the data needed by the applications. Data sparsity should be handled in an efficient manner.

The main characteristics of OLAP are as follows:

Multidimensional conceptual view: OLAP systems let business users have a dimensional and logical view of the data in the data warehouse. It helps in carrying slice and dice operations.

Multi-User Support: Since the OLAP techniques are shared, the OLAP operation should provide normal database operations, containing retrieval, update, adequacy control, integrity, and security.

Accessibility: OLAP acts as a mediator between data warehouses and front-end. The OLAP operations should be sitting between data sources (e.g., data warehouses) and an OLAP front-end.

Storing OLAP results: OLAP results are kept separate from data sources.

Uniform documenting performance: Increasing the number of dimensions or database size should not significantly degrade the reporting performance of the OLAP system.

OLAP provides for distinguishing between zero values and missing values so that aggregates are computed correctly.

OLAP system should ignore all missing values and compute correct aggregate values.

OLAP facilitate interactive query and complex analysis for the users.

OLAP allows users to drill down for greater details or roll up for aggregations of metrics along a single business dimension or across multiple dimension.

OLAP provides the ability to perform intricate calculations and comparisons.

OLAP presents results in a number of meaningful ways, including charts and graphs.

Typical OLAP operations

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)

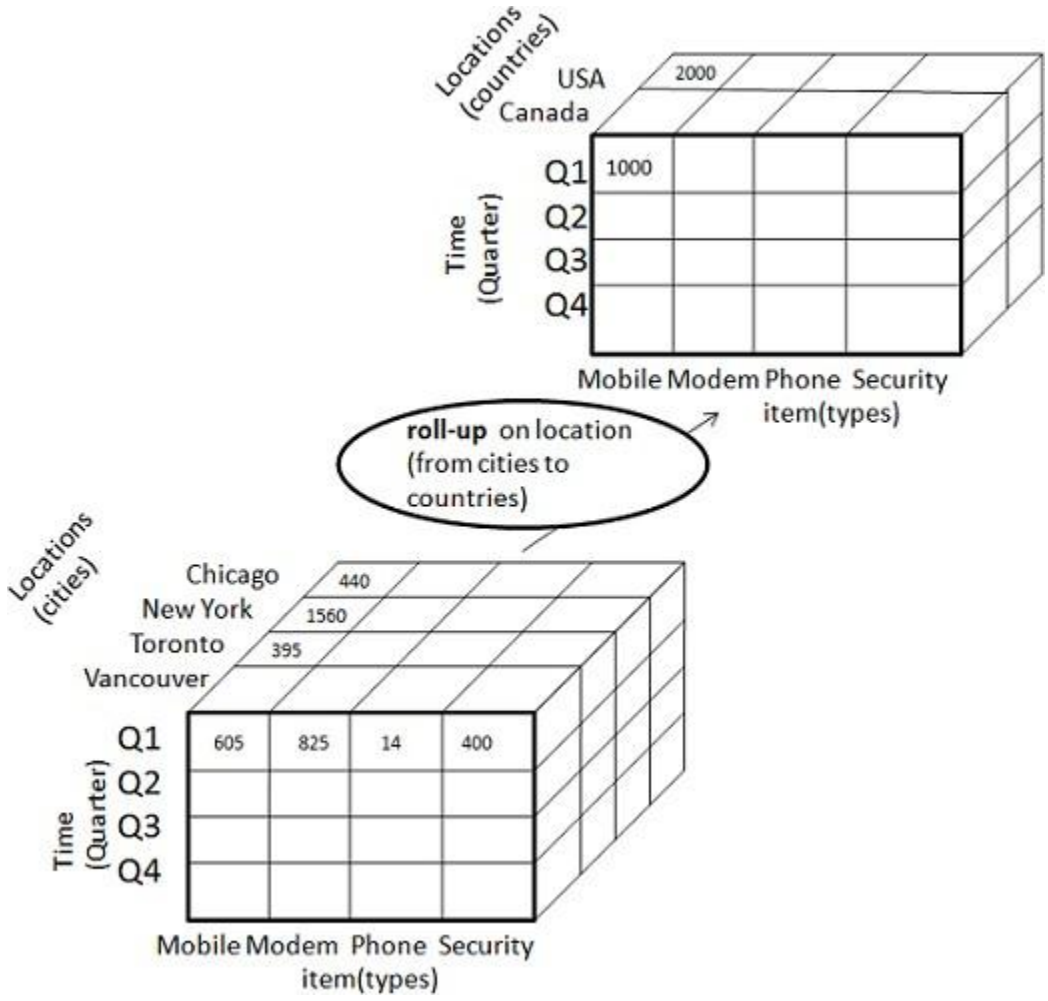
Roll-up

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.

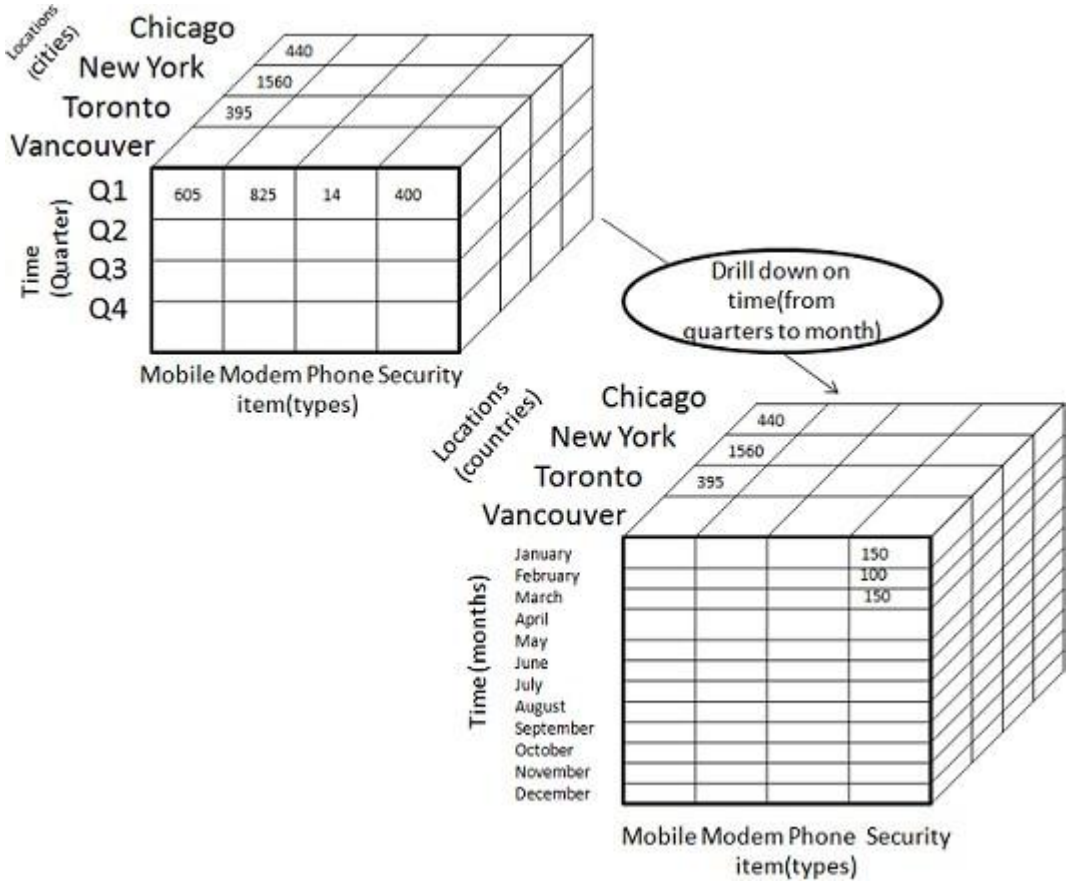
Drill-down

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."

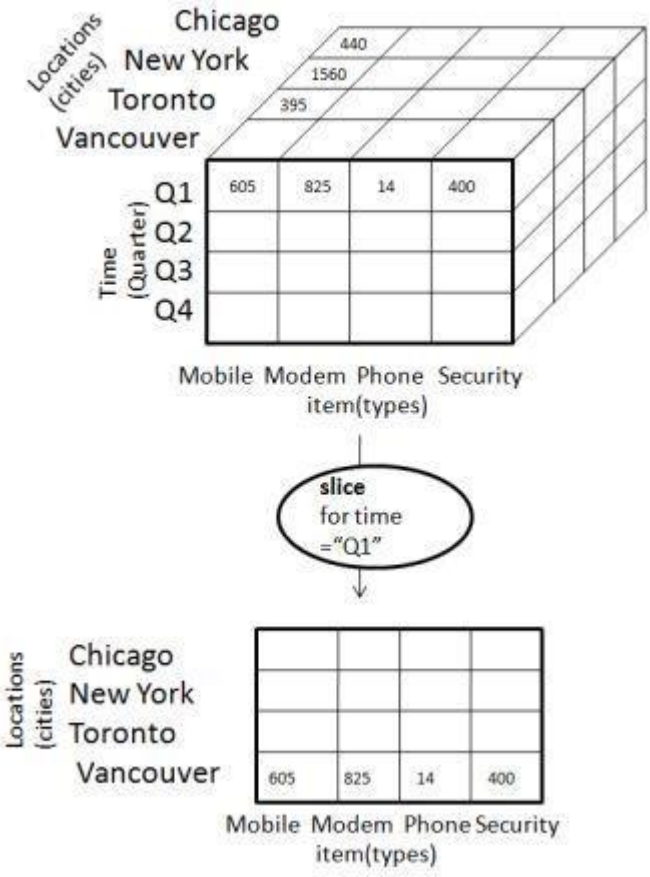
On drilling down, the time dimension is descended from the level of quarter to the level of month.

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.

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.

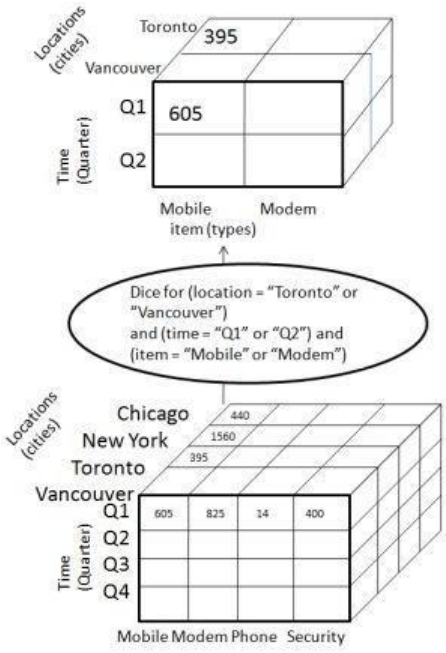


Here Slice is performed for the dimension "time" using the criterion time = "Q1".

It will form a new sub-cube by selecting one or more dimensions.

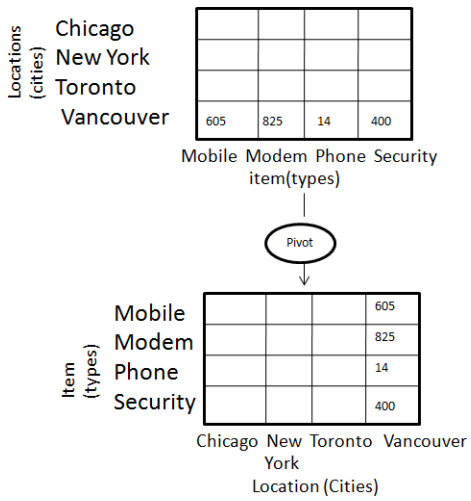
Dice

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")
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.



OLAP V/S OLTP

OLAP (Online analytical processing)	OLTP (Online transaction processing)
Consists of historical data from various Databases.	Consists only operational current data.
It is subject oriented. Used for Data Mining, Analytics, Decision making,etc.	It is application oriented. Used for business tasks.
The data is used in planning, problem solving and decision making.	The data is used to perform day to day fundamental operations.
It reveals a snapshot of present business tasks.	It provides a multi-dimensional view of different business tasks.
Large amount of data is stored typically in TB, PB	The size of the data is relatively small as the historical data is archived. For ex MB, GB
Relatively slow as the amount of data involved is large. Queries may take hours.	Very Fast as the queries operate on 5% of the data.

It only need backup from time to time as compared to OLTP.	Backup and recovery process is maintained religiously
This data is generally managed by CEO, MD, GM.	This data is managed by clerks, managers.
Only read and rarely write operation.	Both read and write operations.

UNIT II

Data Mining – Introduction : Introduction to Data Mining Systems – Knowledge Discovery Process - Data Mining Techniques – Issues – applications- Data Objects and attribute types, Statistical description of data, Data Preprocessing – Cleaning, Integration, Reduction, Transformation and discretization, Data Visualization, Data similarity and dissimilarity measures.

DATA MINING

The process of extracting information to identify patterns, trends, and useful data that would allow the business to take the data-driven decision from huge sets of data is called Data Mining.

In other words, we can say that Data Mining is the process of investigating hidden patterns of information to various perspectives for categorization into useful data, which is collected and assembled in particular areas such as data warehouses, efficient analysis, data mining algorithm, helping decision making and other data requirement to eventually cost-cutting and generating revenue.

Data mining is the act of automatically searching for large stores of information to find trends and patterns that go beyond simple analysis procedures. Data mining utilizes complex mathematical algorithms for data segments and evaluates the probability of future events. Data Mining is also called Knowledge Discovery of Data (KDD).

Data Mining is a process used by organizations to extract specific data from huge databases to solve business problems. It primarily turns raw data into useful information.

Data Mining is similar to Data Science carried out by a person, in a specific situation, on a particular data set, with an objective. This process includes various types of services such as text mining, web mining, audio and video mining, pictorial data mining, and social media mining. It is done through software that is simple or highly specific. By outsourcing data mining, all the work can be done faster with low operation costs. Specialized firms can also use new technologies to collect data that is impossible to locate manually. There are tonnes of information available on various platforms, but very little knowledge is accessible. The biggest challenge is to analyze the data to extract important information that can be used to solve a problem or for company development. There are many powerful instruments and techniques available to mine data and find better insight from it.

Types of Data Mining

Data mining can be performed on the following types of data:

Relational Database:

A relational database is a collection of multiple data sets formally organized by tables, records, and columns from which data can be accessed in various ways without having to recognize the database tables. Tables convey and share information, which facilitates data searchability, reporting, and organization.

Data warehouses:

A Data Warehouse is the technology that collects the data from various sources within the organization to provide meaningful business insights. The huge amount of data comes from multiple places such as Marketing and Finance. The extracted data is utilized for analytical purposes and helps in decision- making for a business organization. The data warehouse is designed for the analysis of data rather than transaction processing.

Data Repositories:

The Data Repository generally refers to a destination for data storage. However, many IT professionals utilize the term more clearly to refer to a specific kind of setup within an IT structure. For example, a group of databases, where an organization has kept various kinds of information.

Object-Relational Database:

A combination of an object-oriented database model and relational database model is called an object-relational model. It supports Classes, Objects, Inheritance, etc.

One of the primary objectives of the Object-relational data model is to close the gap between the Relational database and the object-oriented model practices frequently utilized in many programming languages, for example, C++, Java, C#, and so on.

Transactional Database:

A transactional database refers to a database management system (DBMS) that has the potential to undo a database transaction if it is not performed appropriately. Even though this was a unique capability a very long while back, today, most of the relational database systems support transactional database activities.

Advantages of Data Mining

The Data Mining technique enables organizations to obtain knowledge-based data.

Data mining enables organizations to make lucrative modifications in operation and production.

Compared with other statistical data applications, data mining is a cost-efficient.

Data Mining helps the decision-making process of an organization.

It Facilitates the automated discovery of hidden patterns as well as the prediction of trends and behaviors.

It can be induced in the new system as well as the existing platforms.

It is a quick process that makes it easy for new users to analyze enormous amounts of data in a short time.

Disadvantages of Data Mining

There is a probability that the organizations may sell useful data of customers to other organizations for money. As per the report, American Express has sold credit card purchases of their customers to other organizations.

Many data mining analytics software is difficult to operate and needs advance training to work on.

Different data mining instruments operate in distinct ways due to the different algorithms used in their design. Therefore, the selection of the right data mining tools is a very challenging task.

The data mining techniques are not precise, so that it may lead to severe consequences in certain conditions.

Knowledge Discovery Process

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

Why we need Data Mining?

Volume of information is increasing everyday that we can handle from business transactions, scientific data, sensor data, Pictures, videos, etc. So, we need a system that will be capable of extracting essence of information available and that can automatically generate report,

views or summary of data for better decision-making.

Why Data Mining is used in Business?

Data mining is used in business to make better managerial decisions by:

Automatic summarization of data

Extracting essence of information stored.

Discovering patterns in raw data.

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.

Steps Involved in KDD Process:

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.

Data Integration: Data integration is defined as heterogeneous data from multiple sources combined in a common source(DataWarehouse).

Data integration using Data Migration tools.

Data integration using Data Synchronization tools.

Data integration using ETL(Extract-Load-Transformation) process.

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.

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.

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.

Pattern Evaluation: Pattern Evaluation is definedas identifying strictly increasing patterns representing knowledge based on given measures.

Find interestingness score of each pattern.

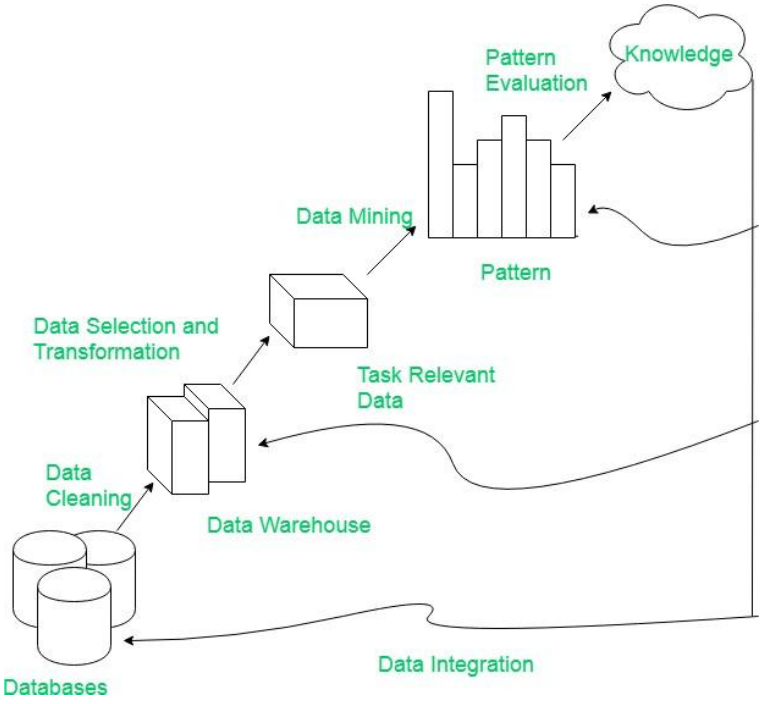
Uses summarization and Visualization to make data understandable by user.

Knowledge representation: Knowledge representation is defined as technique which utilizes visualization tools to represent data mining results.

Generate reports.

Generate tables.

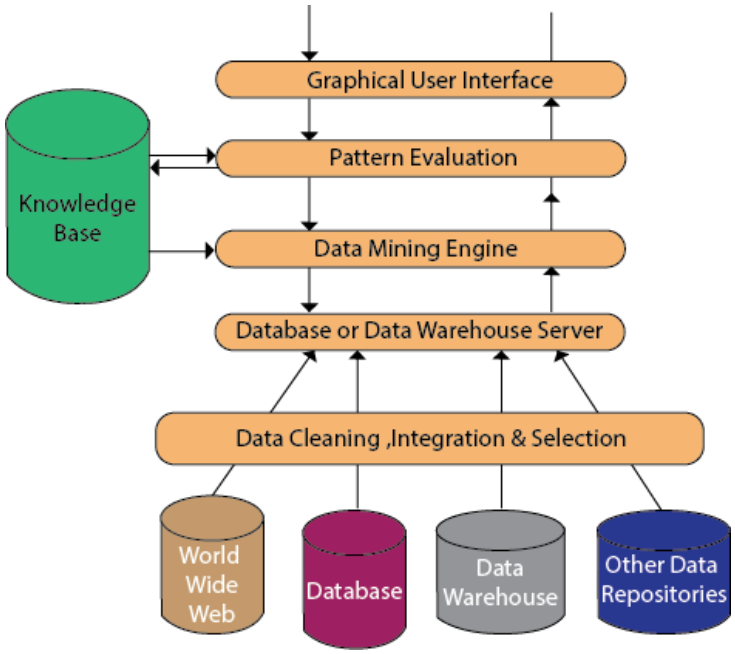
Generate discriminant rules, classification rules, characterization rules, etc.



KDP PROCESS

Data Mining Architecture

The significant components of data mining systems are a data source, data mining engine, data warehouse server, the pattern evaluation module, graphical user interface, and knowledge base.



Data Source:

The actual source of data is the Database, data warehouse, World Wide Web (WWW), text files, and other documents. You need a huge amount of historical data for data mining to be successful. Organizations typically store data in databases or data warehouses. Data warehouses may comprise one or more databases, text files spreadsheets, or other repositories of data. Sometimes, even plain text files or spreadsheets may contain information. Another primary source of data is the World Wide Web or the internet.

Different processes:

Before passing the data to the database or data warehouse server, the data must be cleaned, integrated, and selected. As the information comes from various sources and in different formats, it can't be used directly for the data mining procedure because the data may not be complete and accurate. So, the first data requires to be cleaned and unified. More information than needed will be collected from various data sources, and only the data of interest will have to be selected and passed to the server. These procedures are not as easy as we think. Several methods may be performed on the data as part of selection, integration, and cleaning.

Database or Data Warehouse Server:

The database or data warehouse server consists of the original data that is ready to be processed. Hence, the server is cause for retrieving the relevant data that is based on data mining as per user request.

Data Mining Engine:

The data mining engine is a major component of any data mining system. It contains several modules for operating data mining tasks, including association, characterization, classification, clustering, prediction, time-series analysis, etc.

In other words, we can say data mining is the root of our data mining architecture. It comprises instruments and software used to obtain insights and knowledge from data collected from various data sources and stored within the data warehouse.

Pattern Evaluation Module:

The Pattern evaluation module is primarily responsible for the measure of investigation of the pattern by using a threshold value. It collaborates with the data mining engine to focus the search on exciting patterns.

This segment commonly employs stake measures that cooperate with the data mining modules to focus the search towards fascinating patterns. It might utilize a stake threshold to filter out discovered patterns. On the other hand, the pattern evaluation module might be coordinated with the mining module, depending on the implementation of the data mining techniques used. For efficient data mining, it is abnormally suggested to push the evaluation of pattern stake as much as possible into the mining procedure to confine the search to only fascinating patterns.

Graphical User Interface:

The graphical user interface (GUI) module communicates between the data mining system and the user. This module helps the user to easily and efficiently use the system without knowing the complexity of the process. This module cooperates with the data mining system when the user specifies a query or a task and displays the results.

Knowledge Base:

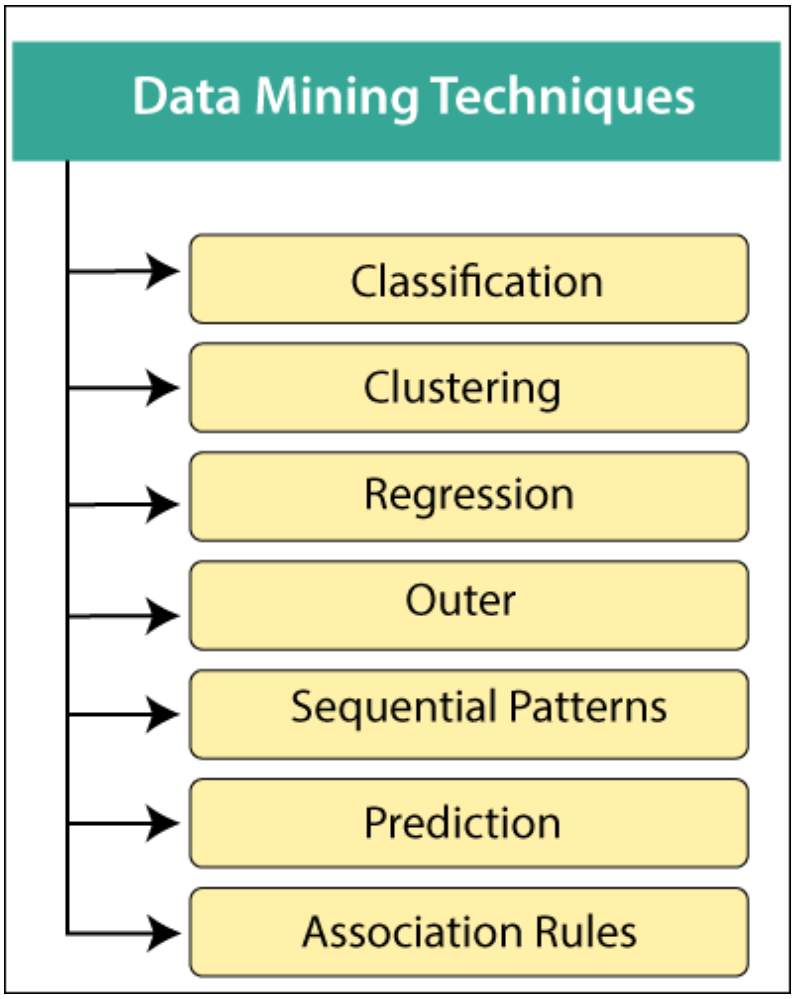
The knowledge base is helpful in the entire process of data mining. It might be helpful to guide the search or evaluate the stake of the result patterns. The knowledge base may even contain user views and data from user experiences that might be helpful in the data mining process. The data mining engine may receive inputs from the knowledge base to make the result more accurate and reliable. The pattern assessment module regularly interacts with the knowledge base to get inputs, and also update it.

3.Data Mining Techniques

Data mining includes the utilization of refined data analysis tools to find previously unknown, valid patterns and relationships in huge data sets. These tools can incorporate statistical models, machine learning techniques, and mathematical algorithms, such as neural networks or decision trees. Thus, data mining incorporates analysis and prediction.

Depending on various methods and technologies from the intersection of machine learning, database management, and statistics, professionals in data mining have devoted their careers to better understanding how to process and make conclusions from the huge amount of data, but what are the methods they use to make it happen?

In recent data mining projects, various major data mining techniques have been developed and used, including association, classification, clustering, prediction, sequential patterns, and regression.



Classification:

This technique is used to obtain important and relevant information about data and metadata. This data mining technique helps to classify data in different classes.

Data mining techniques can be classified by different criteria, as follows:

Classification of Data mining frameworks as per the type of data sources mined: This classification is as per the type of data handled. For example, multimedia, spatial data, text data, time-series data, World Wide Web, and so on..

Classification of data mining frameworks as per the database involved: This classification based on the data model involved. For example. Object-oriented database, transactional database, relational database, and so on..

Classification of data mining frameworks as per the kind of knowledge discovered: This classification depends on the types of knowledge discovered or data mining functionalities. For example, discrimination, classification, clustering, characterization, etc. some frameworks tend to be extensive frameworks offering a few data mining functionalities together..

Classification of data mining frameworks according to data mining techniques used: This classification is as per the data analysis approach utilized, such as neural networks, machine learning, genetic algorithms, visualization, statistics, data warehouse-oriented or database-oriented, etc.

The classification can also take into account, the level of user interaction involved in the data mining procedure, such as query-driven systems, autonomous systems, or interactive exploratory systems.

Clustering:

Clustering is a division of information into groups of connected objects. Describing the data by a few clusters mainly loses certain confine details, but accomplishes improvement. It models data by its clusters. Data modeling puts clustering from a historical point of view rooted in statistics, mathematics, and numerical analysis. From a machine learning point of view, clusters relate to hidden patterns, the search for clusters is unsupervised learning, and the subsequent framework represents a data concept. From a practical point of view, clustering plays an extraordinary job in data mining applications. For example, scientific

data exploration, text mining, information retrieval, spatial database applications, CRM, Web analysis, computational biology, medical diagnostics, and much more.

In other words, we can say that Clustering analysis is a data mining technique to identify similar data. This technique helps to recognize the differences and similarities between the data. Clustering is very similar to the classification, but it involves grouping chunks of data together based on their similarities.

Regression:

Regression analysis is the data mining process is used to identify and analyze the relationship between variables because of the presence of the other factor. It is used to define the probability of the specific variable. Regression, primarily a form of planning and modeling. For example, we might use it to project certain costs, depending on other factors such as availability, consumer demand, and competition. Primarily it gives the exact relationship between two or more variables in the given data set.

Association Rules:

This data mining technique helps to discover a link between two or more items. It finds a hidden pattern in the data set.

Association rules are if-then statements that support to show the probability of interactions between data items within large data sets in different types of databases. Association rule mining has several applications and is commonly used to help sales correlations in data or medical data sets.

The way the algorithm works is that you have various data, For example, a list of grocery items that you have been buying for the last six months. It calculates a percentage of items being purchased together.

These are three major measurements technique:

Lift:

This measurement technique measures the accuracy of the confidence over how often item B is purchased.

(Confidence) / (item B)/ (Entire dataset)

Support:

This measurement technique measures how often multiple items are purchased and

compared it to the overall dataset.

(Item A + Item B) / (Entire dataset)

Confidence:

This measurement technique measures how often item B is purchased when item A is purchased as well.

(Item A + Item B)/ (Item A)

Outer detection:

This type of data mining technique relates to the observation of data items in the data set, which do not match an expected pattern or expected behavior. This technique may be used in various domains like intrusion, detection, fraud detection, etc. It is also known as Outlier Analysis or Outilier mining. The outlier is a data point that diverges too much from the rest of the dataset. The majority of the real-world datasets have an outlier. Outlier detection plays a significant role in the data mining field. Outlier detection is valuable in numerous fields like network interruption identification, credit or debit card fraud detection, detecting outlying in wireless sensor network data, etc.

Sequential Patterns:

The sequential pattern is a data mining technique specialized for evaluating sequential data to discover sequential patterns. It comprises of finding interesting subsequences in a set of sequences, where the stake of a sequence can be measured in terms of different criteria like length, occurrence frequency, etc.

In other words, this technique of data mining helps to discover or recognize similar patterns in transaction data over some time.

Prediction:

Prediction used a combination of other data mining techniques such as trends, clustering, classification, etc. It analyzes past events or instances in the right sequence to predict a future event.

Data Mining Issues

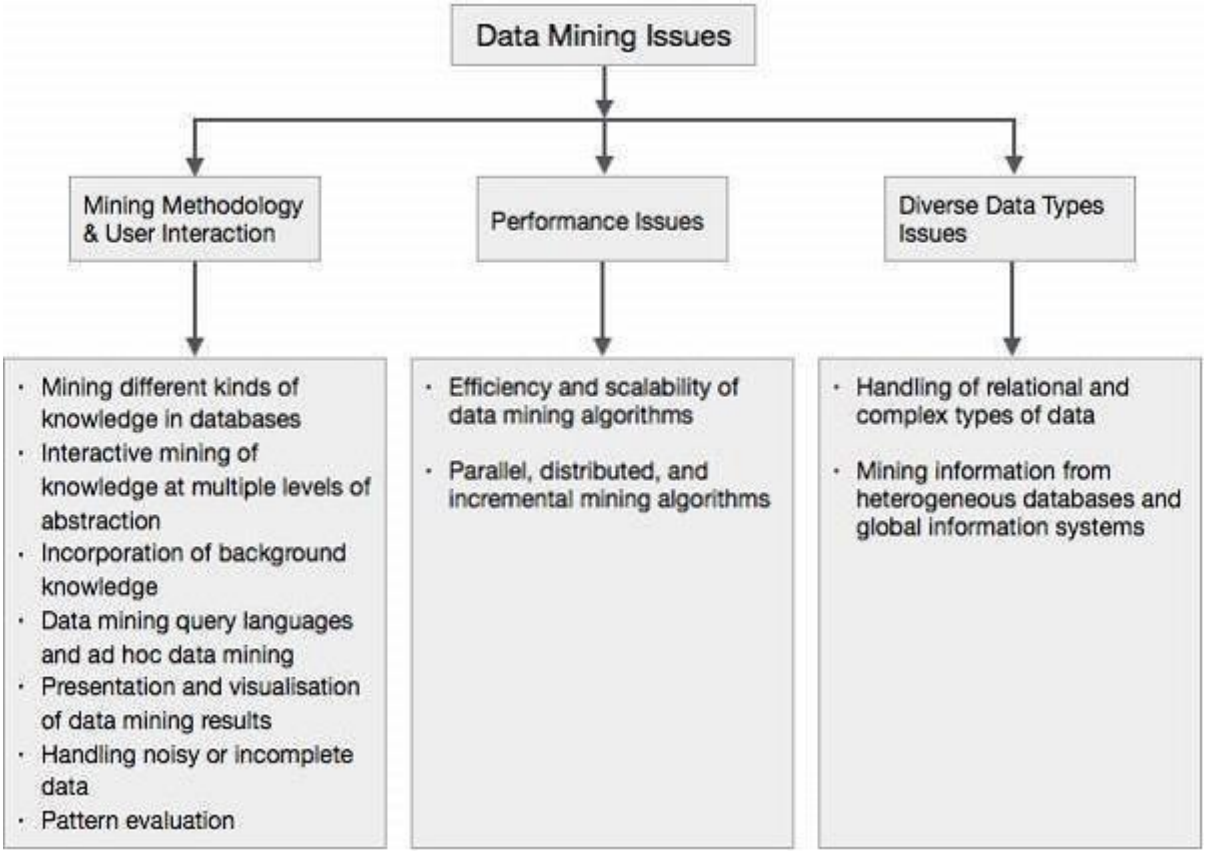
Data mining is not an easy task, as the algorithms used can get very complex and data is not always available at one place. It needs to be integrated from various heterogeneous data sources. These factors also create some issues. Here in this tutorial, we will discuss the major issues regarding −

Mining Methodology and User Interaction

Performance Issues

Diverse Data Types Issues

The following diagram describes the major issues.



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 is 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 Mining Applications

Data Mining is primarily used by organizations with intense consumer demands- Retail, Communication, Financial, marketing company, determine price, consumer preferences, product positioning, and impact on sales, customer satisfaction, and corporate profits. Data mining enables a retailer to use point-of-sale records of customer purchases to develop products and promotions that help the organization to attract the customer.

These are the following areas where data mining is widely used:

Data Mining in Healthcare:

Data mining in healthcare has excellent potential to improve the health system. It uses data and analytics for better insights and to identify best practices that will enhance health care services and reduce costs. Analysts use data mining approaches such as Machine learning, Multi-dimensional database, Data visualization, Soft computing, and statistics. Data Mining can be used to forecast patients in each category. The procedures ensure that the patients get intensive care at the right place and at the right time. Data mining also enables healthcare insurers to recognize fraud and abuse.

Data Mining in Market Basket Analysis:

Market basket analysis is a modeling method based on a hypothesis. If you buy a specific group of products, then you are more likely to buy another group of products. This technique may enable the retailer to understand the purchase behavior of a buyer. This data may assist the retailer in understanding the requirements of the buyer and altering the store's layout accordingly. Using a different analytical comparison of results between various stores, between customers in different demographic groups can be done.

Data mining in Education:

Education data mining is a newly emerging field, concerned with developing techniques that explore knowledge from the data generated from educational Environments. EDM objectives are recognized as affirming student's future learning behavior, studying the impact of educational support, and promoting learning science. An organization can use data mining to make precise decisions and also to predict the results of the student. With the results, the institution can concentrate on what to teach and how to teach.

Data Mining in Manufacturing Engineering:

Knowledge is the best asset possessed by a manufacturing company. Data mining tools can be beneficial to find patterns in a complex manufacturing process. Data mining can be used in system-level designing to obtain the relationships between product architecture, product portfolio, and data needs of the customers. It can also be used to forecast the product development period, cost, and expectations among the other tasks.

Data Mining in CRM (Customer Relationship Management):

Customer Relationship Management (CRM) is all about obtaining and holding Customers, also enhancing customer loyalty and implementing customer-oriented strategies. To get a decent relationship with the customer, a business organization needs to collect data and analyze the data. With data mining technologies, the collected data can be used for analytics.

Data Mining in Fraud detection:

Billions of dollars are lost to the action of frauds. Traditional methods of fraud detection are a little bit time consuming and sophisticated. Data mining provides meaningful patterns and turning data into information. An ideal fraud detection system should protect the data of all the users. Supervised methods consist of a collection of sample records, and these records are classified as fraudulent or non-fraudulent. A model is constructed using this data, and the technique is made to identify whether the document is fraudulent or not.

Data Mining in Lie Detection:

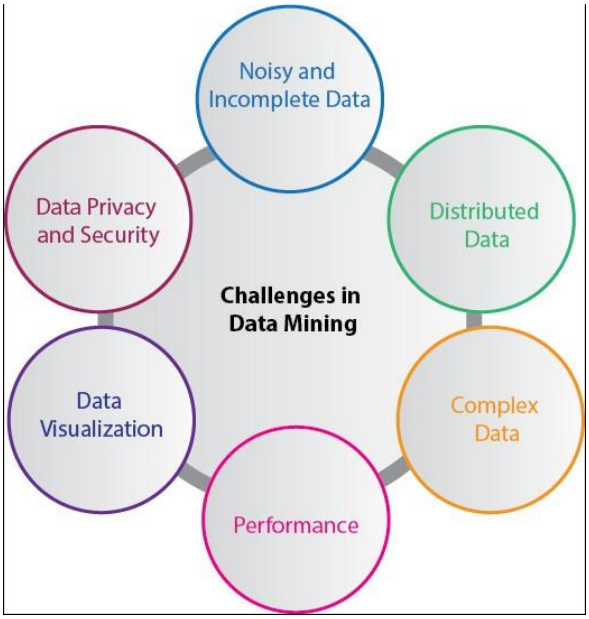
Apprehending a criminal is not a big deal, but bringing out the truth from him is a very challenging task. Law enforcement may use data mining techniques to investigate offenses, monitor suspected terrorist communications, etc. This technique includes text mining also, and it seeks meaningful patterns in data, which is usually unstructured text. The information collected from the previous investigations is compared, and a model for lie detection is constructed.

Data Mining Financial Banking:

The Digitalization of the banking system is supposed to generate an enormous amount of data with every new transaction. The data mining technique can help bankers by solving business-related problems in banking and finance by identifying trends, casualties, and correlations in business information and market costs that are not instantly evident to managers or executives because the data volume is too large or are produced too rapidly on the screen by experts. The manager may find these data for better targeting, acquiring, retaining, segmenting, and maintain a profitable customer.

Challenges of Implementation in Data mining

Although data mining is very powerful, it faces many challenges during its execution. Various challenges could be related to performance, data, methods, and techniques, etc. The process of data mining becomes effective when the challenges or problems are correctly recognized and adequately resolved.



Incomplete and noisy data:

The process of extracting useful data from large volumes of data is data mining. The data in the real-world is heterogeneous, incomplete, and noisy. Data in huge quantities will usually be inaccurate or unreliable. These problems may occur due to data measuring instrument or because of human errors. Suppose a retail chain collects phone numbers of customers who spend more than $ 500, and the accounting employees put the information into their system. The person may make a digit mistake when entering the phone number, which results in incorrect data. Even some customers may not be willing to disclose their phone numbers, which results in incomplete data. The data could get changed due to human or system error. All these consequences (noisy and incomplete data)makes data mining challenging.

Data Distribution:

Real-worlds data is usually stored on various platforms in a distributed computing environment. It might be in a database, individual systems, or even on the internet. Practically, It is a quite tough task to make all the data to a centralized data repository mainly due to organizational and technical concerns. For example, various regional offices may have their servers to store their data. It is not feasible to store, all the data from all the offices on a central server. Therefore, data mining requires the development of tools and algorithms that allow the mining of distributed data.

Complex Data:

Real-world data is heterogeneous, and it could be multimedia data, including audio and video, images, complex data, spatial data, time series, and so on. Managing these various types of data and extracting useful information is a tough task. Most of the time, new technologies, new tools, and methodologies would have to be refined to obtain specific information.

Performance:

The data mining system's performance relies primarily on the efficiency of algorithms and techniques used. If the designed algorithm and techniques are not up to the mark, then the efficiency of the data mining process will be affected adversely.

Data Privacy and Security:

Data mining usually leads to serious issues in terms of data security, governance, and privacy. For example, if a retailer analyzes the details of the purchased items, then it

reveals data about buying habits and preferences of the customers without their permission.

Data Visualization:

In data mining, data visualization is a very important process because it is the primary method that shows the output to the user in a presentable way. The extracted data should convey the exact meaning of what it intends to express. But many times, representing the information to the end-user in a precise and easy way is difficult. The input data and the output information being complicated, very efficient, and successful data visualization processes need to be implemented to make it successful.

Data Objects and Attribute Types

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

What Is an Attribute?

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

The type of an attribute is determined by the set of possible values—nominal, binary, ordinal, or numeric—the attribute can have. In the following subsections, we introduce each type.

Nominal Attributes

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

Example 2.1 Nominal attributes

Suppose that hair_color and marital_status are two attributes describing person objects. In our application, possible values for hair_color are black, brown, blond, red, auburn, gray, and white. The attribute marital_status can take on the values single, married, divorced, and widowed. Both hair_color and marital_status are nominal attributes. Another example of a nominal attribute is occupation, with the values teacher, dentist, programmer, farmer, and so on.

Although we said that the values of a nominal attribute are symbols or "names of things," it is possible to represent such symbols or "names" with numbers. With hair_color, for instance, we can assign a code of 0 for black, 1 for brown, and so on. Another example is customor_ID, with possible values that are all numeric. However, in such cases, the numbers are not intended to be used quantitatively. That is, mathematical operations on values of nominal attributes are not meaningful. It makes no sense to subtract one customer ID number from another, unlike, say, subtracting an age value from another (where age is a numeric attribute). Even though a nominal attribute may have integers as values, it is not considered a numeric attribute because the integers are not meant to be used quantitatively. We will say more on numeric attributes in Section 2.1.5.

Because nominal attribute values do not have any meaningful order about them and are not quantitative, it makes no sense to find the mean (average) value or median (middle) value for such an attribute, given a set of objects. One thing that is of interest, however, is the attribute's most commonly occurring value. This value, known as the mode, is one of the measures of central tendency. You will learn about measures of central tendency in Section 2.2.

Binary Attributes

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

Example 2.2 Binary attributes

Given the attribute smoker describing a patient object, 1 indicates that the patient smokes, while 0 indicates that the patient does not. Similarly, suppose the patient undergoes a medical test that has two possible outcomes. The attribute medical_test is binary, where a value of 1 means the result of the test for the patient is positive, while 0 means the result is negative.

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

A binary attribute is asymmetric if the outcomes of the states are not equally important, such

as the positive and negative outcomes of a medical test for HIV. By convention, we code the most important outcome, which is usually the rarest one, by 1 (e.g., HIV positive) and the other by 0 (e.g., HIV negative).

Ordinal Attributes

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

Example 2.3 Ordinal attributes

Suppose that drink_size corresponds to the size of drinks available at a fast-food restaurant. This nominal attribute has three possible values: small, medium, and large. The values have a meaningful sequence (which corresponds to increasing drink size); however, we cannot tell from the values how much bigger, say, a medium is than a large. Other examples of ordinal attributes include grade (e.g., A+, A, A−, B+, and so on) and professional_rank. Professional ranks can be enumerated in a sequential order: for example, assistant, associate, and full for professors, and private, private first class, specialist, corporal, and sergeant for army ranks.

Ordinal attributes are useful for registering subjective assessments of qualities that cannot be measured objectively; thus ordinal attributes are often used in surveys for ratings. In one survey, participants were asked to rate how satisfied they were as customers. Customer satisfaction had the following ordinal categories: 0: very dissatisfied, 1: somewhat dissatisfied, 2: neutral, 3: satisfied, and 4: very satisfied.

Ordinal attributes may also be obtained from the discretization of numeric quantities by splitting the value range into a finite number of ordered categories as described in Chapter 3 on data reduction.

The central tendency of an ordinal attribute can be represented by its mode and its median (the middle value in an ordered sequence), but the mean cannot be defined.

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

Numeric Attributes

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

Interval-Scaled Attributes

Interval-scaled attributes are measured on a scale of equal-size units. The values of interval-scaled attributes have order and can be positive, 0, or negative. Thus, in addition to providing a

ranking of values, such attributes allow us to compare and quantify the difference between values.

Example 2.4

Interval-scaled attributes

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

Temperatures in Celsius and Fahrenheit do not have a true zero-point, that is, neither 0°C nor 0°F indicates "no temperature." (On the Celsius scale, for example, the unit of measurement is 1/100 of the difference between the melting temperature and the boiling temperature of water in atmospheric pressure.) Although we can compute the difference between temperature values, we cannot talk of one temperature value as being a multiple of another. Without a true zero, we cannot say, for instance, that 10°C is twice as warm as 5°C. That is, we cannot speak of the values in terms of ratios. Similarly, there is no true zero-point for calendar dates. (The year 0 does not correspond to the beginning of time.) This brings us to ratio-scaled attributes, for which a true zero-point exits.

Because interval-scaled attributes are numeric, we can compute their mean value, in addition to the median and mode measures of central tendency.

Ratio-Scaled Attributes

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

Example 2.5

Ratio-scaled attributes

Unlike temperatures in Celsius and Fahrenheit, the Kelvin (K) temperature scale has what is considered a true zero-point (0°K = −273.15°C): It is the point at which the particles that comprise matter have zero kinetic energy. Other examples of ratio-scaled attributes include count attributes such as years_of_experience (e.g., the objects are employees) and number_of_words (e.g., the objects are documents). Additional examples include attributes to measure weight, height, and speed, and longitude and monetary quantities (e.g., you are 100 times richer with $100 than with
$1).

Discrete versus Continuous Attributes

In our presentation, we have organized attributes into nominal, binary, ordinal, and numeric types. There are many ways to organize attribute types. The types are not mutually exclusive.

Classification algorithms developed from the field of machine learning often talk of attributes as being either discrete or continuous. Each type may be processed differently. A discrete attribute has a finite or countably infinite set of values, which may or may not be represented as integers. The attributes hair_color, smoker, medical_test, and drink_size each have a finite number of

values, and so are discrete. Note that discrete attributes may have numeric values, such as 0 and 1 for binary attributes or, the values 0 to 110 for the attribute age. An attribute is countably infinite if the set of possible values is infinite but the values can be put in a one-to-one correspondence with natural numbers. For example, the attribute customer_ID is countably infinite. The number of customers can grow to infinity, but in reality, the actual set of values is countable (where the values can be put in one-to-one correspondence with the set of integers). Zip codes are another example.

If an attribute is not discrete, it is continuous. The terms numeric attribute and continuous attribute are often used interchangeably in the literature. (This can be confusing because, in the classic sense, continuous values are real numbers, whereas numeric values can be either integers or real numbers.) In practice, real values are represented using a finite number of digits. Continuous attributes are typically represented as floating-point variables.

Statistical Methods in Data Mining

Data mining refers to extracting or mining knowledge from large amounts of data. In other words, data mining is the science, art, and technology of discovering large and complex bodies of data in order to discover useful patterns. Theoreticians and practitioners are continually seeking improved techniques to make the process more efficient, cost-effective, and accurate. Any situation can be analyzed in two ways in data mining:

Statistical Analysis: In statistics, data is collected, analyzed, explored, and presented to identify patterns and trends. Alternatively, it is referred to as quantitative analysis.

Non-statistical Analysis: This analysis provides generalized information and includes sound, still images, and moving images.

In statistics, there are two main categories:

Descriptive Statistics: The purpose of descriptive statistics is to organize data and identify the main characteristics of that data. Graphs or numbers summarize the data. Average, Mode, SD(Standard Deviation), and Correlation are some of the commonly used descriptive statistical methods.

Inferential Statistics: The process of drawing conclusions based on probability theory and generalizing the data. By analyzing sample statistics, you can infer parameters about populations and make models of relationships within data.

There are various statistical terms that one should be aware of while dealing with statistics. Some of these are:

Population

Sample

Variable

Quantitative Variable

Qualitative Variable

Discrete Variable

Continuous Variable

Now, let's start discussing statistical methods. This is the analysis of raw data using mathematical formulas, models, and techniques. Through the use of statistical methods, information is extracted from research data, and different ways are available to judge the robustness of research outputs.

As a matter of fact, today's statistical methods used in the data mining field typically are derived from the vast statistical toolkit developed to answer problems arising in other fields. These techniques are taught in science curriculums. It is necessary to check and test several hypotheses. The hypotheses described above help us assess the validity of our data mining endeavor when attempting to infer any inferences from the data under study. When using more complex and sophisticated statistical estimators and tests, these issues become more pronounced.

For extracting knowledge from databases containing different types of observations, a variety of statistical methods are available in Data Mining and some of these are:

Logistic regression analysis

Correlation analysis

Regression analysis

Discriminate analysis

Linear discriminant analysis (LDA)

Classification

Clustering

Outlier detection

Classification and regression trees,

Correspondence analysis

Nonparametric regression,

Statistical pattern recognition,

Categorical data analysis,

Time-series methods for trends and periodicity

Artificial neural networks

Now, let's try to understand some of the important statistical methods which are used in data mining:

Linear Regression: The linear regression method uses the best linear relationship between the independent and dependent variables to predict the target variable. In order to achieve the best fit, make sure that all the distances between the shape and the actual observations at each point are as small as possible. A good fit can be determined by determining that no other position

would produce fewer errors given the shape chosen. Simple linear regression and multiple linear regression are the two major types of linear regression. By fitting a linear relationship to the independent variable, the simple linear regression predicts the dependent variable. Using multiple independent variables, multiple linear regression fits the best linear relationship with the dependent variable. For more details, you can refer linear regression.

Classification: This is a method of data mining in which a collection of data is categorized so that a greater degree of accuracy can be predicted and analyzed. An effective way to analyze very large datasets is to classify them. Classification is one of several methods aimed at improving the efficiency of the analysis process. A Logistic Regression and a Discriminant Analysis stand out as two major classification techniques.

Logistic Regression: It can also be applied to machine learning applications and predictive analytics. In this approach, the dependent variable is either binary (binary regression) or multinomial (multinomial regression): either one of the two or a set of one, two, three, or four options. With a logistic regression equation, one can estimate probabilities regarding the relationship between the independent variable and the dependent variable. For understanding logistic regression analysis in detail, you can refer to logistic regression.

Discriminant Analysis: A Discriminant Analysis is a statistical method of analyzing data based on the measurements of categories or clusters and categorizing new observations into one or more populations that were identified a priori. The discriminant analysis models each response class independently then uses Bayes's theorem to flip these projections around to estimate the likelihood of each response category given the value of X. These models can be either linear or quadratic.

Linear Discriminant Analysis: According to Linear Discriminant Analysis, each observation is assigned a discriminant score to classify it into a response variable class. By combining the independent variables in a linear fashion, these scores can be obtained. Based on this model, observations are drawn from a Gaussian distribution, and the predictor variables are correlated across all k levels of the response variable, Y. and for further details linear discriminant analysis

Quadratic Discriminant Analysis: An alternative approach is provided by Quadratic Discriminant Analysis. LDA and QDA both assume Gaussian distributions for the observations of the Y classes. Unlike LDA, QDA considers each class to have its own covariance matrix. As a result, the predictor variables have different variances across the k levels in Y.

Correlation Analysis: In statistical terms, correlation analysis captures the relationship between variables in a pair. The value of such variables is usually stored in a column or rows of a database table and represents a property of an object.

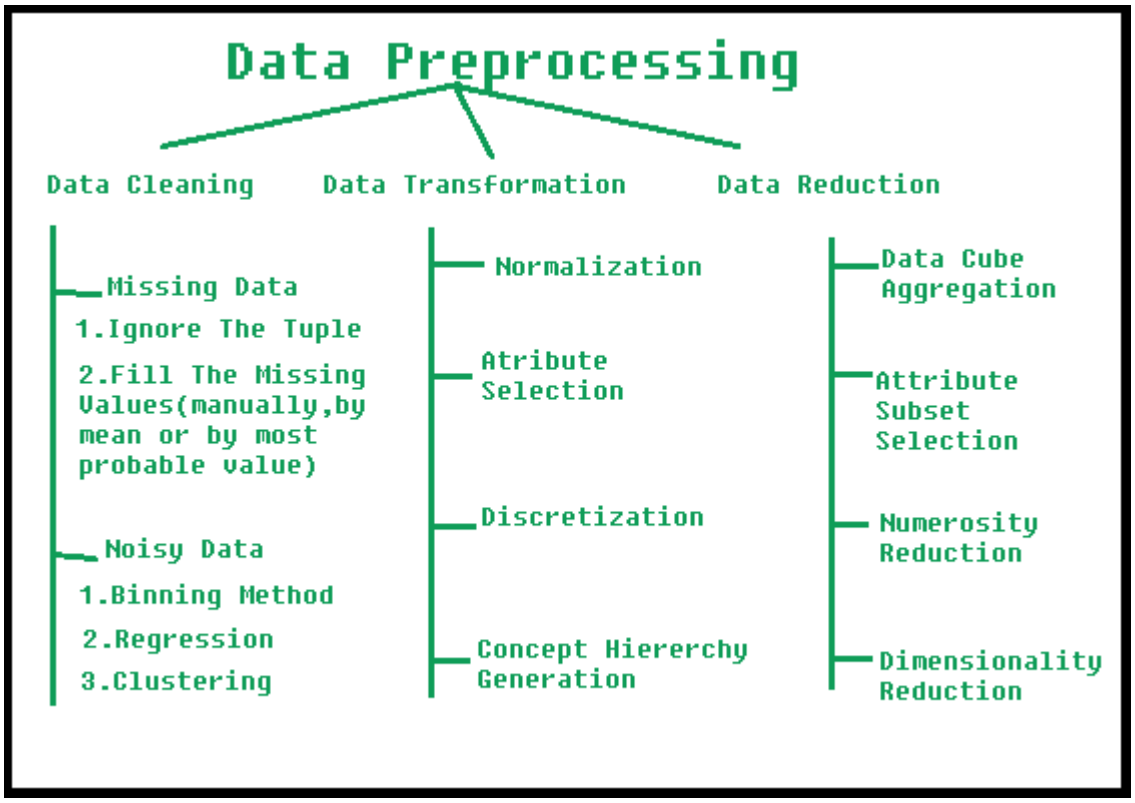
Regression Analysis: Based on a set of numeric data, regression is a data mining method that predicts a range of numerical values (also known as continuous values). You could, for instance, use regression to predict the cost of goods and services based on other variables. A regression model is used across numerous industries for forecasting financial data, modeling environmental conditions, and analyzing trends.

The first step in creating good statistics is having good data that was derived with an aim in mind. There are two main types of data: an input (independent or predictor) variable, which we control or are able to measure, and an output (dependent or response) variable which is observed. A few will be quantitative measurements, but others may be qualitative or categorical variables (called factors).

Data Preprocessing

Preprocessing in Data Mining:

Data preprocessing is a data mining technique which is used to transform the raw data in a useful and efficient format.



Steps Involved in Data Preprocessing:

Data Cleaning:

The data can have many irrelevant and missing parts. To handle this part, data cleaning is done. It involves handling of missing data, noisy data etc.

Missing Data:

This situation arises when some data is missing in the data. It can be handled in various ways.

Some of them are:

Ignore the tuples:

This approach is suitable only when the dataset we have is quite large and multiple values are missing within a tuple.

Fill the Missing values:

There are various ways to do this task. You can choose to fill the missing values manually, by attribute mean or the most probable value.

Noisy Data:

Noisy data is a meaningless data that can’t be interpreted by machines.It can be generated due to faulty data collection, data entry errors etc. It can be handled in following ways :

Binning Method:

This method works on sorted data in order to smooth it. The whole data is divided into segments of equal size and then various methods are performed to complete the task. Each segmented is handled separately. One can replace all data in a segment by its mean or boundary values can be used to complete the task.

Regression:

Here data can be made smooth by fitting it to a regression function.The regression used may be linear (having one independent variable) or multiple (having multiple independent variables).

Clustering:

This approach groups the similar data in a cluster. The outliers may be undetected or it will fall outside the clusters.

Data Transformation:

This step is taken in order to transform the data in appropriate forms suitable for mining process. This involves following ways:

Normalization:

It is done in order to scale the data values in a specified range (-1.0 to 1.0 or 0.0 to 1.0)

Attribute Selection:

In this strategy, new attributes are constructed from the given set of attributes to help the mining process.

Discretization:

This is done to replace the raw values of numeric attribute by interval levels or conceptual levels.

Concept Hierarchy Generation:

Here attributes are converted from lower level to higher level in hierarchy. For Example-The attribute “city” can be converted to “country”.

Data Reduction:

Since data mining is a technique that is used to handle huge amount of data. While working with huge volume of data, analysis became harder in such cases. In order to get rid of this, we uses data reduction technique. It aims to increase the storage efficiency and reduce data storage and analysis costs.

The various steps to data reduction are:

Data Cube Aggregation:

Aggregation operation is applied to data for the construction of the data cube.

Attribute Subset Selection:

The highly relevant attributes should be used, rest all can be discarded. For performing attribute selection, one can use level of significance and p- value of the attribute.the attribute having p-value greater than significance level can be discarded.

Numerosity Reduction:

This enable to store the model of data instead of whole data, for example: Regression Models.

Dimensionality Reduction:

This reduce the size of data by encoding mechanisms.It can be lossy or lossless. If after reconstruction from compressed data, original data can be retrieved, such reduction are called lossless reduction else it is called lossy reduction. The two effective methods of dimensionality reduction are:Wavelet transforms and PCA (Principal Component Analysis).

9.Data Integration

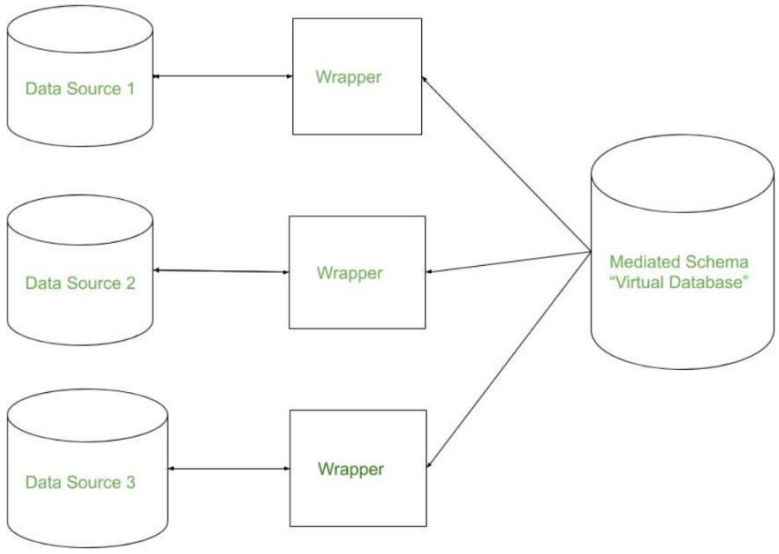
Data Integration is a data preprocessing technique that involves combining data from multiple heterogeneous data sources into a coherent data store and provide a unified view of the data. These sources may include multiple data cubes, databases, or flat files.

The data integration approaches are formally defined as triple <G, S, M> where,

G stand for the global schema,

S stands for the heterogeneous source of schema,

M stands for mapping between the queries of source and global schema.



There are mainly 2 major approaches for data integration – one is the "tight coupling approach" and another is the "loose coupling approach".

Tight Coupling:

Here, a data warehouse is treated as an information retrieval component.

In this coupling, data is combined from different sources into a single physical location through the process of ETL – Extraction, Transformation, and Loading.

Loose Coupling:

Here, an interface is provided that takes the query from the user, transforms it in a way the source database can understand, and then sends the query directly to the source databases to obtain the result.

And the data only remains in the actual source databases.

Issues in Data Integration:

There are no issues to consider during data integration: Schema Integration, Redundancy, Detection, and resolution of data value conflicts. These are explained in brief below.

Schema Integration:

Integrate metadata from different sources.

The real-world entities from multiple sources are matched referred to as the entity identification problem.

Redundancy:

An attribute may be redundant if it can be derived or obtaining from another attribute or set of attributes.

Inconsistencies in attributes can also cause redundancies in the resulting data set.

Some redundancies can be detected by correlation analysis.

Detection and resolution of data value conflicts:

This is the third important issue in data integration.

Attribute values from different sources may differ for the same real- world entity.

An attribute in one system may be recorded at a lower level abstraction than the “same” attribute in another.

9.Data Cleaning

Data cleaning routines attempt to fill in missing values, smooth out noise while identifying outliers, and correct inconsistencies in the data.

Missing values

Ignore the tuple: This is usually done when the class label is missing (assuming the mining task involves classification or description). This method is not very effective, unless the tuple contains several attributes with missing values. It is especially poor when the percentage of missing values per attribute varies considerably.

Fill in the missing value manually: In general, this approach is timeconsuming and may not be feasible given a large data set with many missing values.

Use a global constant to fill in the missing value: Replace all missing attribute values by the same constant, such as a label like ―Unknown". If missing values are replaced by, say, ―Unknown", then the mining program may mistakenly think that they form an interesting concept, since they all have a value in common - that of ―Unknown". Hence, although this method is simple, it is not recommended.

Use the attribute mean to fill in the missing value: For example, suppose that the average income of All Electronics customers is $28,000. Use this value to replace the missing value for income.

Use the attribute mean for all samples belonging to the same class as the given tuple: For example, if classifying customers according to credit risk, replace the missing value with the average income value for customers in the same credit risk category as that of the given tuple.

Use the most probable value to fill in the missing value: This may be determined with inference-based tools using a Bayesian formalism or decision tree induction. For example, using the other customer attributes in your data set, you may construct a decision tree to predict the missing values for income.

Noisy data

Noise is a random error or variance in a measured variable.

Binning methods: Binning methods smooth a sorted data value by consulting the ‖neighborhood", or values around it. The sorted values are distributed into a number of 'buckets', or bins. Because binning methods consult the neighborhood of values, they perform local smoothing. Figure illustrates some binning techniques. In this example, the data for price are first sorted and partitioned into equi-depth bins (of depth 3). In smoothing by bin means, each value in a bin is replaced by the mean value of the bin. For example, the mean of the values 4, 8, and 15 in Bin 1 is 9. Therefore, each original value in this bin is replaced by the value 9. Similarly, smoothing by bin medians can be employed, in which each bin value is replaced by the bin median. In smoothing by bin boundaries, the minimum and maximum values in a given bin are identified as the bin boundaries. Each bin value is then replaced by the closest boundary value.

(i).Sorted data for price (in dollars): 4, 8, 15, 21, 21, 24, 25, 28, 34

(ii).Partition into (equi-width) bins:

Bin 1: 4, 8, 15

Bin 2: 21, 21, 24

Bin 3: 25, 28, 34

(iii).Smoothing by bin means:

Bin 1: 9, 9, 9

, Bin 2: 22, 22, 22

Bin 3: 29, 29, 29

(iv).Smoothing by bin boundaries:

Bin 1: 4, 4, 15

Bin 2: 21, 21, 24

Bin 3: 25, 25, 34

Clustering: Outliers may be detected by clustering, where similar values are organized into groups or ―clusters‖. Intuitively, values which fall outside of the set of clusters may be considered outliers. Figure: Outliers may be detected by clustering analysis.

Combined computer and human inspection: Outliers may be identified through a combination of computer and human inspection. In one application, for example, an information-theoretic measure was used to help identify outlier patterns in a handwritten

character database for classification. The measure's value reflected the ―surprise" content of the predicted character label with respect to the known label. Outlier patterns may be informative or ―garbage". Patterns whose surprise content is above a threshold are output to a list. A human can then sort through the patterns in the list to identify the actual garbage ones

Regression: Data can be smoothed by fitting the data to a function, such as with regression. Page 46 Linear regression involves finding the ―best" line to fit two variables, so that one variable can be used to predict the other. Multiple linear regression is an extension of linear regression, where more than two variables are involved and the data are fit to a multidimensional surface.

Inconsistent data There may be inconsistencies in the data recorded for some transactions. Some data inconsistencies may be corrected manually using external references. For example, errors made at data entry may be corrected by performing a paper trace. This may be coupled with routines designed to help correct the inconsistent use of codes. Knowledge engineering tools may also be used to detect the violation of known data constraints. For example, known functional dependencies between attributes can be used to find values contradicting the functional constraints

9.Data Transformation

The data are transformed in ways that are ideal for mining the data. The data transformation involves steps that are:

Smoothing:

It is a process that is used to remove noise from the dataset using some algorithms It allows for highlighting important features present in the dataset. It helps in predicting the patterns. When collecting data, it can be manipulated to eliminate or reduce any variance or any other noise form.

The concept behind data smoothing is that it will be able to identify simple changes to help predict different trends and patterns. This serves as a help to analysts or traders who need to look at a lot of data which can often be difficult to digest for finding patterns that they wouldn’t see otherwise.

Aggregation:

Data collection or aggregation is the method of storing and presenting data in a summary format. The data may be obtained from multiple data sources to integrate these data sources into a data analysis description. This is a crucial step since the accuracy of data analysis insights is highly dependent on the quantity and quality of the data used. Gathering accurate data of high quality and a large enough quantity is necessary to produce relevant results.

The collection of data is useful for everything from decisions concerning financing or business strategy of the product, pricing, operations, and marketing strategies.

For example, Sales, data may be aggregated to compute monthly& annual total amounts.

Discretization:

It is a process of transforming continuous data into set of small intervals. Most Data Mining activities in the real world require continuous attributes. Yet many of the existing data mining frameworks are unable to handle these attributes. Also, even if a data mining task can manage a continuous attribute, it can significantly improve its efficiency by replacing a constant quality attribute with its discrete values.

For example, (1-10, 11-20) (age:- young, middle age, senior).

Attribute Construction:

Where new attributes are created & applied to assist the mining process from the given set of attributes. This simplifies the original data & makes the mining more efficient.

Generalization:

It converts low-level data attributes to high-level data attributes using concept hierarchy. For Example Age initially in Numerical form (22, 25) is converted into categorical value (young, old).

For example, Categorical attributes, such as house addresses, may be generalized to higher-level definitions, such as town or country.

Normalization: Data normalization involves converting all data variable into a given range.

Techniques that are used for normalization are:

Min-Max Normalization:

This transforms the original data linearly.

Suppose that: min_A is the minima and max_A is the maxima of an attribute, P

We Have the Formula:

Where v is the value you want to plot in the new range.

v’ is the new value you get after normalizing the old value. Solved example:

Suppose the minimum and maximum value for an attribute profit(P) are Rs. 10, 000 and Rs. 100, 000. We want to plot the profit in the range [0, 1]. Using min-max normalization the value of Rs. 20, 000 for attribute profit can be plotted to:

$$\frac{20000 - 10000}{100000 - 10000}(1 - 0) + 0 = 0.11$$

And hence, we get the value of v' as 0.11

Z-Score Normalization:

In z-score normalization (or zero-mean normalization) the values of an attribute (A), are normalized based on the mean of A and its standard deviation

A value, v, of attribute A is normalized to v' by computing

For example:

Let mean of an attribute P = 60, 000, Standard Deviation = 10, 000, for the attribute P. Using z-score normalization, a value of 85000 for P can be transformed to:

$$\frac{85000 - 60000}{10000} = 2.50$$

And hence we get the value of v' to be 2.5

Decimal Scaling:

It normalizes the values of an attribute by changing the position of their decimal points

The number of points by which the decimal point is moved can be determined by the absolute maximum value of attribute A.

A value, v, of attribute A is normalized to v' by computing

$$v' = \frac{v}{10^j}$$

where j is the smallest integer such that Max(|v'|) < 1. For example:

Suppose: Values of an attribute P varies from -99 to 99.

The maximum absolute value of P is 99.

For normalizing the values we divide the numbers by 100 (i.e., j = 2) or (number of integers in the largest number) so that values come out to be as 0.98, 0.97 and so on.

10.Data Reduction

The method of data reduction may achieve a condensed description of the original data which is much smaller in quantity but keeps the quality of the original data.

Methods of data reduction:

These are explained as following below.

Data Cube Aggregation:

This technique is used to aggregate data in a simpler form. For example, imagine that information you gathered for your analysis for the years 2012 to 2014, that data includes the revenue of your company every three months.

They involve you in the annual sales, rather than the quarterly average, So we can summarize the data in such a way that the resulting data summarizes the total sales per year instead of per quarter. It summarizes the data.

Dimension reduction:

Whenever we come across any data which is weakly important, then we use the attribute required for our analysis. It reduces data size as it eliminates outdated or redundant features.

Step-wise Forward Selection –

The selection begins with an empty set of attributes later on we decide best of the original attributes on the set based on their relevance to other attributes. We know it as a p-value in statistics.

Suppose there are the following attributes in the data set in which few attributes are redundant.

Initial attribute Set: {X1, X2, X3, X4, X5, X6} Initial reduced attribute set: { }

Step-1: {X1}

Step-2: {X1, X2}

Step-3: {X1, X2, X5}

Final reduced attribute set: {X1, X2, X5}

Step-wise Backward Selection –

This selection starts with a set of complete attributes in the original data and at each point, it eliminates the worst remaining attribute in the set.

Suppose there are the following attributes in the data set in which few attributes are redundant.

Initial attribute Set: {X1, X2, X3, X4, X5, X6}
Initial reduced attribute set: {X1, X2, X3, X4, X5, X6 }

Step-1: {X1, X2, X3, X4, X5}
Step-2: {X1, X2, X3, X5}
Step-3: {X1, X2, X5}

Final reduced attribute set: {X1, X2, X5}

Combination of forwarding and Backward Selection –

It allows us to remove the worst and select best attributes, saving time and making the process faster.

Data Compression:

The data compression technique reduces the size of the files using different encoding mechanisms (Huffman Encoding & run-length Encoding). We can divide it into two types based on their compression techniques.

Lossless Compression –

Encoding techniques (Run Length Encoding) allows a simple and minimal data size reduction. Lossless data compression uses algorithms to restore the precise original data from the compressed data.

Lossy Compression –

Methods such as Discrete Wavelet transform technique, PCA (principal component analysis) are examples of this compression. For e.g., JPEG image format is a lossy compression, but we can find the meaning equivalent to the original the image. In lossy-data compression, the decompressed data may differ to the original data but are useful enough to retrieve information from them.

Numerosity Reduction:

In this reduction technique the actual data is replaced with mathematical models or smaller representation of the data instead of actual data, it is important to only store the model parameter. Or non-parametric method such as clustering, histogram, sampling. For More Information on Numerosity Reduction Visit the link below:

Discretization & Concept Hierarchy Operation:

Techniques of data discretization are used to divide the attributes of the continuous nature into data with intervals. We replace many constant values of the attributes by labels of small intervals. This means that mining results are shown in a concise, and easily understandable way.

Top-down discretization –

If you first consider one or a couple of points (so-called breakpoints or split points) to divide the whole set of attributes and repeat of this method up to the end, then the process is known as top-down discretization also known as splitting.

Bottom-up discretization –

If you first consider all the constant values as split-points, some are discarded through a combination of the neighbourhood values in the interval, that process is called bottom-up discretization.

Concept Hierarchies:

It reduces the data size by collecting and then replacing the low-level concepts (such as 43 for age) to high-level concepts (categorical variables such as middle age or Senior).

For numeric data following techniques can be followed:

Binning –

Binning is the process of changing numerical variables into categorical counterparts. The number of categorical counterparts depends on the number of bins specified by the user.

Histogram analysis –

Like the process of binning, the histogram is used to partition the value for the attribute X, into disjoint ranges called brackets. There are several partitioning rules:

Equal Frequency partitioning: Partitioning the values based on their number of occurrences in the data set.

Equal Width Partitioning: Partitioning the values in a fixed gap based on the number of bins i.e. a set of values ranging from 0-20.

Clustering: Grouping the similar data together.

Data Discretization

Data discretization refers to a method of converting a huge number of data values into smaller ones so that the evaluation and management of data become easy. In other words, data discretization is a method of converting attributes values of continuous data into a finite set of intervals with minimum data loss. There are two forms of data discretization first is supervised discretization, and the second is unsupervised discretization. Supervised discretization refers to a method in which the class data is used. Unsupervised discretization refers to a method depending upon the way which operation

proceeds. It means it works on the top-down splitting strategy and bottom-up merging strategy. Now, we can understand this concept with the help of an example Suppose we have an attribute of Age with the given values

	1,5,9,4,7,11,14,17,13,18, 19,31,33,36,42,44,46,70,74,78,77

Table before Discretization

Competitive questions on Structures

in

Hindi

Keep Watching

Attribute		Age	Age
		11, 14, 17, 13, 18, 19	31, 33, 36, 42, 44, 46
After Disc		Young	Mature

r e t i z a t i o n				

Another example is analytics, where we gather the static data of website visitors. For example, all visitors who visit the site with the IP address of India are shown under country level.

Some Famous techniques of data discretization

Histogram analysis

Histogram refers to a plot used to represent the underlying frequency distribution of a continuous data set. Histogram assists the data inspection for data distribution. For example, Outliers, skewness representation, normal distribution representation, etc.

Binning

Binning refers to a data smoothing technique that helps to group a huge number of continuous values into smaller values. For data discretization and the development of idea hierarchy, this technique can also be used.

Cluster Analysis

Cluster analysis is a form of data discretization. A clustering algorithm is executed by dividing the values of x numbers into clusters to isolate a computational feature of x.

Data discretization using decision tree analysis

Data discretization refers to a decision tree analysis in which a top-down slicing technique is used. It is done through a supervised procedure. In a numeric attribute discretization, first, you need to select the attribute that has the least entropy, and then you need to run it with the help of a recursive process. The recursive process divides it into various discretized disjoint intervals, from top to bottom, using the same splitting criterion.

Data discretization using correlation analysis

Discretizing data by linear regression technique, you can get the best neighboring interval, and then the large intervals are combined to develop a larger overlap to form the final 20 overlapping intervals. It is a supervised procedure.

Data discretization and concept hierarchy generation

The term hierarchy represents an organizational structure or mapping in which items are ranked according to their levels of importance. In other words, we can say that a hierarchy concept refers to a sequence of mappings with a set of more general concepts to complex concepts. It means mapping is done from low-level concepts to high-level concepts. For example, in computer science, there are different types of hierarchical systems. A document is placed in a folder in windows at a specific place in the tree structure is the best example of a computer hierarchical tree model. There are two types of hierarchy: top-down mapping and the second one is bottom-up mapping.

Let's understand this concept hierarchy for the dimension location with the help of an example.

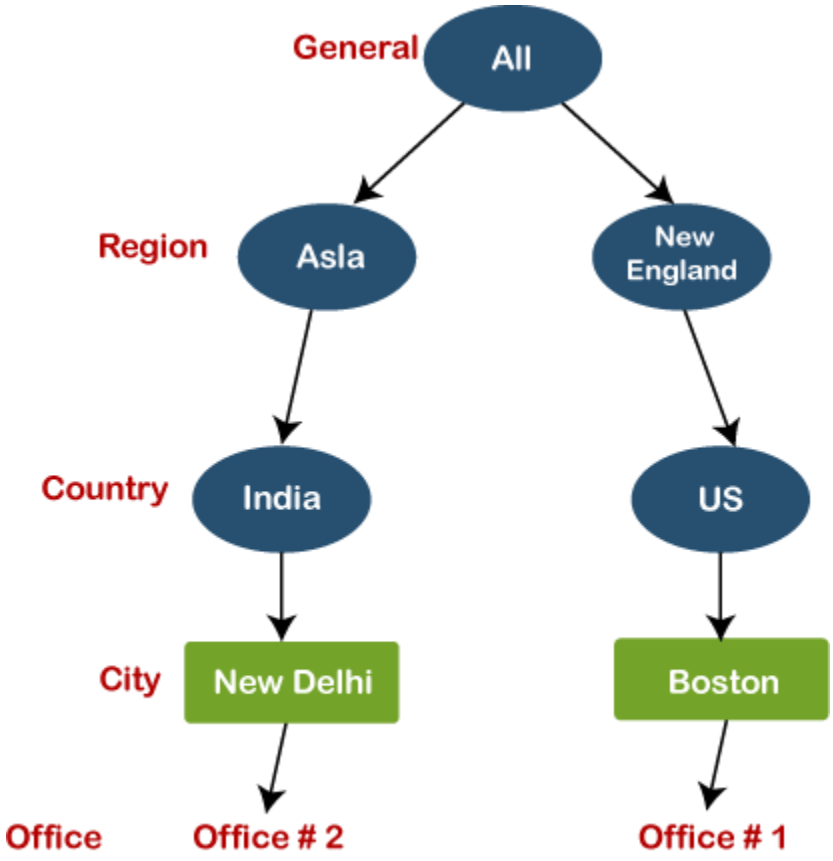
A particular city can map with the belonging country. For example, New Delhi can be mapped to India, and India can be mapped to Asia.

Top-down mapping

Top-down mapping generally starts with the top with some general information and ends with the bottom to the specialized information.

Bottom-up mapping

Bottom-up mapping generally starts with the bottom with some specialized information and ends with the top to the generalized information.



Concept Hierarchy Generation

Data discretization and binarization in data mining

Data discretization is a method of converting attributes values of continuous data into a finite set of intervals with minimum data loss. In contrast, data binarization is used to transform the continuous and discrete attributes into binary attributes.

Why is Discretization important?

As we know, an infinite of degrees of freedom mathematical problem poses with the continuous data. For many purposes, data scientists need the implementation of discretization. It is also used to improve signal noise ratio.

Data Visualization

Data visualization is a graphical representation of quantitative information and data by using visual elements like graphs, charts, and maps.

Data visualization convert large and small data sets into visuals, which is easy to understand and process for humans.

Data visualization tools provide accessible ways to understand outliers, patterns, and trends in the data.

In the world of Big Data, the data visualization tools and technologies are required to analyze vast amounts of information.

Data visualizations are common in your everyday life, but they always appear in the form of graphs and charts. The combination of multiple visualizations and bits of information are still referred to as Infographics.

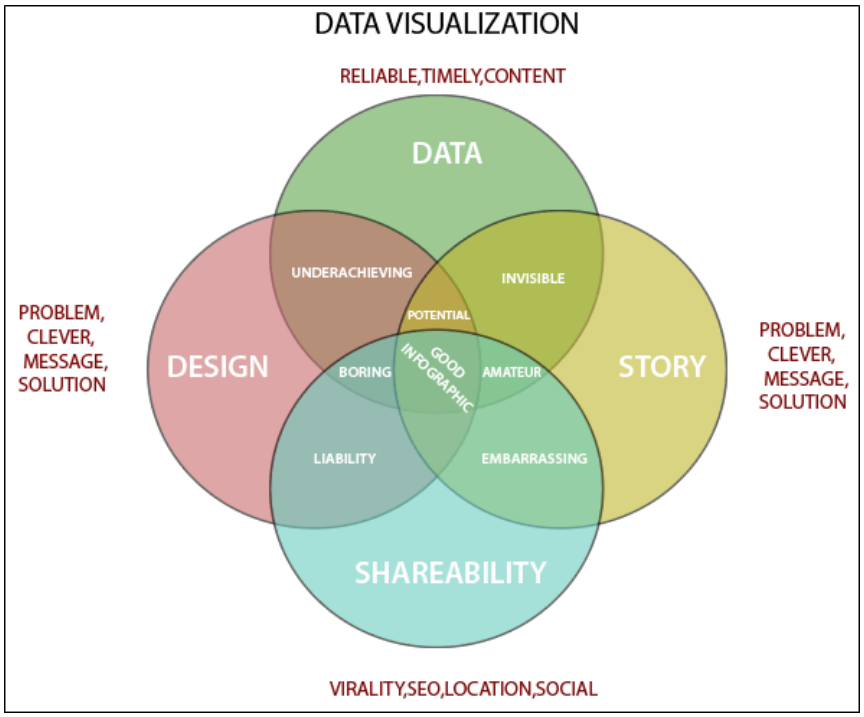
Data visualizations are used to discover unknown facts and trends. You can see visualizations in the form of line charts to display change over time. Bar and column charts are useful for observing relationships and making comparisons. A pie chart is a great way to show parts-of-a-whole. And maps are the best way to share geographical data visually.

Today's data visualization tools go beyond the charts and graphs used in the Microsoft Excel spreadsheet, which displays the data in more sophisticated ways such as dials and gauges, geographic maps, heat maps, pie chart, and fever chart.

What makes Data Visualization Effective?

Effective data visualization are created by communication, data science, and design collide. Data visualizations did right key insights into complicated data sets into meaningful and natural.

American statistician and Yale professor Edward Tufte believe useful data visualizations consist of ?complex ideas communicated with clarity, precision, and efficiency.



To craft an effective data visualization, you need to start with clean data that is well- sourced and complete. After the data is ready to visualize, you need to pick the right chart.

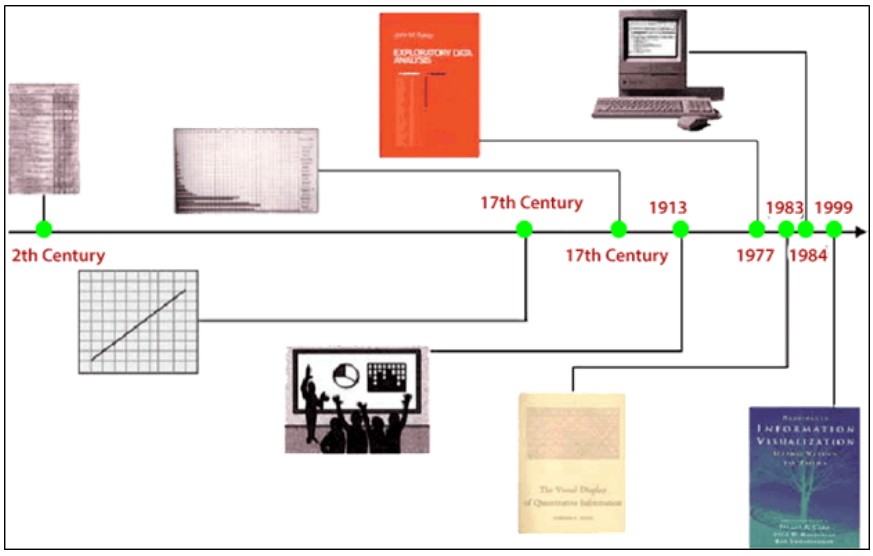
After you have decided the chart type, you need to design and customize your visualization to your liking. Simplicity is essential - you don't want to add any elements that distract from the data.

History of Data Visualization

The concept of using picture was launched in the 17th century to understand the data from the maps and graphs, and then in the early 1800s, it was reinvented to the pie chart.

Several decades later, one of the most advanced examples of statistical graphics occurred when Charles Minard mapped Napoleon's invasion of Russia. The map represents the size of the army and the path of Napoleon's retreat from Moscow - and that information tied to temperature and time scales for a more in-depth understanding of the event.

Computers made it possible to process a large amount of data at lightning-fast speeds. Nowadays, data visualization becomes a fast-evolving blend of art and science that certain to change the corporate landscape over the next few years.



Importance of Data Visualization

Data visualization is important because of the processing of information in human brains. Using graphs and charts to visualize a large amount of the complex data sets is more comfortable in comparison to studying the spreadsheet and reports.

Data visualization is an easy and quick way to convey concepts universally. You can experiment with a different outline by making a slight adjustment.

Data visualization have some more specialties such as:

Data visualization can identify areas that need improvement or modifications.

Data visualization can clarify which factor influence customer behavior.

Data visualization helps you to understand which products to place where.

Data visualization can predict sales volumes.

Data visualization tools have been necessary for democratizing data, analytics, and making data-driven perception available to workers throughout an organization. They are easy to operate in comparison to earlier versions of BI software or traditional statistical analysis software. This guide to a rise in lines of business implementing data visualization tools on their own, without support from IT.

Why Use Data Visualization?

To make easier in understand and remember.

To discover unknown facts, outliers, and trends.

To visualize relationships and patterns quickly.

To ask a better question and make better decisions.

To competitive analyze.

To improve insights.

Data Similarity and dissimilarity measures

Distance or similarity measures are essential in solving many pattern recognition problems such as classification and clustering. Various distance/similarity measures are available in the literature to compare two data distributions. As the names suggest, a similarity measures how close two distributions are. For multivariate data complex summary methods are developed to answer this question.

Similarity Measure

Numerical measure of how alike two data objects often fall between 0 (no similarity) and 1 (complete similarity)

Dissimilarity Measure

Numerical measure of how different two data objects are range from 0 (objects are alike) to ∞ (objects are different)

Proximity

refers to a similarity or dissimilarity Similarity/Dissimilarity for Simple Attributes

Here, p and q are the attribute values for two data objects.

Similarity	Dissimilarity
s={1 if p=q0 if p≠q	d={0 i

	$f \quad p = q \quad 1 \quad if \quad p \neq q \quad d = \|p - q\|_n - 1$
$s = 1 - \|p - q\|_n - 1$ (values mapped to integer 0 to n-1, where n is the number of values)	
$s = 1 - \|p - q\|, s = 11 + \|p - q\|$	$d = \|p - q\|$

Distance, such as the Euclidean distance, is a dissimilarity measure and has some well-known properties: Common Properties of Dissimilarity Measures

$d(p, q) \geq 0$ for all p and q, and $d(p, q) = 0$ if and only if $p = q$,
$d(p, q) = d(q,p)$ for all p and q,

$d(p, r) \leq d(p, q) + d(q, r)$ for all p, q, and r, where d(p, q) is the distance (dissimilarity) between points (data objects), p and q.
A distance that satisfies these properties is called a metric. Following is a list of several common distance measures to compare multivariate data. We will assume that the attributes are all continuous.

Euclidean Distance
Assume that we have measurements xik, i=1,…,N, on variables k=1,…,p (also called attributes).

The Euclidean distance between the ith and jth objects is

$dE(i,j)=(\sum k=1p(xik-xjk)2)12$ for every pair (i, j) of observations.
The weighted Euclidean distance is:

$dWE(i,j)=(\sum k=1pWk(xik-xjk)2)12$

If scales of the attributes differ substantially, standardization is necessary.

Minkowski Distance
The Minkowski distance is a generalization of the Euclidean distance.

With the measurement, xik,i=1,…,N,k=1,…,p, the Minkowski distance is

$dM(i,j)=(\sum k=1p|xik-xjk|\lambda)1\lambda$ where $\lambda \geq 1$. It is also called the $L\lambda$ metric.
$\lambda=1$:L1 metric, Manhattan or City-block distance.
$\lambda=2$:L2 metric, Euclidean distance.
$\lambda \rightarrow \infty$:$L\infty$ metric, Supremum distance.
$\lim\lambda \rightarrow \infty=(\sum k=1p|xik-xjk|\lambda)1\lambda=\max(|xi1-xj1|,...,|xip-xjp|)$

Note that λ and p are two different parameters. Dimension of the data matrix remains finite.

Mahalanobis Distance
Let X be a N × p matrix. Then the ith row of X is

$xiT=(xi1,...,xip)$

The Mahalanobis distance is

$d_{MH}(i,j)=((x_i-x_j)T\Sigma-1(x_i-x_j))12$ where $\sum$ is the $p\times p$ sample covariance matrix.

UNIT III

Data Mining - Frequent Pattern Analysis:

Imagine that you are a sales manager in AllElectronics, and you are talking to a customer who recently bought a PC and a digital camera from the store. What should you recommend to her next? Information about which products are frequently purchased by your customers following their purchases of a PC and a digital camera in sequence would be very helpful in making your recommendation. Frequent patterns and association rules are the knowledge that you want to mine in such a scenario.

Frequent patterns are patterns (such as itemsets, subsequences, or substructures) that appear in a data set frequently. 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. Finding such frequent patterns plays an essential role in mining associations, correlations, and many other interesting relationships among data. Moreover, it helps in data classification, clustering, and other data mining tasks. Thus, frequent pattern mining has become an important data mining task and a focused theme in data mining research.

In this chapter, we introduce the basic concepts of frequent patterns, associations, and correlations (Section 6.1) and study how they can be mined efficiently (Section 6.2). We also discuss how to judge whether the patterns

found are interesting (Section 6.3). In Chapter 7, we extend our discussion to advanced methods of frequent pattern mining, which mine more complex forms of frequent patterns and consider user preferences or constraints to speed up the mining process.

Basic Concepts

Frequent pattern mining searches for recurring relationships in a given data set. This section introduces the basic concepts of frequent pattern mining for the discovery of interesting associations and correlations between itemsets in transactional and relational databases. We begin in Section 6.1.1 by presenting an example of market basket analysis, the earliest form of frequent pattern mining for association rules. The basic concepts of mining frequent patterns and associations are given in Section 6.1.2.

Market Basket Analysis: A Motivating Example

Frequent itemset mining leads to the discovery of associations and correlations among items in large transactional or relational data sets. With massive amounts of data continuously being collected and stored, many industries are becoming interested in mining such patterns from their databases. The discovery of interesting correlation relationships among huge amounts of business transaction records can help in many business decision-making processes, such as catalog design, cross-marketing, and customer shopping behavior analysis.

A typical example of frequent itemset mining is market basket analysis. This process analyzes customer buying habits by finding associations between the different items that customers place in their "shopping baskets" (Figure 6.1). The discovery of such associations can help retailers develop marketing strategies by gaining insight into which items are frequently purchased together by customers. For instance, if customers are buying milk, how likely are they to also buy bread (and what kind of bread) on the same trip to the supermarket?

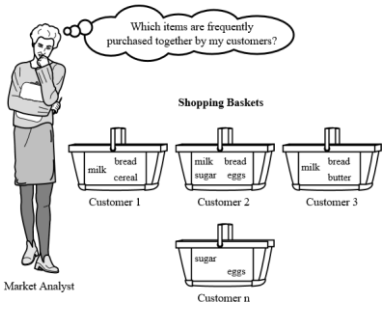


Figure 6.1: Market basket analysis.

Such information can lead to increased sales by helping retailers do selective marketing and plan their shelf space.

Let's look at an example of how market basket analysis can be useful.

Example 6.1 Market basket analysis. Suppose, as manager of an AllElectronicsbranch, you would like to learn more about the buying habits of your customers. Specifically, you wonder, "Which groups or sets of items are customers likely to purchase on a given trip to the store?" To answer your question, market basket analysis may be performed on the retail data of customer transactions at your store. You can then use the results to plan marketing or advertising strategies, or in the design of a new catalog. For instance, market basket analysis may help you design different store layouts. In one strategy, items that are frequently purchased together can be placed in proximity in order to further encourage the sale of such items together. If customers who purchase computers also tend to buy antivirus software at the same time, then placing the hardware display close to the software display may help increase the sales of both items. In an alternative strategy, placing hardware and software at opposite ends of the store may entice customers who purchase such items to pick up other items along the way. For instance, after deciding on an expensive computer, a customer may observe security systems for sale while heading toward the software display to purchase antivirus software and may decide to purchase a home security system as well. Market basket analysis can also help retailers plan which items to put on sale at reduced prices. If customers tend to purchase computers and printers together, then having a sale on printers may encourage the sale of printers as well as computers.

If we think of the universe as the set of items available at the store, then each item has a Boolean variable representing the presence or absence of that item. Each basket can then be represented by a Boolean vector of values assigned to these variables. The Boolean vectors can be analyzed for buying patterns that reflect items that are frequently associated or purchased together. These patterns can be represented in the form of association rules. For example, the information that customers who purchase computers also tend to buy antivirus software at the same time is represented in the association rule below:

computer $\Rightarrow$ antivirus software [support = 2%,confidence= 60%] (6.1)

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 (6.1) means that 2% of all the transactions under analysis show that computer and antivirus 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. Such thresholds can be set by users or domain experts. Additional analysis can be performed to discover interesting statistical correlations between associated items.

Frequent Itemsets, Closed Itemsets, and Association Rules

Let I = {I1, I2,..., Im} be a set of items. Let D, the task-relevant data, be a set of database transactions where each transaction T is a nonempty set of items such that T ⊆ I. Each transaction is associated with an identifier, called TID. Let A be a set of items. A transactionT is said to contain A if A ⊆ T. An association rule is an implication of the form A ⇒ B, where A ⊂ I, B ⊂ I, A 6= ∅, B 6= ∅, and A ∩ B = φ. The rule A ⇒ 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(A ∪ B).1The rule A ⇒ 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). That is,

$$support(A \Rightarrow B) = P(A \cup B) \quad (6.2)$$

$$confidence(A \Rightarrow B) = P(B|A). \quad (6.3)$$

Rules that satisfy bo th a minimum suppo rt th reshold (*min_sup*) a nd a minimum confidence threshold (min conf) are called strong. By convention, we write support and confidence values so as to occur between 0% and 100%, rather than 0 to 1.0.

A set of items is referred to as an itemset.2An itemset that contains k items is a k-itemset. The set {computer, antivirus software} is a 2-itemset. The occurrence frequency of an itemset is the number of transactions that contain the itemset. This is also known, simply, as the frequency, support count, or count of the itemset. Note that the itemset support defined in Equation (6.2) is sometimes referred to as relative support, whereas the occurrence frequency is called the absolute support. If the relative support of an itemset I satisfies a prespecified minimum support threshold (i.e., the absolute support of I satisfies the corresponding minimum support count threshold), then I is a frequent itemset.3The set of frequent k-itemsets is commonly denoted by Lk.4

---

1 Notice that the notation P(A ∪ B) indicates the probability that a transaction contains the union of set A and set B (i.e., it contains every item in A and in B). This should not be confused with P(A or B), which indicates the probability that a transaction contains either A or B.

2In the data mining research literature, “itemset” is more commonly used than “item set.” 3 In early work, itemsets satisfying minimum support were referred to as large. This term,

however, is somewhat confusing as it has connotations to the number of items in an itemset rather
than the frequency of occurrence of the set. Hence, we use the more recent term frequent.
4 Although the term frequent is preferred over large, for historical reasons frequent kitemsets are still denoted as Lk.

From Equation (6.3), we have

$$P(B|A) = \frac{\text{support}(A \cup B)}{\text{support}(A)} = \frac{\text{support count}(A \cup B)}{\text{support count}(A)} \quad (6.4)$$

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

Equation (6.4) shows that the confidence of rule A⇒B can be easily derived from the support counts of A and A ∪ B. That is, once the support counts of A, B, and A ∪ B are found, it is straightforward to derive the corresponding association rules A⇒B and B⇒A and check whether they are strong. Thus the problem of mining association rules can be reduced to that of mining frequent itemsets.

In general, association rule mining can be viewed as a two-step process:

Find all frequent itemsets: By definition, each of these itemsets will occur at least as frequently as a predetermined minimum support count, min sup.

Generate strong association rules from the frequent itemsets: By definition, these rules must satisfy minimum support and minimum confidence.

Additional interestingness measures can be applied for the discovery of correlation relationships between associated items, as will be discussed in Section 6.3. Because the second step is much less costly than the first, the overall performance of mining association rules is determined by the first step.

A major challenge in mining frequent itemsets from a large data set is the fact that such mining often generates a huge number of itemsets satisfying the minimum support (min sup) threshold, especially when min sup is set low. This is because if an itemset is frequent, each of its subsets is frequent as well. A long itemset will contain a combinatorial number of shorter, frequent sub-itemsets. For example, a frequent itemset of length 100, such as {a1, a2,..., a100}, contains $\binom{100}{1}$ = 100 frequent 1-itemsets: {a1}, {a2}, ..., $\{a_{100}\}$, $\binom{100}{2}$ frequent 2-itemsets: {a1, a2}, {a1, a3}, ..., {a99, a100}, and so on. The total number of frequent itemsets that it contains is thus,

$$\binom{100}{1} + \binom{100}{2} + \cdots + \binom{100}{100} = 2^{100} - 1 \approx 1.27 \times 10^{30}. \quad (6.5)$$

This is too huge a number of itemsets for any computer to compute or store. To overcome this difficulty, we introduce the concepts of closed frequent itemset and maximal frequent itemset.

An itemset X is closed in a data set D if there exists no proper superitemset5Y such that Y has the same support count as X in D. An itemset X is a closed frequent itemset in set D if X is both closed and frequent in D.

An itemset X is a maximal frequent itemset (or max-itemset) in a data set D if X is frequent, and there exists no super-itemset Y such that $X \subset Y$ and Y is frequent in D.

Let C be the set of closed frequent itemsets for a data set D satisfying a minimum support threshold, min sup. Let Mbe the set of maximal frequent itemsets for D satisfying min sup. Suppose that we have the support count of each itemset in C and M. Notice that C and its count information can be used to derive the whole set of frequent itemsets. Thus we say that C contains complete information regarding its corresponding frequent itemsets. On the other hand, M registers only the support of the maximal itemsets. It usually does not contain the complete support information regarding its corresponding frequent itemsets. We illustrate these concepts with the following example.

Example 6.2 Closed and maximal frequent itemsets. Suppose that a transaction database has only two transactions: {ha1, a2,..., a100i; ha1, a2,..., a50i}. Let the minimum support count threshold be min sup = 1. We find two closed frequent itemsets and their support counts, that is, C = {{a1, a2,..., a100} : 1; {a1, a2,..., a50} : 2}. There is only one maximal frequent itemset: M = {{a1, a2,..., a100} : 1}. Notice that we cannot include {a1, a2,..., a50} as a maximal frequent itemset because it has a frequent super-set, {a1, a2,..., a100}. Compare this to the above, where we determined that there are $2^{100}-1$ frequent itemsets, which is too huge a set to be enumerated!

The set of closed frequent itemsets contains complete information regarding the frequent itemsets. For example, from C, we can derive, say, (1) {a2, a45 : 2} since {a2, a45} is a sub-itemset of the itemset {a1, a2,..., a50 : 2}; and (2) {a8, a55 : 1} since {a8, a55} is not a sub-itemset of the previous itemset but of the itemset {a1, a2,..., a100 : 1}. However, from the maximal frequent itemset, we can only assert that both itemsets ({a2, a45} and {a8, a55}) are frequent, but we cannot assert their actual support counts.

Frequent Itemset Mining Methods

In this section, you will learn methods for mining the simplest form of frequent patterns, such as those discussed for market basket analysis in Section 6.1.1. We begin by presenting Apriori, the basic algorithm for finding frequent itemsets(Section 6.2.1). In Section 6.2.2, we look at how to generate strong

5Y is a proper super-itemset of X if X is a proper sub-itemset of Y , that is, if $X \subset Y$. In other words, every item of X is contained in Y but there is at least one item of Y that is not in X.

association rules from frequent itemsets. Section 6.2.3 describes several variations to the Apriori algorithm for improved efficiency and scalability. Section 6.2.4 presents pattern-growth methods for mining frequent itemsets that confine the subsequent search space to only the datasets containing the current frequent itemsets. Section 6.2.5 presents methods for mining frequent itemsets that take advantage of vertical data format.

The Apriori Algorithm: Finding Frequent Itemsets by Confined Candidate Generation

Aprioriis a seminal algorithmproposedbyR.AgrawalandR. Srikantin 1994for miningfrequentitemsetsforBooleanassociationrules. Thenameofthealgorithm is based on the fact that the algorithm uses prior knowledge of frequent itemset properties, as we shall see below. Apriori employs an iterative approachknown as a level-wise search, where k-itemsets are used to explore (k + 1)-itemsets. First, the set of frequent 1-itemsets is found by scanning the databaseto accumulate the countforeachitem,andcollectingthoseitemsthatsatisfyminimumsupport. The resulting set is denoted by L1. Next, L1 is used to find L2, the set of frequent 2itemsets,whichisusedto findL3,andso on,untilnomorefrequentk-itemsetscan be found. The finding of each Lk requires one full scan of the database.

To improve the efficiency of the level-wise generation of frequent itemsets, an important property called the Apriori property, presented below, is used to reduce the search space. We will first describe this property, and then show an example illustrating its use.

Apriori property: All nonempty subsets of a frequent itemset must also be frequent.

The Apriori property is based on the following observation. By definition, if an itemset I does not satisfy the minimum support threshold, min sup, then I is not frequent; that is, $P(I) < \text{min sup}$. If an item A is added to the itemset I, then the resulting itemset (i.e., $I \cup A$) cannot occur more frequently than I. Therefore, $I \cup A$ is not frequent either; that is, $P(I \cup A) < \text{min sup}$.

This property belongs to a special category of properties called antimonotonicityin the sense that if a set cannot pass a test, all of its supersets will fail the same test as well. It is called antimonotonicitybecause the property is monotonic in the context of failing a test.6

"How is the Apriori property used in the algorithm?" To understand this, let us look at how L_{k-1} is used to find L_k for $k \geq 2$. A two-step process is followed, consisting of join and prune actions.

The join step: To find L_k, a set of candidate k-itemsets is generated by joining L_{k-1} with itself. This set of candidates is denoted C_k. Let l_1 and l_2 be itemsets in L_{k-1}. The notation $l_i[j]$ refers to the jth item in l_i (e.g., $l_1[k-2]$ refers to the second to the last item in l_1). For efficient implementation, Apriori assumes that items within a transaction or itemset are sorted in lexicographic order. For the $(k-1)$-itemset, l_i, this means that the items are

---

6 The Apriori property has many applications. For example, it can also be used to prune search during data cube computation (Chapter 5).

sorted such that li[1] < li[2] < < li[k −1]. The join, Lk−1 aaLk−1, is performed, where members of Lk−1 are joinable if their first (k −2) items are in common. That is, members l1 and l2 of Lk−1 are joined if (l1[1] = l2[1]) A (l1[2] = l2[2]) A A (l1[k −2] = l2[k −2]) A(l1[k −1] < l2[k −1]). The condition l1[k −1] < l2[k −1] simply ensures Table 6.1: Transactional data for an AllElectronicsbranch.

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

that no duplicates are generated. The resulting itemset formed by joining l1 and l2 is {l1[1], l1[2], , l1[k −2], l1[k −1], l2[k −1]}.

The prune step: Ck is a superset of Lk, that is, its members may or may not be frequent, but all of the frequent k-itemsets are included in Ck. A scan of the database to determine the count of each candidate in Ck would result in the determination of Lk (i.e., all candidates having a count no less than the minimum support count are frequent by definition, and therefore belong to Lk). Ck, however, can be huge, and so this could involve heavy computation. ToreducethesizeofCk,theAprioripropertyisusedasfollows. Any (k −1)-itemset that is not frequent cannot be a subset of a frequent kitemset. Hence, ifany(k −1)-subsetofacandidatek-itemsetis notinLk−1, thenthecandidatecannotbefrequenteitherandsocanberemovedfromCk. This subset testing can be done quickly by maintaining a hash tree of all frequent itemsets.

Example 6.3 Apriori. Let's look at a concrete example, based on the AllElectronicstransaction database, D, of Table 6.1. There are nine transactions in this database, that is, |D| = 9. We use Figure 6.2 to illustrate the Apriori algorithm for finding frequent itemsets in D.

In the first iteration of the algorithm, each item is a member of the set ofcandidate 1- itemsets, C1. The algorithm simply scans all of the transactions in order to count the number of

occurrences of each item.

Suppose that the minimum support count required is 2, that is, min sup = 2. (Here, we are referring to absolute support because we are using a support count. The corresponding relative support is 2/9 = 22%). The set of frequent 1-itemsets, L1, can then be determined. It consists of the candidate 1-itemsets satisfying minimum support. In our example, all of the candidates in C1 satisfy minimum support.

To discover the set of frequent 2-itemsets, L2, the algorithm uses the join

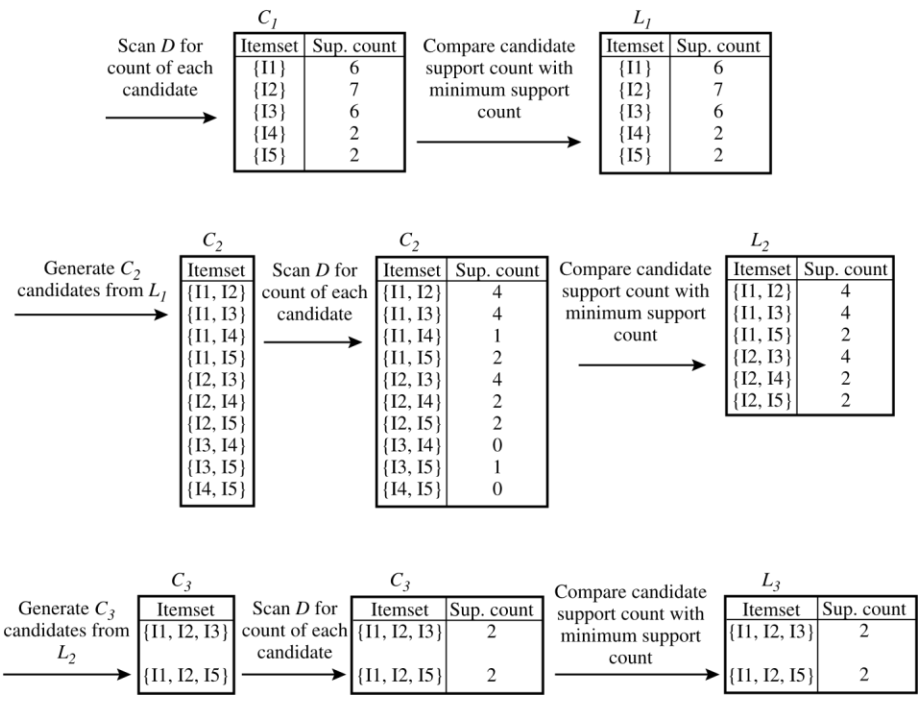


Figure 6.2: Generation of candidate itemsets and frequent itemsets, where the minimum support count is 2.

L1 aaL1 to generate a candidate set of 2-itemsets, C2.7C2 consists of $\binom{|L_1|}{2}$2-itemsets. Note that no candidates are removed from C2 during the prune step because each subset of the candidates is also frequent.

Next, the transactions in D are scanned and the support count of each candidate itemset in C2 is accumulated, as shown in the middle table of the second row in Figure 6.2.

7L1 aaL1 is equivalent to L1 ×L1, since the definition of Lk aaLk requires the two joining itemsets to share k −1

= 0 items.

The set of frequent 2-itemsets, L2, is then determined, consisting of those candidate 2-itemsets in C2 having minimum support.

The generation of the set of candidate 3-itemsets, C3, is detailed in Figure 6.3. From the join step, we first get C3 = L2 aaL2 = {{I1, I2, I3}, {I1, I2, I5}, {I1, I3, I5}, {I2, I3, I4}, {I2, I3, I5}, {I2, I4, I5}}. Based on the Apriori property that all subsets of a frequent itemset must also be frequent, we can determine that the four latter candidates are impossible to be frequent. We therefore remove them from C3, thereby saving the effort of unnecessarily obtaining their counts during the subsequent scan of D to determine

(a) Join: C3 = L2 aaL2 = {{I1, I2}, {I1, I3}, {I1, I5}, {I2, I3}, {I2, I4}, {I2, I5}} aa {{I1, I2}, {I1, I3}, {I1, I5}, {I2, I3}, {I2, I4}, {I2, I5}}

= {{I1, I2, I3}, {I1, I2, I5}, {I1, I3, I5}, {I2, I3, I4}, {I2, I3, I5}, {I2, I4, I5}}.

Prune using the Apriori property: All nonempty subsets of a frequent itemset mustalso be frequent. Do any of the candidates have a subset that is not frequent?

The 2-item subsets of {I1, I2, I3} are {I1, I2}, {I1, I3}, and {I2, I3}. All 2-item subsets of {I1, I2, I3} are members of L2. Therefore, keep {I1, I2, I3} in C3.

The 2-item subsets of {I1, I2, I5} are {I1, I2}, {I1, I5}, and {I2, I5}. All 2-item subsets of {I1, I2, I5} are members of L2. Therefore, keep {I1, I2, I5} in C3.

The 2-item subsets of {I1, I3, I5} are {I1, I3}, {I1, I5}, and {I3, I5}. {I3, I5} is not a member of L2, and so it is not frequent. Therefore, remove {I1, I3, I5} from C3.

The 2-item subsets of {I2, I3, I4} are {I2, I3}, {I2, I4}, and {I3, I4}. {I3, I4} is not a member of L2, and so it is not frequent. Therefore, remove {I2, I3, I4} from C3.

The 2-item subsets of {I2, I3, I5} are {I2, I3}, {I2, I5}, and {I3, I5}. {I3, I5} is not a member of L2, and so it is not frequent. Therefore, remove {I2, I3, I5} from C3.

The 2-item subsets of {I2, I4, I5} are {I2, I4}, {I2, I5}, and {I4, I5}. {I4, I5} is not a member of L2, and so it is not frequent. Therefore, remove {I2, I4, I5} from C3.

Therefore, C3 = {{I1, I2, I3}, {I1, I2, I5}} after pruning.

Figure 6.3: Generation and pruning of candidate 3-itemsets, C3, from L2 using the Apriori property.

L3. Note that when given a candidate k-itemset, we only need to check if its $(k-1)$-subsets are frequent since the Apriori algorithm uses a level-wise searchstrategy. TheresultingprunedversionofC3 isshowninthefirsttable of the bottom row of Figure 6.2.

The transactions in D are scanned in order to determine L3, consisting of those candidate 3-

itemsets in C3 having minimum support (Figure 6.2).

The algorithm uses L3 aaL3 to generate a candidate set of 4-itemsets, C4. Although the join results in {{I1, I2, I3, I5}}, itemset {I1, I2, I3, I5} is pruned because its subset {I2, I3, I5} is not frequent. Thus, C4 = φ, and the algorithm terminates, having found all of the frequent itemsets.

Figure 6.4 shows pseudo-code for the Apriori algorithm and its related procedures. Step 1 of Apriori finds the frequent 1-itemsets, L1. In steps 2 to 10, Lk−1 is used to generate candidates Ck in order to find Lk for k ≥ 2. The apriori gen procedure generates the candidates and then uses the Apriori property to eliminate those having a subset that is not frequent (step 3). This procedure is described below. Once all of the candidates have been generated, the database is scanned (step 4). For each transaction, a subset function is used to find all subsets of the transaction that are candidates (step 5), and the count for each of these candidates is accumulated (steps 6 and 7). Finally, all of those candidates satisfying minimum support (step 9) form the set of frequent itemsets, L (step 11). A procedure can then be called to generate association rules from the frequent itemsets. Such a procedure is described in Section 6.2.2.

Algorithm: Apriori. Find frequent itemsets using an iterative level-wise approach based on candidate generation.

Input:

D, a database of transactions;

min sup, the minimum support count threshold.

Output: L, frequent itemsets in D.

Method:

(1) L1 = find frequent 1-itemsets(D);

(2) for (k = 2;Lk−1 =6 φ;k++) {

Ck = apriorigen(Lk−1);

for each transaction t ∈ D { // scan D for counts

Ct = subset(Ck, t); // get the subsets of t that are candidates

for each candidate c ∈Ct

c.count++;

(8) }

Lk = {c ∈ Ck|c.count≥ min sup}

} (11) return L = UkLk;

procedure apriorigen(Lk−1:frequent (k − 1)-itemsets)

for each itemset l1 ∈ Lk−1

for each itemset l2 ∈ Lk−1

(3) if (l1[1] = l2[1]) A (l1[2] = l2[2])

A... A (l1[k − 2] = l2[k − 2]) A (l1[k − 1] < l2[k − 1]) then {

c = l1 aal2; // join step: generate candidates

if has infrequent subset(c, Lk−1) then -

delete c; // prune step: remove unfruitful candidate

else add c to Ck;

(8) }
(9) return Ck;
procedure has infrequent subset(c: candidate k-itemset;
Lk−1: frequent (k − 1)-itemsets); // use prior knowledge
for each (k − 1)-subset s of c
if s 6∈ Lk−1 then
return TRUE;
return FALSE;

Figure 6.4: The Apriori algorithm for discovering frequent itemsets for mining Boolean association rules.

The apriori gen procedure performs two kinds of actions, namely, join and prune, as described above. In the join component, Lk−1 is joined with Lk−1 to generate potential candidates (steps 1 to 4). The prune component (steps 5 to 7) employs the Apriori property to remove candidates that have a subset that is not frequent. The test for infrequent subsets is shown in procedure has infrequent subset.

Generating Association Rules from Frequent Itemsets

Once the frequent itemsets from transactions in a database D have been found, it is straightforward to generate strong association rules from them (where strong association rules satisfy both minimum support and minimum confidence). This can be done using Equation (6.4) for confidence, which we show again here for completeness:

$$\text{confidence}(A \Rightarrow B) = P(B|A) = \frac{\text{support count}(A \cup B)}{\text{support count}(A)}.$$

The conditional probability is expressed in terms of itemset support count, where support count(AUB) is the number of transactions containing the itemsetsA ∪ B, and support count(A) is the number of transactions containing the itemset A. Based on this equation, association rules can be generated as follows:

For each frequent itemset l, generate all nonempty subsets of l.

For every nonempty subset s of l, output the rule "s ⇒ (l −s)" if

$\frac{\text{support count}(l)}{\text{support count}(s)} \geq \text{min conf}$, where min conf is the minimum confidence

threshold.

Because the rules are generated from frequent itemsets, each one automatically satisfies minimum support. Frequent itemsets can be stored ahead of time in hash tables along with their counts so that they can be accessed quickly.

Example 6.4 Generating association rules. Let's try an example based on the transactional data for AllElectronicsshown in Table 6.1. The data contain frequent itemset X =

{I1, I2, I5}. What are the association rules that can be generated from X? The nonempty subsets of X are {I1, I2}, {I1, I5}, {I2, I5}, {I1}, {I2}, and {I5}. The resulting association rules are as shown below, each listed with its confidence:

{ c
I o
1 n

, I 2 }

⇒ I 5 , { I 1 , I 5 }

⇒ I 2 , { I

confidence = 2/4 = 50% confidence = 2/2 = 100% co

2, I5} ⇒ I1, 2/2 = 100% confidence = 2/6 = 33%

$I2 \Rightarrow \{I1, I5\}$, confidence = 2/7 = 29%

$I5 \Rightarrow \{I1, I2\}$, confidence = 2/2

=

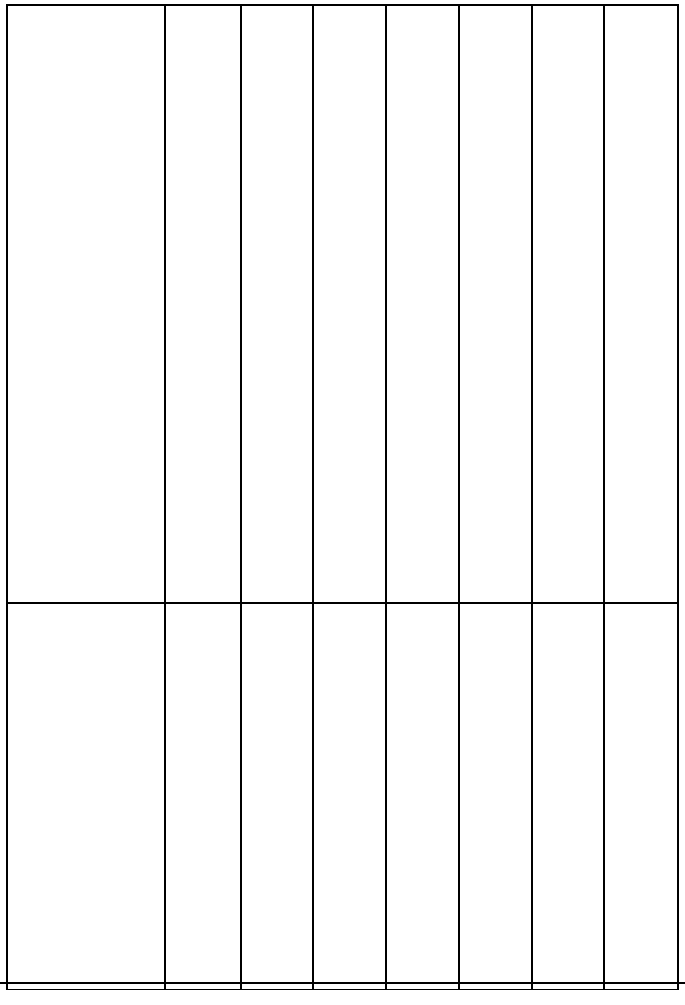
1
0
0
%

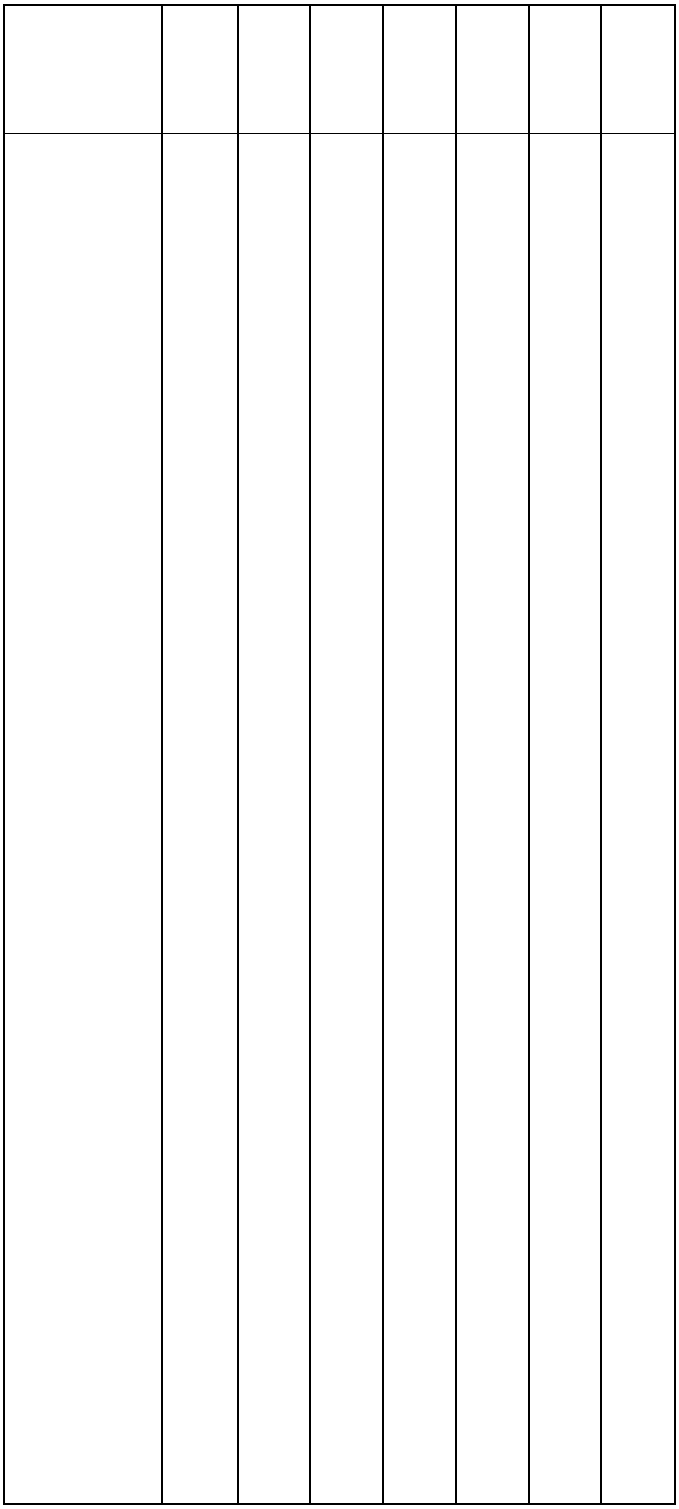
If the minimum confidence threshold is, say, 70%, then only the second, third, and last rules above are output, because these are the only ones generated that are strong. Note that, unlike conventional classification rules, association rules can contain more than one conjunct in the right-hand side of the rule.

Improving the Efficiency of Apriori

"How can we further improve the efficiency of Apriori-based mining?" Many variations of the Apriori algorithm have been proposed that focus on improving the efficiency of the original algorithm. Several of these variations are summarized as follows:

Hash-based technique (hashing itemsets into corresponding buckets): A hash-based technique can be used to reduce the size of the candidate kitemsets, Ck, for k>1. For example, when scanning each transaction in





H2

Create hash table H2 using hash function

h(x, y) 5 ((order of x) 310 1 (order of y)) mod 7

Figure 6.5: Hash table, H2, for candidate 2- itemsets: This hash table was generated by scanning the transactions of Table

6.1 while determining L1. If the minimum support count is, say, 3, then the itemsets in buckets 0, 1, 3, and 4 cannot be frequent and so they should not be included in C2.

the database to generate the frequent 1-itemsets, L1, we can generate all of the 2-itemsets for each transaction, hash (i.e., map) them into the different buckets of a hash table structure, and increase the corresponding bucket counts (Figure 6.5). A 2-itemset whose corresponding bucket count in the hash table is below the support threshold cannot be frequent and thus should be removed from the candidate set. Such a hash-based technique may substantially reduce the number of the candidate k-itemsets examined (especially when k = 2).

Transaction reduction (reducing the number of transactions scanned in future iterations): A transaction that does not contain any frequent kitemsets cannot contain any frequent (k + 1)-itemsets. Therefore, such a transaction can be marked or removed from further consideration because subsequent scans of the database for j-itemsets, where j > k, will not need to consider such a transaction.

Partitioning (partitioning the data to find candidate itemsets): A partitioning technique can be used that requires just two database scans to mine the frequent itemsets (Figure 6.6). It consists of two phases. In Phase I, the algorithm divides the transactions of D into n nonoverlapping partitions. If the minimum relative support threshold for transactions in D is min sup, then the minimum support count for a partition is min sup × the number of transactions in that partition. For each partition, all the local frequent itemsets, i.e., the itemsets frequent within the partition, are found.

A local frequent itemset may or may not be frequent with respect to the entire database, D. However, any itemset that is potentially frequent with respect to D must occur as a frequent itemset in at least one of the partitions8. Therefore, all local frequent itemsets are candidate itemsets with respect to D. The collection of frequent itemsets from all partitions forms the global candidate itemsetswith respect to D. In Phase II, a second scan of D is conducted in which the actual support of each candidate

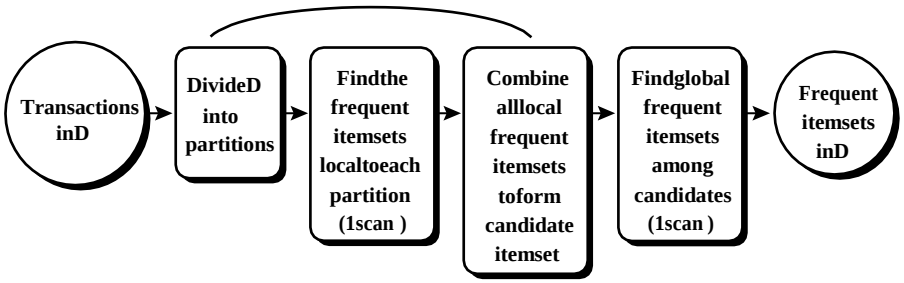


Figure 6.6: Mining by partitioning the data.

is assessed in order to determine the global frequent itemsets. Partition size and the number of partitions are set so that each partition can fit into main memory and therefore be read only once in each phase.

Sampling (mining onasubsetofthegivendata): Thebasicideaofthesampling approach is to pick a random sample S of the given data D, and then search for frequent itemsets in S instead of D. In this way,wetradeoffsomedegree

8The proof of this property is left as an exercise (see Exercise 6.3(d)).

of accuracy against efficiency. The sample size of S is such that the search for frequent itemsets in S canbe done in main memory, and so only one scan of the transactions in S is required overall. Because we are searching for frequent itemsets in S rather than in D, it is possible that we will miss some of the global frequent itemsets. To reduce this possibility, we use a lower support threshold than minimum support to find the frequent itemsets local to S (denoted LS). The rest of the database is then used to compute the actual frequencies of each itemset in LS. A mechanism is used to determine whether allofthe global frequent itemsets are included in LS. If LS actually contains all of the frequent itemsets in D, then only one scan of D is required. Otherwise, a second pass can be done in order to find the frequent itemsets that were missed in the first pass. The sampling approach is especially beneficial when efficiency is of utmost importance, such as in computationally intensive applications that must be run frequently.

Dynamic itemset counting (adding candidate itemsets at different points during a scan): A dynamic itemset counting technique was proposed in which the database is partitioned into blocks marked by start points. In this variation, new candidate itemsets can be added at any start point, unlike in Apriori, which determines new candidate itemsets only immediately before each complete database scan. The technique uses the count-so-far as the lower bound of the actual count. If the count-so-far passes the minimum support, the itemset is added into the frequent itemset collection and can be used to generate longer candidates. This leads to fewer database scans than Apriori for finding all the frequent itemsets.

Other variations are discussed in the next chapter.

A Pattern-Growth Approach for Mining Frequent Itemsets

As we have seen, in many cases the Apriori candidate generate-and-test method significantly reduces the size of candidate sets, leading to good performance gain. However, it can suffer from two nontrivial costs:

It may still need to generate a huge number of candidate sets. For example, if there are 104 frequent 1-itemsets, the Apriori algorithm will need to generate more than 107 candidate 2-itemsets.

It may need to repeatedly scan the whole database and check a large set of candidates by pattern matching.It is costly to go over each transaction in the database to determine the support of the candidate itemsets.

“Can we design a method that mines the complete set of frequent itemsets without such a costly candidate generation process?” An interesting method in

this attempt is called frequent-pattern growth, or simply FP-growth, which adopts a divide-and-conquer strategy as follows. First, it compresses the database representing frequent items into a frequent-pattern tree, or FPtree, which retains the itemset association information. It then divides the compressed database into a set of conditional databases (a special kind of projected database), each associated with one frequent item or "pattern fragment," and mines each such database separately. This approach may substantially reduce the sizes of datasets to be searched along with pattern growth. You'll see how it works with the following example.

Example 6.5 FP-growth (finding frequent itemsets without candidate generation). We re-examine the mining of transaction database, D, of Table 6.1 in Example 6.3 using the frequent-pattern growth approach.

The first scan of the database is the same as Apriori, which derives the set of frequent items (1-itemsets) and their support counts (frequencies). Let the minimum support count be 2. The set of frequent items is sorted in the order of descending support count. This resulting set or list is denoted by L. Thus, we have L ={{I2: 7}, {I1: 6}, {I3: 6}, {I4: 2}, {I5: 2}}.

An FP-tree is then constructed as follows. First, create the root of the tree, labeledwith"null." ScandatabaseDasecondtime. Theitemsineachtransaction areprocessedinLorder(i.e.,sortedaccordingtodescendingsupportcount),anda branchiscreatedforeachtransaction. Forexample,thescanofthefirsttransaction, "T100: I1, I2, I5," which contains three items (I2, I1, I5 in L order), leads to the constructionofthefirstbranchofthetreewiththreenodes,hI2: 1i,hI1:1i,andhI5: 1i,whereI2islinkedasachildtotheroot,I1islinkedtoI2,andI5islinkedtoI1. The secondtransaction,T200,containstheitemsI2andI4inLorder,whichwouldresult inabranchwhereI2islinkedtotherootandI4islinkedtoI2. However,thisbranch would share a common prefix, I2, with the existing path for T100. Therefore, we insteadincrementthecountoftheI2nodeby1,andcreateanewnode,hI4: 1i,which is linked as a child to hI2: 2i. In general, when considering the branch to be added

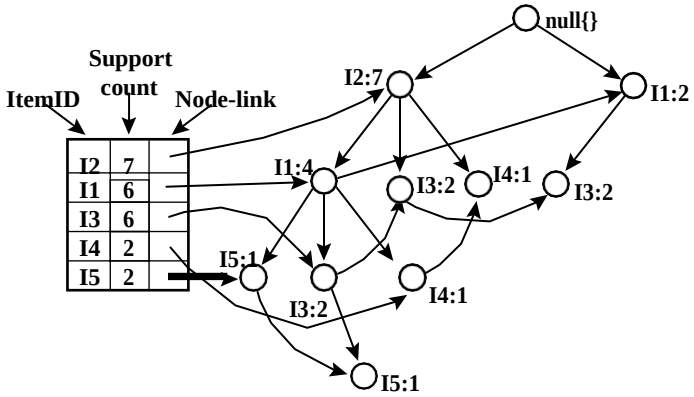


Figure 6.7: An FP-tree registers compressed, frequent pattern information.

for a transaction, the count of each node along a common prefix is incremented by 1, and nodes for the items following the prefix are created and linked accordingly.

To facilitate tree traversal, an item header table is built so that each item points to its occurrences in the tree via a chain of node-links. The tree obtained after scanning all of the transactions is shown in Figure 6.7 with the associated node-links. In this way, the problem of mining frequent patterns in databases is transformed to that of mining the FP-tree.

The FP-tree is mined as follows. Start from each frequent length-1 pattern (as an initial suffix pattern), construct its conditional pattern base(a "sub- database," which consists of the set of prefix paths in the FP-tree cooccurring with the suffix pattern), then construct its (conditional) FP-tree, and perform mining recursively on such a tree. The pattern growth is achieved by the concatenation of the suffix pattern with the frequent patterns generated from a conditional FP-tree.

Mining of the FP-tree is summarized in Table 6.2 and detailed as follows. We first consider I5, which is the last item in L, rather than the first. The reason for starting at the end of the list will become apparent as we explain the FP-tree mining process. I5 occurs in two branches of the FP-tree of Figure 6.7. (The occurrences of I5 can easily be found by following its chain of node-links.) The paths formed by these branches are hI2, I1, I5: 1i and hI2, I1, I3, I5: 1i.

Therefore, considering I5 as a suffix, its corresponding two prefix paths are hI2, I1: 1i and hI2, I1, I3: 1i, which form its conditional pattern base. Using this conditional pattern base as a transaction database, we build an I5-conditional FP-tree, which contains only a single path, hI2: 2, I1: 2i; I3 is not included because its support count of 1 is less than the minimum support count. The single path generates all the combinations of frequent patterns: {I2, I5: 2}, {I1, I5: 2}, {I2, I1, I5: 2}.

For I4, its two prefix paths form the conditional pattern base, {{I2 I1: 1}, {I2: 1}}, which generates a single-node conditional FP-tree, hI2: 2i, and derives one frequent pattern, {I2, I4: 2}.

Table 6.2: Mining the FP-tree by creating conditional (sub-)pattern bases.

Item	Conditional Pattern Base	Conditional FP-tree	Frequent Patterns Generated
I5	{{I2, I1: 1}, {I2, I1, I3: 1}}	{I2: 2, I1: 2}	{I2, I5: 2}, {I1, I5: 2}, {I2, I1, I5: 2}

1
}
,

{
I
2
,

I
1
,

I
3
:

1
}
}

2
}
,

{
I
1
,

I
5
:

2
}
,

{
I
2
,

I
1
,

I
5
:
2
}
{
{
I
2
,

I
1
:

1
}
,

{
I
2
,

I
4
:

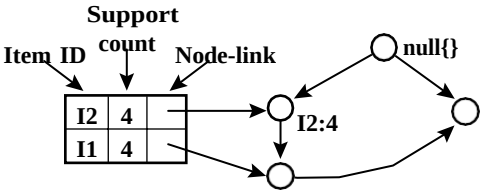
2
}

{
I
2
:

1
}
}

I3 {{I2, I1: 2}, {I2: 2}, {I1: 2}} hI2: 4, I1: 2i, hI1: {I2, I3: 4}, {I1, I3: 4}, {I2, I1, I3:
2i 2}

I1 {{ I2:4 }} hI2:4 i { I2,I1:4 }



I1:2

I1:2

Figure 6.8: The conditional FP-tree associated with the conditional node I3.

Similar to the above analysis, I3's conditional pattern base is {{I2, I1: 2}, {I2: 2}, {I1: 2}}. Its conditional FP-tree has two branches, hI2: 4, I1: 2iand hI1: 2i, as shown in Figure 6.8, which generates the set of patterns {{I2, I3: 4}, {I1, I3: 4},
{I2, I1, I3: 2}}. Finally, I1's conditional pattern base is {{I2: 4}}, whose FP-tree contains only one node, hI2: 4i, which generates one frequent pattern,{I2, I1: 4}. This mining process is summarized in Figure 6.9.

The FP-growth method transforms the problem of finding long frequent patterns to searching for shorter ones in much smaller conditional databases recursively and then concatenating the suffix. It uses the least frequent items as a suffix, offering good selectivity. The method substantially reduces the search costs.

When the database is large, it is sometimes unrealistic to construct a main memory-basedFP-tree. An interesting alternative is to first partition the database into a set of projected databases, and then construct an FP-tree and mine it in each projected database. Such a process can be recursively applied to any projected database if its FP-tree still cannot fit in main memory.

A study on the performance of the FP-growth method shows that it is efficient and scalable for mining both long and short frequent patterns, and is about an order of magnitude faster than the Apriori algorithm.

Mining Frequent Itemsets Using Vertical Data Format

Both the Apriori and FP-growth methods mine frequent patterns from a set of transactions in TID-itemset format (that is, {TID :itemset}), where TID is a transaction-id and itemset is the set of items bought in transaction TID.

Algorithm: FP growth. Mine frequent itemsets using an FP-tree by pattern fragment growth.

Input:

D, a transaction database;

min sup, the minimum support count threshold.

Output: The complete set of frequent patterns.

Method:

The FP-tree is constructed in the following steps:

Scan the transaction database D once. Collect F, the set of frequent items, and their support counts. Sort F in support count descending order as L, the list of frequent items.

Create the root of an FP-tree, and label it as "null." For each transaction Trans in D do the following.

Select and sort the frequent items in Trans according to the order of L. Let the sorted frequent item list in Trans be [p|P], where p is the first element and P is the remaining list. Call insert tree([p|P], T), which is performed as follows. If T has a child N such that N.item-name = p.item-name, then increment N's count by 1; else create a new node N, and let its count be 1, its parent link be linked to T, and its node-link to the nodes with the same item-name via the node-link structure. If P is nonempty, call insert tree(P, N) recursively.

The FP-tree is mined by calling FP growth(FP tree, null), which is implemented as follows.

procedure FP growth(Tree, α)

if Tree contains a single path Pthen

for each combination (denoted as β) of the nodes in the path P

generate pattern β ∪ α with support count = minimum support count of nodes in β;

else for each ai in the header of Tree {

generate pattern β = ai ∪ α with support count = ai.support count;

construct β's conditional pattern base and then β's conditional FP tree Treeβ;

if Treeβ 6= ∅ then

call FP growth(Treeβ, β); }

Figure 6.9: The FP-growth algorithm for discovering frequent itemsets without candidate generation.

Table 6.3: The vertical data format of the transaction data set D of Table 6.1.

itemset	TID set
I1	{T100, T400, T500, T700, T800, T900}
I2	{T100, T200, T300, T400, T600, T800, T900}
I3	{T300, T500, T600, T700, T800, T900}
I4	{T200, T400}
I5	{T100, T800}

This data format is known as the horizontal data format. Alternatively, data can also be presented in item-TID set format (that is, {item :TID set}), where item is an item name, and TID set is the set of transaction identifiers containing the item. This format is known as the vertical data format.

In this section, we look at how frequent itemsets can also be mined efficiently using vertical data format, which is the essence of the ECLAT (Equivalence CLASS Transformation) algorithm.

Example 6.6 Mining frequent itemsets using vertical data format. Consider the horizontal data format of the transaction database, D, of Table 6.1 in Example 6.3. This can

be transformed into the vertical data format shown in Table 6.3 by scanning the data set once. Mining can be performed onthis data set by intersecting the TID sets of every pairoffrequentsingleitems. Theminimumsupportcountis2. Becauseeverysingle itemisfrequentinTable6.3,thereare10intersectionsperformedintotal,whichlead to 8 nonempty 2-itemsets as shown in Table 6.4. Notice that because the itemsets{I1, I4} and {I3, I5} each contain only one transaction, they do not belong to the setof frequent2-itemsets.

Based on the Apriori property, a given 3-itemset is a candidate 3-itemset only if every one of its 2-itemset subsets is frequent. The candidate generation process here will generate only two 3-itemsets: {I1, I2, I3} and {I1, I2, I5}. By intersecting the TID sets of any two corresponding 2-itemsets of these candidate

Table 6.4: The 2-itemsets in vertical data format.

itemset	TID set
{I1, I2}	{T100, T400, T800, T900}

{
T
4
0
0
}

{
T
1
0
0
,

T
8
0
0
}
{
T
3
0
0
,

T
6
0
0
,

T
8
0
0
,

T
9
0
0
}

{
T
2
0
0
,

T
4
0
0
}
{
T
1
0
0
,

T
8
0
0
}
{
T
8
0
0
}

3-itemsets, it derives Table 6.5, where there are only two frequent 3-itemsets: {I1, I2, I3: 2} and {I1, I2, I5: 2}.

Example 6.6 illustrates the process of mining frequent itemsets by exploring the vertical data format. First, we transform the horizontally formatted data to the vertical format by scanning the data set once. The support count of an itemset is simply the length of the TID set of the itemset. Starting with k = 1, the frequent k-itemsets can be used to construct the candidate (k +1)-itemsets based on the Apriori property. The computation is done by intersection of the TID sets of the frequent k-itemsets to compute the TID sets of the corresponding (k + 1)-itemsets. This process repeats, with k incremented by 1 each time, until no frequent itemsets or no candidate itemsets can be found.

Besides taking advantage of the Apriori property in the generation of candidate (k + 1)-itemset

from frequent k-itemsets, another merit of this method is that there is no need to scan the database to find the support of (k+1) itemsets (for $k \geq 1$). This is because the TID set of each k-itemset carries the complete information required for counting such support. However, the TID sets can be

quite long, taking substantial memory space as well as computation time for intersecting the long sets.

To further reduce the cost of registering long TID sets, as well as the subsequent costs of intersections, we can use a technique called diffset, which keeps track of only the differences of the TID sets of a (k + 1)-itemset and a corresponding k-itēmset. For instance, in Example 6.6 we have {I1} = {T100, T400, T500, T700, T800, T900} and {I1, I2} = {T100, T400, T800, T900}. The

diffsetbetween the two is diffset({I1, I2}, {I1}) = {T500, T700}. Thus, rather than recording the four TIDs that make up the intersection of {I1} and {I2}, we can instead usediffset to record just two TIDs indicating the difference between {I1} and {I1, I2}. Experiments show that in certain situations, such as when the data set contains many dense and long patterns, this technique can substantially reduce the total cost of vertical format mining of frequent itemsets.

Mining Closed and Max Patterns

In Section 6.1.2 we saw how frequent itemset mining may generate a huge number of frequent itemsets, especially when the min sup threshold is set low or when there exist long patterns in the data set. Example 6.2 showed that closed

Table 6.5: The 3-itemsets in vertical data format.

{I1, I2, I3}	{T800, T900}
itemset	TID set
{I1, I2, I5}	{T100, T800}

frequent itemsets9can substantially reduce the number of patterns generated in frequent itemset mining while preserving the complete information regarding the set of frequent itemsets. That is, from the set of closed frequent itemsets, we can easily derive the set of frequent itemsets and their support. Thus in practice, it is more desirable to mine the set of closed frequent itemsets rather than the set of all frequent itemsets in most cases.

"How can we mine closed frequent itemsets?" A na¨ıve approach would be to first mine the complete set of frequent itemsets and then remove every frequent itemset that is a proper subset of, and carries the same support as, an existing frequent itemset. However, this is quite costly. As shown in Example 6.2, this method would have to first derive $2^{100} - 1$ frequent itemsets in order to obtain a length-100 frequent itemset, all before it could begin to eliminate redundant

9 Remember that X is a closed frequent itemset in a data set S if there exists no proper super-itemset Y such that Y has the same support count as X in S, and X satisfies minimum support.

itemsets. This is prohibitively expensive. In fact, there exist only a very small number of closed frequent itemsets in the data set of Example 6.2.

A recommended methodology is to search for closed frequent itemsets directly during the mining process. This requires us to prune the search space as soon as we can identify the case of closed itemsets during mining. Pruning strategies include the following:

Item merging: If every transaction containing a frequent itemset X also contains an itemset Y but not any proper superset of Y , then X UY forms a frequent closed itemset and there is no need to search for any itemset containing X but no Y .

For example, in Table6.2 ofExample6.5, the projectedconditionaldatabase for prefix itemset {I5:2} is {{I2, I1},{I2, I1, I3}}, from which we can see that each of its transactions contains itemset {I2, I1} but no proper superset of

{I2, I1}. Itemset {I2, I1} can be merged with {I5} to form the closed itemset,

{I5, I2, I1: 2}, and we do not need to mine for closed itemsets that contain I5 but not {I2, I1}.

Sub-itemset pruning: If a frequent itemset X is a proper subset of an already found frequent closed itemset Y and support count(X) = support count(Y), then X and all of X's descendants in the set enumeration tree cannot be frequent closed itemsets and thus can be pruned.

Similar to Example 6.2, suppose a transaction database has only two transactions: {ha1, a2,..., a100i, ha1, a2,..., a50i}, and the minimum support count is min sup = 2. The projection on the first item, a1, derives the frequent itemset, {a1, a2,..., a50 : 2}, based on the itemset merging optimization. Because support({a2}) = support ({a1, a2, , a50}) = 2, and {a2}

is a proper subset of {a1, a2, , a50}, there is no need to examine a2 and its

projected database. Similar pruning can be done for a3, , a50 as well. Thus

the mining of closed frequent itemsets in this data set terminates after mining a1's projected database.

Item skipping: In the depth-first mining of closed itemsets, at each level, there will be a prefix itemset X associated with a header table and a projected database. If a local frequent item p has the same support in several header tables at different levels, we can safely prune p from the header tables at higher levels.

Consider, for example, the transaction database above having only two transactions: {ha1, a2,..., a100i, ha1, a2,..., a50i}, where min sup = 2. Because a2 in a1's projected database has the same support as a2 in the global header table, a2 can be pruned from the global header table. Similar pruning can be done for a3, , a50. There is no need to mine anything more after mining

a1's projected database.

Besides pruning the search space in the closed itemset mining process, another important optimization is to perform efficient checking of a newly derived frequent itemset to see whether it is closed, because the mining process cannot ensure that every generated frequent itemset is closed.

When a new frequent itemset is derived, it is necessary to perform two kinds of closure checking: (1) superset checking, which checks if this new frequent itemset is a superset of some already found closed itemsets with the same support, and (2) subset checking, which checks whether the newly found itemset is a subset of an already found closed itemset with the same support.

If we adopt the item merging pruning method under a divide-and-conquer framework, then the superset checking is actually built-in and there is no need to explicitly perform superset checking. This is because if a frequent itemset XUY is found later than itemset X, and carries the same support as X, it must be in X's projected database and must have been generated during itemset merging.

To assist in subset checking, a compressed pattern-tree can be constructed to maintain the set of closed itemsets mined so far. The pattern-tree is similar in structure to the FP-tree except that all of the closed itemsets found are stored explicitly in the corresponding tree branches. For efficient subset checking, we can use the following property: If the current itemset Sc can be subsumed by another already found closed itemset Sa, then (1) Sc and Sa have the same support,

(2) the length of Sc is smaller than that of Sa, and (3) all of the items in Sc are contained in Sa. Based on this property, a two-level hash index structure can be built for fast accessing of the pattern-tree: The first level uses the identifier of the last item in Sc as a hash key (since this identifier must be within the branch of Sc), and the second level uses the support of Sc as a hash key (since Sc and Sa have the same support). This will substantially speed up the subset checking process.

The above discussion illustrates methods for efficient mining of closed frequent itemsets. "Can we extend these methods for efficient mining of maximal frequent itemsets?" Because maximal frequent itemsets share many similarities with closed frequent itemsets, many of the optimization techniques developed here can be extended to mining maximal frequent itemsets. However, we leave this method as an exercise for interested readers.

Which Patterns Are Interesting?—Pattern Evaluation Methods

Mostassociationruleminingalgorithmsemployasupport-confidenceframework. Although minimum support and confidence thresholds help weed out or exclude the exploration of a good number of uninteresting rules, many of the rules generated are still not interesting to the users. Unfortunately, this is especially true when mining at low support thresholds or mining for long patterns. This has been a major bottleneck for successful application of association rule mining.

In this section, we first look at how even strong association rules can be uninteresting and misleading (Section 6.3.1). We then discuss how the supportconfidence framework can be supplemented with additional interestingness measures based on correlation analysis (Section 6.3.2). Section

6.3.3 presents additional pattern evaluation measures. It then provides an overall comparison of all of the measures discussed here. By the end, you will learn which pattern evaluation measures are most effective for the discovery of only interesting rules.

Strong Rules Are Not Necessarily Interesting

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

"How can we tell which strong association rules are really interesting?" Let's examine the following example.

Example 6.7 A misleading "strong" association rule. Suppose we are interested in analyzing transactions at AllElectronicswith respect to the purchase of computer games and videos. Let game refer to the transactions containing computer games, and video refer to those containing videos. Of the 10,000 transactions analyzed, the data show that 6,000 of the customer transactions included computer games, while 7,500 included videos, and 4,000 included both computer games and videos. Suppose that a data mining program for discovering association rules is run on the data, using a minimum support of, say, 30% and a minimum confidence of 60%. The following association rule is discovered:

buys(X, "computer games")⇒buys(X, "videos") [support = 40%, confidence = 66%]
(6.6)

Rule (6.6) is a strong association rule and would therefore be reported, since its support value of 104,,000000 = 40% and confidence value of 46,,000000 = 66% satisfy the minimum support and minimum confidence thresholds, respectively.

However, Rule (6.6) is misleading because the probability of purchasing videos is 75%, which is even larger than 66%. In fact, computer games and videos are negatively associated because the purchase of one of these items actually decreases the likelihood of purchasing the other. Without fully understanding this phenomenon, we could easily make unwise business decisions based on Rule (6.6).

The above example also illustrates that the confidence of a rule A ⇒ B can be deceiving. It does not measure the real strength (or lack of strength) of the correlation and implication between A and

B. Hence, alternatives to the support-confidence framework can be useful in mining interesting data relationships.

From Association Analysis to Correlation Analysis

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

A ⇒ B [support, confidence, correlation]. (6.7)

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

Lift is a simple correlation measure that is given as follows. The occurrence of itemset A is independent of the occurrence of itemset B if P(A ∪ B) = P(A)P(B); otherwise, itemsetsA and B are dependent and correlated as events. This definition can easily be extended to more than two itemsets. The lift between the occurrence of A and B can be measured by computing

$$\text{lift}(A, B) = \frac{P(A \cup B)}{P(A)P(B)}. \quad (6.8)$$

If the resulting value of Equation (6.8) is less than 1, then the occurrence of A is negatively correlated with the occurrence of B, meaning that the occurrence of one likely leads to the absence of the other one. If the resulting value is greater than 1, then A and B are positively correlated, meaning that the occurrence of one implies the occurrence of the other. If the resulting value is equal to 1, then A and B are independent and there is no correlation between them.

Equation (6.8) is equivalent to P(B|A)/P(B), or conf(A ⇒ B)/sup(B), which is also referred as the lift of the association (or correlation) rule A ⇒ B. In other words, it assesses the degree to which the occurrence of one "lifts" the occurrence of the other. For example, if A corresponds to the sale of computer games and B corresponds to the sale of videos, then given the current market conditions, the sale of games is said to increase or "lift" the likelihood of the sale of

videos by a factor of the value returned by Equation (6.8).

6.3. WHICH PATTERNS ARE INTERESTING?—PATTERN EVALUATION METHODS27 Let's go back to the computer game and video data of Example 6.7.

Table 6.6: A 2 × 2 contingency table summarizing the transactions with respect to game and video purchases.

	game	game	Σrow
video	4,000	3,500	7,500
video	2,000	500	2,500
Σcol	6,000	4,000	10,000

Table 6.7: The above contingency table, now shown with the expected values.

	game	game	Σrow
	4,000	3,500	
	(4,500)	(3,000)	
	2,000	500	
	(1,500)	(1,000)	

	0)	
	6 , 0 0 0	4 , 0 0 0

Example 6.8 Correlation analysis using lift. To help filter out misleading "strong" associations of the form A ⇒ B from the data of Example 6.7, we need to study how the two itemsets, A and B, are correlated. Let game refer to the transac-

tions of Example 6.7 that do not contain computer games, and video refer to those that do not contain videos. The transactions can be summarized in a contingency table, as shown in Table 6.6. From the table, we can see that the probability of purchasing a computer game is P({game}) = 0.60, the probability of purchasing a video is P({video}) = 0.75, and the probability of purchasing both is P({game,video}) = 0.40. By Equation (6.8), the lift of Rule (6.6) is P({game, video})/(P({game})×P({video})) = 0.40/(0.60×0.75) = 0.89. Because this value is less than 1, there is a negative correlation between the occurrence of {game} and {video}. The numerator is the likelihood of a customer purchasing both, while the denominator is what the likelihood would have been if the two purchases were completely independent. Such a negative correlation cannot be identified by a support-confidence framework.

The second correlation measure that we study is the $\chi 2$ measure, which was introduced in Chapter 3 (Equation 3.1). To compute the $\chi 2$ value, we take the squared difference between the observed and expected value for a slot (A and B pair) in the contingency table, divided by the expected value. This amount is summed for all slots of the contingency table. Let's perform a $\chi 2$ analysis of the above example.

Example 6.9 Correlation analysis using $\chi 2$. To compute the correlation using $\chi 2$ analysis for nominal data, we need the observed value and expected value (displayed in parenthesis) for each slot of the contingency table, as shown in Table 6.7. From
the table, we can compute the $\chi 2$ value as follows:

$\chi 2 =$ Σ(observed - expected)2
= (4,000 −4,500)2 + (3,500 −3,000)2 +
expected 4,500 3,000
(2,000 −1,500)2 (500 −1,000)2
+ = 555.6.
1,500 1,000

Because the $\chi 2$ value is greater than one, and the observed value of the slot (game, video) = 4,000, which is less than the expected value 4,500, buying game and buying video are negatively correlated. This is consistent with the conclusion derived from the analysis of the lift measure in Example 6.9.

A Comparison of Pattern Evaluation Measures

The above discussion shows that instead of using the simple support-confidence framework to evaluate frequent patterns, other measures, such as lift and $\chi 2$, often discloses more intrinsic pattern relationships. How effective are these measures? Should we also consider other alternatives?

Researchers have studied many pattern evaluation measures even before the start of in-depth research on scalable methods for mining frequent patterns. Recently, several other pattern evaluation measures have attracted interest. In this section, we present four such measures: all confidence, max confidence, Kulczynski, and cosine. We'll then compare their effectiveness with respect to one another and with respect to the lift and $\chi 2$ measures.

Given two itemsetsA and B, the all confidencemeasure of A and B is
defined as

$$\text{all\_conf}(A, B) = \frac{sup(A \cup B)}{max\{sup(A), sup(B)\}} = min\{P(A|B), P(B|A)\}, \quad (6.9)$$

where max{sup(A), sup(B)} is the maximum support of the itemsetsA and B. Thus all conf(A,B) is also the minimum confidence of the two association rules related to A and B, namely, "A⇒B" and "B⇒A".

Given two itemsetsA and B, the max confidence measure of A and B is defined as

max conf(A, B) = max{P(A|B),P(B|A)}. (6.10)

The max conf measure is the maximum confidence of the two association rules, “A⇒B” and “B⇒A”.

Given two itemsetsA and B, the Kulczynskimeasure of A and B (abbreviated as Kulc) is defined

as

Kulc $(A, B) = \frac{1}{2}(P(A|B) + P(B|A$)). (6.11)

It was proposed in 1927 by Polish mathematician S. Kulczynski. It can be viewed as an average of two confidence measures. That is, it is the average of two conditional probabilities: the probability of itemset B given itemset A, and the probability of itemset A given itemset B.

Finally, given two itemsetsA and B, the cosine measure of A and B is defined as

P(A ∪ B) sup(A ∪ B)

p

P(A) × P(B) psup(A) × sup(B) p cosine(A, B) =

= = P(A|B) × P(B|A).

(6.12) The cosine measure

can be viewed as a harmonized lift measure: the two formulae are similar except that for cosine, the square root is taken on the product of the probabilities of A and B. This is an important difference, however, because by taking the square root, the cosine value is only influenced by the supports of A, B, and A ∪ B, and not by the total number of transactions.

Each of the four measures defined above has the following property: Its value is only influenced by the supports of A, B, and A ∪ B, or more exactly, by the conditional probabilities of P(A|B) and P(B|A), but not by the total number of transactions. Another common property is that each measure ranges from 0 to 1, and the higher the value, the closer the relationship between A and B.

Now, together with lift and $\chi 2$, we have introduced in total six pattern evaluation measures. You may wonder, "Which is the best in assessing the discovered patten relationships?" To answer this question, we examine their performance on some typical data sets.

Table 6.8: A 2 × 2 contingency table for two items.

	milk	milk	Σrow
coffee	mc	mc	c
coffee	mc	mc	c
Σcol	m	m	Σ

Example 6.10 Comparison of six pattern evaluation measures on typical data sets. The relationships between the purchases of two items, milk and coffee, can be examined by

summarizing their purchase history in Table 6.8, a 2 × 2 contingency table, where

an entry such as mc represents the number of transactions containing both milk and coffee.

Table 6.9 shows a set of transactional data sets with their corresponding contingency tables and the associated values for each of the six evaluation measures. Let's first examine the first four data sets, D1 through D4. From the table, we see that m and c are positively associated in D1 and D2, negatively associated in D3, and neutral in D4. For D1 and D2, m and c are positively associated because mc (10,000) is considerably greater than mc (1,000) and mc (1,000). Intuitively, for people who bought milk (m = 10,000+1,000 = 11,000), it is very likely that they also bought coffee (mc/m = 10/11 = 91%), and vice versa. The results of the four newly introduced measures show that m and c are strongly Table 6.9: Comparison of six pattern evaluation measures using contingency tables for a variety of data sets.

Data Set mc mc mc mc $\chi 2$ lift all conf. max conf.

D1 10,000 1,000 1,000 100,000 90557 9.26 0.91 0.91 0.91 0.91 D2 10,000 1,000 1,000 100 0 1 0.91 0.91 0.91 0.91 D3 100 1,000 1,000 100,000

670 8.44 0.09 0.09 0.09 0.09

positively associated in both data sets by producing a measure value of 0.91. However, lift and $\chi 2$ generate dramatically different measure values for D1 and D2 due to their sensitivity to mc. In fact, in many real-world scenarios mc is usually huge and unstable. For example, in a market basket database, the total number of transactions could fluctuate on a daily basis and overwhelmingly exceed the number of transactions containing any particular itemset. Therefore, a good interestingness measure should not be affected by transactions that do not contain the itemsets of interest; otherwise, it would generate unstable results as illustrated in D1 and D2.

Similarly, in D3, the four new measures correctly show that m and c are strongly negatively associated because the ratio of m to c equals the ratio of mc to m, that is 100/1100 = 9.1%. However, lift and $\chi 2$ both contradict this in an incorrect way: their values for D2 are between those for D1 and D3.

For data set D4, both lift and $\chi 2$ indicate a highly positive association between m and c, whereas the others indicate a "neutral" association because the ratio of mc to mc equals the ratio of mc to mc, which is 1. This means that if a customer buys coffee (or milk), the probability that she will also purchase milk (or coffee) is exactly 50%.

"Why are lift and $\chi 2$ so poor at distinguishing pattern association relationships in the above transactional data sets?" To answer this, we have to consider the null- transactions. A null-transaction is a transaction that does not contain any of the itemsets being examined. In our example, mc represents the number of null- transactions. Lift and $\chi 2$ have difficulty distinguishing interesting pattern association relationships because they are both strongly influenced by mc. Typically, the number of null-transactions can outweigh the number of individual

 purchases, because many people may buy neither milk nor coffee. On the other hand, the other four measures are good indicators of interesting pattern associations because their definitions remove the influence of mc (that is, they are not influenced by the number of null-transactions).

The above discussion shows that it is highly desirable to have a measure whose value is independent of the number of null-transactions. A measure is null- invariant if its value is free from the influence of null-transactions. Nullinvariance is an important property for measuring association patterns in large transaction databases. Among the six discussed measures in this section, only lift and $\chi 2$ are not null-invariant measures.

"Among the all confidence, max confidence, Kulczynski, and cosine measures, which is best at indicating interesting pattern relationships?"

To answer this question, we introduce the imbalance ratio (IR), which assesses the imbalance of two itemsetsA and B in rule implications. It is defined as

$$IR(A,B) = \frac{|sup(A) - sup(B)|}{sup(A) + sup(B) - sup(A \cup B)}, \qquad (6.13)$$

where the numerator is the absolute value of the difference between the support of the itemsetsA and B, and the denominator is the number of transactions containing A or B. If the two directional implications between A and B are the same, IR(A,B) will be zero. Otherwise, the larger the difference between the two is, the larger the imbalance ratio is. This ratio is independent of the number of null-transactions and independent of the total number of transactions. Let's continue examining the remaining datasets of Example 6.10.

Example 6.11 Comparing null-invariant measures in pattern evaluation. Although the four measures introduced in this section are null-invariant, they may present dramatically different values on some subtly different datasets. Let's examine data sets D5 and D6 in Table 6.9, where the two events m and c have unbalanced conditional probabilities. That is, the ratio of mc to c is greater than 0.9. This means that knowing that c occurs should strongly suggest that m occurs also. The ratio of mc to mis less than 0.1. indicating that m implies that c is quite unlikely to occur. The all confidence and cosine measures view both cases as negatively associated and the Kulcmeasure views both as neutral. The max confidence measure claims strong positive associations for these cases. The measures give very diverse results!

"Which measure intuitively reflects the true relationship between the purchase of milk and coffee?" Due to the "balanced" skewness of the data, it is difficult to argue whether the two data sets have positive or negative association. From one point of view, only mc/(mc + mc) = 1,000/(1,000+ 10,000) = 9.09% of milkrelated transactions contain coffee in D5 and this percentage is 1,000/(1,000+ 100,000) = 0.99% in D6, both indicating a negative association. On the other hand, 90.9% of transactions in D5 (that is, mc/(mc+mc) = 1,000/(1,000+100)) and 9% in D6 (that is, 1,000/(1,000 + 10)) containing coffee contain milk as well, which indicates a positive association between milk and coffee. These draw very different conclusions. For such

"balanced" skewness, it could be fair to treat it as neutral, as Kulcdoes, and in the meantime indicate its skewness using the imbalance ratio (IR). According to Equation (6.13), for D4 we have IR(m,c) = 0, a perfectly balanced case; for D5, IR(m,c) = 0.89, a rather imbalanced case; whereas for D6, IR(m,c) = 0.99, a very skewed case. Therefore, the two measures, Kulcand IR, work together, presenting a clear picture for all three datasets, D4 through D6.

In summary, the use of only support and confidence measures to mine associations may generate a large number of rules, many of which can be uninteresting to users. Instead, we can augment the support-confidence framework with a pattern interestingness measure, which helps focus the mining towards rules with strong pattern relationships. The added measure substantially reduces the number of rules generated and leads to the discovery of more meaningful rules. Besides those introduced in this section, many other interestingness measures have been studied in the literature. Unfortunately, most of them do not have the null-invariance property. Because large data sets typically have many null-transactions, it is important to consider the null-invariance property when selecting appropriate interestingness measures for pattern evaluation. Among the four null-invariant measures studied here, namely all confidence, max confidence, Kulc, and cosine, we recommend using Kulcin conjunction with the imbalance ratio.

Summary

The discovery of frequent patterns, association, and correlation relationships among huge amounts of data is useful in selective marketing, decision analysis, and business management. A popular area of application is market basket analysis, which studies the buying habits of customers by searching for sets of items that are frequently purchased together (or in sequence).

Association rule mining consists of first finding frequent itemsets(set of items, such as A and B, satisfying a minimum support threshold, or percentageofthetask-relevanttuples), fromwhichstrongassociationrules in the form of $A \Rightarrow B$ are generated. These rules also satisfy a minimum confidence threshold (a prespecified probability of satisfying B under the condition that A is satisfied). Associations can be further analyzed to uncover correlation rules, which convey statistical correlations between itemsetsA and B.

Many efficient and scalable algorithms have been developed for frequent itemset mining, from which association and correlation rules can be derived. These algorithms can be classified into three categories: (1) Apriorilike algorithms, (2) frequent-pattern growth-based algorithms, such as FPgrowth, and (3) algorithms that use the vertical data format.

TheApriorialgorithmisaseminalalgorithmforminingfrequentitemsetsforBooleanassociationr
ules. Itexploresthelevel-wiseminingAprioriproperty that all nonempty subsets of a frequent itemset must also be frequent. At the kth iteration (for $k \geq 2$), it forms frequent k-itemset candidates based on the frequent $(k-1)$-itemsets, and scans the database once to find the complete set of frequent k-itemsets, Lk.

6.3. WHICH PATTERNS ARE INTERESTING?—PATTERN EVALUATION METHODS33 Variations involving hashing and transaction reduction can be used to make the procedure

more efficient. Other variations include partitioning the data (mining on each partition and

then combining the results) and sampling the data (mining on a subset of the data). These variations can reduce the number of data scans required to as little as two or one.

Frequent pattern growth (FP-growth) is a method of mining frequent itemsets without candidate generation. It constructs a highly compact data

structure(anFP-tree)tocompresstheoriginaltransactiondatabase. Rather than employing the generate-and-test strategy of Apriori-like methods, it focusesonfrequentpattern(fragment)growth,whichavoidscostlycandidate generation, resulting in greater efficiency.

Mining frequent itemsets using vertical data format (ECLAT) is a method that transforms a given data set of transactions in the horizontal data format of TID- itemset into the vertical data format of item-TID set. It mines the transformed data set by TID set intersections based on the Apriori property and additional optimization techniques, such as diffset.

Not all strong association rules are interesting. Therefore, the supportconfidence framework should be augmented with a pattern evaluation measure, which promotes the mining of interesting rules. A measure is nullinvariantif its value is free from the influence of null-transactions, i.e., the transactions that do not contain any of the itemsets being examined. Among many pattern evaluation measures, we examined lift, $\chi 2$, all confidence, max confidence, Kulczynski, and cosine, and showed that only the latter four are null-invariant. We suggest using the Kulczynski measure, together with the imbalance ratio, to present pattern relationships among itemsets.

Exercises

Suppose you have the set C of all frequent closed itemsets on a data set D, as well as the support count for each frequent closed itemset. Describe an algorithm to determine whether a given itemset X is frequent or not, and the support of X if it is frequent.

An itemset X is called a generator on a data set D if there does not exist a proper sub-itemset Y $\subset$ X such that support(X) = support(Y). A generator X is a frequent generator if support(X) passes the minimum support threshold. Let G be the set of all frequent generators on a data set D.

Can you determine whether an itemset A is frequent and the support of A, if it is frequent, using only G and the support counts of all frequent generators? If yes, present your algorithm. Otherwise, what other information is needed? Can you give an algorithm assuming the information needed is available?

What is the relationship between closed itemsets and generators?

The Apriori algorithm makes use of prior knowledge of subset support properties.

Prove that all nonempty subsets of a frequent itemset must also befrequent.

Prove that the support of any nonempty subset s' of itemset s must be at least as great as the support of s.

35

Given frequent itemset l and subset s of l, prove that the confidence of the rule "s′ ⇒ (l −s′)" cannot be more than the confidence of "s ⇒ (l −s)", where s′ is a subset of s.

A partitioning variation of Apriori subdivides the transactions of a database D into n nonoverlapping partitions. Prove that any itemset that is frequent in D must be frequent in at least one partition of D.

Let c be a candidate itemset in Ck generated by the Apriori algorithm. How many length-(k −1) subsets do we need to check in the prune step? According to your answer to the above question, can you give an improved version of procedure hasinfrequentsubsetin Figure 6.4?

Section 6.2.2 describes a method for generating association rules from frequent itemsets. Propose a more efficient method. Explain why it is more efficient than the one proposed there. (Hint: Consider incorporating the properties of Exercise 6.3(b) and 6.3(c) into your design.) 6. A database has 5 transactions. Let min sup = 60% and min conf = 80%.

		items bought
		{M, O, N, K, E,

Y}{D,O,N,K,E,Y}{M,A,K,E}{M,U,C,

		K, Y}
		{C, O, O, K, I, E}

Find all frequent itemsets using Apriori and FP-growth, respectively. Compare the efficiency of the two mining processes.

List all of the strong association rules (with support s and confidence c) matching the following metarule, where X is a variable representing customers, and itemidenotes variables representing items (e.g., “A”, “B”, etc.):

$\forall x \in$ transaction, buys(X,item1)$\wedge$buys(X,item2) $\Rightarrow$ buys(X,item3) [s,c]

(Implementation project) Implement three frequent itemset mining algorithms introduced in this chapter: (1) Apriori [AS94b], (2) FP-growth [HPY00], and (3) Eclat [Zak00] (mining using vertical data format), using a programming language that you are familiar with, such as C++ or Java. Compare the performance of each algorithm with various kinds of large data sets. Write a report to analyze the situations (such as data size, data distribution, minimal support threshold setting, and pattern density) where one algorithm may perform better than the others, and state why.

EXERCISES

A database has four transactions. Let min sup = 60% and min conf = 80%.

		items bought (in the form of brand-item category)
		{King's-Crab, Sunset-Milk, Dairyland-Cheese, Best-Bread} {Best-Cheese, Dairyland-Milk, Goldenfarm-Apple, Tasty-Pie, Wonder-Bread} {Westcoast-Apple, Dairyland-Milk, Wonder-Bread, Tasty-Pie} {Wonder-Bread, Sunset-Milk, Dairyland-Cheese}

At the granularity of item category (e.g., itemicould be "Milk"), for the following rule template,

$\forall X \in$ transaction, buys(X,item1)Abuys(X,item2) $\Rightarrow$ buys(X,item3) [s,c] list the frequent k-itemset for the largest k, and all of the strong association rules (with their support s and confidence c) containing the frequent k-itemset for the largest k.

At the granularity of brand-item category (e.g., itemicould be "Sunset- Milk"), for the following rule template,

$\forall X \in$ customer, buys(X,item1)Abuys(X,item2) $\Rightarrow$ buys(X,item3)

list the frequent k-itemset for the largest k (but do not print any rules).

Suppose that a large store has a transaction database that is distributed among four locations. Transactions in each component database have the same format, namely Tj:{i1,...,im}, where Tjis a transaction identifier, and ik($1 \leq k \leq m$) is the identifier of an item purchased in the transaction. Propose an efficient algorithm to mine global association rules (without

considering multilevel associations). You may present your algorithm in the form of an outline. Your algorithm should not require shipping all of the data to one site and should not cause excessive network communication overhead.

Suppose that frequent itemsets are saved for a large transaction database,DB. Discuss how to efficiently mine the (global) association rules under the same minimum support threshold, if a set of new transactions, denoted as ΔDB, is (incrementally) added in?

Most frequent pattern mining algorithms consider only distinct items in atransaction. However, multiple occurrences of an item in the same shopping basket, such as four cakes and three jugs of milk, can be important in transaction data analysis. How can one mine frequent itemsets efficiently considering multiple occurrences of items? Propose modifications to the well-known algorithms, such as Apriori and FP-growth, to adapt to such a situation.

(Implementation project) Many techniques have been proposed to further improve the performance of frequent-itemset mining algorithms. Taking FP- tree-based frequent pattern-growth algorithms (such as FPgrowth) as an example, implement one of the following optimization techniques. Compare the performance of your new implementation with the unoptimized version.

37

The frequent pattern mining method of Section 6.2.4) uses an FP-treeto generate conditional pattern bases using a bottom-up projection technique, i.e., project on the prefix path of an item p. However, one can develop a top-down projection technique, that is, project on the suffix path of an item p in the generation of a conditional pattern-base. Design and implement such a top- down FP-tree mining method and compare your performance with the bottom- up projection method.

Nodes and pointers are used uniformly in an FP-tree in the design of the FP- growth algorithm. However, such a structure may consume a lot of space when the data are sparse. One possible alternative design is to explore array-and pointer- based hybrid implementation, where a node may store multiple items when it contains no splitting point to multiple sub-branches. Develop such an implementation and compare it with the original one.

It is time- and space- consuming to generate numerous conditionalpattern bases during pattern-growth mining. An interesting alternative is to push right the branches that have been mined for a particular item p, that is, to push them to the remaining branch(es) of the FP-tree. This is done so that fewer conditional pattern bases have to be generated and additional sharing can be explored when mining the remaining branches of the FP-tree. Design and implement such a method and conduct a performance study on it.

Give a short example to show that items in a strong association rule mayactually be negatively correlated.

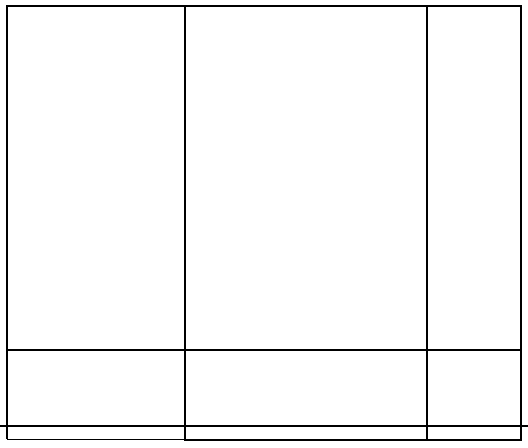
The following contingency table summarizes supermarket transaction data,

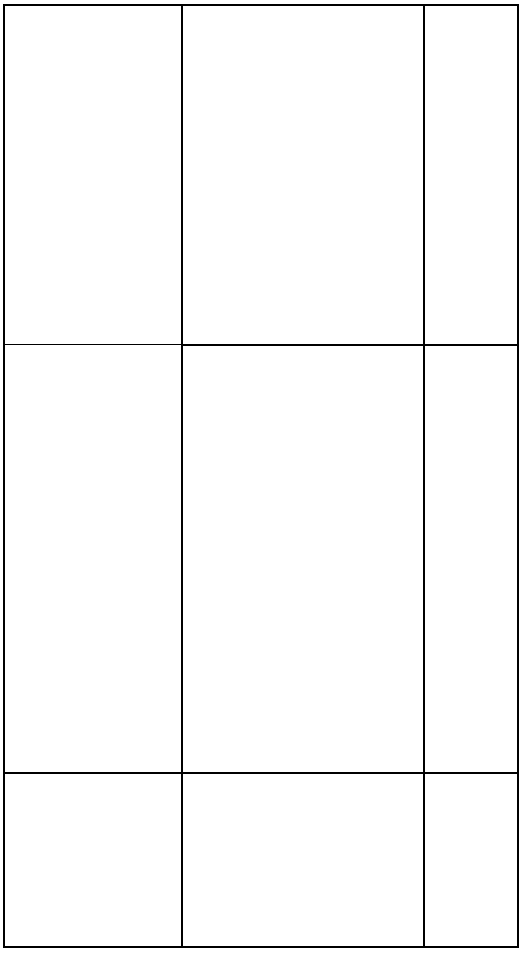
_____

where hot dogs refers to the transactions containing hot dogs, hotdogs refers to the transactions that do not contain hot dogs, hamburgers refers

_____

to the transactions containing hamburgers, and hamburgers refers to the transactions that do not contain hamburgers.





BIBLIOGRAPHIC NOTES

Suppose that the association rule “hot dogs ⇒ hamburgers” is mined. Given a minimum support threshold of 25% and a minimum confidence threshold of 50%, is this association rule strong?

Based on the given data, is the purchase of hot dogs independent of the purchase of hamburgers? If not, what kind of correlation relationship exists between the two?

Compare the use of the all confidence, max confidence, Kulczynski, and cosine measures with lift and correlation on the given data.

(Implementation project) The DBLP dataset (http://www.informatik.unitrier.de/~ley/db/) consists of over one million entries of research papers published in computer science conferences and journals. Among these entries, there are a good number of authors that have coauthor relationships.

Propose a method that can mine efficiently a set of coauthor relationships that are closely correlated (e.g., often coauthoring papers together).

Based on the mining results and the pattern evaluation measuresdiscussed in this chapter, discuss which measure may convincingly uncover close collaboration patterns better than others.

Based on the study above, can you develop a method that can roughly predict advisor and advisee relationships and the approximate period for such advisory supervision?

Bibliographic Notes

Association rule mining was first proposed by Agrawal, Imielinski, and Swami [AIS93]. The Apriori algorithm discussed in Section 6.2.1 for frequent itemset mining was presented in Agrawal and Srikant [AS94b]. A variation of the algorithm using a similar pruning heuristic was developed independently by Mannila, Tiovonen, and Verkamo [MTV94]. A joint publication combining these works later appeared in Agrawal, Mannila, Srikant, Toivonen, and Verkamo [AMS+96]. A method for generating association rules from frequent itemsets is described in Agrawal and Srikant [AS94a].

References for the variations of Apriori described in Section 6.2.3 include the following. The use of hash tables to improve association mining efficiency was studied by Park, Chen, and Yu [PCY95a]. The partitioning technique was proposed by Savasere, Omiecinski, and Navathe [SON95]. The sampling approach is discussed in Toivonen [Toi96]. A dynamic itemset counting approach is given in Brin, Motwani, Ullman, and Tsur [BMUT97]. An efficient incremental updating of mined association rules was proposed by Cheung, Han, Ng, and Wong [CHNW96]. Parallel and distributed association data mining under the Apriori framework was studied by Park, Chen, and Yu [PCY95b], Agrawal and Shafer [AS96], and Cheung, Han, Ng, et al. [CHN+96]. Another parallel association mining method, which explores itemset clustering using a vertical database layout, was proposed in Zaki, Parthasarathy, Ogihara, and Li [ZPOL97].

Other scalable frequent itemset mining methods have been proposed as alternatives to the Apriori-based approach. FP-growth, a pattern-growth approach for mining frequent itemsets without candidate generation, was proposed by Han, Pei, and Yin [HPY00] (Section 6.2.4). An exploration of hyper-structure mining of frequent patterns, called H-Mine, was proposed by Pei, Han, Lu, et al. [PHMA+01]. A method that integrates top-down and bottom-up traversal of FP-trees in pattern-growth mining, was proposed by Liu, Pan, Wang, and Han [LPWH02]. An array-based implementation of prefix-tree- structure for efficient pattern growth mining was proposed by Grahne and Zhu [GZ03]. Eclat, an approach for mining frequent itemsets by exploring the vertical data format, was proposed by Zaki [Zak00]. A depth-first generation of frequent itemsets by a tree projection technique was proposed by Agarwal, Aggarwal, and Prasad [AAP01]. An integration of association mining with relational database systems was studied by Sarawagi, Thomas, and Agrawal [?].

The mining of frequent closed itemsets was proposed in Pasquier, Bastide, Taouil, and Lakhal [?], where an Apriori-based algorithm called A-Close for such mining was presented. CLOSET, an efficient closed itemset mining algorithm based on the frequent pattern growth method, was proposed by Pei, Han, and Mao [?]. CHARM by Zaki and Hsiao [?] develops a compact vertical TID list structure called diffsetwhich only records the difference in the TID list of a candidate pattern from its prefix pattern. A fast hash- based approach is also used in CHARM to prune non-closed patterns. CLOSET+ by Wang, Han and Pei [?] integrates previously proposed effective strategies as well as newly developed techniques such as hybrid tree-projection and item skipping. AFOPT, a method that explores a right push operation on FP-trees during the mining process, was proposed by Liu, Lu, Lou and Yu [?]. A prefix tree-based algorithm integrated with array representation, called FPClose, for mining closed itemsets using pattern-growth approach, was proposed by Grahne and Zhu [GZ03]. Pan, Cong, Tung, et al. [?] proposed CARPENTER, a method for finding closed patterns in long biological datasets, which integrates the advantages of vertical data formats and pattern growth methods. Mining max-patterns was first studied by Bayardo [?], where MaxMiner, an Apriori-based, level- wise, breadth-first search method was proposed to find max-itemset by performing superset frequency pruning and subset infrequency pruning for search space reduction. Another efficient method MAFIA, proposed by Burdick, Calimlim, and Gehrke [?], uses vertical bitmap to compress transaction id list, thus improving the counting efficiency. A FIMI (Frequent Itemset Mining Implementation) workshop dedicated to the implementation methods of frequent itemset mining was reported by Goethals and Zaki [?].

The problem of mining interesting rules has been studied by many researchers. The statistical independence of rules in data mining was studied by PiatetskiShapiro [PS91]. The interestingness problem of strong association rules is discussed in Chen, Han, and Yu [CHY96], Brin, Motwani, and Silverstein [BMS97],

and Aggarwal and Yu [AY99], which cover several interestingness measures including lift. An

efficient method for generalizing associations to correlations is given in Brin, Motwani, and Silverstein [BMS97]. Other alternatives to the support-confidence

framework for assessing the interestingness of association rules are proposed in Brin, Motwani, Ullman, and Tsur [BMUT97] and Ahmed, El-Makky, and Taha [AEMT00]. A method for mining strong gradient relationships among itemsets was proposed by Imielinski, Khachiyan, and Abdulghani [IKA02]. Silverstein, Brin, Motwani, and Ullman [SBMU98] studied the problem of mining causal structures over transaction databases. Some comparative studies of different interestingness measures were done by Hilderman and Hamilton [HH01]. The notion of null transaction invariance was introduced, together with a comparative analysis of interestingness measures, by Tan, Kumar and Srivastava [TKS02]. The use of all confidence as a correlation measure for generating interesting association rules was studied by Omiecinski [Omi03] and by Lee, Kim, Cai and Han [LKCH03]. Wu, Chen and Han [WCH10] introduced the Kulczynski measure for associative patterns and performed a comparative analysis of a set of measures for pattern evaluatio

Bibliography

[AAP01] R. Agarwal, C. C. Aggarwal, and V. V. V. Prasad. A tree projection algorithm for generation of frequent itemsets. J. Parallel and Distributed Computing, 61:350–371, 2001.

[AEMT00] K. M. Ahmed, N. M. El-Makky, and Y. Taha. A note on "beyond market basket: Generalizing association rules to correlations". SIGKDD Explorations, 1:46–48, 2000.

[AIS93] R. Agrawal, T. Imielinski, and A. Swami. Mining association rules between sets of items in large databases. In Proc. 1993 ACMSIGMOD Int. Conf. Management of Data (SIGMOD'93), pages 207–216, Washington, DC, May 1993.

[AMS+96] R. Agrawal, M. Mehta, J. Shafer, R. Srikant, A. Arning, and T. Bollinger. The Quest data mining system. In Proc. 1996 Int. Conf. Data Mining and Knowledge Discovery (KDD'96), pages 244–249, Portland, OR, Aug. 1996.

[AS94a] R. Agrawal and R. Srikant. Fast algorithm for mining association rules in large databases. In Research Report RJ 9839, IBM Almaden Research Center, San Jose, CA, June 1994.

[AS94b] R. Agrawal and R. Srikant. Fast algorithms for mining association rules. In Proc. 1994 Int. Conf. Very Large Data Bases (VLDB'94), pages 487–499, Santiago, Chile, Sept. 1994.

[AS96] R. Agrawal and J. C. Shafer. Parallel mining of association rules: Design, implementation, and experience.

IEEE Trans. Knowledge and Data Engineering, 8:962–969, 1996.

[AY99] C. C. Aggarwal and P. S. Yu. A new framework for itemset generation. In Proc. 1998 ACM Symp. Principles of Database Systems (PODS'98), pages 18–24, Seattle, WA, June 1999.

[BMS97] S. Brin, R. Motwani, and C. Silverstein. Beyond market basket: Generalizing association rules to correlations. In Proc. 1997 ACMSIGMOD Int. Conf. Management of Data (SIGMOD'97), pages 265–276, Tucson, AZ, May 1997.

[BMUT97] S. Brin, R. Motwani, J. D. Ullman, and S. Tsur. Dynamic itemset counting and implication rules for market basket analysis. In Proc. 1997 ACM-SIGMOD Int. Conf. Management of Data (SIGMOD'97), pages 255–264, Tucson, AZ, May 1997.

[CHN+96] D. W. Cheung, J. Han, V. Ng, A. Fu, and Y. Fu. A fast distributed algorithm for mining association rules. In Proc. 1996 Int. Conf. Parallel and Distributed Information Systems, pages 31–44, Miami Beach, FL, Dec. 1996.

[CHNW96] D. W. Cheung, J. Han, V. Ng, and C. Y. Wong. Maintenance of discovered association rules in large databases: An incremental updating technique. In Proc. 1996 Int. Conf. Data Engineering (ICDE'96), pages 106–114, New Orleans, LA, Feb. 1996.

[CHY96] M. S. Chen, J. Han, and P. S. Yu. Data mining: An overview from a database perspective. IEEE Trans.

Knowledge and Data Engineering, 8:866–883, 1996.

[GZ03] G. Grahne and J. Zhu. Efficiently using prefix-trees in mining frequent itemsets. In Proc. ICDM'03 Int.
Workshop on Frequent Itemset Mining Implementations (FIMI'03), Melbourne, FL, Nov. 2003.
[HH01] R. J. Hilderman and H. J. Hamilton. Knowledge Discovery and Measures of Interest. Kluwer Academic, 2001. [HPY00] J. Han, J. Pei, and Y. Yin. Mining frequent patterns without candidate generation. In Proc. 2000 ACM-
SIGMOD Int. Conf. Management of Data (SIGMOD'00), pages 1–12, Dallas, TX, May 2000.
[IKA02] T. Imielinski, L. Khachiyan, and A. Abdulghani. Cubegrades: Generalizing association rules. Data Mining and Knowledge Discovery, 6:219–258, 2002.
[LKCH03] Y.-K. Lee, W.-Y. Kim, Y. D. Cai, and J. Han. CoMine: Efficient mining of correlated patterns. In Proc. 2003 Int.
Conf. Data Mining (ICDM'03), pages 581–584, Melbourne, FL, Nov. 2003.
[LPWH02] J. Liu, Y. Pan, K. Wang, and J. Han. Mining frequent item sets by opportunistic projection. In Proc. 2002 ACM SIGKDD Int. Conf. Knowledge Discovery in Databases (KDD'02), pages 239– 248, Edmonton, Canada, July 2002.
[MTV94] H. Mannila, H. Toivonen, and A. I. Verkamo. Efficient algorithms for discovering association rules. In Proc.
AAAI'94 Workshop Knowledge Discovery in Databases (KDD'94), pages 181–192, Seattle, WA, July 1994.

[Omi03] E. Omiecinski. Alternative interest measures for mining associations. IEEE Trans. Knowledge and Data Engineering, 15:57–69, 2003.
[PCY95a] J. S. Park, M. S. Chen, and P. S. Yu. An effective hash-based algorithm for mining association rules. In Proc. 1995 ACM-SIGMOD Int. Conf. Management of Data (SIGMOD'95), pages 175–186, San Jose, CA, May 1995.
[PCY95b] J. S. Park, M. S. Chen, and P. S. Yu. Efficient parallel mining for association rules. In Proc. 4th Int. Conf.
Information and Knowledge Management, pages 31–36, Baltimore, MD, Nov. 1995.
[PHMA+01] J. Pei, J. Han, B. Mortazavi-Asl, H. Pinto, Q. Chen, U. Dayal, and M.-C. Hsu. PrefixSpan: Mining sequential patterns efficiently by prefix-projected pattern growth. In Proc. 2001 Int. Conf. Data Engineering (ICDE'01), pages 215–224, Heidelberg, Germany, April 2001.
[PS91] G. Piatetsky-Shapiro. Notes of AAAI'91 Workshop Knowledge Discovery in Databases (KDD'91). Anaheim, CA, July 1991.
[SON95] A. Savasere, E. Omiecinski, and S. Navathe. An efficient algorithm for mining association rules in large databases. In Proc. 1995 Int. Conf. Very Large Data Bases (VLDB'95), pages 432–443, Zurich, Switzerland, Sept. 1995.
[TKS02] P.-N. Tan, V. Kumar, and J. Srivastava. Selecting the right interestingness measure

for association patterns. In Proc. 2002 ACM SIGKDD Int. Conf. Knowledge Discovery in Databases (KDD'02), pages 32–41, Edmonton, Canada, July 2002.

[Toi96] H. Toivonen. Sampling large databases for association rules. In Proc. 1996 Int. Conf. Very Large Data Bases (VLDB'96), pages 134–145, Bombay, India, Sept. 1996.

[WCH10] T. Wu, Y. Chen, and J. Han. Re-examination of interestingness measures in pattern mining: A unified framework. Data Mining and Knowledge Discovery, 18, 2010.

[Zak00] M. J. Zaki. Scalable algorithms for association mining. IEEE Trans. Knowledge and Data Engineering, 12:372–390, 2000.

[ZPOL97] M. J. Zaki, S. Parthasarathy, M. Ogihara, and W. Li. Parallel algorithm for discovery of association rules.
Data Mining and Knowledge Discovery, 1:343–374, 1997.

UNIT IV

ClassificationandPrediction:

- Classification and prediction are two forms of data analysis that can be used to extractmodelsdescribingimportant dataclassesor to predict futuredata trends.
- Classificationpredictscategorical(discrete,unordered)labels,predictionmodelscontinuousvalue dfunctions.
- Forexample, we can build a classification model to categorize bankloan applicationsaseither safe orrisky,or a prediction model to predict the expendituresof potential customersoncomputer equipmentgiven their income andoccupation.
- A predictor is constructed that predicts a continuous-valued function, or ordered value, asopposedto a categorical label.
- Regressionanalysisisastatisticalmethodologythatismostoftenusedfornumericprediction.
- Many classification and prediction methods have been proposed by researchers in machinelearning,patternrecognition, and statistics.
- Most algorithms are memory resident, typically assuming a small data size. Recent datamining research has built on such work, developing scalable classification and predictiontechniquescapable ofhandlinglargedisk-residentdata.

IssuesRegardingClassificationandPrediction:

PreparingtheDataforClassificationandPrediction: Thefollowingpreprocessingstepsmaybe appliedtothedatatohelp improvethe accuracy,efficiency,and scalabilityoftheclassification orpredictionprocess.

Datacleaning:

- This refers to the preprocessing of data in order to remove or reduce noise (by applyingsmoothing techniques) and the treatment of missingvalues(e.g., by replacing a missingvalue with the most commonly occurring value for that attribute, or with the most probablevaluebased on statistics).
- Although mostclassification algorithmshave some mechanisms for handling noisy ormissingdata, this stepcanhelp reduceconfusionduringlearning.

Relevanceanalysis:

- Manyof theattributesinthe datamayberedundant. Correlationanalysiscanbeusedtoidentifywhetheranytwogivenattributesarestatisticallyrelate

d.

- For example, a strong correlation between attributes A1 and A2 would suggest that one ofthetwo could be removed from furtheranalysis.
- Adatabasemayalsocontainirrelevantattributes.Attributesubsetselectioncanbeusedin these cases to find a reduced set of attributes such that the resultingprobabilitydistribution of the data classes is as close as possible to the original distribution obtainedusingallattributes.
- Hence,relevanceanalysis,intheformofcorrelationanalysisandattributesubsetselection, can be used to detect attributes that do not contribute to the classification orpredictiontask.
- Suchanalysiscanhelpimproveclassificationefficiencyandscalability.

DataTransformationAndReduction

- Thedata maybetransformed bynormalization, particularlywhen neural networksormethodsinvolvingdistancemeasurementsareused inthelearningstep. Normalizationinvolvesscalingallvaluesfor a givenattributesothattheyfallwithinasmallspecified range, such as-1 to +1 or0 to 1. The data can also be transformed by generalizing it to higher-level concepts.
- Conceptthierarchies may be used for this purpose. This is particularly useful for continuousvaluedattributes.

For example, numeric values for the attribute income can be generalized to discreteranges, such as low, medium, and high. Similarly, categorical attributes, like street, canbegeneralized to higher-level concepts, like city.

- Data can also be reduced by applying many other methods, ranging from wavelettransformationandprinciplecomponentsanalysis todiscretizationtechniques,suchasbinning, histogram analysis, andclustering.

ComparingClassificationandPredictionMethods:

Accuracy:

- The accuracy of a classifier refers to the ability of a given classifier to correctly predicttheclasslabelofneworpreviouslyunseendata(i.e.,tupleswithoutclasslabelinformati on).
- The accuracy of a predictor refers to how well a given predictor can guess the value ofthepredictedattributefor neworpreviouslyunseendata.

Speed:

- Thisreferstothe computationalcosts involvedingeneratingandusingthegiven classifier orpredictor.

Robustness:

- Thisisthe abilityof theclassifier orpredictorto makecorrect predictionsgivennoisydata or datawithmissingvalues.

Scalability:

- This refers to the ability to construct the classifier or predictor efficientlygivenlargeamountsof data.

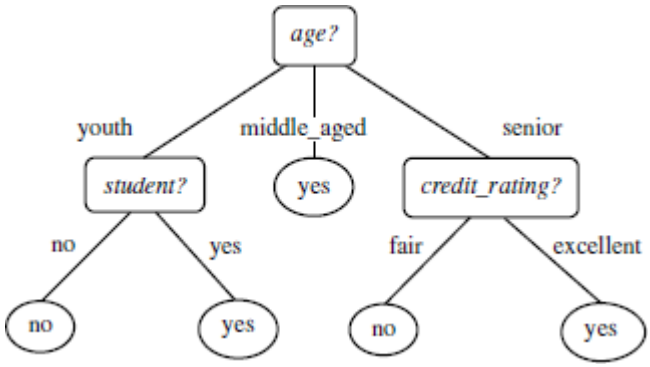
Interpretability:

- This refers to the level of understanding and insight that is providedby the classifier orpredictor.
- Interpretabilityissubjectiveandthereforemoredifficulttoassess.

ClassificationbyDecisionTreeInduction:

Decisiontreeinductionisthelearningofdecisiontreesfromclass- labeledtrainingtuples.Adecision treeis a flowchart-liketreestructure,where

- Eachinternalnodedenotesatestonan attribute.
- Eachbranch representsan outcomeof thetest.
- Eachleafnodeholdsa class label.
- Thetopmost nodeinatreeis therootnode.



- Theconstructionofdecisiontreeclassifiersdoesnotrequireanydomainknowledgeorparamete rsetting,andthereforeIappropriatefor exploratoryknowledgediscovery.
- Decisiontrees canhandlehighdimensionaldata. Theirrepresentationofacquiredknowledgeintreeformisintuitiveandgenerallyeasytoassimilate byhumans. Thelearningandclassificationstepsofdecisiontreeinductionaresimpleandfast.Ingener al, decisiontreeclassifiers have goodaccuracy.
- Decision tree induction algorithmshave been used for classification in many applicationareas, such as medicine,manufacturing and production, financial analysis, astronomy, andmolecularbiology.

AlgorithmForDecisionTreeInduction:

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

Input:

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

Output: A decision tree.

Method:

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

Thealgorithmiscalledwiththreeparameters:
Datapartition
Attributelist
Attributeselectionmethod

- Theparameter attributelistisalistofattributesdescribingthetuples. Attributeselectionmethodspecifiesaheuristicprocedureforselectingtheattributethat
- —best‖discriminates thegiven tuples accordingtoclass. Thetreestartsasasingle node, N,representingthetrainingtuplesinD.
-

- If the tuples in D are all of the same class, then node N becomes a leaf and is labeledwiththat class.
- All of the terminating conditions are explained at the end of the algorithm.Otherwise,thealgorithmcallsAttributeselectionmethodtodeterminethesplit tingcriterion.
- The splittingcriterion tellsuswhich attribute to test at nodeNbydeterminingthe―best‖ waytoseparateorpartitionthetuplesin Dintoindividualclasses.

Therearethreepossiblescenarios.LetAbethesplittingattribute.Ahasvdistinctvalues, {a1,a2,…,av},basedonthetrainingdata.

Aisdiscrete-valued:

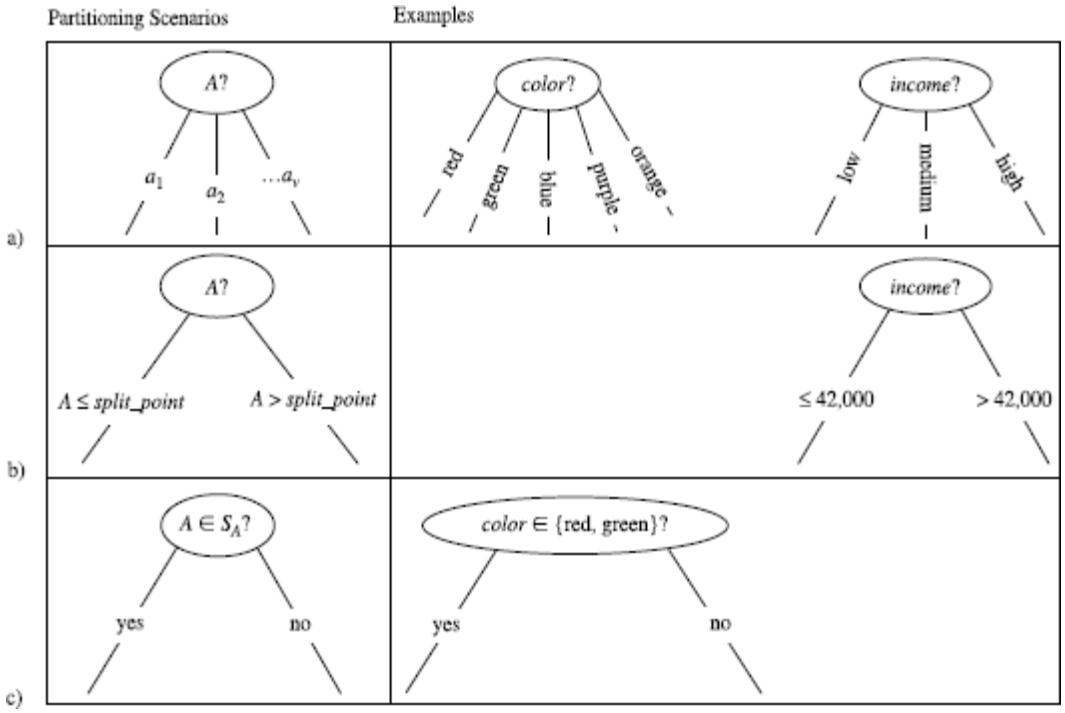
- Inthiscase,theoutcomesofthetestat nodeNcorresponddirectlyto theknownvaluesofA. Abranchiscreatedforeachknownvalue,aj,ofAandlabeled withthatvalue.Aneednot beconsideredin anyfuturepartitioningof thetuples.

Aiscontinuous-valued:

Inthiscase,thetestatnodeNhastwopossibleoutcomes,correspondingtotheconditionsA<=split point and A>split point, respectively
wheresplitpointisthesplit-pointreturnedbyAttributeselectionmethodas partofthesplittingcriterion.

Aisdiscrete-valuedandabinarytreemustbe produced:

Thetest at node Nis of theform―A€SA?‖.
SA is the splitting subset for A, returned by Attribute selection methodas part of the splittingcriterion. It isasubsetof the known values ofA.



If A is Discrete valued (b)If A is continuous valued (c) IfA is discrete-valued and a binarytreemustbeproduced:

BayesianClassification:

- Bayesianclassifiersarestatisticalclassifiers. Theycanpredictclassmembership probabilities,suchasthe probabilitythata
- giventuplebelongstoa particular class.
- BayesianclassificationisbasedonBayes'theorem.

Bayes'Theorem:

- Let Xbeadata tuple.InBayesian terms, Xis considered—evidence.and it is described by measurementsmadeon aset of nattributes.

LetHbesomehypothesis,suchasthatthedatatupleXbelongstoaspecifiedclass C.
Forclassificationproblems,wewanttodetermineP(H|X),theprobabilitythatthehypothesisHholds

given the―evidence‖or observed datatuple X.

P(H|X) is the posterior probability, or a posteriori probability, of H conditioned on X.Bayes'theoremisusefulinthat

itprovidesawayofcalculatingtheposteriorprobability,P(H|X),fromP(H),P(X|H), and P(X).

$$P(H|X) = \frac{P(X|H)P(H)}{P(X)}.$$

NaïveBayesianClassification:

ThenaïveBayesianclassifier,orsimpleBayesian classifier,worksasfollows:

Let D be a training set of tuples and their associated class labels. As usual, each tuple isrepresentedbyann-

dimensionalattributevector,X=(x1,x2,…,xn),depictingnmeasurementsmadeonthe tuplefromn attributes,respectively,A1,A2, …,An.

Supposethattherearem classes,C1,C2,…,Cm.Givenatuple,X,theclassifierwill predictthatXbelongstotheclasshavingthehighestposteriorprobability,conditionedonX.

That is, the naïve Bayesian classifier predicts that tuple X belongs to the class Ci if and onlyif

$$P(C_i|X) > P(C_j|X) \quad \text{for } 1 \le j \le m, j \ne i.$$

ThuswemaximizeP(CijX).TheclassCiforwhichP(CijX)ismaximizediscalledthe maximumposteriorihypothesis.ByBayes'theorem

$$P(C_i|X) = \frac{P(X|C_i)P(C_i)}{P(X)}.$$

As P(X) is constant for all classes, only P(X|Ci)P(Ci) need be maximized. If the classprior probabilities are not known, then it is commonly assumed that the classes are equallylikely, that is, P(C1) = P(C2) = …= P(Cm), and we would therefore maximize P(X|Ci).Otherwise,wemaximizeP(X|Ci)P(Ci).

4. Given data sets with many attributes, it would be extremely computationally expensivetocompute P(X|Ci). In order to reduce computation in evaluating P(X|Ci), the naive assumptionof class conditional independence is made. This presumes that the values of the attributesareconditionallyindependentofoneanother,giventheclasslabelofthetuple.Thus,

$$P(X|C_i) = \prod_{k=1}^{n} P(x_k|C_i)$$
$$= P(x_1|C_i) \times P(x_2|C_i) \times \cdots \times P(x_n|C_i).$$

WecaneasilyestimatetheprobabilitiesP(x1|Ci),P(x2|Ci),:::,P(xn|Ci)fromthetrainingtuples.Fore achattribute,welookatwhethertheattributeiscategoricalorcontinuous-valued.Forinstance, tocomputeP(X|Ci), weconsider thefollowing:

IfAkiscategorical,thenP(xk|Ci)isthenumberoftuplesofclassCiinDhavingthevalue xkforAk, divided by|Ci,D|the number oftuplesofclass Ciin D.

If Akis continuous-valued, then we need to do a bit more work, but the calculationis prettystraightforward.

A continuous-valued attribute is typically assumed tohave a Gaussian distribution with ameanμand standard deviation , defined by

$$g(x, \mu, \sigma) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(x-\mu)^2}{2\sigma^2}},$$

$$P(x_k|C_i) = g(x_k, \mu_{C_i}, \sigma_{C_i}).$$

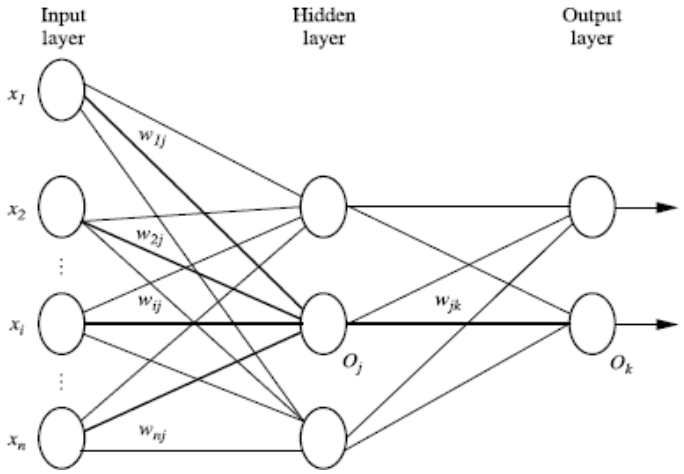
5. Inordertopredictthe classlabelof X,P(XjCi)P(Ci)isevaluatedforeachclass Ci.Theclassifier predicts thattheclass label oftupleXistheclassCiif andonlyif

$$P(X|C_i)P(C_i) > P(X|C_j)P(C_j) \quad \text{for } 1 \le j \le m, j \ne i.$$

AMultilayerFeed-ForwardNeuralNetwork:

- The backpropagation algorithm performs learning on a multilayer feed- forward neuralnetwork.
- Ititeratively learnsasetofweightsforpredictionoftheclasslabeloftuples.A multilayer feed-forward neural network consists of an input layer, one ormorehiddenlayers,andanoutput layer.

Example:



- The inputs to the network correspond to the attributes measured for each training tuple. Theinputs are fed simultaneously into the units making up the input layer. These inputs passthrough the input layer and are then weighted and fed simultaneously to a second layerknownasahidden layer.
- The outputs of the hidden layer units can be input to another hidden layer, and so on. Thenumberof hidden layersis arbitrary.
- Theweightedoutputsofthelasthiddenlayerareinputtounitsmakinguptheoutputlayer, whichemitsthenetwork'spredictionfor giventuples

3.4.1 ClassificationbyBackpropagation:

- Backpropagationisaneural networklearningalgorithm.
- Aneuralnetworkisasetofconnectedinput/outputunitsinwhicheachconnectionhasaweightassoc
- iatedwith it.
- Duringthelearningphase,thenetworklearnsbyadjustingtheweightssoastobeabletopredictthe correct classlabel of theinput tuples.
- Neuralnetworklearningisalsoreferredtoasconnectionistlearningduetotheconnectionsbetween units.

- Neural networks involve long training times and are therefore more suitable forapplicationswherethisis feasible.
- Backpropagationlearnsbyiterativelyprocessingadatasetoftrainingtuples,comparing thenetwork'spredictionforeachtuplewiththeactualknown targetvalue.
- The target value may be the known class label of the training tuple (for classificationproblems)oracontinuous value (forprediction).
- For each training tuple, the weights are modified so as to minimize the mean squarederrorbetweenthe network's prediction andthe actual target value. Thesemodificationsaremadeinthe―backwards‖direction,thatis,fromtheoutputlayer,throughea chhidden layerdown to thefirsthidden layer hencethenameisbackpropagation.
- Although it is not guaranteed, in general the weights will eventually converge, and thelearningprocessstops.

Advantages:

- It include their high tolerance of noisy data as well as their ability to classify patterns onwhichtheyhavenot beentrained.
- They can be used when you may have little knowledge of the relationships betweenattributesandclasses.
- They are well-suited for continuous-valued inputs and outputs, unlike most decision treealgorithms.
- They have been successful on a wide array of real-world data, including handwrittencharacter recognition, pathology and laboratory medicine, and training a computer topronounceEnglishtext.
- Neural network algorithms are inherently parallel; parallelization techniques can be usedtospeed up the computationprocess.

Process:

Initializethe weights:

Theweightsinthenetwork areinitializedtosmallrandom numbers

ranging from-1.0 to 1.0, or -0.5 to 0.5. Each unit has a bias associated with it. The biases aresimilarlyinitialized to small random numbers.

Eachtrainingtuple,X,isprocessedbythefollowingsteps.

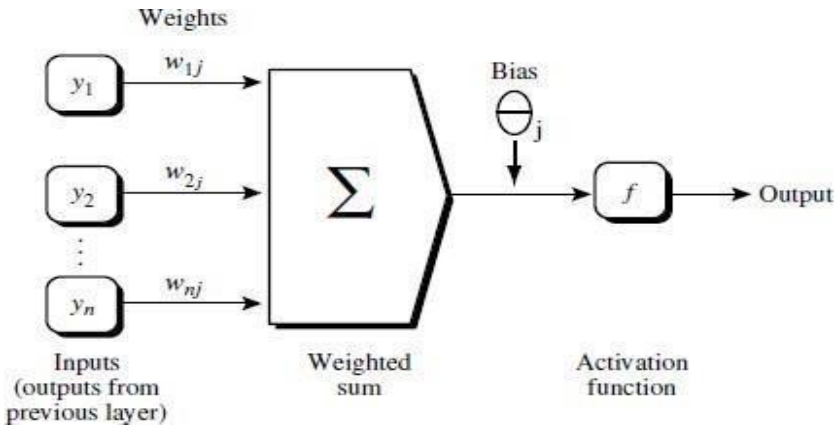
Propagatetheinputsforward:

First, the training tuple is fed to the input layer of thenetwork. The inputs pass through the inputunits, unchanged. That is, for an input unitj, its output, Oj, is equal to its input value, Ij. Next, thenet input and output of eachunit in the hidden and output layers are computed. The net input to aunitin thehiddenoroutputlayers is computed asalinear combination ofitsinputs.
Eachsuchunithasanumberofinputstoitthat are,infact, theoutputsoftheunitsconnectedto itin theprevious layer. Each connection has a weight. To compute the net input to the unit, eachinputconnected tothe unit ismultipliedbyitscorrespondingweight,and thisissummed.

$$I_j = \sum_i w_{ij} O_i + \theta_j,$$

wherewi,jis the weight of the connection from unit iin the previous layer to unit j;Oiistheoutput of unit ifrom theprevious layer
ϴjisthebiasoftheunit&itactsasathresholdinthatitservestovarytheactivityoftheunit.

Eachunitinthehidden andoutputlayerstakesitsnetinputandthenappliesanactivationfunctionto it.



Backpropagatetheerror:

Theerror ispropagated backwardbyupdatingthe weights andbiasestoreflect theerrorof thenetwork's prediction.For aunitjin theoutput layer, theerror Errjis computedby

$$Err_j = O_j(1 - O_j)(T_j - O_j)$$

whereOjistheactualoutputofunit j,andTjistheknowntarget valueofthe giventrainingtuple.
Theerror ofahidden layerunitj is

$$Err_j = O_j(1 - O_j)\sum_k Err_k w_{jk}$$

wherewjkistheweightof theconnectionfromunitjtoaunitkinthenexthigherlayer,andErrkis theerror ofunit k.
Weightsareupdatedbythefollowingequations,where Dwi jisthechangein weight wij:

$$\Delta w_{ij} = (l) Err_j O_i$$
$$w_{ij} = w_{ij} + \Delta w_{ij}$$

Biasesareupdated bythefollowingequations below

$$\Delta\theta_j = (l) Err_j$$

$$\theta_j = \theta_j + \Delta\theta_j$$

Algorithm:

Input:

- D, a data set consisting of the training tuples and their associated target values;
- l, the learning rate;
- *network*, a multilayer feed-forward network.

Output: A trained neural network.

Method:

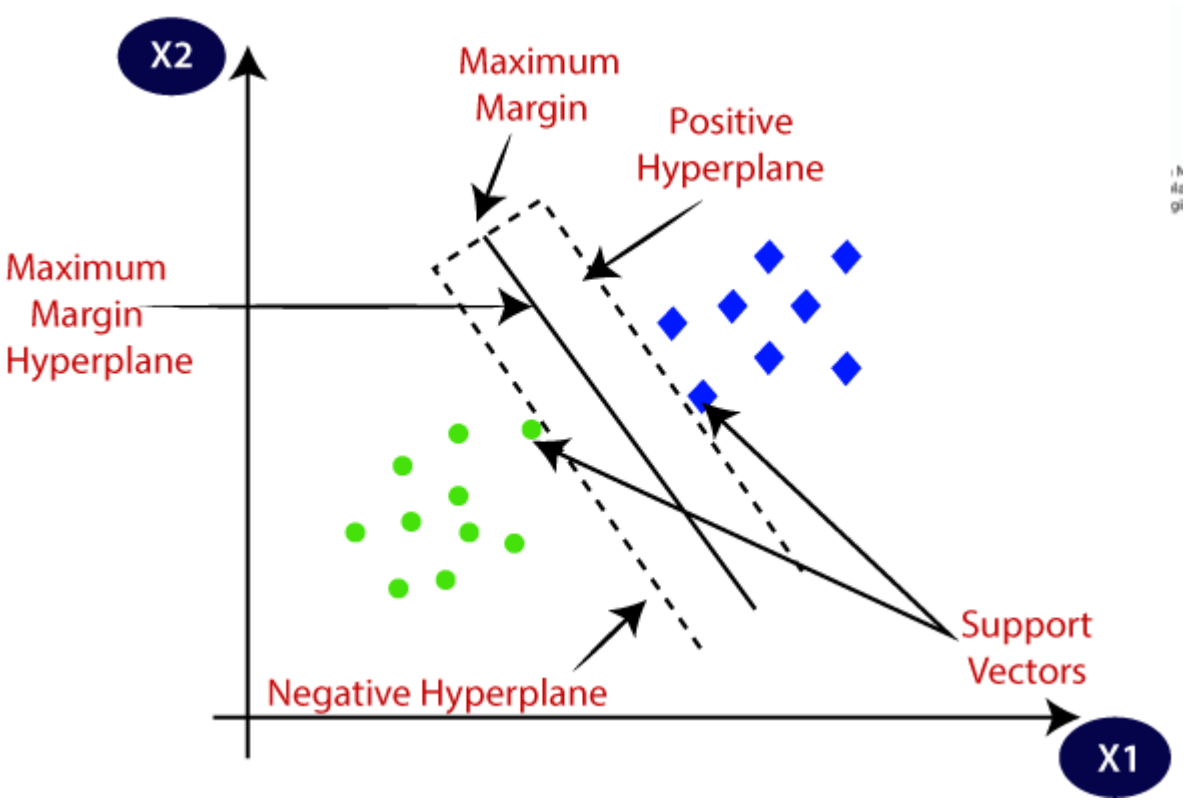
(1) Initialize all weights and biases in *network*;
(2) **while** terminating condition is not satisfied {
(3) **for** each training tuple $\boldsymbol{X}$ in D {
(4) // Propagate the inputs forward:
(5) **for** each input layer unit j {
(6) $O_j = I_j$; // output of an input unit is its actual input value
(7) **for** each hidden or output layer unit j {
(8) $I_j = \sum_i w_{ij} O_i + \theta_j$; //compute the net input of unit j with respect to the previous layer, i
(9) $O_j = \frac{1}{1+e^{-I_j}}$; } // compute the output of each unit j
(10) // Backpropagate the errors:
(11) **for** each unit j in the output layer
(12) $Err_j = O_j(1 - O_j)(T_j - O_j)$; // compute the error
(13) **for** each unit j in the hidden layers, from the last to the first hidden layer
(14) $Err_j = O_j(1 - O_j)\sum_k Err_k w_{jk}$; // compute the error with respect to the next higher layer, k
(15) **for** each weight w_{ij} in *network* {
(16) $\Delta w_{ij} = (l) Err_j O_i$; // weight increment
(17) $w_{ij} = w_{ij} + \Delta w_{ij}$; } // weight update
(18) **for** each bias θ_j in *network* {
(19) $\Delta\theta_j = (l) Err_j$; // bias increment
(20) $\theta_j = \theta_j + \Delta\theta_j$; } // bias update
(21) } }

Support Vector Machine Algorithm

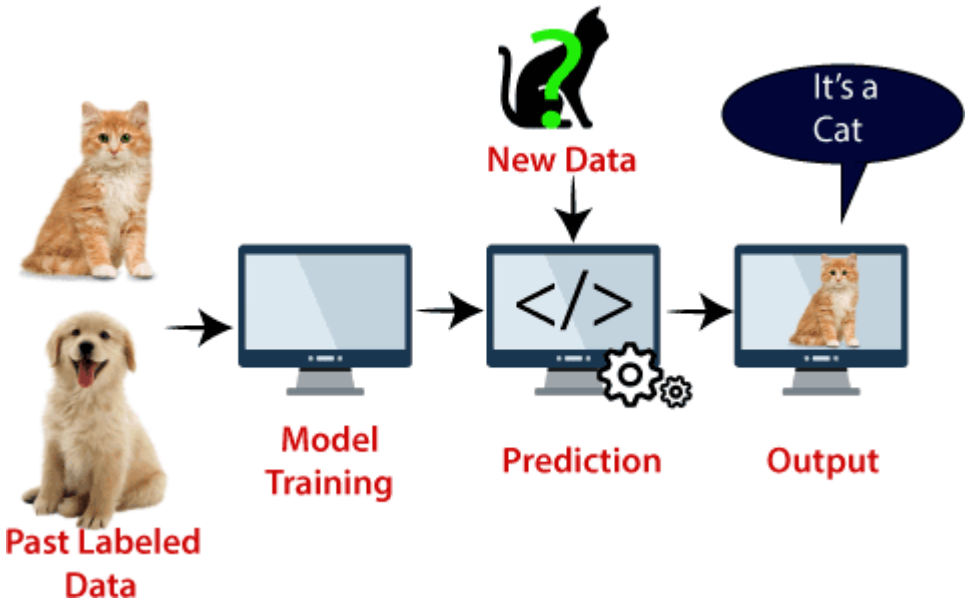
Support Vector Machine or SVM is one of the most popular Supervised Learning algorithms, which is used for Classification as well as Regression problems. However, primarily, it is used for Classification problems in Machine Learning.

The goal of the SVM algorithm is to create the best line or decision boundary that can segregate n-dimensional space into classes so that we can easily put the new data point in the correct category in the future. This best decision boundary is called a hyperplane.

SVM chooses the extreme points/vectors that help in creating the hyperplane. These extreme cases are called as support vectors, and hence algorithm is termed as Support Vector Machine. Consider the below diagram in which there are two different categories that are classified using a decision boundary or hyperplane:



Example: SVM can be understood with the example that we have used in the KNN classifier. Suppose we see a strange cat that also has some features of dogs, so if we want a model that can accurately identify whether it is a cat or dog, so such a model can be created by using the SVM algorithm. We will first train our model with lots of images of cats and dogs so that it can learn about different features of cats and dogs, and then we test it with this strange creature. So as support vector creates a decision boundary between these two data (cat and dog) and choose extreme cases (support vectors), it will see the extreme case of cat and dog. On the basis of the support vectors, it will classify it as a cat. Consider the below diagram:



SVM algorithm can be used for Face detection, image classification, text categorization, etc.

Types of SVM

SVM can be of two types:

Linear SVM: Linear SVM is used for linearly separable data, which means if a dataset can be classified into two classes by using a single straight line, then such data is termed as linearly separable data, and classifier is used called as Linear SVM classifier.

Non-linear SVM: Non-Linear SVM is used for non-linearly separated data, which means if a dataset cannot be classified by using a straight line, then such data is termed as non-linear data and classifier used is called as Non-linear SVM classifier.

Hyperplane and Support Vectors in the SVM algorithm:

Hyperplane: There can be multiple lines/decision boundaries to segregate the classes in n-dimensional space, but we need to find out the best decision boundary that helps to classify the data points. This best boundary is known as the hyperplane of SVM.

The dimensions of the hyperplane depend on the features present in the dataset, which means if there are 2 features (as shown in image), then hyperplane will be a straight line. And if there are 3 features, then hyperplane will be a 2-dimension plane.

We always create a hyperplane that has a maximum margin, which means the maximum distance between the data points.

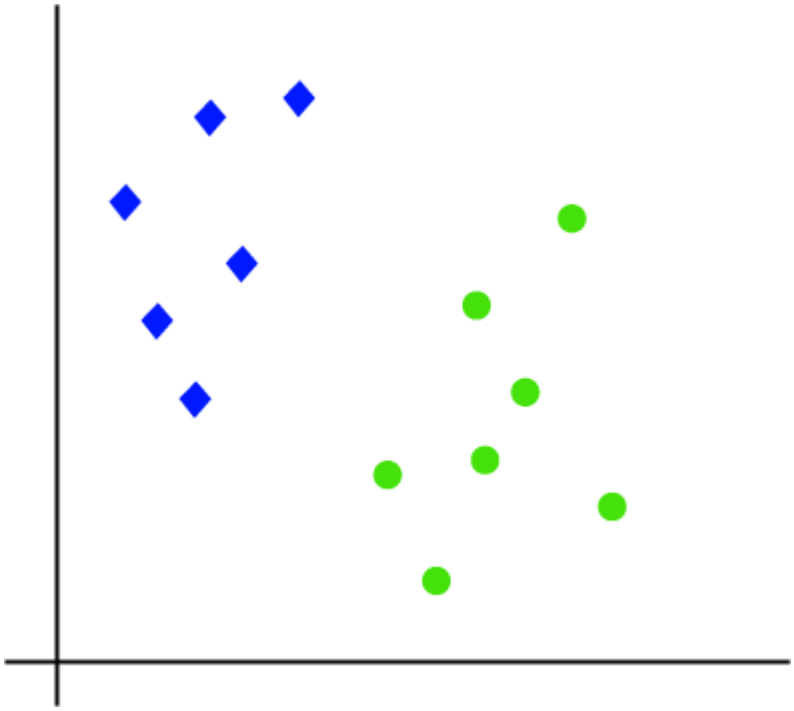
Support Vectors:

The data points or vectors that are the closest to the hyperplane and which affect the position of the hyperplane are termed as Support Vector. Since these vectors support the hyperplane, hence called a Support vector.

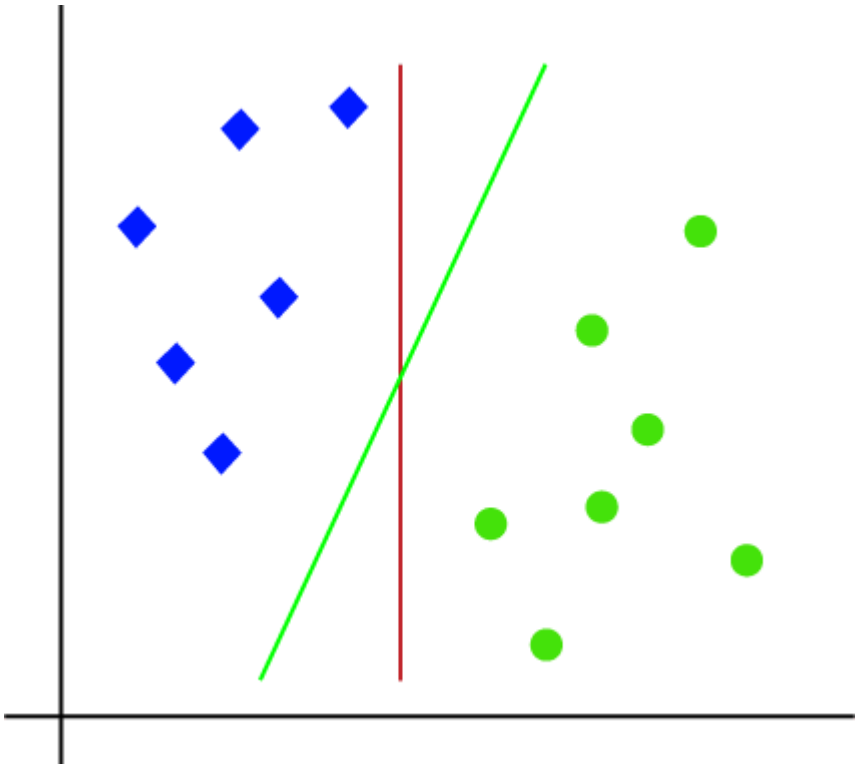
How does SVM works?

Linear SVM:

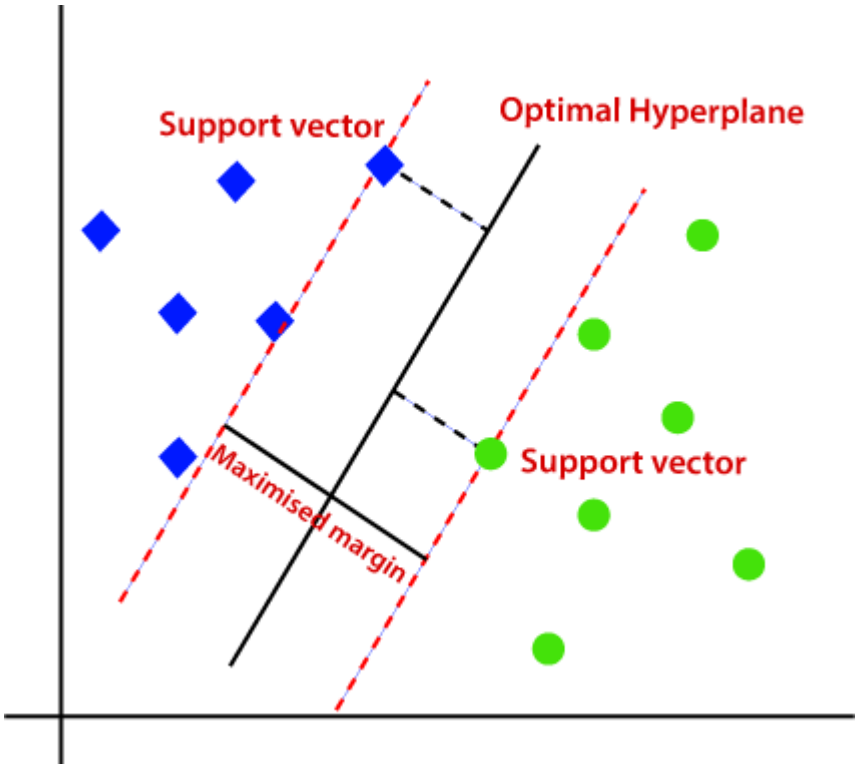
The working of the SVM algorithm can be understood by using an example. Suppose we have a dataset that has two tags (green and blue), and the dataset has two features x1 and x2. We want a classifier that can classify the pair(x1, x2) of coordinates in either green or blue. Consider the below image:



So as it is 2-d space so by just using a straight line, we can easily separate these two classes. But there can be multiple lines that can separate these classes. Consider the below image:

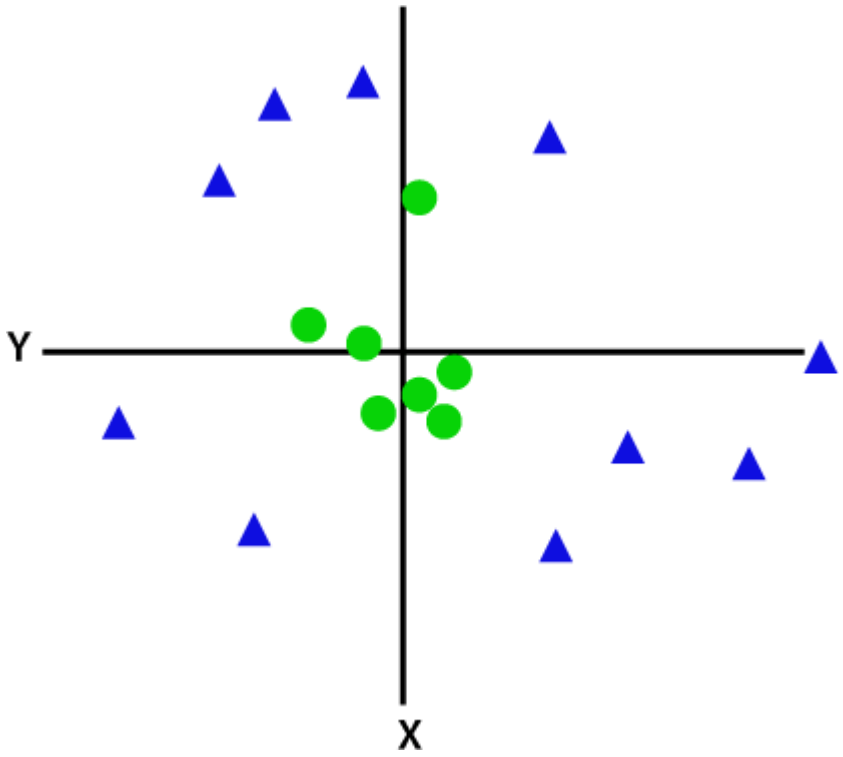


Hence, the SVM algorithm helps to find the best line or decision boundary; this best boundary or region is called as a hyperplane. SVM algorithm finds the closest point of the lines from both the classes. These points are called support vectors. The distance between the vectors and the hyperplane is called as margin. And the goal of SVM is to maximize this margin. The hyperplane with maximum margin is called the optimal hyperplane.



Non-Linear SVM:

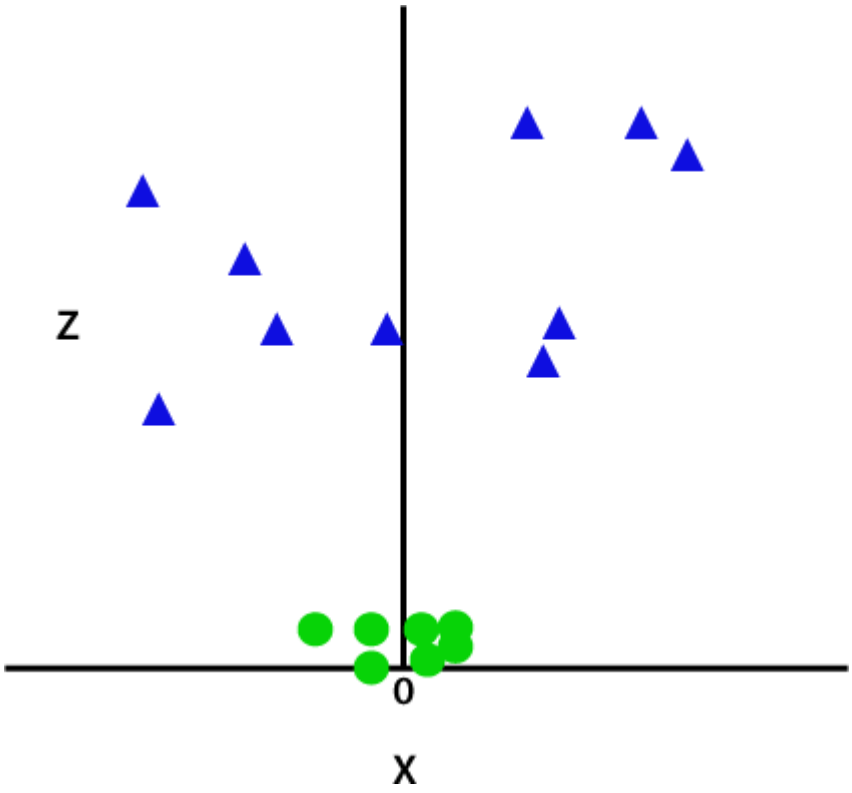
If data is linearly arranged, then we can separate it by using a straight line, but for non-linear data, we cannot draw a single straight line. Consider the below image:



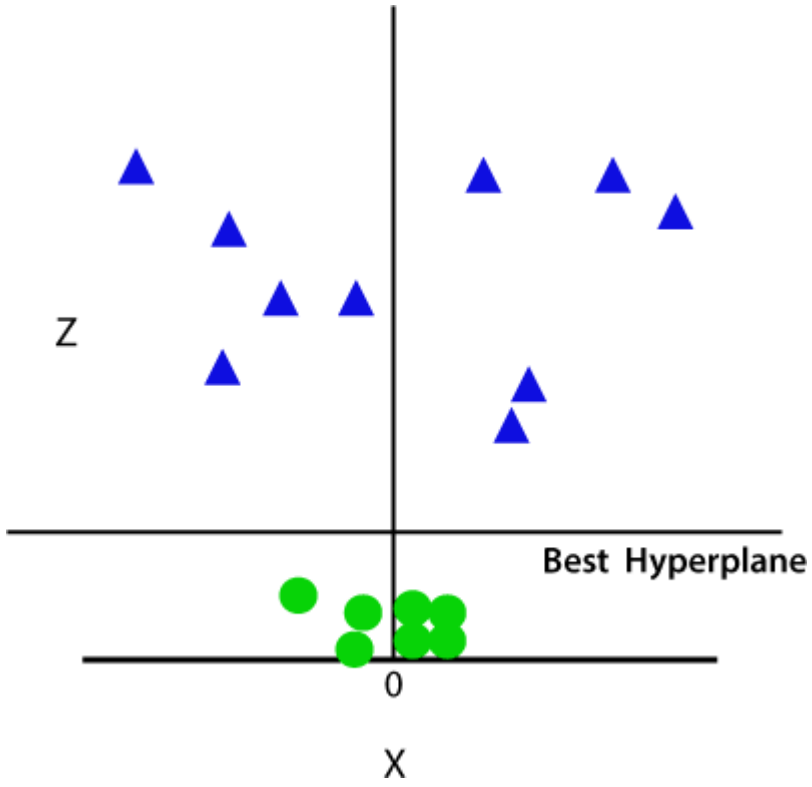
So to separate these data points, we need to add one more dimension. For linear data, we have used two dimensions x and y, so for non-linear data, we will add a third dimension z. It can be calculated as:

z=x2 +y2

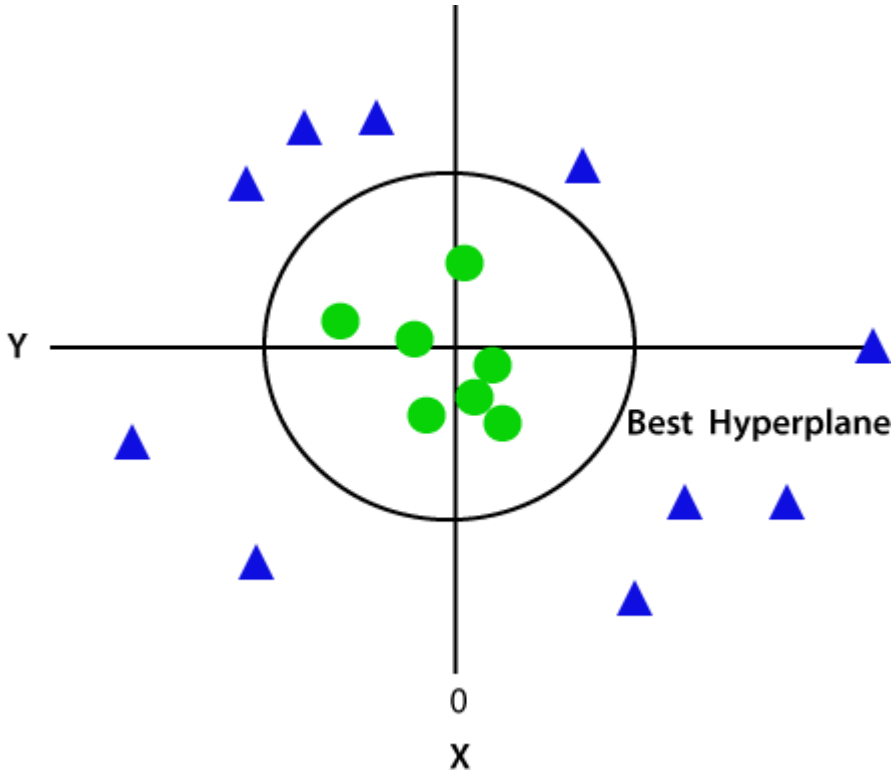
By adding the third dimension, the sample space will become as below image:



So now, SVM will divide the datasets into classes in the following way. Consider the below image:



Since we are in 3-d Space, hence it is looking like a plane parallel to the x-axis. If we convert it in 2d space with z=1, then it will become as:



Hence we get a circumference of radius 1 in case of non-linear data.

k-Nearest-NeighborClassifier:

Nearest-neighbor classifiersarebasedonlearningbyanalogy,thatis,bycomparingagiventesttuplewith trainingtuplesthat aresimilar to it.

The training tuples are describedbyn attributes. Each tuple represents a point in an n-dimensional space. In this way,all of the training tuples are stored in an n- dimensionalpattern space. When given anunknown tuple, a k-nearest-neighbor classifier searches thepattern space for the k trainingtuples that are closest to the unknown tuple. These k trainingtuplesaretheknearestneighbors of theunknowntuple.

Closenessisdefinedintermsofadistancemetric, such asEuclideandistance. TheEuclidean distancebetween twopointsor tuples,say,X1=(x11, x12,… ,x1n) and

X2=(x21, x22, … ,x2n), is

$$dist(X_1, X_2) = \sqrt{\sum_{i=1}^{n} (x_{1i} - x_{2i})^2}.$$

Inotherwords,foreach numericattribute,wetakethedifferencebetweenthecorrespondingvaluesofthatattributeintupleX1a nd intupleX2,squarethisdifference,and accumulateit.

Thesquarerootistakenofthetotalaccumulateddistancecount.

Min-Maxnormalizationcan beused to transformavaluevof anumericattribute Atov0intherange[0, 1]bycomputing

$$v' = \frac{v - min_A}{max_A - min_A},$$

whereminAandmaxAaretheminimumandmaximumvaluesofattribute A

Fork-nearest- neighborclassification,theunknowntupleisassignedthemostcommonclassamongits knearestneighbors.

When k = 1, the unknown tuple is assigned the class of the training tuple that is closest toitin pattern space.

Nearestneighborclassifiers can also be used for prediction, that is, to return a real-valuedpredictionforagiven unknown tuple.

In this case, the classifier returns the averagevalue of the real-valued labels associatedwiththeknearestneighbors ofthe unknowntuple.

OtherClassificationMethods:

GeneticAlgorithms:

Geneticalgorithmsattempttoincorporateideasofnaturalevolution.In general,geneticlearningstartsas follows.

An initial population is created consisting of randomly generated rules. Each rule can

berepresentedbyastringofbits.Asasimpleexample,supposethatsamplesinagiven

trainingset aredescribedbytwoBooleanattributes,A1andA2,andthattherearetwoclasses,C1andC2.

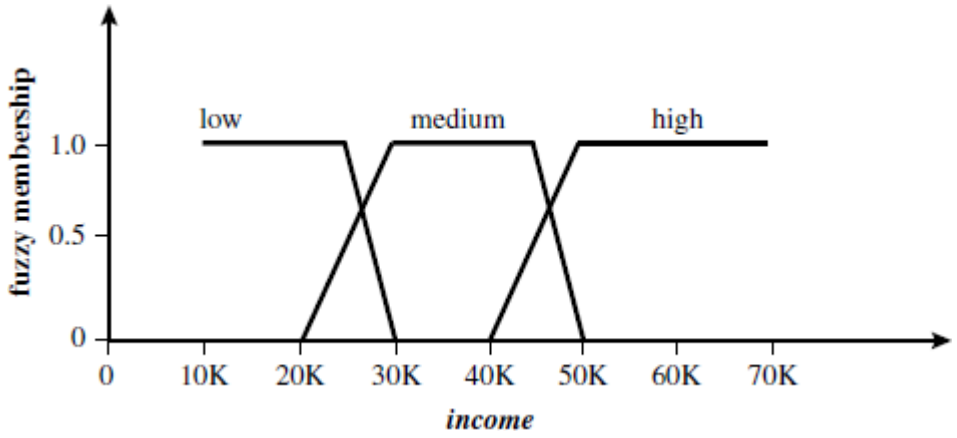
- The rule—IF A1ANDNOTA2THENC2‖can be encoded as the bit string —100,‖where the two leftmost bits represent attributes A1and A2, respectively, and the rightmost bitrepresentstheclass.
- Similarly,therule—IFNOT A1ANDNOTA2THENC1‖can beencodedas —001.‖ Ifanattributehask values,wherek>2,thenkbitsmaybeusedtoencodetheattribute's values.
- Classescanbeencodedinasimilarfashion.
- Basedonthenotionofsurvivalofthefittest,anewpopulationisformed toconsistofthefittestrulesinthecurrent population,aswellasoffspringoftheserules. Typically,thefitnessof aruleisassessedbyitsclassificationaccuracyonaset oftrainingsamples. Offspringarecreated byapplyinggeneticoperatorssuchascrossoverandmutation.In crossover, substrings from pairs of rules are swapped to form new pairs of rules.Inmutation,randomlyselected bits ina rule's stringareinverted.
- Theprocessof generatingnewpopulationsbasedonpriorpopulationsofrulescontinuesuntil a population, P, evolves where each rule in P satisfies a pre specifiedfitnessthreshold.
- Genetic algorithms are easily parallelizable and have been used for classification aswellasother optimizationproblems.Indatamining, theymaybeused toevaluatethefitnessof otheralgorithms.

FuzzySet Approaches:

- Fuzzylogicusestruthvaluesbetween0.0and1.0torepresentthedegree ofmembershipthata certainvaluehasinagiven category.Each categorythen represents afuzzyset.
- Fuzzylogicsystemstypicallyprovide graphicaltoolstoassistusersinconvertingattributevalues to fuzzytruthvalues.
- Fuzzyset theoryis alsoknown as possibilitytheory.

- ItwasproposedbyLotfiZadehin1965asan alternativetotraditionaltwo-valuelogicandprobabilitytheory.
- Itletsusworkatahighlevelofabstractionandoffersameansfordealingwithimprecisemeasurement ofdata.
- Mostimportant, fuzzyset theoryallowsus to dealwithvagueorinexactfacts. Unlikethenotionoftraditional—crisp‖setswhereanelementeitherbelongstoasetSorits complement,
- in fuzzyset theory,elementscanbelongto morethan onefuzzyset. Fuzzyset theoryis useful for data miningsystemsperformingrule-
- basedclassification.Itprovidesoperations for combiningfuzzymeasurements.
- Several procedures exist for translating the resulting fuzzy output into a defuzzifiedor crispvaluethat isreturned bythesystem.
- Fuzzylogicsystemshavebeen usedinnumerousareasforclassification,includingmarketresearch,finance,health care,andenvironmentalengineering.

Example:



RegressionAnalysis:

- Regressionanalysiscan beusedtomodeltherelationshipbetweenoneormoreindependentorpredictorvariables andadependent orresponsevariable whichiscontinuous-valued.
- Inthe contextofdatamining,thepredictorvariablesaretheattributesofinterestdescribingthetuple (i.e., makingupthe attributevector).

- Ingeneral,thevaluesofthepredictorvariablesareknown.

Theresponsevariableiswhatwewanttopredict.

LinearRegression:

Straight-line regression analysis involves a response variable, y, and a single predictorvariablex.

Itisthesimplestformofregression,andmodelsyasalinearfunctionofx.

Thatis,y=b+wx wherethevarianceofyisassumedtobeconstant

bandwareregressioncoefficientsspecifyingthe Y-interceptandslopeoftheline.

Theregressioncoefficients,wandb,canalsobethoughtof asweights,sothatwecanequivalentlywrite,y=w0+w1x

Thesecoefficientscanbesolvedforbythemethodofleastsquares,which estimatesthebest- fitting straight line as the one that minimizes the error between the actual data andtheestimateof theline.

Let D be a training set consisting of values of predictor variable, x, for some population andtheir associated values for response variable, y. The training set contains |D| data points oftheform(x1, y1), (x2, y2),… , (x|D|, y|D|).

Theregressioncoefficientscanbeestimatedusingthismethodwiththefollowingequations:

$$w_1 = \frac{\sum_{i=1}^{|D|}(x_i - \bar{x})(y_i - \bar{y})}{\sum_{i=1}^{|D|}(x_i - \bar{x})^2}$$

$$w_0 = \bar{y} - w_1\bar{x}$$

where x is themeanvalue of x1, x2,… ,x|D|,andyisthemeanvalue ofy1, y2,…,y|D|.Thecoefficientsw0andw1often provide good approximations to otherwisecomplicatedregressionequations.

MultipleLinearRegression:

Itisanextensionofstraight-lineregressionsoas toinvolvemorethanonepredictorvariable.

- Itallowsresponsevariable ytobemodeledasalinearfunctionof,say,npredictorvariablesorattributes,A1,A2, …, An,describingatuple,X.
- An example of a multiple linear regression model basedon two predictor attributes orvariables,A1and A2, isy=w0+w1x1+w2x2 wherex1andx2arethevaluesofattributesA1andA2,respectively,inX.
- Multipleregressionproblemsareinsteadcommonlysolvedwiththeuseofstatisticalsoftwarep ackages,suchasSAS,SPSS, and S-Plus.

NonlinearRegression:

- Itcanbemodeledbyaddingpolynomialterms tothebasiclinearmodel.
- By applying transformations to the variables, we can convert the nonlinear model into alinearonethat can then besolved bythe method ofleastsquares.
- Polynomial Regression is a special case of multiple regression. That is, the addition ofhigh-order terms like x2, x3, and so on, which are simple functions of the single variable, x,canbeconsideredequivalent to addingnewindependent variables.

Transformationofapolynomialregressionmodeltoalinearregressionmodel:

Consideracubicpolynomial relationshipgivenby

y=w0+w1x+w2x2+w3x3

Toconvertthisequation tolinearform,wedefinenewvariables:

x1 =x,x2 = x2 ,x3 =x3

It can then be converted to linear formby applying the above assignments,resulting in theequationy=w0+w1x+w2x2+w3x3 whichiseasilysolvedbythemethodofleastsquaresusingsoftwareforregressionanalysis.

ClassifierAccuracy:

- The accuracy of a classifier on a given test set is the percentage of test set tuples that arecorrectlyclassified bytheclassifier.
- In the pattern recognition literature, this is also referred to as the overall recognition rate oftheclassifier, that is, it reflects how well the classifier recognizes tuples of the variousclasses.

- The error rate or misclassification rate of a classifier,M, which is simply 1- Acc(M),whereAcc(M)is theaccuracyofM. Theconfusionmatrixisausefultoolforanalyzinghowwellyourclassifiercan
- recognizetuplesofdifferent classes.
- Truepositivesrefer to thepositivetuplesthat werecorrectlylabeled
- bytheclassifier.Truenegativesarethe negativetuplesthat
- werecorrectlylabeled bytheclassifier.
- Falsepositives arethe negativetuplesthatwereincorrectlylabeled. Howwelltheclassifiercanrecognize,forthissensitivityandspecificitymeas urescanbeused.

- Accuracyisafunctionofsensitivityandspecificity.

$$accuracy = sensitivity\frac{pos}{(pos+neg)} + specificity\frac{neg}{(pos+neg)}.$$

$$sensitivity = \frac{t\_pos}{pos}$$
$$specificity = \frac{t\_neg}{neg}$$
$$precision = \frac{t\_pos}{(t\_pos + f\_pos)}$$

wheret_posisthenumberoftruepositivespo sisthe number ofpositivetuples

t _negis the number of true negativesnegisthenumberofnegativetuples,f_posis thenumber offalsepositives

The grid-based clustering methods use a multi-resolution grid data structure. It quantizes the object areas into a finite number of cells that form a grid structure on which all of the operations for clustering are implemented. The benefit of the method is its quick processing time, which is generally independent of the number of data objects, still dependent on only the multiple cells in each dimension in the quantized space.

An instance of the grid-based approach involves STING, which explores statistical data stored in the grid cells, WaveCluster, which clusters objects using a wavelet transform approach, and CLIQUE, which defines a grid-and density-based approach for clustering in high-dimensional data space.

STING is a grid-based multiresolution clustering method in which the spatial area is divided into rectangular cells. There are generally several levels of such rectangular cells corresponding to multiple levels of resolution, and these cells form a hierarchical mechanism each cell at a high level is separation to form several cells at the next lower level. Statistical data regarding the attributes in each grid cell (including the mean, maximum, and minimum values) is precomputed and stored.

Statistical parameters of higher-level cells can simply be calculated from the parameters of the lower-level cells. These parameters contain the following: the attribute-independent parameter, count, and the attribute-dependent parameters, mean, stdev (standard deviation), min (minimum), max (maximum); and the type of distribution that the attribute value in the cell follows, including normal, uniform, exponential, or none (if the distribution is anonymous).

When the records are loaded into the database, the parameters count, mean, stdev, min, and a max of the bottom-level cells are computed directly from the records. The value of distribution can be assigned by the user if the distribution type is known beforehand or obtained by hypothesis tests including the $\chi 2$ test.

The kind of distribution of a higher-level cell that can be computed depends on the majority of distribution types of its corresponding lower-level cells in conjunction with a threshold filtering procedure. If the distributions of the lower-level cells disagree with each other and decline the threshold test, the distribution type of the high-level cell is set to none.

The statistical parameters can be used in top-down, grid-based approaches as follows. First, a layer within the hierarchical architecture is decided from which the query-answering procedure is to start. This layer generally includes a small number of cells. For every cell in the current layer, it can compute the confidence interval (or estimated

range of probability) reflecting the cell's relevancy to the given query.

UNIT V

Applications

Data mining is many and varied fields of applications.

Data Mining for Financial Data Analysis

Financial data collected in the banking and financial industry are often relatively complete, reliable, and of high quality, which facilitates systematic data analysis and data mining. A few examples of data mining in the Financial Data Analysis are outlined as follows:

Detect patterns of fraudulent credit card use Identify loyal customers
Predict customers likely to change their credit card affiliation Determine credit card spending by customer groups
Find hidden correlations between different financial indicators Identify stock trading rules from historical market data.

Design and Construction of data warehouses for multidimensional data analysis and data mining

Data warehouses need to be constructed for banking and financial data. Multidimensional data analysis methods should be used to analyze the general properties of such data warehouses, data cubes, multi feature and discovery-driven data cubes, characterization and class comparisons, and outlier analysis all play important roles in financial data analysis and mining.

Loan payment prediction and customer credit policy analysis

Data mining methods, such as attribute selection and attribute relevance ranking, may help identify important factors such as loan-to-value ratio, duration of the loan, debt ratio, payment to-income ratio, customer income level, education level, residence region, and credit history and eliminate irrelevant ones for loan payment prediction and customer credit rating.

Classification and clustering of customers for targeted marketing

Classification technique is used to identify the most crucial factors that may influence a customer's decision regarding banking. Customers with similar behaviors regarding loan payments may be identified by multidimensional clustering techniques. These data mining techniques helps to identify customer groups, associate a new customer with an appropriate customer group, and facilitate targeted marketing. Multiple data analysis tools such as data visualization tools, linkage analysis tools, classification tools, clustering tools, outlier analysis tools and sequential pattern analysis tools can then be used to detect unusual patterns, such as large amount of cash flow at certain periods, by certain groups of customers and also may identify important relationships and patterns of activities for further examination.

Data Mining for the Retail Industry

The retail industry is a major application area for data mining, since it collects huge amounts of data on sales, customer shopping history, goods transportation, consumption, and service. Retail data mining can help to:

Identify buying patterns from customers

Discovers customer shopping patterns and trends

Find associations among customer demographic characteristics

Predict response to mailing campaigns Improve the quality of customer service

Achieve better customer retention and satisfaction

Reduces the cost of business

Market basket analysis

A few examples of data mining in the retail industry are outlined as follows

Design and construction of data warehouses based on the benefits of data mining

The outcome of preliminary data mining exercises can be used to help guide the design and development of data warehouse structures. This involves deciding which dimensions and levels to include and what preprocessing to perform in order to facilitate effective data mining.

Multidimensional analysis of sales, customers, products, time, and

region

The retail industry requires timely information regarding customer needs, product sales, trends, and fashions, as well as the quality, cost, profit, and service of commodities. It is therefore important to provide powerful multidimensional analysis and visualization tools and multi feature data cube to facilitate analysis on aggregates with complex conditions.

Analysis of the effectiveness of sales campaigns

The retail industry conducts sales campaigns using advertisements, coupons, and various kinds of discounts and bonuses to promote products and attract customers. Multidimensional analysis can be used for careful analysis of the effectiveness of sales campaigns to improve company profits. Association analysis may disclose which items are likely to be purchased together with the items on sale.

Customer retention – analysis of customer loyalty

Sequential pattern mining can be used to investigate changes in customer consumption or loyalty and suggest adjustments on the pricing and variety of goods in order to help retain customers and attract new ones. Product recommendations can also be advertised on sales receipts, in weekly flyers or on the web to help improve customer service aid customers in selecting items, and increase sales.

Data mining for the Telecommunication Industry

The integration of telecommunication, computer network, internet, and numerous other means of communication and computing is underway. With the deregulation of the telecommunication industry in many countries and the development of new computer and communication technologies, the telecommunication market is rapidly expanding and highly competitive. This creates a great demand for data mining in order to help.

To understand the business involved

To identify telecommunication patterns

To catch fraudulent activities

To make better use of resources

To improve the quality of service.

The following are a few scenarios for which data mining may improve Telecommunication services:

Multidimensional analysis of telecommunication data

The multidimensional analysis of telecommunication data using OLAP and visualization tools can be used to identify and compare the data traffic, system workload, resource usage, user group behavior, and profit.

Fraudulent pattern analysis and the identification of unusual patterns

The multidimensional analysis, cluster analysis, and outlier analysis are used to (1) identify potentially fraudulent users and their a typical usage patterns; (2) detect attempts to gain fraudulent entry to customer accounts; and (3) discover unusual patterns.

Multidimensional association and sequential pattern analysis

The discovery of association and sequential patterns in multidimensional analysis can be used to promote telecommunication service.

Mobile telecommunication services

Mobile Telecommunication, Web and information services, and mobile computing are becoming increasingly integrated and common in our work and life. One important feature of mobile telecommunication data is its association with spatiotemporal information. Data Mining will likely play a major role in the design of adaptive solutions enabling users to obtain useful information with relatively few keystrokes.

Use of visualization tools in telecommunication data analysis

Tools for OLAP visualization, linkage visualization, association visualization, clustering, and outlier visualization have been shown to be very useful for telecommunication data analysis.

Data Mining for biological Data Analysis

Biological data mining has become an essential part of a new research field called bioinformatics. The identification of DNA or amino acid sequence patterns that play roles in various biological function, genetic diseases, and evolution is challenging. Biological data mining helps to:

Characterize patient behaviour to predict office visits

Identify successful medical therapies for different illnesses

Develop effective genomic and proteomic data analysis tools.

DNA sequences form the foundation of the genetic codes of all living organisms. All DNA sequences are comprised of four basic building blocks, called nucleotides: adenine (A), cytosine (C), guanine (G), and thymine (T). These four nucleotides (or bases) are combined to form long sequences or chains that resemble a twisted ladder. A genome is the complete set of genes of an organism. The human genome is estimated to contain around 20,000 to 25,000 genes. Genomic is the analysis of genome sequences.

Proteins are essential molecules for any organism. They perform life functions and make up the majority of cellular structures.

There are 20 amino acids, each of the amino acids is coded for by one or more triplets of nucleotides making up DNA. The end of the chain is coded for by another set of triplets. Thus, a linear string or sequence of DNA is translated into a sequence of amino acids, forming a protein .

DNA sequence CTA CAC ACG TGT AAC
Amino acid L H T C N sequence
Fig : A DNA sequence and corresponding amino acid sequence

A proteome is the complete molecules present in a cell tissue, or organism. Proteomics is the study of proteome sequences.
Data mining may contribute to biological data analysis in the following aspects:
Semantic integration of heterogeneous, distributed genomic and proteomic databases

Data cleaning, data integration, reference reconciliation, classification, and clustering methods will facilitate the systematic and coordinated analysis of genome and biological data and also integrates the biological data and the construction of data warehouses for biological data analysis.

Alignment, indexing, similarity search, and comparative analysis of multiple nucleotide / protein sequences

Multiple sequence alignment is considered a more challenging task. Methods that can help include (1) reducing a multiple alignment to a series of pair wise alignment and then combining the result, and (2) using hidden Markov Models or HMMs. Multiple sequence alignments can be used to identify highly conserved residues among genomes, and such conserved regions can be used to build phylogenetic trees to infer evolutionary relationships among species.

Discovery of structural patterns and analysis of genetic networks and protein pathways

In biology, protein sequences are folded into three-dimensional structures which interact with each other based on their relative position and the distances between which forms basis of genetic networks and protein pathways. Powerful and scalable data mining methods are developed to discover approximate and frequent structural patterns and to study the regularities and irregularities among such interconnected biological networks.

Association and path analysis: Identifying co-occurring gene sequences and linking genes to different stages of disease development

Association analysis methods can be used to help determine the kinds of genes that are likely to co-occur in target samples and also facilitate the discovery of groups of genes and the study of interactions and relationships between them. Path analysis develops pharmaceutical interventions that target the different stages of disease development.

Visualization tools in genetic data analysis

The visually appealing of biological structures and patterns facilitate pattern understanding, knowledge discovery, and interactive data exploration. Visualization and visual data mining therefore play an important role in biological data analysis.

Data Mining in other Scientific Applications

Vast amounts of data have been collected from scientific domain (including geosciences, astronomy, and meteorology) using sophisticated telescopes, multispectral high resolution remote satellite sensors, and global positioning systems to analyze complex data set. Some of the emerging scientific applications of data mining are:

Data warehouses and data preprocessing

Data warehouses are critical for information exchange and data mining. Scientific applications requires methods for integrating data from heterogeneous sources, for identifying events, for the efficient computation of sophisticated spatial aggregates and the handling of spatial-related data streams.

Mining Complex data types

Scientific data sets are heterogeneous in nature, typically involving semi-structured and unstructured data. Robust methods are needed for handling spatiotemporal data, related concept hierarchies, and complex geographic relationships.

Graph-based mining

In graph modeling, each object to be mined is represented by a vertex in a graph, and edges between vertices represent relationships between objects. The success of graph-modeling, however, depends on improvement in the scalability and efficiency of many classical data mining tasks, such as classification, frequent pattern mining, and clustering.

Visualization tools and domain-specific knowledge

High-level graphical user interface and visualization tools are required for scientific data mining systems to guide researcher and general users in searching for patterns, interpreting and visualizing discovered patterns and using discovered knowledge in their decision making.

Text Mining and Web Mining

Text mining is the process of searching large volumes of documents from certain keywords or key phrases. By searching literally thousands of documents various relationships between the documents can be established. Using text mining however, we can easily derive certain patterns in the comments that may help identify a common set of customer perceptions not captured by the other survey questions.

An extension of text mining is web mining. Web mining is an exciting new field that integrates data and text mining within a website. It enhances the web site with intelligent behavior, such as suggesting related links or recommending new products to the consumer. Web mining is especially exciting because it enables tasks that were previously difficult to implement. They can be configured to monitor and gather data from a wide variety of locations and can analyze the data across one or multiple sites. For example the search engines work on the principle of data mining.

Data Mining for Intrusion Detection

An intrusion can be defined as any set of actions that threaten the integrity, confidentiality or availability of a network resource. An intrusion detection system for a large complete network can typically generate thousands or millions of alarms per day representing an overwhelming task for the security analysts.

Anomaly detection builds models of normal network behaviour (called profiles), which it uses to detect new pattern that significantly deviate from the profiles.

The following are areas in which data mining technology may be applied or further developed for intrusion detection.

Development of data mining algorithms for intrusion detection

Data mining algorithms can be used for misuse detection and anomaly detection. Anomaly detection builds models of normal behavior and automatically detects significant deviations from it. Supervised or unsupervised learning can be used. The techniques used must be efficient and scalable, and capable of handling network data of high volume, dimensionality, and heterogeneity.

Association and Correlation analysis, and aggregation to help select and build discriminating attributes

Association and correlation mining can be applied to find relationships between system attributes describing the network data for intrusion detection. Thus it is necessary to study what sequences of events are frequently encountered together, find sequential patterns, and identify outliers.

Distributed data mining

Distributed data mining methods may be used to analyze network data from several network location in order to detect these distributed attacks.

Visualization and querying tools

Visualization tools detects anomalous patterns using associations, clusters, and outliers techniques. These tools are more precise and require for less manual processing and input from human experts.

Higher Education

An important challenge that higher education faces today is predicting paths of students and alumni. Which student will enroll in particular course programs? Who will need additional assistance in order to graduate? Mean while additional issues, enrollment management and time- to-degree, continue to exert pressure on colleges to search for new and faster solutions.

Institutions can better address these students and alumni through the analysis and presentation of data. Data mining has quickly emerged as a highly desirable tool for using current reporting capabilities to uncover and understand hidden patterns in vast databases.

Data Mining System Products

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

Choosing a Data Mining System

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

Data Types − The data mining system may handle formatted text, record-based data, and relational data. The data could also be in ASCII text, relational database data or data warehouse data. Therefore, we should check what exact format the data mining system can handle.

System Issues − We must consider the compatibility of a data mining system with different operating systems. One data mining

system may run on only one operating system or on several. There are also data mining systems that provide web-based user interfaces and allow XML data as input.

Data Sources − Data sources refer to the data formats in which data mining system will operate. Some data mining system may work only on ASCII text files while others on multiple relational sources. Data mining system should also support ODBC connections or OLE DB for ODBC connections.

Data Mining functions and methodologies − There are some data mining systems that provide only one data mining function such as classification while some provides multiple data mining functions such as concept description, discovery-driven OLAP analysis, association mining, linkage analysis, statistical analysis, classification, prediction, clustering, outlier analysis, similarity search, etc.

Coupling data mining with databases or data warehouse systems − Data mining systems need to be coupled with a database or a data warehouse system. The coupled components are integrated into a uniform information processing environment. Here are the types of coupling listed below −

No coupling

Loose Coupling

Semi tight Coupling

Tight Coupling

Scalability − There are two scalability issues in data mining −

Row (Database size) Scalability − A data mining system is considered as row scalable when the number or rows are enlarged 10 times. It takes no more than 10 times to execute a query.

Column (Dimension) Salability − A data mining system is considered as column scalable if the mining query execution time increases linearly with the number of columns.

Visualization Tools − Visualization in data mining can be categorized as follows −

Data Visualization

Mining Results Visualization

Mining process visualization

Visual data mining

Data Mining query language and graphical user interface − An easy-to-use graphical user interface is important to promote user-guided, interactive data mining. Unlike relational database systems, data mining systems do not share underlying data mining query language.

Additional Themes on Data Mining

Due to the broad scope of data mining and the large variety of data mining methodologies, not all of the themes on data mining can be thoroughly covered here.

Visual and Audio Data Mining

Visual data mining discovers implicit and useful knowledge from large data using data and/or knowledge visualization techniques. The eyes and brain, the latter of which can be thought of as a powerful, highly parallel processing and reasoning engine containing a large knowledge base, control the

human visual system. Visual data mining essentially combines the power of these components making it a highly attractive and effective tool for the comprehension of data distributions, patterns, clusters, and outliers in data.

Visual data mining can be viewed as an integration of two disciplines: visualization and data mining.

It is also closely related to computer graphical multimedia systems, human computer interfaces, pattern recognition, and high performance computing. In general, data visualization and data mining can be integrated in the following ways:

Data visualization:

Data in a database or data warehouse can be viewed different levels of granularity or abstraction, or

as different combination attributes or dimensions. Data can be presented in various visual forms, as box plots, 3-D cubes, data distribution charts, curves, surfaces, link graph and so on. Visual display can help give users' a clear impression and overview of the data characteristics in a database.

Data mining result visualization:

Visualization of data mining results is presentation of the results or knowledge obtained from data mining in via forms. Such forms may include scatter plots and boxplots (obtained in descriptive data mining), as well as decision trees, association rule clusti outliers, generalized rules, and so on. For example, scatter plots are shown

in

Data mining process visualization:

This type of visualization presents the various processes of data mining in visual forms so that users

can see how the data are extracted and from which database or data warehouse they are extracted, as well as how the selected data are cleaned, integrated, processed, and mined. Moreover, it may also show which method is selected for data mining, where the results are stored, and how they may be viewed.

Interactive visual data mining:

In (interactive) visual data mining, visualization tools can be used in the data mining process to help users make smart data mining decisions. For example, the data distribution in a set of attributes can be displayed using colored sectors or columns (depending: on. whether the whole space is represented by either a circle or a set of columns). This display may help users determine which sector should first be

selected for classification and where a good split point for this sector may be an interesting alternative to visual mining.

Audio data mining:

Uses audio signals to indicate the patterns of data or the features of data mining results. Although visual data mining may disclose interesting patterns using graphical displays, it requires users to concentrate on watching patterns and identifying interesting or novel features within them. This can sometimes be quite tiresome. If patterns can be transformed into sound and music, then instead of watching pictures, we can listen to pitches, rhythms, tune, and melody in order to identify anything

interesting or unusual. This may relieve some of the burden of visual concentration and be more relaxing than visual mining in many cases. Therefore, audio data mining can be an interesting alternative to visual mining.

Scientific and Statistical Data Mining

The data mining techniques described in this book are primarily database oriented, that is, designed for the efficient handling of huge amounts of data tin are typically multidimensional and

possibly of various complex types. There are however, many well-established statistical techniques for data analysis, particularly for numeric data. These techniques have been applied extensively to scientific data (e.g., data from experiments in psychology, medicine, electrical engineering and manufacturing), as well as to data from economics and the social sciences.

Some of these techniques, such as principal component analysis, regression, and clustering, have already been addressed in this book. A thorough discussion of major statistical methods for data analysis is beyond the scope of this work; however, several methods are mentioned below for the sake of-completeness.

Regression: - .

In general, these methods are used to predict the value of a response (dependent) variable from one

or more predictor (independent) variables where the variables are numeric. There are various forms of regression, such as, linear, multiple, weighted polynomial, non-parametric and robust (where robust methods are useful when errors fail to satisfy normalcy conditions or when the data contain significant outliers).

Generalized linear models:

These models and their generalization (generalized additive

models), allow a categorical response variable (or some transformation of it) to be related to a set of predictor variables in a manner similar to the modeling of a numeric response variable using linear regression. Generalized linear models include logistic regression and Poisson regression.

Regression trees:

These can be used for classification and prediction. The trees constructed are binary. A regression tree is similar to a decision tree in the sense that tests are performed at the internal nodes. A major difference is at the leaf level-while in a decision tree a majority voting is performed to assign a class label to the leaf, in a regression tree the mean of the objective attribute is computed and used as the predicted value.

Analysis of variance:

These techniques analyze experimental data for two or more populations described by a numeric response variable and one or more categorical variables (factors). In general, an ANOVA (single-factor analysis of variance) problem involves a comparison of population or treatment means to determine if at least two of the means are different. More complex ANOVA problems also exist.

Mixed-effect models:

These models are for analyzing grouped data-data that can be classified according to one or more grouping variables. They typically describe relationships between a response variable and some covariates in data grouped according to one or more factors. Common areas of application include multilevel data, repeated measures data, block designs, and longitudinal data.

Factor analysis:

This method is use& to determine which variables are combined to generate a given factor. For example, for many psychiatric data it is not possible to measure a certain factor of interest directly (such as intelligence); however, it is often possible to measure other quantities (such as student test scores) that reflect the factor of interest. Here, none of the variables are designated as dependent.

Discriminant analysis:

This technique is used to predict a categorical response variable. Unlike generalized linear models, it assumes that the independent variables follow a multivariate normal distribution. The procedure, attempts to determine several discriminant functions (linear combinations of the independent variables) that discriminate among the groups defined by the response variable. Discriminant analysis is commonly used in the social sciences.

Time series:

These are many statistical techniques for analyzing time-series data, such as autoregression methods, univariate ARIMA (autoregressive integrated moving average) modeling, and long-memory time-series modeling.

Survival analysis:

Several well-established statistical techniques exist for survival analysis, which originally were

designed to predict the probability that a patient undergoing a

medical treatment would survive at least to time r. Methods for survival analysis, however, are also commonly applied to manufacturing settings to estimate the life span of industrial equipment. Popular methods include Kaplan-Meier estimates of survival, Cox proportional hazards regression models, and their extensions.

Quality control:

Various statistics can be used to prepare charts for quality control, such as Shewhart charts and cusum charts (both of which display group summary statistics). These statistics include the mean, standard deviation, range, count, moving average, moving standard deviation, and moving range.

Social Impacts of Data Mining

With the fast computerization of society, the social impacts of data mining should not be underestimated. Is data mining a hype, or is it really here to stay? What obstacles must be met in order for data mining to become accepted as a mainstream technology for business, and eventually, for everyone's personal use? What can be done toward protecting data privacy and security? This section addresses each of these questions.

Is Data Mining Hype or a Persistent, Steadily Growing Business?

Data mining has recently become very popular, with many people jumping into data mining research, development, or business, or claiming their software systems to be data mining products.

Observing this, you may wonder, "Is data mining a hype, or is it here to stay? How well accepted is it, as a technology?"

Granted, there has been a great deal of hype regarding data mining since its emergence during the late 1980s, especially because many people expect that data mining will become an essential tool for deriving knowledge from data, to help business executives make strategic decisions, to sharpen the competitive edge of a business;, and do many other wonderful things.

Data mining is a technology. Like any other technology, data mining wi ll require time and effort

to research, to develop, and to mature, and its adoption will, likely go through a life cycle consisting of the fol lowing stages.

Innovators: The new technology starts to take form as researchers begin to realize the need for methods to solve a particular (possibly new) problem. Early adopters: Interest increases as more and more methods for the technology are proposed.

Chasm: This represents the "hurdles" or challenges that must be met before the technology can become widely accepted as mainstream.

Early majority: The technology becomes mature and is generally accepted and

used.

Late majority: The technology is well accepted, but interest in it declines as the initial

problem either becomes less important or is replaced, by other needs.

Laggards: Use of the technology stars to die out, as it becomes old and outdated.

“So, at what stage is data mining?” Several recent discussions have placed data mining at a chasm. In order for data mining to become fully accepted, as a technology, further research and developments

are needed in the many areas mentioned as-efficiency and scalability, increased user interaction, incorporation of background knowledge and visualization techniques, the evolution of a standardized data mining query language, effective methods for finding interesting patterns, improved handling of complex data types, Web mining, and so on.

For data Mining, to "climb out" of the chasm, we also need to focus on the integrated of data mining into existing business technology. Currently, there exists a good variety of generic data mining systems. However, many of these tend to be designed for specifically trained experts who are familiar with data mining jargon and data analysis techniques, like association, classification, and clustering. This makes

such systems difficult to use for business executives and the general public. Moreover, these systems tend to be designed to provide horizontal solutions that are geared to work for all kinds of business but are not specially designed to provide business-specific data mining solutions. Since effective data mining requires

the smooth integration of business logic with data mining functions, one cannot expect that generic data mining systems can achieve as great a success in business intelligence as domain-independent relational

database systems have done in business transaction and query processing.

Many data mining researchers and developers believe that a promising direction for data mining is to construct data mining systems that provide vertical solutions, that is, the integration of in-depth domainspecific

business logic into data mining systems. Business conducted on the Web, or e-commerce, is an

obvious venue for data mining, as more companies collect large amounts of data from e-stores set up on the Web (also called Web stores). We will therefore examine how to provide domain-specific data mining solutions for e-commerce applications.

Currently, more tailored systems are required that facilitate marketing campaign management (often called e-marketing). Ideally, such closed-loop systems bring together customer data analysis (with OLAP and mining technologies embedded under a user-friendly interface), customer profiling (or one-to-one segments'), campaign roll-out, and campaign analysis.

These systems increasingly use data mining for customer relationship management (CRM), which helps companies to provide more customized, personal service to their customers in lieu of mass

marketing. By studying browsing and purchasing patterns on Web stores (e.g., by analyzing click streams, the information that consumers provide by clicks of the mouse), companies can learn more about individual customers or customer groups. This information can be applied to the benefit of both the company and the customer involved. For example, by having more accurate models of their customers, companies should gain a better understanding of customer needs. Serving these needs can result in greater success regarding cross-selling of related products, up-selling, one-to-one promotions, product affinities, larger baskets, and customer retention. By tailoring advertisements and promotions to customer profiles, customers are less likely to be annoyed with unwanted mass mailings or junk mail. These actions can result in substantial cost savings for companies. The customer further benefits in that she is more likely to be notified of offers that are actually of interest to her, resulting in less waste of personal time and greater satisfaction. Customer-tailored advertisements are not limited to company mail-outs or ads planed on Web stores: In the future, digital television and on-line books and newspapers may also provide advertisements that are designed and selected specifically for the given viewer or viewer group based on customer profiling information and demographics. It is

important to note that- data mining is just one piece of the integrated solution. Other components, such as data cleaning and data integration, OLAP, user security, inventory and order management, product management, and so on. Must also be in place.

Is Data Mining Merely Managers Business or Everyone's Business?

Data mining will surely help company executives a great deal in understanding the market and their business. However, "is data mining merely managers' business or every ones business?" Since more and more data are being made available on the Web or possibly on your own disks, it is likely you will need data mining to understand the data you can access to benefit your work and daily life. Moreover, in the years to come, it is expected that more and more powerful, user-friendly, diversified, and affordable data mining systems or components will be made available. Therefore, one can expect that everyone will have needs and the means for data mining. In other words, it is unlikely that data mining will remain reserved for today's traditional knowledge workers consisting of managers and business analysts. Instead, data mining will become increasingly available to everyone.

“But, what could I do at home with data mining?" Data mining can have multiple personal uses- For example, you might like to mine your family's medical history, identifying patterns relating to genetically related medical conditions, such as cancer or chromosome abnormalities. Such knowledge may help in

making decisions about your lifestyle and health. In the future, you may be able to mine the records of the companies you deal with in order to evaluate their service to you as a customer, or to choose the best companies to deal with, based on customer service. You could apply content-based text mining to search your e-mail messages, or automatically create a classification system to help organize your archived messages you could mine data on stocks and company performance to assist in your financial investments. Other examples include mining Web stores to find the best deal on a particular item or type of vacation. Thus, as data mining crosses the chasm and becomes more affordable, and with the- increased availability of personal computers and data on the Web, it is expected that data mining will become increasingly accessible to the general public and will eventually become a handy tool for everyone.

Is Data Mining a Threat to Privacy and Data Security?

With more and more information accessible in electronic forms and available on the Web, and with increasingly powerful data mining tools being developed and put into use, you may wonder, "is data mining a threat to my privacy and information security?" Like any other technology, data mining can be used for good or bad. Since data mining may disclose patterns and various kinds of knowledge that are difficult to find otherwise, it may pose a threat to privacy and information security if not done or used properly.

Most consumers don't mind providing companies with personal information if they think it will

enable the companies to better service their needs. For example, shoppers are usually happy to sign up for loyalty cards at-the local supermarket if it means they can get discounts in return,

Have you ever stopped to think about just, how much information is recorded about you, and what

that information says? Profiling information can be collected every time you use your credit card, debit card, supermarket loyalty card, or frequent flyer card, or apply for any of the above. It can be collected when you surf the Web, reply to an Internet newsgroup, subscribe to a magazine, rent a video, join a club, fill out a contest entry form, give information about your new baby (in order to receive coupons, free samples, or gifts), pay for prescription drugs, or present you medical care number when visiting the doctor. Clearly, the information that can easily be collected is not limited to our retail purchasing behavior, but may even reflect our hobbies as well as financial, medical, and insurance data. If you stop to think about this the next time you do any of the above actions, you

may get the feeling that “Big Brother" or “Big Banker" or “Big Business” is carefully watching you.

While the collection of our personal data may prove beneficial for companies and consumers, there

is also potential for its misuse. What if the data are used for other purposes such as, say, to help insurance companies determine your level of fat consumption based on the food items you purchase? One supermarket recently tried to use loyalty-card data to show that a shopper who slipped and fell was actually a heavy drinker (based on the amount of alcohol purchases). Although the case was dropped, it illustrates how data that are “invisibly” collected on consumers may be used against them.

While pondering the above, you may wonder:

“When I provide a company with information about myself, are these data going to be used in ways I don't expect?"

“Will the data be sold to other companies"? "Can I end out what is recorded about me?"

"How can I find out which companies have information about me?"

"Do I have the right or the means to refuse companies to use the profiling information they have

about me?"

"Are there any means set up by which I can correct any errors in the profile data recorded about me? What if' I Want to erase, complete, amend, or update the data?"

"Will the information about me be `anonymized,' or will it be traceable to me?" "How secure are the data?"

“How accountable is the company who collects or stores my data, if these data are stolen or misused?”

There are no easy answers to these questions. International guidelines, known as fair information practices, were established for data privacy protection and cover aspects relating to data collection, use, quality, openness, individual participation, and accountability. They include the following principles:

Purpose specification and use limitation:

The purposes for which personal data are collected should be specified at the time of collection, and

the data collected should not exceed the stated purpose. Data mining is typically a secondary purpose of the data collection. It has been argued that attaching a "disclaimer" that the data may also be used for mining is generally not accepted as sufficient disclosure of intent. Due to the exploratory nature of data mining, it is impossible to know what patterns may be discovered; therefore, there is no certainty over how they may be used.

Openness:

Individuals have the right to know what information is collected about them, who have access to the data and how the data are being used.

One social concern of data mining is the issue of privacy and information security. Opt-out policies, which allow consumers to specify limitations on the use of their personal data, are one approach toward data privacy protection, while data security-enhancing techniques can anonymize information for security and privacy.

Trends in Data Mining

The diversity of data, data mining tasks, and data mining approaches poses many challenging research issues in data mining. The development of efficient and effective data mining methods and systems, the construction of interactive and integrated data mining environments, the design of data mining languages, and the application of data mining techniques to solve large application problems are important tasks for data mining researchers and data mining system and application developers. Some of the trends in data mining that reflect the pursuit of these challenges are:

Application exploration

Data mining is increasingly used for the exploration of applications in other areas, such as financial analysis, telecommunications; biomedicine, wireless security and science.

Scalable and interactive data mining methods

Constraint-based mining handles huge amounts of data efficiently with added control by allowing the specification and use of constraints to guide data mining systems in their search for interesting patterns.

Web database systems

The Web database systems will ensure data availability, data mining portability, scalability, high performance, and an integrated information processing environment for multidimensional data analysis and exploration.

Standardization of data mining language

A standard data mining language will facilitate the systematic development of data mining solutions, improve interoperability among multiple data mining systems and functions, and promote the education and use of data mining systems in industry and society.

Visual data mining

Visual data mining is an effective way to discover knowledge from huge amounts of data.

New methods for mining complex types of data

New methods should be adopted for mining complex types of data

to bridge a huge gap between the needs for these applications and the available technology.

Biological data mining

Mining DNA and protein sequences, mining high dimensional microarray data, biological pathway and network analysis, link analysis across heterogeneous biological data, and

information integration of biological data by data mining are interesting topics for biological data mining research.

Data mining and software engineering

Further development of data mining methodologies for software debugging will enhance software robustness and bring new vigor to software engineering.

Web Mining

Due to vast amount of information available on the Web. Web content mining, Web log mining, and data mining services on the Internet becomes one of the most important and flourishing subfields in data mining.

Distributed data mining

Advances in distributed data mining methods are expected to work in distributed computing environments.

Real-time or time-critical data mining

Many applications involving stream data (such as e=commerce, Web mining, stock analysis, intrusion detection, mobile data mining, and data mining for counter terrorism) require dynamic data mining models to be built in real time. Graph modeling is also useful for analyzing links in Web structure mining.

Multi-relational and multi-database data mining

Multi-relational data mining methods search for patterns involving multiple tables from a relational database. Multi-database mining searches for patterns across multiple databases.

Privacy protection and information security in data mining

Privacy protection and information security is to be provided by the data mining system.

Need of data mining

The massive growth of data from terabytes to perabytes is due to the wide availability of data in automated form from various sources as WWW, Business, Science, Society and many more. But we are drowning in data but deficient of knowledge data is useless, if it cannot deliver knowledge. That is why data mining is gaining wide acceptance in today's world. A lot has been done in this field and lot more need to be done.