UNIT I

DATA MINING AND DATA PREPROCESSING

DATA MINING

- **Motivation**
- **Definition**
- **Datamining on Kinds of Data**
- **Functionalities**
- **Classification**
- **Data Mining Task primitives**
- **Major Issues in Data Mining**

DATA PREPROCESSING

- **Definition**
- **Data Cleaning**
- **Integration and Transformation**
- **Data Reduction**

MOTIVATION AND IMPORTANCE:

- Data Mining is defined as the procedure of extracting information from huge sets of data.
- Data mining is mining knowledge from data.
- The terminologies involved in data mining and then gradually moves on to cover topics such as knowledge discovery, query language, classification and prediction, decision tree induction, cluster analysis, and how to mine the Web.
- Here is a huge amount of data available in the Information Industry.
- This data is of no use until it is converted into useful information.
- It is necessary to analyse this huge amount of data and extract useful information from it.
- Extraction of information is not the only process we need to perform.
- Data mining also involves other processes such as Data Cleaning, Data Integration, Data Transformation, Data Mining, Pattern Evaluation and Data Presentation.

DEFINITION OF DATA MINING?

Data Mining is defined as extracting information from huge sets of data. In other words, we can say that data mining is the procedure of mining knowledge from data. The information or knowledge extracted so can be used for any of the following applications

- Market Analysis
- Fraud Detection
- Customer Retention
- Production Control
- Science Exploration

Major Sources of data: -

Business –Web, E-commerce, Transactions, Stocks - Science – Remote Sensing, Bio informatics, Scientific Simulation - Society and Everyone – News, Digital Cameras, You Tube * Need for turning data into knowledge – Drowning in data, but starving for knowledge.

Definition of Data Mining?

Extracting and 'Mining' knowledge from large amounts of data. "Gold Mining from rock or sand" is same as "Knowledge mining from data"

Other terms for Data Mining:

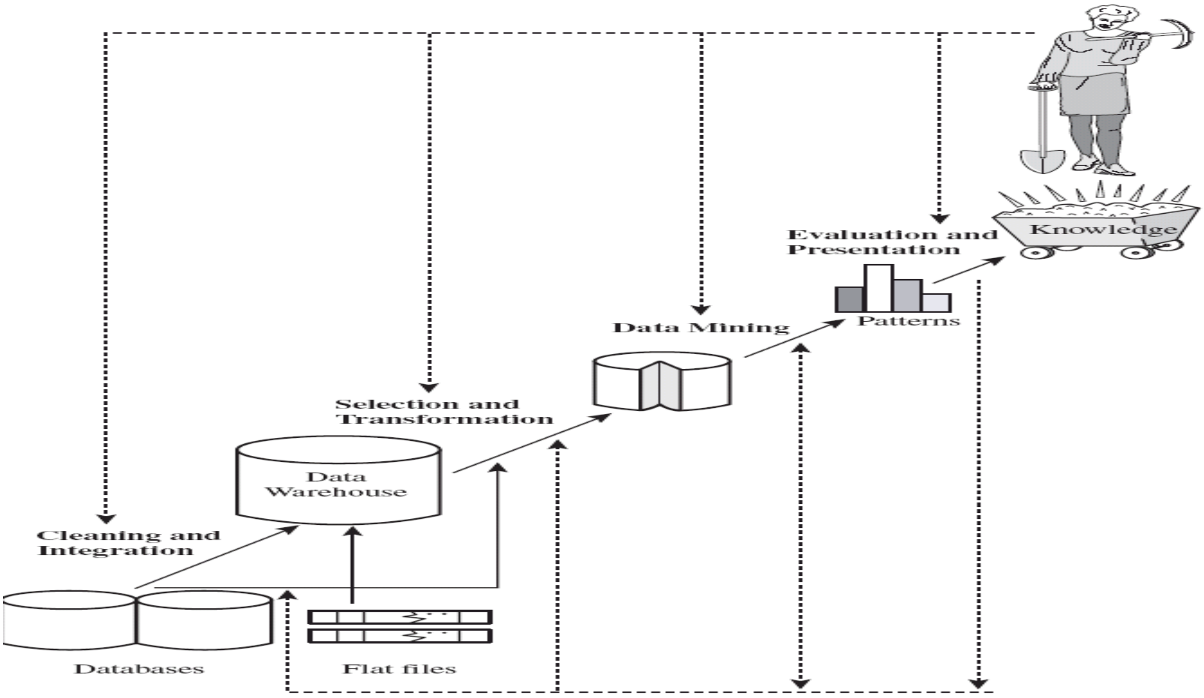
- Knowledge Mining
- Knowledge Extraction o Pattern Analysis

KNOWLEDGE DISCOVERY (KDD) PROCESS:

Several Key Steps:

- ▶ Data processing
  - ▶ **Data cleaning (**remove noise and inconsistent data)
  - ▶ **Data integration** (multiple data sources maybe combined)
  - ▶ **Data selection** (data relevant to the analysis task are retrieved from database)

Data transformation (data transformed or consolidated into forms)



- appropriate for mining)

  (Done with data preprocessing)

- **Data mining** (an essential process where intelligent methods are applied to extract

  data patterns)

- **Pattern evaluation** (identify the truly interesting patterns)

- **Knowledge presentation** (mined knowledge is presented to the user with visualization or representation techniques)

<u>DATA MINING ON WHAT KIND OF DATA? (TYPES OF DATA</u>):

RELATIONAL DATABASES:

- **A database system**, also called a database management system (DBMS), consists of a collection of interrelated data, known as a database, and a set of software programs to manage and access the data.
- **A relational database:** is a collection of tables, each of which is assigned a unique name.
- Each table consists of a set of attributes (*columns* or *fields*) and usually stores a large set of tuples (*records* or *rows*).
- Each tuple in a relational table represents an object identified by a unique *key* and described by a set of attribute values.

- **A semantic data model**, such as an entity-relationship (ER) data model, is often constructed for relational databases.
- An **ER data model** represents the database as a set of entities and their relationships.

DATA WAREHOUSE:

- A data warehouse is a repository of information collected from multiple sources, stored under a unified schema, and that usually resides at a single site.
- Data warehouses are constructed via a process of data cleaning, data integration, data transformation, data loading, and periodic data refreshing.
- The data are stored to provide information from a *historical perspective* and are typically *summarized*.
- A data warehouse is usually modelled by a multidimensional database structure.
- where each dimension corresponds to an attribute or a set of attributes in the schema, and each cell stores the value of some aggregate measure, such as *count* or *sales amount.*
- A data cube provides a multidimensional view of data and allows the precomputation and fast accessing of summarized data

What is the difference between a data warehouse and a data mart?":

- **A data warehouse** collects information about subjects that span an *entire organization*, and thus its scope is *enterprise-wide*.
- **A data mart** is a department subset of a data warehouse. It focuses on selected subjects, and thus its scope is *department-wide*.
- Data warehouse systems are well suited for on-line analytical processing, or OLAP.
- **Examples** of OLAP operations include drill-down and roll-up, which allow the user to view the data at differing degrees of summarization,

Transactional Databases:

- Transactional database consists of a file where each record represents a transaction.
- A transaction typically includes a unique transaction identity number (*trans ID*) and a list of the items making up the transaction.
- The transactional database may have additional tables associated with it, which contain other information regarding the sale, such as the date of the transaction, the customer ID number, the ID ,number of the salesperson and of the branch at which the sale occurred, and so on.

-

ADVANCED DATA AND INFORMATION SYSTEMS AND ADVANCED APPLICATIONS:

The new database applications include handling spatial data (such as maps), engineering design data (such as the design of buildings, system components, or integrated circuits), hypertext and multimedia data (including text, image, video, and audio data), time-related data (such as historical records or stock exchange data), stream data (such as video surveillance and sensor data, where data flow in and out like streams), and the World Wide Web (a huge, widely distributed information repository made available by the Internet).

OBJECT-RELATIONAL DATABASES:

- Object-relational databases are constructed based on an object-relational data model.
- This model extends the relational model by providing a rich data type for handling complex objects and object orientation object-relational databases are becoming increasingly popular in industry and applications.
- The object-relational data model inherits the essential concepts of object-oriented databases Each object has associated with it the following:
- **A set of variables** that describe the objects. These correspond to attributes in the entity relationship and relational models.
- **A set of messages** that the object can use to communicate with other objects, or with the rest of the database system.
- **A set of methods**, where each method holds the code to implement a message. Upon receiving a message, the method returns a value in response. **For instance**: the method for the message *get photo*(*employee*) will retrieve and return a photo of the given employee object.
- Objects that share a common set of properties can be grouped into an object class.
- Each object is an instance of its class. Object classes can be organized into class/subclass hierarchies so that each class represents properties that are common to objects in that class

TEMPORAL DATABASES, SEQUENCE DATABASES, AND TIME-SERIES DATABASES:

TEMPORAL DATABASE:

Typically stores relational data that include time-related attributes. These attributes may involve several timestamps, each having different semantics.

A sequence database stores sequences of ordered events, with or without a concrete notion of time.

Examples: include customer shopping sequences, Web click streams, and biological sequences. A time series database stores sequences of values or events obtained over repeated measurements of time (e.g., hourly, daily, weekly).

SPATIAL DATABASES AND SPATIOTEMPORAL DATABASES:

SPATIAL DATABASES:

- It contain spatial-related information. Examples include geographic (map) databases, very large-scale integration (VLSI) or computed-aided design databases, and medical and satellite image databases.
- Spatial data may be represented in raster format, consisting of *n*-dimensional bit maps or pixel maps.
- **Example:** a 2-D satellite image may be represented as raster data, where each pixel registers the rainfall in a given area.
- Maps can be represented in vector format, where roads, bridges, buildings, and lakes are represented as unions or overlays of basic geometric constructs, such as points, lines, polygons, and the partitions and networks formed by these components.

"What kind of data mining can be performed on spatial databases?" :

- Data mining may uncover patterns describing the characteristics of houses located near a specified kind of location, such as a park, for instance.
- A spatial database that stores spatial objects that change with time is called a spatiotemporal database, from which interesting information can be mined

Text Databases and Multimedia Databases

TEXT DATABASES:

- Databases that contain word descriptions for objects.
- These word descriptions are usually not simple keywords but rather long sentences or paragraphs, such as product specifications, error or bug reports, warning messages, summary reports, notes, or other documents.

- Text databases may be highly unstructured (such as some Web pages on the WorldWideWeb). Some text databases may be somewhat structured, that is, *semistructured* (such as e-mail messages and many HTML/XML Web pages).

- Text databases with highly regular structures typically can be implemented using relational database systems.

MULTIMEDIA DATABASES:

- Store image, audio, and video data. They are used in applications such as picture content-based retrieval, voice-mail systems, video-on-demand systems, the World Wide Web, and speech-based user interfaces that recognize spoken commands.
- Multimedia databases must support large objects, because data objects such as video can require gigabytes of storage.
- Specialized storage and search techniques are also required. Because video and audio data require real-time retrieval at a steady and predetermined rate in order to avoid picture or sound gaps and system buffer overflows, such data are referred to as continuous-media data.

HETEROGENEOUS DATABASES AND LEGACY DATABASES:

HETEROGENEOUS DATABASE:

It consists of a set of interconnected, autonomous component databases. The components communicate in order to exchange information and answer queries. Objects in one component database may differ greatly from objects in other component databases, making it difficult to assimilate their semantics into the overall heterogeneous database.

LEGACY DATABASE:

It is a group of ***heterogeneous databases*** that combines different kinds of data systems, such as relational or object-oriented databases, hierarchical databases, network databases, spreadsheets, multimedia databases, or file systems. The heterogeneous databases in a legacy database may be connected by intra or inter-computer networks.

DATA STREAMS:

- Many applications involve the generation and analysis of a new kind of data, called stream data, where data flow in and out of an observation platform (or window) dynamically.
- Such data streams have the following unique features:
- huge or possibly infinite volume, dynamically changing, flowing in and out in a fixed order, allowing only one or a small number of scans, and demanding fast (often real-time) response time.

Examples: streams include various kinds of scientific and engineering data, time-series data, and data produced in other dynamic environments, such as power supply, network traffic, stock exchange, telecommunications, Web click streams, video surveillance, and weather or environment monitoring.

THE WORLD WIDE WEB:

- The World Wide Web and its associated distributed information services, such as Yahoo!, Google, America Online, and AltaVista, provide rich, worldwide, on-line information services, where data objects are linked together to facilitate interactive access.

Example: understanding user access patterns will not only help improve system design (by providing efficient access between highly correlated objects.)

DATA MINING FUNCTIONALITIES—WHAT KINDS OF PATTERNS CAN BE MINED?:

Data mining functionalities are used to specify the kind of patterns to be found in data mining tasks. data mining tasks can be classified into two categories: descriptive and predictive. Descriptive mining tasks characterize the general properties of the data in the database. Predictive mining tasks perform inference on the current data in order to make predictions.

CONCEPT/CLASS DESCRIPTION: CHARACTERIZATION AND DISCRIMINATION:

- Data can be associated with classes or concepts.
- Example: *AllElectronics* store, classes of items for sale include *computers* and *printers*, and concepts of customers include ***bigSpenders*** and ***budgetSpenders*.**
- It can be useful to describe individual classes and concepts in summarized, concise, and yet precise terms. Such descriptions of a class or a concept are called class/concept descriptions. These descriptions can be derived via
- ***data characterization***, by summarizing the data of the class under study (often called the target class) in general terms,
- ***data discrimination***, by comparison of the target class with one or a set of comparative classes (often called the contrasting classes), or both data characterization and discrimination.

Data characterization:

- It is a summarization of the general characteristics or features of a target class of data.

- The data corresponding to the user-specified class are typically collected by a database query the output of data characterization can be presented in various forms.

Examples include pie charts, bar charts, curves, multidimensional data cubes, and multidimensional tables, including crosstabs.

Data discrimination:

- It is a comparison of the general features of target class data objects with the general features of objects from one or a set of contrasting classes.
- The target and contrasting classes can be specified by the user, and the corresponding data objects retrieved through database queries.

MINING FREQUENT PATTERNS, ASSOCIATIONS, AND CORRELATIONS:

Frequent patterns, as the name suggests, are patterns that occur frequently in data. There are many kinds of frequent patterns, including item sets, sub sequences, and substructures.

A *frequent itemset* typically refers to a set of items that frequently appear together in a transactional data set, such as Computer and Software.

Example: Association analysis. Suppose, as a marketing manager of *AllElectronics*, you would like to determine which items are frequently purchased together within the same transactions.

Example of such a rule, mined from the *AllElectronics* transactional database, is ***buys*(*X*;—*computer*‖) *buys*(*X*; —*software*‖) [*support* = 1%, *confidence* = 50%]**.

where *X* is a variable representing a customer. A confidence, or certainty, of 50% means that if a customer buys a computer, there is a 50% chance that she will buy software as well. A 1% support means that 1% of all of the transactions under analysis showed that computer and software were purchased together.

CLASSIFICATION AND PREDICTION:

Classification is the process of finding a model (or function) that describes and distinguishes data classes or concepts, for the purpose of being able to use the model to predict the class of objects whose class label is unknown.

*“**How is the derived model presented?”:***

The derived model may be represented in various forms, such as ***classification (IF-THEN) rules***, *decision trees*, *mathematical formulae*, or *neural networks*.

A decision tree is a flow-chart-like tree structure, where each node denotes a test on an attribute value, each branch represents an outcome of the test, and tree leaves represent classes or class distributions. Decision trees can easily be converted to classification rules.

A neural network, when used for classification, is typically a collection of neuron-like processing units with weighted connections between the units. There are many other methods for constructing classification models, such as naïve Bayesian classification, support vector machines, and *k*-nearest neighbour classification.

Whereas classification predicts categorical (discrete, unordered) labels, prediction models Continuous-valued functions. That is, it is used to predict missing or unavailable *numerical data values* rather than class labels. Although the term *prediction* may refer to both numeric prediction and class label prediction,

Cluster Analysis

Classification and prediction analyse class-labelled data objects, where as **clustering** analyzes data objects without consulting a known class label.

Outlier Analysis

A database may contain data objects that do not comply with the general behavior or model of the data. These data objects are outliers. Most data mining methods discard outliers as noise or exceptions.

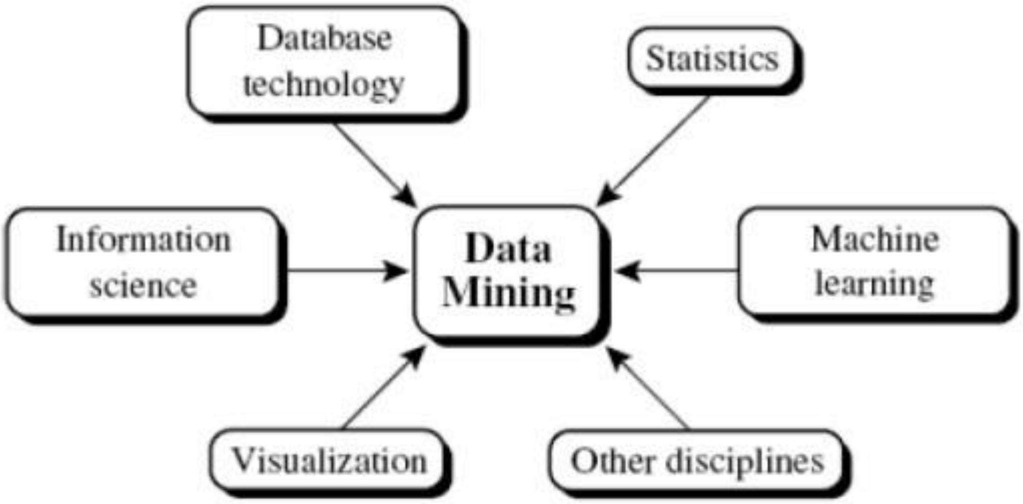
Evolution Analysis

Data evolution analysis describes and models regularities or trends for objects whose behavior changes over time. Although this may include characterization, discrimination, association and correlation analysis, classification, prediction, or clustering of *time related* data.

CLASSIFICATION OF DATA MINING SYSTEMS:

Data mining is an interdisciplinary field, the confluence of a set of disciplines, including database systems, statistics, machine learning, visualization, and information science.

Moreover, depending on the data mining approach used, techniques from other disciplines may be applied, such as neural networks, fuzzy and/or rough set theory, knowledge representation, inductive logic programming, or high-performance computing.



Data mining systems can be categorized according to various criteria, as follows:

Classification according to the *kinds of databases* mined:

A data mining system can be classified according to the kinds of databases mined. Database systems can be classified according to different criteria (such as data models, or the types of data or applications involved), each of which may require its own data mining technique.

Classification according to the *kinds of knowledge* mined:

Data mining systems can be categorized according to the kinds of knowledge they mine, that is, based on data mining functionalities, such as characterization, discrimination, association and correlation analysis, classification, prediction, clustering, outlier analysis, and evolution analysis.

Classification according to the *kinds of techniques* utilized:

Data mining systems can be categorized according to the underlying data mining techniques employed. These techniques can be described according to the degree of user interaction involved (e.g., autonomous systems, interactive exploratory systems, query-driven systems) or the methods of data analysis employed (e.g., database-oriented or data warehouse– oriented techniques, machine learning, statistics, visualization, pattern recognition, neural networks, and so on).

Classification according to the *applications adapted*:

Data mining systems can also be categorized according to the applications they adapt. **For example,** data mining systems may be tailored specifically for finance, telecommunications, DNA, stock markets, e-mail, and so on. Different applications often require the integration of application-specific methods.

DATA MINING TASK PRIMITIVES:

A data mining query is defined in terms of the following primitives:

Task-relevant data:

This is the database portion to be investigated. For example, suppose that you are a manager of All Electronics in charge of sales in the United States and Canada. In particular, you would like to study the buying trends of customers in Canada. Rather than mining on the entire database. These are referred to as relevant attributes

The kinds of knowledge to be mined:

This specifies the data mining functions to be performed, such as characterization, discrimination, association, classification, clustering, or evolution analysis. For instance, if studying the buying habits of customers in Canada.

Background knowledge:

Users can specify background knowledge, or knowledge about the domain to be mined. This knowledge is useful for guiding the knowledge discovery process, and for evaluating the patterns found. There are several kinds of background knowledge.

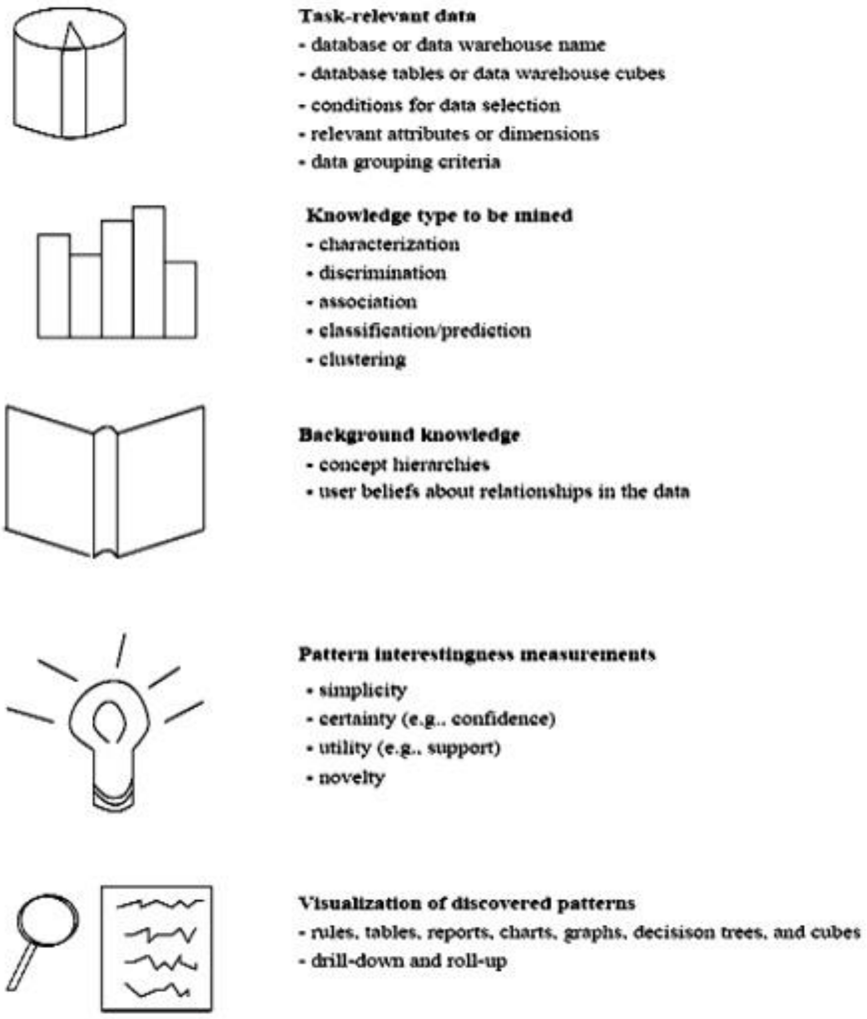
Interestingness measures:

These functions are used to separate uninteresting patterns from knowledge. They may be used to guide the mining process, or after discovery, to evaluate the discovered patterns. Different kinds of knowledge may have different interestingness measures.

Presentation and visualization of discovered patterns:

This refers to the form in which discovered patterns are to be displayed. Users can choose from different forms for knowledge presentation, such as rules, tables, charts, graphs, decision trees, and cubes.

Figure : Primitives for specifying a data mining task.



Task-relevant data
- database or data warehouse name
- database tables or data warehouse cubes
- conditions for data selection
- relevant attributes or dimensions
- data grouping criteria

Knowledge type to be mined
- characterization
- discrimination
- association
- classification/prediction
- clustering

Background knowledge
- concept hierarchies
- user beliefs about relationships in the data

Pattern interestingness measurements
- simplicity
- certainty (e.g., confidence)
- utility (e.g., support)
- novelty

Visualization of discovered patterns
- rules, tables, reports, charts, graphs, decisison trees, and cubes
- drill-down and roll-up

MAJOR ISSUES IN DATA MINING:

Mining different kinds of knowledge in databases. - The need of different users is not the same. And Different user may be in interested in different kind of knowledge. Therefore it is necessary for data mining to cover broad range of knowledge discovery task.

Interactive mining of knowledge at multiple levels of abstraction. **-** The data mining process needs to be interactive because it allows users to focus the search for patterns, providing and refining data mining requests based on returned results.

Incorporation of background knowledge. - To guide discovery process and to express the discovered patterns, the background knowledge can be used. Background knowledge may be used to express the discovered patterns not only in concise terms but at multiple level of abstraction.

Data mining query languages and ad hoc data mining. - Data Mining Query language that allows the user to describe ad hoc mining tasks, should be integrated with a data warehouse query language and optimized for efficient and flexible data mining. Presentation and visualization of data mining results. - Once the patterns are discovered it needs to be expressed in high level languages, visual representations. This representations should be easily understandable by the users.

Handling noisy or incomplete data. - The data cleaning methods are required that can handle the noise, incomplete objects while mining the data regularities. If data cleaning methods are not there then the accuracy of the discovered patterns will be poor.

Pattern evaluation. - It refers to interestingness of the problem. The patterns discovered should be interesting because either they represent common knowledge or lack novelty.

Efficiency and scalability of data mining algorithms. - In order to effectively extract the information from huge amount of data in databases, data mining algorithm must be efficient and scalable.

Parallel, distributed, and incremental mining algorithms. –

The factors such as huge size of databases, wide distribution of data,and complexity of data mining methods motivate the development of parallel and distributed data mining algorithms. These algorithm divide the data into partitions which is further processed parallel. Then the results from the partitions is merged. The incremental algorithms, updates databases without having mine the data again from scratch.

DATA PREPROCESSING:

Definition - What does Data Preprocessing mean?

Data preprocessing is a data mining technique that involves transforming raw data into an understandable format. Real-world data is often incomplete, inconsistent, and/or lacking in certain behaviors or trends, and is likely to contain many errors. Data preprocessing is a proven method of resolving such issues. Data preprocessing prepares raw data for further processing. Data preprocessing is used database-driven applications such as customer relationship management and rule-based applications (like neural networks). Data goes through a series of steps during preprocessing:

- **Data Cleaning:** Data is cleansed through processes such as filling in missing values, smoothing the noisy data, or resolving the inconsistencies in the data.

- **Data Integration:** Data with different representations are put together and conflicts within the data are resolved.
- **Data Transformation:** Data is normalized, aggregated and generalized.
- **Data Reduction:** This step aims to present a reduced representation of the data in a data warehouse.
- **Data Discretization:** Involves the reduction of a number of values of a continuous attribute by dividing the range of attribute intervals.

Why Data Pre-processing? Data preprocessing prepares raw data for further processing. The traditional data preprocessing method is reacting as it starts with data that is assumed ready for analysis and there is no feedback and impart for the way of data collection. The data inconsistency between data sets is the main difficulty for the data preprocessing.

1 . Data Cleaning.

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

(i). Missing values

1. 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.

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

3. 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.

4. 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.

5. 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.

6. 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.

(ii). Noisy data

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

1. Binning methods:

Binning methods smooth a sorted data value by consulting the neighbourhood", or values around it. The sorted values are distributed into a number of 'buckets', or bins. Because binning methods consult the neighbourhood of values, they perform local smoothing.

In this example, the data for price are first sorted and partitioned into equal-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

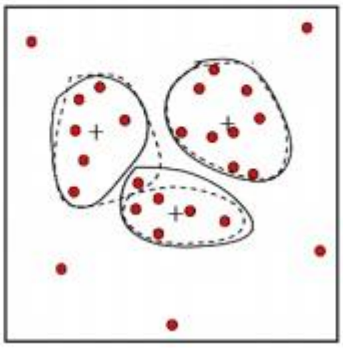
Bin 2: 21, 21, 24

Bin 3: 25, 25, 34

2. Clustering:

Outliers may be detected by clustering, where similar values are organized into groups or clusters.

Figure: Outliers may be detected by clustering analysis.



3. 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.

4. Regression: Data can be smoothed by fitting the data to a function, such as with regression.

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.

(iii). 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.

2. DATA TRANSFORMATION:

In data transformation, the data are transformed or consolidated into forms appropriate for mining. Data transformation can involve the following:

Normalization, where the attribute data are scaled so as to fall within a small specified range, such as -1.0 to 1.0, or 0 to 1.0.

There are three main methods for data normalization :

- **min-max normalization**
- **z-score normalization**

- **normalization by decimal scaling.**

(i).Min-max normalization performs a linear transformation on the original data. Suppose that minA and maxA are the minimum and maximum values of an attribute A. Min-max normalization maps a value v of A to v0 in the range [new minA; new maxA] by computing

$$v' = \frac{v - min_A}{max_A - min_A}(new\_max_A - new\_min_A) + new\_min_A.$$

(ii).z-score normalization (or zero-mean normalization), the values for an attribute A are normalized based on the mean and standard deviation of A. A value v of A is normalized to v0 by computing where mean A and stand dev A are the mean and standard deviation, respectively, of attribute A.

$$v' = \frac{v - mean_A}{stand\_dev_A}$$

(iii). **Normalization by decimal scaling** normalizes by moving the decimal point of values of attribute A. The number of decimal points moved depends on the maximum absolute value of A. A value v of A is normalized to v0by computing where j is the smallest integer such that

$$Max(|v'|) < 1.$$

Smoothing, which works to remove the noise from data? Such techniques include binning, clustering, and regression.

(i). Binning methods:

Binning methods smooth a sorted data value by consulting the neighbourhood, or values around it. The sorted values are distributed into a number of 'buckets', or bins. Because binning methods consult the neighbourhood of values, they perform local smoothing. Figure illustrates some binning techniques.

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

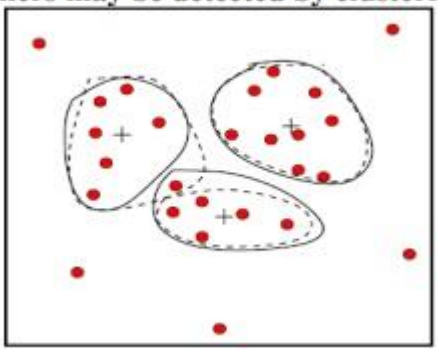
Bin 3: 25, 25, 34

(ii). 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.

Figure: Outliers may be detected by clustering analysis.



Aggregation: where summary or aggregation operations are applied to the data. For example, the daily sales data may be aggregated so as to compute monthly and annual total amounts.

Generalization of the data: where low level or 'primitive' (raw) data are replaced by higher level concepts through the use of concept hierarchies. For example, categorical attributes, like street, can be generalized to higher level concepts, like city or county.

3. DATA REDUCTION:

Data reduction techniques can be applied to obtain a reduced representation of the data set that is much smaller in volume, yet closely maintains the integrity of the original data.

Strategies for data reduction include the following:

Data cube aggregation:where aggregation operations are applied to the data in the construction of a data cube.

Dimension reduction: where irrelevant, weakly relevant or redundant attributes or dimensions may be detected and removed.

Data compression, where encoding mechanisms are used to reduce the data set size.

Numerosity reduction: where the data are replaced or estimated by alternative, smaller data representations such as parametric models (which need store only the model parameters instead of the actual data), or nonparametric methods such as clustering, sampling, and the use of histograms.

Discretization and concept hierarchy generation: where raw data values for attributes are replaced by ranges or higher conceptual levels. Concept hierarchies allow the mining of data at multiple levels of abstraction, and are a powerful tool for data mining.

Data Cube Aggregation

- The lowest level of a data cube
- the aggregated data for an individual entity of interest
- e.g., a customer in a phone calling data warehouse.
- Multiple levels of aggregation in data cubes
- Further reduce the size of data to deal with
- Reference appropriate levels
- Use the smallest representation which is enough to solve the task
- Queries regarding aggregated information should be answered using data cube, when possible

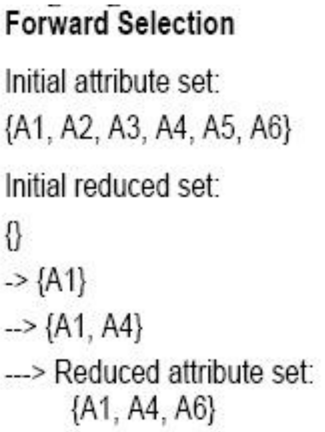
DIMENSIONALITY REDUCTION:

FEATURE SELECTION (ATTRIBUTE SUBSET SELECTION):

Select a minimum set of features such that the probability distribution of different classes given the values for those features is as close as possible to the original distribution given the values of all features reduce of patterns in the patterns, easier to understand

Heuristic methods:

Step-wise forward selection: The procedure starts with an empty set of attributes. The best of the original attributes is determined and added to the set. At each subsequent iteration or step, the best of the remaining original attributes is added to the set.

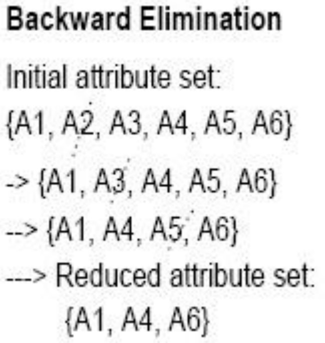


Forward Selection

Initial attribute set:
{A1, A2, A3, A4, A5, A6}

Initial reduced set:
{}
-> {A1}
--> {A1, A4}
---> Reduced attribute set:
{A1, A4, A6}

Step-wise backward elimination: The procedure starts with the full set of attributes. At each step, it removes the worst attribute remaining in the set.

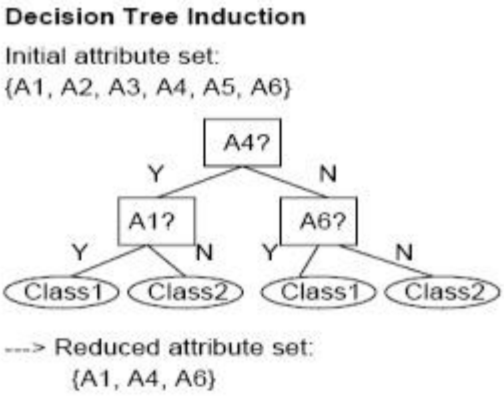


Backward Elimination

Initial attribute set:
{A1, A2, A3, A4, A5, A6}
-> {A1, A3, A4, A5, A6}
--> {A1, A4, A5, A6}
---> Reduced attribute set:
{A1, A4, A6}

Combination forward selection and backward elimination: The step-wise forward selection and backward elimination methods can be combined, where at each step one selects the best attribute and removes the other.

Decision tree induction: Decision tree algorithms, such as ID3 and C4.5, were originally intended for classifcation. Decision tree induction constructs a flow-chart-like structure where each internal (non-leaf) node denotes a test on an attribute, each branch corresponds to an outcome of the test, and each external (leaf) node denotes a class prediction. At each node, the algorithm chooses the ―best" attribute to partition the data into individual classes.



DATA COMPRESSION:

- In data compression, data encoding or transformations are applied so as to obtain a reduced or ‖compressed" representation of the original data.
- If the original data can be reconstructed from the compressed data without any loss of information, the data compression technique used is called lossless.
- we can reconstruct only an approximation of the original data, then the data compression technique is called lossy.

The two popular and effective methods of lossy data compression**:**

WAVELET TRANSFORMS, AND PRINCIPAL COMPONENTS ANALYSIS:

WAVELET TRANSFORMS:

- The discrete wavelet transform (DWT) is a linear signal processing technique that, when applied to a data vector D, transforms it to a numerically different vector, D0, of wavelet coefficients. The two vectors are of the same length.
- The DWT is closely related to the discrete Fourier transform (DFT), a signal processing technique involving sines and cosines. In general, however, the DWT achieves better lossy compression.
- The general algorithm for a discrete wavelet transform is as follows.
- The length, L, of the input data vector must be an integer power of two. This condition can be met by padding the data vector with zeros, as necessary.
- Each transform involves applying two functions. The first applies some data smoothing, such as a sum or weighted average. The second performs a weighted difference.

- The two functions are applied to pairs of the input data, resulting in two sets of data of length L=2. In general, these respectively represent a smoothed version of the input data, and the high-frequency content of it.
- The two functions are recursively applied to the sets of data obtained in the previous loop, until the resulting data sets obtained are of desired length.
- A selection of values from the data sets obtained in the above iterations are designated the wavelet coefficients of the transformed data.

PRINCIPAL COMPONENTS ANALYSIS:

Principal components analysis (PCA) searches for c k-dimensional orthogonal vectors that can best be used to represent the data, where c << N. The original data is thus projected onto a much smaller space, resulting in data compression. PCA can be used as a form of dimensionality reduction. The initial data can then be projected onto this smaller set.

The basic procedure is as follows.

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

NUMEROSITY REDUCTION REGRESSION AND LOG-LINEAR MODELS:

Regression and log-linear models can be used to approximate the given data. In linear regression, the data are modelled to fit a straight line. For example, a random variable, Y (called a response variable), can be modelled as a linear function of another random variable, X (called a predictor variable), with the equation where the variance of Y is assumed to be constant. These coefficients can be solved for by the method of least squares, which minimizes the error between the actual line separating the data and the estimate of the line.

Multiple regression is an extension of linear regression allowing a response variable Y to be modelled as a linear function of a multidimensional feature vector.

Log-linear models approximate discrete multidimensional probability distributions. The method can be used to estimate the probability of each cell in a base cuboid for a set of discretized attributes, based on the smaller cuboids making up the data cube lattice

HISTOGRAMS

A histogram for an attribute A partitions the data distribution of A into disjoint subsets, or buckets. The buckets are displayed on a horizontal axis, while the height (and area) of a bucket typically reects the average frequency of the values represented by the bucket.

Equal width: In an equi-width histogram, the width of each bucket range is constant .

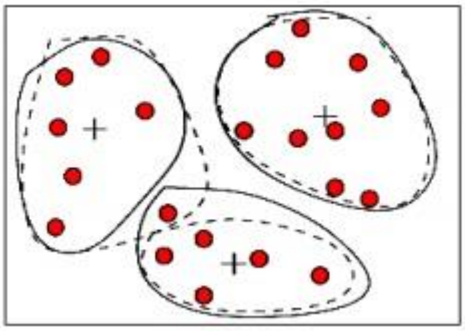
Equal-depth (or equi-height): In an equal-depth histogram, the buckets are created so that, roughly, the frequency of each bucket is constant (that is, each bucket contains roughly the same number of contiguous data samples).

V-Optimal: If we consider all of the possible histograms for a given number of buckets, the V-optimal histogram is the one with the least variance. Histogram variance is a weighted sum of the original values that each bucket represents, where bucket weight is equal to the number of values in the bucket.

MaxDiff: In a MaxDiff histogram, we consider the difference between each pair of adjacent values. A bucket boundary is established between each pair for pairs having the Beta largest differences, where Beta-1 is user-specified.

Clustering:

Clustering techniques consider data tuples as objects. They partition the objects into groups or clusters, so that objects within a cluster are ―similar" to one another and ―dissimilar" to objects in other clusters. Similarity is commonly defined in terms of how ―close" the objects are in space, based on a distance function. The ―quality" of a cluster may be represented by its diameter, the maximum distance between any two objects in the cluster. Centroid distance is an alternative measure of cluster quality, and is defined as the average distance of each cluster object from the cluster centroid.



SAMPLING:

Sampling can be used as a data reduction technique since it allows a large data set to be represented by a much smaller random sample (or subset) of the data. Suppose that a large data set, D, contains N tuples. Let's have a look at some possible samples for D.

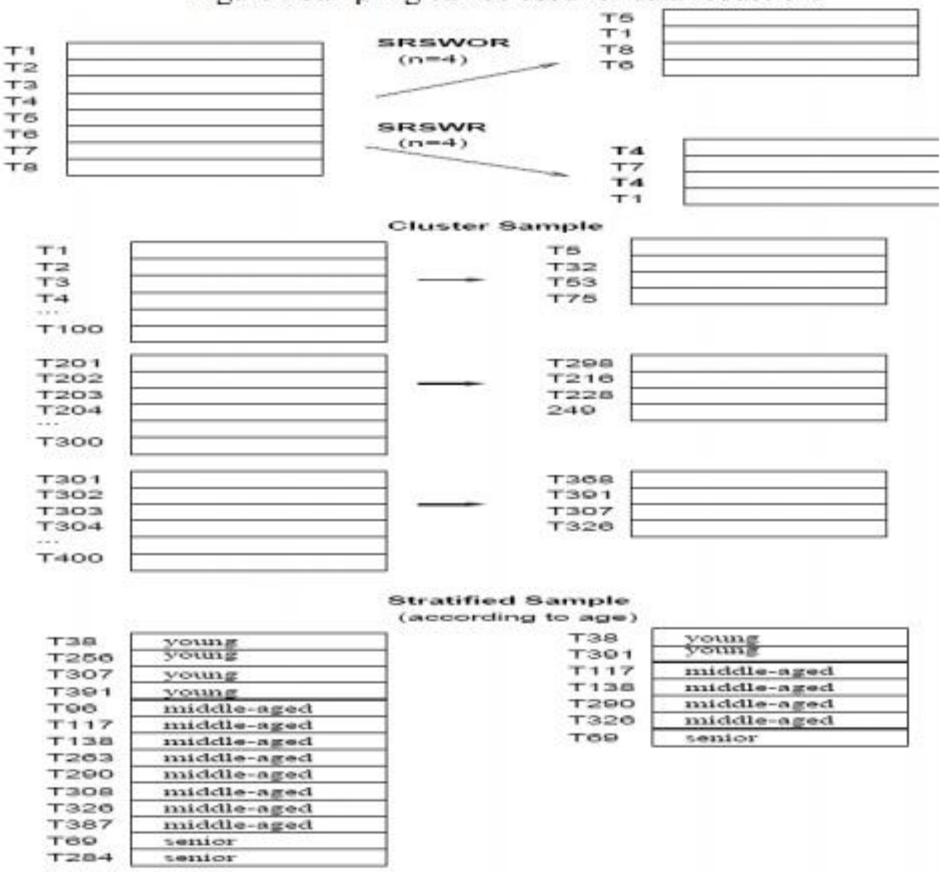
Simple random sample without replacement (SRSWOR) of size n: This is created by drawing n of the N tuples from D (n < N), where the probably of drawing any tuple in D is 1=N, i.e., all tuples are equally likely.

Simple random sample with replacement (SRSWR) of size n: This is similar to SRSWOR, except that each time a tuple is drawn from D, it is recorded and then replaced. That is, after a tuple is drawn, it is placed back in D so that it may be drawn again.

Cluster sample: If the tuples in D are grouped into M mutually disjoint clusters", then a SRS of m clusters can be obtained, where m < M. A reduced data representation can be obtained by applying, say, SRSWOR to the pages, resulting in a cluster sample of the tuples.

Stratified sample: If D is divided into mutually disjoint parts called —strata", a stratified sample of D is generated by obtaining a SRS at each stratum. This helps to ensure a representative sample, especially when the data are skewed. For example, a stratified sample may be obtained from customer data, where stratum is created for each customer age group.

Figure : Sampling can be used for data reduction.



UNIT II

DATA WAREHOUSING

- Multidimensional Data Model
- Data Warehouse Architecture
- Data Warehouse Implementation
- From Data Warehousing to Data Mining
- Online Analytical Processing
- Online Analytical Mining

MULTIDIMENSIONAL DATA MODEL:

- Multidimensional data model stores data in the form of data cube. Mostly, data warehousing supports two or three-dimensional cubes.
- A data cube allows data to be viewed in multiple dimensions.
- A dimensions are entities with respect to which an organization wants to keep records.

-For example in store sales record, dimensions allow the store to keep track of things like monthly sales of items and the branches and locations.

- A multidimensional databases helps to provide data-related answers to complex business queries quickly and accurately.
- Data warehouses and Online Analytical Processing (OLAP) tools are based on a multidimensional data model.
- OLAP in data warehousing enables users to view data from different angles and dimensions.

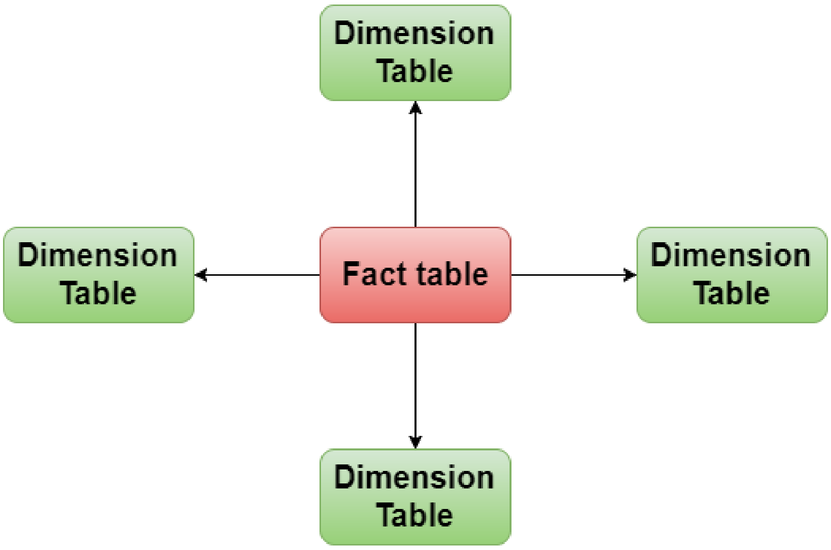
Schemas for Multidimensional Data Model :-

- Star Schema
- Snowflakes Schema
- Fact Constellations Schema

Star Schemas for Multidimensional Model:

- The simplest data warehouse schema is star schema because its structure resembles a star.
- Star schema consists of data in the form of facts and dimensions.

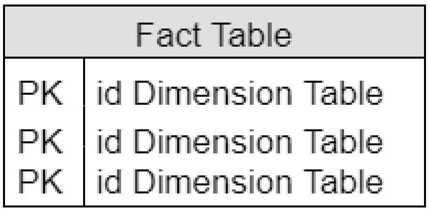
- The fact table present in the center of star and points of the star are the dimension tables.
- In star schema fact table contain a large amount of data, with no redundancy.
- Each dimension table is joined with the fact table using a primary or foreign key.



STAR SCHEMAS FOR MULTIDIMENSIONAL MODAL:

Fact Tables:

A fact table has two types of columns: one column of foreign keys (pointing to the dimension tables) and other of numeric values.



Fact Table	
PK	id Dimension Table
PK	id Dimension Table
PK	id Dimension Table

Dimension Tables:

Dimension table is generally small in size as compared to a fact table. The primary key of a dimension table is a foreign key in a fact table.

Example of Dimension Tables:-

- Time dimension table
- Product dimension table
- Employee dimension table
- Geography dimension table

- The main characteristics of star schema are that it is easy to understand and small number of tables can join.

SNOWFLAKE SCHEMAS FOR MULTIDIMENSIONAL MODEL:

- The snowflake schema is a more complex than star schema because dimension tables of the snowflake are normalized.
- The snowflake schema is represented by centralized fact table which is connected to multiple dimension table and this dimension table can be normalized into additional dimension tables.

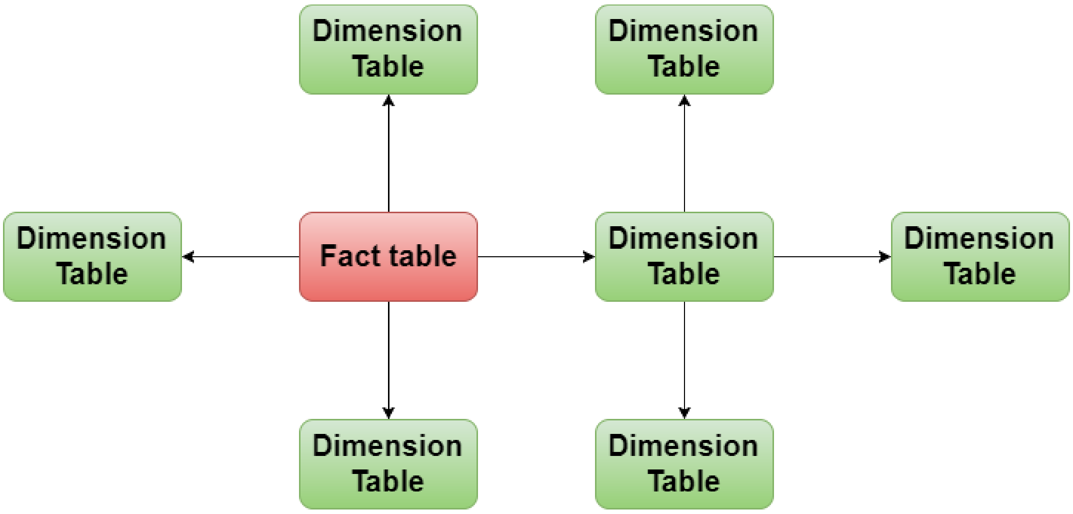


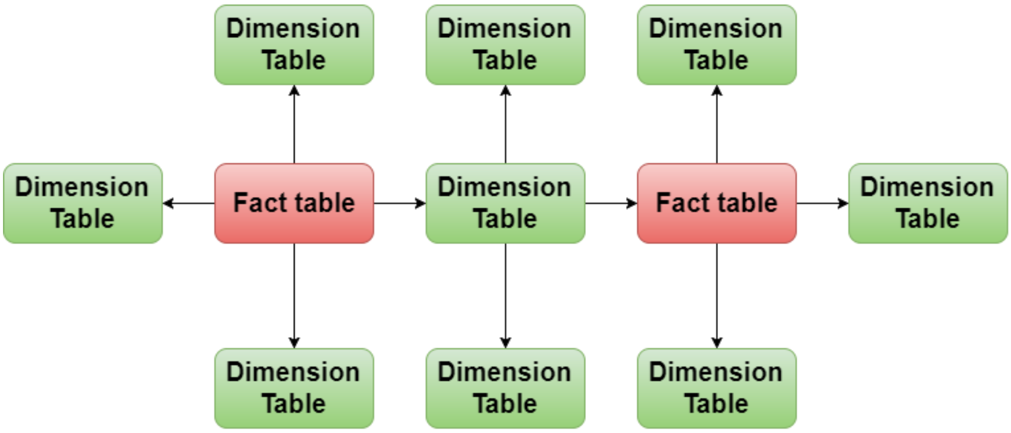
Fig:Snowflake Schemas for Multidimensional Model

- The major difference between the snowflake and star schema models is that the dimension tables of the snowflake model are normalized to reduce redundancies.

FACT CONSTELLATION SCHEMAS FOR MULTIDIMENSIONAL MODEL:

- A fact constellation can have multiple fact tables that share many dimension tables.
- This type of schema can be viewed as a collection of stars, Snowflak

and hence is called a galaxy schema or a fact constellation.



Fact constellation Schemas for Multidimensional Modal:

- The main disadvantage of fact constellation schemas is its more complicated design.

DATA WAREHOUSE ARCHITECTURE:

Data warehouse is a subject oriented, integrated, time-variant, and non-volatile collection of data. This data helps analysts to take informed decisions in an organization.

Steps for the Design and Construction of Data Warehouse:

- This subsection presents a business analysis framework for data warehouse design. The basic steps involved in the design process are also described.
- The Design of a Data Warehouse:
  A Business Analysis Framework Four different views regarding the design of a data warehouse must be considered:
  - the top-down view,
  - the data source view,
  - the data warehouse view,
  - the business query view.

• The top-down view allows the selection of relevant information necessary for the data warehouse.

• The data source view exposes the information being captured, stored and managed by operational systems.

- The data warehouse view includes fact tables and dimension tables

- Finally the business query view is the Perspective of data in the data warehouse from the viewpoint of the end user.

THREE-TIER DATA WAREHOUSE ARCHITECTURE :

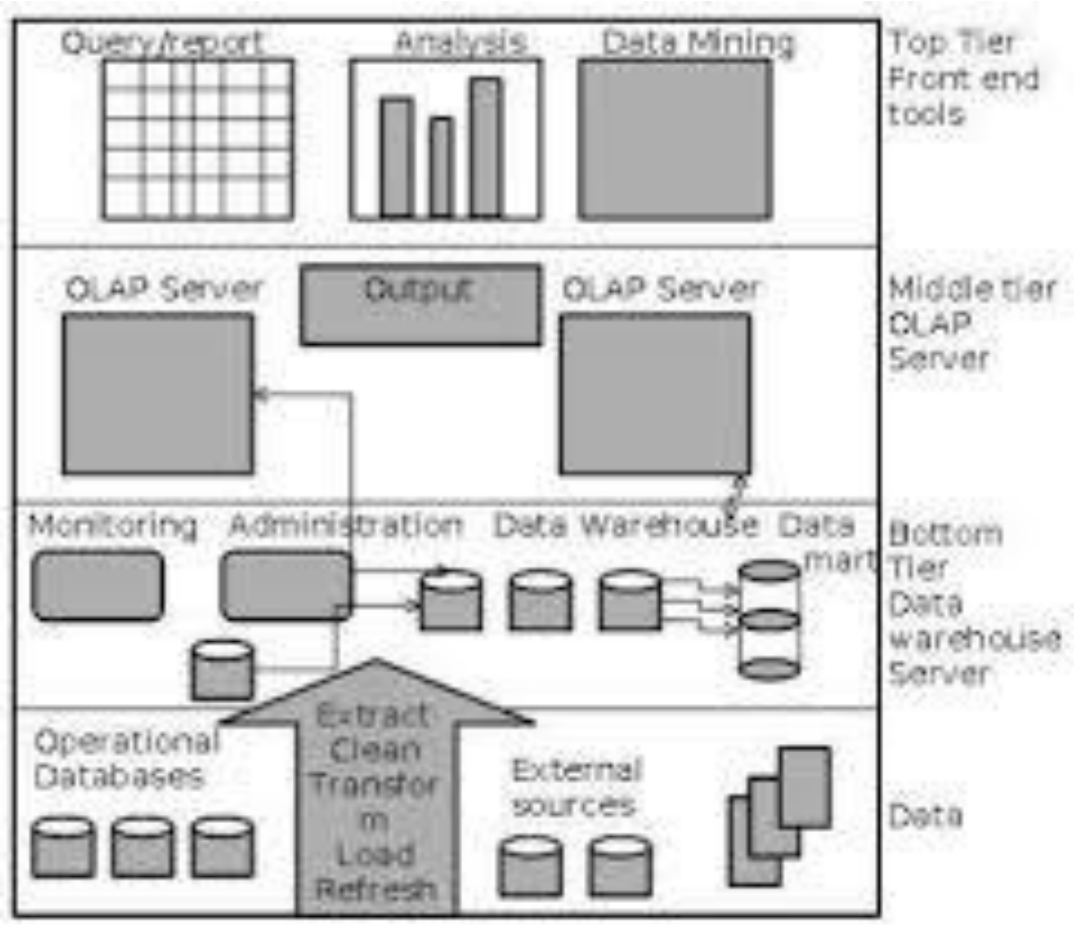
- **The bottom tier** is ware-house database server which is almost always a relational database system.
- **The middle tier** is an OLAP server which is typically implemented using either

  (1) a Relational OLAP (ROLAP) model.

  (2) a Multidimensional OLAP (MOLAP) model.

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

Three-tier Data warehouse architecture



From the architecture point of view, there are three data warehouse models:

- ➢ the enterprise warehouse.
- ➢ the data mart.

- ➢ the virtual warehouse.

- **Enterprise warehouse:**
  - An enterprise warehouse collects all of the information about subjects spanning the entire organization.
  - It provides corporate-wide data integration, usually from one or more operational systems or external information providers, and is cross-functional in scope.
  - It typically contains detailed data as well as summarized data, 13 and can range in size from a few gigabytes to hundreds of gigabytes, terabytes, or beyond.

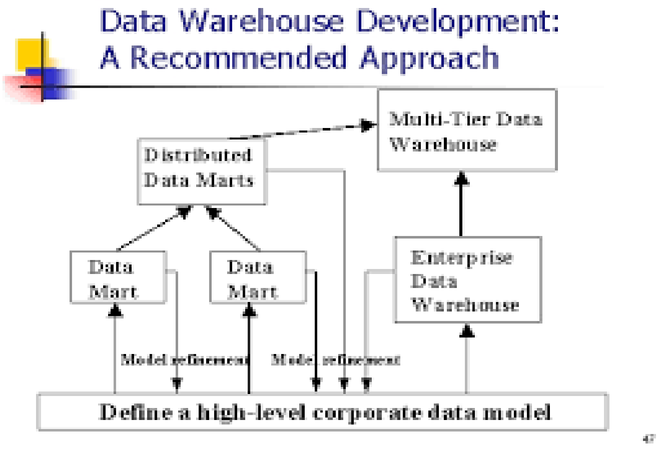
- **Data mart:**
  - A data mart contains a subset of corporate-wide data that is of value to a specific group of users.
  - The scope is connected to specific, selected subjects.
  - **For example**:a marketing data mart may connect its subjects to customer, item, and sales.
  - The data contained in data marts tend to be summarized. Depending on the source of data, data marts can be categorized into the following two classes:

(i).**Independent data marts** are sourced from data captured from one or more operational systems or external information providers, or from data generated locally within a particular department or geographic area.

(ii).**Dependent data marts** are sourced directly from enterprise data warehouses.

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

Figure: A recommended approach for data warehouse development. Data warehouse Back-End Tools and Utilities The ETL (Extract Transformation Load) process



Data Warehouse Development: A Recommended Approach

Multi-Tier Data Warehouse

Distributed Data Marts

Data Mart

Data Mart

Enterprise Data Warehouse

Model refinement

Model refinement

Define a high-level corporate data model

47

TYPES OF OLAP SERVERS:

ROLAP versus MOLAP versus HOLAP :

1. Relational OLAP (ROLAP) ν Use relational or extended-relational DBMS to store and manage warehouse data and OLAP middle ware to support missing pieces include optimization of DBMS backend, implementation of aggregation navigation logic, and additional tools and services ν greater scalability

2. Multidimensional OLAP (MOLAP) ν Array-based multidimensional storage engine (sparse matrix techniques) ν fast indexing to pre-computed summarized data

3. Hybrid OLAP (HOLAP) ν User flexibility, e.g., low level: relational, high-level: array

4. Specialized SQL servers ν specialized support for SQL queries over star/snowflake schemas.

DATA WAREHOUSE IMPLEMENTATION:

Efficient Computation of Data Cubes Data cube can be viewed as a lattice of cuboids

- The bottom-most cuboid is the **base cuboid**
- The top-most cuboid (**apex**) contains only one cell

- How many cuboids in an n-dimensional cube with L levels? Materialization of data cube
- Materialize every (cuboid) (full materialization), none (no materialization), or some (partial materialization)

- Selection of which cuboids to materialize

- Based on size, sharing, access frequency, etc. Cube Operation

- Cube definition and computation in DMQL define cube sales [item, city, year]: sum(sales_in_dollars) compute cube sales

- Transform it into a SQL-like language (with a new operator cube by, introduced by Gray et al. '96)

SELECT item, city, year, SUM (amount) FROM SALES CUBE BY item, city, year

- Need compute the following Group-Bys (date, product, customer), (date,product),(date, customer), (product, customer), (date), (product), (customer) 23 Cube Computation:

ROLAP-Based Method

- Efficient cube computation methods
  - ❖ ROLAP-based cubing algorithms (Agarwal et al'96)
  - ❖ Array-based cubing algorithm (Zhao et al'97)
  - ❖ Bottom-up computation method (Bayer & Ramarkrishnan'99)

- **ROLAP-based cubing algorithms :**
  - ❖ Sorting, hashing, and grouping operations are applied to the dimension attributes in order to reorder and cluster related tuples
  - ❖ Grouping is performed on some sub aggregates as a ―partial grouping step‖
  - ❖ Aggregates may be computed from previously computed aggregates, rather than from the base fact table Multi-way Array Aggregation for Cube
  - ❖ Computation
  - ❖ Partition arrays into chunks (a small sub cube which fits in memory).
  - ❖ Compressed sparse array addressing: (chunk_id, offset) • Compute aggregates in multiway by visiting cube cells in the order which minimizes the number of times to visit each cell, and reduces memory access and storage cost.

Indexing OLAP data:

The bitmap indexing method is popular in OLAP products because it allows quick searching in data cubes.

The bitmap index:

- It is an alternative representation of the record ID (RID) list.
- In the bitmap index for a given attribute, there is a distinct bit vector, By, for each value v in the domain of the attribute.
- If the domain of a given attribute consists of n values, then n bits are needed for each entry in the bitmap index.

Join index:

- The join indexing method gained popularity from its use in relational database query processing.
- Traditional indexing maps the value in a given column to a list of rows having that value.
- In contrast, join indexing registers the joinable rows of two relations from a relational database.

  For example: if two relations R(RID;A) and S(B; SID) join on the attributes A and B, then the join index record contains the pair (RID; SID), where RID and SID are record identifiers from the R and S relations, respectively. Efficient processing of OLAP queries

1. Determine which operations should be performed on the available cuboids. This involves transforming any selection, projection, roll-up (group-by) and drill-down operations specified in the query into corresponding SQL and/or OLAP operations. For example, slicing and dicing of a data cube may correspond to selection and/or projection operations on a materialized cuboid.

2. Determine to which materialized cuboid(s) the relevant operations should be applied. This involves identifying all of the materialized cuboids that may potentially be used to answer the query.

OLAP:

- Online Analytical Processing is based on the multidimensional data model that allow user to extract and view data from different points of view.
- OLAP data stored in multidimensional data.

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

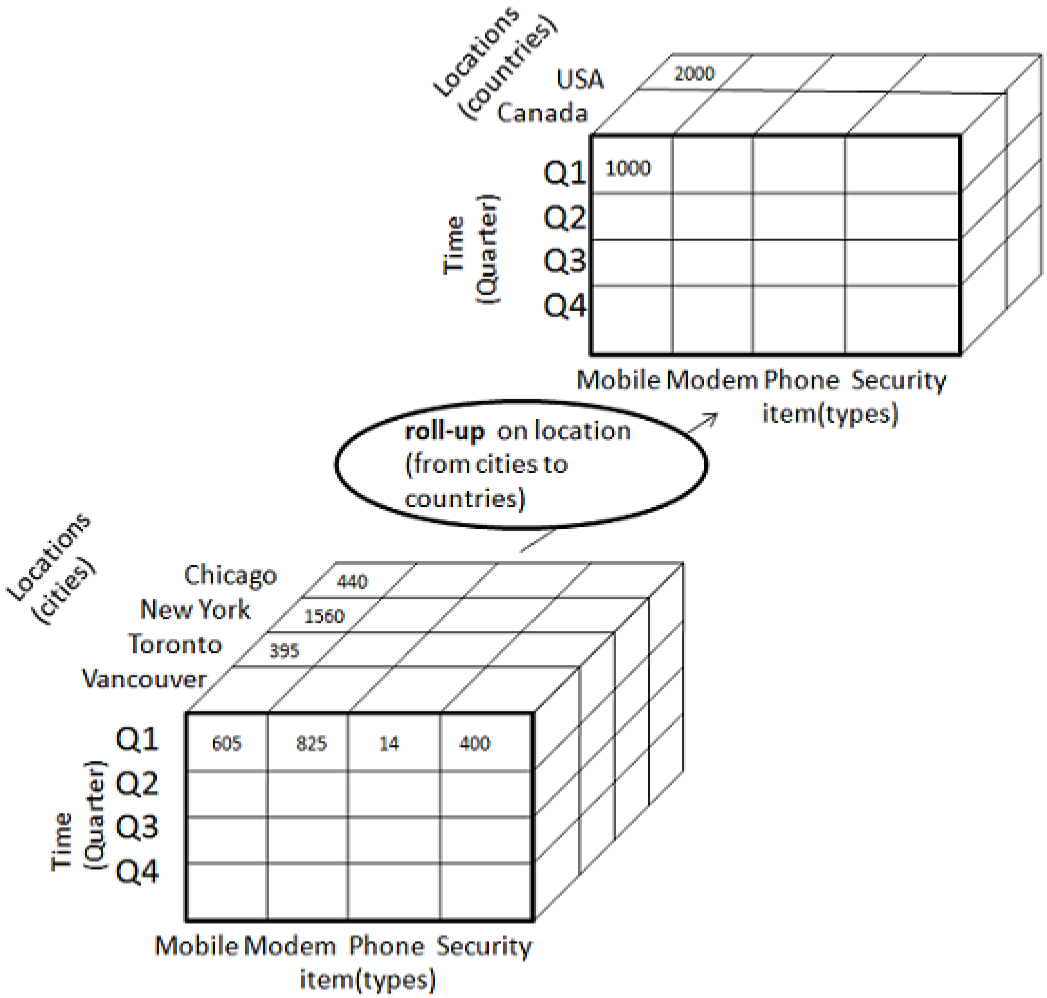
- Drill-down
- Slice and dice
- Pivot (rotate)

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.

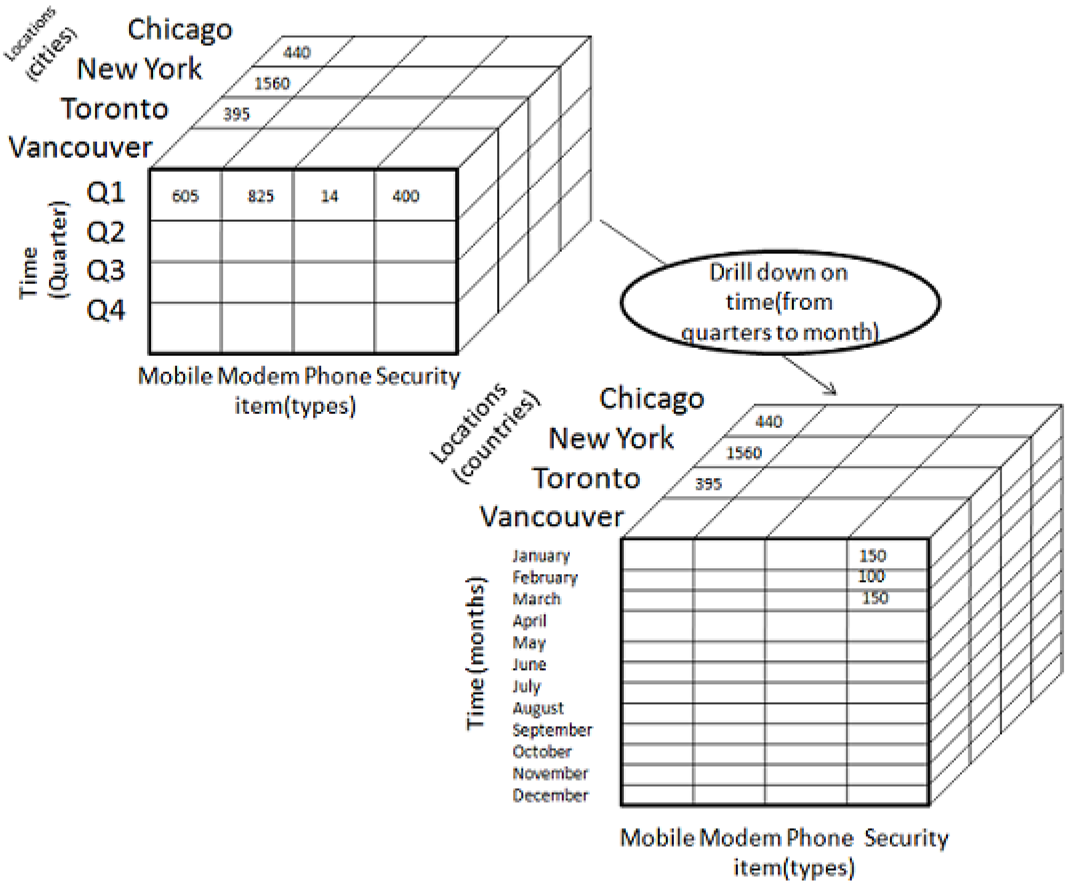
- When roll-up is performed, one or more dimensions from the data cube are removed.

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.

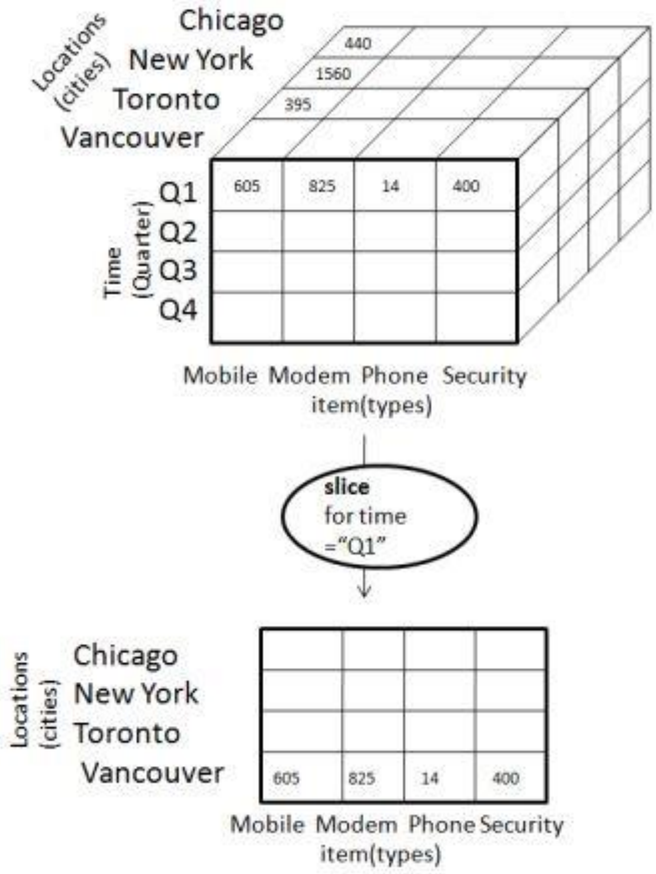
The following diagram illustrates how drill-down works −



- 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

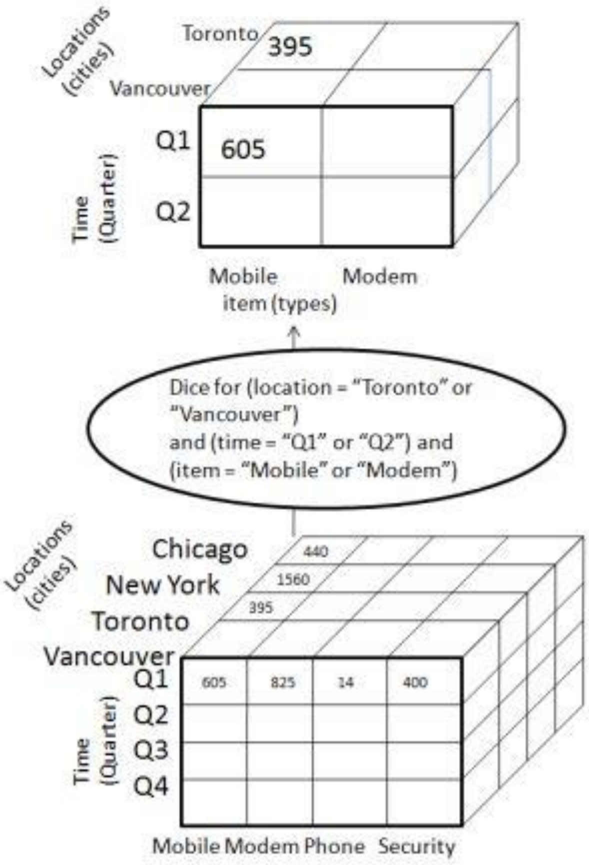
The slice operation selects one particular dimension from a given cube and provides a new sub-cube. Consider the following diagram that shows how slice works.



- 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.

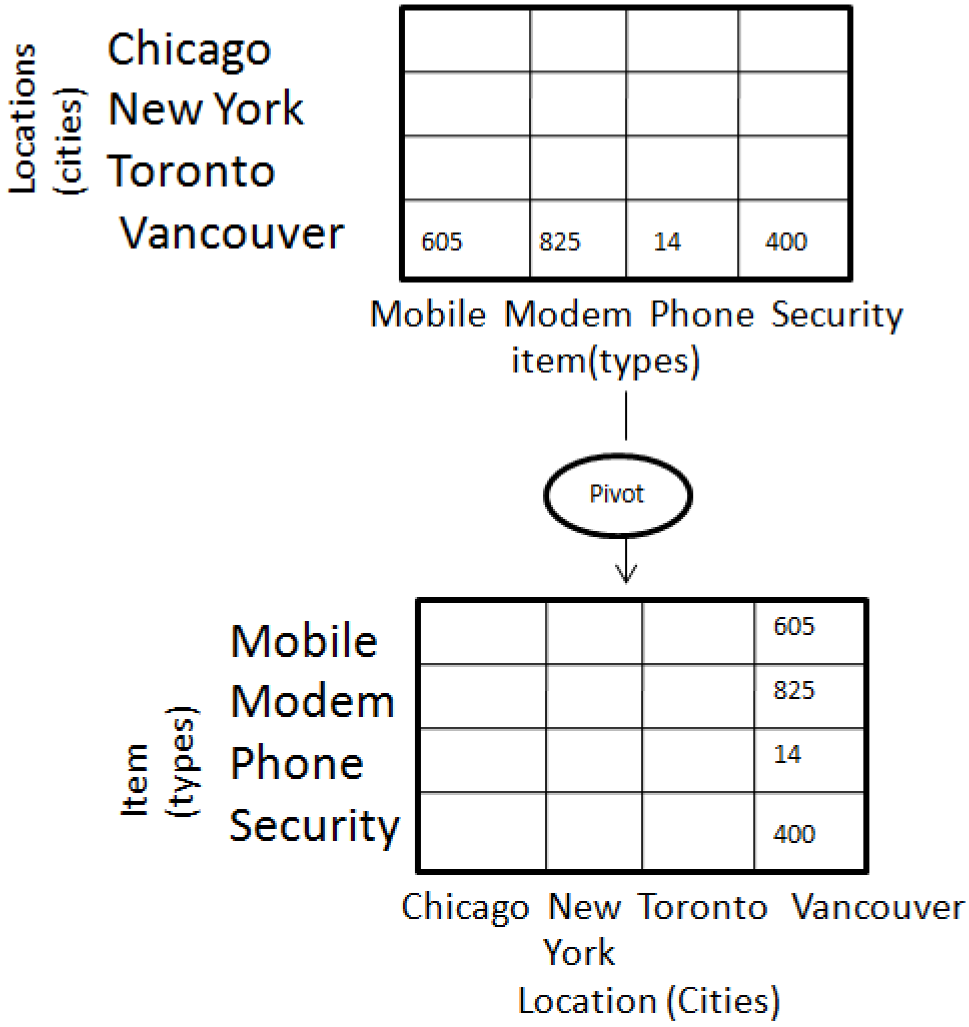


The dice operation on the cube based on the following selection criteria involves three dimensions.

- (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.



FROM DATA WAREHOUSING TO DATA MINING:

“How do data warehousing and OLAP relate to data *mining*.We also introduce on-line analytical mining (OLAM), a powerful paradigm that integrates OLAP with data mining technology.

DATA WAREHOUSE USAGE:

Three kinds of data warehouse applications are

- Information processing vs supports querying, basic statistical analysis, and reporting using crosstabs, tables, charts and graphs .

- Analytical processing νs multidimensional analysis of data warehouse data ν supports basic OLAP operations, slice-dice, drilling, pivoting
- Data mining νs knowledge discovery from hidden patterns ν supports associations, constructing analytical models, performing classification and prediction, and presenting the mining results using visualization tools. ν Differences among the three tasks.

FROM ON-LINE ANALYTICAL PROCESSING TO ON LINE ANALYTICAL MINING (OLAM):

From On-Line Analytical Processing to On Line Analytical Mining (OLAM)is called from data warehousing to data mining From on-line analytical processing to on-line analytical mining.

On-Line Analytical Mining (OLAM) (also called OLAP mining), which integrates online analytical processing (OLAP) with data mining and mining knowledge in multidimensional databases, is particularly important for the following reasons.

- **High quality of data in data warehouses:**
  Most data mining tools need to work on integrated, consistent, and cleaned data, which requires costly data cleaning, data transformation and data integration as preprocessing steps. A data warehouse constructed by such preprocessing serves as a valuable source of high quality data for OLAP as well as for data mining.
- **Available information processing infrastructure surrounding data warehouses:**
  Comprehensive information processing and data analysis infrastructures have been or will be systematically constructed surrounding data warehouses, which include accessing, integration, consolidation, and transformation of multiple, heterogeneous databases ODBC/OLEDB connections, Web-accessing and service facilities, reporting and OLAP analysis tools.
- **OLAP-based exploratory data analysis**:

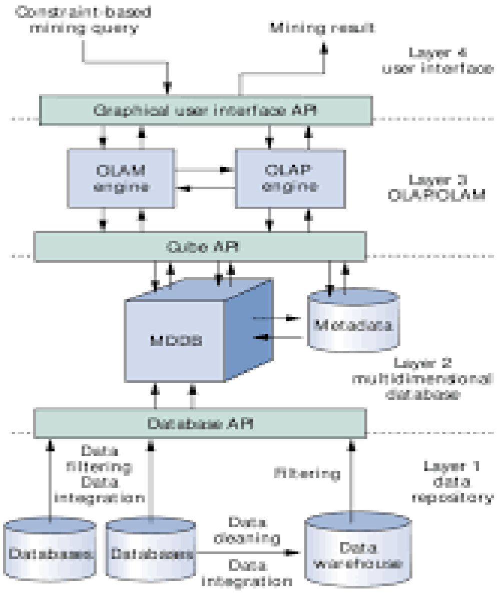
  Effective data mining needs exploratory data analysis. A user will often want to traverse through a database, select portions of relevant data, analyse them at differ

  rent granularities, and present knowledge/results in different forms. On-line analytical mining provides facilities for data mining on different subsets of data and at different levels of abstraction, by drilling, pivoting, filtering, dicing and slicing on a data cube and on some intermediate data mining results.

- **On-line selection of data mining functions:**
  By integrating OLAP with multiple data mining functions, on-line analytical mining provides users with the exibility to select desired data mining functions and swap data mining tasks dynamically.

ARCHITECTURE FOR ON-LINE ANALYTICAL MINING(OLAM):

- An OLAM engine performs analytical mining in data cubes in a similar manner as an OLAP engine performs on-line analytical processing.
- An integrated OLAM and OLAP architecture is shown in Figure, where the OLAM and OLAP engines both accept users' on-line queries via a User GUI API and work with the data cube in the data analysis via a Cube API.
- A metadata directory is used to guide the access of the data cube. The data cube can be constructed by accessing and/or integrating multiple databases and/or by filtering a data warehouse via a Database API which may support OLEDB or ODBC connections.
- Since an OLAM engine may perform multiple data mining tasks, such as concept description, association, classification, prediction, clustering, time-series analysis,etc., it usually consists of multiple, integrated data mining modules and is more sophisticated than an OLAP engine.



UNIT III

FRQUENT PATTERN ASSOCIATION RULE AND CALSSIFICATION

- The Apriori Algorithm
- Definition of Classification and Prediction
- Classification by Decision Tree Induction
- Bayesian Classification
- Rule Based Classification
- Classification by Back Propagation Lazzy Learners
- K-Nearest Neighbour
- Other Classification Methods

APRIORI ALGORITHM:

- Apriori employs an iterative approach known as a level-wise search, where k-itemsets are used to explore (k+1)-itemsets.
- First, the set of frequent 1-itemsetsisfound by scanning the database to accumulate the count for each item, and collecting those items that satisfy minimum support.
- The resulting set is denoted L1. Next, L1 is used to find L2, the set of frequent 2-itemsets, which is used to find L3, and so on, until no more frequent k-itemsets can be found.
- The finding of each Lk requires one full scan of the database.

Apriori property

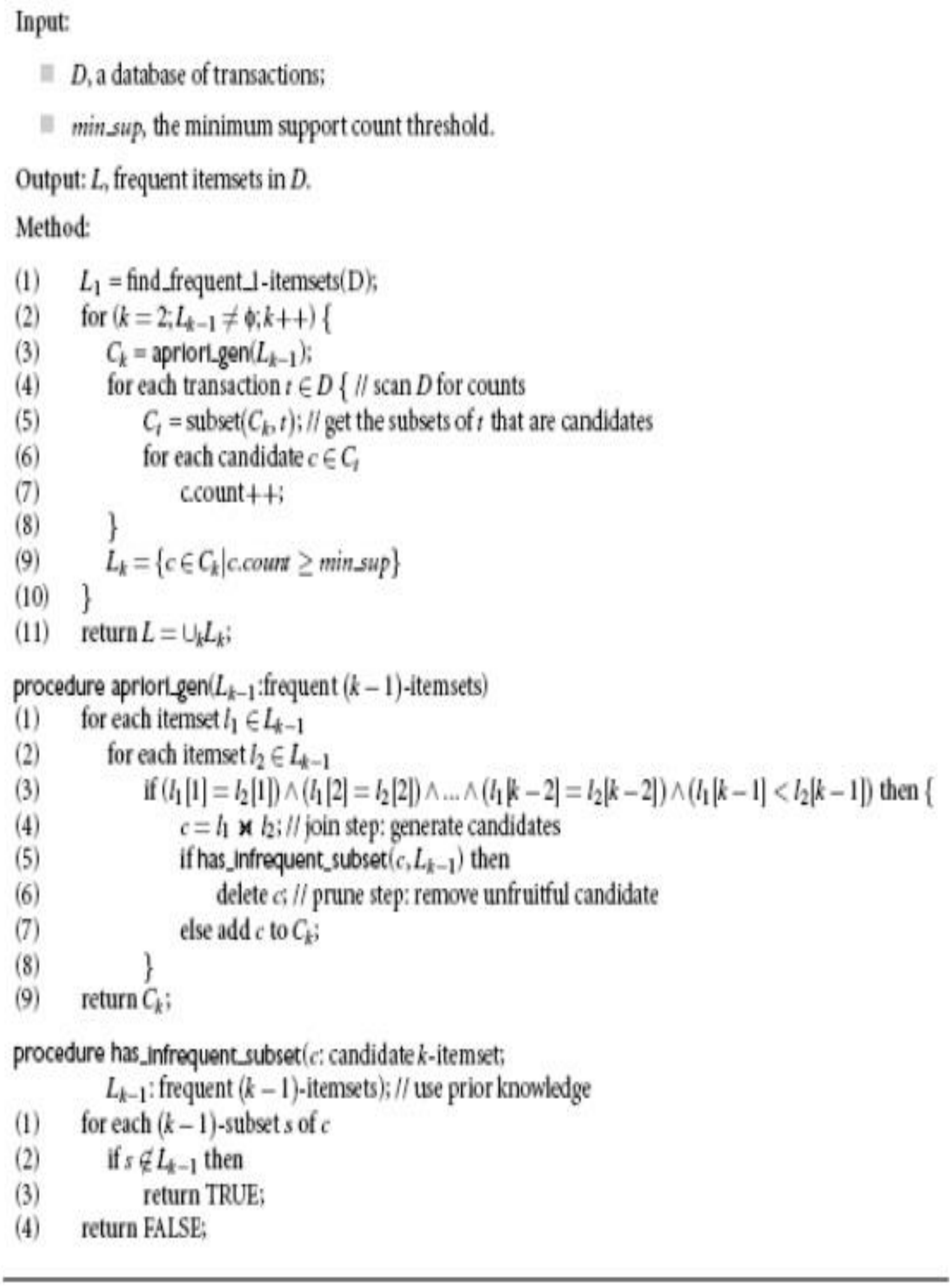
All nonempty subsets of a frequent item set most also be frequent.

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

THE APRIORI ALGORITHM:

Join Step: C_k is generated by joining $L_{k-1}$with itself

Prune Step: Any (k-1)-itemset that is not frequent cannot be a subset of a frequent k-itemset



Input:

- D, a database of transactions;
- *min_sup*, the minimum support count threshold.

Output: L, frequent itemsets in D.

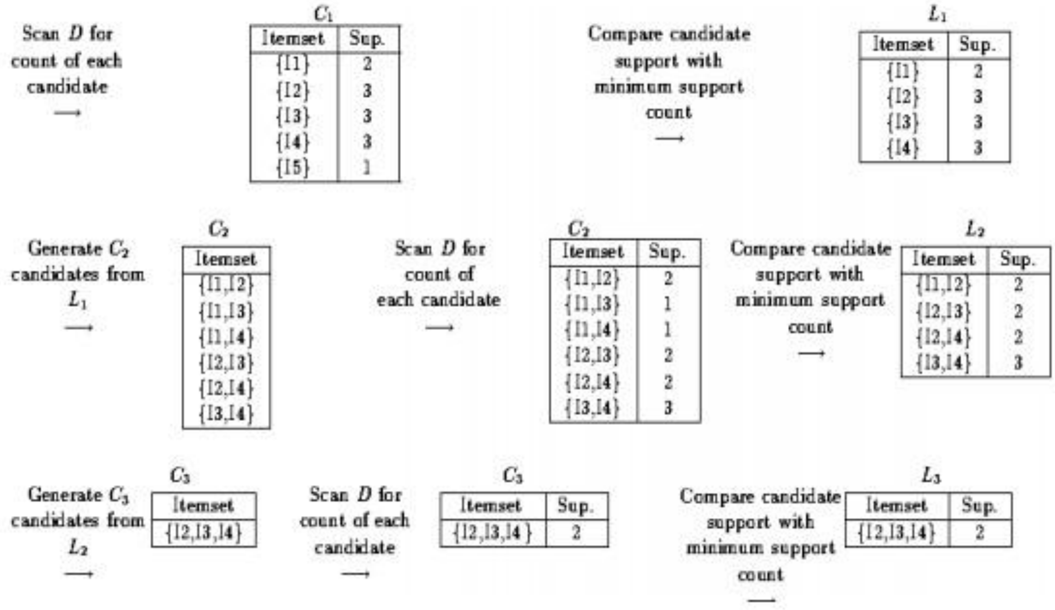
Method:

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

```
procedure apriori_gen($L_{k-1}$:frequent ($k-1$)-itemsets)
for each itemset $l_1 \in L_{k-1}$
    for each itemset $l_2 \in L_{k-1}$
        if $(l_1[1] = l_2[1]) \wedge (l_1[2] = l_2[2]) \wedge ... \wedge (l_1[k-2] = l_2[k-2]) \wedge (l_1[k-1] < l_2[k-1])$ then {
            $c = l_1 \bowtie l_2$; // join step: generate candidates
            if has_infrequent_subset($c, L_{k-1}$) then
                delete $c$; // prune step: remove unfruitful candidate
            else add $c$ to $C_k$;
        }
return $C_k$;
```

```
procedure has_infrequent_subset($c$: candidate $k$-itemset;
        $L_{k-1}$: frequent ($k-1$)-itemsets); // use prior knowledge
for each ($k-1$)-subset $s$ of $c$
    if $s \notin L_{k-1}$ then
        return TRUE;
return FALSE;
```

Example



Scan D for count of each candidate $\longrightarrow$

C_1

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

Compare candidate support with minimum support count $\longrightarrow$

L_1

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

Generate C_2 candidates from L_1 $\longrightarrow$

C_2

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

Scan D for count of each candidate $\longrightarrow$

C_2

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

Compare candidate support with minimum support count $\longrightarrow$

L_2

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

Generate C_3 candidates from L_2 $\longrightarrow$

C_3

Itemset
{I2,I3,I4}

Scan D for count of each candidate $\longrightarrow$

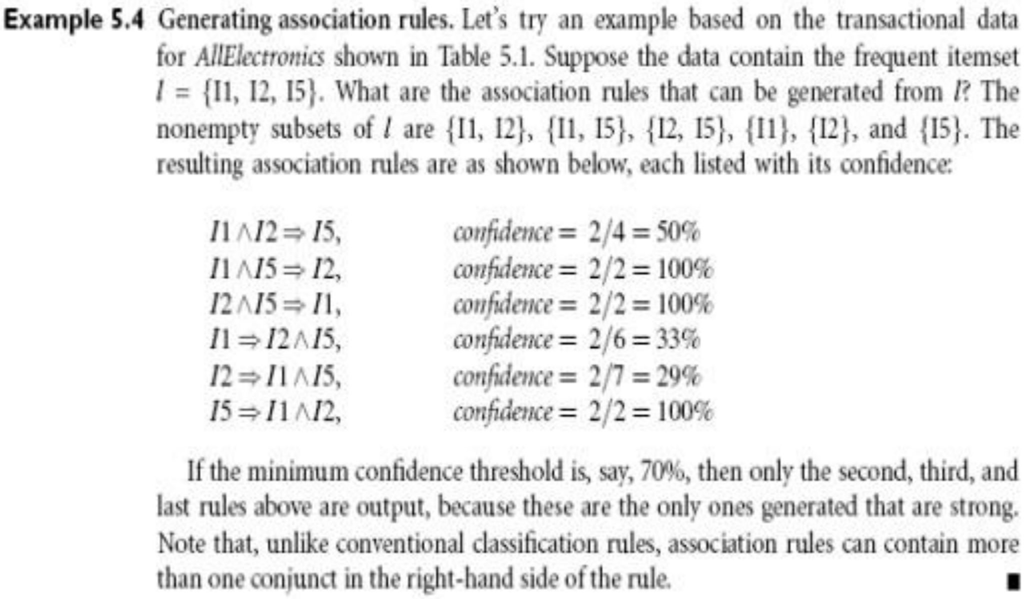
C_3

Itemset	Sup.
{I2,I3,I4}	2

Compare candidate support with minimum support count $\longrightarrow$

L_3

Itemset	Sup.
{I2,I3,I4}	2



Example 5.4 Generating association rules. Let's try an example based on the transactional data for *AllElectronics* shown in Table 5.1. Suppose the data contain the frequent itemset l = {I1, I2, I5}. What are the association rules that can be generated from l? The nonempty subsets of l are {I1, I2}, {I1, I5}, {I2, I5}, {I1}, {I2}, and {I5}. The resulting association rules are as shown below, each listed with its confidence:

$$
\begin{array}{ll}
I1 \wedge I2 \Rightarrow I5, & \textit{confidence} = 2/4 = 50\% \\
I1 \wedge I5 \Rightarrow I2, & \textit{confidence} = 2/2 = 100\% \\
I2 \wedge I5 \Rightarrow I1, & \textit{confidence} = 2/2 = 100\% \\
I1 \Rightarrow I2 \wedge I5, & \textit{confidence} = 2/6 = 33\% \\
I2 \Rightarrow I1 \wedge I5, & \textit{confidence} = 2/7 = 29\% \\
I5 \Rightarrow I1 \wedge I2, & \textit{confidence} = 2/2 = 100\%
\end{array}
$$

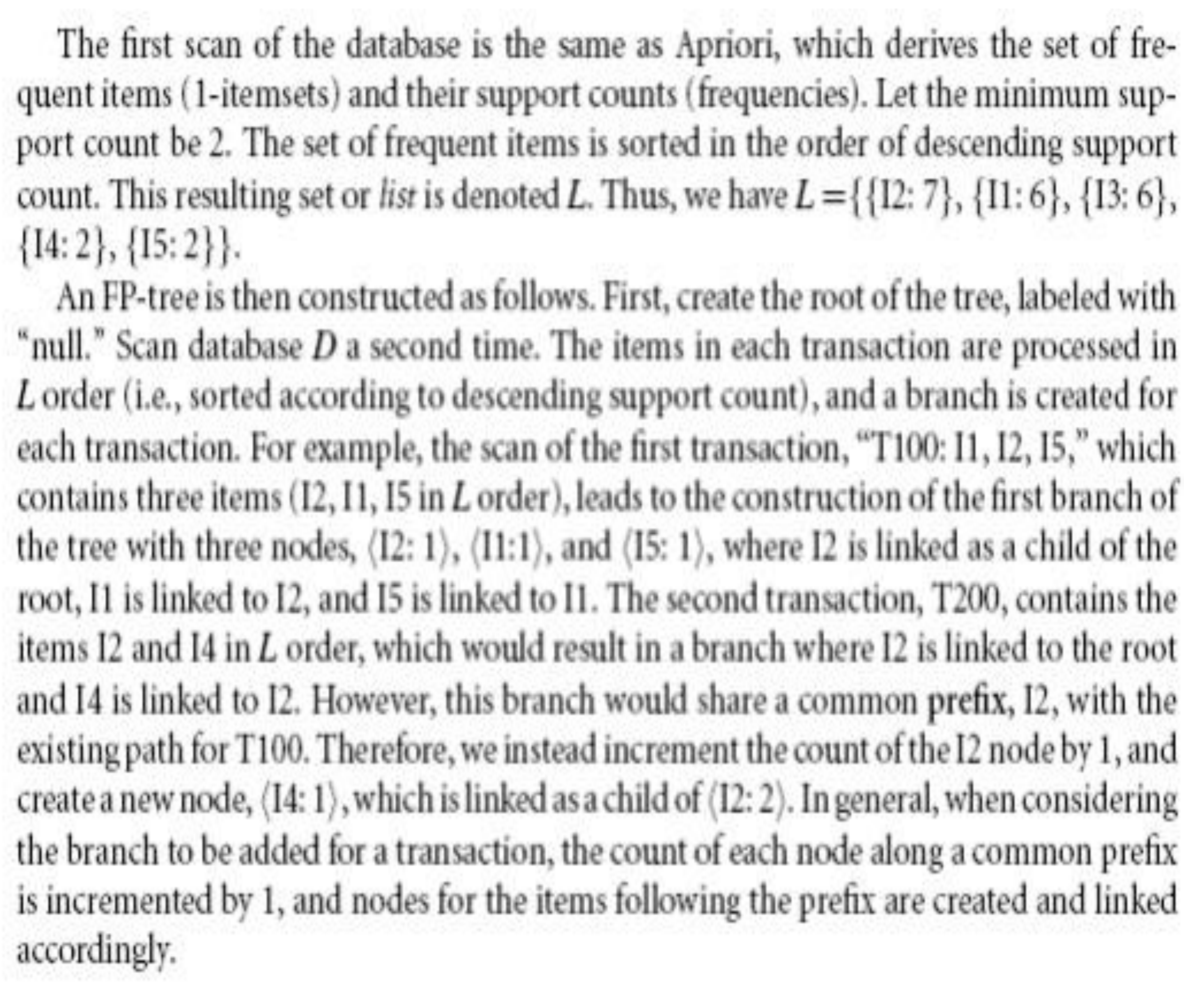
If the minimum confidence threshold is, say, 70%, then only the second, third, and last rules above are output, because these are the only ones generated that are strong. Note that, unlike conventional classification rules, association rules can contain more than one conjunct in the right-hand side of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

The method that mines the complete set of frequent itemsets without generation.

- Compress a large database into a compact, Frequent-Pattern tree (FP-tree) structure
- highly condensed, but complete for frequent pattern mining
- avoid costly database scans
- Develop an efficient, FP-tree-based frequent pattern mining method

- A divide-and-conquer methodology: decompose mining tasks into smaller ones
- Avoid candidate generation: sub-database test only

Example 5.5 FP-growth (finding frequent itemsets without candidate generation). We re-examine the mining of transaction database, *D*, of Table 5.1 in Example 5.3 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 *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, labeled with "null." Scan database *D* a second time. The items in each transaction are processed in *L* order (i.e., sorted according to descending support count), and a branch is created for each transaction. For example, the scan of the first transaction, "T100: I1, I2, I5," which contains three items (I2, I1, I5 in *L* order), leads to the construction of the first branch of the tree with three nodes, ⟨I2: 1⟩, ⟨I1:1⟩, and ⟨I5: 1⟩, where I2 is linked as a child of the root, I1 is linked to I2, and I5 is linked to I1. The second transaction, T200, contains the items I2 and I4 in *L* order, which would result in a branch where I2 is linked to the root and I4 is linked to I2. However, this branch would share a common prefix, I2, with the existing path for T100. Therefore, we instead increment the count of the I2 node by 1, and create a new node, ⟨I4: 1⟩, which is linked as a child of ⟨I2: 2⟩. In general, when considering the branch to be added 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.

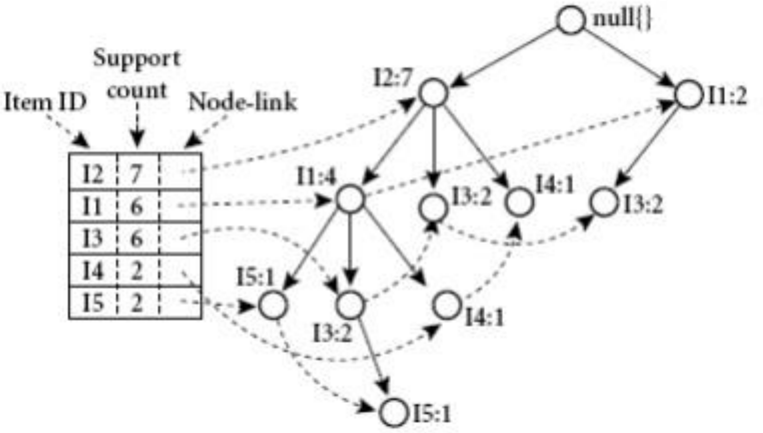
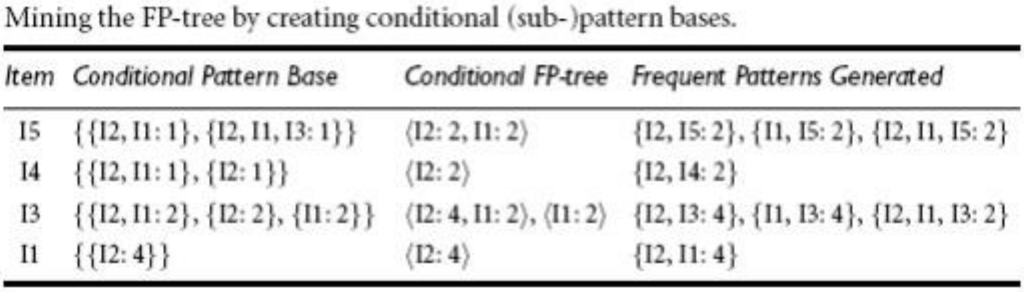


Figure An FP-tree registers compressed, frequent pattern information.



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

Item	*Conditional Pattern Base*	*Conditional FP-tree*	*Frequent Patterns Generated*
I5	{{I2, I1: 1}, {I2, I1, I3: 1}}	⟨I2: 2, I1: 2⟩	{I2, I5: 2}, {I1, I5: 2}, {I2, I1, I5: 2}
I4	{{I2, I1: 1}, {I2: 1}}	⟨I2: 2⟩	{I2, I4: 2}
I3	{{I2, I1: 2}, {I2: 2}, {I1: 2}}	⟨I2: 4, I1: 2⟩, ⟨I1: 2⟩	{I2, I3: 4}, {I1, I3: 4}, {I2, I1, I3: 2}
I1	{{I2: 4}}	⟨I2: 4⟩	{I2, I1: 4}

Benefits of the FP-tree Structure

Completeness:

- never breaks a long pattern of any transaction
- preserves complete information for frequent pattern mining

Compactness

- Reduce irrelevant information—infrequent items are gone
- Frequency descending ordering: more frequent items are more likely to be shared
- Never be larger than the original database (if not count node-links and counts)
- Example: For Connect-4 DB, compression ratio could be over 100

MINING FREQUENT PATTERNS USING FP-TREE

General idea (divide-and-conquer)

- o Recursively grow frequent pattern path using the FP-tree

Method

- o For each item, construct its conditional pattern-base, and then its conditional FP-tree
- o Repeat the process on each newly created conditional FP-tree
- o Until the resulting FP-tree is empty, or it contains only one path (single path will generate all the combinations of its sub-paths, each of which is a frequent pattern)

MAJOR STEPS TO MINE FP-TREE

1. Construct conditional pattern base for each node in the FP-tree
2. Construct conditional FP-tree from each conditional pattern-base
3. Recursively mine conditional FP-trees and grow frequent patterns obtained so far

- ◦ If the conditional FP-tree contains a single path, simply enumerate all the patterns

PRINCIPLES OF FREQUENT PATTERN GROWTH

- Pattern growth property
- Let α be a frequent itemset in DB, B be α's conditional pattern base, and β be an itemset in B. Then α ∪ β is a frequent itemset in DB iff β is frequent in B.
- *abcdef* ‖ is a frequent pattern, if and only if
- *abcde* ‖ is a frequent pattern, and
- *f* ‖ is frequent in the set of transactions containing —*abcde*

Why Is Frequent Pattern Growth Fast?:

Our performance study shows

- FP-growth is an order of magnitude faster than Apriori, and is also faster than tree-projection

- No candidate generation, no candidate test
- Use compact data structure
- Eliminate repeated database scan
- Basic operation is counting and FP-tree building

DEFINITION OF CLASSIFICATION AND PREDICTION:

There are two forms of data analysis that can be used for extracting models describing important classes or to predict future data trends. These two forms are as follows −

- Classification
- Prediction

Classification models predict categorical class labels; and prediction models predict continuous valued functions. For example, we can build a classification model to categorize bank loan applications as either safe or risky, or a prediction model to predict the expenditures in dollars of potential customers on computer equipment given their income and occupation.

What is classification?

Following are the examples of cases where the data analysis task is Classification

- A bank loan officer wants to analyze the data in order to know which customer (loan applicant) are risky or which are safe.
- A marketing manager at a company needs to analyze a customer with a given profile, who will buy a new computer.

In both of the above examples, a model or classifier is constructed to predict the categorical labels. These labels are risky or safe for loan application data and yes or no for marketing data.

What is prediction?

Following are the examples of cases where the data analysis task is Prediction −

Suppose the marketing manager needs to predict how much a given customer will spend during a sale at his company. In this example we are bothered to predict a numeric value. Therefore the data analysis task is an example of numeric prediction. In this case, a model or a predictor will be constructed that predicts a continuous-valued-function or ordered value.

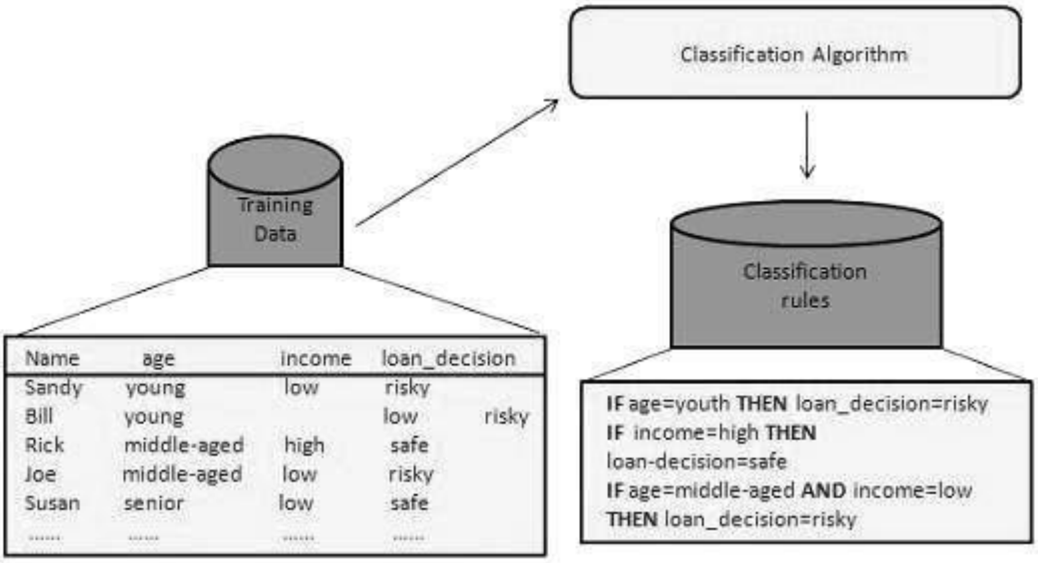
How Does Classification Works?

With the help of the bank loan application that we have discussed above, let us understand the working of classification. The Data Classification process includes two steps −

- Building the Classifier or Model
- Using Classifier for Classification

Building the Classifier or Model:

- This step is the learning step or the learning phase.
- In this step the classification algorithms build the classifier.
- The classifier is built from the training set made up of database tuples and their associated class labels.
- Each tuple that constitutes the training set is referred to as a category or class. These tuples can also be referred to as sample, object or data points.

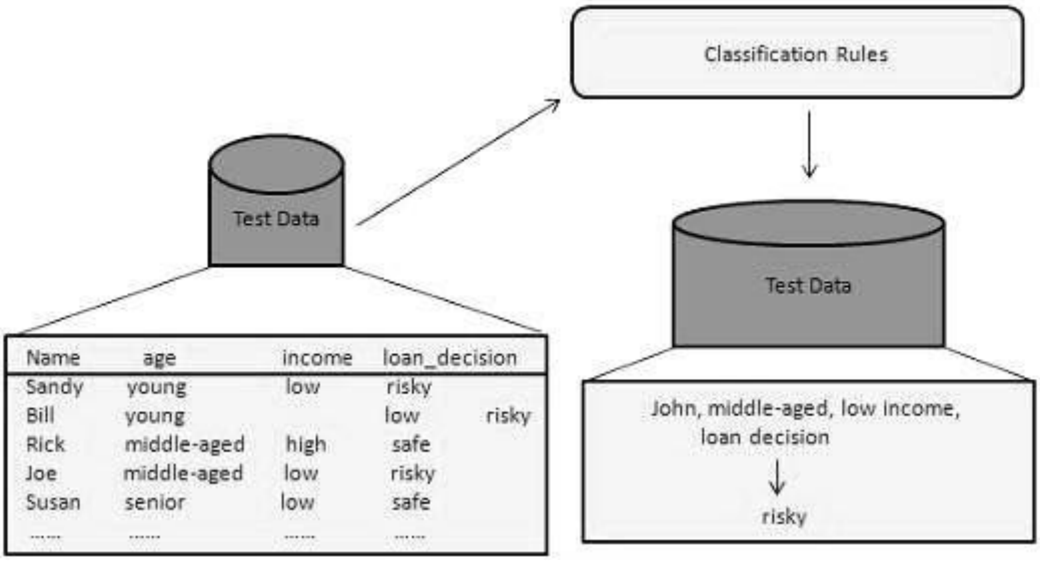


Name	age	income	loan_decision	
Sandy	young	low	risky	
Bill	young		low	risky
Rick	middle-aged	high	safe	
Joe	middle-aged	low	risky	
Susan	senior	low	safe	
......	......	......	......	

IF age=youth **THEN** loan_decision=risky
IF income=high **THEN**
loan-decision=safe
IF age=middle-aged **AND** income=low
THEN loan_decision=risky

Using Classifier for Classification:

In this step, the classifier is used for classification. Here the test data is used to estimate the accuracy of classification rules. The classification rules can be applied to the new data tuples if the accuracy is considered acceptable.



Name	age	income	loan_decision	
Sandy	young	low	risky	
Bill	young		low	risky
Rick	middle-aged	high	safe	
Joe	middle-aged	low	risky	
Susan	senior	low	safe	
......	......	......	......	

Classification and Prediction Issues

The major issue is preparing the data for Classification and Prediction. Preparing the data involves the following activities −

- **Data Cleaning** − Data cleaning involves removing the noise and treatment of missing values. The noise is removed by applying smoothing techniques and the problem of missing values is solved by replacing a missing value with most commonly occurring value for that attribute.
- **Relevance Analysis** − Database may also have the irrelevant attributes. Correlation analysis is used to know whether any two given attributes are related.
- **Data Transformation and reduction** − The data can be transformed by any of the following methods.
  - **Normalization** − The data is transformed using normalization. Normalization involves scaling all values for given attribute in order to make them fall within a small specified range. Normalization is used when in the learning step, the neural networks or the methods involving measurements are used.
  - **Generalization** − The data can also be transformed by generalizing it to the higher concept. For this purpose we can use the concept hierarchies.

Comparison of Classification and Prediction Methods:

Here is the criteria for comparing the methods of Classification and Prediction −

- **Accuracy** − Accuracy of classifier refers to the ability of classifier. It predict the class label correctly and the accuracy of the predictor refers to how well a given predictor can guess the value of predicted attribute for a new data.

- **Speed** − This refers to the computational cost in generating and using the classifier or predictor.
- **Robustness** − It refers to the ability of classifier or predictor to make correct predictions from given noisy data.
- **Scalability** − Scalability refers to the ability to construct the classifier or predictor efficiently; given large amount of data.
- **Interpretability** − It refers to what extent the classifier or predictor understands.

CLASSIFICATION BY DECISION TREE INDUCTION:

Decision tree

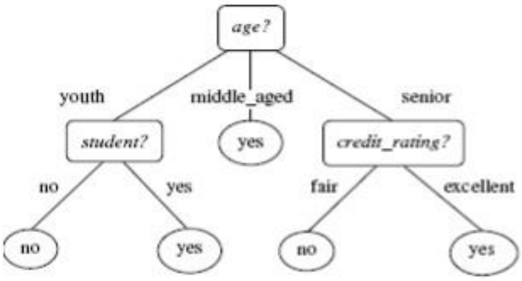
- A flow-chart-like tree structure.
- Internal node denotes a test on an attribute node (nonleaf node) denotes a test on an attribute.
- Branch represents an outcome of the test.
- Leaf nodes represent class labels or class distribution(Terminal node).
- The topmost node in a tree is the root node.

Decision tree generation consists of two phases

- Tree construction
- At start, all the training examples are at the root
- Partition examples recursively based on selected attributes

Tree pruning:

- Identify and remove branches that reflect noise or outliers



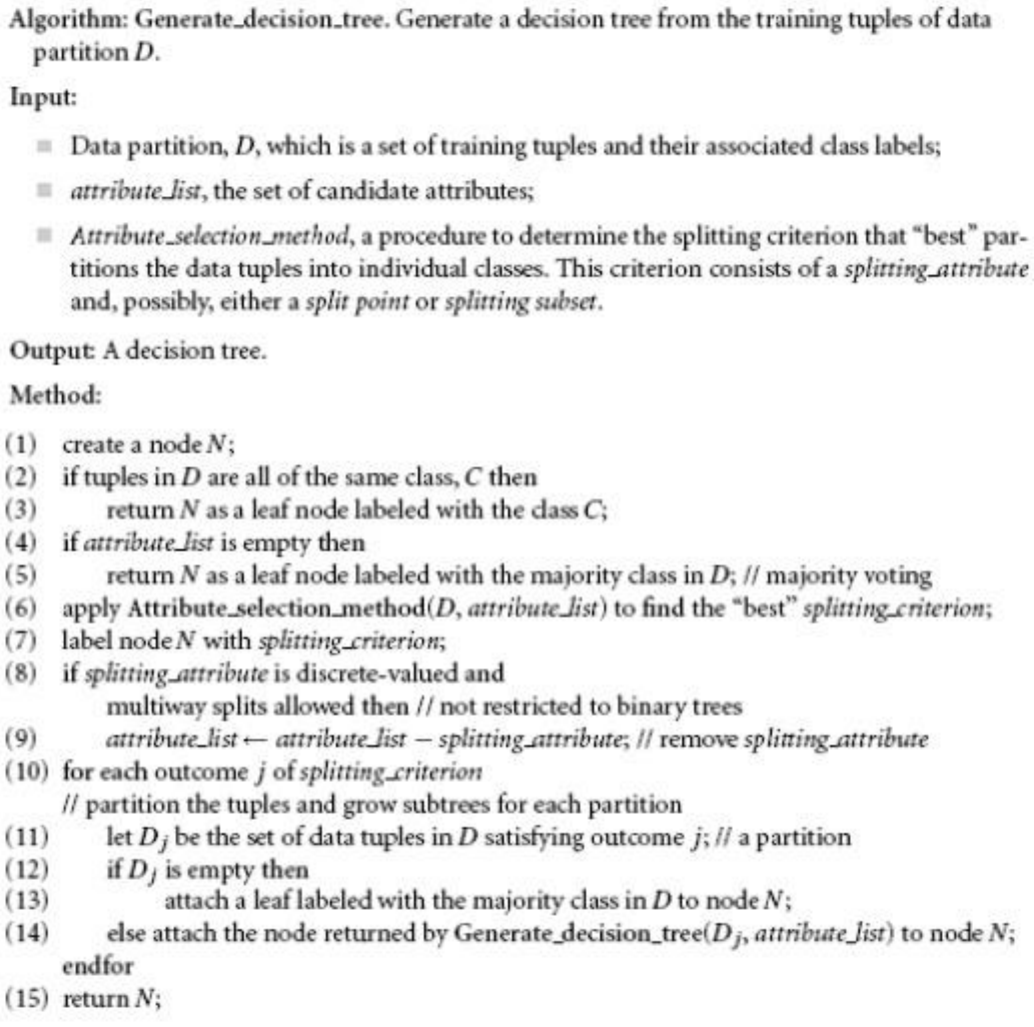
A decision tree for the concept *buys_computer*, indicating whether a customer at *AllElectronics* is likely to purchase a computer. Each internal (nonleaf) node represents a test on an attribute. Each leaf node represents a class (either *buys_computer* = *yes* or *buys_computer* = *no*).

A typical decision tree is shown in Figure. It represents the concept *buys computer*, that is, it predicts whether a customer at *AllElectronics* is likely to purchase a computer.

Internal nodes are denoted by rectangles, and leaf nodes are denoted by ovals. Some decision tree algorithms produce only *binary* trees (where each internal node branches to exactly two other nodes), whereas others can produce non binary trees.

"How are decision trees used for classification?": Given a tuple, ***X***, for which the associated class label is unknown, the attribute values of the tuple are tested against the decision tree. A path is traced from the root to a leaf node, which holds the class prediction for that tuple. Decision trees can easily be converted to classification rules.

Decision Tree Induction



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.

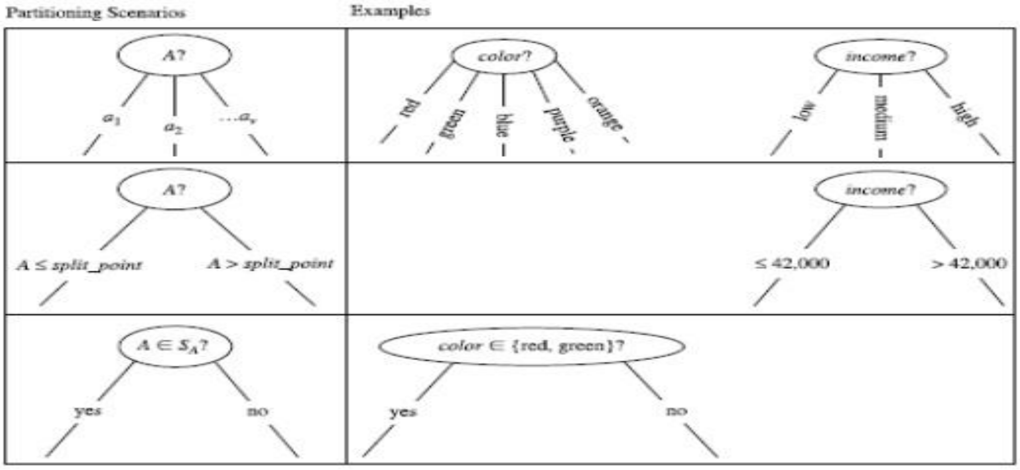
Method:

```
(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 $D_j$ be the set of data tuples in $D$ satisfying outcome $j$; // a partition
(12)     if $D_j$ is empty then
(13)         attach a leaf labeled with the majority class in $D$ to node $N$;
(14)     else attach the node returned by Generate_decision_tree($D_j$, attribute_list) to node $N$;
     endfor
(15) return $N$;
```

Basic algorithm for inducing a decision tree from training tuples.

The tree starts as a single node, *N*, representing the training tuples in *D* (step 1)

- If the tuples in *D* are all of the same class, then node *N* becomes a leaf and is labeled with that class (steps 2 and 3). Note that steps 4 and 5 are terminating conditions. All of the terminating conditions are explained at the end of the algorithm.
- Otherwise, the algorithm calls *Attribute selection method* to determine the splitting criterion. The splitting criterion tells us which attribute to test at node *N* by determining the —best‖ way to separate or partition the tuples in *D* into individual classes(step 6). The splitting criterion also tells us which branches to grow from node *N* with respect to the outcomes of the chosen test. More specifically, the splitting criterion indicates the splitting attribute and may also indicate either a split-point or a splitting subset. The splitting criterion is determined so that, ideally, the resulting partitions at each branch are as —pure‖ as possible.
- A partition is pure if all of the tuples in it belong to the same class. In other words, if we were to split up the tuples in *D* according to the mutually exclusive outcomes of the splitting criterion, we hope for the resulting partitions to be as pure as possible.
- The node *N* is labeled with the splitting criterion, which serves as a test at the node (step 7). A branch is grown from node *N* for each of the outcomes of the splitting criterion. The tuples in *D* are partitioned accordingly (steps 10 to 11). There are three possible scenarios, as illustrated in Figure. Let *A* be the splitting attribute. *A* has *v* distinct values, {*a*1, *a*2, : : : , *av*}, based on the training data.



Three possibilities for partitioning tuples based on the splitting criterion, shown with examples. Let A be the splitting attribute. (a) If A is discrete-valued, then one branch is grown for each known value of A. (b) If A is continuous-valued, then two branches are grown, corresponding to $A \le split\_point$ and $A > split\_point$. (c) If A is discrete-valued and a binary tree must be produced, then the test is of the form $A \in S_A$, where S_A is the splitting subset for A.

Attribute Selection Measures:

- An attribute selection measure is a heuristic for selecting the splitting criterion that best separates a given data partition, *D*, of class-labeled training tuples into individual classes.
- If we were to split *D* into smaller partitions according to the outcomes of the splitting criterion,
- If the splitting attribute is continuous-valued or if we are restricted to binary trees then, respectively, either a *split point* or a *splitting subset* must also be determined as part of the splitting criterion .
- This section describes three popular attribute selection measures—*information gain, gain ratio,* and *gini index.*

INFORMATION GAIN:

ID3 uses information gain as its attribute selection measure.

$$Info(D) = -\sum_{i=1}^{m} p_i \log_2(p_i),$$

$$Info_A(D) = \sum_{j=1}^{v} \frac{|D_j|}{|D|} \times Info(D_j).$$

Information gain is defined as the difference between the original information requirement (i.e., based on just the proportion of classes) and the new requirement (i.e., obtained after partitioning on *A*). That is,

$$Gain(A) = Info(D) - Info_A(D).$$

In other words, *Gain(A)* tells us how much would be gained by branching on *A*. It is the expected reduction in the information requirement caused by knowing the value of *A*. The attribute *A* with the highest information gain, (*Gain*(*A*)), is chosen as the splitting attribute at node *N*.

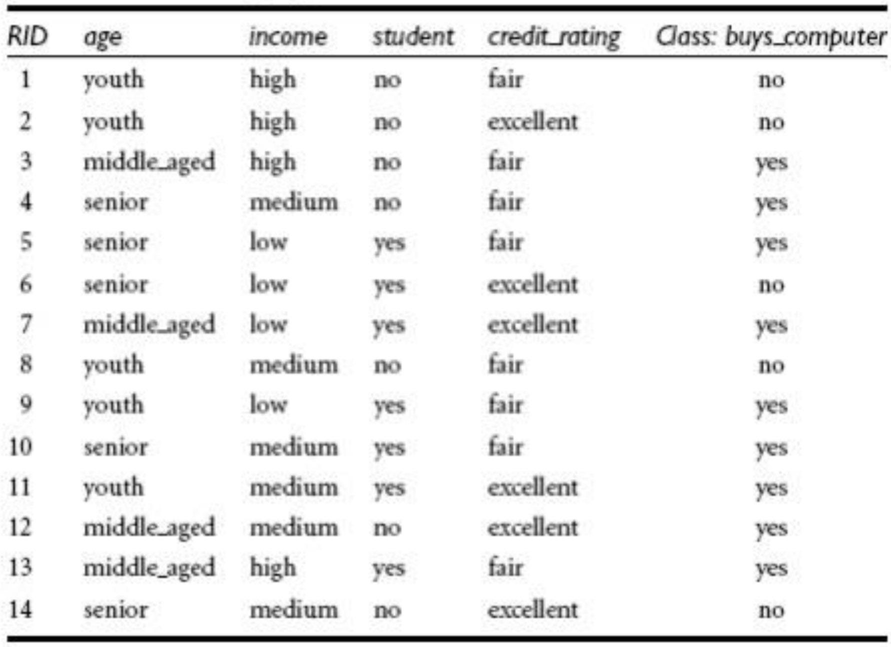
Example Induction of a decision tree using information gain.

Table 6.1 presents a training set, *D*, of class-labeled tuples randomly selected from the *AllElectronics* customer database. (The data are adapted from [Qui86]. In this example, each attribute is discrete-valued. Continuous-valued attributes have been generalized.) The class label attribute, *buys computer*, has two distinct values (namely, {*yes, no}*); therefore, there are two distinct classes (that is, *m* =2).

Let class *C*1 correspond to *yes* and class *C*2 correspond to *no*. There are nine tuples of class *yes* and five tuples of class *no*. A (root) node *N* is created for the tuples in *D*. To find the splitting criterion for these tuples, we must compute the information gain of each attribute. We first use Equation (6.1) to compute the expected information needed to classify a tuple in *D*:

$$Info(D) = -\frac{9}{14}\log_2\left(\frac{9}{14}\right) - \frac{5}{14}\log_2\left(\frac{5}{14}\right) = 0.940 \text{ bits.}$$

Table 6.1 Class-labeled training tuples from the *AllElectronics* customer database.



RID	*age*	*income*	*student*	*credit_rating*	*Class: buys_computer*
1	youth	high	no	fair	no
2	youth	high	no	excellent	no
3	middle_aged	high	no	fair	yes
4	senior	medium	no	fair	yes
5	senior	low	yes	fair	yes
6	senior	low	yes	excellent	no
7	middle_aged	low	yes	excellent	yes
8	youth	medium	no	fair	no
9	youth	low	yes	fair	yes
10	senior	medium	yes	fair	yes
11	youth	medium	yes	excellent	yes
12	middle_aged	medium	no	excellent	yes
13	middle_aged	high	yes	fair	yes
14	senior	medium	no	excellent	no

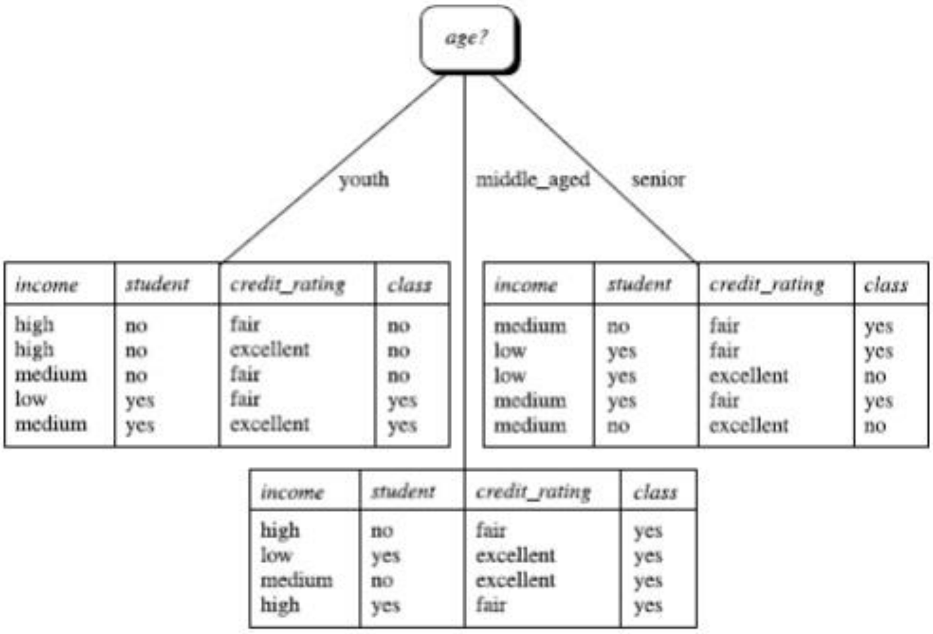
The expected information needed to classify a tuple in *D* if the tuples are partitioned according to *age* is

$$\begin{aligned} Info_{age}(D) &= \frac{5}{14} \times \left(-\frac{2}{5}\log_2\frac{2}{5} - \frac{3}{5}\log_2\frac{3}{5}\right) \\ &+ \frac{4}{14} \times \left(-\frac{4}{4}\log_2\frac{4}{4} - \frac{0}{4}\log_2\frac{0}{4}\right) \\ &+ \frac{5}{14} \times \left(-\frac{3}{5}\log_2\frac{3}{5} - \frac{2}{5}\log_2\frac{2}{5}\right) \\ &= 0.694 \text{ bits.} \end{aligned}$$

Hence, the gain in information from such a partitioning would be

$$Gain(age) = Info(D) - Info_{age}(D) = 0.940 - 0.694 = 0.246 \text{ bits.}$$

Similarly, we can compute *Gain*(*income*) = 0.029 bits, *Gain*(*student*) = 0.151 bits, and *Gain*(*credit rating*) = 0.048 bits. Because *age* has the highest information gain among the attributes, it is selected as the splitting attribute. Node *N* is labeled with *age*, and branches are grown for each of the attribute's values. The tuples are then partitioned accordingly, as shown in Figure 6.5. Notice that the tuples falling into the partition for *age* = *middle aged* all belong to the same class. Because they all belong to class *"yes,"* a leaf should therefore be created at the end of this branch and labeled with *"yes."* The final decision tree returned by the algorithm is shown in Figure 6.5.



income	*student*	*credit_rating*	*class*
high	no	fair	no
high	no	excellent	no
medium	no	fair	no
low	yes	fair	yes
medium	yes	excellent	yes

income	*student*	*credit_rating*	*class*
medium	no	fair	yes
low	yes	fair	yes
low	yes	excellent	no
medium	yes	fair	yes
medium	no	excellent	no

income	*student*	*credit_rating*	*class*
high	no	fair	yes
low	yes	excellent	yes
medium	no	excellent	yes
high	yes	fair	yes

Figure 6.5 The attribute *age* has the highest information gain and therefore becomes the splitting attribute at the root node of the decision tree. Branches are grown for each outcome of *age*. The tuples are shown partitioned accordingly.

BAYESIAN CLASSIFICATION:

- Bayesian classifiers are statistical classifiers. They can predict class membership probabilities, such as the probability that a given tuple belongs to a particular class.
- Bayesian classification is based on Bayes' theorem, a simple Bayesian classifier known as the *naïve Bayesian classifier*.
- Bayesian classifiers have also exhibited high accuracy and speed when applied to large databases.

1. Bayes' Theorem

Let ***X*** be a data tuple. In Bayesian terms, ***X*** is considered —evidence on a set of *n* attributes. Let *H* be some hypothesis, such as that the data tuple ***X*** belongs to a specified class *C*.

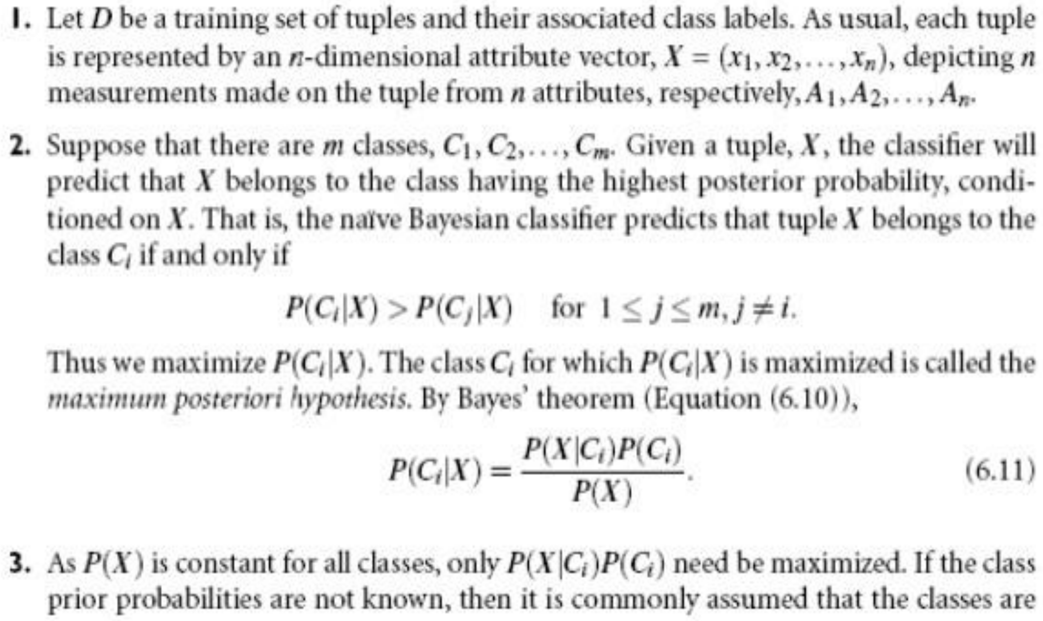
For classification problems, we want to determine *P*(*H*j***X***), the probability that the hypothesis *H* holds given the ―evidence‖ or observed data tuple ***X***.

In other words, we are looking for the probability that tuple ***X*** belongs to class *C*, given that we know the attribute description of ***X***.

*"**<u>How are these probabilities estimated?</u>**"* *P*(*H*), *P*(***X***j*H*), and *P*(***X***) may be estimated from the given data. it provides a way of calculating the posterior probability, *P*(*H*j***X***), from *P*(*H*), *P*(***X***j*H*), and *P*(***X***). Bayes' theorem is

$$P(H|X) = \frac{P(X|H)P(H)}{P(X)}.$$

2. Naïve Bayesian Classification:



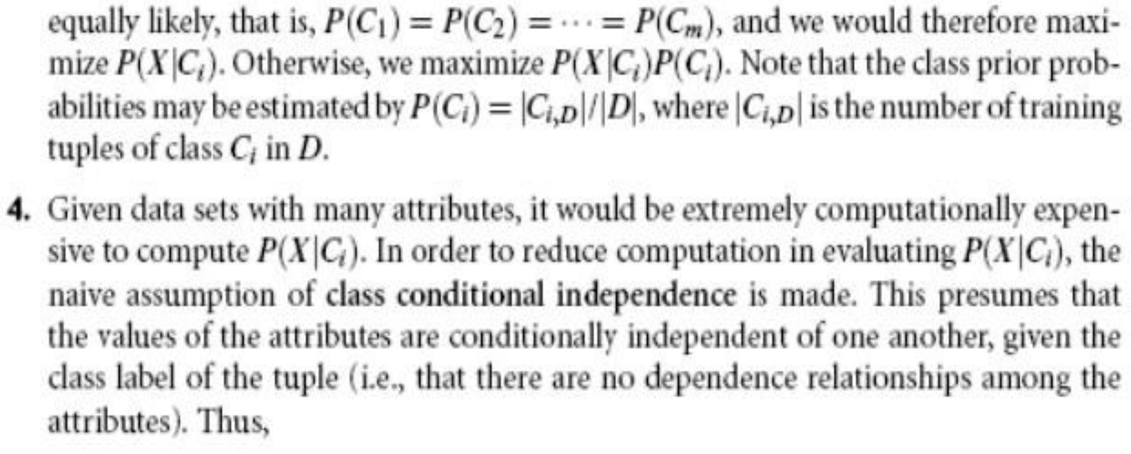
1. Let D be a training set of tuples and their associated class labels. As usual, each tuple is represented by an n-dimensional attribute vector, $X = (x_1, x_2, \ldots, x_n)$, depicting n measurements made on the tuple from n attributes, respectively, $A_1, A_2, \ldots, A_n$.
2. Suppose that there are m classes, $C_1, C_2, \ldots, C_m$. Given a tuple, X, the classifier will predict that X belongs to the class having the highest posterior probability, conditioned on X. That is, the naïve Bayesian classifier predicts that tuple X belongs to the class C_i if and only if

$$P(C_i|X) > P(C_j|X) \quad \text{for } 1 \le j \le m, j \ne i.$$

Thus we maximize $P(C_i|X)$. The class C_i for which $P(C_i|X)$ is maximized is called the *maximum posteriori hypothesis*. By Bayes' theorem (Equation (6.10)),

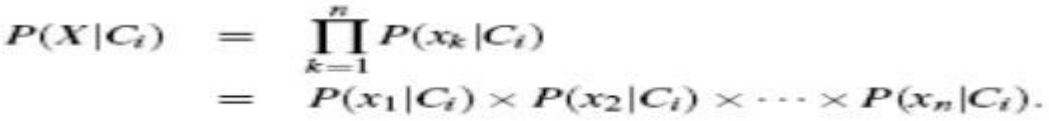
$$P(C_i|X) = \frac{P(X|C_i)P(C_i)}{P(X)}. \qquad (6.11)$$

3. As $P(X)$ is constant for all classes, only $P(X|C_i)P(C_i)$ need be maximized. If the class prior probabilities are not known, then it is commonly assumed that the classes are

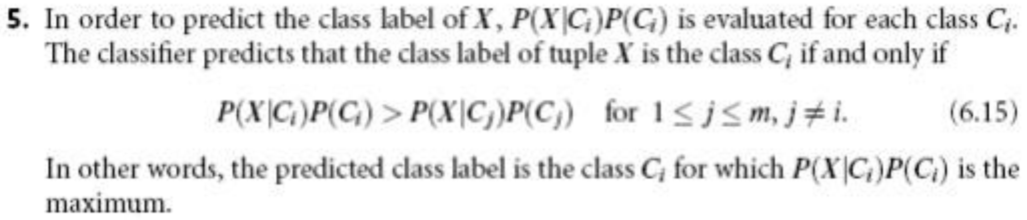


equally likely, that is, $P(C_1) = P(C_2) = \cdots = P(C_m)$, and we would therefore maximize $P(X|C_i)$. Otherwise, we maximize $P(X|C_i)P(C_i)$. Note that the class prior probabilities may be estimated by $P(C_i) = |C_{i,D}|/|D|$, where $|C_{i,D}|$ is the number of training tuples of class C_i in D.

4. Given data sets with many attributes, it would be extremely computationally expensive to compute $P(X|C_i)$. In order to reduce computation in evaluating $P(X|C_i)$, the naive assumption of **class conditional independence** is made. This presumes that the values of the attributes are conditionally independent of one another, given the class label of the tuple (i.e., that there are no dependence relationships among the attributes). Thus,



$$\begin{aligned} P(X|C_i) &= \prod_{k=1}^{n} P(x_k|C_i) \\ &= P(x_1|C_i) \times P(x_2|C_i) \times \cdots \times P(x_n|C_i). \end{aligned}$$



5. In order to predict the class label of X, $P(X|C_i)P(C_i)$ is evaluated for each class C_i. The classifier predicts that the class label of tuple X is the class C_i if and only if

$$P(X|C_i)P(C_i) > P(X|C_j)P(C_j) \quad \text{for } 1 \le j \le m, j \ne i. \qquad (6.15)$$

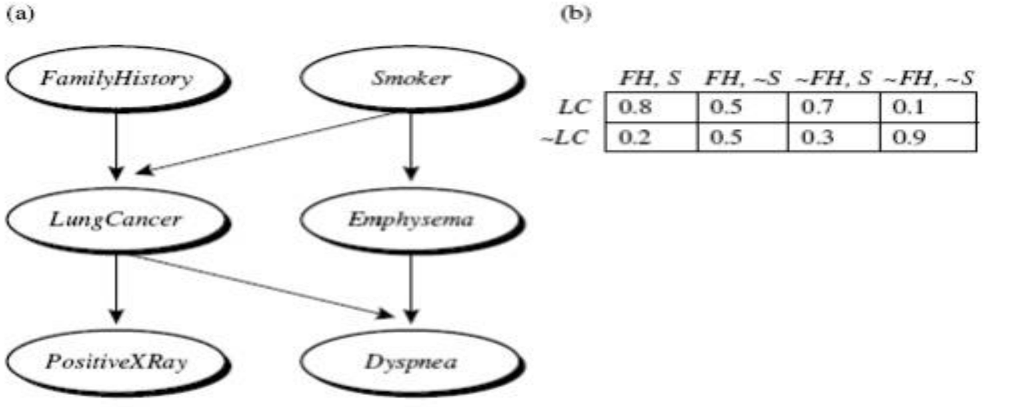
In other words, the predicted class label is the class C_i for which $P(X|C_i)P(C_i)$ is the maximum.

Bayesian Belief Networks

A belief network is defined by two components

- a *directed acyclic graph* and
- a set of *conditional probability tables.*

- The variables may be discrete or continuous-valued.
- They may correspond to actual attributes given in the data or to —hidden variables‖ believed to form a relationship (e.g., in the case of medical data).
- Each arc represents a probabilistic dependence.
- If an arc is drawn from a node *Y* to a node *Z,* then *Y* is a parent or immediate predecessor of *Z,* and *Z* is a descendant of *Y.*
- *Each variable is conditionally independent of its non descendants in the graph, given its parents.*



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

A simple Bayesian belief network: (a) A proposed causal model, represented by a directed acyclic graph. (b) The conditional probability table for the values of the variable *LungCancer (LC)* showing each possible combination of the values of its parent nodes, *FamilyHistory (FH)* and *Smoker (S)*. Figure is adapted from [RBKK95].

- A belief network has one conditional probability table (CPT) for each variable. The CPT for a variable *Y* specifies the conditional distribution *P*(*Y*j*Parents*(*Y*)), where *Parents*(*Y*) are the parents of *Y*.
- Figure(b) shows a CPT for the variable *LungCancer*.
- The conditional probability for each known value of *LungCancer* is given for each possible combination of values of its parents. F
- or instance, from the upper leftmost and bottom rightmost entries, respectively, we see that

$$P(LungCancer = yes \mid FamilyHistory = yes, Smoker = yes) = 0.8$$
$$P(LungCancer = no \mid FamilyHistory = no, Smoker = no) = 0.9$$

Let ***X*** = (*x*1, : : : , *xn*) be a data tuple described by the variables or attributes *Y*1, : : : , *Yn*, respectively.

following equation:

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

RULE BASED CLASSIFICATION

Using IF-THEN Rules for Classification :

Represent the knowledge in the form of IF-THEN rules

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

Rule antecedent/precondition vs. rule consequent

Assessment of a rule: *coverage* and *accuracy*

- n_{covers} = # of tuples covered by R
- $n_{correct}$ = # of tuples correctly classified by R
- o coverage(R) = n_{covers} /|D| /* D: training data set */
- accuracy(R) = $n_{correct}$ / n_{covers}

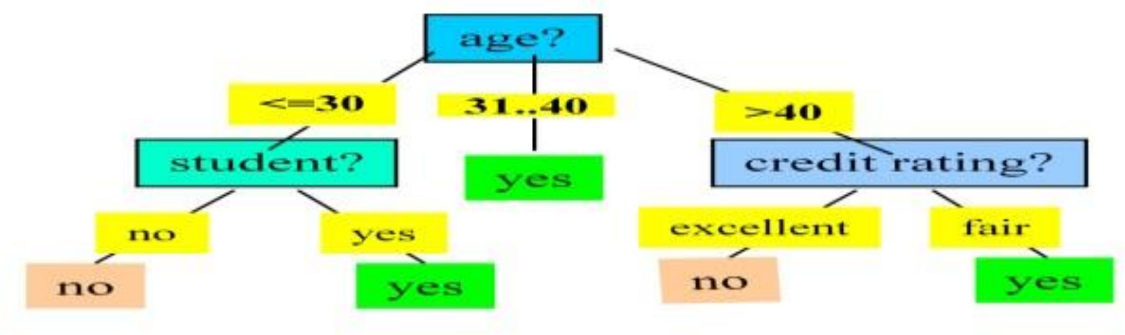
If more than one rule is triggered, need conflict resolution

- **Size ordering:** assign the highest priority to the triggering rules th at has the toughest‖ requirement (i.e., with the *most attribute test*)
- **Class-based ordering:** decreasing order of *prevalence or misclassification cost per class*

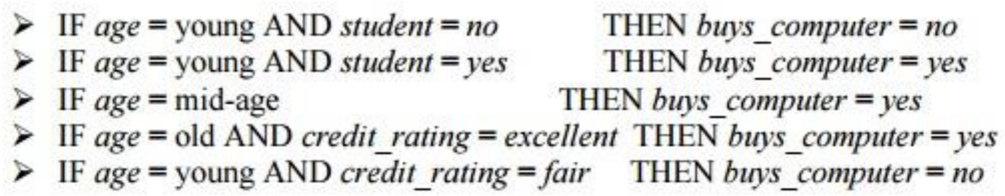
- **Rule-based ordering (decision list):** rules are organized into one long priority list, according to some measure of rule quality or by experts

Rule Extraction from a Decision Tree:

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



Example: Rule extraction from our *buys_computer* decision-tree



- IF *age* = young AND *student* = *no* THEN *buys_computer* = *no*
- IF *age* = young AND *student* = *yes* THEN *buys_computer* = *yes*
- IF *age* = mid-age THEN *buys_computer* = *yes*
- IF *age* = old AND *credit_rating* = *excellent* THEN *buys_computer* = *yes*
- IF *age* = young AND *credit_rating* = *fair* THEN *buys_computer* = *no*

Rule Induction Using a Sequential Covering Algorithm:

- Sequential covering algorithm: Extracts rules directly from training data
- Typical sequential covering algorithms: FOIL, AQ, CN2, RIPPER.
- Rules are learned *sequentially*, each for a given class C_i will cover many tuples of C_i but none (or few) of the tuples of other classes.

Steps:

- Rules are learned one at a time
- Each time a rule is learned, the tuples covered by the rules are removed
- The process repeats on the remaining tuples unless *termination condition*, e.g., when no more training examples or when the quality of a rule returned is below a user-specified threshold

- Comp. w. decision-tree induction: learning a set of rules *simultaneously*

```
Algorithm: Sequential covering. Learn a set of IF-THEN rules for classification.
Input: D, a data set class-labeled tuples;
Att vals, the set of all attributes and their possible values.
Output: A set of IF-THEN rules.
Method: (1) Rule set = {};
 // initial set of rules learned is empty
(2) for each class c do
 (3) repeat
(4) Rule = Learn One Rule(D, Att vals, c);
(5) remove tuples covered by Rule from D;
 (6) until terminating condition;
 (7) Rule set = Rule set +Rule; // add new rule to rule set
 (8) endfor
 (9) return Rule Set;
```

Rule Quality Measures :

- Learn One Rule needs a measure of rule quality.
- Every time it considers an attribute test, it must check to see if appending such a test to the current rule's condition will result in an improved rule.

Example :

- Choosing between two rules based on accuracy.
- Consider the two rules as illustrated in Figure 6.14. Both are for the class loan decision = accept.
- We use "a" to represent the tuples of class "accept" and "r" for the tuples of class "reject."
- Rule R1 correctly classifies 38 of the 40 tuples it covers.
- Rule R2 covers only two tuples, which it correctly classifies.
- Their respective accuracies are 95% and 100%. Thus, R2 has greater accuracy than R1, but it is not the better rule because of its small coverage.
- we are learning rules for the class c.
- Our current rule is R: IF condition THEN class = c.

- We want to see if logically ANDing a given attribute test to condition would result in a better rule.
- We call the new condition, condition0 , where R 0 : IF condition0 THEN class = c is our potential new rule. In other words, we want to see if R 0 is any better than R.
- Let pos (neg) be the number of positive (negative) tuples covered by R.
- Let pos0 (neg0) be the number of positive (negative) tuples covered by R 0 .
- FOIL assesses the information gained by extending condition as

FOIL Gain = pos0 × ³ log2 pos0 pos0 +neg0 −log2 pos pos+neg

- It favors rules that have high accuracy and cover many positive tuples.
- We can use the likelihood ratio statistic,

Likelihood Ratio = 2 m ∑ i=1 fi log³ fi ei ′

- where m is the number of classes.
- For tuples satisfying the rule, fi is the observed frequency of each class i among the tuples. ei is what we would expect the frequency of each class i to be if the rule made random predictions.

Rule Pruning:

The rule is pruned is due to the following reason .

The Assessment of quality is made on the original set of training data. The rule may perform well on training data but less well on subsequent data. That's why the rule pruning is required.

- The rule is pruned by removing conjunct. The rule R is pruned, if pruned version of R has greater quality than what was assessed on an independent set of tuples.
- FOIL is one of the simple and effective method for rule pruning. For a given rule R,

FOIL_Prune = pos - neg / pos + neg

where pos and neg is the number of positive tuples covered by R, respectively.

CLASSIFICATION BY BACKPROPAGATION:

- Backpropagation: A **neural network** learning algorithm.
- Started by psychologists and neurobiologists to develop and test computational analogues of neurons.

- A neural network: A set of connected input/output units where each connection has a **weight** associated with it.
- During the learning phase, the **network learns by adjusting the weights** so as to be able to predict the correct class label of the input tuples
- Also referred to as **connectionist learning** due to the connections between units.

Neural Network as a Classifier:

Weakness

o Long training time

o Require a number of parameters typically best determined empirically, e.g., the network topology or ``structure."

o Poor interpretability: Difficult to interpret the symbolic meaning behind the learned weights and of ``hidden units" in the network

Strength:

o High tolerance to noisy data

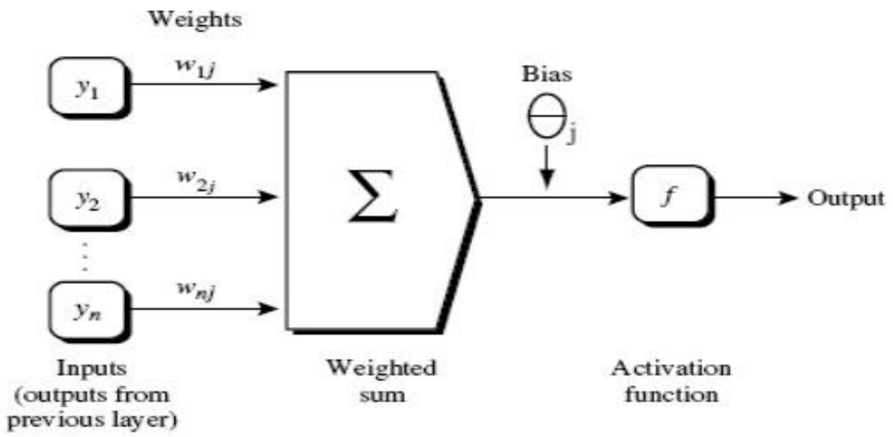
o Ability to classify untrained patterns

o Well-suited for continuous-valued inputs and outputs o Successful on a wide array of real-world data

o Algorithms are inherently parallel

o Techniques have recently been developed for the extraction of rules from trained neural networks

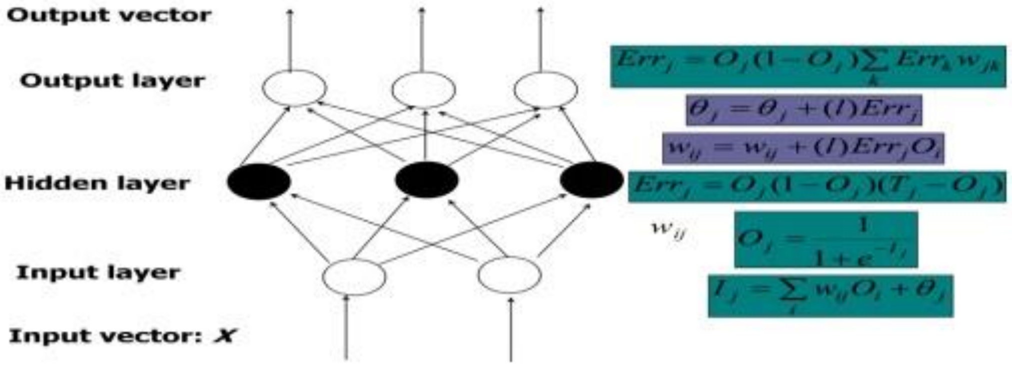
ANeuron(=aperceptron)



A hidden or output layer unit j: The inputs to unit j are outputs from the previous layer. These are multiplied by their corresponding weights in order to form a weighted sum, which is added to the bias associated with unit j. A nonlinear activation function is applied to the net input. (For ease of explanation, the inputs to unit j are labeled $y_1, y_2, \ldots, y_n$. If unit j were in the first hidden layer, then these inputs would correspond to the input tuple $(x_1, x_2, \ldots, x_n)$.)

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

A Multi-Layer Feed-Forward Neural Network


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

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

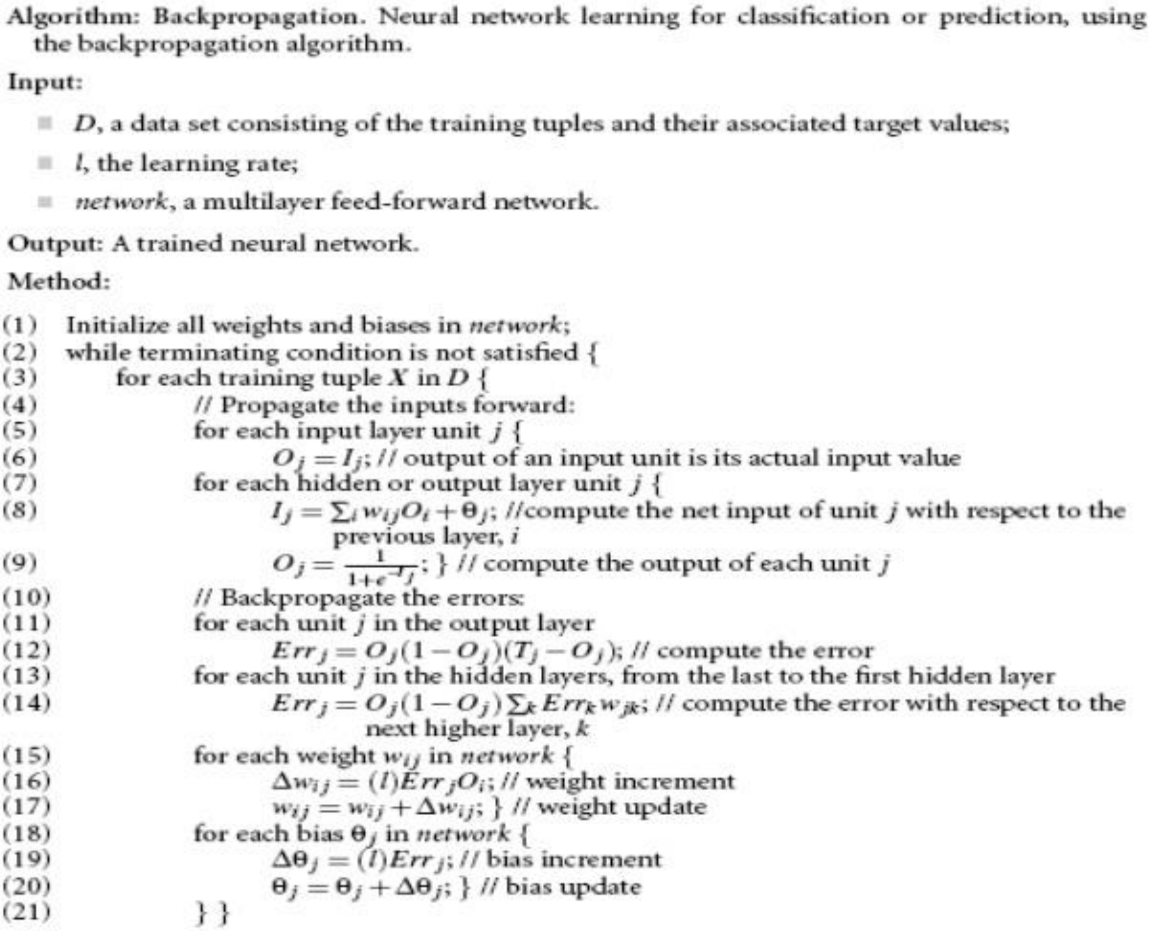
Defining a Network Topology:

- Decide the network topology: Specify # of units in the *input layer*, # of *hidden layers* (if > 1), # of units in *each hidden layer*, and # of units in the *output layer*
- Normalize the input values for each attribute measured in the training tuples to [0.0—1.0]
- One input unit per domain value, each initialized to 0
- Output, if for classification and more than two classes, one output unit per class is used

Once a network has been trained and its accuracy is unacceptable, repeat the training process with a *different.*

Backpropagation

- Iteratively process a set of training tuples & compare the network's prediction with the actual known target value
- For each training tuple, the weights are modified to minimize the mean squared error between the network's prediction and the actual target value
- Modifications are made in the “backwards” direction: from the output layer, through each hidden layer down to the first hidden layer, hence “backpropagation”
- Steps
  - Initialize weights to small random numbers, associated with biases
  - Propagate the inputs forward (by applying activation function)
  - Backpropagate the error (by updating weights and biases)
  - Terminating condition (when error is very small, etc.)



Algorithm: Backpropagation. Neural network learning for classification or prediction, using the backpropagation 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:

```
Initialize all weights and biases in network;
while terminating condition is not satisfied {
    for each training tuple $X$ in $D$ {
        // Propagate the inputs forward:
        for each input layer unit $j$ {
            $O_j = I_j$; // output of an input unit is its actual input value
        for each hidden or output layer unit $j$ {
            $I_j = \sum_i w_{ij} O_i + \theta_j$; //compute the net input of unit $j$ with respect to the
                previous layer, $i$
            $O_j = \frac{1}{1+e^{-I_j}}$; } // compute the output of each unit $j$
        // Backpropagate the errors:
        for each unit $j$ in the output layer
            $Err_j = O_j(1-O_j)(T_j-O_j)$; // compute the error
        for each unit $j$ in the hidden layers, from the last to the first hidden layer
            $Err_j = O_j(1-O_j)\sum_k Err_k w_{jk}$; // compute the error with respect to the
                next higher layer, $k$
        for each weight $w_{ij}$ in network {
            $\Delta w_{ij} = (l) Err_j O_i$; // weight increment
            $w_{ij} = w_{ij} + \Delta w_{ij}$; } // weight update
        for each bias $\theta_j$ in network {
            $\Delta \theta_j = (l) Err_j$; // bias increment
            $\theta_j = \theta_j + \Delta \theta_j$; } // bias update
        } }
```

- **Efficiency** of backpropagation: Each epoch (one iteration through the training set) takes O(|D| * *w*), with |D| tuples and *w* weights, but # of epochs can be exponential to n, the number of inputs, in worst case
- For easier comprehension: **Rule extraction** by network pruning
  - Simplify the network structure by removing weighted links that have the least effect on the trained network
  - Then perform link, unit, or activation value clustering
  - The set of input and activation values are studied to derive rules describing the relationship between the input and hidden unit layers
- **Sensitivity analysis**: assess the impact that a given input variable has on a network output. The knowledge gained from this analysis can be represented in rules.

LAZY LEARNERS (OR LEARNING FROM YOUR NEIGHBORS):

- classification by backpropagation, support vector machines, and classification based on association rule mining—are all examples of *eager learners.*
- Eager learners, when given a set of training tuples, will construct a generalization (i.e., classification) model before receiving new (e.g., test) tuples to classify.
- We can think of the learned model as being ready and eager to classify previously unseen tuples.

k-Nearest-Neighbour Classifiers

- The method is labour intensive when given large training sets, and did not gain popularity until the 1960s when increased computing power became available. It has since been widely used in the area of pattern recognition.
- Nearest-neighbour classifiers are based on learning by analogy, that is, by comparing a given test tuple with training tuples that are similar to it.
- The training tuples are described by *n* attributes. Each tuple represents a point in an *n*-dimensional space.
- In this way, all of the training tuples are stored in an *n*-dimensional pattern space.
- When given an unknown tuple, a ***k***-nearest-neighbor classifier searches the pattern space for the *k* training tuples that are closest to the unknown tuple. These *k* training tuples are the *k* —nearest neighbors‖ of the unknown tuple.
- Closeness‖ is defined in terms of a distance metric, such as Euclidean distance. The
- Euclidean distance between two points or tuples, say, ***X***1 = (*x*11, *x*12, : : : , *x*1*n*) and ***X***2 = (*x*21, *x*22, : : , *x*2*n*), is

$$dist(X_1, X_2) = \sqrt{\sum_{i=1}^{n} (x_{1i} - x_{2i})^2}.$$

CASE-BASED REASONING

- Case-based reasoning (CBR) classifiers use a database of problem solutions to solve new problems.
- Unlike nearest-neighbour classifiers, which store training tuples as points in Euclidean space, CBR stores the tuples or ―cases‖ for problem solving as complex symbolic descriptions.
- Business applications of CBR include problem resolution for customer service help desks, where cases describe product-related diagnostic problems.
- CBR has also been applied to areas such as engineering and law, where cases are either technical designs or legal rulings, respectively.
- Medical education is another area for CBR, where patient case histories and treatments are used to help diagnose and treat new patients.
- When given a new case to classify, a case-based reasoner will first check if an identical training case exists.
- If one is found, then the accompanying solution to that case is returned.
- If no identical case is found, then the case-based reasoner will search for training cases having SCE Department of Information Technology components that are similar to those of the new case.
- Conceptually, these training cases may be considered as neighbours of the new case. If cases are represented as graphs, this involves searching for subgraphs that are similar to subgraphs within the new case.
- The case-based reasoner tries to combine the solutions of the neighbouring training cases in order to propose a solution for the new case.
- If incompatibilities arise with the individual solutions, then backtracking to search for other solutions may be necessary.
- The case-based reasoner may employ background knowledge and problem-solving strategies in order to propose a feasible combined solution.

OTHER CLASSIFICATION METHODS

GENETIC ALGORITHMS:

- Genetic Algorithm: based on an analogy to biological evolution.
- An initial **population** is created consisting of randomly generated rules
- Each rule is represented by a string of bits.

E.g., if A_1 and $\neg A_2$ then C_2 can be encoded as 100

o If an attribute has k > 2 values, k bits can be used.

- Based on the notion of survival of the **fittest**, a new population is formed to consist of the fittest rules and their offsprings.
- The fitness of a rule is represented by its *classification accuracy* on a set of training examples.
- Offsprings are generated by *crossover* and *mutation*.
- The process continues until a population P evolves *when each rule in P satisfies a prespecified threshold*.
- Slow but easily parallelizable

Rough Set Approach:

- Rough sets are used to **approximately or —roughly‖ define equivalent classes**.
- A rough set for a given class C is approximated by two sets: a lower approximation (certain to be in C) and an upper approximation (cannot be described as not belonging to C).
- Finding the minimal subsets (**reducts**) of attributes for feature reduction is NP-hard but a **discernibility matrix** (which stores the differences between attribute values for each pair of data tuples) is used to reduce the computation intensity

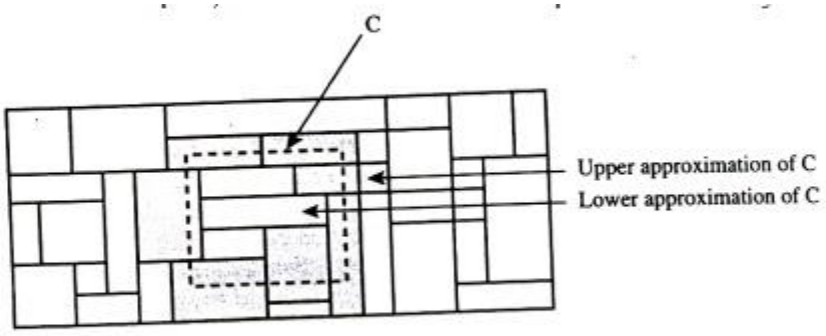


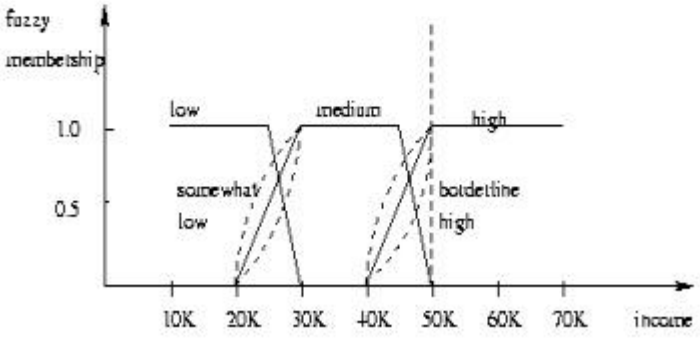
Figure: A rough set approximation of the set of tuples of the class C suing lower and upper approximation sets of C. The rectangular regions represent equivalence classes

Fuzzy Set approaches:

- Fuzzy logic uses truth values between 0.0 and 1.0 to represent the degree of membership (such as using fuzzy membership graph)
- Attribute values are converted to fuzzy values

e.g., income is mapped into the discrete categories {low, medium, high} with fuzzy values calculated

- For a given new sample, more than one fuzzy value may apply.
- Each applicable rule contributes a vote for membership in the categories.
- Typically, the truth values for each predicted category are summed, and these sums are combined.



UNIT IV

CLUSTER ANALYSIS

- Definition-Types of data in cluster analysis
- Categorization of clustering techniques
- Partitioning method
- Hierarichal method
  - -BIRCH
  - -ROCK
- Grid based methods
- Model based method
- Outlier analysis

What is Cluster Analysis?

A cluster is a collection of data objects that are similar to one another within the same cluster and are dissimilar to the objects in other clusters.

The following are typical requirements of clustering in data mining:

- Scalability
- Ability to deal with different types of attributes.
- Discovery of clusters with arbitrary shape
- Minimal requirements for domain knowledge to determine input parameters
- Ability to deal with noisy data
- Incremental clustering and insensitivity to the order of input records
- High dimensionality
- Constraint-based clustering
- Interpretability and usability.

TYPE OF DATA IN CLUSTERING ANALYSIS:

Main memory-based clustering algorithms typically operate on either of the following two data structures.

Data structure Data matrix (two modes) object by variable Structure:

This represents n objects, such as persons, with p variables (also called measurements or attributes)

The structure is in the form of a relational table, or n-by-p matrix (n objects ×p variables):

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

Dissimilarity matrix (one mode) object –by-object structure:

- This stores a collection of proximities that are available for all pairs of n objects. It is often represented by an n-by-n table.

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

- We describe how object dissimilarity can be computed for object by Interval-scaled variables,
- Binary variables, Nominal, ordinal, and ratio variables, Variables of mixed types
- Interval-Scaled variables (continuous measurement of a roughly linear scale) Standardize data

Calculate the mean absolute deviation:

$$s_f = \frac{1}{n}(|x_{1f} - m_f| + |x_{2f} - m_f| + \ldots + |x_{nf} - m_f|)$$

Where

$$m_f = \frac{1}{n}(x_{1f} + x_{2f} + \ldots + x_{nf}).$$

Calculate the standardized measurement (*z-score*)

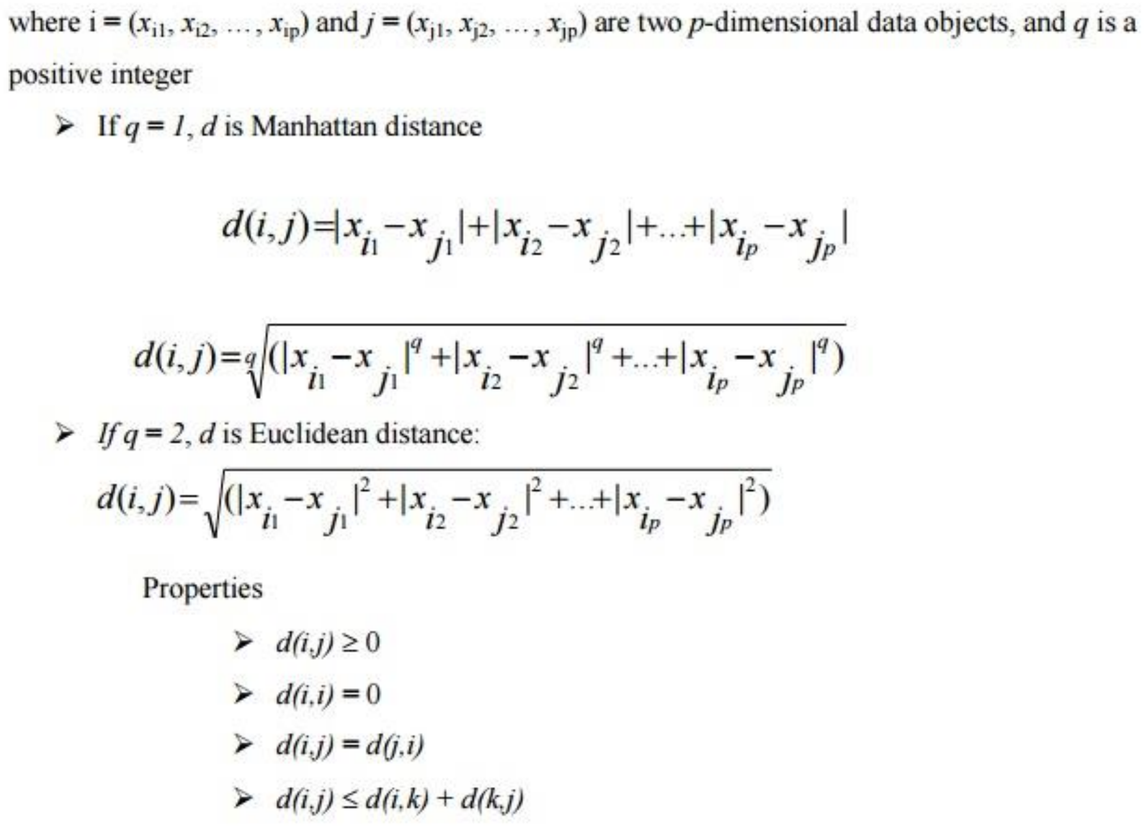
$$z_{if} = \frac{x_{if} - m_f}{s_f}$$

- Using mean absolute deviation is more robust than using standard deviation.

Similarity and Dissimilarity Between Objects:

Distances are normally used to measure the similarity or dissimilarity between two data objects

Some popular ones include: *Minkowski distance*:



where i = ($x_{i1}, x_{i2}, \ldots, x_{ip}$) and *j* = ($x_{j1}, x_{j2}, \ldots, x_{jp}$) are two *p*-dimensional data objects, and *q* is a positive integer

- If $q = 1$, d is Manhattan distance

$$d(i,j) = |x_{i_1} - x_{j_1}| + |x_{i_2} - x_{j_2}| + \ldots + |x_{i_p} - x_{j_p}|$$

$$d(i,j) = \sqrt[q]{(|x_{i_1} - x_{j_1}|^q + |x_{i_2} - x_{j_2}|^q + \ldots + |x_{i_p} - x_{j_p}|^q)}$$

- *If* $q = 2$, d is Euclidean distance:

$$d(i,j) = \sqrt{(|x_{i_1} - x_{j_1}|^2 + |x_{i_2} - x_{j_2}|^2 + \ldots + |x_{i_p} - x_{j_p}|^2)}$$

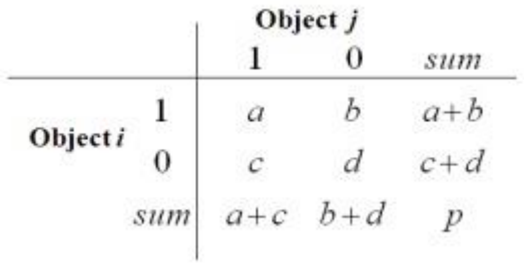
Properties

- $d(i,j) \geq 0$
- $d(i,i) = 0$
- $d(i,j) = d(j,i)$
- $d(i,j) \leq d(i,k) + d(k,j)$

- Also, one can use weighted distance, parametric Pearson product moment correlation, or other dissimilarity measures

<u>Binary Variables</u>

- A contingency table for binary data



		Object *j*		
		1	0	*sum*
Object *i*	1	*a*	*b*	*a+b*
	0	*c*	*d*	*c+d*
	sum	*a+c*	*b+d*	*p*

- Distance measure for symmetric binary variables:

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

- Distance measure for asymmetric binary variables:

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

- Jaccard coefficient (*similarity* measure for *asymmetric* binary variables):

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

Categorical variables:

A generalization of the binary variable in that it can take more than 2 states, **e.g.,** red, yellow, blue, green

Method 1: Simple matching

m: # of matches, *p*: total # of variables

$$d(i,j)=\frac{p-m}{p}$$

Method 2: use a large number of binary variables.

- creating a new binary variable for each of the *M* nominal states

Ordinal Variables

- An ordinal variable can be discrete or continuous
- Order is important, **e.g., rank**
- Can be treated like interval-scaled
- replace x_{if} by their rank
- map the range of each variable onto [0, 1] by replacing *i*-th object in the *f*-th variable

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

- compute the dissimilarity using methods for interval-scaled variables

Ratio-scaled variable:

A positive measurement on a nonlinear scale, approximately at exponential scale, such as Ae^{Bt} or $_{Ae}$-*Bt*

Methods:

- Apply logarithmic transformation $y_{if} = log(x_{if})$.
- continuous ordinal data treat their rank as interval-scaled.

Variables of Mixed Types

A database may contain all the six types of variables symmetric binary, asymmetric binary,nominal, ordinal, interval and ratio

One may use a weighted formula to combine their effects

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

Vector Objects

Vector objects: keywords in documents, gene features in micro-arrays, etc.

Broad applications: information retrieval, biologic taxonomy, etc.

Cosine measure:

$$s(\vec{X},\vec{Y}) = \frac{\vec{X}^t \cdot \vec{Y}}{|\vec{X}||\vec{Y}|},$$

A variant: Tanimoto coefficient

$$s(\vec{X},\vec{Y}) = \frac{\vec{X}^t \cdot \vec{Y}}{\vec{X}^t \cdot \vec{X} + \vec{Y}^t \cdot \vec{Y} - \vec{X}^t \cdot \vec{Y}},$$

Clustering Methods:

The clustering methods can be classified into following categories:

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

1. K-means

Given *k*, the *k-means* algorithm is implemented in four steps:

1. Partition objects into *k* nonempty subsets
2. Compute seed points as the centroids of the clusters of the current partition (the centroid is the center, i.e., *mean point*, of the cluster)
3. Assign each object to the cluster with the nearest seed point
4. Go back to Step 2, stop when no more new assignment

2. Partitioning Method

Suppose we are given a database of n objects, the partitioning method construct k partition of data.

Each partition will represent a cluster and k≤n. It means that it will classify the data into k groups, which satisfy the following requirements:

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

Typical methods:

- **K-means**: where each cluster is represented by the mean value of the objects in the cluster
- **k-medoids**: where each cluster is represented by one of the objects located near the center of the cluster.
- **CLARANS**:

3 Hierarchical Methods

- A hierarchical method creates a hierarchical decomposition of the given set of data objects.
- This method creates the hierarchical decomposition of the given set of data objects.:
  - Agglomerative Approach
  - Divisive Approach

Density-based methods:

- Most partitioning methods cluster objects based on the distance between objects.
- Such methods can find only spherical-shaped clusters and encounter difficulty at discovering clusters of arbitrary shapes.
- Other clustering methods have been developed based on the notion of *density*. Their general idea is to continue growing the given cluster as long as the density (number of objects or datapoints) in the "neighborhood" exceeds some threshold,
  - DBSCAN
  - OPTICS
  - DENCLUE

Grid-based methods:

- Grid-based methods quantize the object space into a finite number of cells that form a grid structure.
- All of the clustering operations are performed on the grid structure (i.e., on the quantized space).
- The main advantage of this approach is its fast processing time.

➢ STING

Model-based methods:

- Model-based methods hypothesize a model for each of the clusters and find the best fit of the data to the given model.
- A model-based algorithm may locate clusters by constructing a density function that reflects the spatial distribution of the data points.
- EM is an algorithm that performs expectation-maximization analysis based on statistical modeling.
- COBWEB is a conceptual learning algorithm that performs probability analysis and takes *concepts* as a model for clusters.
- SOM (or self-organizing feature map) is a neural network base algorithm.

Constraint-based clustering:

- It is a clustering approach that performs clustering by incorporationof user-specified or application-oriented constraints.
- A constraint expressesa user's expectation or describes "properties" of the desired clustering results, and provides an effective means for communicating with the clustering process.

- Various kinds of constraints can be specified, either by a user or as per application requirements.

PARTITIONING METHODS:

The most well-known and commonly used partitioning methods are

- The k-Means Method
- k-Medoids Method

Centroid-Based Technique:

The K-Means Method: The k-means algorithm takes the input parameter, k, and partitions a set of n objects into k clusters so that the resulting intra cluster similarity is high but the inter cluster similarity is low.

Cluster similarity is measured in regard to the mean value of the objects in a cluster, which can be viewed as the cluster's centroid or centre of gravity.

The k-means algorithm proceeds as follows:

- First, it randomly selects k of the objects, each of which initially represents a cluster mean or centre.
- For each of the remaining objects, an object is assigned to the cluster to which it is the most similar, based on the distance between the object and the cluster mean.
- It then computes the new mean for each cluster.
- This process iterates until the criterion function converges.
- Typically, the square-error criterion is used, defined as
  - Where E is the sum of the square error for all objects in the data set
  - p is the point in space representing a given object
  - mi is the mean of cluster Ci.

The k-means partitioning algorithm:

The k-means algorithm for partitioning, where each cluster's center is represented by the mean value of the objects.

Algorithm: *k*-means. The *k*-means algorithm for partitioning, where each cluster's

center is represented by the mean value of the objects in the cluster.

Input:

k: the number of clusters,

D

: a data set containing *n* objects.

Output: A set of *k* clusters.

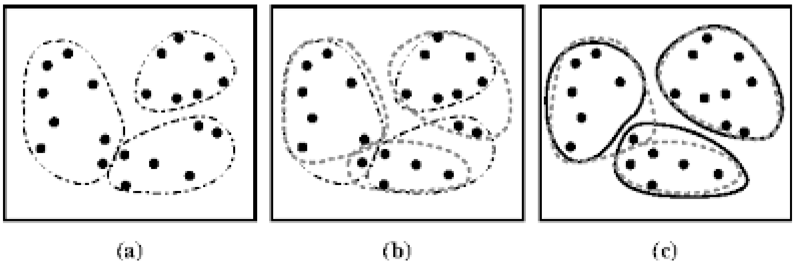
Method:

(1) arbitrarily choose *k* objects from *D* as the initial cluster centers;

(2) repeat

(3) (re)assign each object to the cluster to which the object is the most similar, based on the mean value of the objects in the cluster;
(4) update the cluster means, i.e., calculate the mean value of the objects for each cluster;
(5) until no change;

Figure 7.2 The *k*-means partitioning algorithm.



The k-Medoids Method:

- The k-means algorithm is sensitive to outliers because an object with an extremely large value may substantially distort the distribution of data.
- This effect is particularly exacerbated due to the use of the square-error function.
- Instead of taking the mean value of the objects in a cluster as a reference point, we can pick actual objects to represent the clusters, using one representative object per cluster.
- Each remaining object is clustered with the representative object to which it is the most similar.
- The partitioning method is then performed based on the principle of minimizing the sum of the dissimilarities between each object and its corresponding reference point.
- That is, an absolute-error criterion is used, defined as

$$E =_k \sum_{j=1} \sum_{p2C_j} |p-o_j|$$

- Where E is the sum of the absolute error for all objects in the data set
- p is the point in space representing a given object in cluster.
- Cj ojis the representative object of Cj.
- The initial representative objects are chosen arbitrarily.
- The iterative process of replacing representative objects by non representative objects continues as long as the quality of the resulting clustering is improved.
- This quality is estimated using a cost function that measures the average dissimilaritybetween an object and the representative object of its cluster.
- To determine whether a non representative object, oj random, is a good replacement for a current representativeobject, oj, the following four cases are examined for each of the nonrepresentative objects.

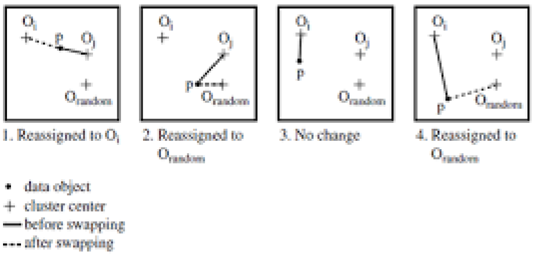
Case 1:

P currently belongs to representative object, oj . If ojis replaced by o_{random} as a representative object and p is closest to one of the other representative objects, oi ,i≠j, then p is reassigned to oi .

Case 2: p currently belongs to representative object, oj. If ojis replaced by o_{random} asa representative object and p is closest to o_{random} then p is reassigned to o_{random}.

Case 3: p currently belongs to representative object, oi , i≠j. If ojis replaced by o_{random} as a representative object and p is still closest to oi , then the assignment does notchange.

Case 4: p currently belongs to representative object, oi, i≠j. If ojis replaced by o_{random} as a representative object and p is closest to o_{random}, then p is reassigned to o_{random}



1. Reassigned to O$_i$ 2. Reassigned to O$_{random}$ 3. No change 4. Reassigned to O$_{random}$

Figure 7.4 Four cases of the cost function for *k*-medoids clustering.

The k-Medoids Algorithm:

The k-medoids algorithm for partitioning based on medoid or central objects.

Algorithm: *k*-medoids. PAM, a *k*-medoids algorithm for partitioning based on medoid
or central objects.
Input:
k: the number of clusters,
D: a data set containing *n* objects.
Output: A set of *k* clusters.
Method:

```
(1) arbitrarily choose k objects in D as the initial representative objects or seeds;
(2) repeat
(3) assign each remaining object to the cluster with the nearest representative object;
(4) randomly select a nonrepresentative object, orandom;
(5) compute the total cost, S, of swapping representative object, oj, with orandom;
(6) if S < 0 then swap oj with orandom to form the new set of k representative objects;
 (7) until no change;
```

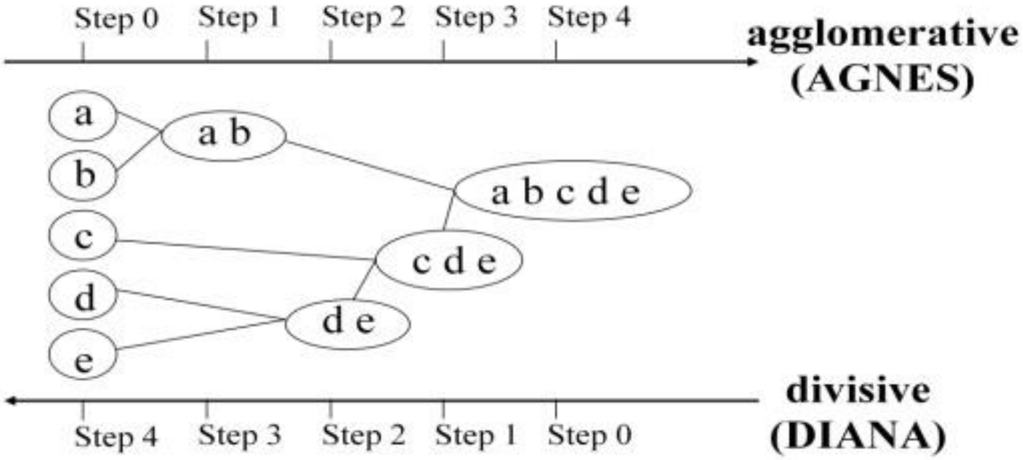
HIERARCHICAL METHODS:

AGGLOMERATIVE METHOD:

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

Divisive Approach

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



Disadvantage

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

Approaches to improve quality of Hierarchical clustering

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

Perform careful analysis of object linkages at each hierarchical partitioning.

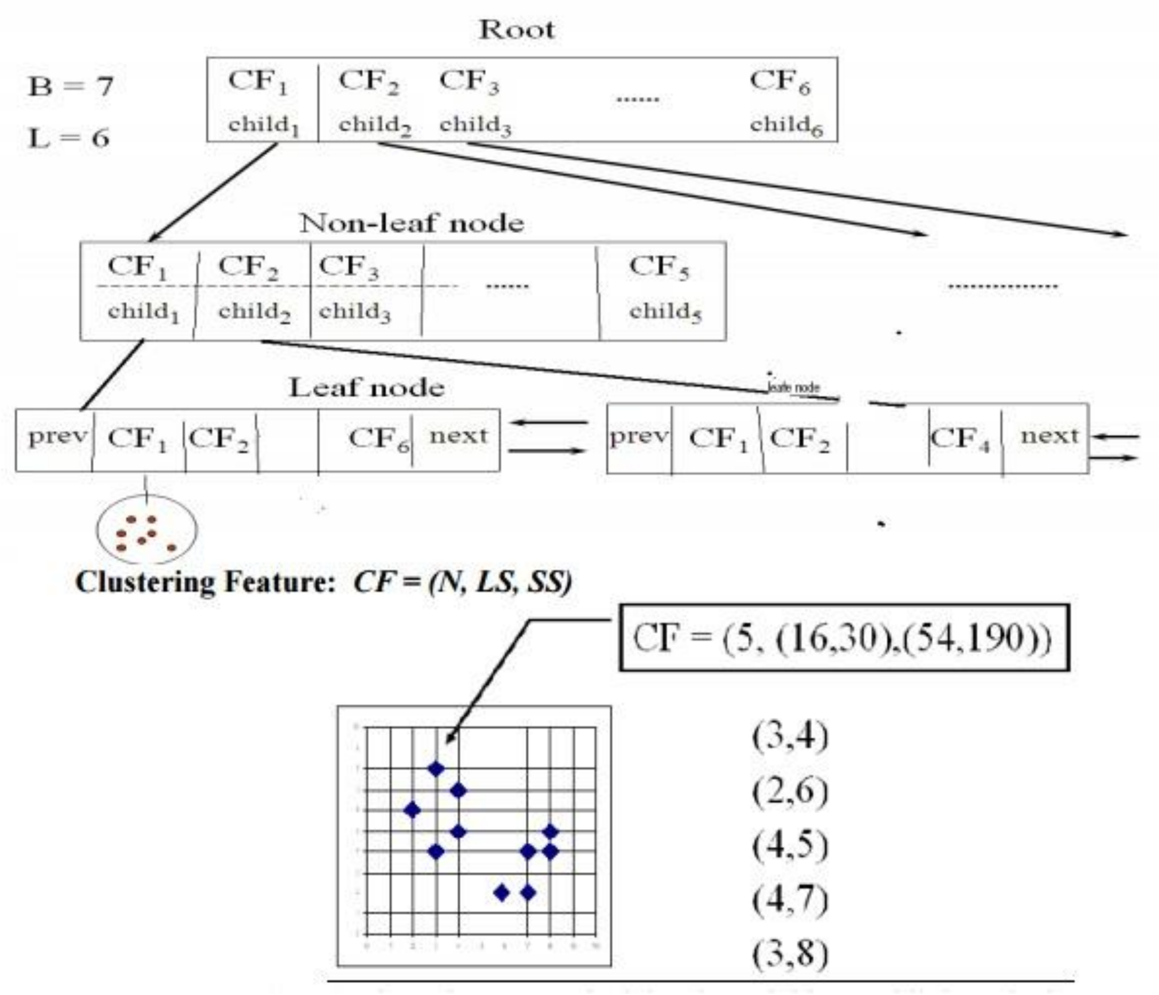
Integrate hierarchical agglomeration by first using a hierarchical agglomerative algorithm to group objects into microclusters, and then performing macroclustering on the microclusters. Typical methods: Diana, Agnes, BIRCH, ROCK, CAMELEON

<u>BIRCH (1996)BALANCED ITERATIVE REDUCING AND CLUSTERING USING HIERARCHIES</u>:

- uses CF-tree and incrementally adjusts the quality of sub-clusters
- Incrementally construct a CF (Clustering Feature) tree, a hierarchical data structure for multiphase clustering

Phase 1: scan DB to build an initial in-memory CF tree (a multi-level compression of the data that tries to preserve the inherent clustering structure of the data)

Phase 2: use an arbitrary clustering algorithm to cluster the leaf nodes of the CF-tree.



ROCK (1999):

- clustering categorical data by neighbor and link analysis .
  - Robust Clustering using links

Major ideas

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

Algorithm: sampling-based clustering

- o Draw random sample
- o Cluster with links
- o Label data in disk

DENSITY-BASED METHOD:

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

Major features:

- o Discover clusters of arbitrary shape
- o Handle noise
- o One scan
- o Need density parameters as termination condition

Two parameters*:*

- o *Eps*: Maximum radius of the neighbourhood
- o *MinPts*: Minimum number of points in an Eps-neighbourhood of that point

Typical methods: DBSACN, OPTICS, DenClue.

DBSCAN: Density Based Spatial Clustering of Applications with Noise.

Relies on a *density-based* notion of cluster: A *cluster* is defined as a maximal set of density-connected points.

Discovers clusters of arbitrary shape in spatial databases with noise

DBSCAN: The Algorithm

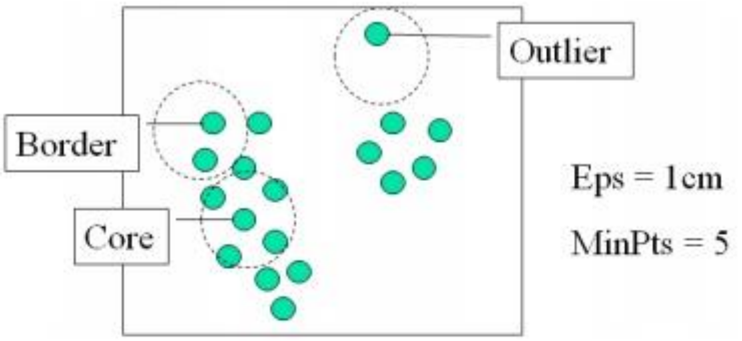
Arbitrary select a point *p*

Retrieve all points density-reachable from *p* w.r.t. *Eps* and *MinPts*.

If *p* is a core point, a cluster is formed.

If *p* is a border point, no points are density-reachable from *p* and DBSCAN visits the next point of the database.

Continue the process until all of the points have been processed.



OPTICS: Ordering Points to Identify the Clustering Structure

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

DENCLUE: DENsity-based CLUstEring:

Major features

- Solid mathematical foundation
- Good for data sets with large amounts of noise
- Allows a compact mathematical description of arbitrarily shaped clusters in high-dimensional data sets
- Significant faster than existing algorithm (e.g., DBSCAN)

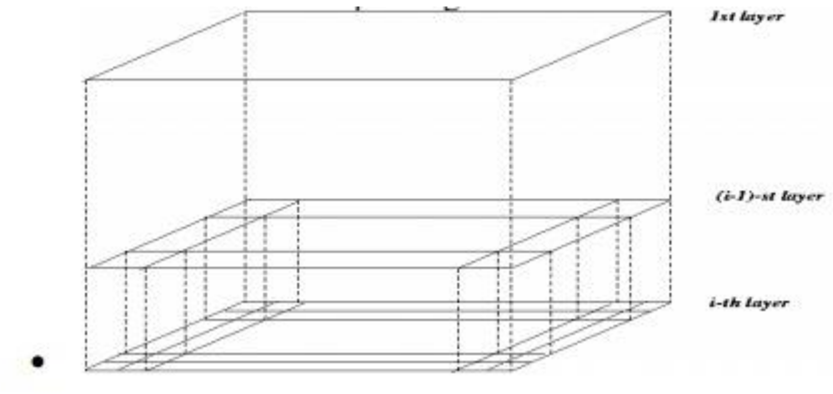
- But needs a large number of parameters

5. GRID-BASED METHOD

Using multi-resolution grid data structure:

Advantage

- The major advantage of this method is fast processing time.
- It is dependent only on the number of cells in each dimension in the quantized space.
- Typical methods: STING, Wave Cluster, CLIQUE
- STING: a STatistical INformation Grid approach
- The spatial area area is divided into rectangular cells
- There are several levels of cells corresponding to different levels of resolution



- Each cell at a high level is partitioned into a number of smaller cells in the next lower level
- Statistical info of each cell is calculated and stored beforehand and is used to answer queries
- Parameters of higher level cells can be easily calculated from parameters of lower level cell

count, *mean*, *s*, *min*, *max*

type of distribution—normal, *uniform*, etc.

- Use a top-down approach to answer spatial data queries
- Start from a pre-selected layer—typically with a small number of cells
- For each cell in the current level compute the confidence interval.

WaveCluster: Clustering by Wavelet Analysis.

A multi-resolution clustering approach which applies wavelet transform to the feature space.

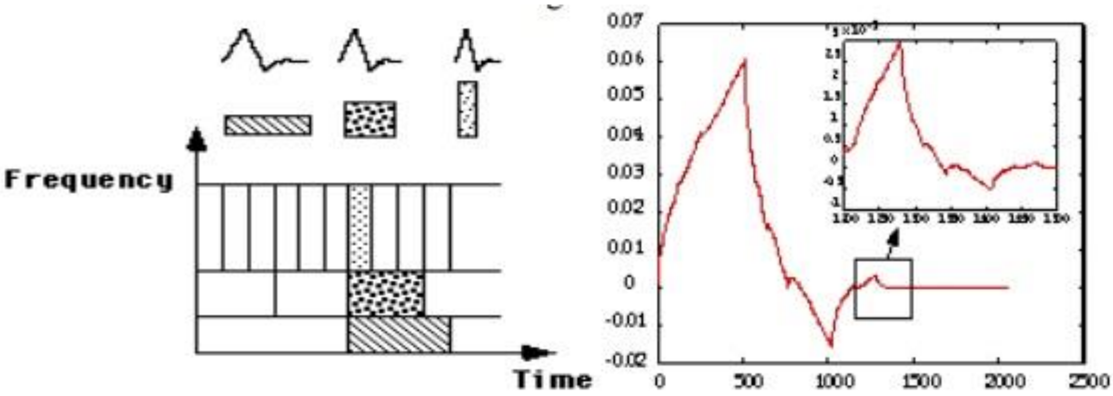
How to apply wavelet transform to find clusters

- o Summarizes the data by imposing a multidimensional grid structure onto data space.
- o These multidimensional spatial data objects are represented in a n-dimensional feature space.
- o Apply wavelet transform on feature space to find the dense regions in the feature space.
- o Apply wavelet transform multiple times which result in clusters at different scales from fine to coarse.

- **Wavelet transform:** A signal processing technique that decomposes a signal into different frequency sub-band (can be applied to n-dimensional signals).

  Data are transformed to preserve relative distance between objects at different levels of resolution.

  Allows natural clusters to become more distinguishable



MODEL-BASED METHODS:

- Attempt to optimize the fit between the given data and some mathematical model
- Based on the assumption: Data are generated by a mixture of underlying probability distribution
- In this method a model is hypothesize for each cluster and find the best fit of data to the given model.

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

Typical methods:

- ❖ EM,
- ❖ SOM
- ❖ COBWEB

EM :EXPECTATION AND MAXIMIZATION

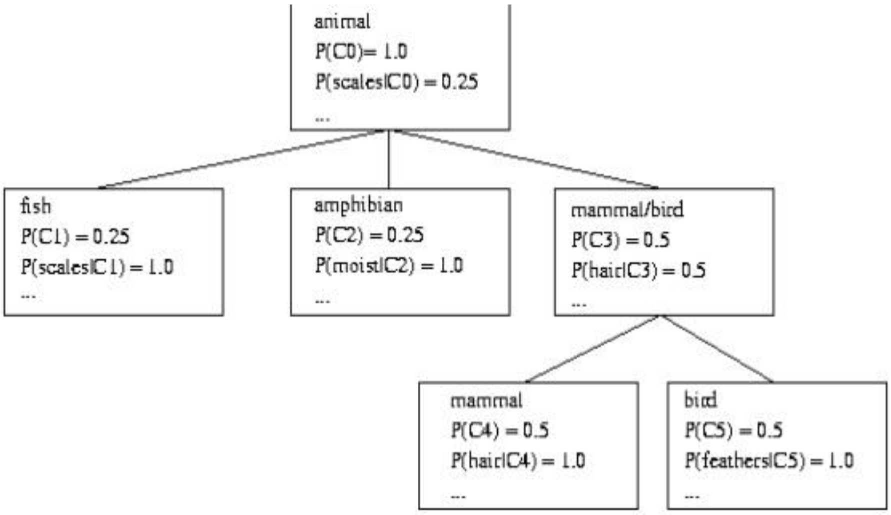
A popular iterative refinement algorithm

- An extension to k-means
- Assign each object to a cluster according to a weight (prob. distribution)
- New means are computed based on weighted measures
- Starts with an initial estimate of the parameter vector
- Iteratively rescores the patterns against the mixture density produced by the parameter vector
- The rescored patterns are used to update the parameter updates
- Patterns belonging to the same cluster, if they are placed by their scores in a particular component
- Algorithm converges fast but may not be in global optima

CONCEPTUAL CLUSTERING:

COBWEB (Fisher'87)

- A popular a simple method of incremental conceptual learning
- Creates a hierarchical clustering in the form of a classification tree
- Each node refers to a concept and contains a probabilistic description of that concept



SOM (Soft-Organizing feature Map):

o Competitive learning

º Involves a hierarchical architecture of several units (neurons)

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

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

- It maps all the points in a high-dimensional source space into a 2 to 3-d target space, the distance and proximity relationship (i.e., topology) are preserved as much as possible
- Similar to k-means: cluster centers tend to lie in a low-dimensional manifold in the feature space
- Clustering is performed by having several units competing for the current object.
- The unit whose weight vector is closest to the current object wins.
- The winner and its neighbours learn by having their weights adjusted
- SOMs are believed to resemble processing that can occur in the brain
- Useful for visualizing high-dimensional data in 2- or 3-D space.

OUTLIER ANALYSIS

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

o Example: Sports: Michael Jordon, Wayne Gretzky,

Problem: Define and find outliers in large data sets

Applications:

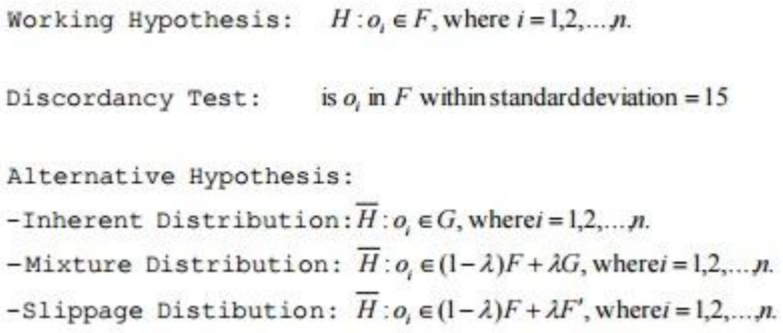
- Credit card fraud detection
- Telecom fraud detection
- Customer segmentation
- Medical analysis

Statistical Distribution-based outlier detection-Identify the outlier with respect to the model using discordancy test

How discordancy test work

Data is assumed to be part of a working hypothesis (working hypothesis)-H

Each data object in the dataset is compared to the working hypothesis and is either accepted in the working hypothesis or rejected as discordant into an alternative hypothesis (outliers)- H



Working Hypothesis: $H : o_i \in F$, where $i = 1,2,\ldots,n.$

Discordancy Test: is o_i in F withinstandarddeviation $= 15$

Alternative Hypothesis:

-Inherent Distribution: $\overline{H} : o_i \in G$, where$i = 1,2,\ldots,n.$

-Mixture Distribution: $\overline{H} : o_i \in (1-\lambda)F + \lambda G$, where$i = 1,2,\ldots,n.$

-Slippage Distibution: $\overline{H} : o_i \in (1-\lambda)F + \lambda F'$, where$i = 1,2,\ldots,n.$

Distance-Based outlier detection

- Imposed by statistical methods
- We need multi-dimensional analysis without knowing data distribution Algorithms for mining distance-based outliers

Index-based algorithm

- Indexing Structures such as R-tree (R+-tree), K-D (K-D-B) tree are built for the multi-dimensional database
- The index is used to search for neighbors of each object O within radius D around that object.
- Once K (K = N(1-p)) neighbors of object O are found, O is not an outlier.

Worst-case computation complexity is O(K*n^2), K is the dimensionality and n is the number of objects in the dataset.

Pros: scale well with K

- **Cons:** the index construction process may cost much time
- Nested-loop algorithm
- Divides the buffer space into two halves (first and second arrays)
- Break data into blocks and then feed two blocks into the arrays.
- Directly computes the distance between each pair of objects, inside the array or between arrays

Example:

Same computational complexity as the index-based algorithm

- Pros: Avoid index structure construction
- Try to minimize the I/Os ^{n} cell based algorithm

- Divide the dataset into cells with length
- K is the dimensionality, D is the distance

Define Layer-1 neighbors – all the intermediate neighbor cells. The maximum distance between a cell and its neighbor cells is D

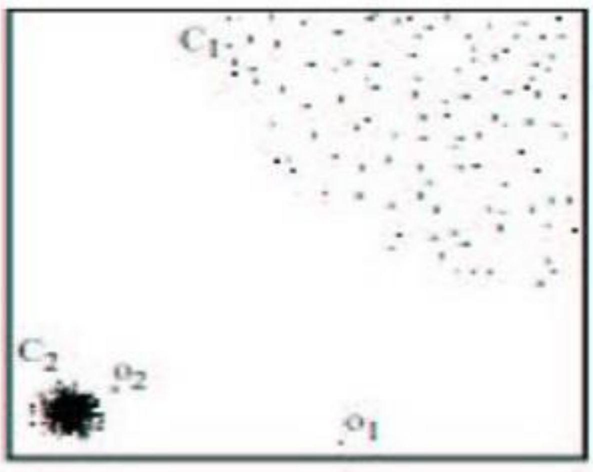
Define Layer-2 neighbors – the cells within 3 cell of a certain cell. The minimum distance between a cell and the cells outside of Layer-2 neighbors is D

Criteria

- Search a cell internally. If there are M objects inside, all the objects in this cell are not outlier.
- Search its layer-1 neighbors. If there are M objects inside a cell and its layer-1 neighbors, all the objects in this cell are not outlier.
- Search its layer-2 neighbors. If there are less than M objects inside a cell, its layer-1 neighbor cells, and its layer-2 neighbor cells, all the objects in this cell are outlier
- Otherwise, the objects in this cell could be outlier, and then need to calculate the distance between the objects in this cell and the objects in the cells in the layer-2 neighbor cells to see whether the total points within D distance is more than M or not.

Density-Based Local Outlier Detection

- Distance-based outlier detection is based on global distance distribution
- It encounters difficulties to identify outliers if data is not uniformly distributed
- **Ex.** C_1 contains 400 loosely distributed points, C_2 has 100 tightly condensed points, 2 outlier points o_1, o_2
- Some outliers can be defined as global outliers, some can be defined as local outliers to a given cluster
- O_2 would not normally be considered an outlier with regular distance-based outlier detection, since it looks at the global picture
- Each data object is assigned a *local outlier factor (LOF)*
- Objects which are closer to dense clusters receive a higher LOF
- LOF varies according to the parameter MinPts



Deviation-Based Outlier detection

- Identifies outliers by examining the main characteristics of objects in a group
- Objects that ―deviate‖ from this description are considered outliers

Sequential exception technique

- simulates the way in which humans can distinguish unusual objects from among a series of supposedly like objects
- Dissimilarities are assed between subsets in the sequence the techniques introduce the

OLAP data cube technique

- Deviation detection process is overlapped with cube computation.
- Recomputed measures indicating data exceptions are needed.
- A cell value is considered an exception if it is significantly different from the expected value, based on a statistical model.
- Use visual cues such as background color to reflect the degree of exception.

UNIT V

SPATIAL, MULTIMEDIA, TEXT AND WEB DATA

- Spatial Data Mining
- Multimedia Data Mining
- Text Mining
- Mining The World Wide Web
- Data Mining Application
- Trends in Data Mining

SPATIAL DATA MINING:

- A spatial database stores a large amount of space-related data, such as maps, pre-processed remote sensing or medical imaging data, and VLSI chip layout data.
- Spatial databases have many features distinguishing them from relational databases.
- Spatial data mining refers to the extraction of knowledge, spatial relationships, or other interesting patterns not explicitly stored in spatial databases.
- Such mining demands an integration of data mining with spatial database technologies.
- It can be used for understanding spatial data, discovering spatial relationships and relationships between spatial and nonspatial data, constructing spatial knowledge bases, reorganizing spatial databases, and optimizing spatial queries.
- It is expected to have wide applications in geographic information systems, geo marketing, remote sensing, image database exploration, medical imaging, navigation, traffic control, environmental studies, and many other areas where spatial data are used.
- A crucial challenge to spatial data mining is the exploration of *efficient* spatial data mining techniques due to the huge amount of spatial data and the complexity of spatial data types and spatial access methods.

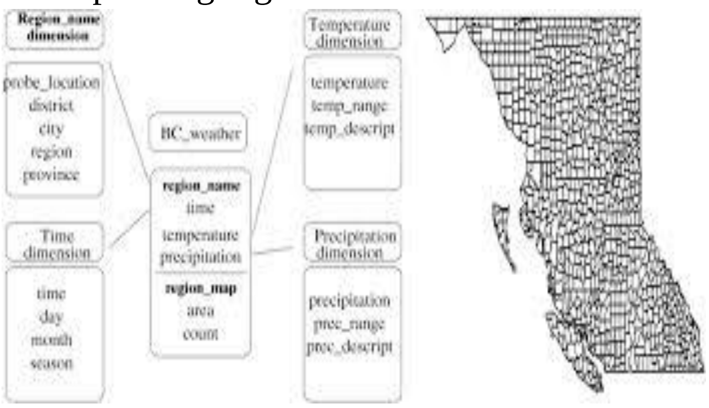
"Can we construct a spatial data warehouse" :
A spatial data warehouse is a *subject-oriented, integrated, time-variant*, and *nonvolatile* collection of both spatial and nonspatial data in support of spatial data mining and spatial-datarelated decision-making processes.

<u>There are three types of *dimensions* in a spatial data cube</u>

- A **nonspatial dimension** contains only nonspatial data. (such as *"hot"* for *temperature* and *"wet"* for *precipitation*)
- A **spatial-to-nonspatial dimension** is a dimension whose primitive-level data are spatial but whose generalization, starting at a certain high level, becomes nonspatial.eg:-city
- **A spatial-to-spatial dimension** is a dimension whose primitive level and all of its highlevel generalized data are spatial.eg:equitemp.

Example :Numerical versus spatial measures.

- A star schema for the BC weather warehouse . It consists of four dimensions: region temperature, time, and precipitation, and three measures: region map, area, and count. A concept hierarchy for each dimension can be created by users or experts, or generated automatically .Spatial Data Mining 603 by data clustering analysis. Figure 10.3 presents hierarchies for each of the dimensions in the BC weather warehouse. Of the three measures, area and count are numerical measures that can be computed similarly as for nonspatial data cubes; region map is a spatial measure that represents a collection of spatial pointers to the corresponding regions.



-
- Figure 10.2 A star schema of the BC weather spatial data warehouse and corresponding BC weather probes map.

Two types of *measures* in a spatial data cube:

- A numerical measure contains only numerical data. For example, one measure in a
- spatial data warehouse could be the *monthly revenue* of a region
- A spatial measure contains a collection of pointers to spatial objects.eg *temperature* and *precipitation*

There are several challenging issues regarding the construction and utilization of spatial datawarehouses:

- The first challenge is the integration of spatial data from heterogeneous sources and systems.
- The second challenge is the realization of fast and flexible on-line analytical processing in spatial data warehouses.

Mining Spatial Association

- A spatial association rule is of the form *A*=>*B* [*s*%;*c*%], where *A* and *B* are sets of spatial or nonspatial predicates, *s*% is the support of the rule, and *c*%is the confidence of the rule.
- For example, the following is a spatial association rule: *is a*(*X*; "*school*")^*close to*(*X*; "*sports center*"))=>*close to*(*X*; "*park*") [0:5%;80%].
- This rule states that 80% of schools that are close to sports centers are also close to parks, and 0.5% of the data belongs to such a case.

MULTIMEDIA DATA MINING:

A multimedia database system stores and manages a large collection of *multimedia data*, such as audio, video, image, graphics, speech, text, document, and hypertext data, which contain text, text markups, and linkages.

Similarity Search in Multimedia Data

For similarity searching in multimedia data, we consider two main families of multimedia indexing and retrieval systems:

- **description-based retrieval systems**:which build indices and perform object retrieval based on image descriptions, such as keywords, captions, size, and time of creation;
- **content-based retrieval systems**: which support retrieval based on the image content, such as color histogram, texture, pattern, image topology, and the shape of objects and their layouts and locations within the image.

<u>In a content-based image retrieval system, there are often two kinds of queries:</u>

Image sample- based queries and image feature specification queries:

· Image-sample-based queries find all of the images that are similar to the given image sample. This search compares the feature vector (or signature) extracted from the sample with the feature vectors of images that have already been extracted and indexed in the image database. Based on this comparison, images that are close to the sample image are returned.

· Image feature specification queries specify or sketch image features like color, texture, or shape, which are translated into a feature vector to be matched with the feature vectors of the images in the database

<u>Mining Associations in Multimedia Data:</u>

- **Associations between image content and non image content features:**

A rule like "If at least 50% of the upper part of the picture is blue, then it is likely to represent sky" belongs to this category since it links the image content to the keyword sky.

- **Associations among image contents that are not related to spatial relationships:**

  A rule like "If a picture contains two blue squares, then it is likely to contain one red circle a swell" belongs to this category since the associations are all regarding image contents.

- **Associations among image contents related to spatial relationships:**

  A rule like "If a red triangle is between two yellow squares, then it is likely a big oval-shaped object is underneath" belongs to this category since it associates objects in the image with spatial relationship.

Several approaches have been proposed and studied for similarity-based retrieval in image databases, based on image signature

- **Color histogram–based signature:**

  In this approach, the signature of an image includes color histograms based on the color composition of an image regardless of its scale or orientation. This method does not contain any information about shape, image topology, or texture.

- **Multifeature composed signature**:

  In this approach, the signature of an image includes a composition of multiple features: color histogram, shape, image topology, and texture.

TEXT MINING:

Text databases (or document databases), which consist of large collections of documents from various sources, such as news articles, research papers, books, digital libraries, e-mail messages, and Web pages. Text databases are rapidly growing due to the increasing amount of information available in electronic form, such as electronic publications, various kinds of electronic documents, e-mail, and the World Wide Web .

IR(INFORMATION RETRIEVAL SYSTEM):

A typical information retrieval problem is to locate relevant documents in a document collection based on a user's query, which is often some keywords describing an information need, although it could also be an example relevant document. In such a search problem, a user takes the initiative to "pull" the relevant information out from the collection; this is most appropriate when a user has some ad hoc (i.e., short-term)information need, such as finding information to buy a used car. When a user has a long-term information need (e.g., a researcher's interests), a retrieval system may also take the initiative to "push" any newly arrived information item to a user if the item is judged as being relevant to the user's information need. Such an information access process is called information filtering, and the corresponding systems are often called filtering systems or recommender systems.

Basic Measures for Text Retrieval: Precision and Recall

"Suppose that a text retrieval system has just retrieved a number of documents for me based on my input in the form of a query. How can we assess how accurate or correct the system?

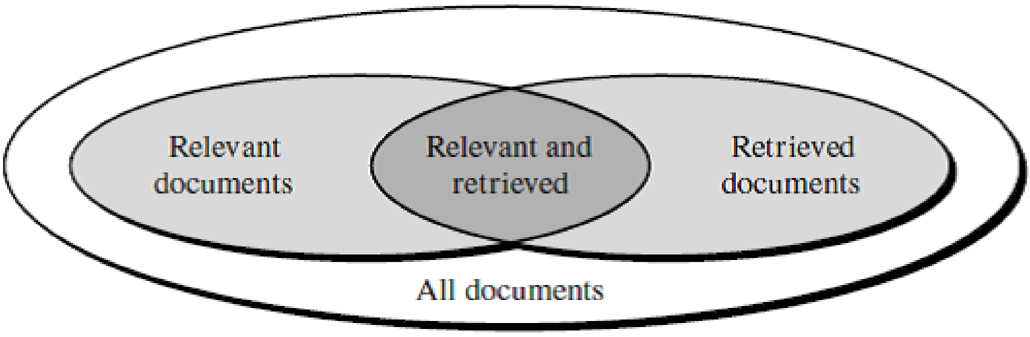
<u>Precision:</u> This is the percentage of retrieved documents that are in fact relevant to the query (i.e., "correct" responses). It is formally defined as

$$precision = \frac{|\{Relevant\} \cap \{Retrieved\}|}{|\{Retrieved\}|}.$$

<u>Recall:</u> This is the percentage of documents that are relevant to the query and were,

in fact, retrieved. It is formally defined as

$$recall = \frac{|\{Relevant\} \cap \{Retrieved\}|}{|\{Relevant\}|}.$$



HOW MINING THEWORLD WIDEWEB IS DONE:

The World Wide web serves as a huge, widely distributed, global information service center for news, advertisements, consumer information, financial management, education, government, e-commerce, and many other information services. The Web also contains a rich and dynamic collection of hyperlink information and Web page access and usage information, providing rich sources for data mining.

challenges for effective resource and knowledge discovery in web

- The Web seems to be too huge for effective data warehousing and data mining. The size of the Web is in the order of hundreds of terabytes and is still growing rapidly. Many organizations and societies place most of their public-accessible information on the Web. It is barely possible to set up a data warehouse to replicate, store, or integrate all of the data on the Web.
- The complexity of Web pages is far greater than that of any traditional text document
  collection. Web pages lack a unifying structure.
- The Web is a highly dynamic information source. Not only does the Web grow rapidly,
- but its information is also constantly updated.
- TheWeb serves a broad diversity of user communities. The Internet currently connects
- more than 100 million workstations, and its user community is still rapidly expanding.

These challenges have promoted research into efficient and effective discovery and use of resources on the Internet.

MINING THE WWW

Mining theWeb Page Layout Structure:

- The basic structure of a Web page is its DOM(Document Object Model) structure. The DOM structure of a Web page is a tree structure, where every HTML tag in the page corresponds to a node in the DOM tree. The Web page can be segmented by some predefined structural tags. Thus the DOM structure can be used to facilitate information extraction.
- Here, we introduce an algorithm called VIsion-based Page Segmentation (VIPS).
- VIPS aims to extract the semantic structure of a Web page based on its visual presentation.

Mining the Web's Link Structures to Identify Authoritative Web Pages:

- The Web consists not only of pages, but also of hyperlinks pointing from one page to another.
- These hyperlinks contain an enormous amount of latent human annotation that can help automatically infer the notion of authority. These properties of Web link structures have led researchers to consider another important category of Web pages called a hub. A hub is one or a set pages that provides collections of links to authorities.

DATA MINING APPLICATIONS:

Data Mining for Financial Data Analysis:

- **Design and construction of data warehouses for multidimensional data analysis and data mining:**

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. One may like to view the debt and revenue changes by month, by region, by sector, and by other factors, along with maximum, minimum, total, average, trend, and other statistical information.

- **Loan payment prediction and customer credit policy analysis**:

Loan payment prediction and customer credit analysis are critical to the business of a bank. Many factors can strongly or weakly influence loan payment performance and customer credit rating.

- **Classification and clustering of customers for targeted marketing**:

Classification and clustering methods can be used for customer group identification and targeted marketing.

For example: we can use classification to identify the most crucial factors that may influence a customer's decision regarding banking. Customers with similar behaviours regarding loan payments may be identified by multidimensional clustering techniques.

- **Detection of money laundering and other financial crimes**:

To detect money laundering and other financial crimes, it is important to integrate information from multiple databases (like bank transaction databases, and federal or state crime history databases), as long as they are potentially related to the study

Data Mining for the Retail Industry:

- **Design and construction of data warehouses based on the benefits of data mining:**

Because retail data cover a wide spectrum (including sales, customers, employees, goods transportation, consumption, and services), there can be many ways to design a data warehouse for this industry.

- **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

- **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

- **Customer retention—analysis of customer loyalty**:

With customer loyalty card information, one can register sequences of purchases of particular customers. Customer loyalty and purchase trends can be analyzed systematically

- **Product recommendation and cross-referencing of items**:

By mining associations from sales records, one may discover that a customer who buys a digital camera is likely to buy another set of items. Such information can be used to form product recommendations. Collaborative recommender systems use data mining techniques to make personalized product recommendations during live customer transactions,
based on the opinions of other customers.

.

Data Mining for the Telecommunication Industry:

- **Fraudulent pattern analysis and the identification of unusual patterns:**

Fraudulent activity costs the telecommunication industry millions of dollars per year.

Itis important to

- identify potentially fraudulent users and their atypical usage patterns;
- detect attempts to gain fraudulent entry to customer accounts; and
- discover unusual patterns that may need special attention, such as busy-hour frustrated call attempts, switch and route congestion patterns, and periodic calls from automatic dial-out equipment (like fax machines) that have been improperly programmed

Multidimensional association and sequential pattern analysis:

The discovery of association and sequential patterns in multidimensional analysis can be used to promote telecommunication services. For example, suppose you would like to find usage patterns for a set of communication services by customer group, by month, and by time of day.

Mobile telecommunication services: Mobile telecommunication, Web and information
services, and mobile computing are becoming increasingly integrated and common in our work and life.

The Social Impacts of Data Mining:

- Ubiquitous data mining is the ever presence of data mining in many aspects of our daily lives. It can influence how we shop, work, search for information, and use a computer, as well as our leisure time, health, and well-being. In invisible data mining, “smart” software, such as Web search engines, customer-adaptive Web services (e.g., using recommender algorithms), e-mail managers, and so on, incorporates data mining into its functional components, often unbeknownst to the user.
- From grocery stores that print personalized coupons on customer receipts to on-line stores that recommend additional items based on customer interests, data mining has innovatively influenced what we buy, the way we shop, as well as our experience while shopping.
- Data mining has shaped the on-line shopping experience. Many shoppers routinely turn to on-line stores to purchase books, music, movies, and toys
- Many companies increasingly use data mining for customer relationship management (CRM), which helps provide more customized, personal service addressing individual customer’s needs, in lieu of mass marketing
- While you are viewing the results of your Google query, various ads pop up relating

- to your query. Google’s strategy of tailoring advertising to match the user’s interests is
- successful—it has increased the clicks for the companies involved by four to five times.
  - Web-wide tracking is a technology that tracks a user across each site she visits. So,while
- Surfing the Web, information about every site you visit may be recorded,which can provide
- marketers with information reflecting your interests, lifestyle, and habits
  - Finally, data mining can contribute toward our health and well-being. Several pharmaceutical companies use data mining software to analyze data when developing drugs and to find associations between patients, drugs, and outcomes. It is also being used to detect beneficial side effects of drugs.

TRENDS IN DATA MINING:

Trends in data mining include further efforts toward the exploration of new application areas, improved scalable and interactive methods (including constraint-based mining), the integration of data mining with data warehousing and database systems, the standardization of data mining languages, visualization methods, and new methods for handling complex data types. Other trends include biological data mining, mining software bugs, Web mining, distributed and real-time mining, graph mining, social network analysis, multi relational and multi database data mining, data privacy protection, and data security.

- **Application Exploration:**

  The exploration of data mining for businesses continues to expand as e-commerce and e-marketing have become mainstream elements of the retail industry. Data mining is increasingly used for the exploration of applications in other areas, such as financial analysis, telecommunications, biomedicine, and science. Emerging application areas include data mining for counterterrorism (including and beyond intrusion detection) and mobile (wireless) data mining.

- **Scalable and interactive data mining methods:**

  Data analysis methods, data mining must be able to handle huge amounts of data efficiently and, if possible, interactively. Because the amount of data being collected continues to increase rapidly, scalable algorithms for individual and integrated data mining functions become essential.

- **Integration of data mining with database systems, data warehouse systems and web database systems:**

  Database systems, data warehouse systems, and the Web have become mainstream information processing systems. It is important to ensure that data mining serves as an essential data analysis component that can be smoothly integrated into such an information processing environment.

- **Standardization of data mining query language:**

A standard data mining language or other standardization efforts 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.**

data mining progress has been made in mining stream, time-series, sequence, graph, spatiotemporal, multimedia, and text data, there is still a huge gap between the needs for these applications and the available technology

- **Biological data mining:**

  Biological data mining can be considered under "application exploration" or "mining complex types of data," the unique combination of complexity, richness, size, and importance of biological data warrants special attention in data mining. Mining DNA and protein sequences, mining high dimensional microarray data, biological pathway and network analysis

- **Data mining and software engineering:**

As software programs become increasingly bulky in size, sophisticated in complexity, and tend to originate from the integration of multiple components developed by different software teams, it is an increasingly challenging task to ensure software robustness and reliability

- **Web mining:**

  The huge amount of information available on the Web and the increasingly important role that the Web plays in today's society, Web content mining, Weblog mining, and data mining services on the Internet will become one of the most important and flourishing subfields in data mining

- **Distributed data mining:**

  Traditional data mining methods, designed to work at a centralized location, do not work well in many of the distributed computing environments present today.

- **Real time data mining:**

  Many applications involving stream data (such as e-commerce, Web mining, stock analysis, intrusion detection, mobile data mining.

- **Multi database data mining:**

  Most realworld data and information are spread across multiple tables and databases. Multi relational data mining methods search for patterns involving multiple tables (relations) from a relational database.

- **Privacy protection and information security in data mining:**

  An abundance of recorded personal information available in electronic forms and on the Web, coupled with increasingly powerful data mining tools, poses a threat to our privacy and data security.