DATA MINING

Introduction

Data mining refers to extracting or ―mining‖ knowledge from large amounts of data.

Many other terms carry a similar or slightly different meaning to data mining, such as knowledge mining from data, knowledge extraction, data/pattern analysis, another popularly used term, Knowledge Discovery from Data, or KDD

Essential step in the Process of knowledge discovery. Knowledge discovery as a process is depicted in Figure consists of an iterative sequence of the following steps:

Data cleaning: to remove noise and inconsistent data

Data integration: where multiple data sources may be combined

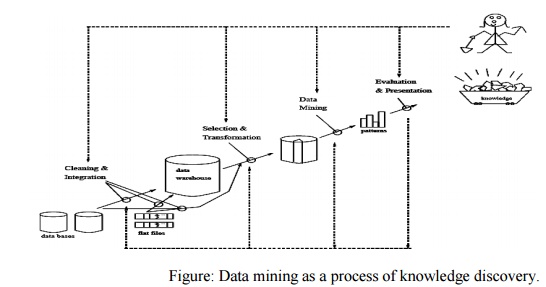
Data selection: where data relevant to the analysis task are retrieved from the database.

Data transformation: where data are transformed or consolidated into forms appropriate for mining by performing summary or aggregation operations, for instance

Data mining: an essential process where intelligent methods are applied in order to extract data patterns

Pattern evaluation to identify the truly interesting patterns representing knowledge based on some interestingness measures;

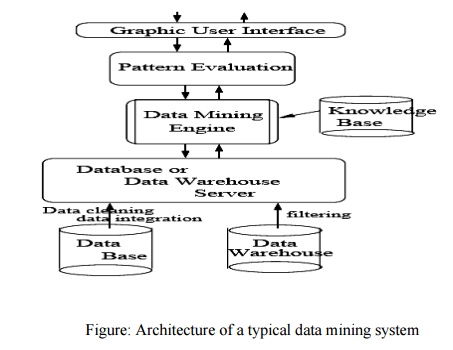
Knowledge presentation where visualization and knowledge representation techniques are used to present the mined knowledge to the user



The architecture of a typical data mining system may have the following major components Database, data warehouse, Worldwide Web, or other information repository: This is one or a set of databases, data warehouses, spreadsheets, or other kinds of information repositories. Data cleaning and data integration techniques may be performed on the data.

Database or data warehouse server: The database or data warehouse server is responsible for fetching the relevant data, based on the user’s data mining request.

Knowledge base: This is the domain knowledge that is used to guide the search or evaluate the interestingness of resulting patterns. Such knowledge can include concept hierarchies, used to organize attributes or attribute values into different levels of abstraction. Other examples of domain knowledge are additional interestingness constraints or thresholds, and metadata (e.g., describing data from multiple heterogeneous sources).



Data mining engine: This is essential to the data mining system and ideally consists of a set of functional modules for tasks such as characterization, association and correlation analysis, classification, prediction, cluster analysis, outlier analysis, and evolution analysis.

Pattern evaluation module: This component typically employs interestingness measures (and interacts with the data mining modules so as to *focus* the search toward interesting patterns. It may use interestingness thresholds to filter out discovered patterns. Alternatively, the pattern evaluation module may be integrated with the mining module, depending on the implementation of the data mining method used..

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

**DATA**

Collection of data objects and their attributes

An attribute is a property or characteristic of an object

Examples: eye color of a person, temperature, etc. Attribute is also known as variable, field, characteristic, or feature A collection of attributes describe an object Object is also known as record, point, case, sample, entity, or instance Attributes



Attribute values are numbers or symbols assigned to an attribute Distinction between attributes and attribute values Same attribute can be mapped to different attribute values

**Example**: height can be measured in feet or meters Different attributes can be mapped to thesame set of values

**Example**: Attribute values for ID and age are integers But properties of attribute values can bedifferent ID has no limit but age has a maximum and minimum value

**Types of Attributes**

There are different types of attributes

**Nominal**: Examples: ID numbers, eye color, zip codes

**Ordinal** Examples: rankings (e.g., taste of potato chips on a scale from 1-10), grades, height in{tall, medium, short}

**Interval Examples:** calendar dates, temperatures in Celsius or Fahrenheit. **Ratio Examples**: temperature in Kelvin, length, time, counts.

**Data Mining—On What Kind of Data? ( Types of Data )**

**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 nameEach 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 Warehouses:** A data warehouse is a repository of information collected from multiplesources, 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* (such as from the past 5–10 years) and are typically *summarized*.



A data warehouse is usually modeled by a multidimensional database structure, where each dimension corresponds to an attribute or a set of attributes in the schema, and each cell stores the value of some aggregate measure, such as *count* or *sales amount*



The actual physical structure of a data warehouse may be a relational data store or a multidimensional data cube. 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****?”*you may ask.



**A data warehouse** collects information about subjects that span an*entire organization*, andthus its scope is *enterprise-wide*.



**A data mart**, on the other hand, is a department subset of a data warehouse. It focuses onselected subjects, and thus its scope is *department-wide*. Data warehouse systems are well suited for on-line analytical processing, or OLAP. OLAP operations use background knowledge regarding the domain of the data being studied in order to allow the presentation of data at *different levels of abstraction*. Such operations accommodate different user viewpoints.



Examples of OLAP operations include drill-down and roll-up, which allow the user to view the data at differing degrees of summarization,



**Transactional Databases:** Transactional database consists of a file where each recordrepresents a transaction. A transaction typically includes a unique transaction identity number (*trans ID*) and a list of the items making up the transaction (such as items purchased in a store).



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 WorldWideWeb (a huge, widely distributed information repository made available by the Internet).

These applications require efficient data structures and scalable methods for handling complex object structures; variable-length records; semi structured or unstructured data; text, spatiotemporal, and multimedia data; and database schemas with complex structures and dynamic changes.

**Object-Relational Databases:**Object-relational databases are constructed based on anobject-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 entityrelationship

and relational models.

**A set of messages** that the object can use to communicate with other objects, or with the restof the database system.

**A set of methods**, where each method holds the code to implement a message. Uponreceiving 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**



**A 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 notionof 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). Examples include data collected from the stock exchange, inventory control, and the observation of natural phenomena (like temperature and wind).



**Spatial Databases and Spatiotemporal Databases**

**Spatial databases** 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. For 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?”* you may ask. Data miningmay 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** are databases that contain word descriptions for objects. These worddescriptions 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), whereas others are relatively well structured (such as library catalogue databases). Text databases with highly regular structures typically can be implemented using relational database systems.

*“What can data mining on text databases uncover?”* By mining text data, one may uncovergeneral and concise descriptions of the text documents, keyword or content associations, as well as the clustering behavior of text objects.

**Multimedia databases** store image, audio, and video data. They are used in applications suchas 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**

**A heterogeneous database** consists of a set of interconnected, autonomous componentdatabases. 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.

**A legacy database** is a group of*heterogeneous databases*that combines different kinds of datasystems, 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*.

Typical examples of data 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.

Mining data streams involves the efficient discovery of general patterns and dynamic changes within stream data.

**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. Users seeking information of interest traverse from one object via links to another. Such systems provide ample opportunities and challenges for data mining.

For example, understanding user access patterns will not only help improve system design (by providing efficient access between highly correlated objects), but also leads to better marketing decisions (e.g., by placing advertisements in frequently visited documents, or by providing better customer/user classification and behavior analysis). Capturing user access patterns in such distributed information environments is called Web usage mining (or Weblog mining).

**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. For example, in the *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 insummarized, 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 (3) both data characterization and discrimination.

**Data characterization** is a summarization of the general characteristics or features of a targetclass 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** is a comparison of the general features of target class data objects with thegeneral 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.

*“How are discrimination descriptions output?”*

Discrimination descriptions expressed in rule form are referred to as discriminate rules.

**Mining Frequent Patterns, Associations, and Correlations**

**Frequent patterns**, as the name suggests, are patterns that occur frequently in data. There aremany kinds of frequent patterns, including itemsets, subsequences, and substructures.

**A *frequent itemset*** typically refers to a set of items that frequently appear together in atransactional data set, such as Computer and Software. A frequently occurring subsequence, such as thepattern that customers tend to purchase first a PC, followed by a digital camera, and then a memory card, is a (*frequent*) *sequential pattern*.

**Example:** Association analysis. Suppose, as a marketing manager of*AllElectronics*, you wouldlike to determine which items are frequently purchased together within the same transactions. An 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. This association rule involves a single attribute or predicate (i.e., *buys*) that repeats. Association rules that contain a single predicate are referred to as single-dimensional association rules. Dropping the predicate notation, the above rule can be written simply as ―*compute software* [1%, 50%]‖.

**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. The derived model is based on the analysis of a set of training data (i.e., data objects whose class label is known).

*“How is the derived model presented?”* The derived model may be represented in variousforms, 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 attributevalue, 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-likeprocessing 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 neighbor 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 numericprediction and class label prediction,

**Cluster Analysis**

Classification and prediction analyze class-labeled 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. However, in some applications such as fraud detection, the rare events can be more interesting than the more regularly occurring ones. The analysis of outlier data is referred to as outlier mining.

**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, distinct features of such an analysis include time-series data analysis, Sequence or periodicity pattern matching, and similarity-based data analysis.

**Interestingness Of Patterns**

A data mining system has the potential to generate thousands or even millions of patterns, or rules. then *“are all of the patterns interesting?”* Typically not—only a small fraction of the patternspotentially generated would actually be of interest to any given user.

This raises some serious questions for data mining. You may wonder, *“****What makes a*** ***pattern interesting? Can a data mining system generate all of the interesting patterns? Can a data mining system generate only interesting patterns?”***

**To answer the first question**, a pattern is interesting if it is

*easily understood* by humans,

(2)*valid* on new or test data with some degree of *certainty*,

potentially *useful*, and

*novel*.

A pattern is also interesting if it validates a hypothesis that the user *sought to confirm*. An interesting pattern represents **knowledge.**

Several objective measures of pattern interestingness exist. These are based on the structure of discovered patterns and the statistics underlying them. An objective measure for association rules of the form *X Y* is rule support, representing the percentage of transactions from a transaction database that the given rule satisfies.

This is taken to be the probability *P*(XU*Y*),where *XUY* indicates that a transaction contains both *X* and *Y*, that is, the union of itemsets *X* and *Y*. Another objective measure for association rules is confidence, which assesses the degree of certainty of the detected association. This is taken to be the conditional probability *P*(Y *| X*), that is, the probability that a transaction containing *X* also contains *Y*. More formally, support and confidence are defined as

***support*(*X Y*) = *P*(*XUY*) *confidence*(*X Y*) = *P*(*Y | X*)**

In general, each interestingness measure is associated with a threshold, which may be controlled by the user. For example, rules that do not satisfy a confidence threshold of, say, 50% can be considered uninteresting. Rules below the threshold threshold likely reflect noise, exceptions, or minority cases and are probably of less value.

The second question—―***Can a data mining system generate all of the interesting*** ***patterns****?*‖—refers to the completeness of a data mining algorithm. It is often unrealistic andinefficient for data mining systems to generate all of the possible patterns. Instead, user-provided constraints and interestingness measures should be used to focus the search.

Finally,

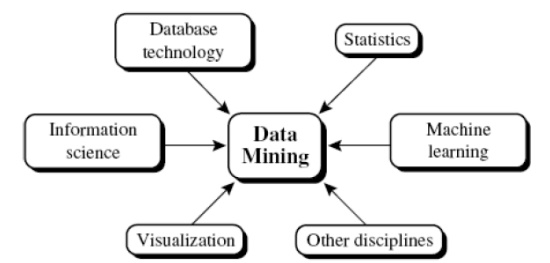
The third question—*“****Can a data mining system generate only interesting atterns****?”*—is an optimization problem in data mining. It is highly desirable for data mining systems to generate only interesting patterns. This would be much more efficient for users and data mining systems, because neither would have to search through the patterns generated in order to identify the truly interesting ones. Progress has been made in this direction; however, such optimization remains a challenging issue in data mining.

**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. Depending on the kinds of data to be mined or on the given data mining application, the data mining system may also integrate techniques from spatial data analysis, information retrieval, pattern recognition, image analysis, signal processing, computer graphics, Web technology, economics, business, bioinformatics, or psychology.

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 classifiedaccording 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. Data mining systems can therefore be classified accordingly.

**Classification according to the *kinds of knowledge* mined**: Data mining systems can becategorized 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. A comprehensive data mining system usually provides multiple and/or integrated data mining functionalities.

**Classification according to the *kinds of techniques* utilized**: Data mining systems can becategorized 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). A sophisticated data mining system will often adopt multiple data mining techniques or work out an effective, integrated technique that combines the merits of a few individual approaches.

**Classification according to the *applications adapted*:** Data mining systems can also becategorized 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. Therefore, a generic, all-purpose data mining system may not fit domain-specific mining tasks.

**Integration Of A Data Mining System With A Database Or Data Warehouse System**

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

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

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

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

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

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

**Data mining primitives.**

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

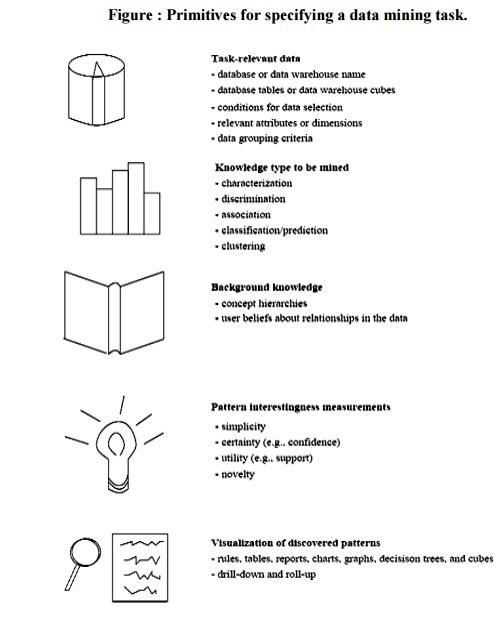
**Task-relevant data:** This is the database portion to be investigated. For example, suppose thatyou are a manager of All Electronics in charge of sales in the United States and Canada. In particular, you would like to study the buying trends of customers in Canada. Rather than mining on the entire 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, you may choose to mine associations between customer profiles and the items that these customers like to buy

**Background knowledge:** Users can specify background knowledge, or knowledge about thedomain 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 fromknowledge. 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 whichdiscovered patterns are to be displayed. Users can choose from different forms for knowledge presentation, such as rules, tables, charts, graphs, decision trees, and cubes.



**Major Issues In Data Mining**

The scope of this book addresses major issues in data mining regarding mining methodology, user interaction, performance, and diverse data types. These issues are introduced below:

**1. Mining methodology and user-interaction issues.** These reect the kinds of knowledge mined,the ability to mine knowledge at multiple granularities, the use of domain knowledge, ad-hoc mining, and knowledge visualization.

**Mining different kinds of knowledge in databases.**

Since different users can be interested in different kinds of knowledge, data mining should cover a wide spectrum of data analysis and knowledge discovery tasks, including data characterization, discrimination, association, classification, clustering, trend and deviation analysis, and similarity analysis. These tasks may use the same database in different ways and require the development of numerous data mining techniques.

**Interactive mining of knowledge at multiple levels of abstraction.**

Since it is difficult to know exactly what can be discovered within a database, the data mining process should be interactive. For databases containing a huge amount of data, appropriate sampling technique can first be applied to facilitate interactive data exploration. Interactive mining allows users to focus the search for patterns, providing and refining data mining requests based on returned results. Specifically, knowledge should be mined by drilling-down, rolling-up, and pivoting through the data space and knowledge space interactively, similar to what OLAP can do on data cubes. In this way, the user can interact with the data mining system to view data and discovered patterns at multiple granularities and from different angles.

**Incorporation of background knowledge.**

Background knowledge, or information regarding the domain under study, may be used to guide the discovery process and allow discovered patterns to be expressed in concise terms and at different levels of abstraction. Domain knowledge related to databases, such as integrity constraints and deduction rules, can help focus and speed up a data mining process, or judge the interestingness of discovered patterns.

**Data mining query languages and ad-hoc data mining.**

Relational query languages (such as SQL) allow users to pose ad-hoc queries for data retrieval. In a similar vein, high-level data mining query languages need to be developed to allow users to describe ad-hoc data mining tasks by facilitating the speci\_cation of the relevant sets of data for analysis, the domain knowledge, the kinds of knowledge to be mined, and the conditions and interestingness constraints to be enforced on the discovered patterns. Such a language should be integrated with a database or data warehouse query language, and optimized for e\_cient and exible data mining.

**Presentation and visualization of data mining results.**

Discovered knowledge should be expressed in high-level languages, visual representations, or other expressive forms so that the knowledge can be easily understood and directly usable by humans. This is especially crucial if the data mining system is to be interactive. This requires the system to adopt expressive knowledge representation techniques, such as trees, tables, rules, graphs, charts, crosstabs, matrices, or curves.

**Handling outlier or incomplete data.**

The data stored in a database may reect outliers | noise, exceptional cases, or incomplete data objects. These objects may confuse the analysis process, causing over\_tting of the data to the knowledge modelconstructed. As a result, the accuracy of the discovered patterns can be poor. Data cleaning methods and data analysis methods which can handle outliers are required. While most methods discard outlier data, such data may be of interest in itself such as in fraud detection for Finding unusual usage of tele-communication services or credit cards. This form of data analysis is known as outlier mining.

**Pattern evaluation: the interestingness problem.**

A data mining system can uncover thousands of patterns. Many of the patterns discovered may be uninteresting to the given user, representing common knowledge or lacking novelty. Several challenges remain regarding the development of techniques to assess the interestingness of discovered patterns, particularly with regard to subjective measures which estimate the value of patterns with respect to a given user class, based on user beliefs or expectations. The use of interestingness measures to guide the discovery process and reduce the search space is another active area of research.

**2. Performance issues.** These include efficiency, scalability, and parallelization of data miningalgorithms.

**Efficiency and scalability of data mining algorithms.**

To effectively extract information from a huge amount of data in databases, data mining algorithms must be efficient and scalable. That is, the running time of a data mining algorithm must be predictable and acceptable in large databases. Algorithms with exponential or even medium-order polynomial complexity will not be of practical use. From a database perspective on knowledge discovery, efficiency and scalability are key issues in the implementation of data mining systems. Many of the issues discussed above under mining methodology and user-interaction must also consider efficiency and scalability.

**Parallel, distributed, and incremental updating algorithms.**

The huge size of many databases, the wide distribution of data, and the computational complexity of some data mining methods are factors motivating the development of parallel and distributed data mining algorithms. Such algorithms divide the data into partitions, which are processed in parallel. The results from the partitions are then merged. Moreover, the high cost of some data mining processes promotes the need for incremental data mining algorithms which incorporate database updates without having to mine the entire data again \from scratch". Such algorithms perform knowledge modification incrementally to amend and strengthen what was previously discovered.

**3. Issues relating to the diversity of database types.**

**Handling of relational and complex types of data.**

There are many kinds of data stored in databases and data warehouses. Since relational databases and data warehouses are widely used, the development of efficient and effective data mining systems for such data is important. However, other databases may contain complex data objects, hypertext and multimedia data, spatial data, temporal data, or transaction data. It is unrealistic to expect one system to mine all kinds of data due to the diversity of data types and different goals of data mining. Specific data mining systems should be constructed for mining specific kinds of data. Therefore, one may expect to have different data mining systems for different kinds of data.

**Mining information from heterogeneous databases and global information systems.**

Local and wide-area computer networks (such as the Internet) connect many sources of data, forming huge, distributed, and heterogeneous databases. The discovery of knowledge from di\_erent sources of structured, semi-structured, or unstructured data with diverse data semantics poses great challenges to data mining. Data mining may help disclose high-level data regularities in multiple heterogeneous databases that are unlikely to be discovered by simple query systems and may improve information exchange and interoperability in heterogeneous databases.

**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 taskinvolves 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 manua**

**lly:** In general, this approach is time-consuming and may not be feasible given a large data set withmany missing values.

**3. Use a global constant to fill in the missing value:** Replace all missing attribute values by thesame 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 averageincome 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:** Forexample, 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 withinference-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 ‖neighborhood", or values around it. The sorted values are distributed into a number of 'buckets', or bins. Because binning methods consult the neighborhood of values, they perform local smoothing. Figure illustrates some binning techniques.

In this example, the data for price are first sorted and partitioned into equi-depth bins (of depth 3). In smoothing by bin means, each value in a bin is replaced by the mean value of the bin. For example, the mean of the values 4, 8, and 15 in Bin 1 is 9. Therefore, each original value in this bin is replaced by the value 9. Similarly, smoothing by bin medians can be employed, in which each bin value is replaced by the bin median. In smoothing by bin boundaries, the minimum and maximum values in a given bin are identified as the bin boundaries. Each bin value is then replaced by the closest boundary value.

(i).Sorted data for price (in dollars): 4, 8, 15, 21, 21, 24, 25, 28, 34 (ii).Partition into (equi-width) bins:

Bin 1: 4, 8, 15

Bin 2: 21, 21, 24

Bin 3: 25, 28, 34

(iii).Smoothing by bin means:

Bin 1: 9, 9, 9,

Bin 2: 22, 22, 22

Bin 3: 29, 29, 29

(iv).Smoothing by bin boundaries:

Bin 1: 4, 4, 15

Bin 2: 21, 21, 24

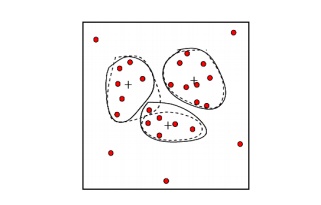
Bin 3: 25, 25, 34

**2. 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.**



**3. Combined computer and human inspection:** Outliers may be identified through a combinationof computer and human inspection. In one application, for example, an information-theoretic measure was used to help identify outlier patterns in a handwritten character database for classification. The measure's value reflected the ―surprise" content of the predicted character label with respect to the known label. Outlier patterns may be informative or ―garbage". Patterns whose surprise content is above a threshold are output to a list. A human can then sort through the patterns in the list to identify the actual garbage ones

**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. Knowledge engineering tools may also be used to detect the violation of known data constraints. For example, known functional dependencies between attributes can be used to find values contradicting the functional constraints.

**2. Data Transformation.**

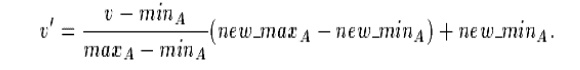
In data transformation, the data are transformed or consolidated into forms appropriate for mining. Data transformation can involve the following:

**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, and normalization by decimal scaling.**

**(i).Min-max normalization** performs a linear transformation on the original data. Suppose thatminA 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



**(ii).z-score normalization (or zero-mean normalization),** the values for an attribute A arenormalized 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. This method of normalization is useful when the actual minimum and maximum of attribute A are unknown, or when there are outliers which dominate the min-max normalization.

http://img.brainkart.com/extra/G91i5GF.jpg

(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

http://img.brainkart.com/extra/6WbfNTu.jpg

**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 ‖neighborhood", or values around it. The sorted values are distributed into a number of 'buckets', or bins. Because binning methods consult the neighborhood of values, they perform local smoothing. Figure illustrates some binning techniques.

In this example, the data for price are first sorted and partitioned into equi-depth bins (of depth 3). In smoothing by bin means, each value in a bin is replaced by the mean value of the bin. For example, the mean of the values 4, 8, and 15 in Bin 1 is 9. Therefore, each original value in this bin is replaced by the value 9. Similarly, smoothing by bin medians can be employed, in which each bin value is replaced by the bin median. In smoothing by bin boundaries, the minimum and maximum values in a given bin are identified as the bin boundaries. Each bin value is then replaced by the closest boundary value.

(i).Sorted data for price (in dollars): 4, 8, 15, 21, 21, 24, 25, 28, 34 (ii).Partition into (equi-width) bins:

Bin 1: 4, 8, 15

Bin 2: 21, 21, 24

Bin 3: 25, 28, 34

(iii).Smoothing by bin means:

Bin 1: 9, 9, 9,

Bin 2: 22, 22, 22

Bin 3: 29, 29, 29

(iv).Smoothing by bin boundaries:

Bin 1: 4, 4, 15

Bin 2: 21, 21, 24

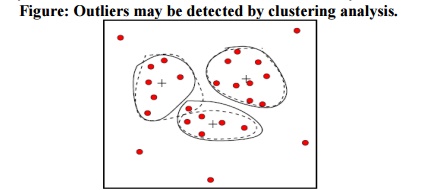
Bin 3: 25, 25, 34

**(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.**



**Aggregation,** where summary or aggregation operations are applied to the data. For example, thedaily 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 levelconcepts 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. That is, mining on the reduced data set should be more efficient yet produce the same (or almost the same) analytical results.

Strategies for data reduction include the following.

**Data cube aggregation,** where aggregation operations are applied to the data in the constructionof a data cube.

**Dimension reduction**, where irrelevant, weakly relevant or redundant attributes or dimensionsmay 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 datarepresentations 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 arereplaced 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

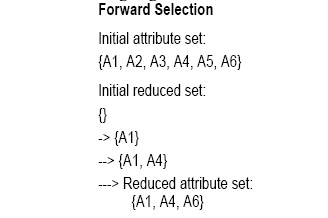
**Feature selection** (i.e., attribute subset selection):

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

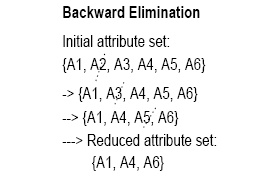
reduce # of patterns in the patterns, easier to understand

**Heuristic methods**:

**Step-wise forward selection:** The procedure starts with an empty set of attributes. The best ofthe 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.

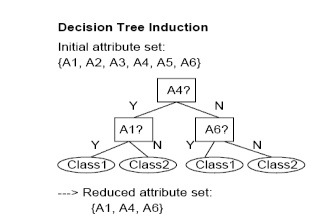


**Step-wise backward elimination:** The procedure starts with the full set of attributes. At eachstep, it removes the worst attribute remaining in the set.



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

**Decision tree induction:** Decision tree algorithms, such as ID3 and C4.5, were originallyintended 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. If, instead, 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 modeled to fit a straight line. For example, a random variable, Y (called a response variable), can be modeled 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 tobe modeled as a linear function of a multidimensional feature vector.

**Log-linear models** approximate discrete multidimensional probability distributions. Themethod 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.

**DATA MINING**

**Define data mining**

Data mining is a process of extracting or mining knowledge from huge amount of data.

**Define pattern evaluation**

Pattern evaluation is used to identify the truly interesting patterns representing knowledge based on some interesting measures.

**Define knowledge representation**

Knowledge representation techniques are used to present the mined knowledge to the user.

**List the five primitives for specification of a data mining task.**

task-relevant data

kind of knowledge to be mined

background knowledge

interestingness measures

knowledge presentation and visualization techniques to be used for displaying the discovered patterns

**What is Visualization?**

Visualization is for depiction of data and to gain intuition about data being observed. It assists the analysts in selecting display formats, viewer perspectives and data representation schema

Mention some of the application areas of data mining

DNA analysis

Market analysis

Financial data analysis

Banking industry

Retail Industry

Health care analysis.

Telecommunication industry

**Define data cleaning**

Data cleaning means removing the inconsistent data or noise and collecting necessary information

**Define Data integration.**

Integration of multiple databases, data cubes, or files

**Why we need Data transformation**

Smoothing: remove noise from data min-max normalization

Aggregation: summarization, data z-score normalization cube construction normalization by decimal scaling

Generalization: concept hierarchy Attribute/feature construction climbing

Normalization: scaled to fall within a        New attributes constructed from the   given ones small, specified range

**Define Data reduction.**

Data reduction Obtains reduced representation in volume but produces the same or similar analytical results.

**What is meant by Data discretization**

It can be defined as Part of data reduction but with particular importance, especially for numerical data

**What is the discretization processes involved in data preprocessing?**

It reduces the number of values for a given continuous attribute by dividing the range of the attribute into intervals. Interval labels can then be used to replace actual data values.

**Define Concept hierarchy.**

It reduce the data by collecting and replacing low level concepts (such as numeric values for the attribute age) by higher level concepts (such as young, middle-aged, or senior).

**Why we need data preprocessing.** Data in the real world is dirty

incomplete: lacking attribute values, lacking certain attributes of interest, or containing only aggregate data

noisy: containing errors or outliers

inconsistent: containing discrepancies in codes or names

**Give some data mining tools.**

DBMiner

GeoMiner

Multimedia miner

WeblogMiner

**Describe the use of DBMiner.**

Used to perform data mining functions, including characterization, association, classification, prediction and clustering.

**Applications of DBMiner.**

The DBMiner system can be used as a general-purpose online analytical mining system for both OLAP and data mining in relational database and datawarehouses.Used in medium to large relational databases with fast response time.

**What are the types of knowledge to be mined?**

Characterization

Discrimination

Association

Classification

prediction



Clustering

Outlier analysis

Other data mining tasks

**Define Relational databases.**

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.

**Define Transactional Databases.**

A 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.

21. Define Spatial Databases.

Spatial databases contain spatial-related information. Such databases include geographic (map) databases, VLSI chip 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.

**22. What is Temporal Database?**

Temporal database store time related data .It usually stores relational data that include time related attributes. These attributes may involve several time stamps, each having different semantics.

**23. What are Time-Series databases?**

A Time-Series database stores sequences of values that change with time,such as data collected regarding the stock exchange.

**24. What is Legacy database?**

A Legacy database is a group of heterogeneous databases that combines different kinds of data systems, such as relational or object-oriented databases, hierarchical databases, network databases, spread sheets, multimedia databases or file systems.

**25 What are the steps in the data mining process?**

Data cleaning



Data integration



Data selection



Data transformation



Data mining



Pattern evaluation



Knowledge representation



**What is Characterization?**

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

**27. What is 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.

**28. What is Classification?**

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. The derived model is based on the analysis of a set of training data

**What are the classification of data mining system**

Classification according to the *kinds of databases* mined

Classification according to the *kinds of knowledge* mined

Classification according to the *kinds of techniques* utilized

Classification according to the *applications adapted*:

**30 what are the scheme of integrating data mining system with a data warehouse**

No coupling

Loose coupling:



Semi tight coupling

Tight coupling

**What are the issues of data mining**

Mining methodology and user interaction issues

Mining different kinds of knowledge in databases:

Interactive mining of knowledge at multiple levels of abstraction

Incorporation of background knowledge

Data mining query languages and ad hoc data mining

Presentation and visualization of data mining results

Handling noisy or incomplete data

Pattern evaluation

Performance issues:

Efficiency and scalability of data mining algorithms

Parallel, distributed, and incremental mining algorithms

Issues relating to the diversity of database types: Handling of relational and complex types of data

Mining information from heterogeneous databases and global information systems

**What is data pre processing.**

The real world data’s are normally noise data so before organizing the data warehouse we need to Preprocess the data

**What is preprocessing technique?**

Data cleaning

Data integration



Data transformation

Data reduction



Define data cleaning

Data cleaning means removing the inconsistent data or noise and collecting necessary information

**Define Data integration.**

Integration of multiple databases, data cubes, or files

**Why we need Data transformation**

Smoothing: remove noise from data

Aggregation: summarization, data cube construction

Generalization: concept hierarchy climbing

Normalization: scaled to fall within a small, specified range

Min-max normalization

Z-score normalization

Normalization by decimal scaling

Attribute/feature construction: New attributes constructed from the given ones

**Define Data reduction.**

Data reduction Obtains reduced representation in volume but produces the same or similar analytical results.

**What is meant by Data discretization**

It can be defined as Part of data reduction but with particular importance, especially for numerical data

**What is the discretization processes involved in data preprocessing?**

It reduces the number of values for a given continuous attribute by dividing the range of the attribute into intervals. Interval labels can then be used to replace actual data values.

**Define Concept hierarchy.**

It reduce the data by collecting and replacing low level concepts (such as numeric values for the attribute age) by higher level concepts (such as young, middle-aged, or senior).

**Why we need data preprocessing. Data in the real world is dirty**

Incomplete: lacking attribute values, lacking certain attributes of interest, or containing only aggregate data

Noisy: containing errors or outliers

Inconsistent: containing discrepancies in codes or names

Equi-width: In an equi-width histogram, the width of each bucket range is constant (such as the width of $10 for the buckets in Figure 3.8).

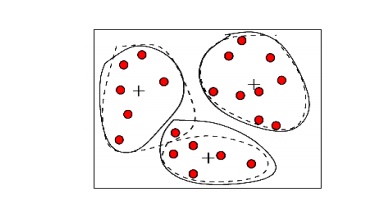
Equi-depth (or equi-height): In an equi-depth histogram, the buckets are created so that, roughly, the frequency of each bucket is constant (that is, each bucket contains roughly the same number of contiguous data samples).

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 drawingn 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 ofm 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 sampleof 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.

**Major Issues In Data Mining**

The scope of this book addresses major issues in data mining regarding mining methodology, user interaction, performance, and diverse data types. These issues are introduced below:

**1. Mining methodology and user-interaction issues.** These reect the kinds of knowledge mined,the ability to mine knowledge at multiple granularities, the use of domain knowledge, ad-hoc mining, and knowledge visualization.

**Mining different kinds of knowledge in databases.**

Since different users can be interested in different kinds of knowledge, data mining should cover a wide spectrum of data analysis and knowledge discovery tasks, including data characterization, discrimination, association, classification, clustering, trend and deviation analysis, and similarity analysis. These tasks may use the same database in different ways and require the development of numerous data mining techniques.

**Interactive mining of knowledge at multiple levels of abstraction.**

Since it is difficult to know exactly what can be discovered within a database, the data mining process should be interactive. For databases containing a huge amount of data, appropriate sampling technique can first be applied to facilitate interactive data exploration. Interactive mining allows users to focus the search for patterns, providing and refining data mining requests based on returned results. Specifically, knowledge should be mined by drilling-down, rolling-up, and pivoting through the data space and knowledge space interactively, similar to what OLAP can do on data cubes. In this way, the user can interact with the data mining system to view data and discovered patterns at multiple granularities and from different angles.

**Incorporation of background knowledge.**

Background knowledge, or information regarding the domain under study, may be used to guide the discovery process and allow discovered patterns to be expressed in concise terms and at different levels of abstraction. Domain knowledge related to databases, such as integrity constraints and deduction rules, can help focus and speed up a data mining process, or judge the interestingness of discovered patterns.

**Data mining query languages and ad-hoc data mining.**

Relational query languages (such as SQL) allow users to pose ad-hoc queries for data retrieval. In a similar vein, high-level data mining query languages need to be developed to allow users to describe ad-hoc data mining tasks by facilitating the speci\_cation of the relevant sets of data for analysis, the domain knowledge, the kinds of knowledge to be mined, and the conditions and interestingness constraints to be enforced on the discovered patterns. Such a language should be integrated with a database or data warehouse query language, and optimized for e\_cient and exible data mining.

**Presentation and visualization of data mining results.**

Discovered knowledge should be expressed in high-level languages, visual representations, or other expressive forms so that the knowledge can be easily understood and directly usable by humans. This is especially crucial if the data mining system is to be interactive. This requires the system to adopt expressive knowledge representation techniques, such as trees, tables, rules, graphs, charts, crosstabs, matrices, or curves.

**Handling outlier or incomplete data.**

The data stored in a database may reect outliers | noise, exceptional cases, or incomplete data objects. These objects may confuse the analysis process, causing over\_tting of the data to the knowledge modelconstructed. As a result, the accuracy of the discovered patterns can be poor. Data cleaning methods and data analysis methods which can handle outliers are required. While most methods discard outlier data, such data may be of interest in itself such as in fraud detection for Finding unusual usage of tele-communication services or credit cards. This form of data analysis is known as outlier mining.

**Pattern evaluation: the interestingness problem.**

A data mining system can uncover thousands of patterns. Many of the patterns discovered may be uninteresting to the given user, representing common knowledge or lacking novelty. Several challenges remain regarding the development of techniques to assess the interestingness of discovered patterns, particularly with regard to subjective measures which estimate the value of patterns with respect to a given user class, based on user beliefs or expectations. The use of interestingness measures to guide the discovery process and reduce the search space is another active area of research.

**2. Performance issues.** These include efficiency, scalability, and parallelization of data miningalgorithms.

**Efficiency and scalability of data mining algorithms.**

To effectively extract information from a huge amount of data in databases, data mining algorithms must be efficient and scalable. That is, the running time of a data mining algorithm must be predictable and acceptable in large databases. Algorithms with exponential or even medium-order polynomial complexity will not be of practical use. From a database perspective on knowledge discovery, efficiency and scalability are key issues in the implementation of data mining systems. Many of the issues discussed above under mining methodology and user-interaction must also consider efficiency and scalability.

**Parallel, distributed, and incremental updating algorithms.**

The huge size of many databases, the wide distribution of data, and the computational complexity of some data mining methods are factors motivating the development of parallel and distributed data mining algorithms. Such algorithms divide the data into partitions, which are processed in parallel. The results from the partitions are then merged. Moreover, the high cost of some data mining processes promotes the need for incremental data mining algorithms which incorporate database updates without having to mine the entire data again \from scratch". Such algorithms perform knowledge modification incrementally to amend and strengthen what was previously discovered.

**3. Issues relating to the diversity of database types.**

**Handling of relational and complex types of data.**

There are many kinds of data stored in databases and data warehouses. Since relational databases and data warehouses are widely used, the development of efficient and effective data mining systems for such data is important. However, other databases may contain complex data objects, hypertext and multimedia data, spatial data, temporal data, or transaction data. It is unrealistic to expect one system to mine all kinds of data due to the diversity of data types and different goals of data mining. Specific data mining systems should be constructed for mining specific kinds of data. Therefore, one may expect to have different data mining systems for different kinds of data.

**Mining information from heterogeneous databases and global information systems.**

Local and wide-area computer networks (such as the Internet) connect many sources of data, forming huge, distributed, and heterogeneous databases. The discovery of knowledge from deferent sources of structured, semi-structured, or unstructured data with diverse data semantics poses great challenges to data mining. Data mining may help disclose high-level data regularities in multiple heterogeneous databases that are unlikely to be discovered by simple query systems and may improve information exchange and interoperability in heterogeneous databases.

**Frequent Itemsets, Closed Itemsets, and Association Rules**

 A set of items is referred to as an **itemset.**



 An itemset that contains *k* items is a ***k*-itemset**.



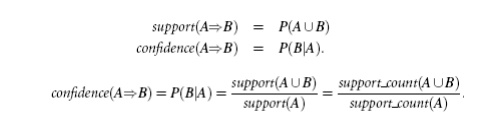
 The set {*computer, antivirus software}* is a **2-itemset**.



 The occurrence frequency of an itemset is the number of transactions that contain the itemset.



        This is also known, simply, as the **frequency, support count,** or **count** of the itemset.



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

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

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

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

**Association Mining**

**Association rule mining**: Finding frequent patterns, associations, correlations, or causalstructures among sets of items or objects in transaction databases, relational databases, and other information repositories.



**Applications:** Basket data analysis, cross-marketing, catalog design, loss-leader analysis,clustering, classification, etc.

Examples.



Rule form: ―Body ® Head [support, confidence]‖. buys(x, ―diapers‖) ® buys(x, ―beers‖) [0.5%, 60%]

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

**Association Rule: Basic Concepts**

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



Find: all rules that correlate the presence of one set of items with that of another set of items E.g., *98% of people who purchase tires and auto accessories also get automotive services done*

Applications

*Maintenance Agreement* (What the store should do to boost Maintenance Agreement sales) *Home Electronics  \** (What other products should the store stocks up?) Attached mailing indirect marketing Detecting ―ping-pong‖ing of patients, faulty ―collisions‖

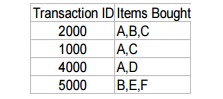
***Rule Measures: Support and Confidence***

Find all the rules *X & Y* ** *Z* with minimum confidence and support Support, *s*, probability that a transaction contains {X  Y  Z}

Confidence, *c,* conditional probability that a transaction having {X  Y} also contains *Z* Let minimum support 50%, and minimum confidence 50%, we have

A  C  (50%, 66.6%)

C  A (50%, 100%)



**Association Rule Mining: A Road Map**

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

buys(x, ―SQLServer‖) ^ buys(x, ―DMBook‖) ® buys(x, ―DBMiner‖) [0.2%, 60%] o age(x, ―30..39‖) ^ income(x, ―42..48K‖) ® buys(x, ―PC‖) [1%, 75%]

Single dimension vs. multiple dimensional associations (see ex. Above)

 Single level vs. multiple-level analysis

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

 Various extensions

Correlation, causality analysis

Association does not necessarily imply correlation or causality o Maxpatterns and closed itemsets

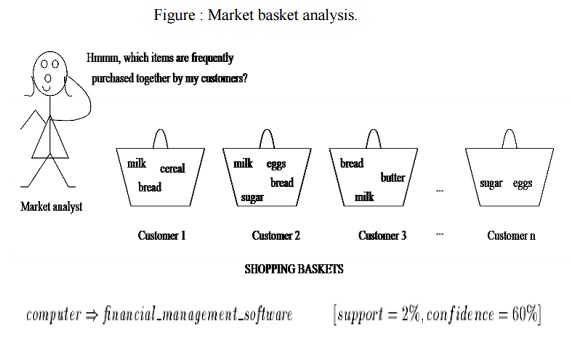
Constraints enforced

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

**Market – Basket analysis**

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

Figure : Market basket analysis.



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

**Mining Methods**

**The method that mines the complete set of frequent itemsets with candidate generation. Apriori property & The Apriori Algorithm.**

**Apriori property**

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

o   An item set I does not satisfy the minimum support threshold, min-sup, then I is not frequent, i.e., support(I) < min-sup

o   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.

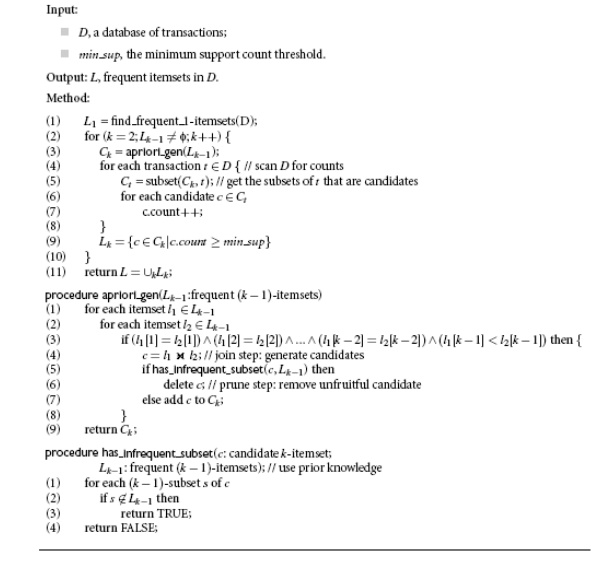


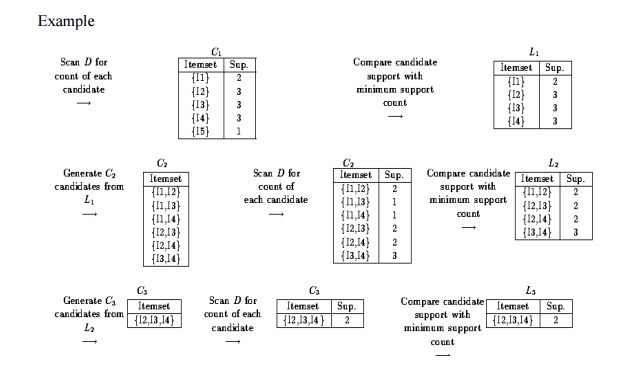
This property is monotonic in failing the test.

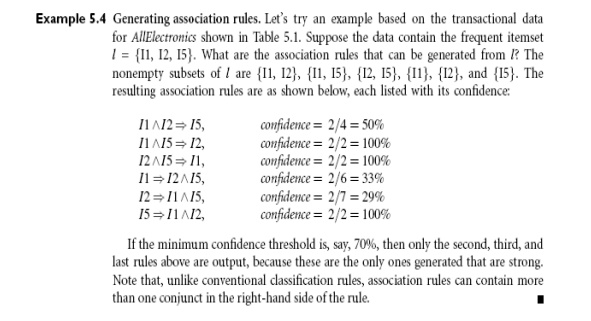
**The Apriori Algorithm**

Join Step: Ck is generated by joining Lk-1with itself

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







**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

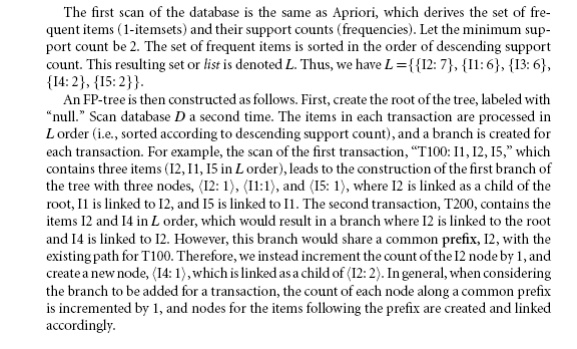
avoid costly database scans

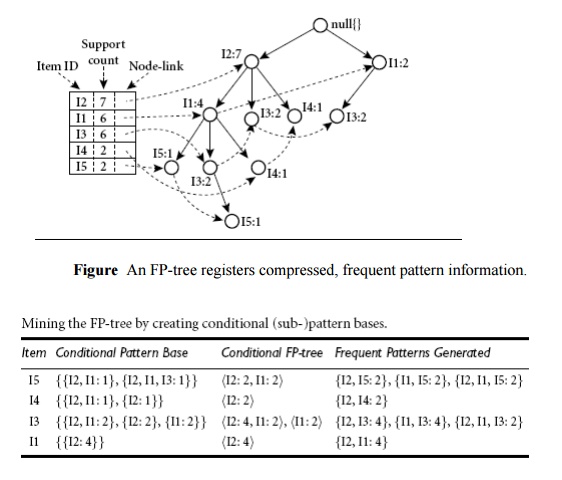
Develop an efficient, FP-tree-based frequent pattern mining method

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-examinethe mining of transaction database, *D*, of Table 5.1 in Example 5.3 using the frequent pattern growth approach.





**Benefits of the FP-tree Structure**

Completeness:

o   never breaks a long pattern of any transaction

o   preserves complete information for frequent pattern mining

Compactness

o   reduce irrelevant information—infrequent items are gone

o   frequency descending ordering: more frequent items are more likely to be shared

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

o   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

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

**Principles of Frequent Pattern Growth**

Pattern growth property

Let  be a frequent itemset in DB, B be 's conditional pattern base, and  be an itemset in B. Then  is a frequent itemset in DB iff  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

Reasoning

No candidate generation, no candidate test

Use compact data structure

Eliminate repeated database scan

Basic operation is counting and FP-tree building

**Mining Various Kinds of Association Rules**

**1. Mining Multilevel Association Rules**

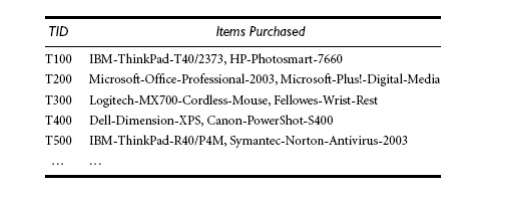
For many applications, it is difficult to find strong associations among data items at low or primitive levels of abstraction due to the sparsity of data at those levels. Strong associations discovered at high levels of abstraction may represent commonsense knowledge.

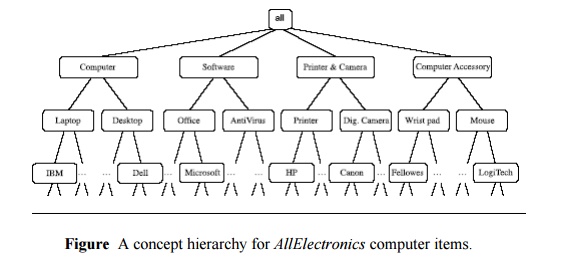
. Therefore, data mining systems should provide capabilities for mining association rules at multiple levels of abstraction, with sufficient flexibility for easy traversal among different abstraction spaces.

Let’s examine the following example.

Mining multilevel association rules. Suppose we are given the task-relevant set of transactional data in Table for sales in an *AllElectronics* store, showing the items purchased for each transaction.

The concept hierarchy for the items is shown in Figure . A concept hierarchy defines a sequence of mappings from a set of low-level concepts to higher level, more general concepts. Data can be generalized by replacing low-level concepts within the data by their higher-level concepts, or *ancestors*, from a concept hierarchy.



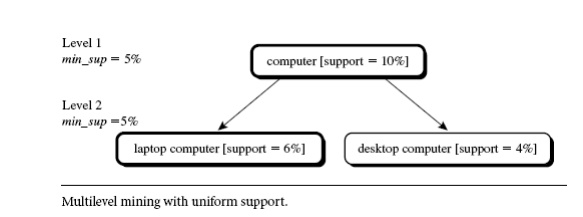


Association rules generated from mining data at multiple levels of abstraction are called multiple-level or multilevel association rules. Multilevel association rules can be mined efficiently using concept hierarchies under a support-confidence framework. In general, a top-down strategy is employed, For each level, any algorithm for discovering frequent itemsets may be used, such as Apriori or its variations.

**Using uniform minimum support for all levels (referred to as uniform support):** Thesame minimum support threshold is used when mining at each level of abstraction. For example, in Figure 5.11, a minimum support threshold of 5% is used throughout (e.g., for mining from

*“computer”* down to *“laptop computer”*). Both *“computer”* and *“laptop computer”* are found tobe frequent, while *“desktop computer”* is not.

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



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



**Using item or group-based minimum support (referred to as group-based support):**



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

**2. Mining Multidimensional Association Rules from Relational Databases and Data Warehouses**

We have studied association rules that imply a single predicate, that is, the predicate *buys*. For instance, in mining our *AllElectronics* database, we may discover the Boolean association rule

http://img.brainkart.com/extra/erLLHGe.jpg

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

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

http://img.brainkart.com/extra/lYqB8xt.jpg

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

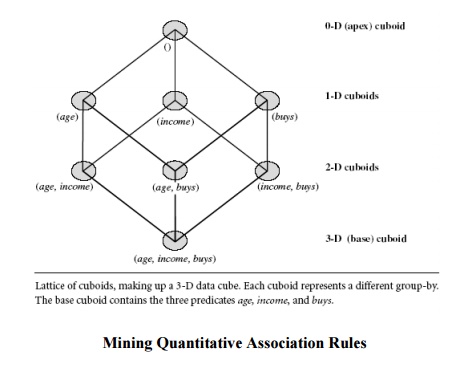
http://img.brainkart.com/extra/P3dIj1X.jpg

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

**Mining Multidimensional Association Rules Using Static Discretization of Quantitative**

**Attributes**

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



**Mining Quantitative Association Rules**

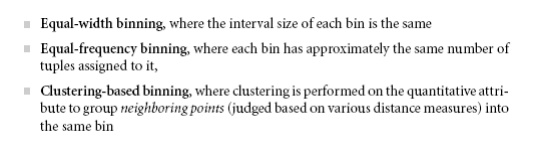
Quantitative association rules are multidimensional association rules in which the numeric attributes are *dynamically* discretized during the mining process so as to satisfy some mining criteria, such as maximizing the confidence or compactness of the rules mined. In this section, we focus specifically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

http://img.brainkart.com/extra/xfmh5X0.jpg

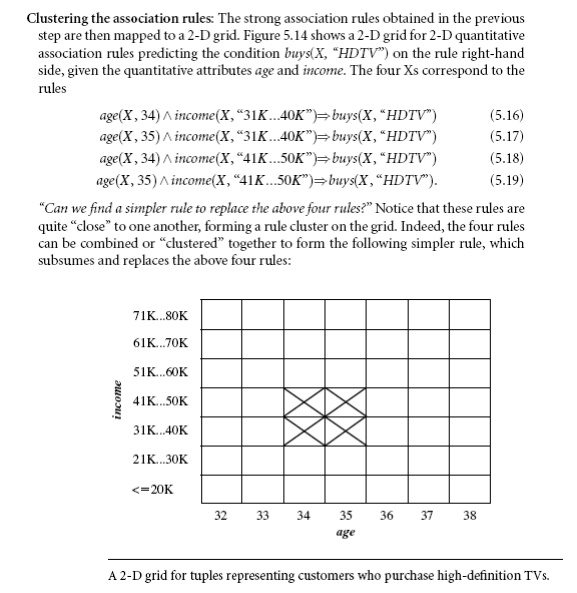
where *Aquan*1 and *Aquan*2 are tests on quantitative attribute intervals (where the intervals are dynamically determined), and *Acat* tests a categorical attribute from the task-relevant data. Such rules have been referred to as two-dimensional quantitative association rules, because they contain two quantitative dimensions. For instance, suppose you are curious about the association relationship between pairs of quantitative attributes, like customer age and income, and the type of television (such as *high-definition TV,* i.e., *HDTV*) that customers like to buy. An example of such a 2-D quantitative association rule is

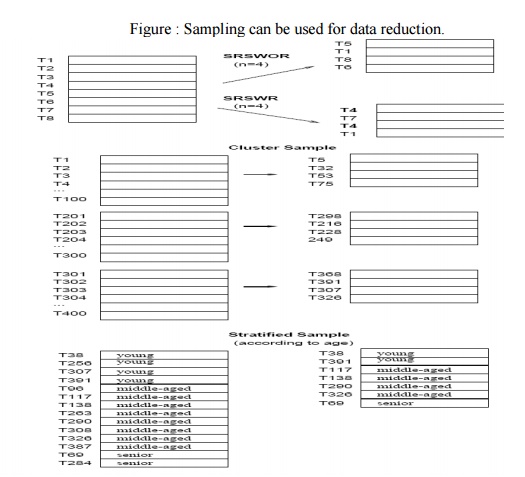
http://img.brainkart.com/extra/PogC27h.jpg

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



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





**Association Mining to Correlation Analysis**

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

**1)Strong Rules Are Not Necessarily Interesting: An Example**

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

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

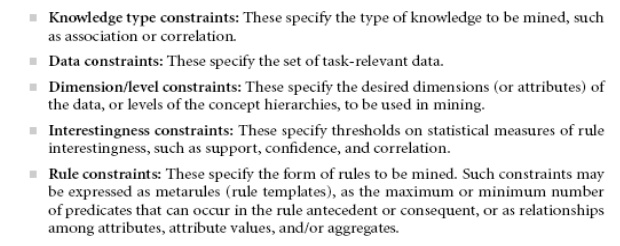
http://img.brainkart.com/extra/Frb6piw.jpg

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

**Constraint-Based Association Mining**

A data mining process may uncover thousands of rules from a given set of data, most of which end up being unrelated or uninteresting to the users. Often, users have a good sense of which

―direction‖ of mining may lead to interesting patterns and the ―form‖ of the patterns or rules they would like to find. Thus, a good heuristic is to have the users specify such intuition or expectations as *constraints* to confine the search space. This strategy is known as constraint-based mining. The constraints can include the following:



**1. Metarule-Guided Mining of Association Rules**

*“How are metarules useful?”* Metarules allow users to specify the syntactic form of rules thatthey are interested in mining. The rule forms can be used as constraints to help improve the efficiency of the mining process. Metarules may be based on the analyst’s experience, expectations, or intuition regarding the data or may be automatically generated based on the database schema.

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

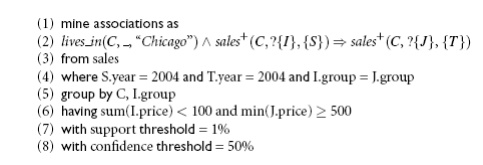
http://img.brainkart.com/extra/vH6IwTv.jpg

where *P*1 and *P*2 are predicate variables that are instantiated to attributes from the given database during the mining process, *X* is a variable representing a customer, and *Y* and *W* take on values of the attributes assigned to *P*1 and *P*2, respectively. Typically, a user will specify a list of attributes to be considered for instantiation with *P*1 and *P*2. Otherwise, a default set may be used.

**2. Constraint Pushing: Mining Guided by Rule Constraints**

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

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



**Classification and Prediction**

**Classification:**



o   predicts categorical class labels



o   classifies data (constructs a model) based on the training set and the values (class labels) in a classifying attribute and uses it in classifying new data

**Prediction**

models continuous-valued functions, i.e., predicts unknown or missing values

**Typical applications**

o   Credit approval

o   Target marketing

o   Medical diagnosis

o   Fraud detection

**Classification: Basic Concepts**

Supervised learning (classification)

o   Supervision: The training data (observations, measurements, etc.) are accompanied by **labels** indicating the class of the observations a New data is classified based on the training set

Unsupervised learning (clustering)

o   The class labels of training data is unknown

o   Given a set of measurements, observations, etc. with the aim of establishing the existence of classes or clusters in the data

**Classification vs. Numeric Prediction**

Classification

o   predicts categorical class labels (discrete or nominal)

o   classifies data (constructs a model) based on the training set and the values (class labels) in a classifying attribute and uses it in classifying new data

Numeric Prediction

o   models continuous-valued functions, i.e., predicts unknown or missing values

Typical applications

o   Credit/loan approval:

o   Medical diagnosis: if a tumor is cancerous or benign

o   o Fraud detection: if a transaction is fraudulent

o   Web page categorization: which category it is

**Classification—A Two-Step Process**

Model construction: describing a set of predetermined classes

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

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

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

Model usage: for classifying future or unknown objects

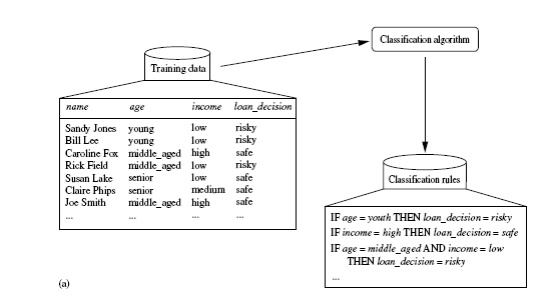
Estimate accuracy of the model

o   The known label of test sample is compared with the classified result from the model

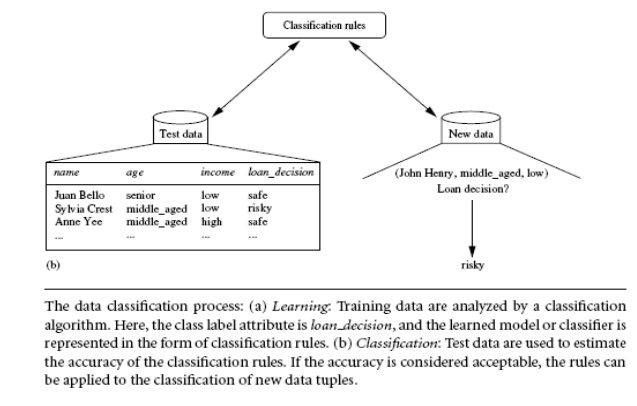
o   Accuracy rate is the percentage of test set samples that are correctly classified by the model

o   Test set is independent of training set, otherwise over-fitting will occur

**Process (1): Model Construction**



**Process (2): Using the Model in Prediction**



**Issues Regarding Classification and Prediction**

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



**Relevance analysis**: Many of the attributes in the data may be*redundant*. Correlationanalysis can be used to identify whether any two given attributes are statistically related.



**Data transformation and reduction:** The data may be transformed by normalization,particularly when neural networks or methods involving distance measurements are used in the learning step. Normalization involves scaling all values for a given attribute so that they fall within a small specified range, such as -1.0 to 1.0, or 0.0 to 1.0. In methods that use distance

**Comparing Classification and Prediction Methods**

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

**Accuracy**

**Speed**

**Robustness**

**Scalability**

**Interpretability**