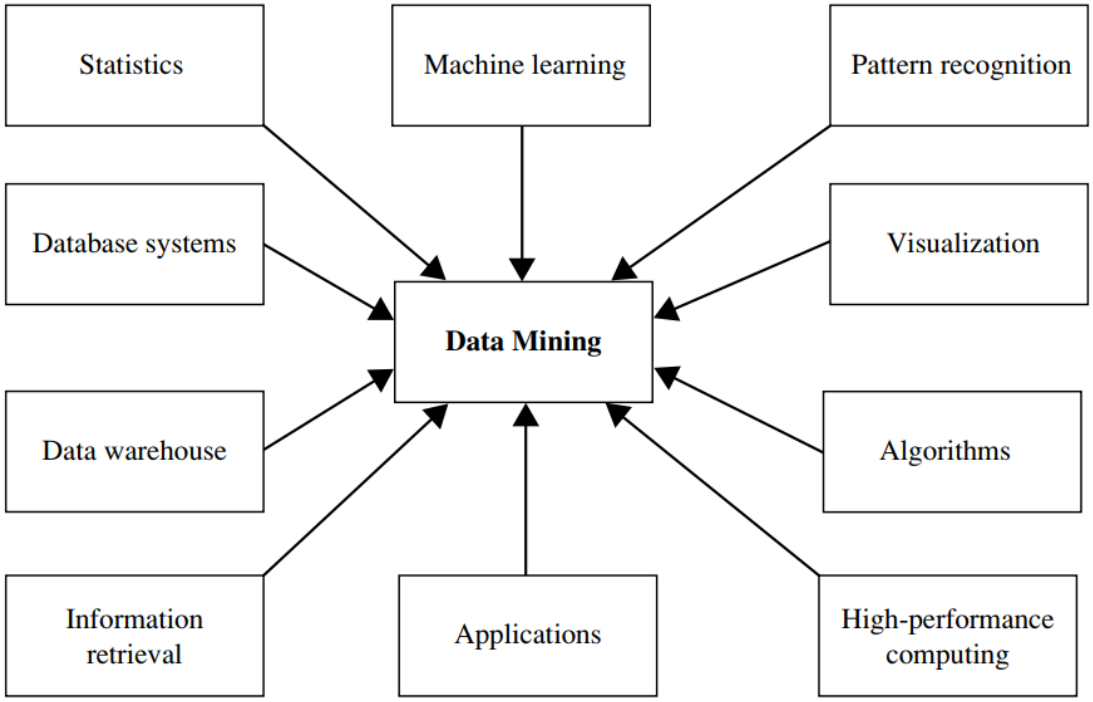
UNIT-I

Data Mining

We live in a world where vast amounts of data are collected daily. Analyzing such data is an important need. Data mining can meet this need by providing tools to discover knowledge from data.

Data Mining is a fast growing field also known as knowledge discovery from data, or KDD or knowledge mining from data, knowledge extraction, data/pattern analysis, data archaeology, and data dredging.

This explosive growth of available data volume is a result of the computerization of our society and the fast development of powerful data collection and storage tools.

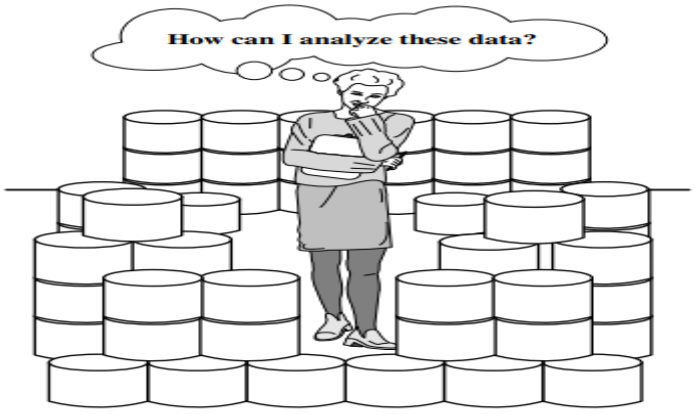


For example, large stores, such as Wal-Mart, handle hundreds of millions of transactions per week at thousands of branches around the world.

Global backbone telecommunication networks carry tens of petabytes of data traffic every day. The medical and health industry generates tremendous amounts of data from medical records, patient monitoring, and medical imaging. Billions of Web searches supported by search engines process tens of petabytes of data daily. Communities and social media have become increasingly important data sources, producing digital pictures and videos, blogs, Web communities, and various kinds of social networks. The list of sources that generate huge amounts of data is endless.

This explosively growing, widely available, and gigantic body of data makes our time truly the data age. Powerful and versatile tools are badly needed to automatically uncover valuable information from the tremendous amounts of data and to transform such data into organized knowledge. This necessity has led to the birth of data mining.

Data mining turns a large collection of data into knowledge. how data mining can turn a large collection of data into knowledge that can help meet a current global challenge.



The world is data rich but information poor.

Data Mining, can be defined in many different ways. The term data mining does not really present all the major components in the picture. knowledge discovery from data, or KDD, while others view data mining as merely an essential step in the process of knowledge discovery.

The knowledge discovery process is an iterative sequence of the following steps:

1. Data cleaning (to remove noise and inconsistent data)
2. Data integration (where multiple data sources may be combined)

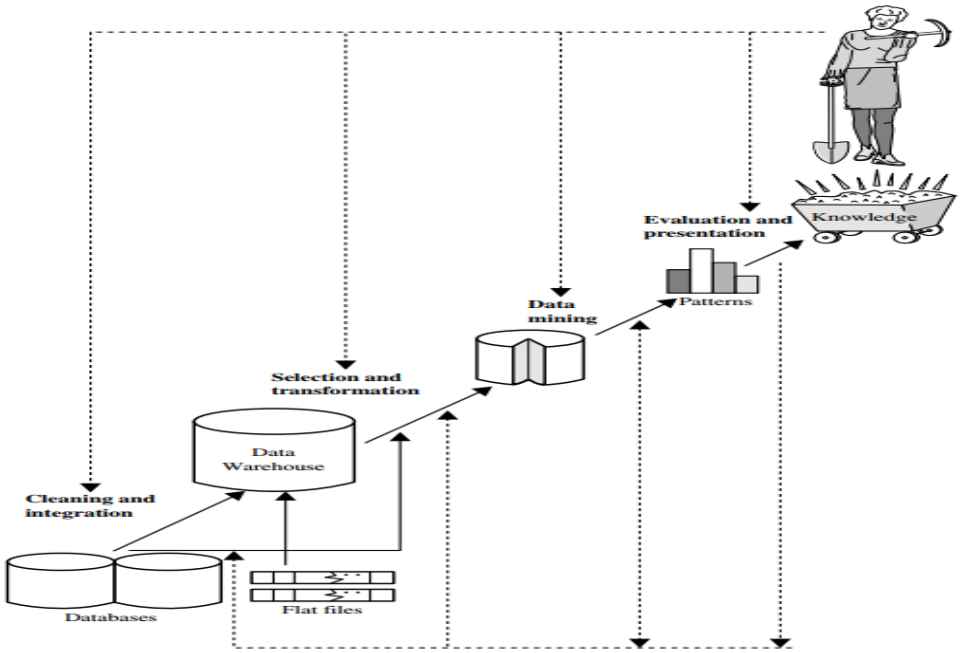


Figure 1.4 Data mining as a step in the process of knowledge discovery.

3. Data selection (where data relevant to the analysis task are retrieved from the database)
4. Data transformation (where data are transformed and consolidated into forms appropriate for mining by performing summary or aggregation operations)
5. Data mining (an essential process where intelligent methods are applied to extract data patterns)
6. Pattern evaluation (to identify the truly interesting patterns representing knowledge based on *interestingness measures*)
7. Knowledge presentation (where visualization and knowledge representation techniques are used to present mined knowledge to users)

Steps 1 through 4 are different forms of data preprocessing, where data are prepared for mining. The data mining step may interact with the user or a knowledge base. The interesting patterns are presented to the user and may be stored as new knowledge in the knowledge base.

The preceding view shows data mining as one step in the knowledge discovery process, an essential one because it uncovers hidden patterns for evaluation.

Data mining is the process of discovering interesting patterns and knowledge from large amounts of data. The data sources can include databases, data warehouses, the Web, other information repositories, or data that are streamed into the system dynamically.

Kinds of Data Can Be Mined:

Data Mining can be applied to any kind of data as long as the data are meaningful for a target application. The most basic forms of data for mining applications are database data, data warehouse data, and transactional data.

Data mining can also be applied to other forms of data e.g., data streams, ordered/sequence data, graph or networked data, spatial data, text data, multimedia data, and the WWW.

Database Data

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. The software programs provide mechanisms for defining database structures and data storage; for specifying and managing concurrent, shared, or distributed data access; and for ensuring consistency and security of the information stored despite system crashes or attempts at unauthorized access.

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. An ER data model represents the database as a set of entities and their relationships.

Relational data can be accessed by database queries written in a relational query language (e.g., SQL) or with the assistance of graphical user interfaces. A given query is transformed into a set of relational operations, such as join, selection, and projection, and is then optimized for efficient processing.

Example:

customer	(*cust_ID, name, address, age, occupation, annual_income, credit_information, category, . . .*)
item	(*item_ID, brand, category, type, price, place_made, supplier, cost, . . .*)
employee	(*empl_ID, name, category, group, salary, commission, . . .*)
branch	(*branch_ID, name, address, . . .*)
purchases	(*trans_ID, cust_ID, empl_ID, date, time, method_paid, amount*)
items_sold	(*trans_ID, item_ID, qty*)
works_at	(*empl_ID, branch_ID*)

Relational schema for a relational database, *AllElectronics*.

Relational data can be accessed by database queries written in a relational query language (e.g., SQL) or with the assistance of graphical user interfaces. A given query is transformed into a set of relational operations, such as join, selection, and projection, and is then optimized for efficient processing. A query allows retrieval of specified subsets of the data.

When mining relational databases, we can go further by searching for trends or data patterns. For example, data mining systems can analyze customer data to predict the credit risk of new customers based on their income, age, and previous credit information.

Data Warehouses

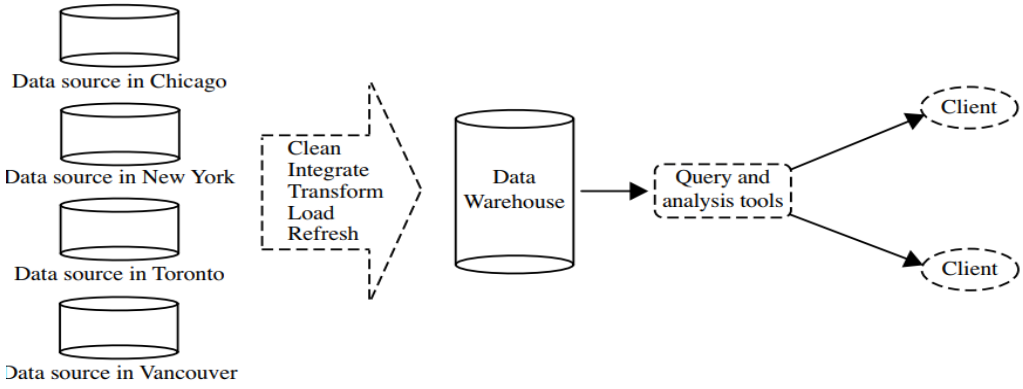
Suppose that AllElectronics is a successful international company with branches around the world. Each branch has its own set of databases. The president of AllElectronics has asked you to provide an analysis of the company's sales per item type per branch for the third quarter. This is a difficult task, particularly since the relevant data are spread out over several databases physically located at numerous sites.

A data warehouse is a repository of information collected from multiple sources, stored under a unified schema, and usually residing at a single site. Data warehouses are constructed via a process of data cleaning, data integration, data transformation, data loading, and periodic data refreshing.

To facilitate decision making, the data in a data warehouse are organized around major subjects. The data are stored to provide information from a historical perspective, such as in the past 6 to 12 months, and are typically summarized. The data warehouse may store a summary of the transactions per item type for each store or, summarized to a higher level, for each sales region

A data warehouse is usually modeled by a multidimensional data structure, called a data cube, in which 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 sum.sales amount. A

data cube provides a multidimensional view of data and allows the precomputation and fast access of summarized data.



By providing multidimensional data views and the precomputation of summarized data, data warehouse systems can provide inherent support for OLAP. Online analytical processing operations make use of background knowledge regarding the domain of the data being studied to allow the presentation of data at different levels of abstraction. Such operations accommodate different user viewpoints.

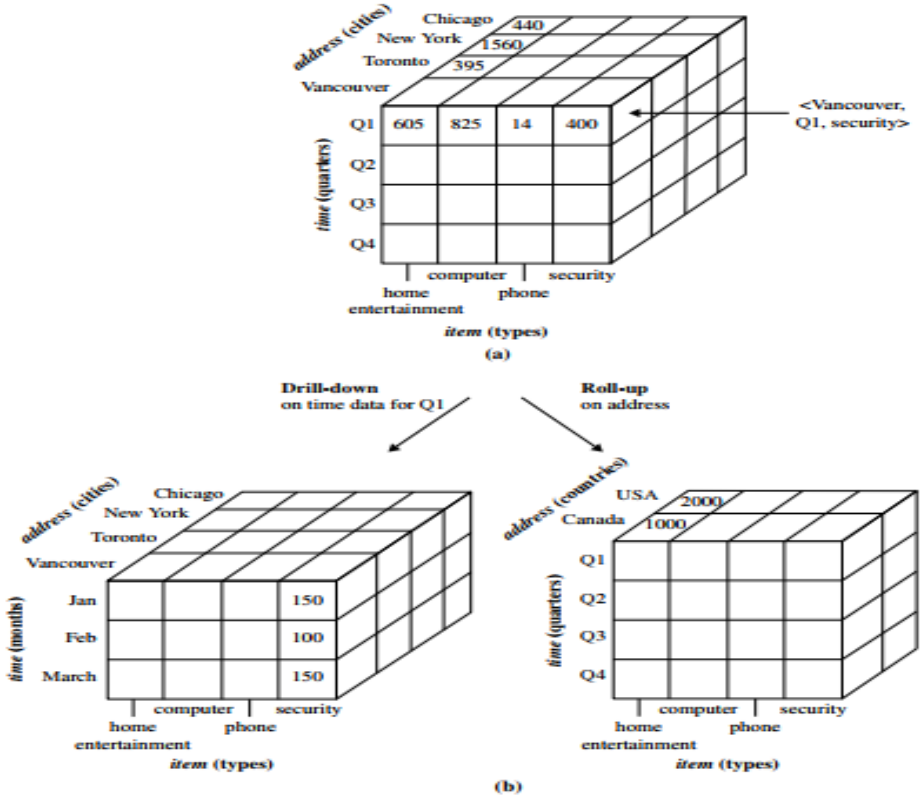


Figure 1.7 A multidimensional data cube, commonly used for data warehousing, (a) showing summarized data for *AllElectronics* and (b) showing summarized data resulting from drill-down and roll-up operations on the cube in (a). For improved readability, only some of the cube cell values are shown.

Transactional Data

In general, each record in a transactional database captures a transaction, such as a customer's purchase, a flight booking, or a user's clicks on a web page. A transaction typically includes a unique transaction identity number (*trans ID*) and a list of the items making up the transaction, such as the items purchased in the transaction. A transactional database may have additional tables, which contain other information related to the transactions, such as item description, information about the salesperson or the branch, and so on.

This kind of market basket data analysis would enable you to bundle groups of items together as a strategy for boosting sales.

For example, given the knowledge that printers are commonly purchased together with computers, you could offer certain printers at asteep discount (or even for free) to customers buying selected computers, in the hopes of selling more computers.

trans_ID	*list_of_item_IDs*
T100	I1, I3, I8, I16
T200	I2, I8
. . .	. . .

There are many other kinds of data can be seen in many applications: time-related or sequence data (e.g., historical records, stock exchange data, and time-series and biological sequence data), data streams (e.g., video surveillance and sensor data, which are continuously transmitted), spatial data (e.g., maps), engineering design data (e.g., the design of buildings, system components, or integrated circuits), hypertext and multimedia data (including text, image, video, and audio data), graph and networked data(e.g., social and information networks), and the Web (a huge, widely distributed information repository made available by the Internet). These applications bring about new challenges, like how to handle data carrying special structures (e.g., sequences, trees, graphs, and networks) and specific semantics (such as ordering, image, audio and video contents, and connectivity), and how to mine patterns that carry rich structures and semantics.

Kinds of Patterns Can Be Mined

Tasks can be classified into two categories: descriptive and predictive.

Descriptive mining tasks characterize properties of the data in a target data set.

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

Characterization and Discrimination

It can be useful to describe individual classes and concepts in summarized, concise, and yet precise terms. Such descriptions of a class or a concept are called class/concept descriptions. These descriptions can be derived using

(1) data characterization, by summarizing the data of the class under study (often called the target class) in general terms, or

(2) 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 target class of data. The data corresponding to the user-specified class are typically collected by a query. For example, to study the characteristics of software products with sales that increased by 10% in the previous year, the data related to such products can be collected by executing an SQL query on the sales database.

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. The resulting descriptions can also be presented as generalized relations or in rule form

Example:

A customer relationship manager at AllElectronics may order the following data mining task: Summarize the characteristics of customers who spend more than $5000 a year at AllElectronics. The result is a general profile of these customers, such as that they are 40 to 50 years old, employed, and have excellent credit ratings.

Data discrimination is a comparison of the general features of the target class data objects against the general features of objects from one or multiple contrasting classes. The target and contrasting classes can be specified by a user, and the corresponding data objects can be retrieved through database queries.

A customer relationship manager at AllElectronics may want to compare two groups of customers—those who shop for computer products regularly (e.g., more than twice a month) and those who rarely shop for such products (e.g.,less than three times a year). The resulting description provides a general comparative profile of these customers, such as that 80% of the customers who frequently purchase computer products are between 20 and 40 years old and have a university education, whereas 60% of the customers who infrequently buy such products are either seniors or youths, and have no university degree.

Mining Frequent Patterns, Associations, and Correlations

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

A frequent itemset refers to a set of items that often appear together in a transactional data set for example, milk and bread, which are frequently bought together in grocery stores by many customers.

A substructure can refer to different structural forms (e.g., graphs, trees, or lattices) that may be combined with itemsets or subsequences.

Example:

Association analysis.

Suppose that, as a marketing manager at AllElectronics, you want to know which items are frequently purchased together (i.e., within the same transaction).

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, 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 the transactions under analysis show that computer and software are purchased together.

Multidimensional association rule.

Age(X,“20..29”)^income(X,“40K..49K”)=>buys(X,“laptop”)

[support=2%,confidence= 60%].

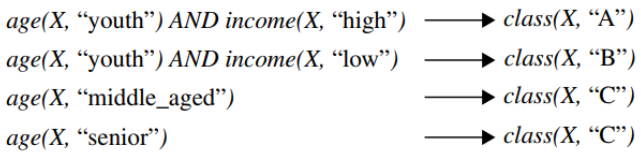
customers under study, 2% are 20 to 29 years old with an income of $40,000 to $49,000 and have purchased a laptop

The terminology used in multidimensional databases, where each attribute is referred to as a dimension, the above rule can be referred to as a multidimensional association rule.

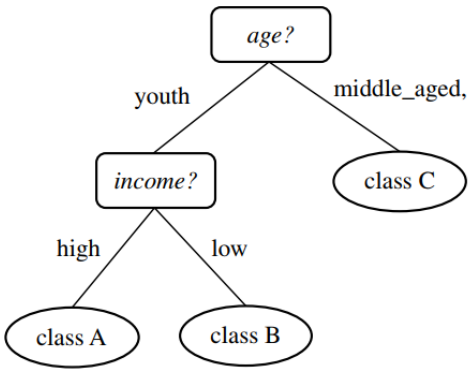
Classification and Regression for Predictive Analysis

Classification is the process of finding a model (or function) that describes and distinguishes data classes or concepts. The model are derived based on the analysis of a set of training data (i.e., data objects for which the class labels are known). The model is used to predict the class label of objects for which the the class label is unknown.

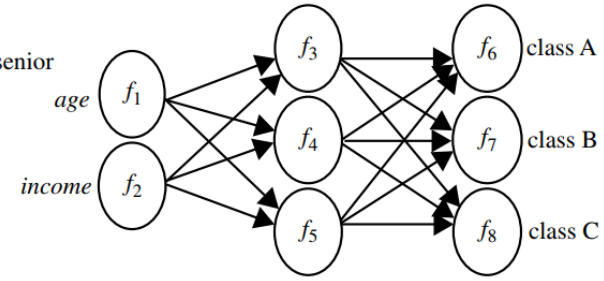
The derived model may be represented in various forms, such as *classification rules* (i.e., *IF-THEN rules*), *decision trees*, *mathematical formulae*, or *neural networks*. A decision tree is a flowchart-like tree structure, where each node denotes a test on an attribute value, each branch represents an outcome of the test, and tree leaves represent classes or class distributions.



(a)



(b)



(c)

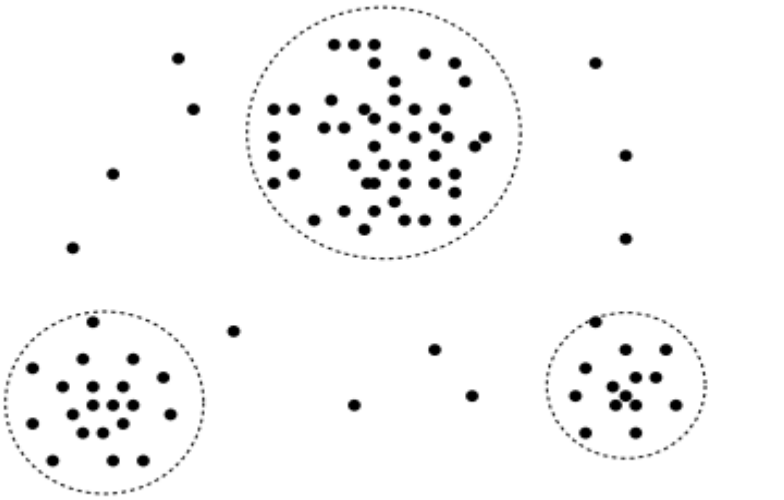
Classification predicts categorical (discrete, unordered) labels, regression models continuous-valued functions. Regression is used to predict missing or unavailable numerical data values rather than class labels. The term prediction refers to both numeric prediction and class label prediction.

Cluster Analysis

Clustering analyzes data objects without consulting class labels. In many cases, class labeled data may simply not exist at the beginning. Clustering can be used to generate class labels for a group of data. The objects are clustered or grouped based on the principle of maximizing the intra class similarity and minimizing the interclass similarity. That is, clusters of objects are formed so that objects within a cluster have high similarity in comparison to one another, but are rather dissimilar to objects in other clusters. Each cluster so formed can be viewed as a class of objects, from which rules can be derived.

Outlier Analysis

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



.10 A 2-D plot of customer data with respect to customer locations in a city, showing three data clusters.

The analysis of outlier data is referred to as outlier analysis or anomaly mining. Outliers may be detected using statistical tests that assume a distribution or probability model for the data, or using distance measures where objects that are remote from any other cluster are considered outliers.

Example:

Outlier analysis. Outlier analysis may uncover fraudulent usage of credit cards by detecting purchases of unusually large amounts for a given account number in comparison to regular charges incurred by the same account.

Patterns Interesting:

A data mining system has the potential to generate thousands or even millions of patterns, or rules. a small fraction of the patterns potentially generated would actually be of interest to a given user.

A pattern is interesting if it is

(1) easily understood by humans,

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

(3)potentially useful, and

(4) novel.

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

An interesting pattern represents knowledge.

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(X UY), where X U Y indicates that a transaction contains both X and Y , that is, the union of itemsets X and Y.

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.

Support(X=>Y)=P(XUY),
confidence(X => Y) =P(Y/X)

Accuracy tells us the percentage of data that are correctly classified by a rule. Coverage is similar to support, in that it tells us the percentage of data to which a rule applies.

Subjective interestingness measures are based on user beliefs in the data. These measures find patterns interesting if the patterns are unexpected (contradicting a user's belief) or offer strategic information on which the user can act.

Major Issues in Data Mining :

The major issues in data mining research, partitioning them into five groups:

- mining methodology,
- user interaction,
- efficiency and scalability,
- diversity of data types, and
- data mining and society.

Mining Methodology

Researchers have been vigorously developing new data mining methodologies. This involves the investigation of new kinds of knowledge, mining in multidimensional space, integrating methods from other disciplines, and the consideration of semantic ties among data objects. In addition, mining methodologies should consider issues such as data uncertainty, noise, and incompleteness. Some mining methods explore how user specified measures can be used to assess the interestingness of discovered patterns as well as guide the discovery process.

Mining various and new kinds of knowledge: Data mining covers a wide spectrum of data analysis and knowledge discovery tasks, from data characterization and discrimination to association and correlation analysis, classification, regression, clustering, outlier analysis, sequence analysis, and trend and evolution analysis. These tasks may use the same database in different ways and require the development of numerous data mining techniques. Due to the diversity of applications, new mining tasks continue to emerge, making data mining a dynamic and fast-growing field.

Mining various and new kinds of knowledge: Data mining covers a wide spectrum of data analysis and knowledge discovery tasks, from data characterization and discrimination to association and correlation analysis, classification, regression, clustering, outlier analysis, sequence analysis, and trend and evolution analysis. These tasks may use the same database in different ways and require the development of numerous data mining techniques. Due to the diversity of applications, new mining tasks continue to emerge, making data mining a dynamic and fast-growing field.

Data mining—an interdisciplinary effort: The power of data mining can be substantially enhanced by integrating new methods from multiple disciplines.

Boosting the power of discovery in a networked environment: Most data objects reside in a linked or interconnected environment, whether it be the Web, database relations, files, or documents. Semantic links across multiple data objects can be used to advantage in data mining. Knowledge derived in one set of objects can be used to boost the discovery of knowledge in a "related" or semantically linked set of objects.

Handling uncertainty, noise, or incompleteness of data: Data often contain noise, errors, exceptions, or uncertainty, or are incomplete. Errors and noise may confuse the data mining process, leading to the derivation of erroneous patterns. Data cleaning, data pre processing, outlier detection and removal, and uncertainty reasoning are examples of techniques that need to be integrated with the data mining process.

Pattern evaluation and pattern- or constraint-guided mining: Not all the patterns generated by data mining processes are interesting. What makes a pattern interesting may vary from user to user. Therefore, techniques are needed to assess the interestingness of discovered patterns based on subjective measures. These estimate the value of patterns with respect to a given user class, based on user beliefs or expectations. Moreover, by using interestingness measures or user-specified constraints to guide the discovery process, we may generate more interesting patterns and reduce the search space.

User Interaction

The user plays an important role in the data mining process. Interesting areas of research include how to interact with a data mining system, how to incorporate a user's background knowledge in mining, and how to visualize and comprehend data mining results.

Interactive mining: The data mining process should be highly interactive. Thus, it is important to build flexible user interfaces and an exploratory mining environment, facilitating the user's interaction with the system. A user may like to first sample a set of data, explore general characteristics of the data, and estimate potential mining results. Interactive mining should allow users to dynamically change the focus of a search, to refine mining requests based on returned results, and to drill, dice, and pivot through the data and knowledge space interactively, dynamically exploring "cube space" while mining.

Incorporation of background knowledge: Background knowledge, constraints, rules, and other information regarding the domain under study should be incorporated Major Issues in Data Mining into the knowledge discovery process. Such knowledge can be used for pattern evaluation as well as to guide the search toward interesting patterns.

Ad hoc data mining and data mining query languages: Query languages have played an important role in flexible searching because they allow users to pose ad hoc queries. Similarly, high-level data mining query languages or other high-level flexible user interfaces will give users the freedom to define ad hoc data mining tasks. This should facilitate specification of the relevant sets of data for analysis, the domain knowledge, the kinds of knowledge to be mined, and the conditions and constraints to be enforced on the discovered patterns.

Presentation and visualization of data mining results: How can a data mining system present data mining results, vividly and flexibly, so that the discovered knowledge can be easily understood and directly usable by humans? This is especially crucial if the data mining process is interactive. It requires the system to adopt expressive knowledge representations, user-friendly interfaces, and visualization techniques.

***Efficiency and Scalability*:** Efficiency and scalability are always considered when comparing data mining algorithms. As data amounts continue to multiply, these two factors are especially critical.

Efficiency and scalability of data mining algorithms: Data mining algorithms must be efficient and scalable in order to effectively extract information from huge amounts of data in many data repositories or in dynamic data streams. In other words, the running time of a data mining algorithm must be predictable, short, and acceptable by applications. Efficiency, scalability, performance, optimization, and the ability to execute in real time are key criteria that drive the development of many new data mining algorithms.

Parallel, distributed, and incremental mining algorithms: The humongous size of many data sets, the wide distribution of data, and the computational complexity of some data mining methods are factors that motivate the development of parallel and distributed data-intensive mining algorithms.

Such algorithms first partition the data into “pieces.” Each piece is processed, in parallel, by searching for patterns. The parallel processes may interact with one another. The patterns from each partition are eventually merged

Diversity of Database Types

The wide diversity of database types brings about challenges to data mining. These include Handling complex types of data: Diverse applications generate a wide spectrum of new data types, from structured data such as relational and data warehouse data to semi-structured and unstructured data; from stable data repositories to dynamic data streams; from simple data objects to temporal data, biological sequences, sensor data, spatial data, hypertext data, multimedia data, software program code, Web data, and social network data. It is unrealistic to expect one data mining system to mine all kinds of data, given the diversity of data types and the different goals of data mining.

Domain- or application-dedicated data mining systems are being constructed for indepth mining of specific kinds of data. The construction of effective and efficient data mining tools for diverse applications remains a challenging and active area of research.

Mining dynamic, networked, and global data repositories: Multiple sources of data are connected by the Internet and various kinds of networks, forming gigantic, distributed, and heterogeneous global information systems and networks. The discovery of knowledge from different sources of structured, semi-structured, or unstructured yet interconnected data with diverse data semantics poses great challenges to data mining. Mining such gigantic, interconnected information networks may help disclose many more patterns and knowledge in heterogeneous data sets than can be discovered from a small set of isolated data repositories. Web mining, multisource datamining, and information network mining have become challenging and fast-evolving data mining fields.

Data Mining and Society

Social impacts of data mining: With data mining penetrating our everyday lives, it is important to study the impact of data mining on society. The improper disclosure or use of data and the potential violation of individual privacy and data protection rights are areas of concern that need to be addressed.

Privacy-preserving data mining: Data mining will help scientific discovery, business management, economy recovery, and security protection. Studies on privacy-preserving data publishing and data mining are ongoing. The philosophy is to observe data sensitivity and preserve people’s privacy while performing successful data mining.

Invisible data mining: We cannot expect everyone in society to learn and master data mining techniques. More and more systems should have data mining functions built within so that people can perform data mining or use data mining results simply by mouse clicking, without any knowledge of data mining algorithms.

Intelligent search engines and Internet-based stores perform such invisible data mining by incorporating data mining into their components to improve their functionality and performance.

For example, when purchasing items online, users may be unaware that the store is likely collecting data on the buying patterns of its customers, which may be used to recommend other items for purchase in the future.

Data Mining Task Primitives

A data mining task can be specified in the form of a data mining query, which is input to the data mining system. A data mining query is defined in terms of data mining task primitives.

These primitives allow the user to a interactively communicate with the data mining system during discovery in order to direct the mining process, or examine the findings from different angles or depths.

The set of task-relevant data to be mined: This specifies the portions of the database or the set of data in which the user is interested. This includes the database attributes or data warehouse dimensions of interest.

The kind of knowledge **to be mined:** This specifies the data mining functions to be performed, such as characterization, discrimination, association or correlation analysis, classification, prediction, clustering, outlier analysis, or evolution analysis.
The background knowledge to be used in the discovery process: This knowledge about the domain to be mined is useful for guiding the knowledge discovery process and for evaluating the patterns found.

The interestingness measures and thresholds for pattern evaluation: They may be used to guide the mining process or, after discovery, to evaluate the discovered patterns. Different kinds of knowledge may have different interestingness measures. For example, interestingness measures for association rules include support and confidence. Rules whose support and confidence values are below user-specified thresholds are considered uninteresting.

The expected representation for visualizing the discovered patterns: This refers to the form in which discovered patterns are to be displayed, which may include rules, tables, charts, graphs, decision trees, and cubes.

A data mining query language can be designed to incorporate these primitives, allowing users to flexibly interact with data mining systems.

This facilitates a data mining system’s communication with other information systems and its integration with the overall information processing environment.

Real-world databases are highly susceptible to noisy, missing, and inconsistent data due to their typically huge size (often several gigabytes or more) and their likely origin from multiple, heterogenous sources. Low-quality data will lead to low-quality mining results.

There are several data preprocessing techniques.

Data cleaning can be applied to remove noise and correct inconsistencies in data.

Data integration merges data from multiple sources into a coherent data store such as a data warehouse.

Data reduction can reduce data size by, for instance, aggregating, eliminating redundant features, or clustering.

Data transformations (e.g., normalization) may be applied, where data are scaled to fall within a smaller range like 0.0 to 1.0. This can improve the accuracy and efficiency of mining algorithms involving distance measurements.

These techniques are not mutually exclusive; they may work together.

For example, data cleaning can involve transformations to correct wrong data, such as by transforming all entries for a date field to a common format.

Data quality, including accuracy, completeness, consistency, timeliness, believability, and interpretability. Users of your database system have reported errors, unusual values, and inconsistencies in the data recorded for some transactions.

In other words, the data you wish to analyze by data mining techniques are incomplete (lacking attribute values or certain attributes of interest, or containing only aggregate data); inaccurate or noisy (containing errors, or values that deviate from the expected); and inconsistent

the elements defining data quality: accuracy, completeness, and consistency. Inaccurate, incomplete, and inconsistent data are commonplace properties of large real-world databases and data warehouses. There are many possible reasons for inaccurate data (i.e., having incorrect attribute values). The data collection instruments used may be faulty. There may have been human or computer errors occurring at data entry. Users may purposely submit incorrect data values for mandatory fields when they do not wish to submit personal information

This is known as disguised missing data. Errors in data transmission can also occur. There may be technology limitations such as limited buffer size for coordinating synchronized data transfer and consumption. Incorrect data may also result from inconsistencies in naming conventions or data codes, or inconsistent formats for input fields

Incomplete data can occur for a number of reasons. Attributes of interest may not always be available, such as customer information for sales transaction data. Other data may not be included simply because they were not considered important at the time of entry.

Relevant data may not be recorded due to a misunderstanding or because of equipment malfunctions. Missing data, particularly for tuples with missing values for some attributes, may need to be inferred.

Data quality depends on the intended use of the data. Two different users may have very different assessments of the quality of a given database. For example, a marketing analyst may need to access the database mentioned before for a list of customer

addresses. Some of the addresses are outdated or incorrect, yet overall, 80% of the addresses are accurate. The marketing analyst considers this to be a large customer database for target marketing purposes and is pleased with the database's accuracy, although, as sales manager, you found the data inaccurate. Timeliness also affects data quality.

Suppose that you are overseeing the distribution of monthly sales bonuses to the top sales representatives at AllElectronics. Several sales representatives, however, fail to submit their sales records on time at the end of the month. There are also a number of corrections and adjustments that flow in after the month's end. For a period of time following each month, the data stored in the database are incomplete. However, once all of the data are received, it is correct. The fact that the month-end data are not updated in a timely fashion has a negative impact on the data quality.

Two other factors affecting data quality are believability and interpretability. Believability reflects how much the data are trusted by users, while interpretability reflects how easy the data are understood. Suppose that a database, at one point, had several errors, all of which have since been corrected. The past errors, however, had caused many problems for sales department users, and so they no longer trust the data. The data also use many accounting codes, which the sales department does not know how to interpret. Even though the database is now accurate, complete, consistent, and timely, sales department users may regard it as of low quality due to poor believability and interpretability

Major Tasks in Data Preprocessing

Data preprocessing, namely, data cleaning, data integration, data reduction,

Data cleaning routines work to "clean" the data by filling in missing values, smoothing noisy data, identifying or removing outliers, and resolving inconsistencies. If users believe the data are dirty, they are unlikely to trust the results of any data mining that has been applied.

Furthermore, dirty data can cause confusion for the mining procedure, resulting in unreliable output. Although most mining routines have some procedures for dealing with incomplete or noisy data, they are not always robust.

Getting back to your task at AllElectronics, suppose that you would like to include data from multiple sources in your analysis. This would involve integrating multiple databases, data cubes, or files (i.e., data integration).

Data reduction obtains a reduced representation of the data set that is much smaller in volume, yet produces the same (or almost the same) analytical results. Data reduction strategies include dimensionality reduction and numerosity reduction.

In dimensionality reduction, data encoding schemes are applied so as to obtain a reduced or "compressed" representation of the original data. Examples include data compression techniques (e.g., wavelet transforms and principal components analysis),

Attribute subset selection (e.g., removing irrelevant attributes), and attribute construction (e.g., where a small set of more useful attributes is derived from the original set)

In numerosity reduction, the data are replaced by alternative, smaller representations using parametric models (e.g., regression or log-linear models) or nonparametric models (e.g., histograms, clusters, sampling, or data aggregation).

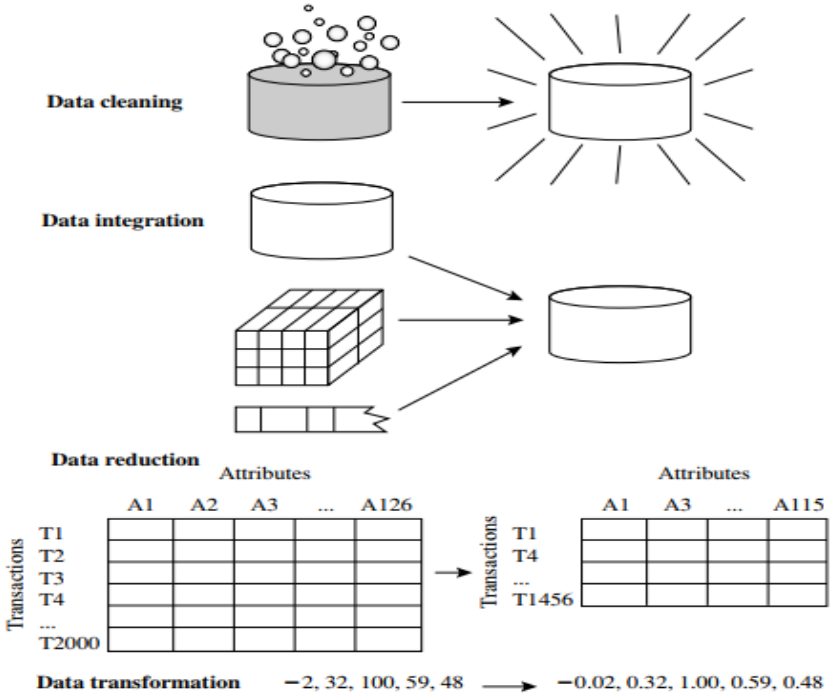


Figure 3.1 Forms of data preprocessing.

Getting back to your data, you have decided, say, that you would like to use a distance based mining algorithm for your analysis, such as neural networks, nearest-neighbor classifiers, or clustering.

Such methods provide better results if the data to be analyzed have been normalized, that is, scaled to a smaller range such as [0.0, 1.0]. Your customer data, for example, contain the attributes age and annual salary. The annual salary attribute usually takes much larger values than age. Therefore, if the attributes are left unnormalized, the distance measurements taken on annual salary will generally outweigh distance measurements taken on age. Discretization and concept hierarchy generation can also be useful, where raw data values for attributes are replaced by ranges or higher conceptual levels.

For example, raw values for age may be replaced by higher-level concepts, such as youth, adult, or senior.

Real-world data tend to be incomplete, noisy, and inconsistent. Data cleaning (or data cleansing) routines attempt to fill in missing values, smooth out noise while identifying outliers, and correct inconsistencies in the data.

Missing Values

Imagine that you need to analyze AllElectronics sales and customer data. You note that many tuples have no recorded value for several attributes such as customer income. How can you go about filling in the missing values for this attribute? Let's look at the following methods.

1. ***Ignore the tuple*:** This is usually done when the class label is missing. 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. By ignoring the tuple, we do not make use of the remaining attributes' values in the tuple. Such data could have been useful to the task at hand.

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

3. ***Use a global constant to fill in the missing value***: Replace all missing attribute values by the same constant such as a label like "Unknown" or -1. 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 foolproof.

4. ***Use a measure of central tendency for the attribute (e.g., the mean or median) to fill in the missing value*:** Suppose that the data distribution regarding the income of AllElectronics customers is symmetric and that the mean income is $56,000. Use this value to replace the missing value for income.

5. ***Use the attribute mean or median for all samples belonging to the same class as the given tuple*:** if classifying customers according to credit risk, we may replace the missing value with the mean income value for customers in the same credit risk category as that of the given tuple. If the data distribution for a given class is skewed, the median value is a better choice.

6. ***Use the most probable value to fill in the missing value*:** This may be determined with regression, inference-based tools using a Bayesian formalism, or decision tree

Noisy Data

Noise *is a random error or variance in a measured variable.* we saw how some basic statistical description techniques (e.g., boxplots and scatter plots), and methods of data visualization can be used to identify outliers, which may represent noise.

Binning: Binning methods smooth a sorted data value by consulting its "neighborhood," that is, the 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.

In this example, the data for price are first sorted and then partitioned into equal-frequency bins of size 3 (i.e., each bin contains three values). 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.

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

Partition into (equal-frequency) bins:

Bin 1: 4, 8, 15
Bin 2: 21, 21, 24
Bin 3: 25, 28, 34

Smoothing by bin means:

Bin 1: 9, 9, 9
Bin 2: 22, 22, 22
Bin 3: 29, 29, 29

Smoothing by bin boundaries:

Bin 1: 4, 4, 15
Bin 2: 21, 21, 24
Bin 3: 25, 25, 34

Binning methods for data smoothing.

Regression:

Data smoothing can also be done by regression, a technique that conforms data values to a function. Linear regression involves finding the "best" line to fit two attributes so that one attribute can be used to predict the other. Multiple linear regression is an extension of linear regression, where more than two attributes are involved and the data are fit to a multidimensional surface.

Outlier analysis:

Outliers may be detected by clustering, for example, where similar values are organized into groups, or "clusters." Intuitively, values that fall outside of the set of clusters may be considered outliers.

Many data smoothing methods are also used for data discretization (a form of data transformation) and data reduction. For example, the binning techniques described before reduce the number of distinct values per attribute. This acts as a form of data reduction for logic-based data mining methods, such as decision tree induction, which repeatedly makes value comparisons on sorted data.

Concept hierarchies are a form of data discretization that can also be used for data smoothing. A concept hierarchy for price, for example, may map real price values into inexpensive, moderately priced, and expensive.

Data Cleaning as a Process

Missing values, noise, and inconsistencies contribute to inaccurate data. So far, we have looked at techniques for handling missing data and for smoothing data.

The first step in data cleaning as a process is discrepancy detection. Discrepancies can be caused by several factors, including poorly designed data entry forms that have many optional fields, human error in data entry, deliberate errors (e.g., respondents not wanting to divulge information about themselves), and data decay (e.g., outdated addresses). Discrepancies may also arise from inconsistent data representations and inconsistent use of codes.

Field overloading is another error source that typically results when developers squeeze new attribute definitions into unused (bit) portions of already defined attributes

A unique rule says that each value of the given attribute must be different from all other values for that attribute. A consecutive rule says that there can be no missing values between the lowest and highest values for the attribute, and that all values must also be unique (e.g., as in check numbers). A null rule specifies the use of blanks, question marks, special characters, or other strings that may indicate the null condition (e.g., where a value for a given attribute is not available), and how such values should be handled.

Data scrubbing tools use simple domain knowledge (e.g., knowledge of postal addresses and spell-checking) to detect errors and make corrections in the data. These tools rely on parsing and fuzzy matching techniques when cleaning data from multiple sources.

Data auditing tools find discrepancies by analyzing the data to discover rules and relationships, and detecting data that violate such conditions. They are variants of data mining tools.

Data analysis task will involve data integration, which combines data from multiple sources into a coherent data store, as in data warehousing. These sources may include multiple databases, data cubes, or flat files. There are a number of issues to consider during data integration.

This is referred to as the entity identification problem.

For example, how can the data analyst or the computer be sure that customer id in one database and cust number in another refer to the same attribute? Examples of metadata for each attribute include the name, meaning, data type, and range of values permitted for the attribute, and null rules for handling blank, zero, or null values.

When matching attributes from one database to another during integration, special attention must be paid to the structure of the data. This is to ensure that any attribute functional dependencies and referential constraints in the source system match those in the target system.

For example, in one system, a discount may be applied to the order, whereas in another system it is applied to each individual line item within the order. If this is not caught before integration, items in the target system may be improperly discounted.

Redundancy and Correlation Analysis

Redundancy is another important issue in data integration. An attribute (such as annual revenue, for instance) may be redundant if it can be "derived" from another attribute or set of attributes. Inconsistencies in attribute or dimension naming can also cause redundancies in the resulting data set.

Some redundancies can be detected by *correlation analysis*. Given two attributes, such analysis can measure how strongly one attribute implies the other, based on the available data. For nominal data, we use the χ2 (chi-square) test. For numeric attributes, we can use the correlation coefficient and covariance.

χ*2* Correlation Test for Nominal Data

For nominal data, a correlation relationship between two attributes, A *and* B*, can be discovered by a χ2 (chi-square) test. Suppose* A *has* c *distinct values, namely* a*1*,a*2,…,*ac. B *has* r *distinct values, namely* b*1*,b*2,….,*br. *The data tuples described by* A *and* B *can be shown as a contingency table, with the* c *values of* A *making up the columns and the* r *values of* B *making up the rows. Let (*ai,bj*) denote the joint event that attribute* A *takes on value* ai *and attribute* B *takes on value* bj*, that is, where* A= ai,B=bj. *Each and every possible (*Ai,Bj*) joint event has its own cell (or slot) in the table. The χ2 value (also known as the* Pearson *χ2* statistic*) is computed as*

$$\chi^2 = \sum_{i=1}^{c} \sum_{j=1}^{r} \frac{(o_{ij} - e_{ij})^2}{e_{ij}},$$

$$e_{ij} = \frac{count(A = a_i) \times count(B = b_j)}{n},$$

where n is the number of data tuples, count. A = |ai| is the number of tuples having value ai for A, and count.B= bj is the number of tuples having value bj for B. The sum is computed over all of the r × c cells.

The χ2 statistic tests the hypothesis that A and B are independent, that is, there is no correlation between them. The test is based on a significance level, with (r – 1) × (c – 1) degrees of freedom.

where oij is the observed frequency (i.e., actual count) of the joint event (Ai,Bj) and eij is
the expected frequency of (Ai,Bj), which can be computed as

Example 2.1's 2 × 2 Contingency Table Data

	male	*female*	*Total*
fiction	250 (90)	200 (360)	450
non_fiction	50 (210)	1000 (840)	1050
Total	300	1200	1500

Note: Are *gender* and *preferred_reading* correlated?

Using Eq. (3.1) for χ^2 computation, we get

$$\chi^2 = \frac{(250-90)^2}{90} + \frac{(50-210)^2}{210} + \frac{(200-360)^2}{360} + \frac{(1000-840)^2}{840}$$
$$= 284.44 + 121.90 + 71.11 + 30.48 = 507.93.$$

Correlation Coefficient for Numeric Data

For numeric attributes, we can evaluate the correlation between two attributes, A and B, by computing the correlation coefficient (also known as Pearson's product moment coefficient

$$r_{A,B} = \frac{\sum_{i=1}^{n}(a_i - \bar{A})(b_i - \bar{B})}{n\sigma_A\sigma_B} = \frac{\sum_{i=1}^{n}(a_i b_i) - n\bar{A}\bar{B}}{n\sigma_A\sigma_B},$$

where n is the number of tuples, ai and bi are the respective values of A and B in tuple i,

A_N and B_N are the respective mean values of A and B, σ_A and σ_B are the respective standard deviations of A and B

$-1 \leq r_{A,B} \leq 1$.

If $r_{A,B}$ *is greater than 0, then* A *and* B *are* positively correlated, *meaning that the values of* A *increase as the values of* B *increase.*

The higher the value, the stronger the correlation Hence, a higher value may indicate that A *(or* B*) may be removed as a redundancy.*

Covariance of Numeric Data

Consider two numeric attributes A and B, and a set of n observations {(a1,b1),……,(an,bn)}. The mean values of A and B, respectively, are also known as the expected values on A and B, that is,

$$E(A) = \bar{A} = \frac{\sum_{i=1}^{n} a_i}{n} \qquad E(B) = \bar{B} = \frac{\sum_{i=1}^{n} b_i}{n}.$$

The covariance between A and B is defined as

$$Cov(A, B) = E((A - \bar{A})(B - \bar{B})) = \frac{\sum_{i=1}^{n}(a_i - \bar{A})(b_i - \bar{B})}{n}.$$

$$r_{A,B} = \frac{Cov(A, B)}{\sigma_A \sigma_B}$$

the covariance between A and B is positive. On the other hand, if one of the attributes tends to be above its expected value when the other attribute is below its expected value, then the covariance of A and B is negative.

Covariance analysis of numeric attributes. Consider Table 3.2, which presents a simplified example of stock prices observed at five time points for *AllElectronics* and *HighTech*, a high-tech company. If the stocks are affected by the same industry trends, will their prices rise or fall together?

$$E(AllElectronics) = \frac{6 + 5 + 4 + 3 + 2}{5} = \frac{20}{5} = \$4$$

and

$$E(HighTech) = \frac{20 + 10 + 14 + 5 + 5}{5} = \frac{54}{5} = \$10.80.$$

Thus, using Eq. (3.4), we compute

$$Cov(AllElectroncis, HighTech) = \frac{6 \times 20 + 5 \times 10 + 4 \times 14 + 3 \times 5 + 2 \times 5}{5} - 4 \times 10.80$$
$$= 50.2 - 43.2 = 7.$$

Therefore, given the positive covariance we can say that stock prices for both companies rise together. ■

Tuple Duplication

In addition to detecting redundancies between attributes, duplication should also be detected at the tuple level. The use of denormalized tables is another source of data redundancy.

Inconsistencies often arise between various duplicates, due to inaccurate data entry or updating some but not all data occurrences.

Data Value Conflict Detection and Resolution:

Data integration also involves the detection and resolution of data value conflicts.

For example, for the same real-world entity, attribute values from different sources may differ. This may be due to differences in representation, scaling, or encoding.

For instance, a weight attribute may be stored in metric units in one system and British imperial units in another.

For a hotel chain, the price of rooms in different cities may involve not only different currencies but also different services (e.g., free breakfast) and taxes. When exchanging information between schools, for example, each school may have its own curriculum and grading scheme.

One university may adopt a quarter system, offer three courses on database systems, and assign grades from A to F, whereas another may adopt a semester system, offer two courses on databases, and assign grades from 1 to 10. It is difficult to work out precise course-to-grade transformation rules between the two universities, making information exchange difficult.

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.

Mining on the reduced data set should be more efficient yet produce the same (or almost the same) analytical results

Overview of Data Reduction Strategies

Data reduction strategies include dimensionality reduction, numerosity reduction, and data compression.

Dimensionality reduction is the process of reducing the number of random variables or attributes under consideration. Dimensionality reduction methods include wavelet transforms and principal components analysis, which transform or project the original data onto a smaller space.

Attribute subset selection is a method of dimensionality reduction in which irrelevant, weakly relevant, or redundant attributes or dimensions are detected and removed

Numerosity reduction techniques replace the original data volume by alternative, smaller forms of data representation. These techniques may be parametric or nonparametric. For parametric methods, a model is used to estimate the data, so that typically only the data parameters need to be stored, instead of the actual data. Regression and log-linear models are examples.

Nonparametric methods for storing reduced representations of the data include histograms, clustering, sampling, and datacube aggregation (Section 3.4.9).

In data compression, 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 information loss, the data reduction is called

lossless. If, instead, we can reconstruct only an approximation of the original data, then the data reduction is called lossy.

Dimensionality reduction and numerosity reduction techniques can also be considered forms of data compression.

There are many other ways of organizing methods of data reduction. The computational time spent on data reduction should not outweigh or "erase" the time saved by mining on a reduced data set size.

Wavelet Transforms

The discrete wavelet transform (DWT) is a linear signal processing technique that, when applied to a data vector *X*, transforms it to a numerically different vector, *X*0, of wavelet coefficients. The two vectors are of the same length. When applying this technique to data reduction,

we consider each tuple as an n-dimensional data vector, that is, $X = \{x1,x2,.....,xn\}$, depicting n measurements made on the tuple from n database attributes.

The usefulness lies in the fact that the wavelet transformed data can be truncated. A compressed approximation of the data can be retained by storing only a small fraction of the strongest of the wavelet coefficients.

For example, all wavelet coefficients larger than some user-specified threshold can be retained. All other coefficients are set to 0. The resulting data representation is therefore very sparse, so that operations that can take advantage of data sparsity are computationally very fast if performed in wavelet space. The technique also works to remove noise without smoothing out the main features of the data, making it effective for data.

Given a set of coefficients, an approximation of the original data can be constructed by applying the inverse of the DWT used. 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. That is, if the same number of coefficients is retained for a DWT and a DFT of a given data vector, the DWT version will provide a more accurate approximation of the original data.

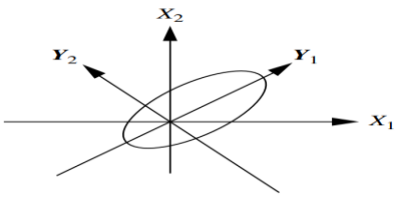
Suppose that the data to be reduced consist of tuples or data vectors described by n attributes or dimensions. Principal components analysis (PCA; also called the Karhunen-Loeve, or K-L, method) searches for k n-dimensional orthogonal vectors that can best be used to represent the data, where $k \leq n$. The original data are thus projected onto a much smaller space, resulting in dimensionality reduction.

The basic procedure is as follows:

1. 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.

2. PCA computes k orthonormal vectors that 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.

3. 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. That is, the sorted axes are such that the first axis shows the most variance among the data, the second axis shows the next highest variance, and so on.



Principal components analysis. Y_1 and Y_2 are the first two principal components for the given data.

4. Because the components are sorted in decreasing order of "significance," the data size can be reduced by eliminating the weaker components, that is, those with low variance. Using the strongest principal components, it should be possible to reconstruct a good approximation of the original data.

PCA can be applied to ordered and unordered attributes, and can handle sparse data and skewed data. Multidimensional data of more than two dimensions can be handled by reducing the problem to two dimensions. Principal components may be used as inputs to multiple regression and cluster analysis.

Attribute Subset Selection

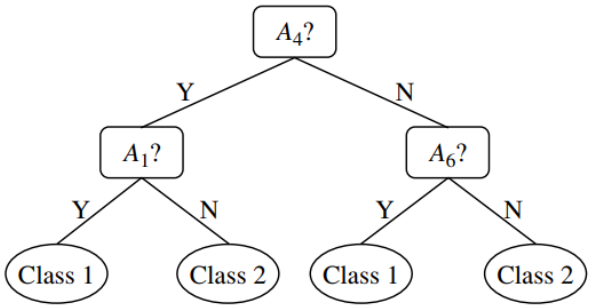
Data sets for analysis may contain hundreds of attributes, many of which may be irrelevant to the mining task or redundant.

Attribute subset selection4 reduces the data set size by removing irrelevant or redundant attributes (or dimensions).

The goal of attribute subset selection is to find a minimum set of attributes such that the resulting probability distribution of the data classes is as close as possible to the original distribution obtained using all attributes.

Mining on a reduced set of attributes has an additional benefit: It reduces the number

of attributes appearing in the discovered patterns, helping to make the patterns easier to understand.

Forward selection	Backward elimination	Decision tree induction
Initial attribute set: $\{A_1, A_2, A_3, A_4, A_5, A_6\}$ Initial reduced set: $\{\}$ $=> \{A_1\}$ $=> \{A_1, A_4\}$ => Reduced attribute set: $\{A_1, A_4, A_6\}$	Initial attribute set: $\{A_1, A_2, A_3, A_4, A_5, A_6\}$ $=> \{A_1, A_3, A_4, A_5, A_6\}$ $=> \{A_1, A_4, A_5, A_6\}$ => Reduced attribute set: $\{A_1, A_4, A_6\}$	Initial attribute set: $\{A_1, A_2, A_3, A_4, A_5, A_6\}$  => Reduced attribute set: $\{A_1, A_4, A_6\}$

Greedy (heuristic) methods for attribute subset selection.

1. Stepwise forward selection: The procedure starts with an empty set of attributes as the reduced set. The best of the original attributes is determined and added to the reduced set. At each subsequent iteration or step, the best of the remaining original attributes is added to the set.
2. Stepwise backward elimination: The procedure starts with the full set of attributes. At each step, it removes the worst attribute remaining in the set.
3. Combination of forward selection and backward elimination: The stepwise forward selection and backward elimination methods can be combined so that, at each step, the procedure selects the best attribute and removes the worst from among the remaining attributes.
4. Decision tree induction: Decision tree algorithms (e.g., ID3, C4.5, and CART) were originally intended for classification. Decision tree induction constructs a flowchartlike structure where each internal (nonleaf) 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.

Regression and Log-Linear Models: Parametric

Data Reduction Regression and log-linear models can be used to approximate the given data. In (simple) 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

$$y = wx + b$$

where the variance of y is assumed to be constant. In the context of data mining, x and y are numeric database attributes. The coefficients, w and b

Multiple linear regressionis an extension of (simple) linear regression, which allows a response variable, y, to be modeled as a linear function of two or more predictor variables.

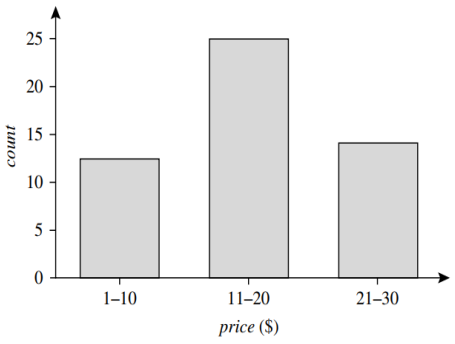
Log-linear models approximate discrete multidimensional probability distributions. Given a set of tuples in n dimensions (e.g., described by n attributes), we can consider each tuple as a point in an n-dimensional space.

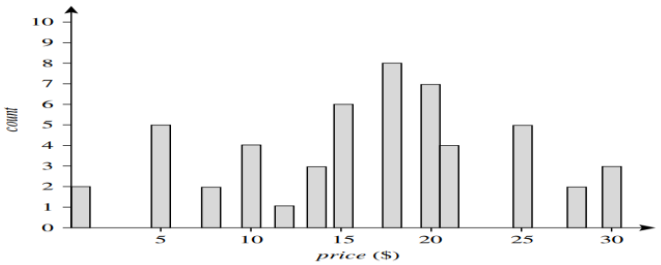
Log-linear models can be used to estimate the probability of each point in a multidimensional space for a set of discretized attributes, based on a smaller subset of dimensional combinations. This allows a higher-dimensional data space to be constructed from lower-dimensional spaces.

Histograms

Histograms use binning to approximate data distributions and are a popular form of data reduction. Histograms were introduced in Section 2.2.3. A histogram *for an attribute, A, partitions the data distribution of A into disjoint subsets, referred to as buckets or bins. If each bucket represents only a single attribute–value/frequency pair, the buckets are called singleton buckets. Often, buckets instead represent continuous ranges for the given attribute.*

Histograms. The following data are a list of AllElectronics prices for commonly sold items (rounded to the nearest dollar). The numbers have been sorted: 1, 1, 5, 5, 5, 5, 5, 8, 8, 10, 10, 10, 10, 12, 14, 14, 14, 15, 15, 15, 15, 15, 15, 18, 18, 18, 18, 18, 18, 18, 18, 20, 20, 20, 20, 20, 20, 20, 21, 21, 21, 21, 25, 25, 25, 25, 25, 28, 28, 30, 30, 30.





Equal-width*: In an equal-width histogram, the width of each bucket range is uniform.*

Equal-frequency*: In an equal-frequency histogram, the buckets are created so that, roughly, the frequency of each bucket is constant.*

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 because it allows a large data set to be represented by a much smaller random data sample (or subset). Suppose that a large data set, D*, contains* N *tuples.*

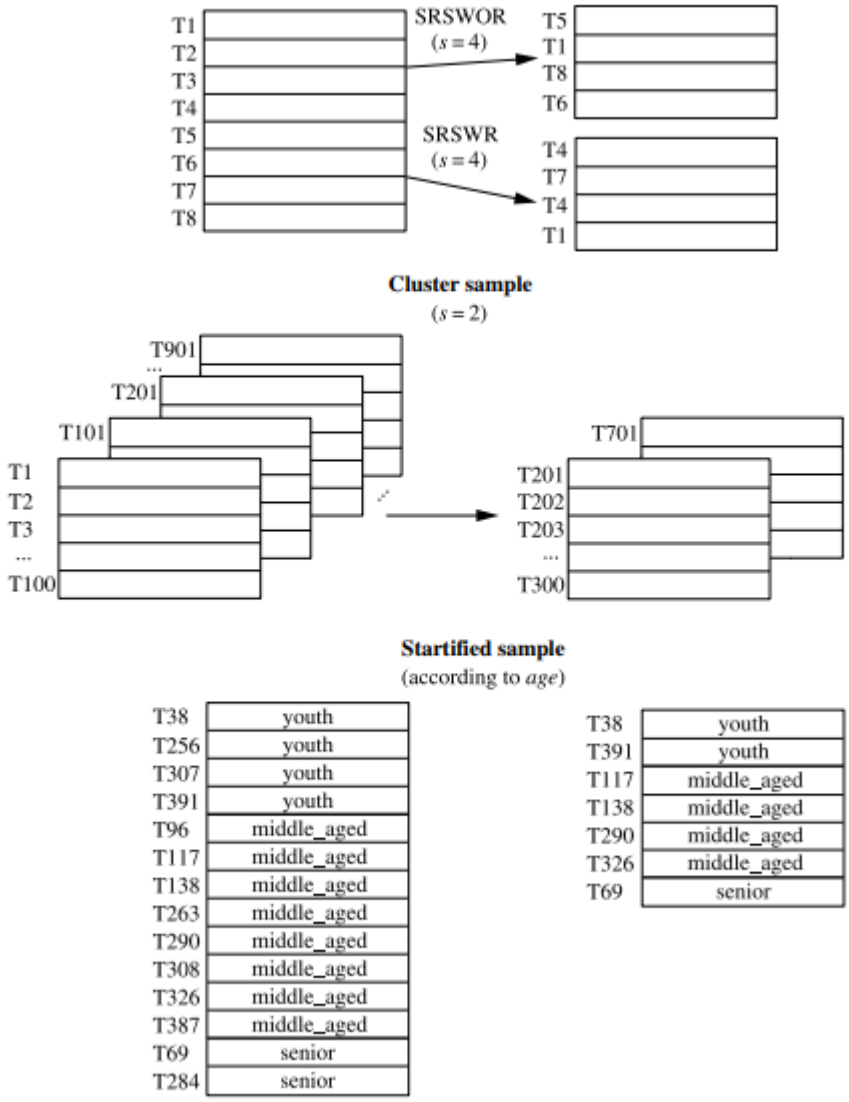
Simple random sample without replacement (SRSWOR) of size s***:*** *This is created by drawing* s *of the* N *tuples from* D *(*s < N*), where the probability of drawing any tuple in* D *is 1*=N*, that is, all tuples are equally likely to be sampled.*

Simple random sample with replacement (SRSWR) of size s***:*** *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 an SRS of* s *clusters can be obtained, where* s < M*. For example, tuples in a database are usually retrieved a page at a time, so that each page can be considered* a cluster.

A reduced data representation can be obtained by applying, say, SRSWOR to the pages, resulting in a cluster sample of the tuples. Other clustering criteria conveying rich semantics can also be explored. For example, in a spatial database, we may choose to define clusters geographically based on how closely different areas are located.

Stratified sample: If D is divided into mutually disjoint parts called strata, a stratified sample of D is generated by obtaining an SRS at each stratum. This helps ensure a representative sample, especially when the data are skewed. For example, a stratified sample may be obtained from customer data, where a stratum is created for each customer age group. In this way, the age group having the smallest number of customers will be sure to be represented.



Data Transformation and Data Discretization

In this preprocessing step, the data are transformed or consolidated so that the resulting mining process may be more efficient, and the patterns found may be easier to understand

Data Transformation Strategies Overview

In data transformation, the data are transformed or consolidated into forms appropriate for mining. Strategies for data transformation include the following:
1. *Smoothing*, which works to remove noise from the data. Techniques include binning, regression, and clustering.

2. *Attribute construction* (or feature construction), where new attributes are constructed and added from the given set of attributes to help the mining process.

3. *Aggregation*, where summary or aggregation operations are applied to the data. For example, the daily sales data may be aggregated so as to compute monthly and annual total amounts. This step is typically used in constructing a data cube for data analysis at multiple abstraction levels.

4. *Normalization*, where the attribute data are scaled so as to fall within a smaller range, such as -1.0 to 1.0, or 0.0 to 1.0.

5. *Discretization*, where the raw values of a numeric attribute (e.g., age) are replaced by interval labels (e.g., 0–10, 11–20, etc.) or conceptual labels (e.g., youth, adult, senior). The labels, in turn, can be recursively organized into higher-level concepts, resulting in a concept hierarchy for the numeric attribute. Figure 3.12 shows a concept hierarchy for the attribute price. More than one concept hierarchy can be defined for the same attribute to accommodate the needs of various users.

6. *Concept hierarchy generation for nominal data*, where attributes such as street can be generalized to higher-level concepts, like city or country. Many hierarchies for nominal attributes are implicit within the database schema and can be automatically defined at the schema definition level.

Data Transformation by Normalization

To help avoid dependence on the choice of measurement units, the data should be normalized or standardized. This involves transforming the data to fall within a smaller or common range such as [-1,1] or [0.0, 1.0].

Min-max normalization performs a linear transformation on the original data. Suppose that minA and maxA are the minimum and maximum values of an attribute, A. Min-max normalization maps a value

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

In z-score normalization (or *zero-mean normalization*), the values for an attribute, *A*, are normalized based on the mean (i.e., average) and standard deviation of *A*. A value, *vi*, of *A* is normalized to *vi*0 by computing

$$v_i' = \frac{v_i - \bar{A}}{\sigma_A},$$

Concept Hierarchy Generation for Nominal Data

Nominal attributes have a finite (but possibly large) number of distinct values, with no ordering among the values.

Examples include geographic location, job category, and item type.

The concept hierarchies can be used to transform the data into multiple levels of granularity. For example, data mining patterns regarding sales may be found relating to specific regions or countries, in addition to individual branch locations.

Specification of a partial ordering of attributes explicitly at the schema level by users or experts:

Concept hierarchies for nominal attributes or dimensions typically involve a group of attributes. A user or expert can easily define a concept hierarchy by specifying a partial or total ordering of the attributes at the schema level.

For example, suppose that a relational database contains the following group of attributes: street, city, province or state, and country. Similarly, a data warehouse location dimension may contain the same attributes. A hierarchy can be defined by specifying the total ordering among these attributes at the schema level such as street < city < province or state < country.

Specification of a set of attributes, but not of their partial ordering: A user may specify a set of attributes forming a concept hierarchy, but omit to explicitly state their partial ordering. The system can then try to automatically generate the attribute ordering so as to construct a meaningful concept hierarchy