UNIT III

DATA MINING

Introduction – Data – Types of Data – Data Mining Functionalities – Interestingness of Patterns – Classification of Data Mining Systems – Data Mining Task Primitives – Integration of a Data Mining System with a Data Warehouse – Issues –Data Preprocessing.

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

Tid	Refund	Marital Status	Taxable Income	Cheat
1	Yes	Single	125K	No
2	No	Married	100K	No
3	No	Single	70K	No
4	Yes	Married	120K	No
5	No	Divorced	95K	Yes
6	No	Married	60K	No
7	Yes	Divorced	220K	No
8	No	Single	85K	Yes
9	No	Married	75K	No
10	No	Single	90K	Yes

Objects

Attribute Values

- 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 the same set of values

- Example: Attribute values for ID and age are integers
- But properties of attribute values can be different
  - 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

Evolution of Database Technology

Data mining primitives.

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

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

2. 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

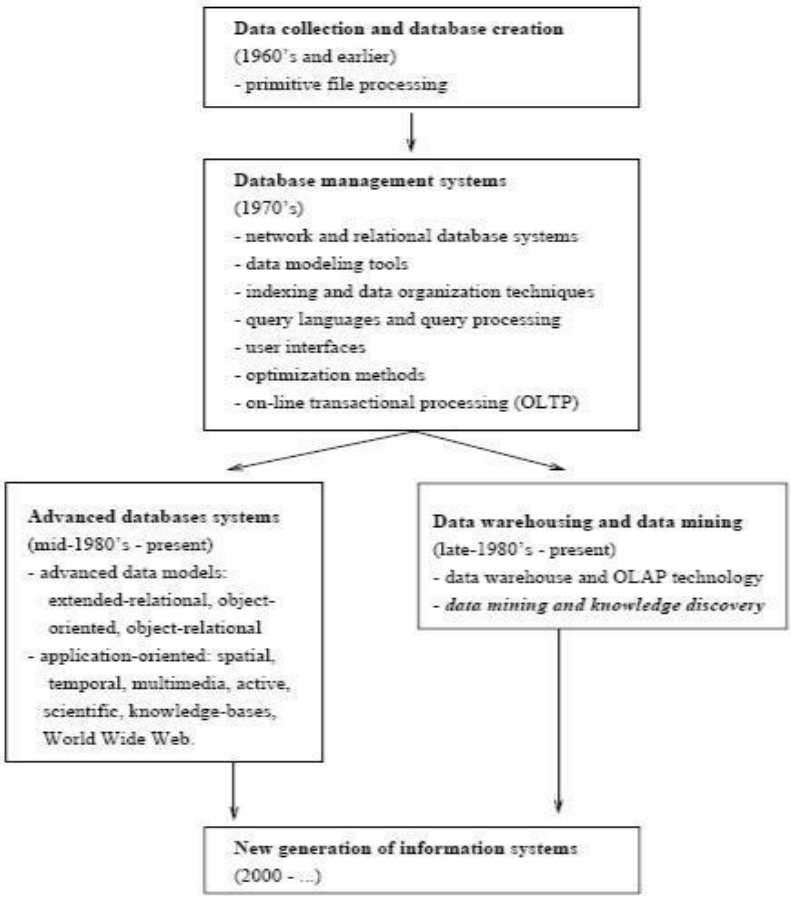


Figure 1.1: The evolution of database technology.

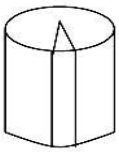
3. Background knowledge: Users can specify background knowledge, or knowledge about the domain to be mined. This knowledge is useful for guiding the knowledge discovery process, and for evaluating the patterns found. There are several kinds of background knowledge.

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

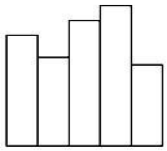
5. Presentation and visualization of discovered patterns: This refers to the form in which discovered patterns are to be displayed. Users can choose from different forms for knowledge presentation, such as rules, tables, charts, graphs, decision trees, and cubes.

Figure : Primitives for specifying a data mining task.

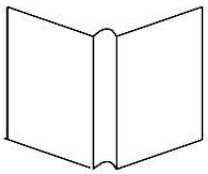
$$Ent(S) - E(T, S) > \delta$$



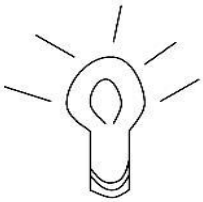
Task-relevant data
- database or data warehouse name
- database tables or data warehouse cubes
- conditions for data selection
- relevant attributes or dimensions
- data grouping criteria



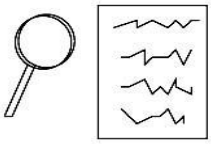
Knowledge type to be mined
- characterization
- discrimination
- association
- classification/prediction
- clustering



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



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



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

Knowledge Discovery in Databases or KDD

Knowledge discovery as a process is depicted and consists of an iterative sequence of the following steps:

- Data cleaning (to remove noise or irrelevant 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;), and
- Knowledge presentation (where visualization and knowledge representation techniques are used to present the mined knowledge to the user).

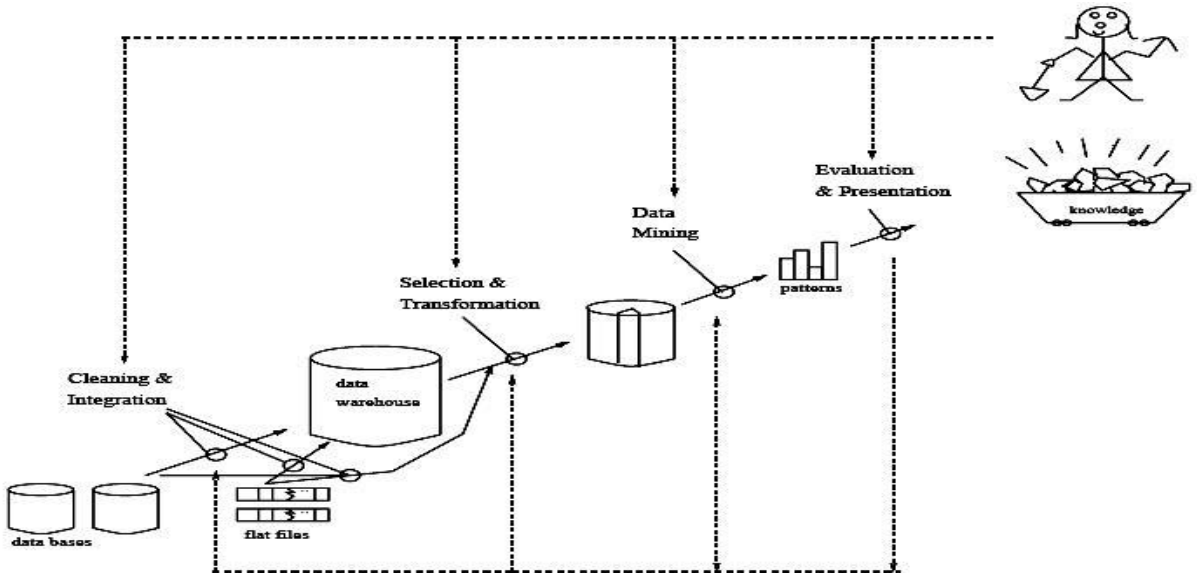


Figure: Data mining as a process of knowledge discovery.

Architecture of a typical data mining system.

The architecture of a typical data mining system may have the following major components

1. Database, data warehouse, or other information repository. This is one or a set of databases, data warehouses, spread sheets, or other kinds of information repositories. Data cleaning and data integration techniques may be performed on the data.

2. 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.

3. 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. Knowledge such as user beliefs, which can be used to assess a pattern's interestingness based on its unexpectedness, may also be included.

4. 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 analysis, classification, evolution and deviation analysis.

5. Pattern evaluation module. This component typically employs interestingness measures and interacts with the data mining modules so as to focus the search towards interesting patterns. It may access interestingness thresholds stored in the knowledge base. Alternatively, the pattern evaluation module may be integrated with the mining module, depending on the implementation of the data mining method used.

6. Graphical 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.

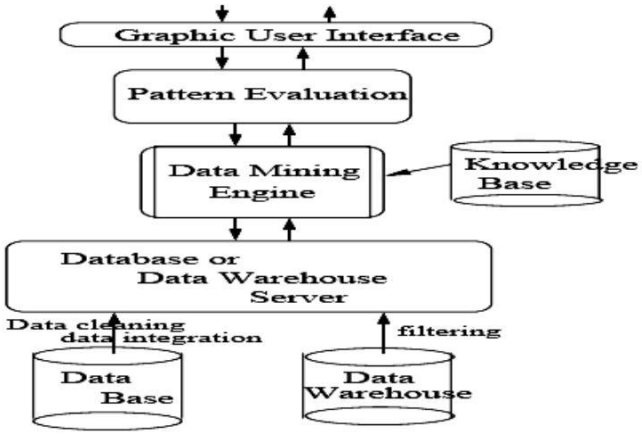


Figure: Architecture of a typical data mining system

Data mining functionalities

Data mining functionalities are used to specify the kind of patterns to be found in data mining tasks. In general, 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. In some cases, users may have no idea of which kinds of patterns in their data may be interesting, and hence may like to search for several different kinds of patterns in parallel. Thus it is important to have a data mining system that can mine multiple kinds of patterns to accommodate di_erent user expectations or applications. Furthermore, data mining systems should be able to discover patterns at various granularities. To encourage interactive and exploratory mining, users should be able to easily \play" with the output patterns, such as by mouse clicking. Operations that can be speci_ed by simple mouse clicks include adding or dropping a dimension (or an attribute), swapping rows and columns (pivoting, or axis rotation), changing dimension representations (e.g., from a 3-D cube to a sequence of 2-D cross tabulations, or crosstabs), or using OLAP roll-up or drill-down operations along

dimensions. Such operations allow data patterns to be expressed from different angles of view and at multiple levels of abstraction.

Data mining systems should also allow users to specify hints to guide or focus the search for interesting patterns. Since some patterns may not hold for all of the data in the database, a measure of certainty or \trustworthiness" is usually associated with each discovered pattern. Data mining functionalities, and the kinds of patterns they can discover, are described below.

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 in summarized, concise, and yet precise terms. Such descriptions of a class or a concept are called class/concept descriptions. These descriptions can be derived via (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 database query. For example, to study the characteristics of software products whose sales increased by 10% in the last year, one can collect the data related to such products by executing an SQL query. There are several methods for e_ective data summarization and characterization. For instance, the data cube- based OLAP roll-up operation can be used to perform user-controlled data summarization along a specified dimension. This process is further detailed in Chapter 2 which discusses data warehousing. An attribute- oriented induction technique can be used to perform data generalization and characterization without step-by-step user interaction. 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 (called characteristic rules).

Association analysis

Association analysis is the discovery of association rules showing attribute-value conditions that occur frequently together in a given set of data. Association analysis is widely used for market basket or transaction data analysis. More formally, association rules are of the form X) Y , i.e., \A1 ^ _ _ _ ^Am !B1 ^ _ _ _^Bn", where Ai (for i 2 f1; : : :;mg) and Bj (for j 2 f1; : : :; ng) are attribute-value pairs. The association rule X) Y is interpreted as \database tuples that satisfy the conditions in X are also likely to satisfy the conditions in Y ".

An association between more than one attribute, or predicate (i.e., age, income, and buys). Adopting 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. Suppose, as a marketing manager of AllElectronics, you would like to determine which items are frequently purchased together within the same transactions. An example of such a rule is

Contains

(T; \computer")) contains(T; \software") [support = 1%; confidence = 50%]

meaning that if a transaction T contains \computer", there is a 50% chance that it contains \software" as well, and 1% of all of the transactions contain both. This association rule involves a single attribute or predicate (i.e., contains) which 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 \computer) software [1%, 50%]".

Classification and prediction

Classification is the processing of finding a set of models (or functions) which describe and distinguish data classes or concepts, for the purposes 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). The derived model may be represented in various forms, such as classi_cation (IF-THEN) rules, decision trees, mathematical formulae, or neural networks. A decision tree is a ow-chart-like tree structure, where each node denotes a test on an attribute value, each branch represents an outcome of the test, and tree leaves represent classes or class distributions. Decision trees can be easily converted to classi_cation rules. A neural network is a collection of linear threshold units that can be trained to distinguish objects of different classes.Classification can be used for predicting the class label of data objects. However, in many applications, one may like to predict some missing or unavailable data values rather than class labels. This is usually the case when the predicted values are numerical data, and is often specifically referred to as prediction. Although prediction may refer to both data value prediction and class label prediction, it is usually con_ned to data value prediction and thus is distinct from classification. Prediction also encompasses the identification of distribution trends based on the available data. Classification and prediction may need to be preceded by relevance analysis which attempts to identify at tributes that do not contribute to the classification or prediction process.

Clustering analysis

Clustering analyzes data objects without consulting a known class label. In general, the class labels are not present in the training data simply because they are not known to begin with. Clustering can be used to generate such labels. The objects are clustered or grouped based on the principle of maximizing the intraclass 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 very dissimilar to objects in other clusters. Each cluster that is formed can be viewed as a class of objects, from which rules can be derived.

Evolution and deviation analysis

Data evolution analysis describes and models regularities or trends for objects whose behavior changes over time. Although this may include characterization, discrimination, association, classi_cation, 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 Patterns

A data mining system has the potential to generate thousands or even millions of patterns, or rules. This raises some serious questions for data mining:

A pattern is interesting if (1) it is 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 con_rm. 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 XU Y is rule support, representing the percentage of data samples that the given rule satisfies. Another objective measure for association rules is confidence, which assesses the degree of certainty of the detected association. It is defined as the conditional probability that a pattern Y is true given that X is true. More formally, support and confidence aredefined as

support(X) Y) = Prob{XUY}g

confidence (X) Y) = Prob{Y |X}g

A classification of data mining systems

Data mining is an interdisciplinary field, the conuence 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, or psychology. Because of the diversity of disciplines contributing to data mining, data mining research is expected to generate

a large variety of data mining systems. Therefore, it is necessary to provide a clear classi_cation of data mining systems. Such a classi_cation may help potential users distinguish data mining systems and identify those that best match their needs. Data mining systems can be categorized according to various criteria, as follows._ Classi_cation according to the kinds of databases mined. A data mining system can be classi_ed according to the kinds of databases mined. Database systems themselves can be classi_ed according to di_erent 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 classi_ed accordingly.

For instance, if classifying according to data models, we may have a relational, transactional, object-oriented, object-relational, or data warehouse mining system. If classifying according to the special types of data handled, we may have a spatial, time-series, text, or multimedia data mining system, or a World-Wide Web mining system. Other system types include heterogeneous data mining systems, and legacy data mining systems.

Classification according to the kinds of knowledge mined. Data mining systems can be categorized according to the kinds of knowledge they mine, i.e., based on data mining functionalities, such as characterization, discrimination, association, classi_cation, clustering, trend and evolution analysis, deviation analysis, similarity analysis, etc. A comprehensive data mining system usually provides multiple and/or integrated data mining functionalities. Moreover, data mining systems can also be distinguished based on the granularity or levels of abstraction of the knowledge mined, including generalized knowledge (at a high level of abstraction), primitive-level knowledge (at a raw data level), or knowledge at multiple levels

(considering several levels of abstraction). An advanced data mining system should facilitate the discovery of knowledge at multiple levels of abstraction.

Classification according to the kinds of knowledge mined.

Data mining systems can be categorized according to the kinds of knowledge they mine, i.e., based on data mining functionalities, such as characterization, discrimination, association, classi_cation, clustering, trend and evolution analysis, deviation analysis, similarity analysis, etc. A comprehensive data mining system usually provides multiple and/or integrated data mining functionalities.Moreover, data mining systems can also be distinguished based on the granularity or levels of abstraction of the knowledge mined, including generalized knowledge (at a high level of abstraction), primitive-level knowledge (at a raw data level), or knowledge at multiple levels (considering several levels of abstraction). An advanced data mining system should facilitate the discovery of knowledge at multiple levels of abstraction.

Classification according to the kinds of techniques utilized

Data mining systems can also be categorized according to the underlying data mining techniques employed. These techniques can be described according to the degree of user interaction involved (e.g., autonomous systems, interactive exploratory systems, query-driven systems), or the methods of data analysis employed (e.g., database-oriented or data warehouse-oriented techniques, machine learning, statistics, visualization, pattern recognition, neural networks, and so on). A sophisticated data mining system will often adopt multiple data mining techniques or work out an e_ective, integrated technique which combines the merits of a few individual approaches.

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 _nding 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 mining algorithms.

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.

DataPreprocessing

Data cleaning.

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

(i). Missing values

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

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

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

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

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

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

using the other customer attributes in your data set, you may construct a decision tree to predict the missing values for income.

(ii). Noisy data

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

1. Binning methods:

Binning methods smooth a sorted data value by consulting the ”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

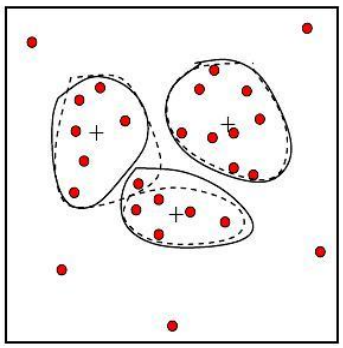
(iv).Smoothing by bin boundaries:

- 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 combination of computer and human inspection. In one application, for example, an information-theoretic measure was used to help identify outlier patterns in a handwritten character database for classification. The measure's value reflected the “surprise" content of the predicted character label with respect to the known label. Outlier patterns may be informative or “garbage". Patterns whose surprise content is above a threshold are output to

a list. A human can then sort through the patterns in the list to identify the actual garbage ones

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.

Data transformation.

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

1. **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 that minA and maxA are the minimum and maximum values of an attribute A. Min-max normalization maps a value v of A to v0 in the range [new minA; new maxA] by computing

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

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

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

(iii). **Normalization by decimal scaling** normalizes by moving the decimal point of values of attribute A. The number of decimal points moved depends on the maximum absolute value of A. A value v of A is normalized to v0by computing where j is the smallest integer such that $Max(|v'|) < 1$.

2. **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

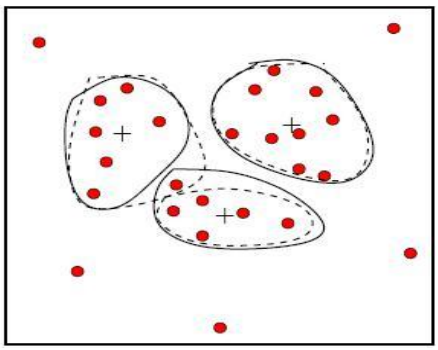
(iv).Smoothing by bin boundaries:

- 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.



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.

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

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.

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

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

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

4. Numerosity reduction, where the data are replaced or estimated by alternative, smaller data representations such as parametric models (which need store only the model

parameters instead of the actual data), or nonparametric methods such as clustering, sampling, and the use of histograms.

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

Data Cube Aggregation

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

Dimensionality Reduction

Feature selection (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:

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

Forward Selection

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

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

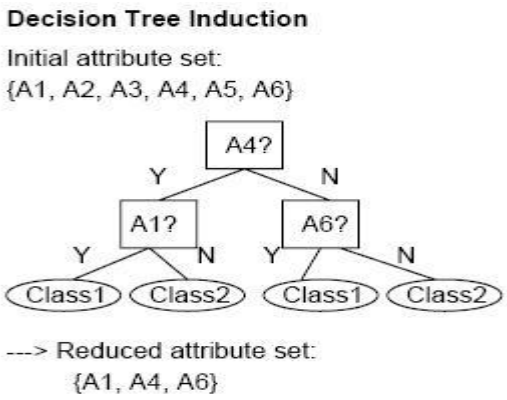
2. Step-wise backward elimination: The procedure starts with the full set of attributes. At each step, it removes the worst attribute remaining in the set.

Backward Elimination

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

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

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



Data compression

In data compression, data encoding or transformations are applied so as to obtain a reduced or ”compressed" representation of the original data. If the original data can be reconstructed from the compressed data without any loss of information, the data compression technique used is called lossless. 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.

1. 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.
2. 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.

3. 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.
4. 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.
5. 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.

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 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.

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.

4. 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 to be modeled as a linear function of a multidimensional feature vector.

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

Histograms

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

1. 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).

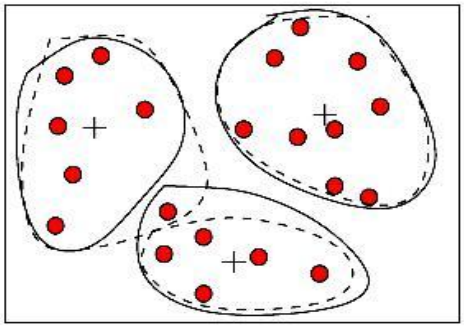
2. Equi-depth (or equi-height): In an equi-depth histogram, the buckets are created so that, roughly, the frequency of each bucket is constant (that is, each bucket contains roughly the same number of contiguous data samples).

3. 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.

4. 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 $\beta-1$ largest differences, where β 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.

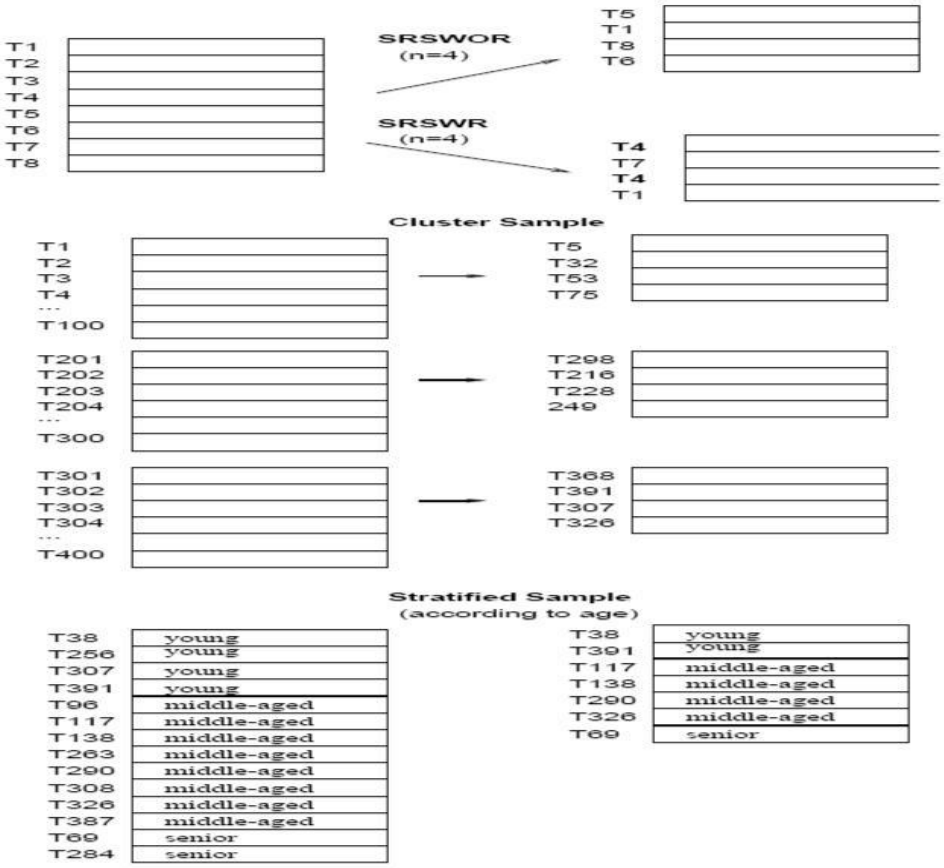
1. Simple random sample without replacement (SRSWOR) of size n: This is created by drawing n of the N tuples from D (n < N), where the probably of drawing any tuple in D is 1=N, i.e., all tuples are equally likely.

2. 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.

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

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

Figure : Sampling can be used for data reduction.



Major Issues in Data Warehousing and Mining

- Mining methodology and user interaction
  - Mining different kinds of knowledge in databases
  - Interactive mining of knowledge at multiple levels of abstraction
  - Incorporation of background knowledge
  - Data mining query languages and ad-hoc data mining
  - Expression and visualization of data mining results
  - Handling noise and incomplete data
  - Pattern evaluation: the interestingness problem
- Performance and scalability
  - Efficiency and scalability of data mining algorithms
  - Parallel, distributed and incremental mining methods

- Issues relating to the diversity of data types
  - Handling relational and complex types of data
  - Mining information from heterogeneous databases and global information systems (WWW)
- Issues related to applications and social impacts
  - Application of discovered knowledge
    - Domain-specific data mining tools