UNIT-1

TOPIC-1

INTRODUCTION

What Is Data Mining?

Data mining is the *process* of discovering interesting patterns, previously unknown 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.

Many other terms have a similar meaning to data mining—for example,

- ***knowledge mining from data,***
- ***knowledge extraction,***
- ***data/pattern analysis,***
- ***data archaeology, and*** data ***dredging.***

Why do we need 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 be viewed as a result of the natural evolution of information technology.

There is a huge amount of data available in the Information Industry. This data is of no use until it is converted into useful information. It is necessary to analyze this huge amount of data and extract useful information from it.

Extraction of information is not the only process we need to perform; data mining also involves other processes such as Data Cleaning, Data Integration, Data Transformation, Data Mining, Pattern Evaluation and Data Presentation. Once all these processes are over, we would be able to use this information in many applications such as Fraud Detection, Market Analysis, Production Control, Science Exploration, etc.

“We are living in the information age” is a popular saying; however, we are actually living in the data age. Terabytes or petabytes of data pour into our computer networks, the World Wide Web, and various data storage devices every day from business,

Consider a search engine (e.g., Google) receives hundreds of millions of queries every day. Each query can be viewed as a transaction where the user describes her/his information need. some patterns found in user search queries can disclose invaluable knowledge that cannot be obtained by reading individual data items alone. say, Google’s *Flu Trends* uses specific search terms as indicators of flu activity. It found a close relationship between the number of people who search for flu related information and the number of people who actually have flu symptoms. A pattern emerges when all of the search queries related to flu are aggregated. Using aggregated Google search data, *Flu Trends* can estimate flu activity up to two weeks faster than traditional systems can. This example shows how data mining can turn a large collection of data into knowledge that can help meet a current global challenge.

Data mining plays a crucial role in various domains, including cybersecurity, e-commerce, analyzing traffic patterns, monitoring Twitter data, managing sensor networks, and conducting computer simulations.

Cybersecurity:

Data mining helps in identifying patterns of malicious activities, such as cyber attacks, malware infections, and unauthorized access attempts.

- It can detect anomalies in network traffic, user behavior, or system activities that might indicate potential security breaches.
- By analyzing historical data, data mining techniques can predict potential security threats and help in developing proactive security measures.
- It assists in identifying trends and patterns in cyber threats, which can aid in improving intrusion detection and prevention systems.

E-commerce:

In e-commerce, data mining is used for market basket analysis to understand customer buying behavior and preferences.

- It helps in identifying cross-selling and upselling opportunities by analyzing purchase patterns and customer segmentation.
- Data mining techniques enable personalized recommendations based on past purchases, browsing history, and demographic information.
- Fraud detection in online transactions is another crucial application where data mining helps identify suspicious activities and fraudulent transactions.

Traffic Patterns:

Data mining is utilized to analyze traffic patterns in transportation systems, such as roads, railways, and airways.

- It helps in optimizing traffic flow, reducing congestion, and improving transportation efficiency.
- By analyzing historical traffic data, data mining techniques can predict future traffic patterns and assist in urban planning and infrastructure development.

Twitter:

- Data mining of Twitter data enables sentiment analysis, trend detection, and opinion mining.
- It helps in understanding public opinion, identifying emerging trends, and monitoring brand reputation.
- By analyzing Twitter data, organizations can gain insights into customer preferences, market trends, and competitor activities.

Sensor Networks:

In sensor networks, data mining is used to analyze sensor data streams generated by various IoT devices.

- It helps in detecting patterns, anomalies, and events in sensor data, such as environmental monitoring, industrial processes, and smart grid systems.
- Data mining techniques enable predictive maintenance by identifying equipment failures or performance degradation based on sensor data patterns.

Computer Simulations:

- Data mining assists in analyzing simulation output data to extract meaningful insights and identify significant trends or patterns.
- It helps in validating simulation models, calibrating parameters, and improving simulation accuracy.
- By mining simulation data, researchers can discover new knowledge, optimize processes, and make informed decisions in various domains, such as healthcare, manufacturing, and scientific research.

UNIT-1

TOPIC-3

What Kinds of Data Can Be Mined?

As a general technology, 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

1. Database data
2. Data warehouse data and
3. transactional data

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

Example: consider a relational database for ***AllElectronics***. The company is described by the **following relation tables: customer, item, employee, and branch**.

The relation **customer** consists of a set of attributes describing the customer information, including a unique customer identity number (cust ID), customer name, address, age, occupation, annual income, credit information, and category.

Similarly, each of the relations item, employee, and branch consists of a set of attributes describing the properties of these entities.

Tables can also be used to represent the relationships between or among multiple entities. In our example, these include *purchases* (customer purchases items, creating a sales transaction handled by an employee), *items sold* (lists items sold in a given transaction), and *works at* (employee works at a branch of *AllElectronics*).

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)
item_ 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 query allows retrieval of specified subsets of the data.

Relational languages also use aggregate functions such as sum, avg (average), count, max (maximum), and min (minimum). Using aggregates allows you to ask: *"**Show me the total sales of the last month, grouped by branch,"*** **or** ***"How many sales transactions occurred in the month of December?"*** **or** ***"Which salesperson had the highest sales?"***

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 mining systems may also detect deviations—that is, **items with sales that are far from those expected in comparison with the previous year**. Such

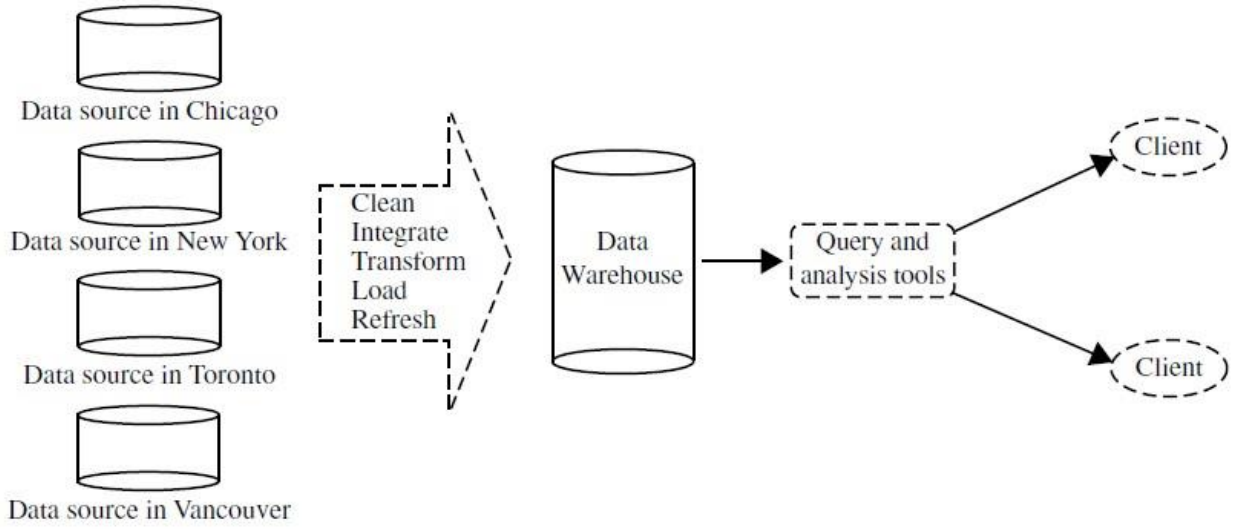
deviations can then be further investigated. For example, data mining may discover that there has been a change in packaging of an item or a significant increase in price.

Relational databases are one of the most commonly available and richest information repositories, and thus they are a major data form in the study of data mining

Data Warehouse Data

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.

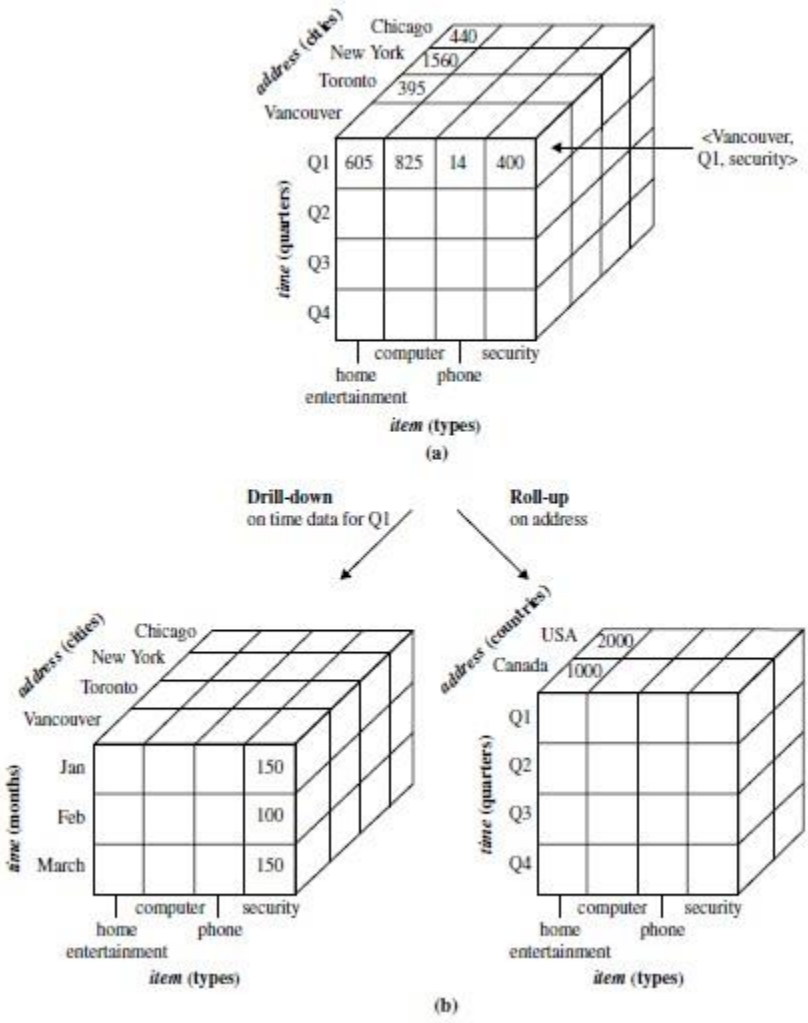
To facilitate decision making, the data in a data warehouse are organized around *major subjects* (e.g., customer, item, supplier, and activity). The data are stored to provide information from a *historical perspective*, such as in the past 6 to 12 months, and are typically *summarized*. For example, rather than storing the details of each sales transaction, 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.



Typical framework of a data warehouse for *AllElectronics*.

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.



A multidimensional data cube, commonly used for data warehousing, (a) showing summarized data for *AllElectronics* and (b) showing summarized data resulting fromdrill-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.

Example transaction list of Items : which items are sold together – frequent itemsets/ market basket analysis.

trans ID	*list of item IDs*
T100	I1, I3, I8, I16
T200	I2, I8
….	….

Other Kinds of Data

Besides relational database data, data warehouse data, and transaction data, there are many other kinds of data that have versatile forms and structures and rather different semantic meanings. Such 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.

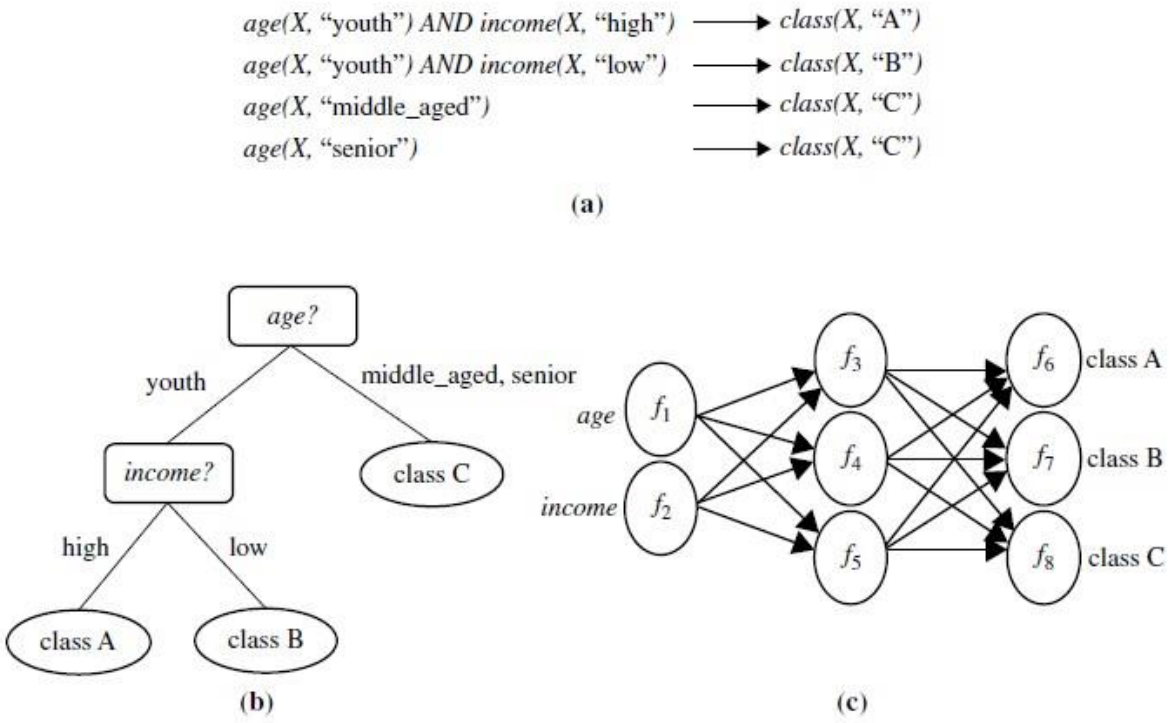
UNIT-1

TOPIC 4.2

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

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.
"How is the derived model presented?" 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 classification model can be represented in various forms: (a) IF-THEN rules, (b) a decision tree, or (c) a neural network.

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. Decision trees can easily be converted to classification rules.

A **neural network**, when used for classification, is typically a collection of neuron-like processing units with weighted connections between the units. There are many other

methods for constructing classification models, such as na¨ıve Bayesian classification, support vector machines, and *k*-nearest-neighbor classification.

Whereas classification predicts categorical (discrete, unordered) labels, **regression** models continuous-valued functions. That is, regression is used to predict missing or unavailable *numerical data values* rather than (discrete) class labels. The term *prediction* refers to both numeric prediction and class label prediction. **Regression analysis** is a statistical methodology that is most often used for numeric prediction, although other methods exist as well. Regression also encompasses the identification of distribution *trends* based on the available data.

Example Classification and regression. Suppose as a sales manager of *AllElectronics* you want to classify a large set of items in the store, based on three kinds of responses to a sales campaign: *good response*, *mild response* and *no response*. You want to derive a model for each of these three classes based on the descriptive features of the items, such as *price*, *brand, place made, type*, and *category*. The resulting classification should maximally distinguish each class from the others, presenting an organized picture of the data set. to predict the amount of revenue that each item will generate during an upcoming sale at *AllElectronics*, based on the previous sales data is an example of regression analysis because the regression model constructed will predict a continuous function (or ordered value.)

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. However, in some applications (e.g., fraud detection) the rare events can be more interesting than the more regularly occurring ones. The analysis of outlier data is referred to as **outlier 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. Outlier values may also be detected with respect to the locations and types of purchase, or the purchase frequency.

Evolution Analysis

Evolution Analysis refers to the study of data sets that may have been through a phase of transformation or change. The evolution analysis models capture evolutionary trends in data, which further contributes to data characterization, classification, or discrimination and clustering for multivariate time series.

Are All Patterns Interesting?

A data mining system has the potential to generate thousands or even millions of patterns, or rules.

"Are all of the patterns interesting?" Typically, the answer is no—only a small fraction of the patterns potentially generated would actually be of interest to a given user. *"What makes a pattern interesting? Can a data mining system generate all of the interesting patterns? Or, Can the system generate only the interesting ones?"*

1) Is 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**.

Several **objective measures of pattern interestingness** exist. These are based on the structure of discovered patterns and the statistics underlying them. (support, confidence, accuracy, coverage)

Subjective interestingness measures are based on user beliefs in the data. These measures find patterns interesting if the patterns are **unexpected or expected.**

2) "*Can a data mining system generate* all *of the interesting patterns?*"— refers to the **completeness** of a data mining algorithm. It is often unrealistic and inefficient for data mining systems to generate all possible patterns. Instead, userprovided constraints and interestingness measures should be used to focus the search.

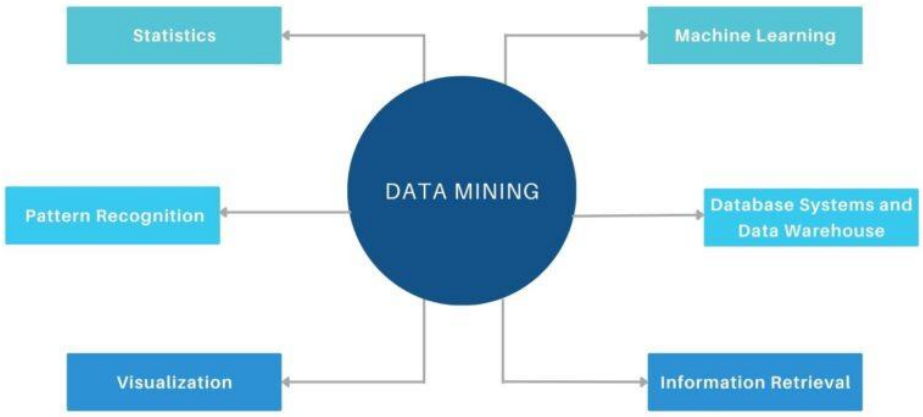
3)*"Can a data mining system generate only interesting patterns?"*— is an optimization problem in data mining. It is highly desirable for data mining systems to generate only interesting patterns.

UNIT-1

TOPIC-5

WHICH TECHNOLOGIES ARE USED?

Data mining has incorporated many techniques from other domain fields like machine learning, statistics, information retrieval, data warehouse, pattern recognition, algorithms, and high-performance computing. Since it is a highly application-driven domain, the interdisciplinary nature is typically very significant. Research and development in data mining and its applications prove quite useful in implementing it. The following are the major technologies utilized in data mining.



I. Statistics:

- Statistics studies the collection, analysis, interpretation or explanation, and presentation of data.
- Data mining has an inherent connection with statistics.

- Statistics is a component of data mining that provides the tools and analytics techniques for dealing with large amounts of data.
- Statistics is useful for mining various patterns from data as well as for understanding the mechanisms which are generating and affecting the patterns.

II. Machine Learning:

- Machine learning investigates how computers can learn or improve their performance based on data.
- Data mining uses techniques developed by machine learning for predicting the outcome.
- For example, a typical machine learning problem is to program a computer so that it can automatically recognize handwritten code on mail after learning from a set of examples.
- Supervised learning is basically a synonym for classification.
- Unsupervised learning is essentially a synonym for clustering.
- Semi-supervised learning is a class of machine learning techniques that make use of both labelled and unlabeled examples when learning a model.
- Active learning is a machine learning approach that lets users play an active role in the learning process.

III. Database Systems and Data Warehouse:

- Database systems research focuses on the creation, maintenance, and use of databases for organizations and end-users.
- Database System is used in traditional way of storing and retrieving data.
- The major task of database system is to perform query processing.
- Data Warehouse is the place where huge amount of data is stored.

IV. Information Retrieval:

- Information retrieval is the process of searching for documents or information in the documents.
- Documents can be text, multimedia and many other formats stores on a web.

- In information retrieval the data under search are unstructured.
- The queries are formed mainly by keywords, which do not have complex structure (unlike SQL queries in database systems).

V. Visualization:

- Visualization of data mining results is the presentation of the results or knowledge obtained from data mining in visual forms.
- Data visualization is the graphical representation of information and data in a pictorial or graphical format (Example: charts, graphs, and maps).
- Data visualization tools provide an accessible way to see and understand trends, patterns in data, and outliers.
- Data visualization tools and technologies are essential to analyzing massive amounts of information and making data-driven decisions.

VI. Pattern Recognition:

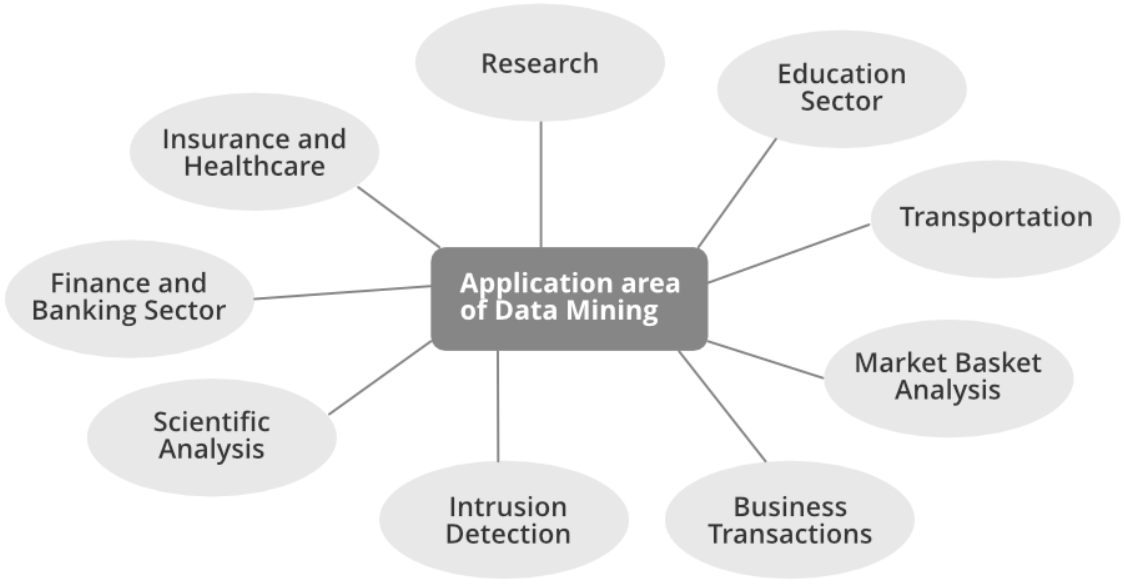
- Pattern is everything around in this digital world.
- A pattern can either be seen physically or it can be observed mathematically by applying algorithms.
- Pattern recognition is the process of recognizing patterns by using a machine learning algorithm.
- Pattern recognition can be defined as the classification of data based on knowledge already gained or on statistical information extracted from patterns and/or their representation.

What kind of applications are targeted?

Data mining is the computational process of analyzing data from different perspectives, dimensions, angles and categorizing/summarizing it into meaningful information. Data Mining can be applied to any type of data e.g. Data Warehouses, Transactional Databases, Relational Databases, Multimedia Databases, Spatial Databases, Time-series Databases, World Wide Web.

Data mining provides competitive advantages in the knowledge economy. It does this by providing the maximum knowledge needed to rapidly make valuable business decisions despite the enormous amounts of available data.

There are many measurable benefits that have been achieved in different application areas from data mining. So, let's discuss different applications of Data Mining:



Scientific Analysis: Scientific simulations are generating bulks of data every day. This includes data collected from nuclear laboratories, data about human psychology, etc. Data mining techniques are capable of the analysis of these data. Now we can capture and store more new data faster than we can analyze the old data already accumulated. Example of scientific analysis:

- Sequence analysis in bioinformatics
- Classification of astronomical objects
- Medical decision support.

Intrusion Detection: A network intrusion refers to any unauthorized activity on a digital network. Network intrusions often involve stealing valuable network resources. Data mining technique plays a vital role in searching intrusion detection, network attacks, and anomalies. These techniques help in selecting and refining useful and relevant information from large data sets. Data mining technique helps in classify relevant data for Intrusion Detection

System. Intrusion Detection system generates alarms for the network traffic about the foreign invasions in the system. For example:

- Detect security violations
- Misuse Detection
- Anomaly Detection

Business Transactions: Every business industry is memorized for perpetuity. Such transactions are usually time-related and can be inter-business deals or intra-business operations. The effective and in-time use of the data in a reasonable time frame for competitive decision-making is definitely the most important problem to solve for businesses that struggle to survive in a highly competitive world. Data mining helps to analyze these business transactions and identify marketing approaches and decision-making. Example :

- Direct mail targeting
- Stock trading
- Customer segmentation
- Churn prediction (Churn prediction is one of the most popular Big Data use cases in business)

Market Basket Analysis: Market Basket Analysis is a technique that gives the careful study of purchases done by a customer in a supermarket. This concept identifies the pattern of frequent purchase items by customers. This analysis can help to promote deals, offers, sale by the companies and data mining techniques helps to achieve this analysis task. Example:

- Data mining concepts are in use for Sales and marketing to provide better customer service, to improve cross-selling opportunities, to increase direct mail response rates.
- Customer Retention in the form of pattern identification and prediction of likely defections is possible by Data mining.
- Risk Assessment and Fraud area also use the data-mining concept for identifying inappropriate or unusual behavior etc.

Education: For analyzing the education sector, data mining uses Educational Data Mining (EDM) method. This method generates patterns that can be used both by learners and educators. By using data mining EDM we can perform some educational task:

- Predicting students admission in higher education

- Predicting students profiling
- Predicting student performance
- Teachers teaching performance
- Curriculum development
- Predicting student placement opportunities

Research: A data mining technique can perform predictions, classification, clustering, associations, and grouping of data with perfection in the research area. Rules generated by data mining are unique to find results. In most of the technical research in data mining, we create a training model and testing model. The training/testing model is a strategy to measure the precision of the proposed model. It is called Train/Test because we split the data set into two sets: a training data set and a testing data set. A training data set used to design the training model whereas testing data set is used in the testing model. Example:

- Classification of uncertain data.
- Information-based clustering.
- Decision support system
- Web Mining
- Domain-driven data mining
- IoT (Internet of Things)and Cybersecurity
- Smart farming IoT(Internet of Things)

Healthcare and Insurance: A Pharmaceutical sector can examine its new deals force activity and their outcomes to improve the focusing of high-value physicians and figure out which promoting activities will have the best effect in the following upcoming months, Whereas the Insurance sector, data mining can help to predict which customers will buy new policies, identify behavior patterns of risky customers and identify fraudulent behavior of customers.

- Claims analysis i.e which medical procedures are claimed together.
- Identify successful medical therapies for different illnesses.
- Characterizes patient behavior to predict office visits.

Transportation: A diversified transportation company with a large direct sales force can apply data mining to identify the best prospects for its services. A large consumer merchandise organization can apply information mining to improve its business cycle to retailers.

- Determine the distribution schedules among outlets.

- Analyze loading patterns.

Financial/Banking Sector: A credit card company can leverage its vast warehouse of customer transaction data to identify customers most likely to be interested in a new credit product.

- Credit card fraud detection.
- Identify 'Loyal' customers.
- Extraction of information related to customers.
- Determine credit card spending by customer groups.

Major Issues In Data Mining:

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

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

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

Data mining query languages and ad hoc data mining. - Data Mining Query language that allows the user to describe ad hoc mining tasks, should be integrated with a data warehouse query language and optimized for efficient and flexible data mining.

Presentation and visualization of data mining results. - Once the patterns are discovered it needs to be expressed in high level languages, visual representations. This representations should be easily understandable by the users.

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

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

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

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

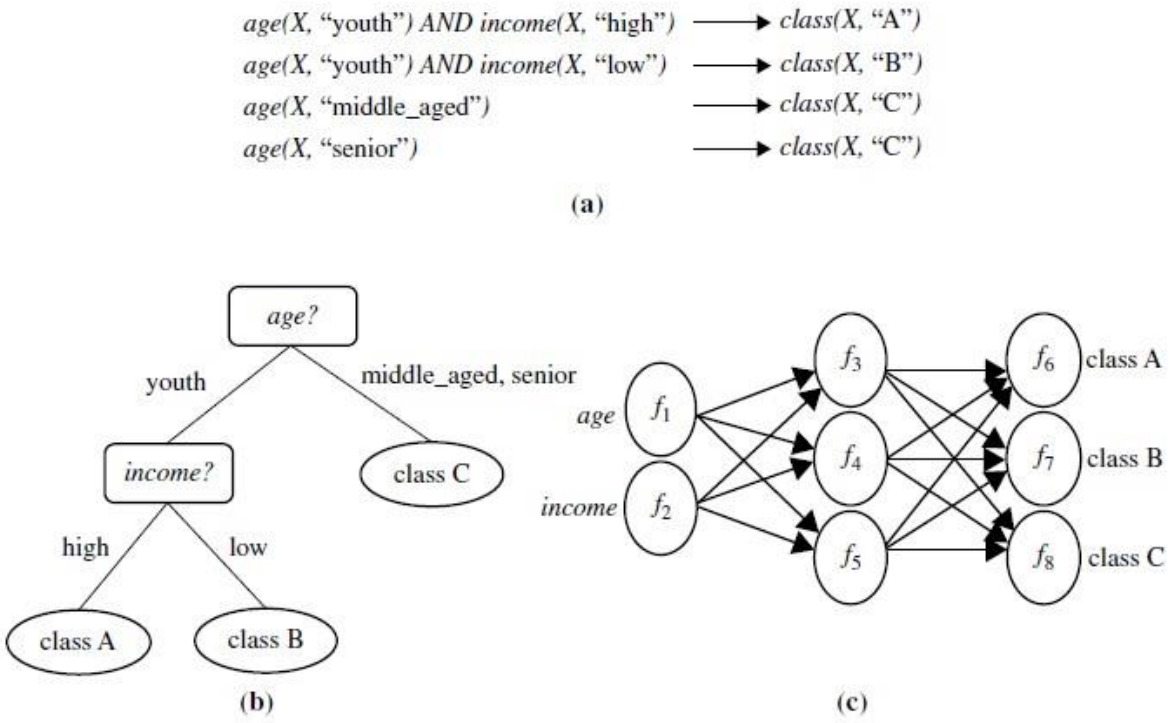
UNIT-1

TOPIC 4.2

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

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.
"How is the derived model presented?" 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 classification model can be represented in various forms: (a) IF-THEN rules, (b) a decision tree, or (c) a neural network.

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. Decision trees can easily be converted to classification rules.

A **neural network**, when used for classification, is typically a collection of neuron-like processing units with weighted connections between the units. There are many other

methods for constructing classification models, such as na¨ıve Bayesian classification, support vector machines, and *k*-nearest-neighbor classification.

Whereas classification predicts categorical (discrete, unordered) labels, **regression** models continuous-valued functions. That is, regression is used to predict missing or unavailable *numerical data values* rather than (discrete) class labels. The term *prediction* refers to both numeric prediction and class label prediction. **Regression analysis** is a statistical methodology that is most often used for numeric prediction, although other methods exist as well. Regression also encompasses the identification of distribution *trends* based on the available data.

Example Classification and regression. Suppose as a sales manager of *AllElectronics* you want to classify a large set of items in the store, based on three kinds of responses to a sales campaign: *good response*, *mild response* and *no response*. You want to derive a model for each of these three classes based on the descriptive features of the items, such as *price*, *brand, place made, type*, and *category*. The resulting classification should maximally distinguish each class from the others, presenting an organized picture of the data set. to predict the amount of revenue that each item will generate during an upcoming sale at *AllElectronics*, based on the previous sales data is an example of regression analysis because the regression model constructed will predict a continuous function (or ordered value.)

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. However, in some applications (e.g., fraud detection) the rare events can be more interesting than the more regularly occurring ones. The analysis of outlier data is referred to as **outlier 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. Outlier values may also be detected with respect to the locations and types of purchase, or the purchase frequency.

Evolution Analysis

Evolution Analysis refers to the study of data sets that may have been through a phase of transformation or change. The evolution analysis models capture evolutionary trends in data, which further contributes to data characterization, classification, or discrimination and clustering for multivariate time series.

Are All Patterns Interesting?

A data mining system has the potential to generate thousands or even millions of patterns, or rules.

"Are all of the patterns interesting?" Typically, the answer is no—only a small fraction of the patterns potentially generated would actually be of interest to a given user. *"What makes a pattern interesting? Can a data mining system generate all of the interesting patterns? Or, Can the system generate only the interesting ones?"*

1) Is 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**.

Several **objective measures of pattern interestingness** exist. These are based on the structure of discovered patterns and the statistics underlying them. (support, confidence, accuracy, coverage)

Subjective interestingness measures are based on user beliefs in the data. These measures find patterns interesting if the patterns are **unexpected or expected.**

2) "*Can a data mining system generate* all *of the interesting patterns?*"— refers to the **completeness** of a data mining algorithm. It is often unrealistic and inefficient for data mining systems to generate all possible patterns. Instead, userprovided constraints and interestingness measures should be used to focus the search.

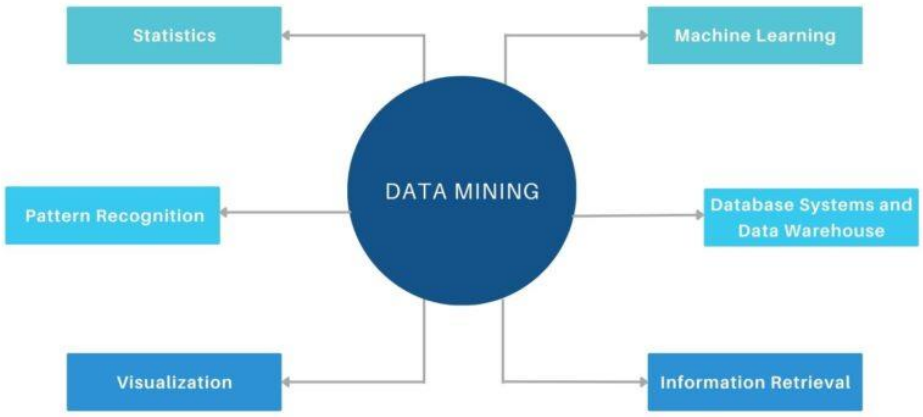
3)*"Can a data mining system generate only interesting patterns?"*— is an optimization problem in data mining. It is highly desirable for data mining systems to generate only interesting patterns.

UNIT-1

TOPIC-5

WHICH TECHNOLOGIES ARE USED?

Data mining has incorporated many techniques from other domain fields like machine learning, statistics, information retrieval, data warehouse, pattern recognition, algorithms, and high-performance computing. Since it is a highly application-driven domain, the interdisciplinary nature is typically very significant. Research and development in data mining and its applications prove quite useful in implementing it. The following are the major technologies utilized in data mining.



I. Statistics:

- Statistics studies the collection, analysis, interpretation or explanation, and presentation of data.
- Data mining has an inherent connection with statistics.

- Statistics is a component of data mining that provides the tools and analytics techniques for dealing with large amounts of data.
- Statistics is useful for mining various patterns from data as well as for understanding the mechanisms which are generating and affecting the patterns.

II. Machine Learning:

- Machine learning investigates how computers can learn or improve their performance based on data.
- Data mining uses techniques developed by machine learning for predicting the outcome.
- For example, a typical machine learning problem is to program a computer so that it can automatically recognize handwritten code on mail after learning from a set of examples.
- Supervised learning is basically a synonym for classification.
- Unsupervised learning is essentially a synonym for clustering.
- Semi-supervised learning is a class of machine learning techniques that make use of both labelled and unlabeled examples when learning a model.
- Active learning is a machine learning approach that lets users play an active role in the learning process.

III. Database Systems and Data Warehouse:

- Database systems research focuses on the creation, maintenance, and use of databases for organizations and end-users.
- Database System is used in traditional way of storing and retrieving data.
- The major task of database system is to perform query processing.
- Data Warehouse is the place where huge amount of data is stored.

IV. Information Retrieval:

- Information retrieval is the process of searching for documents or information in the documents.
- Documents can be text, multimedia and many other formats stores on a web.

- In information retrieval the data under search are unstructured.
- The queries are formed mainly by keywords, which do not have complex structure (unlike SQL queries in database systems).

V. Visualization:

- Visualization of data mining results is the presentation of the results or knowledge obtained from data mining in visual forms.
- Data visualization is the graphical representation of information and data in a pictorial or graphical format (Example: charts, graphs, and maps).
- Data visualization tools provide an accessible way to see and understand trends, patterns in data, and outliers.
- Data visualization tools and technologies are essential to analyzing massive amounts of information and making data-driven decisions.

VI. Pattern Recognition:

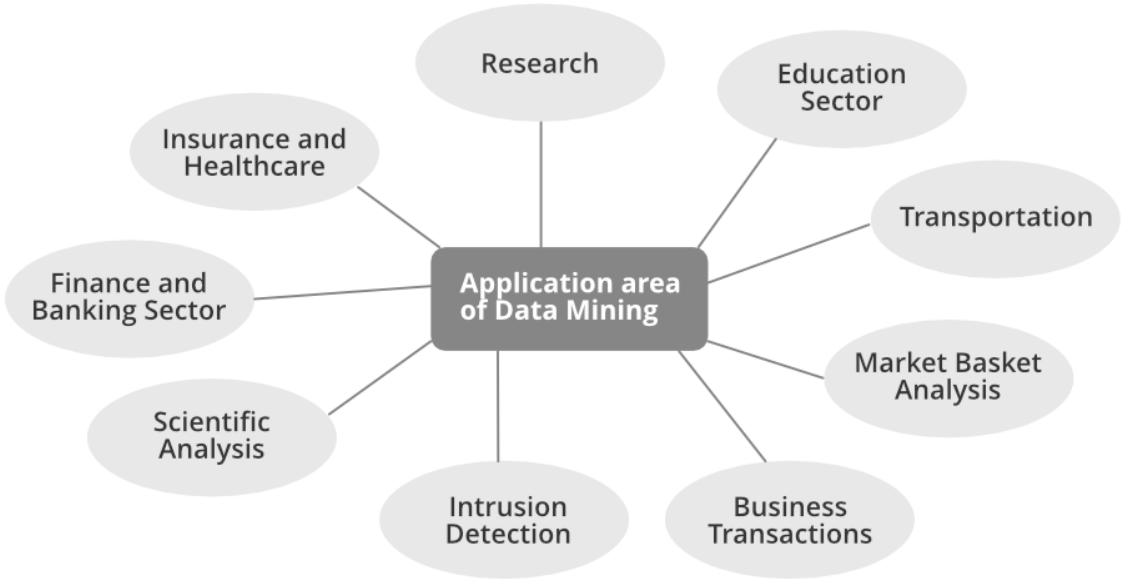
- Pattern is everything around in this digital world.
- A pattern can either be seen physically or it can be observed mathematically by applying algorithms.
- Pattern recognition is the process of recognizing patterns by using a machine learning algorithm.
- Pattern recognition can be defined as the classification of data based on knowledge already gained or on statistical information extracted from patterns and/or their representation.

What kind of applications are targeted?

Data mining is the computational process of analyzing data from different perspectives, dimensions, angles and categorizing/summarizing it into meaningful information. Data Mining can be applied to any type of data e.g. Data Warehouses, Transactional Databases, Relational Databases, Multimedia Databases, Spatial Databases, Time-series Databases, World Wide Web.

Data mining provides competitive advantages in the knowledge economy. It does this by providing the maximum knowledge needed to rapidly make valuable business decisions despite the enormous amounts of available data.

There are many measurable benefits that have been achieved in different application areas from data mining. So, let's discuss different applications of Data Mining:



Scientific Analysis: Scientific simulations are generating bulks of data every day. This includes data collected from nuclear laboratories, data about human psychology, etc. Data mining techniques are capable of the analysis of these data. Now we can capture and store more new data faster than we can analyze the old data already accumulated. Example of scientific analysis:

- Sequence analysis in bioinformatics
- Classification of astronomical objects
- Medical decision support.

Intrusion Detection: A network intrusion refers to any unauthorized activity on a digital network. Network intrusions often involve stealing valuable network resources. Data mining technique plays a vital role in searching intrusion detection, network attacks, and anomalies. These techniques help in selecting and refining useful and relevant information from large data sets. Data mining technique helps in classify relevant data for Intrusion Detection

System. Intrusion Detection system generates alarms for the network traffic about the foreign invasions in the system. For example:

- Detect security violations
- Misuse Detection
- Anomaly Detection

Business Transactions: Every business industry is memorized for perpetuity. Such transactions are usually time-related and can be inter-business deals or intra-business operations. The effective and in-time use of the data in a reasonable time frame for competitive decision-making is definitely the most important problem to solve for businesses that struggle to survive in a highly competitive world. Data mining helps to analyze these business transactions and identify marketing approaches and decision-making. Example :

- Direct mail targeting
- Stock trading
- Customer segmentation
- Churn prediction (Churn prediction is one of the most popular Big Data use cases in business)

Market Basket Analysis: Market Basket Analysis is a technique that gives the careful study of purchases done by a customer in a supermarket. This concept identifies the pattern of frequent purchase items by customers. This analysis can help to promote deals, offers, sale by the companies and data mining techniques helps to achieve this analysis task. Example:

- Data mining concepts are in use for Sales and marketing to provide better customer service, to improve cross-selling opportunities, to increase direct mail response rates.
- Customer Retention in the form of pattern identification and prediction of likely defections is possible by Data mining.
- Risk Assessment and Fraud area also use the data-mining concept for identifying inappropriate or unusual behavior etc.

Education: For analyzing the education sector, data mining uses Educational Data Mining (EDM) method. This method generates patterns that can be used both by learners and educators. By using data mining EDM we can perform some educational task:

- Predicting students admission in higher education

- Predicting students profiling
- Predicting student performance
- Teachers teaching performance
- Curriculum development
- Predicting student placement opportunities

Research: A data mining technique can perform predictions, classification, clustering, associations, and grouping of data with perfection in the research area. Rules generated by data mining are unique to find results. In most of the technical research in data mining, we create a training model and testing model. The training/testing model is a strategy to measure the precision of the proposed model. It is called Train/Test because we split the data set into two sets: a training data set and a testing data set. A training data set used to design the training model whereas testing data set is used in the testing model. Example:

- Classification of uncertain data.
- Information-based clustering.
- Decision support system
- Web Mining
- Domain-driven data mining
- IoT (Internet of Things)and Cybersecurity
- Smart farming IoT(Internet of Things)

Healthcare and Insurance: A Pharmaceutical sector can examine its new deals force activity and their outcomes to improve the focusing of high-value physicians and figure out which promoting activities will have the best effect in the following upcoming months, Whereas the Insurance sector, data mining can help to predict which customers will buy new policies, identify behavior patterns of risky customers and identify fraudulent behavior of customers.

- Claims analysis i.e which medical procedures are claimed together.
- Identify successful medical therapies for different illnesses.
- Characterizes patient behavior to predict office visits.

Transportation: A diversified transportation company with a large direct sales force can apply data mining to identify the best prospects for its services. A large consumer merchandise organization can apply information mining to improve its business cycle to retailers.

- Determine the distribution schedules among outlets.

- Analyze loading patterns.

Financial/Banking Sector: A credit card company can leverage its vast warehouse of customer transaction data to identify customers most likely to be interested in a new credit product.

- Credit card fraud detection.
- Identify 'Loyal' customers.
- Extraction of information related to customers.
- Determine credit card spending by customer groups.

Major Issues In Data Mining:

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

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

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

Data mining query languages and ad hoc data mining. - Data Mining Query language that allows the user to describe ad hoc mining tasks, should be integrated with a data warehouse query language and optimized for efficient and flexible data mining.

Presentation and visualization of data mining results. - Once the patterns are discovered it needs to be expressed in high level languages, visual representations. This representations should be easily understandable by the users.

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

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

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

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

UNIT-1

Types of Data

Data Objects and Attribute Types

Data sets are made up of data objects.

A **data object** represents an entity . The objects may be customers, store items, and sales in a sale database; in a medical database, the objects may be patients in a university database, the objects may be students, professors, and courses. Data objects are typically described by attributes.

Attribute

An **attribute** is a data field, representing a characteristic or feature of a data object. The term *dimension* is commonly used in data warehousing. Data mining and database commonly use the term *attribute*.

Attributes describing a customer object can include, for example, *customer ID*, *name*, and *address*. Observed values for a given attribute are known as *observations*. A set of attributes used to describe a given object is called an *attribute vector* (or *feature vector*). The distribution of data involving one attribute (or variable) is called *univariate*. A *bivariate* distribution involves two attributes, and so on.

The **type** of an attribute is determined by the set of possible values—**nominal, binary, ordinal, or numeric**—the attribute can have.

1. Nominal Attributes

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

Example Nominal attributes. The attribute *hair color* have possible values are *black*, *brown*, *red*, *auburn*, *gray*, and *white*. The attribute *marital status* can take on the values *single, married, divorced*, and *widowed*. Both *hair color* and *marital status* are nominal attributes

2. Binary Attributes

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

Example Binary attributes. The attribute *smoker* describing a *patient* object, 1 indicates that the patient smokes, while 0 indicates that the patient does not. Similarly, suppose the patient undergoes a medical test that has two possible outcomes. The attribute *medical test*

is binary, where a value of 1 means the result of the test for the patient is positive, while 0 means the result is negative.
A binary attribute is **symmetric** if both of its states are equally valuable and carry the same weight eg: gender (male,female), and is **asymmetric** if the outcomes of the states are not equally important, such as the *positive* and *negative* outcomes of a medical test for HIV.

3. Ordinal Attributes

An **ordinal attribute** is an attribute with possible values that have a meaningful order or *ranking* among them, but the magnitude between successive values is not known.
Example Ordinal attributes. Suppose that *drink size* corresponds to the size of drinks available at a fast-food restaurant. This nominal attribute has three possible values: *small, medium*, and *large*. The values have a meaningful sequence (which corresponds to increasing drink size) *grade* (e.g., *A+, A, A-, B+*,and so on) and *professional rank(assistant*, *associate*, and *full* for professors) Customer satisfaction had the following ordinal categories: *0: very dissatisfied, 1: somewhat dissatisfied, 2: neutral, 3: satisfied*, and *4: very satisfied. N*ominal, binary, and ordinal attributes are *qualitative*. That is, they *describe* a feature of an object without giving an actual size or quantity.

4. Numeric Attributes

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

a. Interval-Scaled Attributes

Interval-scaled attributes are measured on a scale of equal-size units. The values of intervalscaled attributes have order and can be positive, 0, or negative. Thus, in addition to providing a ranking of values, such attributes allow us to compare and quantify the *difference* between values.
Example Interval-scaled attributes. A *temperature* attribute is interval-scaled. Suppose that we have the outdoor *temperature* value for a number of different days, where each day is an object. By ordering the values, we obtain a ranking of the objects with respect to *temperature*.
20ºC is five degrees higher than a temperature of 15ºC.Calendar dates are another example. For instance, the years 2002 and 2010 are eight years apart. Temperatures in Celsius and Fahrenheit do not have a true zero-point, that is,
neither 0_C nor 0_F indicates “no temperature.”

b. Ratio-Scaled Attributes

A **ratio-scaled attribute** is a numeric attribute with an inherent zero-point. That is, if a measurement is ratio-scaled, we can speak of a value as being a multiple (or ratio) of another value. In addition, the values are ordered, and we can also compute the difference between values, as well as the mean, median, and mode.
Example Ratio-scaled attributes. Attributes include *count* attributes such as *years of experience* (e.g., the objects are employees) and *number of words* (e.g., the objects are

documents). Additional examples include attributes to measure weight, height, latitude and longitude

5. Discrete versus Continuous Attributes

The types are not mutually exclusive.
Classification algorithms developed from the field of machine learning often talk of attributes as being either *discrete* or *continuous*. Each type may be processed differently.

A **discrete attribute** has a finite or countably infinite set of values, which may or may not be represented as integers. The attributes *hair color*, *smoker*, *medical test*, and *drink size* each have a finite number of values, and so are discrete. Note that discrete attributes may have numeric values, such as 0 and 1 for binary attributes or, the values 0 to 110 for the attribute *age*. An attribute is *countably infinite* if the set of possible values is infinite but the values can be put in a one-to-one correspondence with natural numbers. For example, the attribute *customer ID* is countably infinite. The number of customers can grow to infinity, but in reality, the actual set of values is countable (where the values can be put in one-to-one correspondence with the set of integers). Zip codes are another example.

If an attribute is not discrete, it is **continuous**. The terms *numeric attribute* and *continuous attribute* are often used interchangeably in the literature. In practice, real values are represented using a finite number of digits. Continuous attributes are typically represented as floating-point variables.

UNIT-1

Basic Statistical Descriptions of Data

For data preprocessing to be successful, it is essential to have an overall picture of the data. Basic statistical descriptions can be used to identify properties of the data and highlight which data values should be treated as noise or outliers.

Measuring the Central Tendency: Mean, Median, and Mode

Suppose that we have some attribute *X*, like *salary*, which has been recorded for a set of objects.
Let $x_1, x_2, \ldots, x_N$ be the set of *N* observed values or *observations* for *X*. Here, these values may also be referred to as the data set (for *X*). If we were to plot the observations for *salary*, where would most of the values fall? This gives us an idea of the central tendency of the data.

Measures of central tendency include the mean, median, mode, and midrange.
The most common and effective numeric measure of the "center" of a set of data is the *(arithmetic) mean*. Let $x_1, x_2, \ldots, x_N$ be a set of *N* values or *observations*, such as for some numeric attribute *X*, like *salary*. The **mean** of this set of values is

$$\bar{x} = \frac{\sum_{i=1}^{N} x_i}{N} = \frac{x_1 + x_2 + \cdots + x_N}{N}. \qquad (2.1)$$

This corresponds to the built-in aggregate function, *average* (avg() in SQL), provided in relational database systems.

Example . Suppose we have the following values for *salary* (in thousands of dollars), shown in increasing order: 30, 36, 47, 50, 52, 52, 56, 60, 63, 70, 70, 110. Using Eq. (2.1), we have

$$\bar{x} = \frac{30 + 36 + 47 + 50 + 52 + 52 + 56 + 60 + 63 + 70 + 70 + 110}{12}$$

$$= \frac{696}{12} = 58.$$

Thus, the mean salary is $58,000.

Sometimes, each value *xi* in a set may be associated with a weight *wi* for $i = 1, \ldots, N$.
The weights reflect the significance, importance, or occurrence frequency attached to their respective values. In this case, we can compute

$$\bar{x} = \frac{\sum_{i=1}^{N} w_i x_i}{\sum_{i=1}^{N} w_i} = \frac{w_1 x_1 + w_2 x_2 + \cdots + w_N x_N}{w_1 + w_2 + \cdots + w_N}. \quad (2.2)$$

This is called the weighted arithmetic mean or the weighted average

For skewed (asymmetric) data, a better measure of the center of data is the **median**, which is the middle value in a set of ordered data values. It is the value that separates the higher half of a data set from the lower half.

Example Median: The median is expensive to compute when we have a large number of observations. For numeric attributes, however, we can easily *approximate* the value. Assume that data are grouped in intervals according to their *xi* data values and that the frequency (i.e., number of data values) of each interval is known. For example, employees may be grouped according to their annual salary in intervals such as \$10–20,000, \$20–30,000, and so on. Let the interval that contains the median frequency be the *median interval*. We can approximate the median of the entire data set (e.g., the median salary) by interpolation using the formula

$$median = L_1 + \left(\frac{N/2 - (\sum freq)_l}{freq_{median}} \right) width,$$

where L_1 is the lower boundary of the median interval, N is the number of values in the entire data set, $(\sum freq)_l$ is the sum of the frequencies of all of the intervals

The *mode* is another measure of central tendency. The **mode** for a set of data is the value that occurs most frequently in the set. Therefore, it can be determined for qualitative and quantitative attributes. It is possible for the greatest frequency to correspond to several different values, which results in more than one mode. Data sets with one, two, or three modes are respectively called **unimodal**, **bimodal**, and **trimodal**. In general, a data set with two or more modes is**multimodal**. At the other extreme, if each data value occurs only once, then there is no mode.

Example Mode. The data from previous Example are bimodal. The two modes are \$52,000 and \$70,000.

For unimodal numeric data that are moderately skewed (asymmetrical), we have the following empirical relation:

$$mean\text{-}mode \approx 3 * (mean - median). \quad (2.4)$$

This implies that the mode for unimodal frequency curves that are moderately skewed can easily be approximated if the mean and median values are known.

The **midrange** can also be used to assess the central tendency of a numeric data set. It is the average of the largest and smallest values in the set. This measure is easy to compute using the SQL aggregate functions, max() and min().

Example Midrange. The midrange of the data of is (30,000 +110,000) /2 = $70,000.

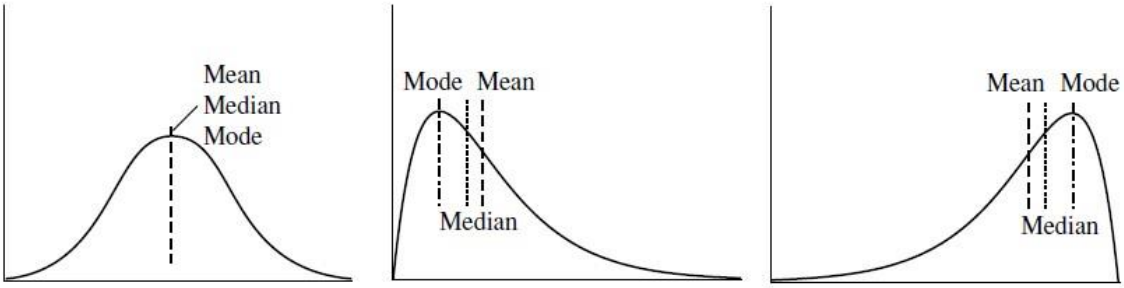


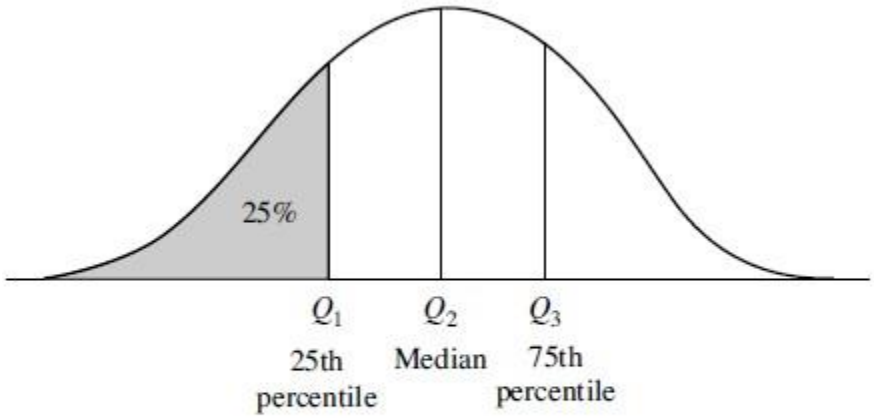
Figure 2.1 Mean, median, and mode of symmetric versus positively and negatively skewed data.

Measuring the Dispersion of Data: Range, Quartiles, Variance, Standard Deviation, and Interquartile Range

Range, Quartiles, and Interquartile Range

The **range** of the set is the difference between the largest (max()) and smallest (min()) values.

Quantiles are points taken at regular intervals of a data distribution, dividing it into essentially equalsize
consecutive sets. The 100-quantiles are more commonly referred to as **percentiles**; they divide the data distribution into 100 equal-sized consecutive sets. The median, quartiles, and percentiles are the most widely used forms of quantiles.



A plot of the data distribution for some attribute X. The quantiles plotted are quartiles. The three quartiles divide the distribution into four equal-size consecutive subsets. The second quartile corresponds to the median.

The distance between the first and third quartiles is a simple measure of spread that gives the range covered by the middle half of the data. This distance is called the **interquartile range** (**IQR**) and is defined as

$$IQR = Q3 - Q1. \qquad (2.5)$$

Example 2.10 Interquartile range. The quartiles are the three values that split the sorted data set into four equal parts. The data of Example 2.6 contain 12 observations, already sorted in increasing order. Thus, the quartiles for this data are the third, sixth, and ninth values, respectively, in the sorted list. Therefore, $Q1$ = \$47,000 and $Q3$ is \$63,000. Thus, the interquartile range is IQR = 63 -47 = \$16,000. (Note that the sixth value is a median, \$52,000, although this data set has two medians since the number of data values is even.)

Five-Number Summary, Boxplots, and Outliers

No single numeric measure of spread (e.g., *IQR*) is very useful for describing skewed distributions. it is more informative to also provide the two quartiles $Q1$ and $Q3$, along with the median. A common rule of thumb for identifying suspected **outliers** is to single out values falling at least 1.5 X *IQR* above the third quartile or below the first quartile.

The **five-number summary** of a distribution consists of the median ($Q2$), the quartiles $Q1$ and $Q3$, and the smallest and largest individual observations, written in the order of *Minimum*, $Q1$, *Median*, $Q3$, *Maximum*.

Boxplots are a popular way of visualizing a distribution. A boxplot incorporates the five-number summary as follows:

- Typically, the ends of the box are at the quartiles so that the box length is the interquartile range.

- The median is marked by a line within the box.
- Two lines (called *whiskers*) outside the box extend to the smallest (*Minimum*) and largest (*Maximum*) observations.

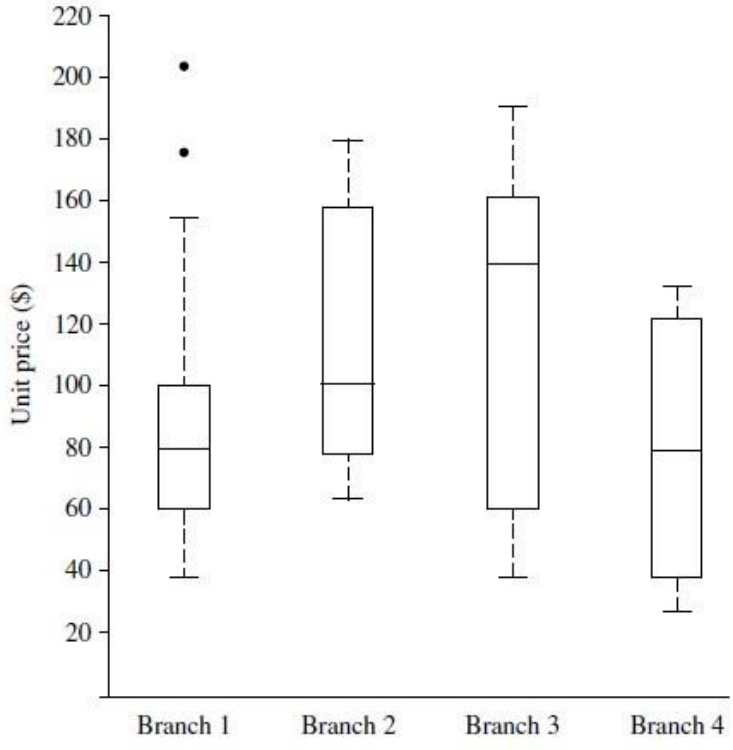


Figure 2.3 Boxplot for the unit price data for items sold at four branches of *AllElectronics* during a given time period.

When dealing with a moderate number of observations, it is worthwhile to plot potential outliers individually. To do this in a boxplot, the whiskers are extended to the extreme low and high observations *only if* these values are less than 1.5 * *IQR* beyond the quartiles.Otherwise, the whiskers terminate at the most extreme observations occurring within 1.5 * *IQR* of the quartiles. The remaining cases are plotted individually. Boxplots can be used in the comparisons of several sets of compatible data.

Example Boxplot. For branch 1, we see that the median price of items sold is $80, *Q*1 is $60, and *Q*3 is $100. Notice that two outlying observations for this branch were plotted individually, as their values of 175 and 202 are more than 1.5 times the IQR here of 40.

Variance and Standard Deviation

Variance and standard deviation are measures of data dispersion. They indicate how spread out a data distribution is. A low standard deviation means that the data observations tend to be very close to the mean, while a high standard deviation indicates that the data are spread out over a large range of values.

The **variance** of N observations, $x_1, x_2, \ldots, x_N$, for a numeric attribute X is

$$\sigma^2 = \frac{1}{N}\sum_{i=1}^{N}(x_i - \bar{x})^2 = \left(\frac{1}{N}\sum_{i=1}^{N}x_i^2\right) - \bar{x}^2, \tag{2.6}$$

where $\bar{x}$ is the mean value of the observations, as defined in Eq. (2.1). The **standard deviation**, σ, of the observations is the square root of the variance, σ^2.

Variance and standard deviation. In Example 2.6, we found $\bar{x} = \$58{,}000$ using Eq. (2.1) for the mean. To determine the variance and standard deviation of the data from that example, we set $N = 12$ and use Eq. (2.6) to obtain

$$\begin{aligned}\sigma^2 &= \frac{1}{12}(30^2 + 36^2 + 47^2 \ldots + 110^2) - 58^2 \\ &\approx 379.17 \\ \sigma &\approx \sqrt{379.17} \approx 19.47.\end{aligned}$$

■

UNIT-1

Graphic Displays of Basic Statistical Descriptions of Data

These include *quantile plots*, *quantile–quantile plots*, *histograms*, and *scatter plots*. Such graphs are helpful for the visual inspection of data, which is useful for data preprocessing. The first three of these show univariate distributions (i.e., data for one attribute), while scatter plots show bivariate distributions (i.e., involving two attributes).

Quantile Plot

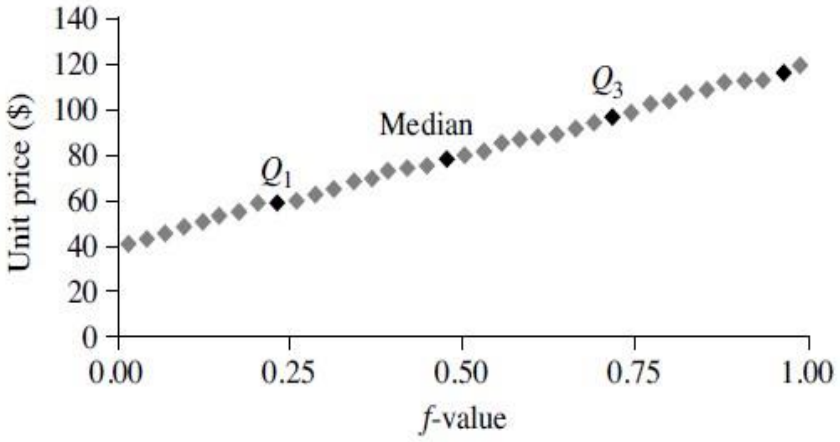
A **quantile plot** is a simple and effective way to have a first look at a univariate data distribution. First, it displays all of the data for the given attribute Second, it plots quantile information . Let *xi* , for $i = 1$ to N, be the data sorted in increasing order so that $x1$ is the smallest observation and xN is the largest for some ordinal or numeric attribute X. Each observation, *xi* , is paired with a percentage, *fi* , which indicates that approximately *fi* *100% of the data are below the value, *xi* .We say "approximately" because there may not be a value with exactly a fraction, *fi* , of the data below *xi* . Note that the 0.25 percentile corresponds to quartile $Q1$, the 0.50 percentile is the median, and the 0.75 percentile is $Q3$.

$$\text{Let } fi = (i - 0.5) / N. \qquad (2.7)$$

These numbers increase in equal steps of $1/N$, ranging from $1/2N$ (which is slightlyabove 0) to $1 - 1/2N$ (which is slightly below 1). On a quantile plot, *xi* is graphed against *fi* . This allows us to compare different distributions based on their quantiles. For example, given the quantile plots of sales data for two different time periods, we can compare their $Q1$, median, $Q3$, and other *fi* values at a glance.

Table 2.1 A Set of Unit Price Data for Items Sold at a Branch of *AllElectronics*

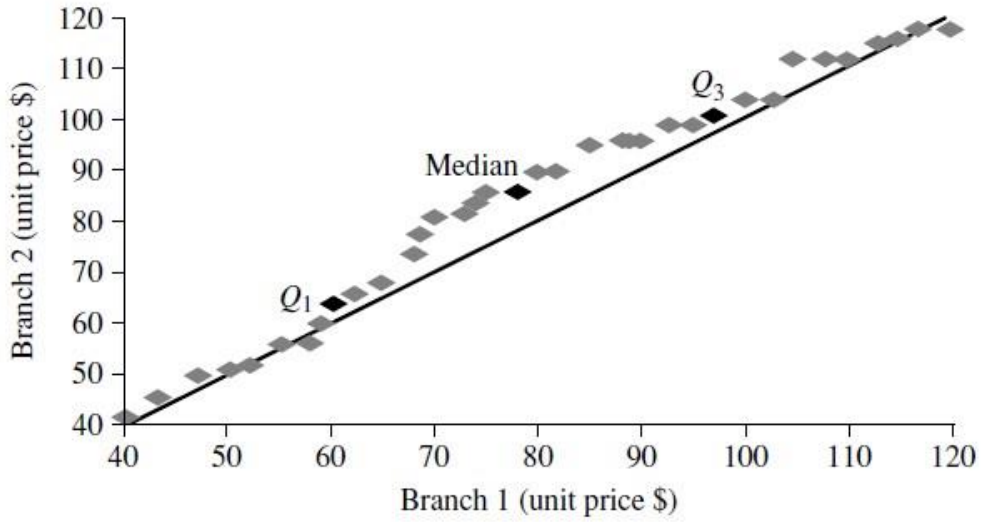
Unit price ($)	*Count of items sold*
40	275
43	300
47	250
—	—
74	360
75	515
78	540
—	—
115	320
117	270
120	350



A quantile plot for the unit price data of Table 2.1.

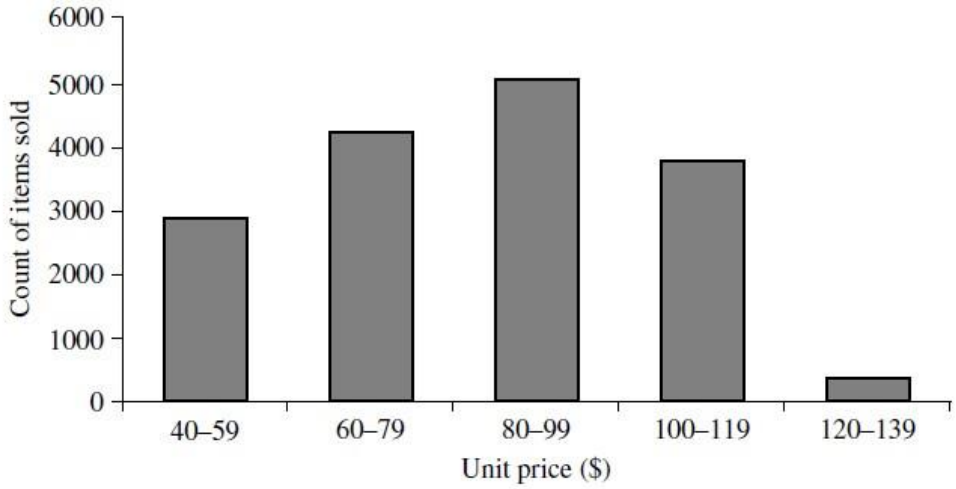
Quantile–Quantile Plot

A**quantile–quantile plot**, or **q-q plot**, graphs the quantiles of one univariate distribution against the corresponding quantiles of another. It is a powerful visualization tool in that it allows the user to view whether there is a shift in going fromone distribution to another.



A q-q plot for unit price data from two *AllElectronics* branches.

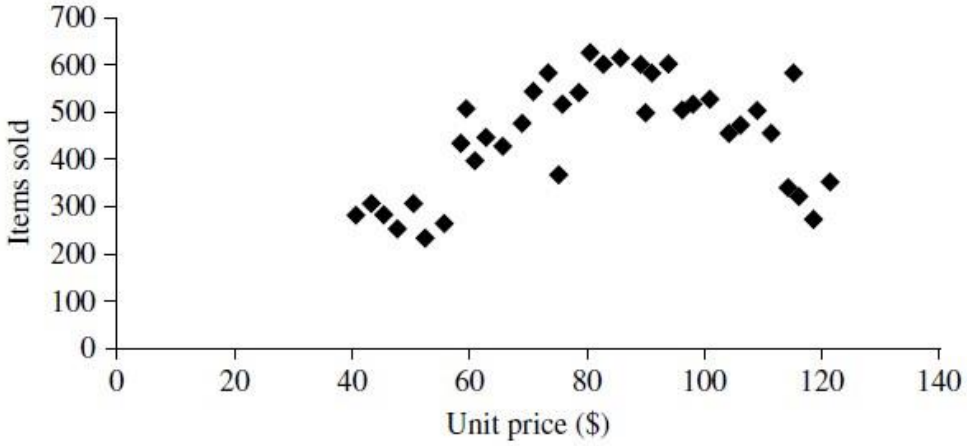
Histograms



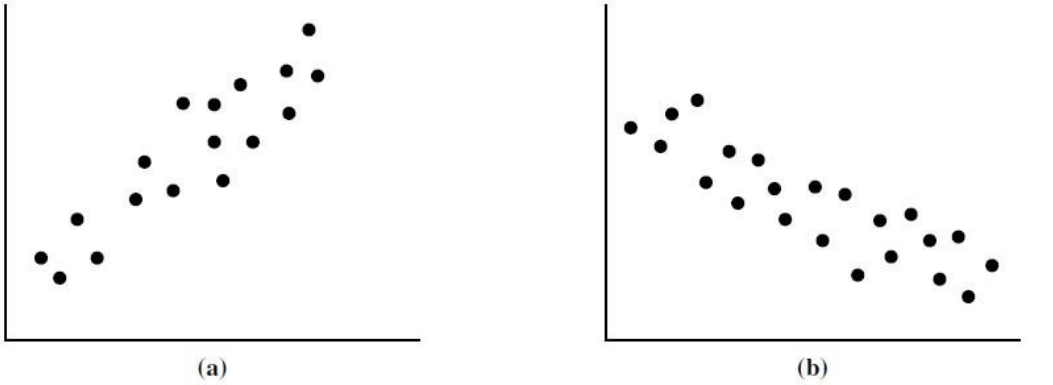
5 A histogram for the Table 2.1 data set.

Scatter Plots and Data Correlation

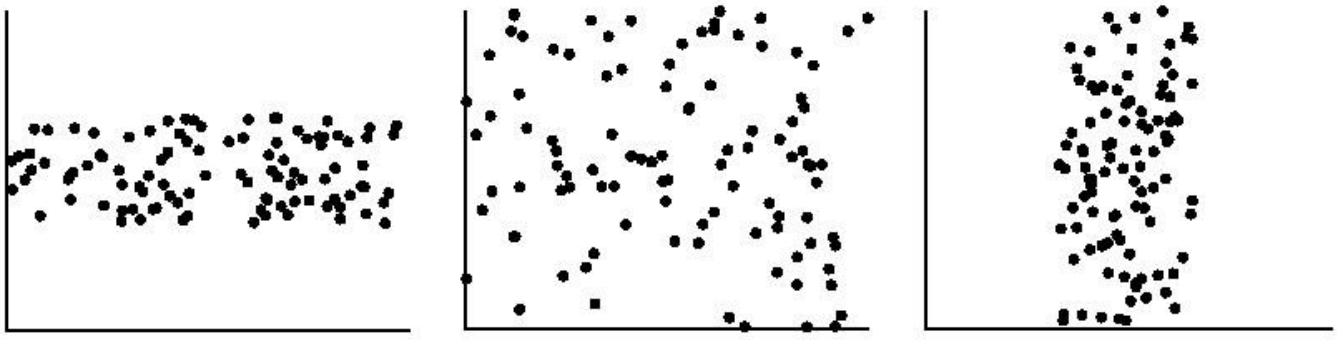
A **scatter plot** is one of the most effective graphical methods for determining if there appears to be a relationship, pattern, or trend between two numeric attributes.



A scatter plot for the Table 2.1 data set.



Scatter plots can be used to find (a) positive or (b) negative correlations between attributes.



Three cases where there is no observed correlation between the two plotted attributes in each of the data sets.

Measuring Data Similarity and Dissimilarity

In data mining applications, such as clustering, outlier analysis, and nearest-neighbor classification, we need ways to assess how alike or unalike objects are in comparison to one another. For example, a store may want to search for clusters of *customer* objects, resulting in groups of customers with similar characteristics (e.g., similar income, area of residence, and age). Such information can then be used for marketing. A **cluster** is a collection of data objects such that the objects within a cluster are *similar* to one
another and *dissimilar* to the objects in other clusters. Outlier analysis also employs clusteringbased techniques to identify potential outliers as objects that are highly dissimilar to others. Knowledge of object similarities can also be used in nearest-neighbor classification schemes where a given object (e.g., a *patient*) is assigned a class label (relating to, say, a *diagnosis*) based on its similarity toward other objects in the model.

Similarity and dissimilarity are related, which are referred to as measures of *proximity*. A similarity measure for two objects, *i* and *j*, will typically return the value 0 if the objects are unalike. The higher
the similarity value, the greater the similarity between objects. (Typically, a value of 1 indicates complete similarity, that is, the objects are identical.) A dissimilarity measure works the opposite way. It returns a value of 0 if the objects are the same (and therefore, far from being dissimilar). The higher the dissimilarity value, the more dissimilar the two objects are.

Data Matrix versus Dissimilarity Matrix

Suppose that we have *n* objects (e.g., persons, items, or courses) described by *p* attributes (also called *measurements* or *features*, such as age, height, weight, or gender). The objects are $x1 = (x11, x12, \ldots, x1p)$, $x2 = (x21, x22, \ldots, x2p)$, and so on, where *xij* is the value for object *xi* of the *j*th attribute. For brevity, we hereafter refer to object *xi* as object *i*. The objects may be tuples in a relational database, and are also referred to as *data samples* or *feature vectors*.

Main memory-based clustering and nearest-neighbor algorithms typically operate on either of the following two data structures:

Data matrix (or object-by-attribute structure): This structure stores the n data objects in the form of a relational table, or n-by-p matrix (n objects p attributes):

$$\begin{bmatrix} x_{11} & \cdots & x_{1f} & \cdots & x_{1p} \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ x_{i1} & \cdots & x_{if} & \cdots & x_{ip} \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ x_{n1} & \cdots & x_{nf} & \cdots & x_{np} \end{bmatrix}. \quad (2.8)$$

Each row corresponds to an object. As part of our notation, we may use f to index through the p attributes.

Dissimilarity matrix (or *object-by-object structure*): This structure stores a collection of proximities that are available for all pairs of n objects. It is often represented by an n-by-n table:

$$\begin{bmatrix} 0 & & & & \\ d(2,1) & 0 & & & \\ d(3,1) & d(3,2) & 0 & & \\ \vdots & \vdots & \vdots & & \\ d(n,1) & d(n,2) & \cdots & \cdots & 0 \end{bmatrix}, \quad \textbf{(2.9)}$$

where $d(i, j)$ is the measured **dissimilarity** or "difference" between objects i and j.

$d(i, j)$ is a non-negative number that is close to 0 when objects i and j are highly similar or "near" each other, and becomes larger the more they differ.

Measures of similarity can often be expressed as a function of measures of dissimilarity. For example, for nominal data

$$sim(i, j) = 1 - d(i, j), \quad (2.10)$$

where $sim(i, j)$ is the similarity between objects i and j.

A data matrix is made up of two entities or "things," namely rows (for objects) and columns (for attributes). Therefore, the data matrix is often called a **two-mode** matrix.

The dissimilarity matrix contains one kind of entity (dissimilarities) and so is called a **one-mode** matrix. Many clustering and nearest-neighbor algorithms operate on a dissimilarity matrix. Data

in the form of a data matrix can be transformed into a dissimilarity matrix before applying such algorithms.

Proximity Measures for Nominal Attributes

Let the number of states of a nominal attribute be *M*. The states can be denoted by letters, symbols, or a set of integers, such as 1, 2, …, *M*. Notice that such integers are used just for data handling and do not represent any specific ordering.

The dissimilarity between two objects *i* and *j* can be computed based on the ratio of mismatches:

$$d(i, j) = \frac{p - m}{p}, \tag{2.11}$$

where *m* is the number of *matches* (i.e., the number of attributes for which *i* and *j* are in the same state), and *p* is the total number of attributes describing the objects.

Example :Dissimilarity between nominal attributes. Suppose that we have the sample data of Table 2.2, except that only the *object-identifier* and the attribute *test-1* are available, where *test-1* is nominal. (We will use *test-2* and *test-3* in later examples.) Let's compute the dissimilarity matrix

$$\begin{bmatrix} 0 & & & \\ d(2,1) & 0 & & \\ d(3,1) & d(3,2) & 0 & \\ d(4,1) & d(4,2) & d(4,3) & 0 \end{bmatrix}.$$

(Eq. 2.9), that is

Since here we have one nominal attribute, *test-1*, we set *p* D 1 in Eq. (2.11) so that *d* (*i*, *j)* evaluates to 0 if objects *i* and *j* match, and 1 if the objects differ. Thus, we get

$$\begin{bmatrix} 0 & & & \\ 1 & 0 & & \\ 1 & 1 & 0 & \\ 0 & 1 & 1 & 0 \end{bmatrix}.$$

From this, we see that all objects are dissimilar except objects 1 and 4 (i.e.,*d(*4,1)=0)

Table 2.2 A Sample Data Table Containing Attributes of Mixed Type

Object Identifier	test-1 (nominal)	test-2 (ordinal)	test-3 (numeric)
1	code A	excellent	45
2	code B	fair	22
3	code C	good	64
4	code A	excellent	28

Alternatively, similarity can be computed as

$$sim(i, j) = 1 - d(i, j) = \frac{m}{p}. \qquad (2.12)$$

Proximity Measures for Binary Attributes

A binary attribute has only one of two states: 0 and 1, where 0 means that the attribute is absent, and 1 means that it is present.

Example: Given the attribute *smoker* describing a patient, for instance, 1 indicates that the

Table 2.3 Contingency Table for Binary Attributes

		Object j		
		1	0	sum
Object i	1	q	r	$q+r$
	0	s	t	$s+t$
	sum	$q+s$	$r+t$	p

patient smokes, while 0 indicates that the patient does not. dissimilarity between i and j is

$$d(i, j) = \frac{r+s}{q+r+s+t}.$$

For asymmetric binary attributes, the two states are not equally important, such as the *positive* (1) and *negative* (0) outcomes of a disease test.

Given two asymmetric binary attributes, the agreement of two 1s (a positive match) is then considered more significant than that of two 0s (a negative match).
Therefore, such binary attributes are often considered "monary" (having one state). The dissimilarity based on these attributes is called **asymmetric binary dissimilarity**

$$d(i, j) = \frac{r + s}{q + r + s}.$$

We can measure the difference between two binary attributes based on the notion of similarity instead of dissimilarity. For example, the **asymmetric binary similarity** between the objects *i* and *j* can be computed as

$$sim(i, j) = \frac{q}{q + r + s} = 1 - d(i, j).$$

The coefficient Sim(i, j) is called the **Jaccard coefficient**

Example Dissimilarity between binary attributes. Suppose that a patient record table contains the attributes *name, gender, fever, cough, test-1, test-2, test-3*, and *test-4*, where *name* is an object identifier, *gender* is a symmetric attribute, and the remaining attributes are asymmetric binary.
For asymmetric attribute values, let the values *Y* (*yes*) and *P* (*positive*) be set to 1, and the value *N* (*no* or *negative*) be set to 0. Suppose that the distance between objects (patients) is computed based only on the asymmetric attributes

Table 2.4 Relational Table Where Patients Are Described by Binary Attributes

name	*gender*	*fever*	*cough*	*test-1*	*test-2*	*test-3*	*test-4*
Jack	M	Y	N	P	N	N	N
Jim	M	Y	Y	N	N	N	N
Mary	F	Y	N	P	N	P	N
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮

The distance between each pair of the three patients—Jack, Mary, and Jim—is

$$d(Jack, Jim) = \frac{1 + 1}{1 + 1 + 1} = 0.67,$$

$$d(Jack, Mary) = \frac{0 + 1}{2 + 0 + 1} = 0.33,$$

$$d(Jim, Mary) = \frac{1 + 2}{1 + 1 + 2} = 0.75.$$

These measurements suggest that Jim and Mary are unlikely to have a similar disease because they have the highest dissimilarity value among the three pairs. Of the three patients, Jack and Mary are the most likely to have a similar disease.

4. Dissimilarity of Numeric Data: Minkowski Distance

These measures include the ***Euclidean, Manhattan*, and *Minkowski distances***.
In some cases, the data are normalized before applying distance calculations. This involves transforming the data to fall within a smaller or common range, such as [1, 1] or [0.0, 1.0]. Example, *height* attribute, which could be measured in either meters or inches.

$$d(i, j) = \sqrt{(x_{i1} - x_{j1})^2 + (x_{i2} - x_{j2})^2 + \cdots + (x_{ip} - x_{jp})^2}.$$

Another measure is known as Manhattan(or city block) and it is defined as

$$d(i, j) = |x_{i1} - x_{j1}| + |x_{i2} - x_{j2}| + \cdots + |x_{ip} - x_{jp}|.$$

Both the Euclidean and the Manhattan distance satisfy the following mathematical properties:
Non-negativity: $d(i, j) \geq 0$: Distance is a non-negative number.
Identity of indiscernibles: $d(i, i) = 0$: The distance of an object to itself is 0.
Symmetry: $d(i, j) = d(j, i)$: Distance is a symmetric function.
Triangle inequality: $d(i, j) \leq d(i, k) + d(k, j)$: Going directly from object *i* to object *j* in space is no more than making a detour over any other object *k*.
Minkowski distance is a generalization of the Euclidean and Manhattan distances. It is defined as

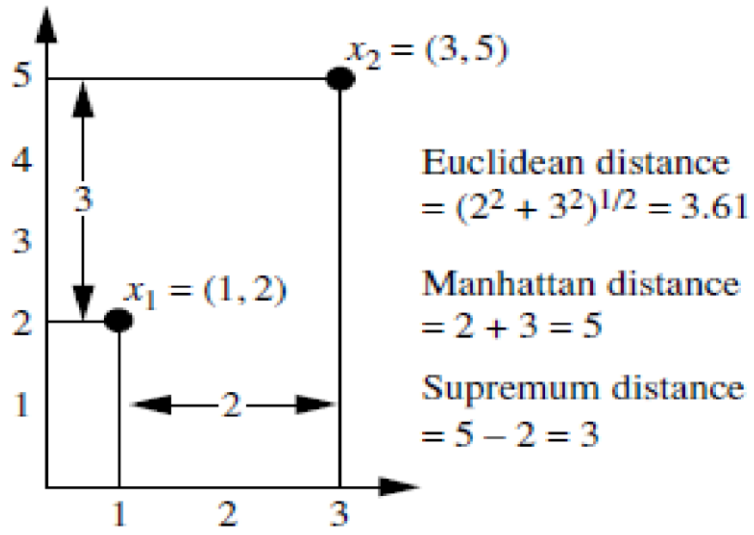
$$d(i, j) = \sqrt[h]{|x_{i1} - x_{j1}|^h + |x_{i2} - x_{j2}|^h + \cdots + |x_{ip} - x_{jp}|^h}, \qquad (2.18)$$

Supremum distance (also known as the **Chebyshev distance**) is a generalization of the Minkowski distance for $h \to$ *infinity*. it is defined as

$$d(i, j) = \lim_{h \to \infty} \left(\sum_{f=1}^{p} |x_{if} - x_{jf}|^h \right)^{\frac{1}{h}} = \max_{f}^{p} |x_{if} - x_{jf}|.$$

The L^∞ norm is also known as the *uniform norm.*

Euclidean, Manhattan, and supremum distances between two objects



If each attribute is assigned a weight according to its perceived importance, the **weighted Euclidean distance** can be computed as

$$d(i, j) = \sqrt{w_1|x_{i1} - x_{j1}|^2 + w_2|x_{i2} - x_{j2}|^2 + \cdots + w_m|x_{ip} - x_{jp}|^2}. \qquad (2.20)$$

Weighting can also be applied to other distance measures as well.

5. Proximity Measures for Ordinal Attributes

The values of an ordinal attribute have a meaningful order or ranking about them, yet the magnitude between successive values is unknown. An example includes the sequence *small*, *medium*, *large* for a *size* attribute. Ordinal attributes may also be obtained from the discretization of numeric attributes by splitting the value range into a finite number of categories. These categories are organized into ranks. Let *M* represent the number of possible states that an ordinal attribute can have. These ordered states define the ranking 1, …. , *Mf* . The treatment of ordinal attributes is quite similar to that of numeric attributes when computing dissimilarity between objects.

$$z_{if} = \frac{r_{if} - 1}{M_f - 1}. \qquad (2.21)$$

Example 2.21 Dissimilarity between ordinal attributes. Suppose that we have the sample data shown earlier in Table 2.2, except that this time only the *object-identifier* and the continuous ordinal attribute, *test-2*, are available. There are three states for *test-2*: *fair*, *good*, and *excellent*, that is, *Mf* D 3. For step 1, if we replace each value for *test-2* by its rank, the our objects are assigned the ranks 3, 1, 2, and 3, respectively. Step 2 normalizes the ranking by mapping rank 1 to 0.0, rank 2 to 0.5, and rank 3 to 1.0. For step 3, we can use, say, the Euclidean distance (Eq. 2.16), which results in the following dissimilarity
matrix:

$$\begin{bmatrix} 0 & & & \\ 1.0 & 0 & & \\ 0.5 & 0.5 & 0 & \\ 0 & 1.0 & 0.5 & 0 \end{bmatrix}$$

Therefore, objects 1 and 2 are the most dissimilar, as are objects 2 and 4 (i.e., *d(*2,1)=1.0 and *d(*4,2)=1.0). This makes intuitive sense since objects 1 and 4 are both *excellent*. Object 2 is *fair*, which is at the opposite end of the range of values for *test-2*. Similarity values for ordinal attributes can be interpreted from dissimilarity as *sim(i, j)* = 1 –*d(i, j)*.

6. Dissimilarity for Attributes of Mixed Types

One approach is to group each type of attribute together, performing separate data mining (e.g., clustering) analysis for each type. This is feasible if these analyses derive compatible results. However, in real applications, it is unlikely that a separate analysis per attribute type will generate compatible results.

A more preferable approach is to process all attribute types together, performing a single analysis. One such technique combines the different attributes into a single dis-similarity matrix, bringing all of the meaningful attributes onto a common scale of the interval [0.0, 1.0]. Suppose that the data set contains p attributes of mixed type. The dissimilarity $d(i, j)$between objects i and j is defined as

$$d(i,j) = \frac{\sum_{f=1}^{p} \delta_{ij}^{(f)} d_{ij}^{(f)}}{\sum_{f=1}^{p} \delta_{ij}^{(f)}}, \qquad (2.22)$$

where the indicator $\delta_{ij}^{(f)} = 0$ if either (1) x_{if} or x_{jf} is missing (i.e., there is no measurement of attribute f for object i or object j), or (2) $x_{if} = x_{jf} = 0$ and attribute f is asymmetric binary; otherwise, $\delta_{ij}^{(f)} = 1$. The contribution of attribute f to the dissimilarity between i and j (i.e., $d_{ij}^{(f)}$) is computed dependent on its type:

- If f is numeric: $d_{ij}^{(f)} = \frac{|x_{if} - x_{jf}|}{max_h x_{hf} - min_h x_{hf}}$, where h runs over all nonmissing objects for attribute f.
- If f is nominal or binary: $d_{ij}^{(f)} = 0$ if $x_{if} = x_{jf}$; otherwise, $d_{ij}^{(f)} = 1$.
- If f is ordinal: compute the ranks r_{if} and $z_{if} = \frac{r_{if} - 1}{M_f - 1}$, and treat z_{if} as numeric.

These steps are identical to what we have already seen for each of the individual attribute types. The only difference is for numeric attributes, where we normalize so that the values map to the interval [0.0, 1.0]. Thus, the dissimilarity between objects can be computed even when the attributes describing the objects are of different types.

Example: Dissimilarity between attributes of mixed type. Let's compute a dissimilarity matrix for the objects in Table 2.2. Now we will consider *all* of the attributes, which are of different types. In Examples 2.17 and 2.21, we worked out the dissimilarity matrices for each of the individual attributes. The procedures we followed for *test-1* (which is nominal) and *test-2* (which is ordinal) are the same as outlined earlier for processing attributes of mixed types. Therefore, we can use the dissimilarity matrices obtained for *test-1* and *test-2* later when we compute Eq. (2.22). First, however, we need to compute the dissimilarity matrix for the third attribute, *test-3* (which is numeric). That is, we must compute $d_{ij}^{(3)}$. Following the case for numeric attributes, we let $max_h x_h = 64$ and $min_h x = 22$. The difference between the two is used in Eq. (2.22) to normalize the values of the dissimilarity matrix. The resulting dissimilarity matrix for *test-3* is

$$\begin{bmatrix} 0 & & & \\ 0.55 & 0 & & \\ 0.45 & 1.00 & 0 & \\ 0.40 & 0.14 & 0.86 & 0 \end{bmatrix}.$$

We can now use the dissimilarity matrices for the three attributes in our computation of Eq. (2.22). The indicator $\delta_{ij}^{(f)} = 1$ for each of the three attributes, f. We get, for example, $d(3, 1) = \frac{1(1)+1(0.50)+1(0.45)}{3} = 0.65$. The resulting dissimilarity matrix obtained for the data described by the three attributes of mixed types is:

$$\begin{bmatrix} 0 & & & \\ 0.85 & 0 & & \\ 0.65 & 0.83 & 0 & \\ 0.13 & 0.71 & 0.79 & 0 \end{bmatrix}.$$

From Table 2.2, we can intuitively guess that objects 1 and 4 are the most similar, based on their values for *test*-1 and *test*-2. This is confirmed by the dissimilarity matrix, where *d(*4, 1) is the lowest value for any pair of different objects. Similarly, the matrix indicates that objects 1 and 2 are the least similar.

7. Cosine Similarity

A document can be represented by thousands of attributes, each recording the frequency of a particular word (such as a keyword) or phrase in the document. Thus, each document is an object represented by what is called a *term-frequency vector*. For example, a *Document* contains five instances of the word *team*, while *hockey* occurs three times. The word *coach* is absent from the entire document, Such data can be highly asymmetric.

Cosine similarity is a measure of similarity that can be used to compare documents or, say, give a ranking of documents with respect to a given vector of query words.

Table 2.5 Document Vector or Term-Frequency Vector

Document	team	coach	hockey	baseball	soccer	penalty	score	win	loss	season
Document1	5	0	3	0	2	0	0	2	0	0
Document2	3	0	2	0	1	1	0	1	0	1
Document3	0	7	0	2	1	0	0	3	0	0
Document4	0	1	0	0	1	2	2	0	3	0

similarity function,

$$sim(\boldsymbol{x}, \boldsymbol{y}) = \frac{\boldsymbol{x} \cdot \boldsymbol{y}}{||\boldsymbol{x}||||\boldsymbol{y}||}, \qquad (2.23)$$

where $||\boldsymbol{x}||$ is the Euclidean norm of vector $\boldsymbol{x} = (x1, x2, \ldots, xp)$, defined as $\sqrt{x_1^2 + x_2^2 + \cdots + x_p^2}$.

The measure computes the cosine of the angle between vectors $\boldsymbol{x}$ and $\boldsymbol{y}$. A cosine value of 0 means that the two vectors are at 90 degrees to each other (orthogonal) and have no match. The closer the cosine value to 1, the smaller the angle and the greater the match between vectors. Note that because the cosine similarity measure does not obey all of the properties of Section 2.4.4 defining metric measures, it is referred to as a *nonmetric measure*.

Cosine similarity between two term-frequency vectors. Suppose that $\boldsymbol{x}$ and $\boldsymbol{y}$ are the first two term-frequency vectors in Table 2.5. That is, $\boldsymbol{x} = (5,0,3,0,2,0,0,2,0,0)$ and $\boldsymbol{y} = (3,0,2,0,1,1,0,1,0,1)$. How similar are $\boldsymbol{x}$ and $\boldsymbol{y}$? Using Eq. (2.23) to compute the cosine similarity between the two vectors, we get:

$$\begin{aligned}
\boldsymbol{x}^t \cdot \boldsymbol{y} &= 5 \times 3 + 0 \times 0 + 3 \times 2 + 0 \times 0 + 2 \times 1 + 0 \times 1 + 0 \times 0 + 2 \times 1 \\
&\quad + 0 \times 0 + 0 \times 1 = 25 \\
||\boldsymbol{x}|| &= \sqrt{5^2 + 0^2 + 3^2 + 0^2 + 2^2 + 0^2 + 0^2 + 2^2 + 0^2 + 0^2} = 6.48 \\
||\boldsymbol{y}|| &= \sqrt{3^2 + 0^2 + 2^2 + 0^2 + 1^2 + 1^2 + 0^2 + 1^2 + 0^2 + 1^2} = 4.12 \\
sim(\boldsymbol{x}, \boldsymbol{y}) &= 0.94
\end{aligned}$$

Therefore, if we were using the cosine similarity measure to compare these documents, they would be considered quite similar. ■

When attributes are binary-valued, the cosine similarity function can be interpreted in terms of shared features or attributes. Suppose an object $\boldsymbol{x}$ possesses the ith attribute if $xi = 1$. Then $\boldsymbol{x}t\ \boldsymbol{y}$ is the number of attributes possessed (i.e., shared) by both $\boldsymbol{x}$ and $\boldsymbol{y}$, and $|\boldsymbol{x}|\ |\boldsymbol{y}|$ is the *geometric mean* of the number of attributes possessed by $\boldsymbol{x}$ and the number possessed by $\boldsymbol{y}$. Thus, Sim($\boldsymbol{x}$, $\boldsymbol{y}$) is a measure of relative possession of common attributes.

A simple variation of cosine similarity for the preceding scenario is

$$sim(x, y) = \frac{x \cdot y}{x \cdot x + y \cdot y - x \cdot y},$$

which is the ratio of the number of attributes shared by $\boldsymbol{x}$ and $\boldsymbol{y}$ to the number of attributes possessed by $\boldsymbol{x}$ or $\boldsymbol{y}$. This function, known as the **Tanimoto coefficient** or **Tanimoto distance**, is frequently used in information retrieval and biology taxonomy.