Semester – V

16CAU502B **DATA MINING** **4H – 4C**

Instruction Hours / week: L: 4 T: 0 P: 0 **Marks:** Int : **40** Ext : **60** Total: **100**

Scope

This course introduce students to the basic concepts and techniques of Data Mining, and the data mining software used for solving practical problems.

Objectives

The primary objectives of the course is teach students:

- the algorithms and methods of data mining.
- to develop data mining programs and applications.
- to program using available data mining tools and general-purpose languages.
- the analysis, metrics, visualization and navigation of data mining results.
- to use a few commercial data mining tools

Unit- I

Introduction : Fundamentals of data mining – Data Mining Functionalities – Classification of Data Mining systems – Major issues in Data Mining.
Data Warehouse and OLAP Technology: An Overview – Data Warehouse – Multidimensional Data Model – Data Warehouse Architecture

Unit-II

Data Preprocessing: Needs Preprocessing the Data – Data Cleaning – Data Integration and Transformation – Data Reduction – Discretization and Concept Hierarchy Generation – Online Data Storage.
Preparing Data for Mining: Variable Measures.

Unit-III

Mining Frequent Patterns, Associations and Correlations: Basic Concepts – Efficient and Scalable Frequent item set Mining Methods – From Association Mining to Correlation Analysis.

Unit-IV

Predictive and descriptive data mining techniques, supervised and unsupervised learning techniques, process of knowledge discovery in databases, pre-processing methods

Unit-V

Data Mining Techniques: Association Rule Mining, classification and regression techniques, clustering, Scalability and data management issues in data mining algorithms, measures of interestingness

Suggested Readings

1. Gupta, G.K., (2006). *Introduction to Data Mining with Case Studies*, PHI.
2. Pang-Ning Tan, Michael Steinbach, Vipin Kumar, (2005). *Introduction to Data Mining*, Pearson Education.
3. Richard Roiger, Michael Geatz, (2003). *Data Mining: A Tutorial Based Primer*, Pearson Education.
4. Soman,K.P., Diwakar Shyam, Ajay,V.,(2006*). Insight Into Data Mining: Theory And Practice*, PHI.

KarpagamAcademy of Higher Education
(Established Under Section 3 of UGC Act 1956)
Eachanari Post, Coimbatore – 641 021. INDIA
Phone : 0422-2611146, 2611082 Fax No : 0422 -2611043

DATA MINING (16CAU502B)

SEMESTER : V **CLASS : III BCA**

LECTURE PLAN
UNIT-I

S.NO	LECTURE DURATION (Hr)	TOPICS TO BE COVERED	SUPPORT MATERIALS
1.	1	Introduction :Fundamentals of Data Mining	R1: 6-10
2.	1	Data Mining Functionalities	R1:16-21,J1
3.	1	Classification of Data Mining Systems	R1:28-30
4.	1	Major Issues in Data Mining	R1:30-33
5.	1	Data warehouse and OLAP Technology : An Overview	R2: 1-9
6.	1	Difference between Operational Database and Data warehouse	R1:39-43
7.	1	Data warehouse	R2:19-36
8.	1	Multidimensional Data Model	R1:44-51
9.	1	Data Cubes and Star Schema	R1:52-54
10.	1	Data warehouse Architecture	R1:65-69,W1
11.	1	Recapitulation and Discussion of Important Questions	

References Books:

R1: Gupta,G.K.,(2006), Introduction to Data Mining with Case Studies,PHI

WEB SITES

W1:www.cs.gsu.edu/~cscv92/DM

Journals

J1: “Data Mining Techniques : A Survey Paper”, IJRET,Vol 2,Issue 11, November 2013.

UNIT-II

S.NO	LECTURE DURATION (Hr)	TOPICS TO BE COVERED	SUPPORT MATERIALS
1.	1	Data Preprocessing: Needs of Preprocessing the Data	R1: 113-114
2.		Measuring the Dispersion of Data	R1:114-115
3.	1	Graphic displays for basic descriptive data	R1:116-118
4.	1	Data Cleaning	R2:109-110
5.		Missing Values and Noisy Data	R2:110-111
6.	1	Data Integration and Data Transformation	R2:111-116
7.	1	Data Reduction	R2 117-119
8.	1	Dimensionality Reduction and Numerosity Reduction	R2:120-125
9.	1	Discretization and Concept Hierarchy Generation	R2:126-131
10.	1	Online Data Storage	R2:348-350,W2
11.	1	Preparing Data for Mining : Variable Measures	R2:351-352
12.	1	Recapitulation and Discussion of Important Questions	

References Books:

R1: Gupta,G.K.,(2006), Introduction to Data Mining with Case Studies,PHI
R2: Pang-Ning Tan, Michael Steinbach, Vipin Kumar, (2005), Introduction to Data Mining, Pearson Education

WEB SITES

W2: www.cs.nyu.edu/courses/spring08

UNIT-III

S.NO	LECTURE DURATION (Hr)	TOPICS TO BE COVERED	SUPPORT MATERIALS
1.	1	Mining Frequent Patterns	R1:227-229,J2
2.	1	Associations and Correlations : Basic Concepts	R1: 230-232
3.	1	Efficient and Scalable Frequent item set mining Methods	R1:235-239
4.	1	The Apriori Algorithm and Improving the efficiency of Apriori Algorithm	R1:240-242
5.	1	Mining Frequent itemset without candidate itemset	R1:243-245
6.	1	Mining Multilevel Association Rules and Approaches to Mining Multilevel Association Rules	R1:249-254
7.	1	Mining Quantitative Association Rules	R1:255-257
8.	1	From Association Mining to Correlation Analysis	R1:261-263
9.	1	Recapitulation and Discussion of Important Questions	

References Books:

R1: Gupta,G.K.,(2006), Introduction to Data Mining with Case Studies,PHI

WEB SITES

W1:www.cs.gsu.edu/~cscv92/DM

Journals

J2: “A Complete Survey on Application of Frequent Pattern Mining”,IJACST,Vol 3,2015.

UNIT-IV

S.NO	LECTURE DURATION (Hr)	TOPICS TO BE COVERED	SUPPORT MATERIALS
1.	1	Predictive and Descriptive Data Mining Techniques	R1:279-287
2.	1	Predictor Error Measures	R1: 282-284
3.	1	Issues regarding Prediction	R1:285-287
4.	1	Supervised and Unsupervised Learning Techniques	R1:288-290
5.	1	Process of Knowledge Discovery in Databases	R1:291-293,W3
6.	1	Pre-processing Methods	R1:294-296
7.	1	Recapitulation and Discussion of Important Questions	

References Books:

R1: Gupta,G.K.,(2006), Introduction to Data Mining with Case Studies,PHI

WEB SITES

W3:www.mimuw.edu.pl

UNIT-V

S.NO	LECTURE DURATION (Hr)	TOPICS TO BE COVERED	SUPPORT MATERIALS
1.	1	Association Rule Mining	R2:279-287
2.	1	Classification and Regression Techniques	R2: 282-284
3.	1	Linear and Nonlinear regression Techniques	R2:285-287
4.	1	Clustering	R2:288-290
5.	1	Scalability and Data Management issues in Data Mining Algorithms	R2:291-293,W4
6.	1	Recapitulation and Discussion of Important Questions	
7.	1	Discussion of Previous ESE Questions	
8.	1	Discussion of Previous ESE Questions	
9.	1	Discussion of Previous ESE Questions	

Reference Books:

R1: Gupta,G.K.,(2006), Introduction to Data Mining with Case Studies,PHI

R2: Pang-Ning Tan, Michael Steinbach, Vipin Kumar, (2005), Introduction to Data Mining, Pearson Education

R3: Richard Roiger, Michael Geatz (2003), Data Mining: A Tutorial Based Primer, Pearson Education

R4:Soman,K.P., Diwakar Shyam, Ajay,V.(2006), Insight into Data Mining: Theory and Practice, PHI.

WEB SITES

W1:www.cs.gsu.edu/~cscv92/DM
W2:www.cs.nyu.edu/courses/spring08
W3:www.mimuw.edu.pl
W4:www.cs.ccsu.edu

Journals

J1: “Data Mining Techniques : A Survey Paper”, IJRET,Vol 2,Issue 11, November 2013.J2: “A Complete Survey on Application of Frequent Pattern Mining”,IJACST,Vol 3,2015.
J3:”Spatial Data Mining Using Cluster Analysis”,IJCSIT,Vol 4,No.4,2012.

UNIT I

Introduction: Fundamentals of data mining - Data Mining Functionalities - Classification of Data Mining systems - Major issues in Data Mining. Data Warehouse and OLAP Technology: An Overview - Data Warehouse - Multidimensional Data Model - Data Warehouse Architecture

INTRODUCTION

What is data mining?

Data mining refers to extracting or mining" knowledge from large amounts of data. There are many other terms related to data mining, such as knowledge mining, knowledge extraction, data/pattern analysis, data archaeology, and data dredging. Many people treat data mining as a synonym for another popularly used term, Knowledge Discovery in Databases", or KDD.

What motivated data mining? Why is it important?

The major reason that data mining has attracted a great deal of attention in information industry in recent years is due to the wide availability of huge amounts of data and the imminent need for turning such data into useful information and knowledge. The information and knowledge gained can be used for applications ranging from business management, production control, and market analysis, to engineering design and science exploration.

Essential step in the process of knowledge discovery in databases

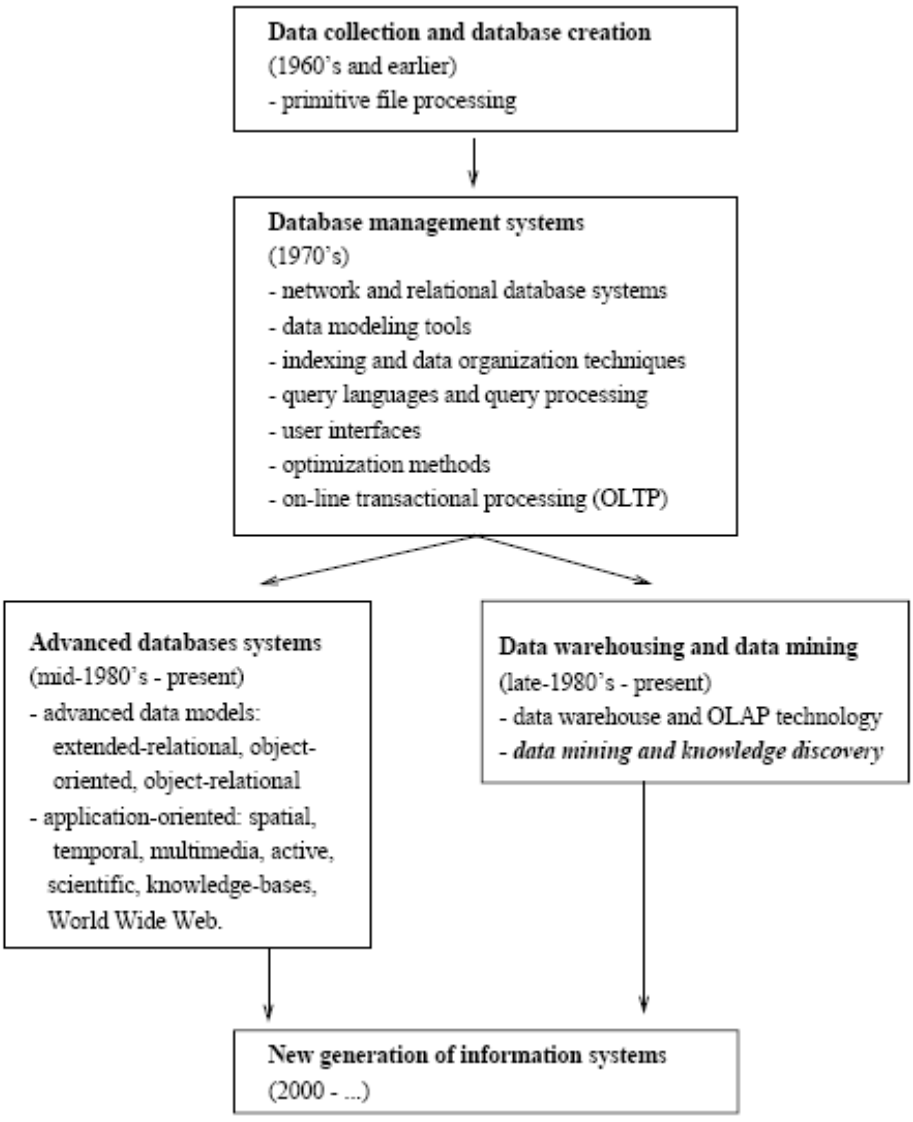
Knowledge discovery as a process is depicted in following figure and consists of an iterative sequence of the following steps:

- **data cleaning: to remove noise or irrelevant data**

□□data integration: where multiple data sources may be combined
□□data selection: where data relevant to the analysis task are retrieved from the database
□□data transformation: where data are transformed or consolidated into forms appropriate for mining by performing summary or aggregation operations
□□data mining :an essential process where intelligent methods are applied in order to extract data patterns.

□□**Pattern evaluation** to identify the truly interesting patterns representing knowledge based on some interestingness measures knowledge presentation: where visualization and knowledge representation techniques are used to present the mined knowledge to the user.

The evolution of database technology



Data mining: on what kind of data?

Describe the following advanced database systems and applications: object-relational databases, spatial databases, text databases, multimedia databases, the World Wide Web.

In principle, data mining should be applicable to any kind of information repository. This includes relational databases, data warehouses, transactional databases, advanced database systems, flat files, and the World-Wide Web. Advanced database systems include object-oriented and object relational databases, and special c application-oriented databases, such as spatial databases, time-series databases, text databases, and multimedia databases.

Flat files: Flat files are actually the most common data source for data mining algorithms, especially at the research level. Flat files are simple data files in text or binary format with a structure known by the data mining algorithm to be applied. The data in these files can be transactions, time-series data, scientific measurements, etc.

Relational Databases: a relational database consists of a set of tables containing either values of entity attributes, or values of attributes from entity relationships. Tables have columns and rows, where columns represent attributes and rows represent tuples. A tuple in a relational table corresponds to either an object or a relationship between objects and is identified by a set of attribute values representing a unique key. In following figure it presents some relations Customer, Items, and Borrow representing business activity in a video store. These relations are just a subset of what could be a database for the video store and is given as an example.

The most commonly used query language for relational database is SQL, which allows retrieval and manipulation of the data stored in the tables, as well as the calculation of aggregate functions such as average, sum, min, max and count. For instance, an SQL query to select the videos grouped by category would be:

SELECT count(*) FROM Items WHERE type=video GROUP BY category.
Data mining algorithms using relational databases can be more versatile than data mining algorithms specifically written for flat files, since they can take advantage of the structure inherent to relational databases. While data mining can benefit from SQL for data selection, transformation and consolidation, it goes beyond what SQL could provide, such as predicting, comparing, detecting deviations, etc.

Data mining Functionalities/Data mining tasks

Data mining functionalities are used to specify the kind of patterns to be found in data mining tasks. In general, data mining tasks can be classified into two categories:
• Descriptive
• Predictive

Descriptive mining tasks characterize the general properties of the data in the database.
Predictive mining tasks perform inference on the current data in order to make predictions.

1. Concept/Class description: characterization and discrimination
Data can be associated with classes or concepts. It describes a given set of data in a concise and

summarative manner, presenting interesting general properties of the data. These descriptions can be derived via
1. data characterization, by summarizing the data of the class under study (often called the target class)
2. data discrimination, by comparison of the target class with one or a set of comparative Classes

Data characterization
It is a summarization of the general characteristics or features of a target class of data.
Example: A data mining system should be able to produce a description summarizing the characteristics of a student who has obtained more than 75% in every semester; the result could be a general profile of the student.

Data Discrimination is a comparison of the general features of target class data objects with the general features of objects from one or a set of contrasting classes.
Example : The general features of students with high GPAs may be compared with the general features of students with low GPAs. The resulting description could be a general comparative profile of the students such as 75% of the students with high GPAs are fourth-year computing science students while 65% of the students with low GPAs are not.
The output of data characterization can be presented in various forms. Examples include pie charts, bar charts, curves, multidimensional data cubes, and multidimensional tables, including crosstabs. The resulting descriptions can also be presented as generalized relations, or in rule form called characteristic rules. Discrimination descriptions expressed in rule form are referred to as discriminate rules.

2. Classification and prediction
Classification:
Classification:
□□It predicts categorical class labels
□□It classifies data (constructs a model) based on the training set and the values (class labels) in a
classifying attribute and uses it in classifying new data
□□Typical Applications
o credit approval
o target marketing
o medical diagnosis
o treatment effectiveness analysis
Classification can be defined as the process of finding a model (or function) that describes and distinguishes data classes or concepts, for the purpose of being able to use the model to predict the class of objects whose class label is unknown. The derived model is based on the analysis of a set of training data (i.e., data objects whose class label is known).
Example:
An airport security screening station is used to deter mine if passengers are potential terrorist or criminals. To do this, the face of each passenger is scanned and its basic pattern(distance between eyes, size, and shape of mouth, head etc) is identified. This pattern is compared to entries in a database to see if it matches any patterns that are associated with known offenders.

Prediction:
Find some missing or unavailable data values rather than class labels referred to as prediction. Although prediction may refer to both data value prediction and class label prediction, it is usually confined to data
value prediction and thus is distinct from classification. Prediction also encompasses the identification of distribution trends based on the available data.
Example:
Predicting flooding is difficult problem. One approach is uses monitors placed at various points in the river. These monitors collect data relevant to flood prediction: water level, rain amount, time, humidity etc. These water levels at a potential flooding point in the river can be predicted based on the data collected by the sensors upriver from this point. The prediction must be made with respect to the time the data were collected.

3. Cluster analysis
Clustering analyzes data objects without consulting a known class label. The objects are clustered or grouped based on the principle of maximizing the intra class similarity and minimizing the interclass similarity.

Each cluster that is formed can be viewed as a class of objects. Clustering can also facilitate taxonomy formation, that is, the organization of observations into a hierarchy of classes that group similar events together.

4.Outlier Analysis

Outliers can occur by chance in any distribution, but they often indicate either measurement error or that the population has a heavy-tailed distribution. In the former case one wishes to discard them or use statistics that are robust to outliers, while in the latter case they indicate that the distribution has high kurtosis and that one should be very cautious in using tools or intuitions that assume a normal distribution. A frequent cause of outliers is a mixture of two distributions, which may be two distinct sub-populations, or may indicate 'correct trial' versus 'measurement error'; this is modeled by a mixture model. Outlier points can therefore indicate faulty data, erroneous procedures, or areas where a certain theory might not be valid.

5. Classification and Regression for Predictive Analysis

- Classification is the process of learning a model that describes different classes of data. The classes are predetermined.
- Example: In a banking application, customers who apply for a credit card may be classify as a “good risk”, a “fair risk” or a “poor risk”. Hence, this type of activity is also called *supervised learning*.
- Once the model is built, then it can be used to classify new data.

Regression

- Regression is a method used to predict continuous values for given input. Regression analysis can be used to model the relationship between one or more *independent* or *predictor* variables and one or more *response* or *dependent* variables.

6.Mining Frequent Patterns, Associations, and Correlations

Frequent patterns are the patterns that occur frequently in the data. Patterns can include itemsets, sequences and subsequences. A frequent itemset refers to a set of items that often appear together in a transactional data set.

ex: bread and milk

Association Analysis

Association Analysis is used for finding frequent patterns, associations, correlations, or causal structures among sets of items or objects in transaction databases, relational databases, and other information repositories.

Association analysis is a method for discovering interesting relations between variables in large databases. It is intended to identify strong rules discovered in databases using different measures of interestingness. Based on the concept of strong rules, For example, the rule $\{\text{onions}, \text{potatoes}\} \Rightarrow \{\text{burger}\}$ found in the sales data of a supermarket would indicate that if a customer buys onions and potatoes together, they are likely to also buy hamburger meat. Such information can be used as the basis for decisions about marketing activities such as, e.g., promotional pricing or product placements.

Classification of data mining systems

There are many data mining systems available or being developed. Some are specialized systems dedicated to a given data source or are confined to limited data mining functionalities, other are more versatile and comprehensive. Data mining systems can be categorized according to various criteria among other classification are the following:

· **Classification according to the type of data source mined:**

This classification categorizes data mining systems according to the type of data handled such as spatial data, multimedia data, time-series data, text data, World Wide Web, etc.

· **Classification according to the data model drawn on:** this classification categorizes data mining systems based on the data model involved such as relational database, object-oriented database, data warehouse, transactional, etc.

· **Classification according to the king of knowledge discovered:** this classification categorizes data mining systems based on the kind of knowledge discovered or data mining functionalities, such as characterization, discrimination, association, classification, clustering, etc. Some systems tend to be comprehensive systems offering several data mining functionalities together.

· **Classification according to mining techniques used:** Data mining systems employ and provide different techniques. This classification categorizes data mining systems according to the data analysis approach used such as machine learning, neural networks, genetic algorithms, statistics, visualization, database oriented or data warehouse-oriented, etc. The classification can also take into account the degree of user interaction involved in the data mining process such as query-driven systems, interactive exploratory systems, or autonomous systems. A comprehensive system would provide a wide variety of data mining techniques to fit different situations and options, and offer different degrees of user interaction.

Major issues in data mining

Major issues in data mining is regarding mining methodology, user interaction, performance, and diverse data types:

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

Incorporation of background knowledge: Background knowledge, or information regarding the domain under study, may be used to guide the discovery patterns. Domain knowledge related to databases, such as integrity constraints and deduction rules, can help focus and speed up a data mining process, or judge the interestingness of discovered patterns.

Data mining query languages and ad-hoc data mining: Knowledge in Relational query languages (such as SQL) required since it allow users to pose ad-hoc queries for data retrieval.

2. User interaction Issue

- Due to Data mining query languages and ad-hoc mining. User will face problems in getting accurate results in Data Mining.
  Due to the Expression and visualization of data mining results, the managers may confuse by the output produced by data mining process. So the Discovered knowledge should be expressed in

high-level languages, visual representations, so that the knowledge can be easily understood and directly usable by humans

- Due to Interactive mining of knowledge at multiple levels of abstraction, Data Mining tools may provide inaccurate output.

3. Diversity of database types Issue:

Handling of relational and complex types of data: Since relational databases and data warehouses are widely used, the development of efficient and effective data mining systems for such data is important.

_ Mining information from heterogeneous databases and global information systems: Local and
wide-area computer networks (such as the Internet) connect many sources of data, forming huge, distributed, and heterogeneous databases. The discovery of knowledge from different sources of structured, semi-structured, or unstructured data with diverse data semantics poses great challenges to
data mining.

4.Efficiency and Scalability Issue

As data is increasing every day, efficiency and scalability are especially critical issue.

Efficiency and Scalability of Data Mining Algorithms :

Data Mining algorithms must be efficient and scalable in order to effectively extract information from huge amounts of data in many data repositories. Efficiency and scalability are the key criteria in the development of new data mining algorithm.

Parallel, distributed, and incremental mining algorithms :

Since cloud computing and cluster computing use computers in a distributed and collaborative way to tackle very large scale computational tasks, Data mining algorithms should support parallel and distributed computing.

5.Data Mining and Society

Social Impacts of Data Mining :

The improper disclosure or use of data leads to the potential violation of individual privacy and data protection rights. We need to guard against the misuse of data mining techniques to society.

Privacy-preserving data mining:

Data Mining will help scientific discovery, business management and Security protection. However, it poses the risk of disclosing an individual's personal information. So we need to find the measures for preserving the privacy of individual's personal information.

Introduction to Data warehouse

A data warehouse is a collection of data marts representing historical data from different operations in the company. This data is stored in a structure optimized for querying and data analysis as a data warehouse. Table design, dimensions and organization should be consistent throughout a data warehouse so that reports or queries across the data warehouse are consistent. A data warehouse can also be viewed as a database for historical data from different functions within a company.

A data warehouse is a subject-oriented, integrated, time-variant and non-volatile collection of data in support of management's decision making process".

Subject Oriented: Data that gives information about a particular subject instead of about a company's ongoing operations.

Integrated: Data that is gathered into the data warehouse from a variety of sources and merged into a coherent whole.

Time-variant*:* All data in the data warehouse is identified with a particular time period.

Non-volatile*:* Data is stable in a data warehouse. More data is added but data is never removed.

This enables management to gain a consistent picture of the business. It is a single, complete and consistent store of data obtained from a variety of different sources made available to end users in what they can understand and use in a business context. It can be used for decision Support used to manage and control business and used by managers and end-users to understand the business and make judgments.
Data Warehousing is an architectural construct of information systems that provides users with current and historical decision support information that is hard to access or present in traditional operational data stores.

Enterprise Data warehouse*:* It collects all information about subjects (*customers, products, sales, assets, personnel*) that span the entire organization.

Data Mart: Departmental subsets that focus on selected subjects. A data mart is a segment of a data warehouse that can provide data for reporting and analysis on a section, unit, department or operation in the company, e.g. sales, payroll, production. Data marts are sometimes complete individual data warehouses which are usually smaller than the corporate data warehouse.

Decision Support System (DSS): Information technology to help the knowledge worker (executive, manager, and analyst) makes faster & better decisions.

Metadata: Data about data. Containing location and description of warehouse system components: names, definition, structure.

Benefits of data warehousing

- Data warehouses are designed to perform well with aggregate queries running on large amounts of data.
- The structure of data warehouses is easier for end users to navigate, understand and query against unlike the relational databases primarily designed to handle lots of transactions.
- Data warehouses enable queries that cut across different segments of a company's operation. E.g. production data could be compared against inventory data even if they were originally stored in different databases with different structures.
- Queries that would be complex in very normalized databases could be easier to build and maintain in data warehouses, decreasing the workload on transaction systems.
- Data warehousing is an efficient way to manage and report on data that is from a variety of sources, non uniform and scattered throughout a company.
- Data warehousing is an efficient way to manage demand for lots of information from lots of users.
- Data warehousing provides the capability to analyze large amounts of historical data for nuggets of wisdom that can provide an organization with competitive advantage.

Differences between Operational Database Systems and Data Warehouses

Because most people are familiar with commercial relational database systems, it is easy to understand what a data warehouse is by comparing these two kinds of systems.
The major task of on-line operational database systems is to perform on-line transaction and query processing. These systems are called on-line transaction processing (OLTP) systems.

	OLTP	**OLAP**
users	clerk, IT professional	knowledge worker
function	day to day operations	decision support
DB design	application-oriented	subject-oriented
data	current, up-to-date detailed, flat relational isolated	historical, summarized, multidimensional integrated, consolidated
usage	repetitive	ad-hoc
access	read/write index/hash on prim. key	lots of scans
unit of work	short, simple transaction	complex query
# records accessed	tens	millions
#users	thousands	hundreds
DB size	100MB-GB	100GB-TB
metric	transaction throughput	query throughput, response

They cover most of the day-to-day operations of an organization, such

as purchasing, inventory, manufacturing, banking, payroll, registration, and accounting. Data warehouse systems, on the other hand, serve users or knowledge workers in the role of data analysis and decision making. Such systems can organize and present data in various formats in order to accommodate the diverse needs of the different users. These systems are known as on-line analytical processing (OLAP) systems.
The major distinguishing features between OLTP and OLAP are summarized as follows:
Users and system orientation: An OLTP system is *customer-oriented* and is used for transaction and query processing by clerks, clients, and information technology professionals. An OLAP systemis *market-oriented* and is used for data analysis by knowledge workers, including managers, executives, and analysts.

Data contents: An OLTP systemmanages current data that, typically, are too detailed to be easily used for decision making. An OLAP system manages large amounts of historical data, provides facilities for summarization and aggregation, and stores and manages information at different levels of granularity. These features make the data easier to use in informed decision making.

Database design: An OLTP system usually adopts an entity-relationship (ER) data model and an application-oriented database design.An OLAP systemtypically adopts either a *star* or *snowflake* model (to be discussed in Section 3.2.2) and a subjectoriented database design.

View: An OLTP system focuses mainly on the current data within an enterprise or department, without referring to historical data or data in different organizations. In contrast, an OLAP system often spans multiple versions of a database schema, due to the evolutionary process of an organization.
Access patterns: The access patterns of an OLTP system consist mainly of short, atomic transactions. Such a system requires concurrency control and recovery mechanisms.

Why Have a Separate DataWarehouse?
Because operational databases store huge amounts of data, you may wonder, *"why not perform on-line analytical processing directly on such databases instead of spending additional time and resources to construct a separate data warehouse?"* A major reason for such a separation is to help promote the *high performance of both system.*

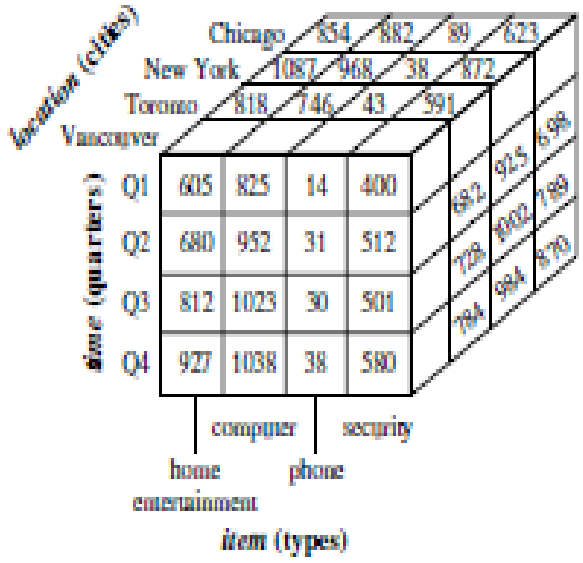
Datawarehouse : A Multidimensional Data Model

Data warehouses and OLAP tools are based on a multidimensional data model. This model views data in the form of a *data cube*. In this section, you will learn how data cubes model *n*-dimensional data. You will also learn about concept hierarchies and how they can be used in basic OLAP operations to allow interactive mining at multiple levels of abstraction.

From Tables and Spreadsheets to Data Cubes

"What is a data cube?" A data cube allows data to be modeled and viewed in multiple dimensions. It is defined by dimensions and facts.

In general terms, dimensions are the perspectives or entities with respect to which an organization wants to keep records. For example, *AllElectronics* may create a *sales* data warehouse in order to keep records of the store's sales with respect to the dimensions *time*, *item*, *branch*, and *location*. These dimensions allow the store to keep track of things like monthly sales of items and the branches and locations.
Each dimension may have a table associated with it, called a dimension table, which further describes the dimension. For example,a dimension table for *item* may contain the attributes *item name*, *brand*, and *type*.
Dimension tables can be specified by users or experts, or automatically generated and adjusted based on data distributions.
A multidimensional data model is typically organized around a central theme, like *sales*, for instance. This theme is represented by a fact table. Facts are numerical measures.



3.1 A 3-D data cube representation of the data in Table 3.3, according to the dimensions *time*, *item*, and *location*. The measure displayed is *dollars_sold* (in thousands).

Stars, Snowflakes, and Fact Constellations:
Schemas for Multidimensional Databases
The entity-relationship data model is commonly used in the design of relational databases, where a database schema consists of a set of entities and the relationships between them. Such a data model is appropriate for on-line transaction processing.
A data warehouse, however, requires a concise, subject-oriented schema that facilitates on-line data analysis. The most popular data model for a data warehouse is a multidimensional model. Such a model can exist in the form of a star schema, a snowflake schema, or a fact constellation Schema.

Star schema: The most common modeling paradigm is the star schema, in which the data warehouse contains (1) a large central table (fact table) containing the bulk of the data, with no redundancy, and (2) a set of smaller attendant tables (dimension tables), one for each dimension. The schema graph resembles a starburst, with the dimension tables displayed in a radial pattern around the central fact table.

Star schema. A star schema for *AllElectronics* sales is shown in Figure 3.4. Sales are considered along four dimensions,namely, *time, item, branch,* and *location*.The schemacontains a central fact table for *sales* that contains keys to each of the four dimensions, along with two measures: *dollars sold* and *units sold*.To minimize the size of the fact table, dimension identifiers (such as *time key* and *item key*) are system-generated identifiers.

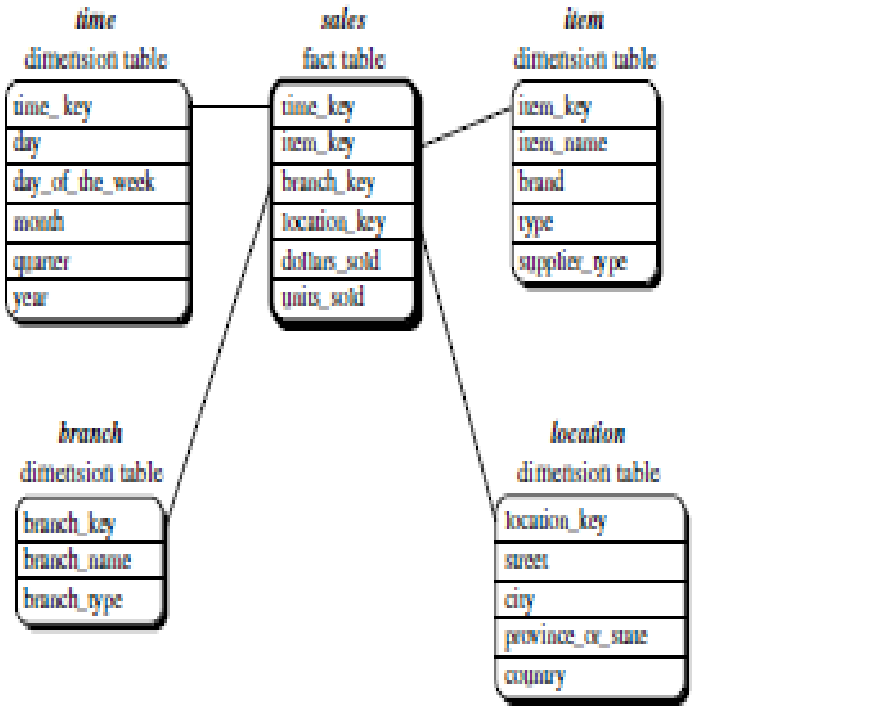


Figure 3.4 Star schema of a data warehouse for sales.

The major difference between the snowflake and star schema models is that the dimension tables of the snowflake model may be kept in normalized form to reduce redundancies. Such a table is easy to maintain and saves storage space. However, this saving of space is negligible in comparison to the typical magnitude of the fact table. Furthermore, the snowflake structure can reduce the effectiveness of browsing, since more joins will be needed to execute a query. Consequently, the system performance

may be adversely impacted. Hence, although the snowflake schema reduces redundancy, it is not as popular as the star schema in data warehouse design.

Snowflake schema. A snowflake schema for *AllElectronics* sales is given in Figure 3.5. Here, the *sales* fact table is identical to that of the star schema in Figure 3.4. The main difference between the two schemas is in the definition of dimension tables. The single dimension table for *item* in the star schema is normalized in the snowflake schema, resulting in new *item* and *supplier* tables. For example, the *item* dimension table now contains the attributes *item key*, *item name*, *brand*, *type*, and *supplier key*, where *supplier key* is linked to the *supplier* dimension table, containing *supplier key* and *supplier type* information. Similarly, the single dimension table for *location* in the star schema can be normalized into two new tables: *location* and *city*. The *city key* in the new *location* table links to the *city* dimension. Notice that further normalization can be performed on *province or state* and *country* in the snowflake schema shown in Figure 3.5, when desirable.

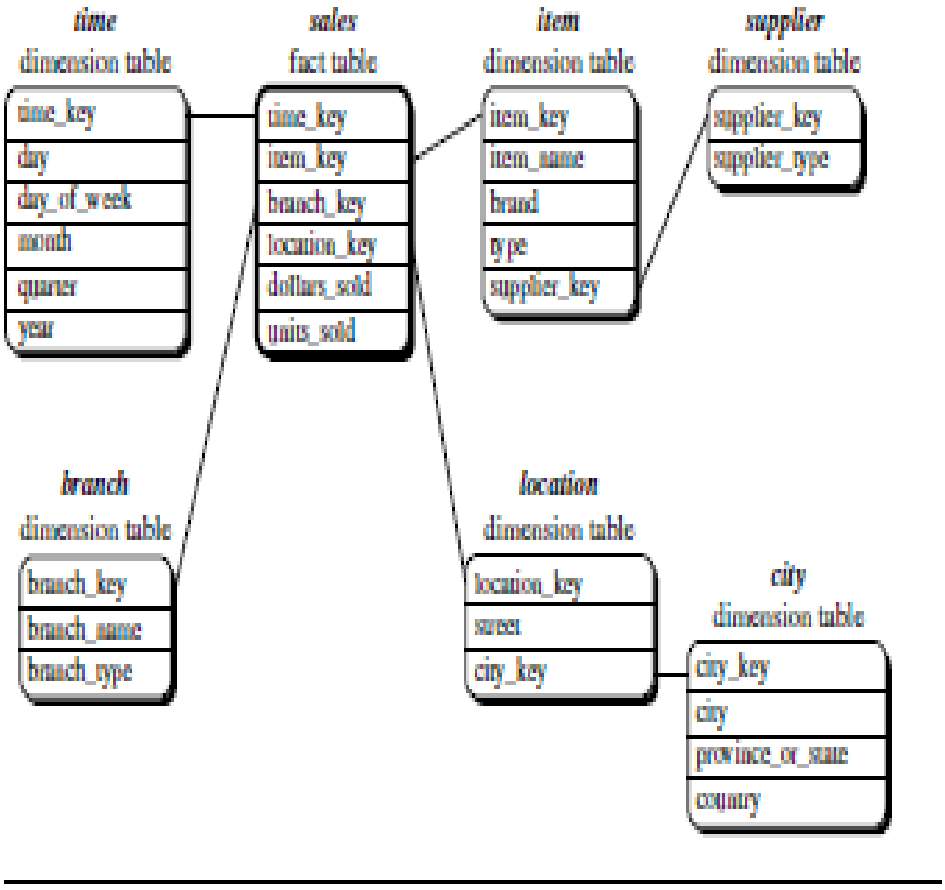


Figure 3.5 Snowflake schema of a data warehouse for sales.

Fact constellation. A fact constellation schema is shown in Figure 3.6. This schema specifies two fact tables, *sales* and *shipping*. The *sales* table definition is identical to that of

the star schema (Figure 3.4). The *shipping* table has five dimensions, or keys: *item key*, *time key*, *shipper key*, *from location*, and *to location*, and two measures: *dollars cost* and *units shipped*. A fact constellation schema allows dimension tables to be shared between fact tables. For example, the dimensions tables for *time*, *item*, and *location* are shared between both the *sales* and *shipping* fact tables.

OLAP Operations in the Multidimensional Data Model

"How are concept hierarchies useful in OLAP?" In the multidimensional model, data are organized into multiple dimensions, and each dimension contains multiple levels of abstraction defined by concept hierarchies. This organization provides users with the flexibility to view data from different perspectives. A number of OLAP data cube operations exist to materialize these different views, allowing interactive querying and analysis of the data at hand. Hence, OLAP provides a user-friendly environment for interactive data analysis.

Roll-up: The roll-up operation (also called the *drill-up* operation by some vendors) performs aggregation on a data cube, either by *climbing up a concept hierarchy* for a dimension or by *dimension reduction*. Figure 3.10 shows the result of a roll-up operation performed on the central cube by climbing up the concept hierarchy for *location* given in Figure 3.7. This hierarchy was defined as the total order "*street* < *city* < *province or state* < *country*." The roll-up operation shown aggregates the data by ascending the *location* hierarchy from the level of *city* to the level of *country*. In other words, rather than grouping the data by city, the resulting cube groups the data by country.

Drill-down: Drill-down is the reverse of roll-up. It navigates from less detailed data to more detailed data. Drill-down can be realized by either *stepping down a concept hierarchy* for a dimension or *introducing additional dimensions*. Figure 3.10 shows the result of a drill-down operation performed on the central cube by stepping down a concept hierarchy for *time* defined as "*day* < *month* < *quarter* < *year*." Drill-down occurs by descending the *time* hierarchy fromthe level of *quarter* to the more detailed level of *month*. The resulting data cube details the total sales per month rather than summarizing them by quarter.

Pivot (rotate): *Pivot* (also called *rotate*) is a visualization operation that rotates the data axes in view in order to provide an alternative presentation of the data. Figure 3.10 shows a pivot operation where the *item* and *location* axes in a 2-D slice are rotated.

DataWarehouse Architecture

Steps for the Design and Construction of DataWarehouses

This subsection presents a business analysis framework for data warehouse design. The basic steps involved in the design process are also described.

The Design of a DataWarehouse: A Business Analysis Framework

"*What can business analysts gain from having a data warehouse?*" First, having a data warehouse may provide a *competitive advantage* by presenting relevant information from which to measure performance and make critical adjustments in order to help win over competitors. Second, a data warehouse can enhance business *productivity* because it is able to quickly and efficiently gather information that accurately describes the organization. Third, a data warehouse facilitates *customer relationship management* because it provides a consistent view of customers and items across all lines of business, all departments, and all markets. Finally, a data warehouse may bring about *cost reduction* by tracking trends, patterns, and exceptions over long periods in a consistent and reliable manner.

Four different views regarding the design of a data warehousemust be considered: the ***top-down view***, the *data source view*, the *data warehouse view*, and the *business query view*.
The top-down view allows the selection of the relevant information necessary for the data warehouse. This information matches the current and future business needs.
The data source view exposes the information being captured, stored, and managed by operational systems. This information may be documented at various levels of detail and accuracy, from individual data source tables to integrated data source tables. Data sources are often modeled by traditional data modeling techniques, such as the entity-relationship model or CASE (computer-aided software engineering) tools.

The data warehouse view includes fact tables and dimension tables. It represents the information that is stored inside the data warehouse, including precalculated totals and counts, as well as information regarding the source, date, and time of origin, added to provide historical context.
Finally, the business query view is the perspective of data in the data warehouse from the viewpoint of the end user.

A Three-Tier DataWarehouse Architecture

Data warehouses often adopt a three-tier architecture, as presented in Figure 3.12.
1. The bottom tier is a warehouse database server that is almost always a relational database system. Back-end tools and utilities are used to feed data into the bottom tier from operational databases or other external sources (such as customer profile information provided by external consultants). These tools and utilities performdata extraction, cleaning, and transformation (e.g., to merge similar data from different sources into a unified format, as well as load and refresh functions to update the data warehouse.

2. The middle tier is an OLAP server that is typically implemented using either (1) a relational OLAP (ROLAP) model, that is, an extended relational DBMS that maps operations on multidimensional data to standard relational operations; or

(2) a multidimensional OLAP (MOLAP) model, that is, a special-purpose server that directly implements multidimensional data and operations.

3. The top tier is a front-end client layer, which contains query and reporting tools, analysis tools, and/or data mining tools (e.g., trend analysis, prediction, and so on). Fromthe architecture point of view, there are three data warehouse models: the *enterprise warehouse*, the *data mart*, and the *virtual warehouse*.

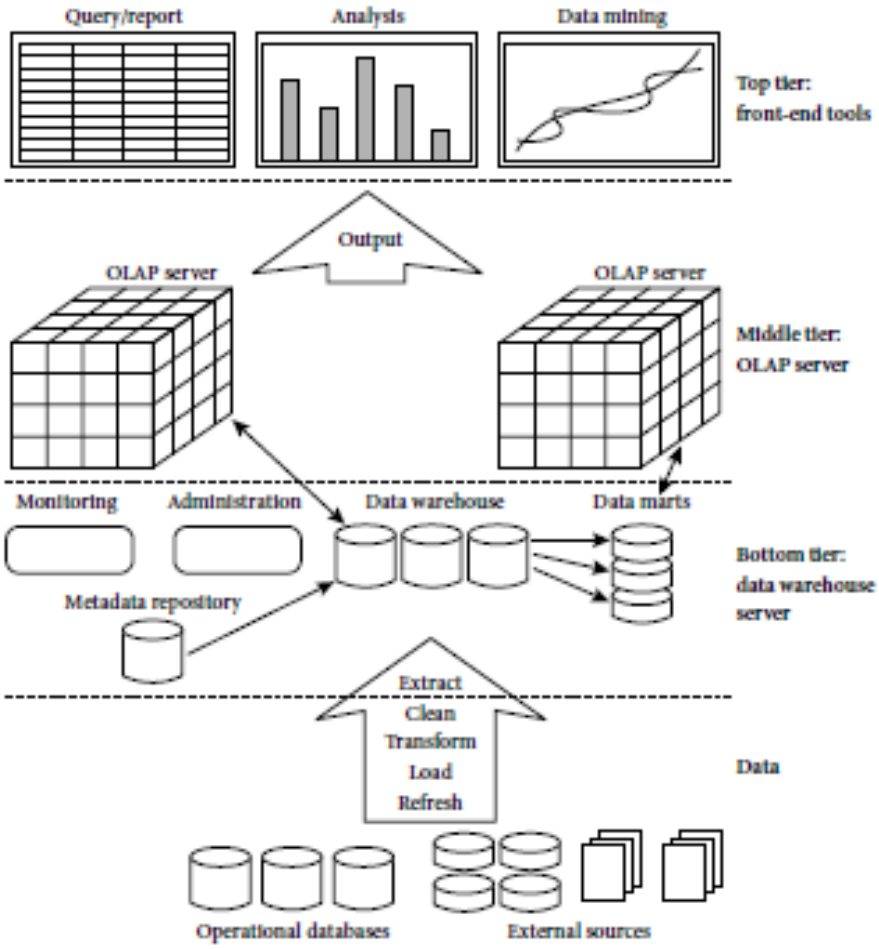


Figure 3.12 A three-tier data warehousing architecture.

Enterprise warehouse: An enterprise warehouse collects all of the information about subjects spanning the entire organization. It provides corporate-wide data integration, usually from one or more operational systems or external information providers, and is cross-functional in scope.

Data mart: A data mart contains a subset of corporate-wide data that is of value to a specific group of users. The scope is confined to specific selected subjects. For example,

a marketing data mart may confine its subjects to customer, item, and sales.

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

DataWarehouse Back-End Tools and Utilities

Data warehouse systems use back-end tools and utilities to populate and refresh their data (Figure 3.12). These tools and utilities include the following functions:
Data extraction,which typically gathers data frommultiple, heterogeneous, and external Sources.

Data cleaning, which detects errors in the data and rectifies them when possible
Data transformation, which converts data from legacy or host format to warehouse Format Load, which sorts, summarizes, consolidates, computes views, checks integrity

Metadata Repository
Metadata are data about data. When used in a data warehouse, metadata are the data that define warehouse objects. Metadata are created for the data names and definitions of the given warehouse. Additional metadata are created and captured for time stamping any extracted data, the source of the extracted data, and missing fields that have been added by data cleaning or integration processes.

A metadata repository should contain the following:

A description of *the structure of the data warehouse*, which includes the warehouse schema, view, dimensions, hierarchies, and derived data definitions, as well as data mart locations and contents.

Operational metadata, which include data lineage (history of migrated data and the sequence of transformations applied to it), currency of data (active, archived, or purged), and monitoring information (warehouse usage statistics, error reports, and audit trails).

The algorithms used for summarization, which include measure and dimension definition algorithms, data on granularity, partitions, subject areas, aggregation, summarization, and predefined queries and reports

Types of OLAP Servers: ROLAP versus MOLAP versus HOLAP

Logically, OLAP servers present business users with multidimensional data from data warehouses or data marts, without concerns regarding how or where the data are stored.

However, the physical architecture and implementation of OLAP servers must consider data storage issues. Implementations of a warehouse server for OLAP processing include the following:

Relational OLAP (ROLAP) servers: These are the intermediate servers that stand in between a relational back-end server and client front-end tools. They use a *relational or extended-relationalDBMS* to store and manage warehouse data, and OLAP middleware to support missing pieces. ROLAP servers include optimization for each DBMS back end, implementation of aggregation navigation logic, and additional tools and services.

Multidimensional OLAP (MOLAP) servers: These servers support multidimensional views of data through *array-based multidimensional storage engines*. They mapmultidimensional
views directly to data cube array structures. The advantage of using a data cube is that it allows fast indexing to precomputed summarized data.

Hybrid OLAP (HOLAP) servers: The hybrid OLAP approach combines ROLAP and MOLAP technology, benefiting from the greater scalability of ROLAP and the faster computation of MOLAP. For example, a HOLAP server may allow large volumes of detail data to be stored in a relational database, while aggregations are kept in a separate MOLAP store. The Microsoft SQL Server 2000 supports a hybrid OLAP server.

Specialized SQL servers: To meet the growing demand of OLAP processing in relational databases, some database systemvendors implement specialized SQL servers that provide advanced query language and query processing support for SQL queries over star and snowflake schemas in a read-only environment.

Part B 2 Mark Questions

1.Define Data warehouse.
2.What are the advantages of Aprori Algorithm?
3.What is Bayesian belief Network?
4.What is Text Mining?
5.What are the four differences between OLTP and OLAP?

PART-C 8 MARK Questions

1.Explain the architecture of a Data Mining System.
2.Discuss Data warehouse implementation process in detail.
3.Explain the functionalities of data mining.
4.Explain the architecture of Data warehouse.
5.Discuss the various classification of Data Mining Systems.
6.Explain the multidimensional data model of Data warehouse.
7.Explain the steps in Knowledge Discovery from Data (KDD).
8.Discuss the differences between OLTP and OLAP.
9.Explain the various steps of Knowledge Discovery process.

10.Discuss the major issues in Data Mining.

Questions	opt1	opt2	opt3	opt4	opt5	opt6	Answer
_____ model makes a prediction about values of data using known results found from different data.	predictive	descriptive	preference	process			predictive
_____ model identifies patterns or relationships in data.	predictive	descriptive	process	preference			descriptive
_____ maps data into predefined groups or classes	regression	classification	clustering	prediction			classification
_____ is used to map a data item to a real valued prediction variable.	regression	classification	clustering	prediction			regression
Classification of input patterns by using its similarity to the predefined classes is called____	prediction	classification	pattern recognition	regression			pattern recognition
A special type of clustering is called _____	classification	segmentation	regression	prediction			segmentation
_____ maps data into subsets with associated simple description.	classification	segmentation	recognition	summarization			summarization
Which rule is used to identify the specific type of data association	delta rule	association rule	classification rule	segmentation rule			association rule
Which is used to determine sequential patterns in data?	sequence discovery	segmentation	regression	random analysis			sequence discovery
_____ is the process of finding useful information and patterns in data	datamining	KDD	selection	transformation			KDD
_____ is the use of algorithms to extract the information and pattern derived by the knowledge discovery	datamining	KDD	selection	transformation			datamining
The KDD process is often said to be _____	trivial	nontrivial	significant	consequential			nontrivial
Extreme values that occur infrequently, may actually be removed is called _____	outliers	outlayers	statistics	frequency			outliers
Traditional graph structure is a _____ visualization techniqe	graphical	geometric	icon-based	pixel-based			graphical
The major goal of _____ is to be able to describe the result in meaningful manner.	datamining	KDD	selection	transformation			KDD
_____ techniques often involve sophisticated multimedia and graphics presentations.	summarization	conceptualization	visualization	decription			visualization

Describing a large database can be viewed as using _____ to help uncover hidden information about the data	approximation	search	induction	compression			approximation
_____ is used to proceed from very specific knowledge to more general information.	approximation	search	induction	compression			induction
_____ occurs when the model does not fit future states.	outliers	interception	overfitting	qureying			overfitting
The problem of increasing over all complexity and decrease the efficiency of an algorithms is _____	Single dimensionality	Multi-Dimensionality	Dimensionality curse	Recursion			Dimensionality curse
What is the expansion of ROI	return on investment	result of invension	research oriented investment	return of interest			return on investment
Find out predictive model datamining task from the followings	clustering	summarization	association rule	regression			regression
Find out descriptive model datamining task from the followings	classification	sequence discovery	prediction	regression			sequence discovery
Flooding, speech recognition, machine learning are_____	classification applicatina	prediction applications	clustering application	regression applications			prediction applications
Link analysis alternatively referred to as a _____	rule analysis	affinity analysis	master analysis	report analysis			affinity analysis
Similarity measure used to identify the _____ of different items in the database.	unlikeness	difference	alikeness	matches			alikeness
Distance measures used to identify the _____ of different items in the database.	unlikeness	matches	alikeness	difference			unlikeness
Decision tree is otherwise called as _____ tree	clustering	classification	regression	correlation			classification
The _____ phase might remove redundant comparisons or remove subtrees to achieve better performance.	classification	clustering	splitting	pruning			pruning
The _____ technique to building a decision tree is based on information theory and attempts to minimize the espected number of comparisons.	bayes rule	backpropagation	ID3	perceptron			ID3

Expansion of CART	clustering and regression trees	classification and regression technique	classification and regression trees	clustering and regression technique			classification and regression trees
_____ is a technique that generates a binary decision tree.	perceptron	CART	bayes rule	backpropagation			CART
In _____ learning, the NN starting state is modified based on feedback of its performance with the training set of data.	unsupervised	nonsupervised	desupervised	supervised			supervised
_____ is a learning technique that adjusts weights in the NN by propagating weight changes backward from the sink to the source nodes.	backpropagation	propagation	regression	correlation			backpropagation
A _____ is a class of function whose value decreases or increases with the distance from a central point.	propagation	association	radial basis function	bayes rule			radial basis function
A _____ is a single neuron with multiple inputs and one output.	CART	ID3	bayes rule	perceptron			perceptron
The _____ contains a predicate that can be evaluated as true or false against each tuple in the database.	consequent	antecedent	propagation	probability			antecedent
A _____is a graph on a 2D axis of points representing the relationship between x and y values.	box plot	histogram	scatter diagram	pie chart			scatter diagram
_____ is a technique to estimate the likelihood of a property given the set of data as evidence or input.	bayes rule	ID3	backpropagation	CART			bayes rule
The initial hypothesis is called the _____ hypothesis	start	source	alternative	null			null
Rejection of the null hypothesis causes _____hypothesis.	not null	alternative	null	sink			alternative
The _____ is a procedure used to test the association between 2 observed variable values and to determine whether the values are statistically significant.	prediction	regression	chi-squared statistics	bayesian			chi-squared statistics

_____ is used to examine the degree to which the values for 2 variables behave similarly.	correlation	regression	similarity	dissimilarity			correlation
Decision tree uses a _____ technique to split the problem search space into subsets.	splitting	divide and conquer	pruning	entropy			divide and conquer
_____ can be viewed as a directed graph with source, sink and internal nodes	decision tree	clustering	classification	neural network			neural network
The _____ of an estimator is the difference between the espected value of the estimator and the actual value	unbias	bias	predict	unpredict			bias
An unbiased estimator is one whose bias is _____	-1	0	1	2			0
The _____ is defined as the expected value of the squared difference between the estimate and the actual value	root mean square	root mean square error	mean squared error	jackknife			mean squared error
the _____ estimate is obtained by omiting one value from the set of observed values	jackknife	mean squared error	root mean square	root mean square error			jackknife
_____ defined as a value proportional to the actual probability that with a specific distribution the given sample exists	root mean square error	jackknife	likelihood	mean squared error			likelihood
The _____ algorithm is an approach that solves the estimation problem with incomplete data	genetic algorithm	expectation maximization	crossover	rule-based class			expectation maximization
The _____ diagram shows the distribution of the data	scatter	box plot	histogram	graph			histogram
In box plot the total range of the data values is divided into four equal parts called _____	quartiles	quarter	quartales	quad			quartiles
In bayes rule h stands for _____	hyperbolic	hypertension	hyperlink	hypothesis			hypothesis
In bayes rule x stands for _____	alternate event	observable event	extensible event	expert event			observable event
_____ testing attempts to find a model that explains the observed data by first creating a hypothesis and then testing against the data	regression	correlation	logarithmic	hypothesis			hypothesis

Alternative hypothesis is denoted by _____	H_0	H_1	H_a	H_n			H_1
In chi-square, _____ tables are used to evaluate the actual value in order to determine its significance	statistical	logarithmic	algebric	analytical			statistical
Both regression and correlation are used to evaluate the strength of a relationship between _____ variables	zero	one	two	more			two
Regression is used to predict future values based on past values by fitting a set of points to a _____	line	square	curve	ellipse			curve

UNIT II

Data Preprocessing: Needs Preprocessing the Data - Data Cleaning - Data Integration and Transformation - Data Reduction - Discretization and Concept Hierarchy Generation - Online Data Storage - Preparing Data for Mining: Variable Measures.

DATA PREPROCESSING

Needs for Preprocess the Data

Imagine that you are a manager at *AllElectronics* and have been charged with analyzing the company's data with respect to the sales at your branch. You immediately set out to perform this task. You carefully inspect the company's database and data warehouse, identifying and selecting the attributes or dimensions to be included in your analysis such as *item*, *price*, and *units sold*. Alas! You notice that several of the attributes for various tuples have no recorded value. For your analysis, you would like to include information as to whether each item purchased was advertised as on sale, yet you discover that this information has not been recorded. Furthermore, users of your database system have reported errors, unusual values, and inconsistencies in the data recorded for some transactions. In other words, the data you wish to analyze by data mining techniques are incomplete (lacking attribute values or certain attributes of interest, or containing only aggregate data), noisy (containing errors, or *outlier* values that deviate from the expected), and inconsistent (e.g., containing discrepancies in the department codes used to categorize items).Welcome to the real world!

Incomplete, noisy, and inconsistent data are commonplace properties of large real world databases and data warehouses. Incomplete data can occur for a number of reasons. Attributes of interest may not always be available, such as customer information for sales transaction data. Other data may not be included simply because it was not considered important at the time of entry. Relevant data may not be recorded due to a misunderstanding, or because of equipment malfunctions. Data that were inconsistent with other recorded data may have been deleted.
There are many possible reasons for noisy data (having incorrect attribute values). The data collection instruments usedmay be faulty. Theremay have been human or computer errors occurring at data entry. Errors in data transmission can also occur. There may be technology limitations, such as limited buffer size for coordinating synchronized data transfer and consumption. Incorrect data may also result from inconsistencies in naming conventions or data codes used, or inconsistent formats for input fields, such as *date*. Duplicate tuples also require data cleaning.

Data cleaning routines work to "clean" the data by filling in missing values, smoothing noisy data, identifying or removing outliers, and resolving inconsistencies. If users believe the data are dirty, they are unlikely to trust the results of any data mining that has been applied to it. Furthermore, dirty data can cause confusion for the mining procedure, resulting in unreliable output. Although most mining routines have some procedures for dealing with incomplete or

noisy data, they are not always robust. Instead, they may concentrate on avoiding overfitting the data to the function being modeled. Therefore, a useful preprocessing step is to run your data through some data cleaning routines.

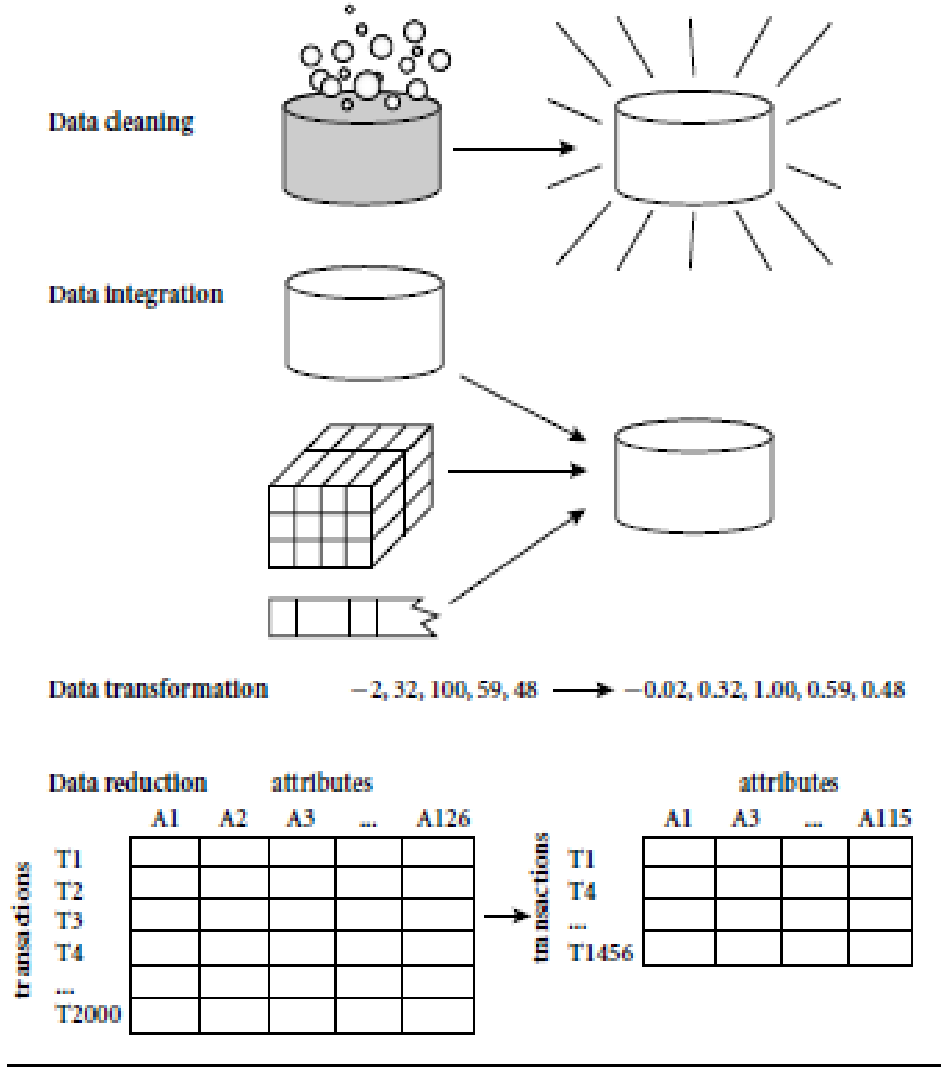


Figure 2.1 Forms of data preprocessing.

Data Cleaning

Real-world data tend to be incomplete, noisy, and inconsistent. *Data cleaning* (or *data cleansing*) routines attempt to fill in missing values, smooth out noise while identifying outliers, and correct inconsistencies in the data. In this section, you will study basic methods for data cleaning.

Missing Values

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

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

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

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

4. Use the attribute mean to fill in the missing value: For example, suppose that the average income of *AllElectronics* customers is $56,000. Use this value to replace the missing value for *income*.

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

6. Use the most probable value to fill in the missing value: This may be determined with regression, inference-based tools using a Bayesian formalism, or decision tree induction. For example, using the other customer attributes in your data set, you may construct a decision tree to predict the missing values for *income.*

Noisy Data

"What is noise?" Noise is a random error or variance in a measured variable. Given a numerical attribute such as, say, *price*, how can we "smooth" out the data to remove the noise? Let's look at the following data smoothing techniques.

Sorted data for *price* (in dollars): 4, 8, 15, 21, 21, 24, 25, 28, 34
Partition into (equal-frequency) bins:
Bin 1: 4, 8, 15
Bin 2: 21, 21, 24
Bin 3: 25, 28, 34

Smoothing by bin means:
Bin 1: 9, 9, 9
Bin 2: 22, 22, 22
Bin 3: 29, 29, 29

Smoothing by bin boundaries:
Bin 1: 4, 4, 15

Bin 2: 21, 21, 24
Bin 3: 25, 25, 34

Binning: Binning methods smooth a sorted data value by consulting its "neighborhood," that is, the values around it. The sorted values are distributed into a number of "buckets," or *bins*. Because binning methods consult the neighborhood of values, they perform*local* smoothing. Figure 2.11 illustrates some binning techniques. In this example, the data for *price* are first sorted and then partitioned into *equal-frequency* bins of size 3 (i.e., each bin contains three values). In smoothing by bin means, each value in a bin is replaced by the mean value of the bin. For example, the mean of the values 4, 8, and 15 in Bin 1 is 9. Therefore, each original value in this bin is replaced by the value 9.

2. Regression: Data can be smoothed by fitting the data to a function, such as with regression. *Linear regression* involves finding the "best" line to fit two attributes (or variables), so that one attribute can be used to predict the other. *Multiple linear regression* is an extension of linear regression, where more than two attributes are involved and the data are fit to a multidimensional surface.

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

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

Data Cleaning as a Process

Missing values, noise, and inconsistencies contribute to inaccurate data. So far, we have looked at techniques for handling missing data and for smoothing data. *"But data cleaning is a big job. What about data cleaning as a process? How exactly does one proceed in tackling this task? Are there any tools out there to help?"*

The first step in data cleaning as a process is ***discrepancy detection***. Discrepancies can be caused by several factors, including poorly designed data entry forms that have many optional fields,humanerror in data entry, deliberate errors (e.g., respondents not wanting to divulge information about themselves), and data decay (e.g., outdated addresses).

Discrepancies may also arise frominconsistent data representations and the inconsistent use of codes.Errors in instrumentationdevices that recorddata,andsystemerrors, areanother source of discrepancies. Errors can also occur when the data are (inadequately) used for purposes other than originally intended. There may also be inconsistencies due to data integration.

The two-step process of discrepancy detection and data transformation (to correct discrepancies) iterates. This process, however, is error-prone and time-consuming. Some transformations may introduce more discrepancies. Some *nested discrepancies* may only be detected after others have been fixed. For example, a typo such as "20004" in a year field may only surface once all date values have been converted to a uniform format. Transformations are often done as a batch process while the user waits without feedback. Only after the transformation is complete can the user go back and check that no new anomalies have been created by mistake. Typically, numerous iterations are required before the user is satisfied. Any tuples that cannot be automatically handled by a given transformation are typically written to a file without any explanation regarding the reasoning behind their failure. As a result, the entire data cleaning process also suffers from a lack of interactivity.

Data Integration and Transformation

Data mining often requires data integration—the merging of data from multiple data stores. The data may also need to be transformed into forms appropriate for mining. This section describes both data integration and data transformation.

Data Integration

It is likely that your data analysis task will involve *data integration*, which combines data from multiple sources into a coherent data store, as in data warehousing. These sources may include multiple databases, data cubes, or flat files.

There are a number of issues to consider during data integration. *Schema integration* and *object matching* can be tricky. How can equivalent real-world entities from multiple data sources be matched up? This is referred to as the entity identification problem. For example, how can the data analyst or the computer be sure that *customer id* in one database and *cust number* in another refer to the same attribute? Examples of metadata for each attribute include the name, meaning, data type, and range of values permitted for the attribute, and null rules for handling blank, zero, or null values (Section 2.3). Such metadata can be used to help avoid errors in schema integration. The metadata may also be used to help transform the data (e.g., where data codes for *pay type* in one database may be *"H"* and *"S"*, and *1* and *2* in another). Hence, this step also relates to data cleaning, as described earlier.

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

Some redundancies can be detected by correlation analysis. Given two attributes, such analysis can measure how strongly one attribute implies the other, based on the available data. For numerical attributes, we can evaluate the correlation between two attributes, *A* and *B*, by computing the correlation coefficient (also known as *Pearson's product moment coefficient*, named after its inventer, Karl Pearson).

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

where *N* is the number of tuples, *ai* and *bi* are the respective values of *A* and *B* in tuple *i*, *A* and *B* are the respective mean values of *A* and *B*, □*A* and □*B* are the respective standard deviations of *A* and *B* .

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

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

where *oi j* is the *observed frequency* (i.e., actual count) of the joint event (*Ai*;*Bj*) and *ei j* is the *expected frequency* of (*Ai*;*Bj*), which can be computed as

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

where *N* is the number of data tuples, *count*(*A*=*ai*) is the number of tuples having value *ai* for *A*, and *count*(*B* = *bj*) is the number of tuples having value *bj* for *B*. The sum in Equation (2.9) is computed over all of the *r_c* cells. Note that the cells that contribute the most to the □2 value are those whose actual count is very different fromthat expected.

Data Transformation

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

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

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

Generalization of the data, where low-level or "primitive" (raw) data are replaced by higher-level concepts through the use of concept hierarchies. For example, categorical attributes, like *street*, can be generalized to higher-level concepts, like *city* or *country*. Similarly, values for numerical attributes, like *age*, may be mapped to higher-level concepts, like *youth*, *middle-aged*, and *senior*.

Normalization, where the attribute data are scaled so as to fall within a small specified range.

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

Data Reduction

Data reduction techniques can be applied to obtain a reduced representation of the data set that ismuch smaller in volume, yet closely maintains the integrity of the original data. That is, mining on the reduced data set should be more efficient yet produce the same (or almost the same) analytical results.

Strategies for data reduction include the following:

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

2. Attribute subset selection, where irrelevant, weakly relevant, or redundant attributes or dimensions may be detected and removed.

3. Dimensionality reduction, where encoding mechanisms are used to reduce the data set size.

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

5. Discretization and concept hierarchy generation,where raw data values for attributes are replaced by ranges or higher conceptual levels. Data discretization is a form of numerosity reduction that is very useful for the automatic generation of concept hierarchies. Discretization and concept hierarchy generation are powerful tools for data mining, in that they allow the mining of data at multiple levels of abstraction.

Data Cube Aggregation

AllElectronics sales per quarter, for the years 2002 to 2004. You are, however, interested in the annual sales (total per year), rather than the total per quarter. Thus the data can be *aggregated* so that the resulting data summarize the total sales per year instead of per quarter. The resulting data set is smaller in volume, without loss of information necessary for the analysis task. Data cubes store multidimensional aggregated information. For example, the following Figure shows a data cube for multidimensional analysis of sales data with respect to annual sales per item type for each *AllElectronics* branch. Each cell holds an aggregate data value, corresponding to the data point in multidimensional space.

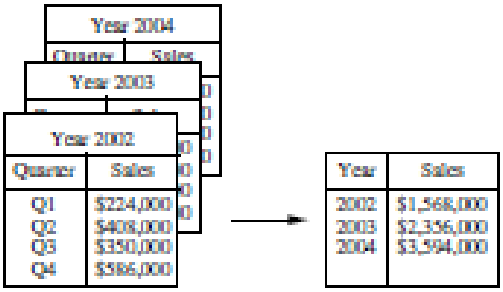
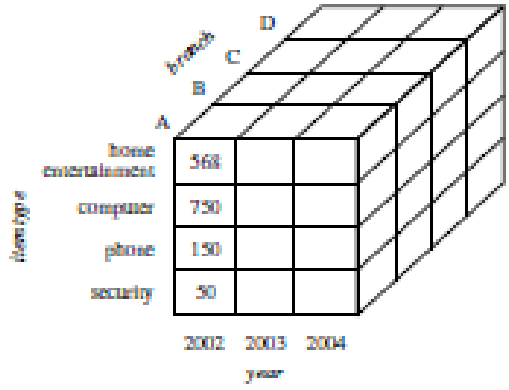


Figure 2.13 Sales data for a given branch of *AllElectronics* for the years 2002 to 2004. On the left, the sales are shown per quarter. On the right, the data are aggregated to provide the annual sales.



Data cubes provide fast access to pre computed, summarized data, thereby benefiting on-line analytical processing as well as data mining. The cube created at the lowest level of abstraction is referred to as the *base cuboid.* The base cuboid should correspond to an individual entity of interest, such as *sales* or *customer*. In other words, the lowest level should be usable, or useful for the analysis. A cube at the highest level of abstraction is the *apex cuboid.*

Attribute Subset Selection

Data sets for analysis may contain hundreds of attributes, many of which may be irrelevant to the mining task or redundant. For example, if the task is to classify customers as to whether or not they are likely to purchase a popular new CD at *AllElectronics* when notified of a sale, attributes such as the customer's telephone number are likely to be irrelevant, unlike attributes such as *age* or *music taste*. Although it may be possible for a domain expert to pick out some of the useful attributes, this can be a difficult and time-consuming task, especially when the behavior of the data is not well known (hence, a reason behind its analysis!). Leaving out relevant attributes or keeping irrelevant attributes may be detrimental, causing confusion for the mining algorithm employed. This can result in discovered patterns of poor quality. In addition, the added volume of irrelevant or redundant attributes can slow down the mining process.
Attribute subset selection6 reduces the data set size by removing irrelevant or redundant attributes (or dimensions). The goal of attribute subset selection is to find a minimum set of attributes such that the resulting probability distribution of the data classes is as close as possible to the original distribution obtained using all attributes. Mining on a reduced set of attributes has an additional benefit. It reduces the number of attributes appearing in the discovered patterns, helping to make the patterns easier to understand

Dimensionality Reduction

In *dimensionality reduction*, data encoding or transformations are applied so as to obtain a reduced or "compressed" representation of the original data. If the original data can be *reconstructed* from the compressed data without any loss of information, the data reduction is called lossless. If, instead, we can reconstruct only an approximation of the original data, then the data reduction is called lossy. There are several well-tuned algorithms for string compression. Although they are typically lossless, they allow only limited manipulation of the data. In this section, we instead focus on two popular and effective methods of lossy dimensionality reduction: *wavelet transforms* and *principal components analysis*.

Wavelet Transforms

The discrete wavelet transform(DWT) is a linear signal processing technique that, when applied to a data vector $\boldsymbol{X}$, transforms it to a numerically different vector, $\boldsymbol{X}0$, of wavelet coefficients. The two vectors are of the same length. When applying this technique to data reduction, we consider each tuple as an n-dimensional data vector, that is, $\boldsymbol{X} = (x1;x2;:::;xn)$, depicting n measurements made on the tuple from$n$ database attributes.8 *"How can this technique be useful for data reduction if the wavelet transformed data are of the same length as the original data?"* The usefulness lies in the fact that the wavelet transformed data can be truncated. A compressed approximation of the data can be retained by storing only a small fraction of the strongest of the wavelet coefficients.

The DWT is closely related to the *discrete Fourier transform(DFT),* a signal processing technique involving sines and cosines. In general, however, theDWT achieves better lossy

compression. That is, if the same number of coefficients is retained for aDWT and a DFT of a given data vector, the DWT version will provide a more accurate approximation of the original data. Hence, for an equivalent approximation, the DWT requires less space than the DFT. Unlike the DFT, wavelets are quite localized in space, contributing to the conservation of local detail.
There is only one DFT, yet there are several families of DWTs. Figure 2.16 shows some wavelet families. Popular wavelet transforms include the Haar-2, Daubechies-4, and Daubechies-6 transforms. The general procedure for applying a discrete wavelet transform uses a hierarchical *pyramid algorithm* that halves the data at each iteration, resulting in fast computational speed. The method is as follows:

1. The length, *L*, of the input data vector must be an integer power of 2. This condition can be met by padding the data vector with zeros as necessary (*L* _ *n*).
2. Each transforminvolves applying two functions. The first applies some data smoothing, such as a sum or weighted average. The second performs a weighted difference, which acts to bring out the detailed features of the data.
3. The two functions are applied to pairs of data points in ***X***, that is, to all pairs of measurements (*x*2*i*;*x*2*i*+1). This results in two sets of data of length *L*=2. In general, these represent a smoothed or low-frequency version of the input data and the highfrequency content of it, respectively.
4. The two functions are recursively applied to the sets of data obtained in the previous loop, until the resulting data sets obtained are of length 2.
5. Selected values from the data sets obtained in the above iterations are designated the wavelet coefficients of the transformed data.

Wavelet transforms can be applied to multidimensional data, such as a data cube. This is done by first applying the transform to the first dimension, then to the second, and so on. The computational complexity involved is linear with respect to the number of cells in the cube. Wavelet transforms give good results on sparse or skewed data and on data with ordered attributes. Lossy compression by wavelets is reportedly better than JPEG compression, the current commercial standard. Wavelet transforms have many real-world applications, including the compression of fingerprint images, computer vision, analysis of time-series data, and data cleaning.

Principal Components Analysis

Suppose that the data to be reduced consist of tuples or data vectors described by *n* attributes or dimensions. Principal components analysis, or PCA (also called the Karhunen-Loeve, or K-L, method), searches for *k n*-dimensional orthogonal vectors that can best be used to represent the data, where *k* _ *n*. The original data are thus projected onto a much smaller space, resulting in dimensionality reduction. Unlike attribute subset selection, which reduces the attribute set size by retaining a subset of the initial set of attributes, PCA “combines” the essence of attributes by creating an alternative, smaller set of variables. The initial data can then be projected onto this smaller set. PCA often reveals relationships that were not previously suspected and thereby allows interpretations that would not ordinarily result.
The basic procedure is as follows:

1. The input data are normalized, so that each attribute falls within the same range. This step helps ensure that attributes with large domains will not dominate attributes with smaller domains.
2. PCA computes *k* orthonormal vectors that provide a basis for the normalized input data. These are unit vectors that each point in a direction perpendicular to the others. These vectors are referred to as the *principal components*. The input data are a linear combination of the principal components.
3. The principal components are sorted in order of decreasing "significance" or strength. The principal components essentially serve as a new set of axes for the data, providing important information about variance. That is, the sorted axes are such that the first axis shows the most variance among the data, the second axis shows the next highest variance, and so on. For example, Figure 2.17 shows the first two principal components, ***Y***1 and ***Y***2, for the given set of data originally mapped to the axes *X*1 and *X*2. This information helps identify groups or patterns within the data.
4. Because the components are sorted according to decreasing order of "significance," the size of the data can be reduced by eliminating the weaker components, that is, those with low variance. Using the strongest principal components, it should be possible to reconstruct a good approximation of the original data.
PCA is computationally inexpensive, can be applied to ordered and unordered attributes, and can handle sparse data and skewed data. Multidimensional data of more than two dimensions can be handled by reducing the problem to two dimensions. Principal components may be used as inputs to multiple regression and cluster analysis. In comparison with wavelet transforms, PCA tends to be better at handling sparse data, whereas wavelet transforms are more suitable for data of high dimensionality.

Numerosity Reduction

"Can we reduce the data volume by choosing alternative, 'smaller' forms of data representation?"
Techniques of *numerosity reduction* can indeed be applied for this purpose.
These techniques may be parametric or nonparametric. For *parametric methods*, a model is used to estimate the data, so that typically only the data parameters need to be stored, instead of the actual data. (Outliers may also be stored.) Log-linear models, which estimate discrete multidimensional probability distributions, are an example.
Nonparametric methods for storing reduced representations of the data include histograms,

Regression and Log-Linear Models

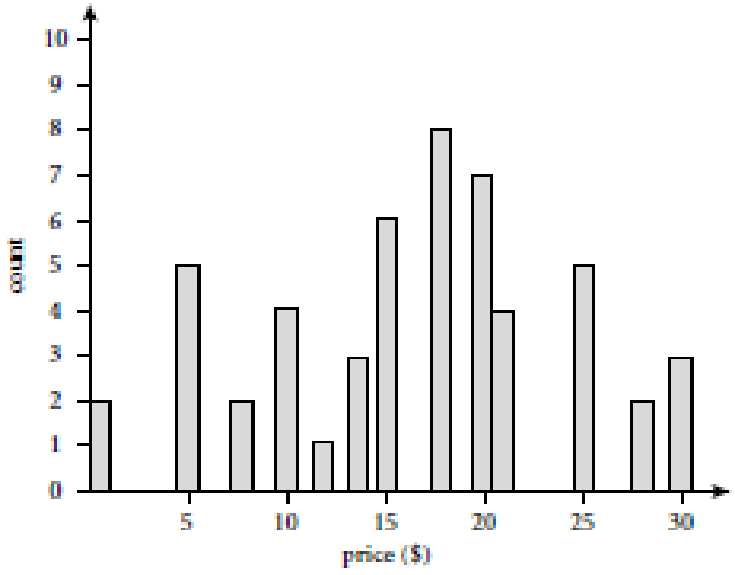
Regression and log-linear models can be used to approximate the given data. In (simple) linear regression, the data are modeled to fit a straight line. For example, a random variable, *y* (called a *response variable*), can be modeled as a linear function of another random variable, *x* (called a *predictor variable*), with the equation
$y = wx+b$, (2.14)
where the variance of *y* is assumed to be constant. In the context of data mining, *x* and *y* are numerical database attributes. The coefficients, *w*and *b* (called *regression coefficients*), specify the slope of the line and the *Y*-intercept, respectively. These coefficients can be solved for by the *method of least squares*, which minimizes the error between the actual line separating the data and the estimate of the line. Multiple linear regression is an extension of (simple) linear regression, which allows a response variable, *y*, to be modeled

as a linear function of two or more predictor variables.

Histograms

Histograms use binning to approximate data distributions and are a popular form of data reduction. Histograms were introduced in Section 2.2.3. A histogram for an attribute, *A*, partitions the data distribution of *A* into disjoint subsets, or *buckets*. If each bucket represents only a single attribute-value/frequency pair, the buckets are called *singleton bucket* clustering, and sampling.

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



A histogram for *price* using singleton buckets—each bucket represents one price-value/ frequency pair.

"How are the buckets determined and the attribute values partitioned?" There are several partitioning rules, including the following:

Equal-width: In an equal-width histogram, the width of each bucket range is uniform (such as the width of $10 for the buckets in the above Figure.

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

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

MaxDiff: In a MaxDiff histogram, we consider the difference between each pair of

adjacent values. A bucket boundary is established between each pair for pairs having the β-1 largest differences, where β is the user-specified number of buckets.

Clustering

Clustering techniques consider data tuples as objects. They partition the objects into groups or *clusters*, so that objects within a cluster are "similar" to one another and "dissimilar" to objects in other clusters. Similarity is commonly defined in terms of how "close" the objects are in space, based on a distance function. The "quality" of a cluster may be represented by its *diameter*, the maximum distance between any two objects in the cluster. *Centroid distance* is an alternative measure of cluster quality and is defined as the average distance of each cluster object from the cluster centroid (denoting the "average object," or average point in space for the cluster). Three data clusters are visible. In data reduction, the cluster representations of the data are used to replace the actual data. The effectiveness of this technique depends on the nature of the data. It is much more effective for data that can be organized into distinct clusters than for smeared data.

In database systems, multidimensional index trees are primarily used for providing fast data access. They can also be used for hierarchical data reduction, providing a multiresolution clustering of the data. This can be used to provide approximate answers to queries. An index tree recursively partitions the multidimensional space for a given set of data objects, with the root node representing the entire space. Such trees are typically balanced, consisting of internal and leaf nodes. Each parent node contains keys and pointers to child nodes that, collectively, represent the space represented by the parent node. Each leaf node contains pointers to the data tuples they represent (or to the actual tuples).

Sampling

Sampling can be used as a data reduction technique because it allows a large data set to be represented by a much smaller random sample (or subset) of the data. Suppose that a large data set, *D*, contains *N* tuples.

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

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

Cluster sample: If the tuples in *D* are grouped into *M* mutually disjoint "clusters," then an SRS of *s* clusters can be obtained, where $s < M$. For example, tuples in a database are usually retrieved a page at a time, so that each page can be considered a cluster. A reduced data representation can be obtained by applying, say, SRSWOR to the pages, resulting in a cluster sample of the tuples. Other clustering criteria conveying

rich semantics can also be explored. For example, in a spatial database, we may choose to define clusters geographically based on how closely different areas are located.

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

Data Discretization and Concept Hierarchy Generation

Data discretization techniques can be used to reduce the number of values for a given continuous attribute by dividing the range of the attribute into intervals. Interval labels can then be used to replace actual data values. Replacing numerous values of a continuous attribute by a small number of interval labels thereby reduces and simplifies the original data. This leads to a concise, easy-to-use, knowledge-level representation of mining results.

Discretization techniques can be categorized based on how the discretization is performed, such as whether it uses class information or which direction it proceeds (i.e., top-down vs. bottom-up). If the discretization process uses class information, then we say it is *supervised discretization*. Otherwise, it is *unsupervised*. If the process starts by first finding one or a few points (called *split points* or *cut points*) to split the entire attribute range, and then repeats this recursively on the resulting intervals, it is called *top-down discretization* or *splitting*. This contrasts with *bottom-up discretization* or *merging*, which starts by considering all of the continuous values as potential split-points, removes some by merging neighborhood values to form intervals, and then recursively applies this process to the resulting intervals. Discretization can be performed recursively on an attribute to provide a hierarchical or multiresolution partitioning of the attribute values, known as a concept hierarchy. Concept hierarchies are useful for mining at multiple levels of abstraction.

A concept hierarchy for a given numerical attribute defines a discretization of the attribute. Concept hierarchies can be used to reduce the data by collecting and replacing low-level concepts (such as numerical values for the attribute *age*) with higher-level concepts (such as *youth*, *middle-aged*, or *senior*). Although detail is lost by such data generalization, the generalized data may be more meaningful and easier to interpret.

Discretization and Concept Hierarchy Generation for Numerical Data

It is difficult and laborious to specify concept hierarchies for numerical attributes because of the wide diversity of possible data ranges and the frequent updates of data values. Such manual specification can also be quite arbitrary. Concept hierarchies for numerical attributes can be constructed automatically based on data discretization. We examine the following methods:

binning, *histogram analysis*,*entropy-based discretization*, c2-*merging*, *cluster analysis*, and *discretization by intuitive partitioning.* In general, each method assumes that the values to be discretized are sorted in ascending order.

Binning

Binning is a top-down splitting technique based on a specified number of bins. Section 2.3.2 discussed binning methods for data smoothing. These methods are also used as discretization methods for numerosity reduction and concept hierarchy generation. For example, attribute values can be discretized by applying equal-width or equal-frequency binning, and then replacing each bin value by the bin mean or median, as in *smoothing by bin means* or *smoothing by bin medians*, respectively. These techniques can be applied recursively to the resulting partitions in order to generate concept hierarchies. Binning does not use class information and is therefore an unsupervised discretization technique. It is sensitive to the user-specified number of bins, as well as the presence of outliers.

Histogram Analysis

Like binning, histogram analysis is an unsupervised discretization technique because it does not use class information. Histograms partition the values for an attribute, *A*, into disjoint ranges called *buckets*. Histograms were introduced in Section 2.2.3. Partitioning rules for defining histograms were described in Section 2.5.4. In an *equal-width* histogram, for example, the values are partitioned into equal-sized partitions or ranges (such as in Figure 2.19 for *price*, where each bucket has a width of $10). With an *equal frequency* histogram, the values are partitioned so that, ideally, each partition contains the same number of data tuples.

The histogram analysis algorithm can be applied recursively to each partition in order to automatically generate a multilevel concept hierarchy, with the procedure terminating once a pre specified number of concept levels has been reached. A *minimum interval size* can also be used per level to control the recursive procedure. This specifies the minimum width of a partition, or the minimum number of values for each partition at each level. Histograms can also be partitioned based on cluster analysis of the data distribution, as described below.

Entropy-Based Discretization

Entropy is one of the most commonly used discretization measures. It was first introduced by Claude Shannon in pioneering work on information theory and the concept of information gain. Entropy-based discretization is a supervised, top-down splitting technique. It explores class distribution information in its calculation and determination of split-points (data values for partitioning an attribute range). To discretize a numerical attribute, *A*, the method selects the value of *A* that has the minimum entropy as a split-point, and recursively partitions the resulting intervals to arrive at a hierarchical discretization. Such discretization forms a concept hierarchy for *A*.

Let *D* consist of data tuples defined by a set of attributes and a class-label attribute. The class-label attribute provides the class information per tuple. The basic method for entropy-based discretization of an attribute *A* within the set is as follows:

1. Each value of *A* can be considered as a potential interval boundary or split-point

(denoted *split point*) to partition the range of *A*. That is, a split-point for *A* can partition the tuples in *D* into two subsets satisfying the conditions *A _ split point* and *A > split point*, respectively, thereby creating a binary discretization.

2. Entropy-based discretization, as mentioned above, uses information regarding the class label of tuples. To explain the intuition behind entropy-based discretization, we must take a glimpse at classification. Suppose we want to classify the tuples in *D* by partitioning on attribute *A* and some split-point. Ideally, we would like this partitioning to result in an exact classification of the tuples. For example, if we had two classes, we would hope that all of the tuples of, say, class C_1 will fall into one partition, and all of the tuples of class C_2 will fall into the other partition. However, this is unlikely. For example, the first partition may contain many tuples of C_1, but also some of C_2. How much more information would we still need for a perfect classification, after this partitioning? This amount is called the *expected information requirement* for classifying a tuple in *D* based on partitioning by *A*. It is given by

$$Info_A(D) = \frac{|D_1|}{|D|} Entropy(D_1) + \frac{|D_2|}{|D|} Entropy(D_2),$$

where D_1 and D_2 correspond to the tuples in *D* satisfying the conditions *A _ split point* and *A > split point*, respectively; j*D*j is the number of tuples in *D*, and so on. The entropy function for a given set is calculated based on the class distribution of the tuples in the set. For example, given *m* classes, $C_1;C_2; : : : ;C_m$, the entropy of D_1 is

$$Entropy(D_1) = -\sum_{i=1}^{m} p_i \log_2(p_i),$$

where p_i is the probability of class C_i in D_1, determined by dividing the number of tuples of class C_i in D_1 by jD_1j, the total number of tuples in D_1. Therefore, when selecting a split-point for attribute *A*, we want to pick the attribute value that gives the minimum expected information requirement (i.e., min($Info_A(D)$)). This would result in the minimum amount of expected information (still) required to perfectly classify the tuples after partitioning by *A_split point* and *A>split point*. This is equivalent to the attribute-value pair with the maximum information gain (the further details of which are given in Chapter 6 on classification.) Note that the value of $Entropy(D_2)$ can be computed similarly as in Equation (2.16).
"But our task is discretization, not classification!", you may exclaim. This is true.We use the split-point to partition the range of *A* into two intervals, corresponding to *A _ split point* and *A > split point*.
3. The process of determining a split-point is recursively applied to each partition obtained, until some stopping criterion is met, such as when the minimum information requirement on all candidate split-points is less than a small threshold, ε, or when the number of intervals is greater than a threshold, *max interval*.
Entropy-based discretization can reduce data size. Unlike the other methods mentioned here so far, entropy-based discretization uses class information. This makes it more likely that the interval boundaries (split-points) are defined to occur in places that may help improve classification accuracy.

Interval Merging by Chi Square χ^2 Analysis

ChiMerge is a χ^2-based discretization method. The discretization methods that we have studied up to this point have all employed a top-down, splitting strategy. This contrasts with ChiMerge, which employs a bottom-up approach by finding the best neighboring intervals and then merging these to form larger intervals, recursively. The method is supervised in that it uses class information. The basic notion is that for accurate discretization, the relative class frequencies should be fairly consistent within an interval. Therefore, if two adjacent intervals have a very similar distribution of classes, then the intervals can be merged. Otherwise, they should remain separate.

ChiMerge proceeds as follows. Initially, each distinct value of a numerical attribute *A* is considered to be one interval. χ^2 tests are performed for every pair of adjacent intervals. Adjacent intervals with the least χ^2 values are merged together, because low χ^2 values for a pair indicate similar class distributions. This merging process proceeds recursively until a predefined stopping criterion is met.

Cluster Analysis

Cluster analysis is a popular data discretization method. A clustering algorithm can be applied to discretize a numerical attribute, *A*, by partitioning the values of *A* into clusters or groups. Clustering takes the distribution of *A* into consideration, as well as the closeness of data points, and therefore is able to produce high-quality discretization results. Clustering can be used to generate a concept hierarchy for *A* by following either a top down splitting strategy or a bottom-up merging strategy, where each cluster forms a node of the concept hierarchy. In the former, each initial cluster or partition may be further decomposed into several subclusters, forming a lower level of the hierarchy.

.

Discretization by Intuitive Partitioning

Although the above discretization methods are useful in the generation of numerical hierarchies, many users would like to see numerical ranges partitioned into relatively uniform, easy-to-read intervals that appear intuitive or "natural." For example, annual salaries broken into ranges like ($50,000, $60,000] are often more desirable than ranges like ($51,263.98, $60,872.34], obtained by, say, some sophisticated clustering analysis.

Concept Hierarchy Generation for Categorical Data

Categorical data are discrete data. Categorical attributes have a finite (but possibly large) number of distinct values, with no ordering among the values. Examples include *geographic location*, *job category*, and *itemtype*. There are several methods for the generation of concept hierarchies for categorical data.

Specification of a partial ordering of attributes explicitly at the schema level by users or experts: Concept hierarchies for categorical attributes or dimensions typically involve a group of attributes. A user or expert can easily define a concept hierarchy by specifying a partial or total ordering of the attributes at the schema level. For example,

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

Online Data Storage

Online data storage refers to the practice of storing electronic data with a third party service accessed via the Internet. It's an alternative to traditional local storage (such as disk or tape drives) and portable storage (such as optical media or flash drives). It can also be called "hosted storage," "Internet storage" or "cloud storage."

In recent years, the number of vendors offering online data storage for both consumers and businesses has increased dramatically. Some services store only a particular kind of data, such as photos, music or backup data, while others will allow users to store any type of file. Most of these vendors offer a small amount of storage for free with additional storage capacity available for a fee, usually paid on a monthly or annual basis.

Benefits of Online Storage

One of the biggest benefits of online storage is the ability to access data from anywhere. As the number of devices the average person uses continues to grow, syncing or transferring data among devices has become more important. Not only does it help transfer data between devices, online data storage also provides the ability to share files among different users. This is particularly helpful for business users, although it's also popular with consumers who want to share photos, videos and similar materials with their friends and family.

Online data storage also offers distinct advantages for backup and disaster recovery situations because it's located off site. In a fire, flood, earthquake or similar situation, on-site backups could be damaged, but online backups won't be affected unless the disaster is very widespread.

However, online data storage does have some potential downsides. Some people worry about the security of cloud storage services, and some vendors have experienced significant outages from time to time, leading to concerns about reliability.

Online Storage Enhances Data Protection and Availability

Before the development of the Internet, computer systems were limited to local storage or portable storage, first in the form of tapes and floppy disks, then CDs, DVDs and USB thumb drives. Generally, using on-site storage is faster than using Internet storage, because user does

not have to wait for files to upload or download. However, on-site storage is more susceptible to loss due to theft, natural disasters or device failure. By contrast, most online data storage facilities offer enhanced physical security and automated backup capabilities to ensure that data is not lost. Online data storage also enables easier data transfer and sharing.

Like local storage, portable storage devices offer fast data transfer along with some data transfer and sharing capabilities. However, portable storage isn't quite as convenient as online data storage, particularly if we want to share files with a large number of users. Portable storage devices are also easy to lose or damage, and they offer limited storage capacity.

Recently, the term cloud storage has become increasingly common. Although many people use the terms "cloud storage" and "online storage" interchangeably, technically, cloud storage is a particular kind of online data storage. In order to be considered cloud storage, a service must be sold on demand, provide elasticity (the user can have as much or as little as desired) and offer self-service capabilities.

Many individuals and organizations use a mix of on-site and online storage capabilities. For example, they might use local storage for files they use frequently and online storage for backup or archive data. Or they might use local storage for personal data and online storage for files that they wish to share with others.

Online Storage Implementation

In most cases, setting up online data storage is incredibly easy, but the exact process will depend on the vendor. For most consumer online storage services, the process entails nothing more than setting up an account with a user name and password, although in some cases users will also need to download and install some software. Using these online data storage services is also very simple, and many offer intuitive drag-and-drop interfaces. Again, the exact details vary by vendor.

For business-oriented online storage services, the set up and use procedures can be slightly more complex because they generally offer more options for configuration, security and reliability. However, because they want to appeal to as many customers as possible, enterprise vendors generally make their services as easy to use and maintain as possible.

Part B TWO MARK QUESTIONS

1.Define Datacube.
2.What do you mean by market basket analysis?
3.What is Principal component Analysis in Data Mining?
4.What is Naive Bayesian Classification?
5.Define outlier analysis.

PART-C 8 MARK QUESTIONS

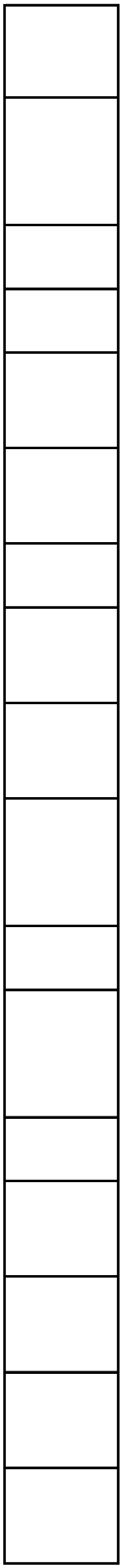
1.Explain Data Discretization and Concept Hierarchy Generation.
2. Discuss Data Cleaning methods in detail.
3.Explain the techniques of numerosity reduction.
4.Discuss the online data storage.
5.Discuss the two methods of dimensionality reduction.
6.Explain the needs of preprocessing the data.
7.Discuss the data integration and transformation process.
8.Explain how wavelet transforms are useful for data reduction.
9.Explain the data transformation process.
10.Discuss the usage of Principal Components Analysis in Data reduction.

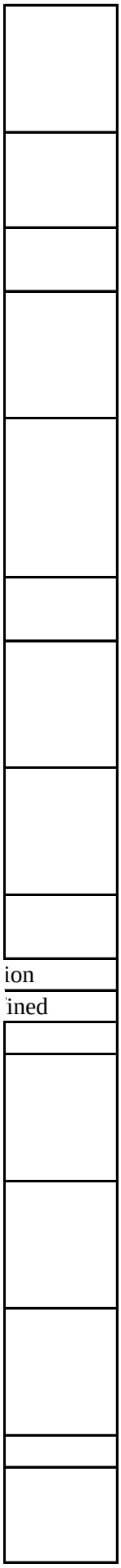
Questions	opt1	opt2	opt3	opt4	opt5	opt6	Answer
_____ model makes a prediction about values of data using known results found from different data.	predictive	descriptive	preference	process			predictive
_____ model identifies patterns or relationships in data.	predictive	descriptive	process	preference			descriptive
_____ maps data into predefined groups or classes	regression	classification	clustering	prediction			classification
_____ is used to map a data item to a real valued prediction variable.	regression	classification	clustering	prediction			regression
Classification of input patterns by using its similarity to the predefined classes is called____	prediction	classification	pattern recognition	regression			pattern recognition
A special type of clustering is called _____	classification	segmentation	regression	prediction			segmentation
_____ maps data into subsets with associated simple description.	classification	segmentation	pattern recognition	summarization			summarization
Which rule is used to identify the specific type of data association	delta rule	association rule	classification rule	segmentation rule			association rule
Which is used to determine sequential patterns in data?	sequence discovery	segmentation	regression	random analysis			sequence discovery
_____ is the process of finding useful information and patterns in data	datamining	KDD	selection	transformation			KDD
_____ is the use of algorithms to extract the information and pattern derived by the knowledge discovery	datamining	KDD	selection	transformation			datamining
The KDD process is often said to be _____	trivial	nontrivial	significant	consequential			nontrivial
Extreme values that occur infrequently, may actually be removed is called _____	outliers	outlayers	statistics	frequency			outliers
Traditional graph structure is a _____ visualization techniqe	graphical	geometric	icon-based	pixel-based			graphical
The major goal of _____ is to be able to describe the result in meaningful manner.	datamining	KDD	selection	transformation			KDD
_____ techniques often involve sophisticated multimedia and graphics presentations.	summarization	conceptualization	visualization	decription			visualization

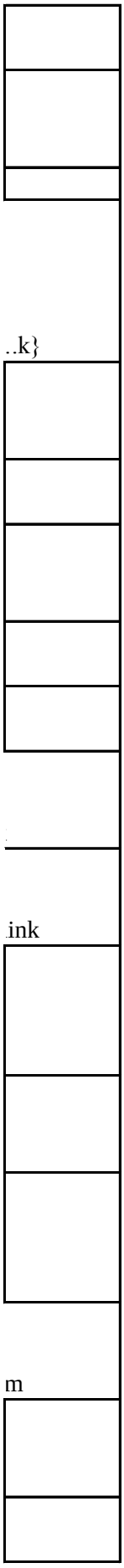
Describing a large database can be viewed as using _____ to help uncover hidden information about the data	approximation	search	induction	compression			approximation
_____ is used to proceed from very specific knowledge to more general information.	approximation	search	induction	compression			induction
_____ occurs when the model does not fit future states.	outliers	interception	overfitting	qureying			overfitting
The problem of increasing over all complexity and decrease the efficiency of an algorithms is _____	Single dimensionality	Multi-Dimensionality	Dimensionality curse	Recursion			Dimensionality curse
What is the expansion of ROI	return on investment	result of invension	research oriented investment	return of interest			return on investment
Find out predictive model datamining task from the followings	clustering	summarization	association rule	regression			regression
Find out descriptive model datamining task from the followings	classification	sequence discovery	prediction	regression			sequence discovery
Flooding, speech recognition, machine learning are_____	classification applicationa	prediction applications	clustering application	regression applications			prediction applications
Link analysis alternatively referred to as a _____	rule analysis	affinity analysis	master analysis	report analysis			affinity analysis
Clustering is similar to _____	Summariz	Classifica	Associati	Regression			Classificati
In clustering the groups are_____	Predefine	redefined	Not Prede	Not Redefined			Not Predef
In clustering the groups are called _____	Clusters	Outliers	Elements	Segments			Clusters
The distance between points in a cluster is _____ than the distance between a point in the cluster and any point outside it.	Lesser	Greater	Equal	More			Lesser
The term similar to clustering is_____ where like tupples in a database are grouped together.	Database segmentation	Distance Measure	Classification	Regression			Database segmentation
The first domain in which clustering was u	disease classification	biological taxonomy	Medicine	Economics			biological taxonomy
The element that donot naturally fall into a	Segment	Domain	Outlier	Medoid			Outlier
Dynamic data in the database implies that cluster membership may_____ over time.	Remain	Change	Increase	Decrease			Change

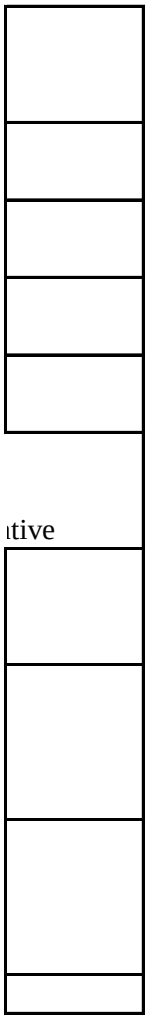
The _____ meaning of each cluster may be difficult.	Static	Dynamic	Semantic	Constant			Semantic
Clustering can be viewed similar to _____ .	Supervised learning	Conditional learning	Predefined learning	Unsupervised learning			Unsupervised learning
Clusters results are _____ .	Dynamic	Static	Increase	Decrease			Dynamic
Given a database D={t1,t2,….tn} of tuples and an integer value k, the clustering problem is to define a mapping _____ where each ti is assigned to one cluster kj, 1≤j≤k.	f:D_ {1,..	f.D={1,..	f:D→{1,.	f/d_{1,…k}			f:D→{1,..
Clustering algorithms themselves may be viewed as hierarchical or _____ .	Categorial	Large Database	Partitiona	Extrinsic			Partitional
In hierarchical a _____ set of clusters is created.	Nested	Unique	Separate	Combined			Nested
At the highest level in hierarchical clustering the all items belong to _____ clusters.	Separate	Same	Different	Distinct			Same
In partitional clustering the algorithm creates _____ set of clusters.	Many	Same	Unique	Only One			Only One
Non Overlapping clusters can be viewed as extrinsic or _____ .	Multrinsic	Intrinsic	Distinct	discrete			Intrinsic
Smallest distance between an element in one cluster and an element in the other is called as_____.	Multiple l	Complete	Single lin	Centroid			Single link
Largest distance between an element in one cluster and an element in the other is called as_____.	Multiple l	Complete	Single lin	Centroid			Complete l
If clusters have a representative centroid, then the centroid distance is defined as the distance between the _____.	Medoid	Outliers	Clusters	Centroid			Centroid
_____ are sample points with values much different from those of the remaning set of data.	Outlier	Centroid	Medoid	Cluster			Outlier
_____ is the process of identifying outlieers in a set of data.	Discordan	Outlier detection	Statistical Techniq ue	sequential test			Outlier detection
A tree data structure, called a _____ can be used to illustrate clustering technique.	Centroid	Outlier	Dendrogr	information gain			Dendrogra
The root in the dendrogram tree contains one cluster where all elements are _____.	Same	Different	Unique	Together			Together
The leaves in the dendrogram consists of a _____ element cluster.	Single	Same	Unique	Distinct			Single

Internal nodes in the dendrogram represent new clusters formed by _____ the clusters.	Seperating	Merging	Creating	Deleting			Merging
Hierarchical clustering algorithm is further divided into _____ .	1	2	3	4			2
The space complexity for hierarchical algorithms is _____.	O(n)	O/n	n	$O(n^2)$			$O(n^2)$
The space required for a dendrogram is _____ .	O(n)	$O(n^2)$	O(kn)	O(k)			O(kn)
The time complexity for hierarchical algorithms is _____ .	O(n)	$O(n^2)$	O(kn)	$O(kn^2)$			$O(kn^2)$
_____ algorithm start with each individual item in its own cluster and iteratively merge clusters.	Hierarchic	Partitiona	Agglomer	Divisive			Agglomera
The _____ link technique is based on the idea of finding maximal connected components in a graph.	Single	Multiple	Multileve	Hybrid			Single
A _____ is a graph in which there exists a path between any two vertices.	Disconne cted Compone nt	Connect ed Compon ent	Distinct compone nt	discrete value			Connecte d Compone nt
Two clusters are merged if there is at least one edge that connects the two clusters is often called as _____.	Single link	K Means clusterin g	Nearest neighbor	isolated value			Nearest neighbor




ion
ined


..k}
ink
m


ıtive

UNIT III

Mining Frequent Patterns, Associations and Correlations: Basic Concepts - Efficient and Scalable Frequent item set Mining Methods - From Association Mining to Correlation Analysis.

Mining Frequent Patterns, Associations and Correlations

Basic Concepts

Frequent patterns are patterns (such as itemsets, subsequences, or substructures) that appear in a data set frequently. For example, a set of items, such as milk and bread, that appear frequently together in a transaction data set is a *frequent itemset*. A subsequence, such as buying first a PC, then a digital camera, and then a memory card, if it occurs frequently in a shopping history database, is a (*frequent*) *sequential pattern*.
A *substructure* can refer to different structural forms, such as subgraphs, subtrees, or sublattices, which may be combined with itemsets or subsequences. If a substructure occurs frequently, it is called a (*frequent*) *structured pattern*.
Finding such frequent patterns plays an essential role in mining associations, correlations, and many other interesting relationships among data. Moreover, it helps in data classification, clustering, and other data mining tasks as well. Thus, frequent pattern mining has become an important data mining task and a focused theme in data mining research.

Frequent pattern mining searches for recurring relationships in a given data set. This section introduces the basic concepts of frequent pattern mining for the discovery of interesting associations and correlations between itemsets in transactional and relational Databases.

Market Basket Analysis

Frequent itemset mining leads to the discovery of associations and correlations among items in large transactional or relational data sets. With massive amounts of data continuously being collected and stored, many industries are becoming interested in mining such patterns from their databases. The discovery of interesting correlation relationships among huge amounts of business transaction records can help in many business decision-making processes, such as catalog design, cross-marketing, and customer shopping behavior analysis.
A typical example of frequent itemset mining is market basket analysis. This process analyzes customer buying habits by finding associations between the different items that customers place in their “shopping baskets” (Figure 3.1). The discovery of such associations can help retailers develop marketing strategies by gaining insight into which items are frequently purchased together by customers. For instance, if customers are

buying milk, how likely are they to also buy bread (and what kind of bread) on the same trip to the supermarket? Such information can lead to increased sales by helping retailers do selective marketing and plan their shelf space.

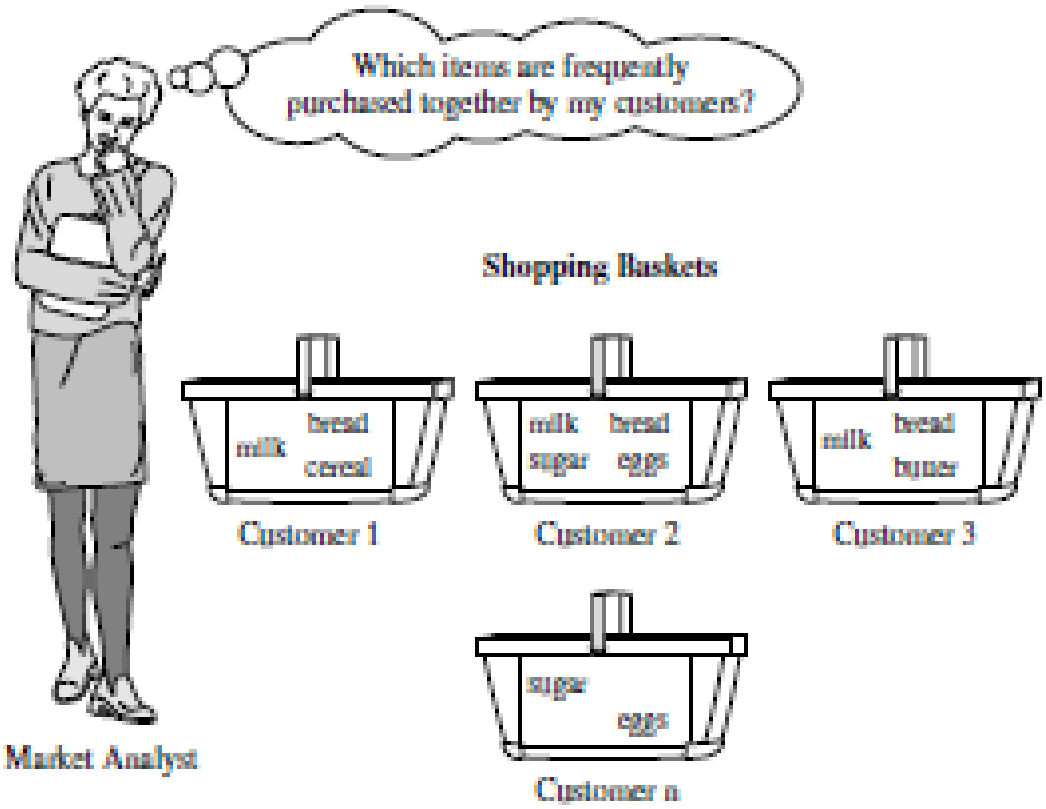


Figure 3.1 Market Basket Analysis

Suppose, as manager of an *AllElectronics* branch, you would like to learn more about the buying habits of your customers. Specifically, you wonder, *"Which groups or sets of items are customers likely to purchase on a given trip to the store?"* To answer your question, market basket analysis may be performed on the retail data of customer transactions at your store. You can then use the results to plan marketing or advertising strategies, or in the design of a new catalog.

For instance, market basket analysis may help you design different store layouts. In one strategy, items that are frequently purchased together can be placed in proximity in order to further encourage the sale of such items together. If customers who purchase computers also tend to buy antivirus software at the same time, then placing the hardware display close to the software display may help increase the sales of both items. In an

alternative strategy, placing hardware and software at opposite ends of the store may entice customers who purchase such items to pick up other items along the way.

For instance, after deciding on an expensive computer, a customer may observe security systems for sale while heading toward the software display to purchase antivirus software and may decide to purchase a home security system as well. Market basket analysis can also help retailers plan which items to put on sale at reduced prices. If customers tend to purchase computers and printers together, then having a sale on printers may encourage the sale of printers *as well as* computers.

For example, the information that customers who purchase computers also tend to buy antivirus software at the same time is represented in Association Rule (5.1) below:

Rule support and confidence are two measures of rule interestingness. They respectively reflect the usefulness and certainty of discovered rules. A support of 2% for Association Rule (5.1) means that 2% of all the transactions under analysis show that computer and antivirus software are purchased together.

A confidence of 60% means that 60% of the customers who purchased a computer also bought the software. Typically, association rules are considered interesting if they satisfy both a minimum support threshold and a minimum confidence threshold. Such thresholds can be set by users or domain experts. Additional analysis can be performed to uncover interesting statistical correlations between associated items.

Frequent Itemsets, Closed Itemsets, and Association Rules

- Itemset X = $\{x_1, \ldots, x_k\}$
- Find all the rules $X \rightarrow Y$ with minimum support and confidence
  - support, *s*, probability that a transaction contains $X \cup Y$
  - confidence, *c,* conditional probability that a transaction having X also contains *Y*

The rule *A*) *B* has confidence *c* in the transaction set *D*, where *c* is the percentage of transactions in *D* containing *A* that also contain *B*. This is taken to be the conditional probability, *P*(*B*j*A*).

Let sup_{min} = 50%, $conf_{min}$ = 50%
Freq. Pat.: {A:3, B:3, D:4, E:3, AD:3}

Association rules:
***A* → *D* (60%, 100%)**
***D* → *A* (60%, 75%)**

Closed Itemset

An itemset X is closed if X is *frequent* and there exists *no super-pattern* Y ⊃ X, *with the same support* as X.

A set of items is referred to as an itemset. An itemset that contains *k* items is a *k*-itemset. The set of {*computer, antivirus software}* is a 2-itemset. The occurrence frequency of an itemset is the number of transactions that contain the itemset.
This is also known, simply, as the frequency, support count, or count of the itemset.
Note that the itemset support defined in Equation (5.2) is sometimes referred to as *relative support*, whereas the occurrence frequency is called the absolute support. If the relative support of an itemset *I* satisfies a prespecified minimum support threshold (i.e., the absolute support of *I* satisfies the corresponding minimum support count threshold), then *I* is a frequent itemset.3 The set of frequent *k*-itemsets is commonly denoted by *Lk*.4

$$support(A \rightarrow B) = P(A \cup B)$$
$$confidence(A \rightarrow B) = P(B|A).$$

$$confidence(A \rightarrow B) = P(B|A) = \frac{support(A \cup B)}{support(A)} = \frac{support\_count(A \cup B)}{support\_count(A)}.$$

The above equations show that the confidence of rule $A \rightarrow B$ can be easily derived from the support counts of *A* and *A U B*. That is, once the support counts of *A*, *B*, and A U *B* are found, it is straightforward to derive the corresponding association rules A→*B* and *B*→*A* and check whether they are strong. Thus the problem of mining association rules can be reduced to that of mining frequent itemsets.

In general, association rule mining can be viewed as a two-step process:
1. Find all frequent itemsets: By definition, each of these itemsets will occur at least as frequently as a predetermined minimum support count, *min sup*.

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

Efficient and Scalable Frequent Itemset Mining Methods

The Apriori Algorithm: Finding Frequent Itemsets Using Candidate Generation

Apriori is a seminal algorithm proposed by R. Agrawal and R. Srikant in 1994 for mining frequent itemsets for Boolean association rules. The name of the algorithm is based on the fact that the algorithm uses *prior knowledge* of frequent itemset properties, as we shall see following. Apriori employs an iterative approach known as a *level-wise* search, where *k*-itemsets are usedtoexplore (*k*+1)-itemsets. First, the setof frequent 1-itemsets is found by scanning the database to accumulate the count for each item, and collecting those items that satisfy minimum support. The resulting set is denoted *L*1.Next, *L*1 is used to find *L*2, the set of frequent 2-itemsets, which is used to find *L*3, and so on, until no more frequent *k*-itemsets can be found. The finding of each *Lk* requires one full scan of the database. To improve the efficiency of the level-wise generation of frequent itemsets, an important property called the Apriori property, presented below, is used to reduce the search space.We will first describe this property, and then show an example illustrating its use.

Apriori property:

All nonempty subsets of a frequent itemset must also be frequent. The Apriori property is based on the following observation. By definition, if an itemset *I* does not satisfy the minimum support threshold, *min sup*, then *I* is not frequent; that is, *P*(*I*) < *min sup*. If an item *A* is added to the itemset *I*, then the resulting itemset (i.e., *I* [*A*) cannot occur more frequently than *I*. Therefore, *I* [*A* is not frequent either; that is, *P*(*I* [*A]* < *min_sup*. This property belongs to a special category of properties called anti monotone in the sense that *if a set cannot pass a test, all of its supersets will fail the same test as well*. It is called *antimonotone* because the property is monotonic in the context of failing a test.7 *"How is the Apriori property used in the algorithm?"* To understand this, let us look at how *Lk*-1 is used to find *Lk* for *k*>= 2. A two-step process is followed, consisting of join and prune actions.

1.The Join Step:

To find *Lk*, a set of candidate *k*-itemsets is generated by joining *Lk*-1 with itself. This set of candidates is denoted *Ck*. Let L1 and L2 be itemsets in *Lk-1*. The notation L*i*[*j*] refers to the *j* th item in L*i* (e.g.,L1[*k-2*] refers to the second to the last item in L1).

2. The Prune step:

Ck is a superset of *Lk*, that is, its members may or may not be frequent, but all of the frequent *k*-itemsets are included in *Ck*. A scan of the database to determine the count of each candidate in *Ck* would result in the determination of *Lk* (i.e., all candidates having a count no less than the minimum support count are frequent by definition, and therefore belong to *Lk*).

Apriori. Let's look at a concrete example, based on the *AllElectronics* transaction database, *D*. There are nine transactions in this database, that is, *D*j = 9.
In the first iteration of the algorithm, each item is a member of the set of candidate 1-itemsets, *C*1. The algorithm simply scans all of the transactions in order to count the number of occurrences of each item.

2. Suppose that the minimum support count required is 2, that is, *min sup* = 2. (Here, we are referring to *absolute* support because we are using a support count. The corresponding relative support is 2/9 = 22%). The set of frequent 1-itemsets, *L*1, can then be determined. It consists of the candidate 1-itemsets satisfying minimum support.
In our example, all of the candidates in *C*1 satisfy minimum support.

3. To discover the set of frequent 2-itemsets, *L*2, the algorithm uses the join *L*1 on *L*1 to generate a candidate set of 2-itemsets, *C*2.8 *C*2 consists of □j*L*1j 2-itemsets. Note that no candidates are removed from*C*2 during the prune step because each subset of the candidates is also frequent.

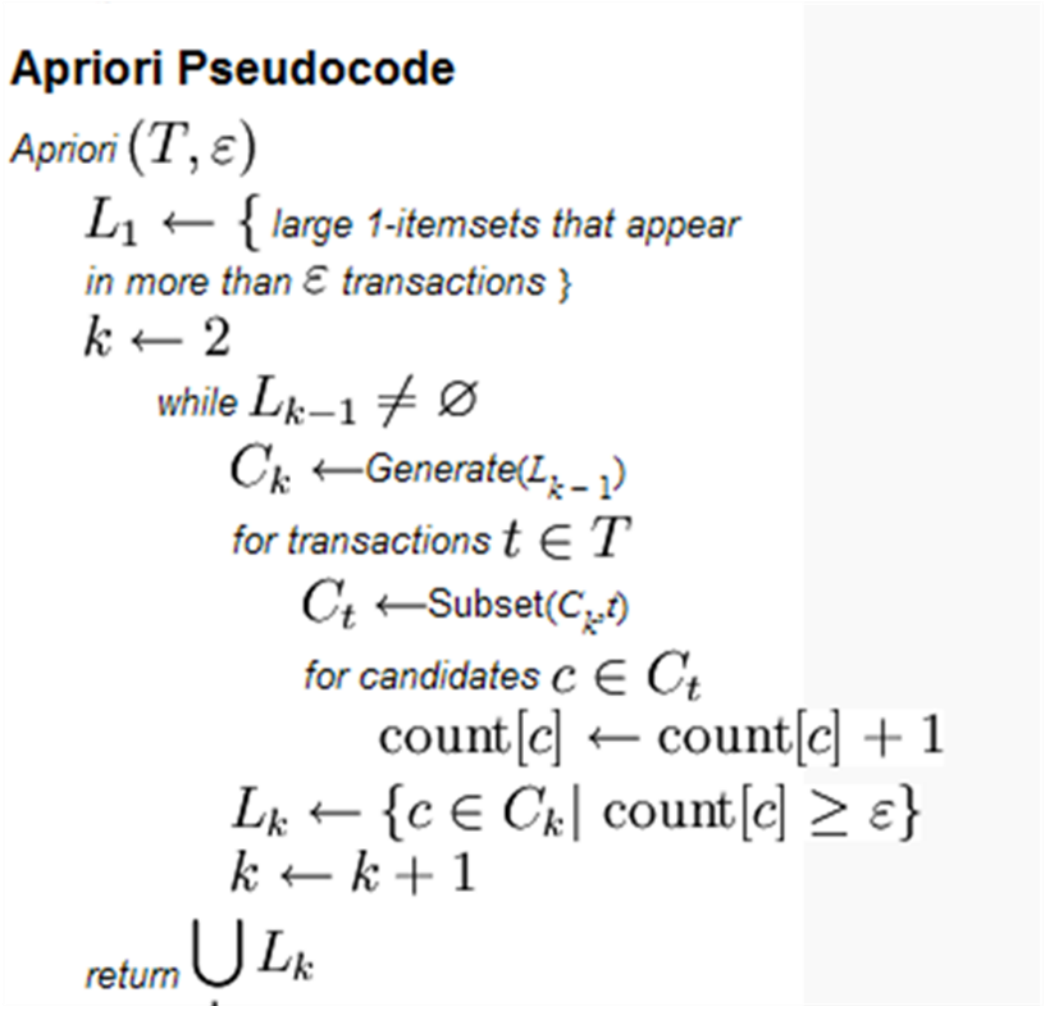
4. Next, the transactions in*D*are scanned and the support count of each candidate itemset. In *C*2 is accumulated, as shown in the middle table of the second row.

5. The set of frequent 2-itemsets, *L*2, is then determined, consisting of those candidate 2-itemsets in *C*2 having minimum support.

6. Based on the Apriori property that all subsets of a frequent itemset must also be frequent, we can determine that the four latter candidates cannot possibly be frequent.We therefore remove them from*C*3, thereby saving the effort of unnecessarily obtaining their counts during the subsequent scan of*D*to determine *L*3.

APRIORI ALGORITHM

L_1= {frequent items};
for (k= 2; L_{k-1} !=∅; k++) **do begin**
C_k= candidates generated from L_{k-1} (that is: cartesian product L_{k-1} x L_{k-1} and eliminating any
k-1 size itemset that is not frequent);
for each transaction t in database **do**
increment the count of all candidates in
C_k that are contained in t
L_k = candidates in C_k with *min_sup*
end
return $\cup_k L_k$;



Generating Association Rules from Frequent Itemsets

Once the frequent itemsets from transactions in a database *D* have been found, it is straightforward to generate strong association rules from them (where *strong* association rules satisfy both minimum support and minimum confidence). This can be done using the following Equation for confidence, which we show again here for completeness:

$$confidence(A \Rightarrow B) = P(B|A) = \frac{support\_count(A \cup B)}{support\_count(A)}.$$

The conditional probability is expressed in terms of itemset support count, where *support count*(*A U B*) is the number of transactions containing the itemsets *AUB*, and *support count*(*A*) is the number of transactions containing the itemset *A*. Based on this equation, association rules can be generated as follows:

For each frequent itemset *l*, generate all non empty subsets of *l*.

Generating association rules. Let's try an example based on the transactional data for *AllElectronics* shown in Table 5.1. Suppose the data contain the frequent itemset *l* = {I1,I2, I3}. What are the association rules that can be generated from *l* ? The non empty subsets of *l* are {I1,I2},{I1,I5},{I2,I5},{I1},{I2},{I5} The resulting association rules are as shown below, each listed with its confidence:

$$I1 \wedge I2 \Rightarrow I5, \quad confidence = 2/4 = 50\%$$
$$I1 \wedge I5 \Rightarrow I2, \quad confidence = 2/2 = 100\%$$
$$I2 \wedge I5 \Rightarrow I1, \quad confidence = 2/2 = 100\%$$
$$I1 \Rightarrow I2 \wedge I5, \quad confidence = 2/6 = 33\%$$
$$I2 \Rightarrow I1 \wedge I5, \quad confidence = 2/7 = 29\%$$
$$I5 \Rightarrow I1 \wedge I2, \quad confidence = 2/2 = 100\%$$

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

Improving the Efficiency of Apriori

"How can we further improve the efficiency of Apriori-based mining?" Many variations of the Apriori algorithm have been proposed that focus on improving the efficiency of the original algorithm. Several of these variations are summarized as follows:

- **Hash-based technique** (hashing itemsets into corresponding buckets):

A hash-based technique can be used to reduce the size of the candidate *k*-itemsets, *Ck*, for $k > 1$. For example, when scanning each transaction in the database to generate the frequent 1-itemsets, *L*1, from the candidate 1-itemsets in *C*1, we can generate all of the 2-itemsets for each transaction, hash (i.e., map) them into the different *buckets* of a *hash table* structure, and increase the corresponding bucket counts.
A 2-itemset whose corresponding bucket count in the hash table is below the support.

- **Transaction reduction (reducing the number of transactions scanned in future iterations):**

A transaction that does not contain any frequent *k*-itemsets cannot contain any frequent (*k*+1)-itemsets. Therefore, such a transaction can be marked or removed from further consideration because subsequent scans of the database for *j*-itemsets, where $j > k$, will not require it.

- **Partitioning (partitioning the data to find candidate itemsets):**

A partitioning technique can be used that requires just two database scans to mine the frequent itemsets . It consists of two phases. In Phase I, the algorithm subdivides the transactions of *D* into *n* non overlapping partitions.

- **Sampling (mining on a subset of the given data):**

The basic idea of the sampling approach is to pick a random sample *S* of the given data *D*, and then search for frequent itemsets in *S* instead of *D*. In this way, we trade off some degree of accuracy. against efficiency. The sample size of *S* is such that the search for frequent itemsets in *S* can be done in main memory, and so only one scan of the transactions in *S* is required overall. Because we are searching for frequent itemsets in *S* rather than in *D*, it is possible that we will miss some of the global frequent itemsets.

- **Dynamic itemset counting (adding candidate itemsets at different points during a scan):**

A dynamic itemset counting technique was proposed in which the database is partitioned into blocks marked by start points. In this variation, new candidate itemsets can be added at any start point, unlike in Apriori, which determines new candidate itemsets only immediately before each complete database scan.

Mining Frequent Itemsets without Candidate Generation

As we have seen, in many cases the Apriori candidate generate-and-test method significantly reduces the size of candidate sets, leading to good performance gain. However, it can suffer from two nontrivial costs:

It may need to generate a huge number of candidate sets. For example, if there are 10^4 frequent 1-itemsets, the Apriori algorithm will need to generate more than 10^7 candidate 2-itemsets.

It may need to repeatedly scan the database and check a large set of candidates by pattern matching. It is costly to go over each transaction in the database to determine the support of the candidate itemsets.

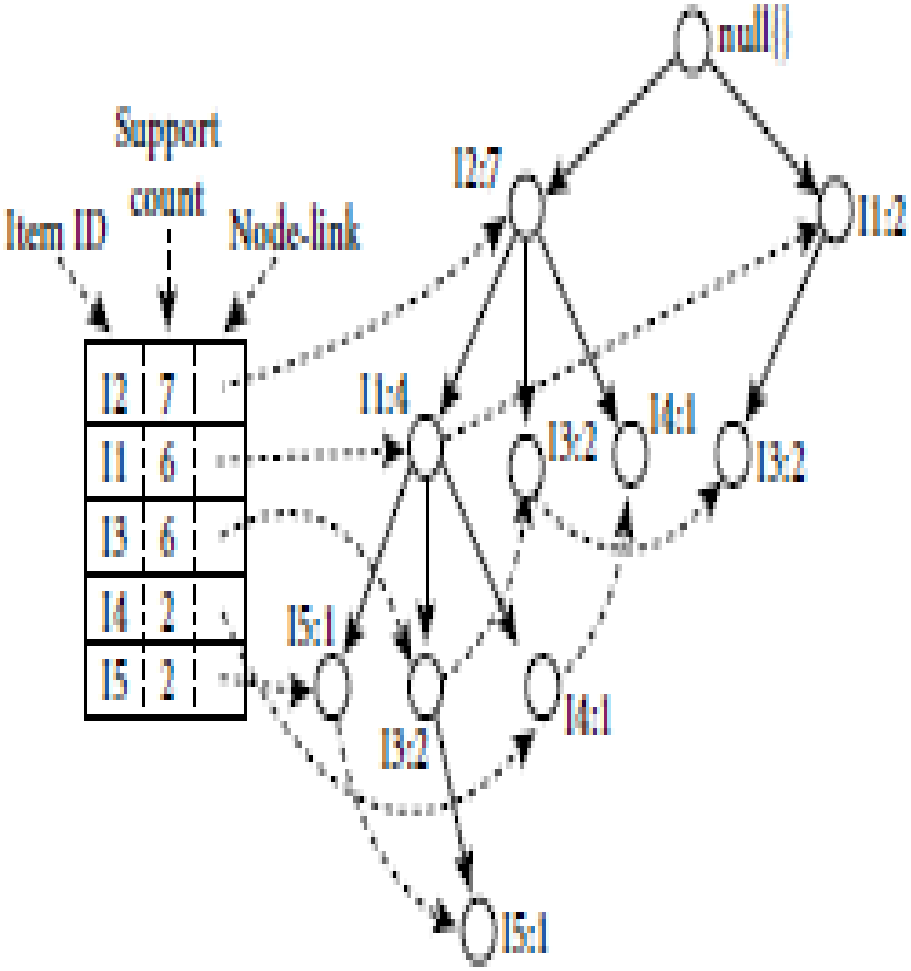
*"Can we design a method that mines the complete set of frequent itemsets without candidate generation?"*An interestingmethod in this attempt is called frequent-pattern growth, or simply FP-growth, which adopts a *divide-and-conquer* strategy as follows. First, it compresses the database representing frequent items into a frequent-pattern tree, or FP-tree,which retains the itemset association information. It then divides the compressed database into a set of *conditional databases* (a special kind of projected database), each associated with one frequent item or "pattern fragment," and mines each such database separately.

FP-growth (finding frequent itemsets without candidate generation). We re-examine the mining of transaction database, *D* using the frequent pattern growth (FP-growth) approach.

An FP-tree is then constructed as follows. First, create the root of the tree, labeled with "null." Scan database *D* a second time. The items in each transaction are processed in *L* order (i.e., sorted according to descending support count), and a branch is created for each transaction.

To facilitate tree traversal, an item header table is built so that each item points to its occurrences in the tree via a chain of node-links. The tree obtained after scanning all of the transactions is shown in the following Figure with the associated node-links. In this way, the problem of mining frequent patterns in databases is transformed to that of mining the FP-tree.

A study on the performance of the FP-growth method shows that it is efficient and scalable for mining both long and short frequent patterns, and is about an order of magnitude faster than the Apriori algorithm. It is also faster than a Tree-Projection algorithm,which recursively projects a database into a tree of projected databases.



Mining Frequent Itemsets Using Vertical Data Format

Both the Apriori and FP-growth methods mine frequent patterns from a set of transactions in *TID-itemset* format (that is, f*TID* : *itemset*g), where *TID* is a transaction-id and *itemset* is the set of items bought in transaction *TID*. This data format is known as horizontal data format. Alternatively, data can also be presented in *item-TID set* format (that is, f*item* : *TID set*g), where *item* is an item name, and *TID set* is the set of transaction identifiers containing the item. This format is known as vertical data format.
In this section, we look at how frequent itemsets can also be mined efficiently using vertical data format, which is the essence of the ECLAT (Equivalence CLASS Transformation) algorithm.

The vertical data format of the transaction data set *D* of Table 5.1.

Itemset	TID_set
I1	{T100, T400, T500, T700, T800, T900}
I2	{T100, T200, T300, T400, T600, T800, T900}
I3	{T300, T500, T600, T700, T800, T900}
I4	{T200, T400}
I5	{T100, T800}

Mining can be performed on this data set by intersecting the TID sets of every pair of frequent single items. The minimum support count is 2. There are 10 intersections performed in total, which lead to 8 nonempty 2-itemsets. Notice that because the itemsets {I1, I4} and {I3, I5} each contain only one transaction, they do not belong to the set of frequent 2-itemsets.

Based on the Apriori property, a given 3-itemset is a candidate 3-itemset only if every one of its 2-itemset subsets is frequent. The candidate generation process here will generate only two 3-itemsets: {I1, I2, I3} and {I1, I2, I5}. By intersecting the TID sets of any two corresponding 2-itemsets of these candidate 3-itemsets, it derives Table 5.5, where there are only two frequent 3-itemsets: {I1, I2, I3} and {I1, I2, I5}.

Algorithm: FP_growth. Mine frequent itemsets using an FP-tree by pattern fragment growth.

Input:

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

Output: The complete set of frequent patterns.

Method:

1. The FP-tree is constructed in the following steps:

   (a) Scan the transaction database D once. Collect F, the set of frequent items, and their support counts. Sort F in support count descending order as L, the *list* of frequent items.

   (b) Create the root of an FP-tree, and label it as "null." For each transaction *Trans* in D do the following. Select and sort the frequent items in *Trans* according to the order of L. Let the sorted frequent item list in *Trans* be $[p|P]$, where p is the first element and P is the remaining list. Call insert_tree($[p|P]$, T), which is performed as follows. If T has a child N such that $N.item\text{-}name = p.item\text{-}name$, then increment N's count by 1; else create a new node N, and let its count be 1, its parent link be linked to T, and its node-link to the nodes with the same *item-name* via the node-link structure. If P is nonempty, call insert_tree(P, N) recursively.

2. The FP-tree is mined by calling FP_growth(*FP_tree*, *null*), which is implemented as follows.

```
procedure FP_growth(Tree, α)
(1)  if Tree contains a single path P then
(2)      for each combination (denoted as β) of the nodes in the path P
(3)          generate pattern β ∪ α with support_count = minimum support count of nodes in β;
(4)  else for each a_i in the header of Tree {
(5)      generate pattern β = a_i ∪ α with support_count = a_i.support_count;
(6)      construct β's conditional pattern base and then β's conditional FP_tree Tree_β;
(7)      if Tree_β ≠ ∅ then
(8)          call FP_growth(Tree_β, β); }
```

The FP-growth algorithm for discovering frequent itemsets without candidate generation.

Mining Closed Frequent Itemsets

We saw how frequent itemset mining may generate a huge number of frequent itemsets, especially when the *min sup* threshold is set low or when there exist long patterns in the data set. Example showed that closed frequent itemsets can substantially reduce the number of patterns generated in frequent itemset mining while preserving the complete information regarding the set of frequent itemsets. That is, from the set of closed frequent itemsets, we can easily derive the set of frequent itemsets and their support. Thus in practice, it is more desirable to mine the set of closed frequent itemsets rather than the set of all frequent itemsets in most cases.

"How can we mine closed frequent itemsets?" A naïve approach would be to first mine the complete set of frequent itemsets and then remove every frequent itemset that is a proper subset of, and carries the same support as, an existing frequent itemset. However, this is quite costly.

- **Item merging:**

If every transaction containing a frequent itemset X also contains an itemset Y but not any proper superset of Y, then XUY forms a frequent closed itemset and there is no need to search for any itemset containing X but no Y.

- **Sub-itemset pruning:**

If a frequent itemset X is a proper subset of an already found frequent closed itemset Y and support count(*X*) = *support count*(*Y*)*, then X and all of X's descendants in the set enumeration tree cannot be frequent closed itemsets and thus can be pruned.*

- **Item skipping:**

In the depth-first mining of closed itemsets, at each level, there will be a prefix itemset X associated with a header table and a projected database. If a local frequent item p has the same support in several header tables at different levels, we can safely prune p from the header tables at higher levels.

Besides pruning the search space in the closed itemset mining process, another important optimization is to perform efficient checking of a newly derived frequent itemset to see whether it is closed, because the mining process cannot ensure that every generated frequent itemset is closed.

When a newfrequent itemset is derived, it is necessary to performtwo kinds of closure checking: (1) *superset checking*, which checks if this new frequent itemset is a superset of some already found closed itemsetswith the same support, and (2) *subset checking*,which checks whether the newly found itemset is a subset of an already found closed itemset with the same support.

If we adopt the *item merging* pruning method under a divide-and-conquer framework,then the superset checking is actually built-in and there is no need to explicitly perform superset checking. This is because if a frequent itemset X [Y is found later than itemset X, and carries the same support as X, it must be in X's projected database and must have been generated during itemset merging.

From Association Mining to Correlation Analysis

Most association rule mining algorithms employ a support-confidence framework. Often, many interesting rules can be found using low support thresholds. Although minimum support and confidence thresholds *help* weed out or exclude the exploration of a good number of uninteresting rules, many rules so generated are still not interesting to the users. Unfortunately, this is especially true *when mining at low support thresholds or mining for long patterns*. This has been one of the major bottlenecks for successful application of association rule mining.

In this section, we first look at how even strong association rules can be uninteresting and misleading. We then discuss how the support-confidence framework can be supplemented with additional interestingness measures based on statistical significance and correlation analysis.

- **Strong Rules Are Not Necessarily Interesting: An Example**

Whether or not a rule is interesting can be assessed either subjectively or objectively. Ultimately, only the user can judge if a given rule is interesting, and this judgment, being subjective, may differ fromone user to another. However, objective interestingness measures, based on the statistics "behind" the data, can be used as one step toward the goal of weeding out uninteresting rules from presentation to the user.
"How can we tell which strong association rules are really interesting?" Let's examine the following example :

A misleading "strong" association rule. Suppose we are interested in analyzing transactions at *AllElectronics* with respect to the purchase of computer games and videos. Let *game* refer to the transactions containing computer games, and *video* refer to those containing videos. Of the 10,000 transactions analyzed, the data show that 6,000 of the customer transactions included computer games, while 7,500 included videos, and 4,000 included both computer games and videos. Suppose that a data mining program for discovering association rules is run on the data, using a minimum support of, say, 30% and a minimum confidence of 60%. The following association rule is discovered:

$$buys(X, \text{"computer games"}) \Rightarrow buys(X, \text{"videos"}) \quad [support = 40\%, confidence = 66\%]$$

Rule is a strong association rule and would therefore be reported, since its support value of 4,000/ 10,000 =40% and confidence value of 4,000/ 6,000 =66% satisfy the minimum support and minimum confidence thresholds, respectively. However, the above Rule is misleading because the probability of purchasing videos is 75%, which is even larger than 66%. In fact, computer games and videos are negatively associated because the purchase of one of these items actually decreases the likelihood of purchasing the other. Without fully understanding this phenomenon, we could easily make unwise business decisions based on Rule.

- **From Association Analysis to Correlation Analysis**

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

$$A \Rightarrow B \ [support, confidence, correlation].$$

That is, a correlation rule is measured not only by its support and confidence but also by the correlation between itemsets *A* and *B*. There are many different correlation measures from which to choose. In this section, we study various correlation measures to determine which would be good for mining large data sets.
Lift is a simple correlation measure that is given as follows. The occurrence of itemset *A* is independent of the occurrence of itemset *B* if *P(AU B)* = *P*(*A*)*P*(*B*); otherwise, itemsets *A* and *B* are dependent and correlated as events. This definition can easily be extended to more than two itemsets. The lift between the occurrence of *A* and *B* can be measured by computing

$$lift(A, B) = \frac{P(A \cup B)}{P(A)P(B)}.$$

If the resulting value of Equation is less than 1, then the occurrence of *A* is *negatively correlated with* the occurrence of *B*. If the resulting value is greater than 1, then *A* and *B* are *positively correlated*, meaning that the occurrence of one implies the occurrence of the other. If the resulting value is equal to 1, then *A* and *B* are *independent* and there is no correlation between them.

Part B TWO MARK QUESTIONS

1.What is knowledge discovery process?

2.Define Data Discretization.

3.What do you mean by Mining frequent patterns?

4.What is decision tree induction?

5.What is spatial data mining?

PART-C 8 MARK QUESTIONS

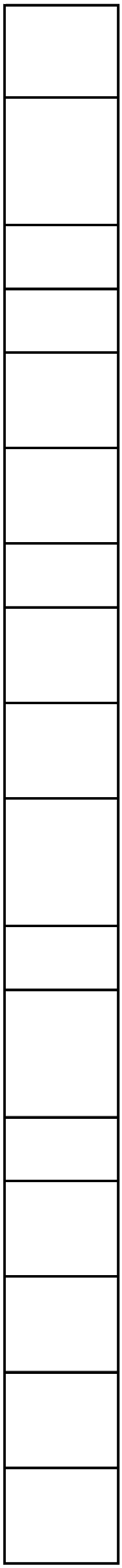
1.Discuss the techniques used to improve the efficiency of Apriori-based mining.
2.Explain the basic concept of mining frequent patterns.
3.Explain Apriori algorithm in detail.
4.Discuss Data Mining multilevel association rules.
5.Explain the Mining Frequent Item sets without Candidate Generation.
6.Explain how correlation analysis supports association rule analysis in Data Mining
7.Discuss market basket analysis with examples.
8.Explain quantitative association rule.
9.Explain any three efficient and scalable item set mining methods.
10.Discuss Multidimensional Association Rules of Data Mining.

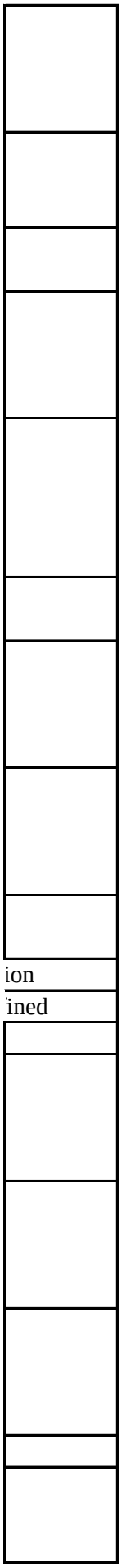
Questions	opt1	opt2	opt3	opt4	opt5	opt6	Answer
_____ model makes a prediction about values of data using known results found from different data.	predictive	descriptive	preference	process			predictive
_____ model identifies patterns or relationships in data.	predictive	descriptive	process	preference			descriptive
_____ maps data into predefined groups or classes	regression	classification	clustering	prediction			classification
_____ is used to map a data item to a real valued prediction variable.	regression	classification	clustering	prediction			regression
Classification of input patterns by using its similarity to the predefined classes is called____	prediction	classification	pattern recognition	regression			pattern recognition
A special type of clustering is called _____	classification	segmentation	regression	prediction			segmentation
_____ maps data into subsets with associated simple description.	classification	segmentation	pattern recognition	summarization			summarization
Which rule is used to identify the specific type of data association	delta rule	association rule	classification rule	segmentation rule			association rule
Which is used to determine sequential patterns in data?	sequence discovery	segmentation	regression	random analysis			sequence discovery
_____ is the process of finding useful information and patterns in data	datamining	KDD	selection	transformation			KDD
_____ is the use of algorithms to extract the information and pattern derived by the knowledge discovery	datamining	KDD	selection	transformation			datamining
The KDD process is often said to be _____	trivial	nontrivial	significant	consequential			nontrivial
Extreme values that occur infrequently, may actually be removed is called _____	outliers	outlayers	statistics	frequency			outliers
Traditional graph structure is a _____ visualization techniqe	graphical	geometric	icon-based	pixel-based			graphical
The major goal of _____ is to be able to describe the result in meaningful manner.	datamining	KDD	selection	transformation			KDD
_____ techniques often involve sophisticated multimedia and graphics presentations.	summarization	conceptualization	visualization	decription			visualization

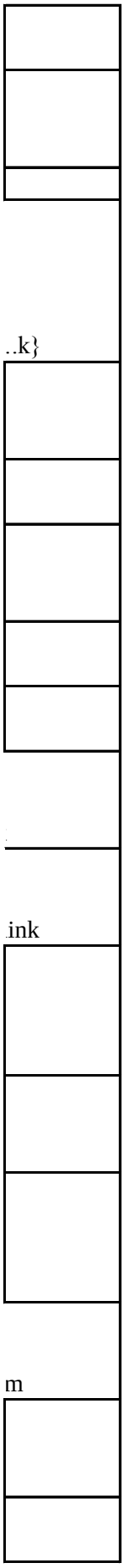
Describing a large database can be viewed as using _____ to help uncover hidden information about the data	approximation	search	induction	compression			approximation
_____ is used to proceed from very specific knowledge to more general information.	approximation	search	induction	compression			induction
_____ occurs when the model does not fit future states.	outliers	interception	overfitting	qureying			overfitting
The problem of increasing over all complexity and decrease the efficiency of an algorithms is _____	Single dimensionality	Multi-Dimensionality	Dimensionality curse	Recursion			Dimensionality curse
What is the expansion of ROI	return on investment	result of invension	research oriented investment	return of interest			return on investment
Find out predictive model datamining task from the followings	clustering	summarization	association rule	regression			regression
Find out descriptive model datamining task from the followings	classification	sequence discovery	prediction	regression			sequence discovery
Flooding, speech recognition, machine learning are_____	classification applicationa	prediction applications	clustering application	regression applications			prediction applications
Link analysis alternatively referred to as a _____	rule analysis	affinity analysis	master analysis	report analysis			affinity analysis
Clustering is similar to _____	Summariz	Classifica	Associati	Regression			Classificati
In clustering the groups are_____	Predefine	redefined	Not Prede	Not Redefined			Not Predef
In clustering the groups are called _____	Clusters	Outliers	Elements	Segments			Clusters
The distance between points in a cluster is _____ than the distance between a point in the cluster and any point outside it.	Lesser	Greater	Equal	More			Lesser
The term similar to clustering is_____ where like tupples in a database are grouped together.	Database segmentation	Distance Measure	Classification	Regression			Database segmentation
The first domain in which clustering was u	disease classification	biological taxonomy	Medicine	Economics			biological taxonomy
The element that donot naturally fall into a	Segment	Domain	Outlier	Medoid			Outlier
Dynamic data in the database implies that cluster membership may_____ over time.	Remain	Change	Increase	Decrease			Change

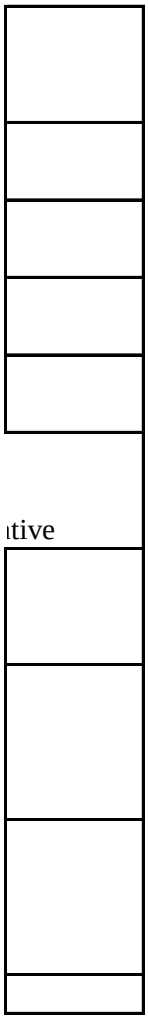
The _____ meaning of each cluster may be difficult.	Static	Dynamic	Semantic	Constant			Semantic
Clustering can be viewed similar to _____ .	Supervised learning	Conditional learning	Predefined learning	Unsupervised learning			Unsupervised learning
Clusters results are _____ .	Dynamic	Static	Increase	Decrease			Dynamic
Given a database D={t1,t2,….tn} of tuples and an integer value k, the clustering problem is to define a mapping _____ where each ti is assigned to one cluster kj, 1≤j≤k.	f:D_ {1,..	f.D={1,..	f:D→{1,.	f/d_{1,…k}			f:D→{1,..
Clustering algorithms themselves may be viewed as hierarchical or _____ .	Categorial	Large Database	Partitiona	Extrinsic			Partitional
In hierarchical a _____ set of clusters is created.	Nested	Unique	Separate	Combined			Nested
At the highest level in hierarchical clustering the all items belong to _____ clusters.	Separate	Same	Different	Distinct			Same
In partitional clustering the algorithm creates _____ set of clusters.	Many	Same	Unique	Only One			Only One
Non Overlapping clusters can be viewed as extrinsic or _____ .	Multrinsic	Intrinsic	Distinct	discrete			Intrinsic
Smallest distance between an element in one cluster and an element in the other is called as_____.	Multiple l	Complete	Single lin	Centroid			Single link
Largest distance between an element in one cluster and an element in the other is called as_____.	Multiple l	Complete	Single lin	Centroid			Complete l
If clusters have a representative centroid, then the centroid distance is defined as the distance between the _____.	Medoid	Outliers	Clusters	Centroid			Centroid
_____ are sample points with values much different from those of the remaning set of data.	Outlier	Centroid	Medoid	Cluster			Outlier
_____ is the process of identifying outlieers in a set of data.	Discordan	Outlier detection	Statistical Techniq ue	sequential test			Outlier detection
A tree data structure, called a _____ can be used to illustrate clustering technique.	Centroid	Outlier	Dendrogr	information gain			Dendrogra
The root in the dendrogram tree contains one cluster where all elements are _____.	Same	Different	Unique	Together			Together
The leaves in the dendrogram consists of a _____ element cluster.	Single	Same	Unique	Distinct			Single

Internal nodes in the dendrogram represent new clusters formed by _____ the clusters.	Seperating	Merging	Creating	Deleting			Merging
Hierarchical clustering algorithm is further divided into _____ .	1	2	3	4			2
The space complexity for hierarchical algorithms is _____.	O(n)	O/n	n	$O(n^2)$			$O(n^2)$
The space required for a dendrogram is _____ .	O(n)	$O(n^2)$	O(kn)	O(k)			O(kn)
The time complexity for hierarchical algorithms is _____ .	O(n)	$O(n^2)$	O(kn)	$O(kn^2)$			$O(kn^2)$
_____ algorithm start with each individual item in its own cluster and iteratively merge clusters.	Hierarchic	Partitiona	Agglomer	Divisive			Agglomera
The _____ link technique is based on the idea of finding maximal connected components in a graph.	Single	Multiple	Multileve	Hybrid			Single
A _____ is a graph in which there exists a path between any two vertices.	Disconne cted Compone nt	Connect ed Compon ent	Distinct compone nt	discrete value			Connecte d Compone nt
Two clusters are merged if there is at least one edge that connects the two clusters is often called as _____.	Single link	K Means clusterin g	Nearest neighbor	isolated value			Nearest neighbor




ion
ined


..k}
ink
m


ıtive

UNIT IV

Syllabus

Predictive and descriptive data mining techniques- supervised and unsupervised learning techniques- process of knowledge discovery in databases- pre-processing methods

Predictive and descriptive data mining techniques

Databases are rich with hidden information that can be used for intelligent decision making. Classification and prediction are two forms of data analysis that can be used to extract models describing important data classes or to predict future data trends. Such analysis can help provide us with a better understanding of the data at large. Whereas *classification* predicts categorical (discrete, unordered) labels, *prediction* models continuous valued functions. For example, we can build a classification model to categorize bank loan applications as either safe or risky, or a prediction model to predict the expenditures in dollars of potential customers on computer equipment given their income and occupation.

Many classification and prediction methods have been proposed by researchers in machine learning, pattern recognition, and statistics. Most algorithms are memory resident, typically assuming a small data size. Recent data mining research has built on such work, developing scalable classification and prediction techniques capable of handling large disk-resident data.

A bank loans officer needs analysis of her data in order to learn which loan applicants are "safe"and which are "risky" for the bank.A marketing manager at *AllElectronics* needs data analysis to help guess whether a customer with a given profile will buy a new computer. A medical researcher wants to analyze breast cancer data in order to predict which one of three specific treatments a patient should receive.

In each of these examples, the data analysis task is classification, where a model or classifier is constructed to predict *categorical labels*, such as "safe" or "risky" for the loan application data; "yes" or "no" for the marketing data; or "treatment A," "treatment B," or "treatment C" for the medical data. These categories can be represented by discrete values, where the ordering among values has no meaning. For example, the values 1, 2, and 3 may be used to represent treatments A, B,and C, where there is no ordering implied among this group of treatment regimes.

This first step of the classification process can also be viewed as the learning of a mapping or function, $y = f(\boldsymbol{X})$, that can predict the associated class label y of a given tuple $\boldsymbol{X}$. In this view, we wish to learn a mapping or function that separates the data classes. Typically, this mapping is represented in the form of classification rules, decision trees, or mathematical formulae. In our example, the mapping is represented as classification rules that identify loan applications as being either safe or risky.

Issues Regarding Classification and Prediction

This section describes issues regarding preprocessing the data for classification and prediction. Criteria for the comparison and evaluation of classification methods are also described.

Preparing the Data for Classification and Prediction

The following preprocessing steps may be applied to the data to help improve the accuracy, efficiency, and scalability of the classification or prediction process.

Data cleaning:

This refers to the preprocessing of data in order to remove or reduce *noise* (by applying smoothing techniques, for example) and the treatment of *missing values* (e.g., by replacing a missing value with the most commonly occurring value for that attribute, or with the most probable value based on statistics). Although most classification algorithms have some mechanisms for handling noisy or missing data, this step can help reduce confusion during learning.

Relevance analysis:

Many of the attributes in the data may be *redundant*. Correlation analysis can be used to identify whether any two given attributes are statistically related. For example, a strong correlation between attributes *A*1 and *A*2 would suggest that one of the two could be removed from further analysis. A database may also contain *irrelevant* attributes. Attribute subset selection4 can be used in these cases to find a reduced set of attributes such that the resulting probability distribution of the data classes is as close as possible to the original distribution obtained using all attributes.

Data transformation and reduction:

The data may be transformed by normalization, particularly when neural networks or methods involving distance measurements are used in the learning step. Normalization involves scaling all values for a given attribute so that they fall within a small specified range, such as -1:0 to 1:0, or 0:0 to 1:0. In methods that use distance measurements, for example, this would prevent attributes with initially large ranges (like, say, *income*) from out weighing attributes with initially smaller ranges (such as binary attributes).

Comparing Classification and Prediction Methods

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

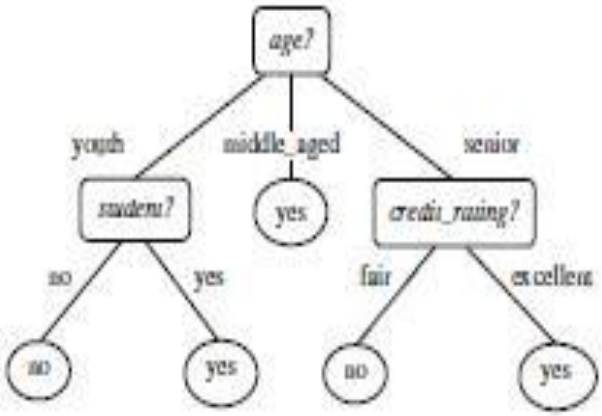
Accuracy:

The accuracy of a classifier refers to the ability of a given classifier to correctly predict the class label of new or previously unseen data (i.e., tuples without class

label information). Similarly, the accuracy of a predictor refers to how well a given predictor can guess the value of the predicted attribute for new or previously unseen data. Accuracy can be estimated using one or more test sets that are independent of the training set.
Speed: This refers to the computational costs involved in generating and using the given classifier or predictor.
Robustness: This is the ability of the classifier or predictor to make correct predictions given noisy data or data with missing values.
Scalability: This refers to the ability to construct the classifier or predictor efficiently given large amounts of data.
Interpretability: This refers to the level of understanding and insight that is provided by the classifier or predictor. Interpretability is subjective and therefore more difficult to assess.

Supervised and unsupervised learning techniques

Decision tree induction is the learning of decision trees from class-labeled training tuples. Adecision tree is a flowchart-like tree structure, where each internal node (nonleaf node) denotes a test on an attribute, each branch represents an outcomeof the test, and each leaf node (or *terminal node*) holds a class label. The topmost node in a tree is the root node.



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

A typical decision tree is shown in the above Figure. It represents the concept *buys computer*, that is, it predicts whether a customer at *AllElectronics* is likely to purchase a computer. Internal nodes are denoted by rectangles, and leaf nodes are denoted by ovals. Some decision tree algorithms produce only *binary* trees (where each internal node branches to exactly two other nodes), whereas others can produce nonbinary trees.

"How are decision trees used for classification?"
Given a tuple, ***X***, for which the associated class label is unknown, the attribute values of the tuple are tested against the decision tree. A path is traced from the root to a leaf node, which holds the class prediction for that tuple. Decision trees can easily be converted to classification rules.
"Why are decision tree classifiers so popular?"
The construction of decision tree classifiers does not require any domain knowledge or parameter setting, and therefore is appropriate for exploratory knowledge discovery.

Decision Tree Induction

During the late 1970s and early 1980s, J. Ross Quinlan, a researcher in machine learning, developed a decision tree algorithm known as ID3 (Iterative Dichotomiser). This work expanded on earlier work on *concept learning systems*, described by E. B. Hunt, J. Marin, and P. T. Stone. Quinlan later presented C4.5 (a successor of ID3), which became a benchmark to which newer supervised learning algorithms are often compared.

In 1984, a group of statisticians (L. Breiman, J. Friedman, R. Olshen, and C.Stone) published the book *Classification and Regression Trees* (CART), which described the generation of binary decision trees. ID3 and CART were invented independently of one another at around the same time, yet follow a similar approach for learning decision trees from training tuples. These two cornerstone algorithms spawned a flurry of work on decisiontree induction.ID3, C4.5, and CART adopt a greedy (i.e., nonbacktracking) approach in which decision trees are constructed in a top-down recursive divide-and-conquer manner. Most algorithms for decision tree induction also follow such a top-down approach.

Algorithm: Generate_decision_tree. Generate a decision tree from the training tuples of data partition D.

Input:

- Data partition, D, which is a set of training tuples and their associated class labels;
- *attribute_list*, the set of candidate attributes;
- *Attribute_selection_method*, a procedure to determine the splitting criterion that "best" partitions the data tuples into individual classes. This criterion consists of a *splitting_attribute* and, possibly, either a *split point* or *splitting subset*.

Output: A decision tree.

Method:

```
(1)  create a node N;
(2)  if tuples in D are all of the same class, C then
(3)      return N as a leaf node labeled with the class C;
(4)  if attribute_list is empty then
(5)      return N as a leaf node labeled with the majority class in D; // majority voting
(6)  apply Attribute_selection_method(D, attribute_list) to find the "best" splitting_criterion;
(7)  label node N with splitting_criterion;
(8)  if splitting_attribute is discrete-valued and
         multiway splits allowed then // not restricted to binary trees
(9)      attribute_list ← attribute_list − splitting_attribute; // remove splitting_attribute
(10) for each outcome j of splitting_criterion
     // partition the tuples and grow subtrees for each partition
(11)     let Dj be the set of data tuples in D satisfying outcome j; // a partition
(12)     if Dj is empty then
(13)         attach a leaf labeled with the majority class in D to node N;
(14)     else attach the node returned by Generate_decision_tree(Dj, attribute_list) to node N;
     endfor
(15) return N;
```

.3 Basic algorithm for inducing a decision tree from training tuples.

ID3, C4.5, and CART adopt a greedy (i.e., nonbacktracking) approach in which decision trees are constructed in a top-down recursive divide-and-conquer manner. Most algorithms for decision tree induction also follow such a top-down approach, which starts with a training set of tuples and their associated class labels.
The training set is recursively partitioned into smaller subsets as the tree is being built.

A basic decision tree algorithm is summarized in Figure 6.3. At first glance, the algorithm may appear long, but fear not! It is quite straightforward. The strategy is as follows.

The algorithm is called with three parameters: *D*, *attribute list*, and *Attribute selection method*. We refer to *D* as a data partition. Initially, it is the complete set of training tuples and their associated class labels. The parameter *attribute list* is a list of attributes describing the tuples.

Attribute selection Measures

An attribute selection measure is a heuristic for selecting the splitting criterion that "best" separates a given data partition, *D*, of class-labeled training tuples into individual classes. If we were to split *D* into smaller partitions according to the outcomes of the splitting criterion, ideally each partition would be pure (i.e., all of the tuples that fall into a given partition would belong to the same class). Conceptually, the "best" splitting criterion is the one that most closely results in such a scenario. Attribute selection measures are also known as splitting rules because they determine how the tuples at a given node are to be split. The attribute selection measure provides a ranking for each attribute describing the given training tuples. The attribute having the best score for the measure6 is chosen as the *splitting attribute* for the given tuples. If the splitting attribute is continuous-valued or if we are restricted to binary trees then, respectively, either a *split point* or a *splitting subset* must also be determined as part of the splitting criterion.

Information gain

ID3 uses information gain as its attribute selection measure. This measure is based on pioneering work by Claude Shannon on information theory, which studied the value or "information content" of messages. Let node *N* represent or hold the tuples of partition *D*. The attribute with the highest information gain is chosen as the splitting attribute for node *N*. This attribute minimizes the information needed to classify the tuples in the resulting partitions and reflects the least randomness or "impurity" in these partitions. Such an approach minimizes the expected number of tests needed to classify a given tuple and guarantees that a simple (but not necessarily the simplest) tree is found.
The expected information needed to classify a tuple in *D* is given by

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

where *pi* is the probability that an arbitrary tuple in *D* belongs to class *Ci* and is estimated by |*Ci*,,*D*|I,|*D*j|. A log function to the base 2 is used, because the information is encoded in bits. *Info(D)* is just the average amount of information needed to identify the class label of a tuple in *D*.
The information we have is based solely on theV proportions of tuples of each class. *Info(D)* is also known as the entropy of *D*. How much more information would we still need (after the partitioning) in order to arrive at an exact classification?

This amount is measured by

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

The term |Dj| / |D| acts as the weight of the *j*th partition. *InfoA*(*D*) is the expected information required to classify a tuple from *D* based on the partitioning by *A*. The smaller the expected information (still) required, the greater the purity of the partitions. Information gain is defined as the difference between the original information requirement (i.e., based on just the proportion of classes) and the new requirement (i.e., obtained after partitioning on *A*).
Information gain is defined as the difference between the original information requirement (i.e., based on just the proportion of classes) and the new requirement (i.e., obtained after partitioning on *A*). That is,

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

In other words, *Gain(A)* tells us how much would be gained by branching on *A*. It is the expected reduction in the information requirement caused by knowing the value of *A*. The attribute *A* with the highest information gain, (*Gain*(*A*)), is chosen as the splitting attribute at node *N*. This is equivalent to saying that we want to partition on the attribute *A* that would do the "best classification," so that the amount of information still required to finish classifying the tuples is minimal (i.e., minimum *InfoA*(*D*)).

Tree Pruning

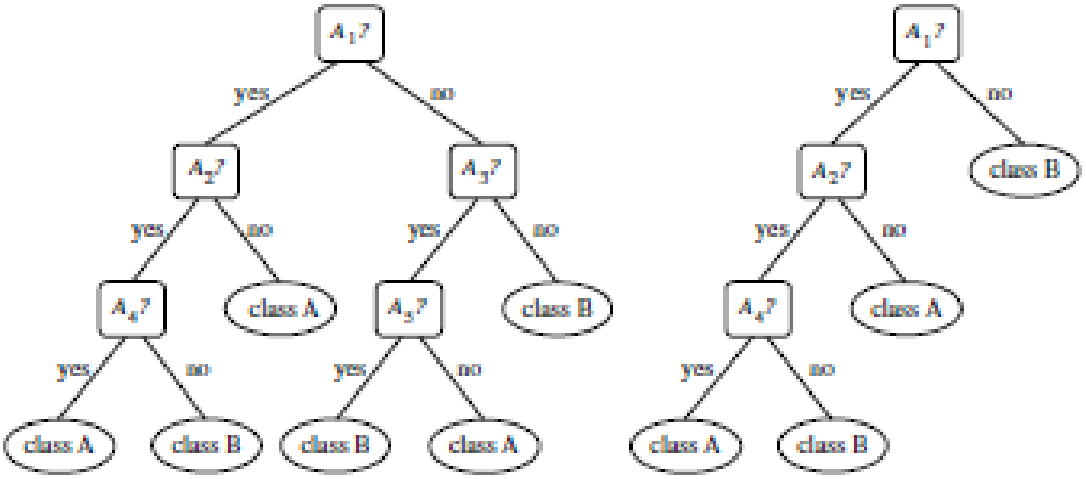
When a decision tree is built, many of the branches will reflect anomalies in the training data due to noise or outliers. Tree pruning methods address this problem of *overfitting* the data. Such methods typically use statistical measures to remove the least reliable branches. An unpruned tree and a pruned version of it are shown in Figure 6.6.
Pruned trees tend to be smaller and less complex and, thus, easier to comprehend. They are usually faster and better at correctly classifying independent test data (i.e., of previously unseen tuples) than unpruned trees.
"How does tree pruning work?" There are two common approaches to tree pruning:

prepruning and *postpruning*.

In the prepruning approach, a tree is "pruned" by halting its construction early (e.g.,by deciding not to further split or partition the subset of training tuples at a given node).Upon halting, the node becomes a leaf. The leaf may hold the most frequent class among the subset tuples or the probability distribution of those tuples. When constructing a tree, measures such as statistical significance, information gain, Gini index, and so on can be used to assess the goodness of a split. If partitioning the tuples at a node would result in a split that falls below a prespecified threshold, then further partitioning of the given subset is halted. There are difficulties, however, in choosing an appropriate threshold. High thresholds could result in oversimplified trees, whereas low thresholds could result in very little simplification.

The second and more common approach is postpruning, which removes subtrees from a "fully grown" tree. A subtree at a given node is pruned by removing its branches and replacing it with a leaf. The leaf is labeled with the most frequent class among the subtree being replaced. For example, notice the subtree at node "*A*3?" in the unpruned tree of Figure 6.6. Suppose that the most common class within this subtree is "*class B*." In the pruned version of the tree, the subtree in question is pruned by replacing it with the leaf "*class B*."

The cost complexity pruning algorithm used in CART is an example of the postpruning approach. This approach considers the cost complexity of a tree to be a function of the number of leaves in the tree and the error rate of the tree (where the error rate is the percentage of tuples misclassified by the tree).



An unpruned decision tree and a pruned version of it.

Process of knowledge discovery in databases

The efficiency of existing decision tree algorithms, such as ID3, C4.5, and CART, has been well established for relatively small data sets. Efficiency becomes an issue of concern when these algorithms are applied to the mining of very large real-world databases. The pioneering decision tree algorithms that we have discussed so far have the restriction that the training tuples should reside *in memory*. In data mining applications, very large training sets of millions of tuples are common.

Most often, the training data will not fit in memory! Decision tree construction therefore becomes inefficient due to swapping of the training tuples in and out of main and cache memories. More scalable approaches, capable of handling training data that are too large to fit in memory, are required. Earlier strategies to "save space" included discretizing continuous-valued attributes and sampling data at each node. These techniques, however, still assume that the training set can fit in memory.

Bayesian Classification

"What are Bayesian classifiers?" Bayesian classifiers are statistical classifiers. They can predict class membership probabilities, such as the probability that a given tuple belongs to a particular class. Bayesian classification is based on Bayes' theorem, described below. Studies comparing classification algorithms have found a simple Bayesian classifier known as the *naïve Bayesian classifier* to be comparable in performance with decision tree and selected neural network classifiers. Bayesian classifiers have also exhibited high accuracy and speed when applied to large databases.

Naïve Bayesian classifiers assume that the effect of an attribute value on a given class is independent of the values of the other attributes. This assumption is called *class conditional independence*. It is made to simplify the computations involved and, in this sense, is considered "naïve." *Bayesian belief networks* are graphical models, which unlike naïve
Bayesian classifiers, allow the representation of dependencies among subsets of attributes. Bayesian belief networks can also be used for classification.

Bayes' Theorem

Bayes' theorem is named after Thomas Bayes, a nonconformist English clergyman who did early work in probability and decision theory during the 18th century. Let $\boldsymbol{X}$ be a data tuple. In Bayesian terms, $\boldsymbol{X}$ is considered "evidence." As usual, it is described by measurements made on a set of n attributes. Let H be some hypothesis, such as that the data tuple $\boldsymbol{X}$ belongs to a specified class C. For classification problems, we want to determine $P(H|\boldsymbol{X})$, the probability that the hypothesis H holds given the "evidence" or observed data tuple $\boldsymbol{X}$. In other words, we are looking for the probability that tuple $\boldsymbol{X}$ belongs to class C, given that we know the attribute description of $\boldsymbol{X}$.

$P(H|\boldsymbol{X})$, is the posterior probability, or *a posteriori probability*, of H conditioned on $\boldsymbol{X}$. For example, suppose our world of data tuples is confined to customers described by the attributes *age* and *income*, respectively, and that $\boldsymbol{X}$ is a 35-year-old customer with an income of $40,000. Suppose that H is the hypothesis that our customer will buy a computer. Then $P(H|\boldsymbol{X})$, reflects the probability that customer $\boldsymbol{X}$ will buy a computer given that we know the customer's age and income. In contrast, $P(H)$ is the prior probability, or *a priori probability*, of H.

Similarly, $P(H|\boldsymbol{X})$ is the posterior probability of $\boldsymbol{X}$ conditioned on H. That is, it is the probability that a customer, $\boldsymbol{X}$, is 35 years old and earns $40,000, given that we know the customer will buy a computer. $P(\boldsymbol{X})$ is the prior probability of $\boldsymbol{X}$. Using our example, it is the probability that a person from our set of customers is 35 years old and earns $40,000.

"How are these probabilities estimated?" $P(H)$, $P(H|\boldsymbol{X})$, and $P(\boldsymbol{X})$ may be estimated from the given data, as we shall see below. Bayes' theorem is useful in that it provides a way of calculating the posterior probability, $P(H|\boldsymbol{X})$, from $P(H)$, $P(H|\boldsymbol{X})$, and $P(\boldsymbol{X})$.

Bayes' theorem is

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

Naïve Bayesian Classification

The naïve Bayesian classifier, or simple Bayesian classifier, works as follows:

1. Let *D* be a training set of tuples and their associated class labels.
As usual, each tuple is represented by an *n*-dimensional attribute vector, ***X*** = (*x*1, *x*2, : : : , *xn*), depicting *n* measurements made on the tuple from *n* attributes, respectively, *A*1, *A*2, : : : , *An*.
2. Suppose that there are *m* classes, *C*1, *C*2, : : : , *Cm*. Given a tuple, ***X***, the classifier will predict that ***X*** belongs to the class having the highest posterior probability, conditioned on ***X***. That is, the naïve Bayesian classifier predicts that tuple ***X*** belongs to the class *Ci* if and only if

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

Thus we maximize $P(\mathrm{Cj}|\boldsymbol{X})$. The class C_i for which $P(\mathrm{Ci}|\boldsymbol{X})$is maximized is called the *maximum posteriori hypothesis*. By Bayes' theorem

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

3. As $P(\boldsymbol{X})$ is constant for all classes, only P(X|Ci))$P(Ci)$ need be maximized. If the class prior probabilities are not known, then it is commonly assumed that the classes are equally likely, that is, $P(C1) = P(C2) = \ldots = P(Cm)$.

4. Given data sets with many attributes, it would be extremely computationally expensive to compute P(X|Ci). In order to reduce computation in evaluating P(X|Ci), the naive assumption of class conditional independence is made.

Bayesian Belief Networks

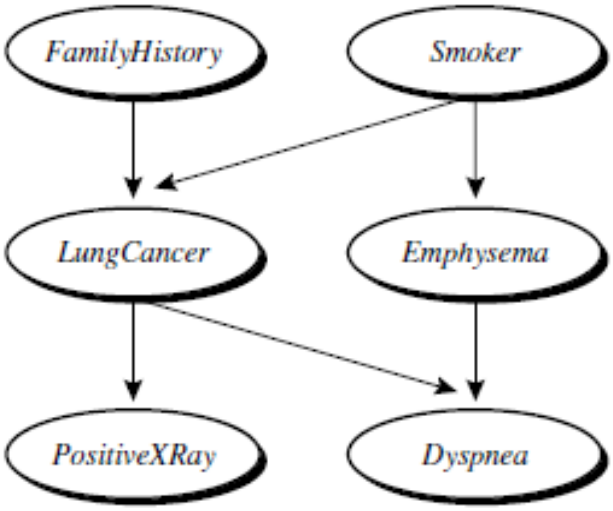
The naive Bayesian classifier makes the assumption of class conditional independence, that is, given the class label of a tuple, the values of the attributes are assumed to be conditionally independent of one another. This simplifies computation. When the assumption holds true, then the naïve Bayesian classifier is the most accurate in comparison with all other classifiers. In practice, however, dependencies can exist between variables.

Bayesian belief networks specify joint conditional probability distributions. They allow class conditional independencies to be defined between subsets of variables. They provide a graphical model of causal relationships, on which learning can be performed. Trained Bayesian belief networks can be used for classification. Bayesian belief networks are also known as belief networks, Bayesian networks, and probabilistic networks. For brevity, we will refer to them as belief networks.

A belief network is defined by two components—a *directed acyclic graph* and a set of *conditional probability tables*. Each node in the directed acyclic graph represents a random variable. The variables may be discrete or continuous-valued. They may correspond to actual attributes given in the data or to "hidden variables" believed to form a relationship (e.g., in the case ofmedical data, a hidden variablemay indicate a syndrome, representing a number of symptoms that, together, characterize a specific disease). Each

arc represents a probabilistic dependence.
If an arc is drawn from a node *Y* to a node *Z*. then *Y* is a parent orimmediate predecessor of *Z*, and *Z* is a descendant of*Y*. *Each variable is conditionally independent of its nondescendants in the graph, given its parents.*

(a)



(b)

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

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

The above figure is a simple belief network for six Boolean variables. The arcs in the above figure allow representation of causal knowledge. For example, having lung cancer is influenced by a person's family history of lung cancer, as well as whether or not the person is a smoker. Note that the variable *PositiveXRay* is independent of whether the patient has a family history of lung cancer or is a smoker, given that we know the patient has lung cancer. For instance, from the upper leftmost and bottom rightmost entries, respectively, we see that

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

Training Bayesian Belief Networks

"How does a Bayesian belief network learn?" In the learning or training of a belief network, a number of scenarios are possible. The network topology (or "layout" of nodes and arcs) may be given in advance or inferred from the data.

The network variables may be *observable* or *hidden* in all or some of the training tuples. The case of hidden data is also referred to as *missing values* or *incomplete data*. Several algorithms exist for learning the network topology from the training data given observable variables. The problem is one of discrete optimization.

Human experts usually have a good grasp of the direct conditional dependencies that hold in the domain under analysis, which helps in network design. Experts must specify conditional probabilities for the nodes that participate in direct dependencies. These probabilities can then be used to compute the remaining probability values.

If the network topology is known and the variables are observable, then training the network is straightforward. It consists of computing the CPT entries, as is similarly done when computing the probabilities involved in naive Bayesian classification.

1. Compute the gradients: For each *i*, *j*, *k*, compute

$$\frac{\partial \ln P_w(D)}{\partial w_{ijk}} = \sum_{d=1}^{|D|} \frac{P(Y_i = y_{ij}, U_i = u_{ik} | \boldsymbol{X}_d)}{w_{ijk}}.$$

.

The probability in the right-hand side of Equation is to be calculated for each training tuple, ***Xd***, in *D*. For brevity, let's refer to this probability simply as p.

2. Take a small step in the direction of the gradient: The weights are updated by

$$w_{ijk} \leftarrow w_{ijk} + (l) \frac{\partial \ln P_w(D)}{\partial w_{ijk}},$$

where *l* is the learning rate representing the step size and $\frac{\partial \ln P_w(D)}{\partial w_{ijk}}$ is computed from Equation . The learning rate is set to a small constant and helps with convergence.

3. **Renormalize the weights:**

Because the weights *wi jk* are probability values, they must be between 0.0 and 1.0, and*å*j *wi jk* must equal 1 for all *i*, *k*. These criteria are achieved by renormalizing the weights after they have been updated by the above Equation.

Rule-Based Classification

We look at rule-based classifiers, where the learned model is represented as a set of IF-THEN rules. We first examine how such rules are used for classification. We then study ways in which they can be generated, either froma decision tree or directly from the training data using a *sequential covering algorithm.*

Using IF-THEN Rules for Classification

Rules are a good way of representing information or bits of knowledge. A rule-based classifier uses a set of IF-THEN rules for classification. An IF-THEN rule is an expression of the form.

IF *condition* THEN *conclusion*.

An example is rule *R*1,

R1: IF *age = youth* AND *student = yes* THEN *buys computer = yes*.

The "IF"-part (or left-hand side)of a rule isknownas the rule antecedent or precondition. The "THEN"-part (or right-hand side) is the rule consequent. In the rule antecedent, the condition consists of one or more *attribute tests* (such as *age = youth*, and *student = yes*) that are logically ANDed. The rule's consequent contains a class prediction.

A rule *R* can be assessed by its coverage and accuracy. Given a tuple, ***X***, from a class labeled data set,*D*, let n_{covers} be the number of tuples covered by *R*; $n_{correct}$ be the number of tuples correctly classified by *R*; and |*D*| be the number of tuples in *D*. We can define the coverage and accuracy of *R* as

$$coverage(R) = \frac{n_{covers}}{|D|}$$

$$accuracy(R) = \frac{n_{correct}}{n_{covers}}.$$

That is, a rule's coverage is the percentage of tuples that are covered by the rule (i.e., whose attribute values hold true for the rule's antecedent). For a rule's accuracy, we look at the tuples that it covers and see what percentage of them the rule can correctly classify.

Rule Extraction from a Decision Tree

We learned how to build a decision tree classifier from a set of training data. Decision tree classifiers are a popular method of classification—it is easy to understand how decision trees work and they are known for their accuracy. Decision trees can become large and difficult to interpret. In this subsection, we look at how to build a rule based classifier by extracting IF-THEN rules from a decision tree. In comparison with a

decision tree, the IF-THEN rules may be easier for humans to understand, particularly if the decision tree is very large.

To extract rules from a decision tree, one rule is created for each path from the root to a leaf node. Each splitting criterion along a given path is logically ANDed to form the rule antecedent ("IF" part). The leaf node holds the class prediction, forming the rule consequent ("THEN" part).

*R*1: IF *age = youth* AND *student = no* THEN *buys computer = no*
*R*2: IF *age = youth* AND *student = yes* THEN *buys computer = yes*
*R*3: IF *age = middle aged* THEN *buys computer = yes*
*R*4: IF *age = senior* AND *credit rating = excellent* THEN *buys computer = yes*
*R*5: IF *age = senior* AND *credit rating = fair* THEN *buys computer = no*

A disjunction (logical OR) is implied between each of the extracted rules. Because the rules are extracted directly from the tree, they are mutually exclusive and exhaustive. By *mutually exclusive*, this means that we cannot have rule conflicts here because no two rules will be triggered for the same tuple.

"How can we prune the rule set?" For a given rule antecedent, any condition that does not improve the estimated accuracy of the rule can be pruned (i.e., removed), thereby generalizing the rule. C4.5 extracts rules from an unpruned tree, and then prunes the rules using a pessimistic approach similar to its tree pruning method. The training tuples and their associated class labels are used to estimate rule accuracy.

Pre-processing methods

Methods of data preprocessing

- Data cleaning: fill in missing values, smooth noisy data, identify or remove outliers, and resolve inconsistencies.
- Data integration: using multiple databases, data cubes, or files.
- Data transformation: normalization and aggregation.
- Data reduction: reducing the volume but producing the same or similar analytical results.
- Data discretization: part of data reduction, replacing numerical attributes with nominal ones.

Data cleaning

1. Fill in missing values (attribute or class value):
    - Ignore the tuple: usually done when class label is missing.

    - Use the attribute mean (or majority nominal value) to fill in the missing value.
    - Use the attribute mean (or majority nominal value) for all samples belonging to the same class.
    - Predict the missing value by using a learning algorithm: consider the attribute with the missing value as a dependent (class) variable and run a learning algorithm (usually Bayes or decision tree) to predict the missing value.
2. Identify outliers and smooth out noisy data:
    - Binning
        - Sort the attribute values and partition them into bins (see "Unsupervised discretization" below);
        - Then smooth by bin means, bin median, or bin boundaries.
    - Clustering: group values in clusters and then detect and remove outliers (automatic or manual)
    - Regression: smooth by fitting the data into regression functions.
3. Correct inconsistent data: use domain knowledge or expert decision.

Data transformation

1. Normalization:
    - Scaling attribute values to fall within a specified range.
        - Example: to transform `V in [min, max]` to `V' in [0,1]`, apply `V'=(V-Min)/(Max-Min)`
    - Scaling by using mean and standard deviation (useful when min and max are unknown or when there are outliers): `V'=(V-Mean)/StDev`
2. Aggregation: moving up in the concept hierarchy on numeric attributes.
3. Generalization: moving up in the concept hierarchy on nominal attributes.
4. Attribute construction: replacing or adding new attributes inferred by existing attributes.

Data reduction

1. Reducing the number of attributes
    - Data cube aggregation: applying roll-up, slice or dice operations.
    - Removing irrelevant attributes: attribute selection (filtering and wrapper methods), searching the attribute space (see Lecture 5: Attribute-oriented analysis).
    - Principle component analysis (numeric attributes only): searching for a lower dimensional space that can best represent the data..
2. Reducing the number of attribute values

- Binning (histograms): reducing the number of attributes by grouping them into intervals (bins).
- Clustering: grouping values in clusters.
- Aggregation or generalization

3. Reducing the number of tuples
    - Sampling

PART-B 2 MARK QUESTIONS

1.List out any four functionalities of Data Mining.

2.What are the benefits of wavelet transforms in Data Mining?

3.Define Quantitative Association rule.

4.List out any four issues in Data Classification.

5.What are the trends in Data Mining?

PART-C 8 MARK QUESTIONS

1.Explain the attribute selection measures of decision tree induction.
2.Discuss the Naive Bayesian classification.
3.Explain how Classification problems are solved using Decision Tree based algorithms.
4.Explain the training process of Bayesian Belief Networks.
5.Explain the constraint-based cluster analysis.
6.Discuss the three methods of prediction.
7.Explain the issues regarding classification and prediction.
8.Explain how to evaluate the accuracy of a Classifier or Predictor in Data Mining.
9.Explain rule induction using a sequential covering algorithm.
10.Discuss the usage of decision tree induction in the data classification process.

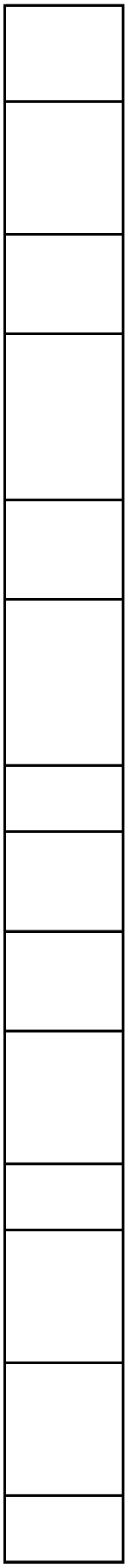
Questions	opt1	opt2	opt3	opt4	opt5	opt6	Answer
The term MST is referred to as _____.	Minimum Spanning tree	Maximum Spanning tree	Multiple Spanning Tree	single Spanning Tree			Minimum Spanning tree
_____ link algorithm looks for cliques rather than connected components.	Single	Complete	Multiple	Average			Complete
A variation of the complete link algorithm is called the _____.	Nearest Neighbor algorithm	Single link algorithm	multiple link algorithm	Farthest neighbor algorithm			Farthest neighbor algorithm
A _____ is a maximal graph in which there is an edge between any two vertices.	Cluster	Dendrogram	Clique	None			Clique
The _____ link technique merges two clusters if the average distance between any two points in the two target clusters is below the distance threshold.	Single	Complete	Multiple	Average			Average
In _____ clustering, all items are initially placed in one cluster.	Divisive	Partitional	Agglomerative	Hierarchical			Divisive
Partitional clustering is otherwise called as _____ .	Hierarchical	Non-Hierarchical	Divisive	Agglomerative			Non-Hierarchical
In _____ clustering only one set of clusters may be created internally within the various algorithms.	Hierarchical	Partitional	Divisive	Agglomerative			Partitional
The common measure is _____ which measures the squared distance from each point to the centroid for the associated cluster.	Squared error metric	Non Squared error Metric	metroid	chi square			Squared error metric
The _____ error clustering algorithm minimizes the squared error.	Squared	Non Squared	metroid	k - means			Squared
The squared error for a cluster is the _____ of the squared Euclidean distances between each element in the cluster.	Difference	Multiple	Sum	union			Sum
_____ is an iterative algorithm in which items are moved among sets of clusters until the desired set is reached.	Squared error metric	Partitional	Hierarchical	K- Means			K- Means
The time complexity of K-means is _____.	O(n)	$O(n^2)$	O(kn)	O(tkn)			O(tkn)

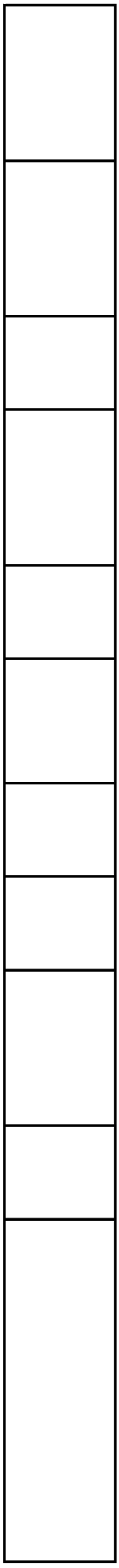
An algorithm similar to the single link technique is called as _____.	Nearest Neighbor algorithm	Multiple link algorithm	Farthest neighbor algorithm	Mutual link algorithm			Nearest Neighbor algorithm
The term PAM refers to _____	Predefined Account Medoids	Partitioning Around Medoids	Partitioning Along Medoids	Predefined Account Mutex			Partitioning Around Medoids
The PAM algorithm also called as _____ algorithm.	K- Means	Clustering	K-Medoids	Partitioning			K-Medoids
The term BEA refers to _____.	Bond Ellipsing Algorithm	Bond Estimated Algorithm	Bond Estimating Algorithm	Bond Energy Algorithm			Bond Energy Algorithm
With BEA, the _____ between database attributes is based on common usage.	Affinity	Vertical fragment	Bond matrix	horizontal segment			Affinity
_____ that use unsupervised learning attempt to find features in data that characterize the desired output	Network	Neural Network	Connected Network	fuzzy			Neural Network
In _____ learning, the weight between two nodes is changed to be proportional to both output values.	Competitive	Self-Organizing	Non-Competitive	isolated			Non-Competitive
Non Competitive learning is otherwise called as _____ .	Self-Organizing	Organizing	Neurons	Hebbian			Hebbian
SOFM stands for _____.	Self-Organizing Feature Map	Self Organized Feature Map	Single Organizing Feature Map	Single Organized Feature Map			Self-Organizing Feature Map
SOM stands for _____.	Self Organized Map	Self - Organizing Map	Single Organized Map	Single Organizing Map			Self - Organizing Map
BIRCH Stands for _____ .	Balanced Iterative Redusing And Clustering Using Hierarchies	Balancing Iterative Redusing And Clustering Using Hierarchies	Balance Iterative Redusing And Clustering Using Hierarchies	Balance Iterative Reliable And Clustering Using Hierarchies			Balanced Iterative Redusing And Clustering Using Hierarchies

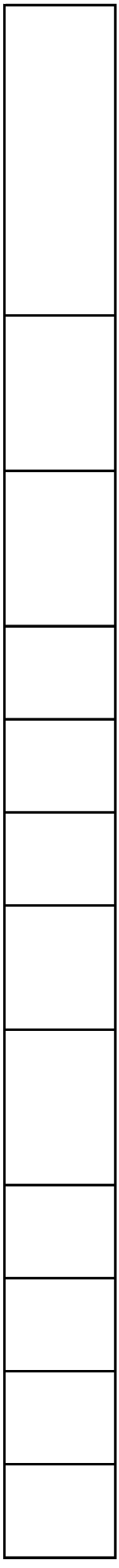
DBSCAN stands for _____ .	Density Based Scanning clustering of Appliances with Noise	Density Based Scanning clustering of Application with Noise	Density Based Spatial Clustering of Application with Noise	Density Based Spatial Clustering of Application without Note			Density Based Spatial Clustering of Application with Noise
CURE stands for _____.	Clustering Using Representatives	Clustering Using REpresentation	Clustered Using Representatives	Clue Using Representatives			Clustering Using REpresentatives
ROCK stands for _____.	ROdant Clustering using linKs	RObust Clustering using linKs	ROtated Clustering using linKs	Rest Clustering using linKs			RObust Clustering using linKs
_____ are frequently used by retail stores to assist in marketing, advertising,….	Classification	Association rules	Clustering	Neural Networks			Association rules
_____ are used to show the relationship between data items.	Classification	Association rules	Clustering	Neural Networks			Association rules
A _____ is an itemset whose number of occurrences is above a threshold.	Large Itemset	Small Itemset	Equal Itemset	least Update			Large Itemset
The _____ is the most well known association rule algorithm.	Priori Algorithm	Posteriori Algorithm	Apriori Algorithm	Aposteriori Algorithm			Apriori Algorithm
The basic idea of the _____ is to generate candidate itemsets of a particular size and then scan the database to count these to see if they are large.	Priori Algorithm	Posteriori Algorithm	Apriori Algorithm	Aposteriori Algorithm			Apriori Algorithm
The itemset in sampling algorithm is viewed as ____ itemsets.	Potentially Small	Potentially Large	Potentially Equal	None			Potentially Large
The candidates in the sampling algorithm is determined by applying the _____ function.	Positive Border	Equal Border	Largest Border	Negative Border			Negative Border
_____ may improve the performance of finding large itemset in several ways.	Partitioning	Combining	Seperating	Dividing			Partitioning
In _____ parallelism, the candidates are partitioned and counted seperately at each processor.	Data	task	Count	mutual			task

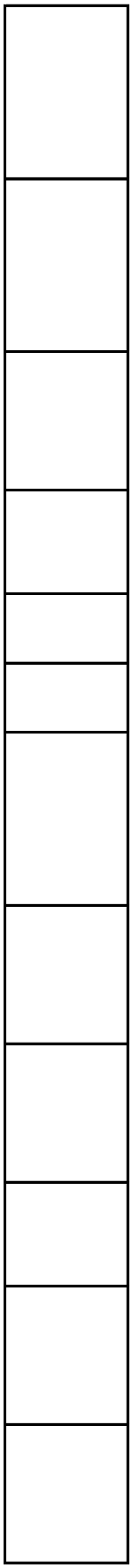
CDA stands for _____.	Count Distribution Algorithm	Count Distributed Algorithm	Count Distributing Algorithm	Cost Distributing Algorithm			Count Distribution Algorithm
DDA stands for _____.	Data Description Algorithm	Data Definition Algorithm	Data Distribution Algorithm	discrete data algorithm			Data Distribution Algorithm
The data distribution algorithm demonstrates _____.	Data parellelism	Task Parellelism	Count Parellelism	sequential Parellelism			Task Parellelism
_____ technique look at how to improve on the performance of an algorithm givrn data distribution.	Efficiency	Effectiveness	Optimization	Update			Optimization
FUP stands for _____.	First UPdate	Fast UPdate	Firm UPdate	target			Fast UPdate
_____ is basec on Apriori algorithm.	First UPdate	Fast UPdate	Firm UPdate	least Update			Fast UPdate
A generalized association rule, _____ is defined like regular association rule the restriction that no item in Y may be above any item in X.	X=Y	X.Y	X!=Y	X=> Y			X=> Y
A _____, X=> Y is defined like regular association rule the restriction that no item in Y may be above any item in X.	Generalized association rule	Increemental rule	Multiple level association rule	single level association rule			Generalized association rule
A variation of generalized rules are _____.	Generalized association rule	Increemental rule	Multiple level association rule	single level association rule			Multiple level association rule
When large itemsets are found at level I, large itemsets are generated for level _____.	i-1	i=1	i<1	i+1			i+1
A _____ is the one that involves categorical and quantitative data.	Generalized association rule	Increemental rule	Multiple level association rule	Quantitative association rule			Quantitative association rule
A _____ is defined as a set of itemsets that are correlated.	Generalized association rule	Correlation rule	Multiple level association rule	Quantitative association rule			Correlation rule

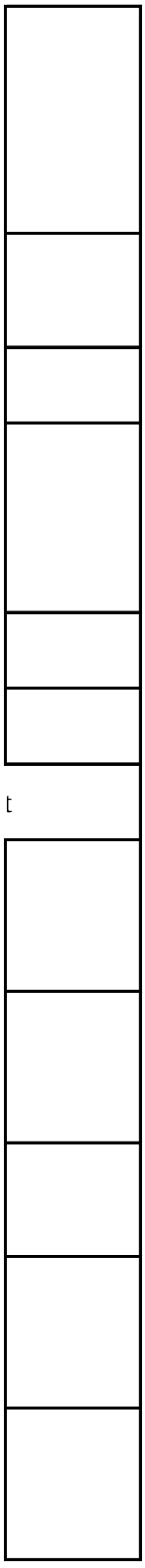
The formula for conviction is _____.	Conviction(A=> B) =P(A)P(¬ B)/P(A,¬ B)	Conviction(A= B) =P(A)P(¬B)/P(A,¬B)	Conviction(A=> B) =P(A)P(¬B)+P(A,¬B)	Conviction(A=> B) =P(A)+P(B)			Conviction(A=> B) =P(A)P(¬ B)/P(A,¬ B)
_____ is the count of number of transactions that contain the items in X.	O(X)	O(n²)	O(kn)	O(n)			O(X)
An _____ is a network of perceptrons	multisection	multifram	multilaye	multiscene			multilayer
_____ theorem states that a mapping between two sets of numbers can be performed using an neural network with only one hidden layer.	boolean	gaussian	bayes	kolmogorov			kolmogorov
A _____ rule consists of if part and then part	classification	associatic	bayes	incremental			classification
The if part in classification rule is called as _____	antecedent	conseque	prepositic	composition			antecedent
The then part in classification rule is called as _____	antecedent	conseque	prepositic	composition			consequent
The _____ algorithm is to cluster output values with the associated hidden nodes and input	genetic algorithm	expectation maximization	RX	crossover			RX
_____ algorithm attempt to generate rules exactly cover a specific class	genetic algorithm	covering	expectation maximization	RX			covering
A _____ of approaches takes multiple techniques and blemd them into a new approach	combined	semantic	analysis	synthesis			synthesis
One simple approach is called _____ which generates a simple set of rules that are equivalent to a DT with only one level	1R	2R	3R	4R			1R
_____ algorithm generate a rule for each leaf node in the decision tree	genetic algorithm	gen	expectation maximization	RX			gen










t

UNIT V

Data Mining Techniques: Association Rule Mining, classification and regression techniques, clustering, Scalability and data management issues in data mining algorithms, measures of interestingness

Association Rule Mining:

If/Then statements that help discover relationships between seemingly independent relational databases or other data repositories.

Most machine learning algorithms work with numeric datasets and hence tend to be mathematical. However, association rule mining is suitable for non-numeric, categorical data and requires just a little bit more than simple counting.

Association rule mining is a procedure which aims to observe frequently occurring patterns, correlations, or associations from datasets found in various kinds of databases such as relational databases, transactional databases, and other forms of repositories.

An association rule has two parts:

- an antecedent (if) and
- a consequent (then).

An antecedent is something that's found in data, and a consequent is an item that is found in combination with the antecedent. Have a look at this rule for instance:

"If a customer buys bread, he's 70% likely of buying milk."

In the above association rule, bread is the antecedent and milk is the consequent. Simply put, it can be understood as a retail store's association rule to target their customers better. If the above rule is a result of thorough analysis of some data sets, it can be used to not only improve customer service but also improve the company's revenue.

Association rules are created by thoroughly analyzing data and looking for frequent if/then patterns. Then, depending on the following two parameters, the important relationships are observed:

1. **Support**: Support indicates how frequently the if/then relationship appears in the database.

2. **Confidence**: Confidence tells about the number of times these relationships have been found to be true.

So, in a given transaction with multiple items, Association Rule Mining primarily tries to find the rules that govern how or why such products/items are often bought together. For example, peanut butter and jelly are frequently purchased together because a lot of people like to make PB&J sandwiches.

Association Rule Mining is sometimes referred to as "Market Basket Analysis", as it was the first application area of association mining. The aim is to discover associations of items occurring together more often than you'd expect from randomly sampling all the possibilities. The classic anecdote of Beer and Diaper will help in understanding this better.

Classification and Regression techniques

Linear and Logistic regressions are usually the first algorithms people learn in predictive modeling. Due to their popularity, a lot of analysts even end up thinking that they are the only form of regressions. The ones who are slightly more involved think that they are the most important amongst all forms of regression analysis.

The truth is that there are innumerable forms of regressions, which can be performed. Each form has its own importance and a specific condition where they are best suited to apply. In this article, I have explained the most commonly used 7 forms of regressions in a simple manner. Through this article, I also hope that people develop an idea of the breadth of regressions, instead of just applying linear / logistic regression to every problem they come across and hoping that they would just fit!

Regression analysis is a form of predictive modelling technique which investigates the relationship between a **dependent** (target) and **independent variable (s)** (predictor). This technique is used for forecasting, time series modelling and finding the causal effect relationship between the variables. For example, relationship between rash driving and number of road accidents by a driver is best studied through regression.

Regression analysis is an important tool for modelling and analyzing data. Here, we fit a curve / line to the data points, in such a manner that the differences between the distances of data points from the curve or line is minimized. I'll explain this in more details in coming sections.

Why do we use Regression Analysis?

As mentioned above, regression analysis estimates the relationship between two or more variables. Let's understand this with an easy example:Let's say, you want to estimate growth in sales of a company based on current economic conditions. You have the recent company data which indicates that the growth in sales is around two and a half times the growth in the economy. Using this insight, we can predict future sales of the company based on current & past information.

There are multiple benefits of using regression analysis. They are as follows:

1. It indicates the **significant relationships** between dependent variable and independent variable.
2. It indicates the **strength of impact** of multiple independent variables on a dependent variable.

Regression analysis also allows us to compare the effects of variables measured on different scales, such as the effect of price changes and the number of promotional activities. These benefits help market researchers / data analysts / data scientists to eliminate and evaluate the best set of variables to be used for building predictive models.

1. Linear Regression

It is one of the most widely known modeling technique. Linear regression is usually among the first few topics which people pick while learning predictive modeling. In this technique, the dependent variable is continuous, independent variable(s) can be continuous or discrete, and nature of regression line is linear.

Linear Regression establishes a relationship between **dependent variable (Y)** and one or more **independent variables (X)** using a **best fit straight line** (also known as regression line).

It is represented by an equation **Y=a+b*X + e**, where a is intercept, b is slope of the line and e is error term. This equation can be used to predict the value of target variable based on given predictor variable(s).

2. Logistic Regression

Logistic regression is used to find the probability of event=Success and event=Failure. We should use logistic regression when the dependent variable is binary (0/ 1, True/

False, Yes/ No) in nature. Here the value of Y ranges from 0 to 1 and it can represented by following equation.

3.Stepwise Regression

This form of regression is used when we deal with multiple independent variables. In this technique, the selection of independent variables is done with the help of an automatic process, which involves *no* human intervention.

CLUSTERING

The process of grouping a set of physical or abstract objects into classes of *similar* objects is called clustering. A cluster is a collection of data objects that are *similar* to one another within the same cluster and are *dissimilar* to the objects in other clusters. A cluster of data objects can be treated collectively as one group and so may be considered as a formof data compression. Although classification is an effective means for distinguishing groups or classes of objects, it requires the often costly collection and labeling of a large set of training tuples or patterns, which the classifier uses to model each group. Clustering is also called data segmentation in some applications because clustering partitions large data sets into groups according to their *similarity*. Clustering can also be used for outlier detection, where outliers (values that are "far away" from any cluster) may be more interesting than common cases.

Clustering is a challenging field of research in which its potential applications pose their own special requirements. The following are typical requirements of clustering in data mining:

- **Scalability:** Many clustering algorithms work well on small data sets containing fewer than several hundred data objects; however, a large database may contain millions of objects. Clustering on a *sample* of a given large data set may lead to biased results. Highly scalable clustering algorithms are needed.

- **Ability to deal with different types of attributes:** Many algorithms are designed to cluster interval-based (numerical) data. However, applications may require clustering other types of data, such as binary, categorical (nominal), and ordinal data, or mixtures of these data types.

- **Discovery of clusters with arbitrary shape:** Many clustering algorithms determine clusters based on Euclidean or Manhattan distance measures. Algorithms based on such distance measures tend to find spherical clusters with

similar size and density. However, a cluster could be of any shape. It is important to develop algorithms that can detect clusters of arbitrary shape.

- **Minimal requirements for domain knowledge to determine input parameters:** Many clustering algorithms require users to input certain parameters in cluster analysis (such as the number of desired clusters). The clustering results can be quite sensitive to input parameters. Parameters are often difficult to determine, especially for data sets containing high-dimensional objects. This not only burdens users, but it also makes the quality of clustering difficult to control.

- **Ability to dealwith noisy data:** Most real-world databases contain outliers or missing, unknown, or erroneous data. Some clustering algorithms are sensitive to such data and may lead to clusters of poor quality.

- **Incremental clustering and insensitivity to the order of input records:** Some clustering algorithms cannot incorporate newly inserted data (i.e., database updates) into existing clustering structures and, instead, must determine a new clustering from scratch. Some clustering algorithms are sensitive to the order of input data.

- **High dimensionality:** A database or a data warehouse can contain several dimensions or attributes. Many clustering algorithms are good at handling low-dimensional data, involving only two to three dimensions. Human eyes are good at judging the quality of clustering for up to three dimensions. Finding clusters of data objects in high dimensional space is challenging, especially considering that such data can be sparse and highly skewed.

Mining methodology and user interaction issues:

These reflect the kinds of knowledge mined, the ability to mine knowledge at multiple granularities, the use of domain knowledge, ad hoc mining, and knowledge visualization.

- Mining different kinds of knowledge databases: Data mining should cover a wide spectrum of data analysis and knowledge discovery tasks, including data characterization, discrimination, association, classification, clustering, tread and deviation analysis, and similarity analysis.
- Interactive mining of knowledge at multiple levels of abstraction: The data mining process should be interactive. Interactive mining allows users to focus the search for patterns, providing and refining data mining requests based on returned results.

- Incorporation of background knowledge: Background knowledge may be used to guide the discovery process and allow discovered patterns to be expressed in concise terms and at different levels of abstraction.
- Data mining query languages and ad hoc mining: Relational query languages (such as SQL) allow users to pose ad hoc queries for data retrieval.
- Presentation and visualization of data mining results: Discovered knowledge should be expressed in high-level languages, visual representations, or other expressive forms so that knowledge can be easily understood and directly usable by humans.
- Handling noisy or incomplete data: When mining data regularities, these objects may confuse the process, causing the knowledge model constructed to overfit the data.
- Pattern evaluation--the interestingness problem: A data mining system can uncover thousands of patterns. Many of the patterns discovered may be uninteresting to the given user, representing common knowledge or lacking novelty.

Performance issues: These include efficiency, scalability, and parallelization of data mining algorithms.

- Efficiency and scalability of data mining algorithms: To effectively extract information from a huge amount of data in databases, data mining algorithms must be efficient and scalable.
- Parallel, distributed, and incremental mining algorithms: The huge size of many databases, the wide distribution of data, and the computational complexity of some data mining methods are factors motivating the development of algorithms that divide data into partitions that can be processed in parallel.

Issues relating to the diversity of database types:

- Handling of relational and complex types of data: Specific data mining systems should be constructed for mining specific kinds of data.

- Mining information from heterogeneous databases and global information systems: Local- and wide-area computer networks (such as the Internet) connect many sources of data, forming huge, distributed, and heterogeneous databases.

The above issues are considered major requirements and challenges for the further evolution of data mining technology. Some of the challenges have been addressed in recent data mining research and development, to a certain extent, and are now considered requirements, while others are still in the research stage.

Measures of Interestingness

Measuring the interestingness of discovered patterns is an active and important area of data mining research. Although much work has been conducted in this area, so far there is no widespread agreement on a formal definition of interestingness in this context. Based on the diversity of definitions presented to-date, interestingness is perhaps best treated as a broad concept that emphasizes conciseness, coverage, reliability, peculiarity, diversity, novelty, surprisingness, utility, and actionability. These nine specific criteria are used to determine whether or not a pattern is interesting. They are described as follows. Conciseness.

A pattern is concise if it contains relatively few attribute-value pairs, while a set of patterns is concise if it contains relatively few patterns. A concise pattern or set of patterns is relatively easy to understand and remember and thus is added more easily to the user's knowledge (set of beliefs). Accordingly, much research has been conducted to find a "minimum set of patterns," using properties such as monotonicity [Padmanabhan and Tuzhilin 2000] and confidence invariance

Generality/Coverage. A pattern is general if it covers a relatively large subset of a dataset. Generality (or coverage) measures the comprehensiveness of a pattern, that is, the fraction of all records in the dataset that matches the pattern. If a pattern characterizes more information in the dataset, it tends to be more interesting

Frequent itemsets are the most studied general patterns in the data mining literature. An itemset is a set of items, such as some items from a grocery basket. An itemset is frequent if its support, the fraction of records in the dataset containing the itemset, is above a given threshold

PART-B 2 MARK QUESTIONS

1.What is data regression?
2.What is numerosity reduction?
3.What are the benefits of correlation analysis in Data Mining?
4.Define Constraint based Cluster Analysis.
5.What do you mean by visual data mining?

PART-C 8 MARK QUESTIONS

1.Discuss the different types of data in cluster analysis.
2.Explain Regression techniques.
3.Explain association rule mining.
4.Discuss the outlier analysis.
5.Explain Spatial Data mining.
6.Discuss the Mining methodology and user interaction issues.
7.Explain the trends in Data mining.
8.Discuss the Partitioning methods of cluster analysis in Data Mining.
9.Explain classification techniques of Data Mining.
10.Discuss the interestingness of Data Mining.

Questions	opt1	opt2	opt3	opt4	opt5	opt6	Answer
_____ takes data from source systems and makes it available to the data warehouse	data cleaning	data abstraction	data extraction	data integration			data extraction
clean and _____ the loaded data into a structure that speeds up queries	transform	extract	partition	aggregate			transform
A _____ is a collection of data objects that are similar to one another	cluster	outlier	prediction	hashing			cluster
The _____ process is the system process that manages the queries and speeds them up by directing queries to the most effective data source	process management	load management	system management	query management			query management
The _____ is the system component that performs all the operations necessary to support the extract and load process	process manager	load manager	system manager	query manager			load manager
The bulk of the effort to develop a load manager should be planned within the fir _____ phase	analysis	design	load	production			production
Meta data describes the _____.	type and format of data	structure of the contents of database	location of data	owner of data			structure of the contents of database
In bottom up approach _____ are used by end-users	data mart	large data warehouse	central database	operational database			data mart
_____ are used to load the information from the operational database	Statistical Techniques	Visualization Techniques	Windowing mechanism	Replication Techniques			Replication Techniques
_____ responses end-users queries in a very short space of time.	Client / Server Technique	Time sharing system	Multiprocessing computer system	Real time system			Multiprocessing computer system
Expert system contain _____.	spatial data	knowledge of specialists	knowledge of business logic	transaction records			knowledge of specialists

OLAP store their data in _____.	table format	object oriented format	a special multi-dimensional format	points in multi-dimensional space			a special multi-dimensional format
OLAP tools	do not learn but create new knowledge	do learn but cannot create new knowledge	do learn and also can create new knowledge	do not learn and cannot create new knowledge			do not learn and cannot create new knowledge
Data mining algorithm should not have a complexity that is higher than____.	logn	n^2	nlogn	n			nlogn
Association rules are defined on _____.	single attribute	n attributes	binary attributes	data attributes			binary attributes
A perceptrons consists of _____.	two layered networks	single layered networks	three layered networks	multiple layered networks			three layered networks
Back propagation method gives _____.	answers and idea as to how they arrived at the answers	answers and no clear idea as to how they arrived at the answers	only ideas as to how to obtain answers	answers and poor ideas as to how they arrived at the answers			answers and no clear idea as to how they arrived at the answers
A Kohonen's self-organizing map is a collection of _____.	networks	neurons	processors	monitors			neurons
Genetic algorithms can be viewed as a kind of _____ strategy.	self learning	meta learning	knowledge discovering	concept learning			meta learning
Voroni diagram divide a sample space into _____.	different groups	different divisions	different sub networks	different regions			different regions
Neural networks are somewhat better at _____ tasks.	problem solving tasks	classification	knowledge engineering	simple			classification

SQL retrieves _____.	hidden knowledge	hidden rules	shallow knowledge	deep knowledge			shallow knowledge
_____ can be found by pattern recognition algorithms	Hidden knowledge	Deep knowledge	Encrypted information	Fine grained segmentation			Hidden knowledge
_____ give a yes or no answer and no explanation of their responses	Genetic algorithm	Neural network	k-nearest neighbor algorithm	(b) and (c)			(b) and (c)
The reporting stage combines _____.	analysis of the results & application of the result to new data	the results & application of the result to new data	analysis of the results & application of the result to existing data	a. analysis of the results & application of the result to new rules			analysis of the results & application of the result to new data
The delivery process is staged in order to _____.	reduce the execution time	to minimize error	minimize risk	measure benefits			minimize risk
Delivery process is designed to deliver _____.	an enterprise data warehouse	a point solution	maximum information	quality solution			an enterprise data warehouse
Delivery process ensures _____.	to reduce the overall delivery time-slice	benefits are delivered incrementally	to reduce the overall delivery time-slice & benefits are delivered incrementally	to reduce investment			to reduce the overall delivery time-slice & benefits are delivered incrementally
The technical blueprint phase must deliver an overall architecture that satisfies the _____ requirements.	short term	long term	today's	mid term			long term

----------_____ is part of day-to-day management of the data warehouse.	Creating / deleting summaries	Extracting data	Cleaning data	Populating data			Creating / deleting summaries
The data extracted from the source systems is loaded into a _____.	data warehouse	data mart	temporary data store	operational database			temporary data store
The purpose of business case is to identify the projected _____.	out put structure	business process	business benefits	business process risks			business benefits
_____ is a typical example of grid based method	Partitioning	STING	Density based	Model –based			STING
SOM stands for	Self-organizing feature map	Simple organizing map	Self oriented map	Summer Opt Machine			Self-organizing feature map
_____ is a clustering approach that performs clustering by incorporation of user specified or application oriented constrained	Partitioning	Hierarchical	Density based	Constrained based clustering			Constrained based clustering
The cost complexity pruning algorithm used in _____	CART	ID3	splitting rule	printer			CART
In linear regression, the n input variables are called _____	generators	response	prediction	predictors			predictors
In linear regression, the one output variables are called _____	generators	response	prediction	predictors			response
_____ is the problem of determining how much alike the two variables actually are	regression	generators	correlation	predictors			correlation
A _____ rule consists of if part and then part	classification	associatic	bayes	incremental			classification
_____ is the count of number of transactions that contain the items in X.	O(X)	O(n²)	O(kn)	O(n)			O(X)
In _____ clustering, all items are initially placed in one cluster.	Divisive	Partional	Agglomerative	Hierarchical			Divisive
Partitional clustering is otherwise called as _____ .	Hierarchical	Non-Hierarchical	Divisive	Agglomerative			Non-Hierarchical
In _____ clustering only one set of clusters may be created internally within the various algorithms.	Hierarchical	Partional	Divisive	Agglomerative			Partional

The common measure is _____ which measures the squared distance from each point to the centroid for the associated cluster.	Squared error metric	Non Squared error Metric	metroid	squared			Non Squared error Metric
The _____ error clustering algorithm minimizes the squared error.	Squared	Non Squared	metroid	divisive			Squared
The squared error for a cluster is the _____ of the squared Euclidean distances between each element in the cluster.	Difference	Multiple	Sum	mean			Sum
_____ is an iterative algorithm in which items are moved among sets of clusters until the desired set is reached.	Squared error metric	Partitional	Hierarchical	K- Means			K- Means
Back propagation method gives _____.	answers and idea as to how they arrived at the answers	answers and no clear idea as to how they arrived at the answers	only ideas as to how to obtain answers	answers and poor ideas as to how they arrived at the answers			answers and no clear idea as to how they arrived at the answers
A Kohonen's self-organizing map is a collection of _____.	networks	neurons	processors	monitors			neurons
Genetic algorithms can be viewed as a kind of _____ strategy.	self learning	meta learning	knowledge discovering	concept learning			meta learning
Voroni diagram divide a sample space into _____.	different groups	different divisions	different sub networks	different regions			different regions
Neural networks are somewhat better at _____ tasks.	problem solving tasks	classification	knowledge engineering	analysis			classification
SQL retrieves _____.	hidden knowledge	hidden rules	shallow knowledge	deep knowledge			shallow knowledge
_____ can be found by pattern recognition algorithms	Hidden knowledge	Deep knowledge	Encrypted information	Fine grained segmentation			Hidden knowledge

_____ give a yes or no answer and no explanation of their responses	Genetic algorithm	Neural network	k-farest neighbor algorithm	Mobile Network			Neural Network
The reporting stage combines _____.	analysis of the results & application of the result to new data	the results & application of the result to new data	analysis of the results & application of the result to existing data	a. analysis of the results & application of the result to new rules			analysis of the results & application of the result to new data
The delivery process is staged in order to _____.	reduce the execution time	to minimize error	minimize risk	measure benefits			minimize risk
Delivery process ensures _____.	to reduce the overall delivery time-slice	benefits are delivered incrementally	to reduce the overall delivery time-slice & benefits are delivered incrementally	to reduce investment			to reduce the overall delivery time-slice & benefits are delivered incrementally

Reg. No.....

[11CAU601]

KARPAGAM UNIVERSITY

(Under Section 3 of UGC Act 1956)
COIMBATORE – 641 021
(For the candidates admitted from 2011 onwards)

BCA DEGREE EXAMINATION, APRIL 2014

Sixth Semester

COMPUTER APPLICATIONS

DATA MINING AND WAREHOUSING

Time: 3 hours Maximum : 100 marks

PART – A (15 x 2 = 30 Marks)
Answer ALL the Questions

1. What is data Mining?
2. Define Date Stream.
3. What are kinds of data warehouse application?
4. What is mean by aggregation?
5. Write the uses of sampling.
6. Define binning.
7. What are the uses of hash-based technique?
8. List out the strategies of constraint based mining.
9. Define confidences.
10. What is classification?
11. Define Pruning.
12. List out the uses of CBR.
13. List out the types of hierarchical clustering method.
14. What is the expansion of DBSCAN?
15. Define spatial Data Mining.

PART B (5 X 14= 70 Marks)
Answer ALL the Questions

6. a. Explain various types of classification of data mining systems.

Or

b. Describe the data warehouse architecture.

17. a. Explain about various forms of data preprocessing.

Or

b. Explain the following :
i. data aggregation ii. attribute subset selection

18. a. How to find the frequent item sets using candidate generation for Apriori algorithm? Explain

Or

b. Explain the process of constraint based association mining.

19. a. Explain about naive Bayesian classification.

Or

b. Discuss about rule based classification.

20. a. List and explain the categorization of major cluster method.

Or

b. Discuss about model based cluster method.

Reg. No.....

[11CAU601]

KARPAGAM UNIVERSITY

(Under Section 3 of UGC Act 1956)
COIMBATORE – 641 021
(For the candidates admitted from 2011 onwards)

BCA DEGREE EXAMINATION, NOVEMBER 2014

Sixth Semester

COMPUTER APPLICATIONS

DATA MINING AND WAREHOUSING

Time: 3 hours Maximum : 100 marks

PART – A (15 x 2 = 30 Marks)
Answer ALL the Questions

1. Define Data Warehouse.
2. What is OLTP?
3. What is a multidimensional Data Model?
4. What is the necessity of Data Reduction?
5. Define Data Transformation.
6. What is a Schema?
7. What is a pattern analysis?
8. Define Sampling.
9. What is associative classification?
10. What is a Supervised Learning?
11. List the advantages and disadvantages of Naïve Bayesian classification.
12. What is rule based classification?
13. Define Density-based approach.
14. What is a statics approach in clustering?
15. Define unsupervised learning.

PART B (5 X 14= 70 Marks)
Answer ALL the Questions

16. a) Elaborately explain the KDD.
Or
b) Explain Data warehouse implementation in detail.

17. a) Explain online data storage in detail.
Or
b) Elaborately explain data preprocessing.

18. a) Discuss the following of mining frequent patterns and associations,
i. Market Basket Analysis
ii. Frequent Itemsets, Closed Itemsets, and Association Rules
Or
b) Explain the Mining of Multilevel Association Rules.

19. a) Explain the classification by Back propagation.
Or
b) Explain Bayesians theorem in detail.

20. a) Explain the following methods for Outlier detection
i. Statistical Distribution-Based Outlier Detection
ii. Distance-Based Outlier Detection
Or
b) Elaborately explain Multimedia Data Mining.

Reg. No.....

[12CAU601]

KARPAGAM UNIVERSITY

(Under Section 3 of UGC Act 1956)
COIMBATORE – 641 021
(For the candidates admitted from 2012 onwards)

BCA DEGREE EXAMINATION, APRIL 2015

Sixth Semester

COMPUTER APPLICATIONS

DATA MINING & WAREHOUSING

Time: 3 hours Maximum : 100 marks

PART – A (15 x 2 = 30 Marks)
Answer ALL the Questions

1. What is data mining?
2. List out any four data mining functionalities.
3. Define cluster analysis.
4. What is data preprocessing?
5. Define the term 'Linear regression'.
6. What is mean by dimensionality reduction?
7. Define association rule mining?
8. List any two techniques to improve the efficiency of apriori algorithm.
9. What is market basket analysis?
10. What is prediction?
11. Define back propagation.
12. What is decision tree?
13. Define categorical variable.
14. What is mean by text mining?
15. Define constraint based clustering.

PART B (5 X 14= 70 Marks)
Answer ALL the Questions

16. a. Discuss about major issues in data mining.

Or

b. Explain about data warehouse implementation.

17. a. Explain various types of preprocessing techniques.

Or

b. Discuss about data cleaning in detail.

18. a. Describe about mining various kinds of association rules.

Or

b. How to find the frequent item sets without candidate generation? Explain.

19. a. Explain about rule based classification.

Or

b. Discuss about accuracy and error measures.

20. **Compulsory :-**

Explain various types of application in data mining.
