

GUDLAVALLERU ENGINEERING COLLEGE
(An Autonomous Institute with Permanent Affiliation to JNTUK, Kakinada)
Seshadri Rao Knowledge Village, Gudlavalleru – 521 356.

Department of Computer Science and Engineering



HANDOUT

on

DATA WAREHOUSING AND MINING

Vision

To be a Centre of Excellence in computer science and engineering education and training to meet the challenging needs of the industry and society.

Mission

- To impart quality education through well-designed curriculum in tune with the growing software needs of the industry.
- To be a Centre of Excellence in computer science and engineering education and training to meet the challenging needs of the industry and society.
- To serve our students by inculcating in them problem solving, leadership, teamwork skills and the value of commitment to quality, ethical behavior & respect for others.
- To foster industry-academia relationship for mutual benefit and growth

Program Educational Objectives

PEO1: Identify, analyze, formulate and solve Computer Science and Engineering problems both independently and in a team environment by using the appropriate modern tools.

PEO2: Manage software projects with significant technical, legal, ethical, social, environmental and economic considerations

PEO3: Demonstrate commitment and progress in lifelong learning, professional development, leadership and Communicate effectively with professional clients and the public.

HANDOUT ON DATA WAREHOUSING AND MINING

Class & Sem. : III B.Tech – II Semester

Year : 2019-20

Branch : CSE

Credits : 3

1. Brief History and Scope of the Subject

The term “Data Mining” was only introduced in the 1990s. Data mining is part of the knowledge discovery process that offers a new way to look at data. Data mining consists of the nontrivial extraction of implicit, previously unknown, and potentially useful information from data. It uses machine learning, statistical and visualization techniques to discover and present knowledge in a form that is easily comprehensible to humans. Data mining is then the process of discovering meaningful new correlations, patterns and trends by sifting through vast amounts of data using statistical and mathematical techniques.

As Fortune 500 organizations continue to amass substantial quantities of information into their respective databases, data mining can offer the opportunity to learn from this data. Furthermore, current trends indicate that more companies implementing Enterprise Resource Planning systems or contracting with ASP vendors could further benefit in using data mining techniques. Integrating a data mining technique alongside these two added value services can prove to be an optimum solution in understanding a company’s data.

2. Pre-Requisites

- Database Management Systems, Basics of Probability and Statistics

3. Course Objectives:

- To introduce the concepts of Data warehousing and Data mining.
- To familiarize with the concepts of association rule mining, classification, clustering techniques and algorithms.

4. Course Outcomes:

- CO1: Outline different types of databases used in data mining
- CO2: Apply pre-processing methods on raw data to make it ready for mining.
- CO3: Illustrate the major concepts and operations of multi dimensional data models.
- CO4: Analyze the performance of association rules mining algorithms for finding frequent item sets from the large databases
- CO5: Simplify the data classification procedure by selecting appropriate classification methods / algorithms
- CO6: Classify various clustering methods and algorithms on data sets to create appropriate clusters.

5. Program Outcomes:

Computer Science and Engineering Graduates will be able to:

1. **Engineering knowledge:** Apply the knowledge of mathematics, science, engineering fundamentals, and an engineering specialization to the solution of complex engineering problems.
2. **Problem analysis:** Identify, formulate, review research literature, and analyze complex engineering problems reaching substantiated conclusions using first principles of mathematics, natural sciences, and engineering sciences.
3. **Design/development of solutions:** Design solutions for complex engineering problems and design system components or processes that meet the specified needs with appropriate consideration for the public health and safety, and the cultural, societal, and environmental considerations.
4. **Conduct investigations of complex problems:** Use research-based knowledge and research methods including design of experiments, analysis and interpretation of data, and synthesis of the information to provide valid conclusions.
5. **Modern tool usage:** Create, select, and apply appropriate techniques, resources, and modern engineering and IT tools including

- prediction and modelling to complex engineering activities with an understanding of the limitations.
6. **The engineer and society:** Apply reasoning informed by the contextual knowledge to assess societal, health, safety, legal and cultural issues and the consequent responsibilities relevant to the professional engineering practice.
 7. **Environment and sustainability:** Understand the impact of the professional engineering solutions in societal and environmental contexts, and demonstrate the knowledge of, and need for sustainable development.
 8. **Ethics:** Apply ethical principles and commit to professional ethics and responsibilities and norms of the engineering practice.
 9. **Individual and team work:** Function effectively as an individual, and as a member or leader in diverse teams, and in multidisciplinary settings.
 10. **Communication:** Communicate effectively on complex engineering activities with the engineering community and with society at large, such as, being able to comprehend and write effective reports and design documentation, make effective presentations, and give and receive clear instructions.
 11. **Project management and finance:** Demonstrate knowledge and understanding of the engineering and management principles and apply these to one's own work, as a member and leader in a team, to manage projects and in multidisciplinary environments.
 12. **Life-long learning:** Recognize the need for, and have the preparation and ability to engage in independent and life-long learning in the broadest context of technological change.

6. Mapping of Course Outcomes with Program Outcomes:

	1	2	3	4	5	6	7	8	9	10	11	12
CO1	3	2	3	2								
CO2	3	2	2		3	2						
CO3			3	2								
CO4	2	2	2		2	3						
CO5		2	2		3							
CO6	2	2	2	3	3							

3- High Level Mapping 2- Medium Level Mapping 1-Low Level Mapping

7. Prescribed Text Books

1. Jiawei Han & Micheline Kamber, & Jian pei, "Data Mining Concepts and Techniques", 3rd edition, Morgan Kaufmann Publisher an imprint of Elsevier.

8. Reference Text Books

- a. Pang-Ning Tan, Michael Steinbach, Vpin Kumar "Introduction to Data Mining", 1st edition, Pearson.
- b. Margaret H Dunham, "Data Mining Introductory and Advanced Topics", 1st edition, Pearson Education

9. URLs and Other E-Learning Resources

- a. <http://www.cs.sfu.ca/~han/dmbook>
- b. <http://db.cs.sfu.ca/>
- c. <http://www.cs.sfu.ca/~han>

10. Digital Learning Materials:

- <http://192.168.0.49/videos/videosListing/270#>

11. Lecture Schedule / Lesson Plan

Topic	No. of Periods
UNIT - I: INTRODUCTION	
Motivation and importance of data mining	2
Types of data to be mined: Relational database, datawarehouses, transactional databases, advanced database systems	4
Data Mining Functionalities	2
	8
UNIT - II: DATA PRE-PROCESSING	
Major tasks in data pre-processing	1

Data cleaning: Missing values, Noisy Data	2
Data reduction: Overview of data reduction strategies, Principal components analysis Attribute subset selection, histograms, sampling	4
Data Transformation: Data transformation strategies overview, data transformation by normalization	3
	10
UNIT - III: DATA WAREHOUSING AND ONLINE ANALYTICAL PROCESSING	
Data warehouse: Basic concepts, OLAP vs OLTP	2
Data warehouse: A multi-tired architecture	1
Data warehouse modeling : Data cube and OLAP	2
Data cube: A multidimensional data model, star, snowflake and fact constellation schemas for multidimensional data models	3
The role of concept hierarchies	1
Typical OLAP operations	1
	10
UNIT - IV: MINING FREQUENT PATTERNS, ASSOCIATIONS, AND CORRELATIONS	
Basic concepts, Frequent item sets, closed item sets and association rules	2
Frequent item set mining methods: Apriori Algorithm, generations, association rules from frequent item sets	3
A Pattern-Growth approach for mining frequent item sets	2
	7
UNIT - V: CLASSIFICATION	
Basic concepts, What is classification, general approach to classification	2
Decision Tree Induction	2
Attribute selection measures : Information gain	3
Bayes classification methods: Bayes' theorem	2

Naïve Bayesian classification	2
	11
UNIT - VI: CLUSTER ANALYSIS	
Introduction, Overview of basic clustering methods	2
Partitioning methods: k-means, k-medoids	3
Hierarchical methods: Agglomerative versus divisive hierarchical clustering	3
Density based method: DBSCAN	2
	10
Total No.of Periods:	56

12. Seminar Topics:

In order to enhance the understanding capability and to prepare the student to face the interviews and audience, to enhance the communication skills and to eliminate stage fear, seminars and group discussions are conducted.

- Data Warehouse and OLAP
- Concept Hierarchy Generation
- Bayesian Classification
- Density-Based Methods

UNIT – I

Objective:

- To gain knowledge on Data mining

Syllabus:

Unit-I:

Motivation and importance of data mining, types of data to be mined: Relational databases, data warehouses, transactional databases, advanced database systems, data mining functionalities.

Learning Outcomes:

At the end of the unit, students will be able to:

1. Understand functionalities of Data Mining
2. Identify and study different databases to implement data mining systems.

Learning Material

Introduction

1.1 Motivation and importance of data mining

- Motivation and Importance of Data Mining in the 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.

Definition : Data mining refers to extracting or “mining” knowledge from large amounts of data.

(or)

It is the process of automatically discovering useful information in large data repositories.

- Data mining should have been more appropriately named “knowledge mining from data,” in short it is “Knowledge mining,” many people treat data mining as a synonym for another popularly used term, Knowledge Discovery from Data, or KDD.

Different views of Data mining

- 1) Data mining as simply an essential step in the process of knowledge discovery.

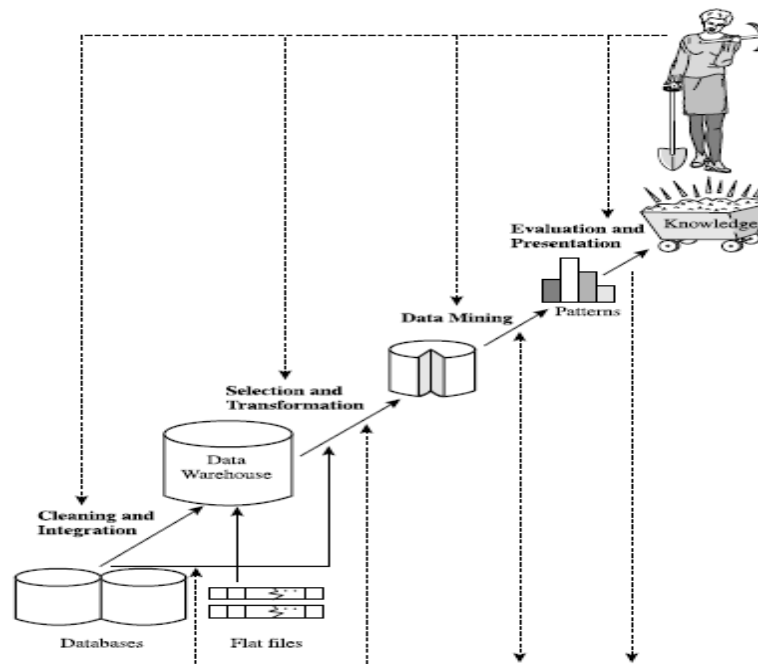


Fig 1: Data mining as a step in the process of knowledge discovery

Above Figure consists of an iterative sequence of the following steps:

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

3. Data selection (where data relevant to the analysis task are retrieved from the database)

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

5. Data mining (an essential process where intelligent methods are applied in order to extract data patterns)

6. Pattern evaluation (to identify the truly interesting patterns representing knowledge based on some interestingness measures)

7. Knowledge presentation (where visualization and knowledge representation techniques are used to present the mined knowledge to the user)

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

according to this view, data mining is only one step in the entire process, and it is essential one because it uncovers hidden patterns for evaluation.

2) Other view of Data mining is the process of discovering interesting knowledge from large amounts of data stored in databases, data warehouses, or other information repositories.

Based on this view, the architecture of a typical data mining system may have the following major components

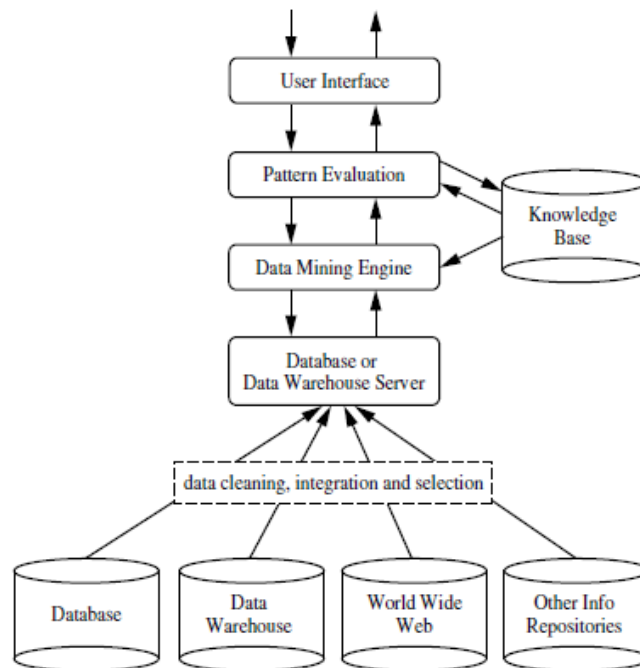


Fig 2 : Architecture of a typical data mining system.

- **Database, data warehouse, WorldWideWeb, or other information repository:** This is one or a set of databases, data warehouses, spreadsheets, or other kinds of information repositories.
- **Data cleaning and data integration** techniques may be performed on the data.
- **Database or data warehouse server:** The database or data warehouse server is responsible for fetching the relevant data, based on the user's data mining request.
- **Knowledge base:** This is the domain knowledge that is used to guide the search or evaluate the interestingness of resulting patterns.
Ex: concept hierarchies, used to organize attributes or attribute values into different levels of abstraction.

- **Data mining engine:** This is essential to the data mining system and ideally consists of a set of functional modules for tasks such as characterization, association and correlation analysis, classification, prediction, cluster analysis, outlier analysis, and evolution analysis.
- **Pattern evaluation module:** This component typically employs interestingness measures. The pattern evaluation module may be integrated with the mining module, depending on the implementation of the data mining method used
- **Graphical User interface:** This module communicates between users and the data mining system.

Allowing the user

- To interact with the system by specifying a data mining query or task.
 - Providing information to help focus the search.
 - Performing exploratory data mining based on the intermediate data mining results.
 - Allows the user to browse database and data warehouse schemas or data structures.
 - Evaluate mined patterns, and visualize the patterns in different forms.
- ❖ Data mining involves an integration of techniques from multiple disciplines such as
- Database technology
 - Statistics
 - Machine learning
 - High-performance computing
 - Pattern recognition
 - Neural networks
 - Data visualization

- Information retrieval
 - Image and signal processing
 - Spatial or Temporal data analysis
- ❖ By performing data mining, interesting knowledge, high-level information can be extracted from databases and viewed from different angles.
 - ❖ The discovered knowledge can be applied to decision making, process control, information management and query processing.
 - ❖ The data mining considered one of the most important frontiers in database systems and one of the most promising interdisciplinary developments in the information industry.

1.2 Data mining should be applicable to any kind of data

1) Relational databases

- A [Relational database](#) is defined as the collection of data organized in tables with rows and columns.
- Physical schema in Relational databases is a schema which defines the structure of tables.
- Logical schema in Relational databases is a schema which defines the relationship among tables.
- Standard API of relational database is [SQL](#).
- **Application:** Data Mining, ROLAP model, etc.

customer

<u>cust_ID</u>	name	address	age	income	credit_info	...
C1	Smith, Sandy	5463 E. Hastings, Burnaby, BC, V5A 4S9, Canada	21	\$27000	1	...
...

STUDENT

ROLL_NO	NAME	ADDRESS	PHONE	AGE
1	RAM	DELHI	9455123451	18
2	RAMESH	GURGAON	9652431543	18
3	SUJIT	ROHTAK	9156253131	20
4	SURESH	DELHI		18

2) DWH

- A data warehouse is defined as the collection of data integrated from multiple sources that will queries and decision making.
- There are three types of data warehouse: **Enterprise** data warehouse, **Data Mart** and **Virtual** Warehouse.
- Two approaches can be used to update data in Data Warehouse: **Query-driven** Approach and **Update-driven** Approach.
- **Application:** Business decision making, Data mining, etc.

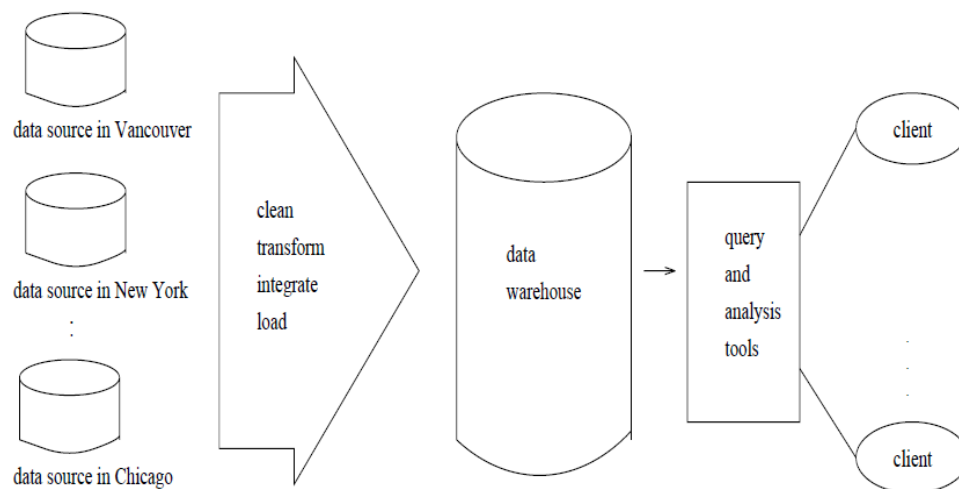


Figure: typical architecture of a data warehouse for AllElectronics.

3) Transactional databases

- Transactional databases is a collection of data organized by time stamps, date, etc to represent transaction in databases.
- This type of database has the capability to roll back or undo its operation when a transaction is not completed or committed.
- Highly flexible system where users can modify information without changing any sensitive information.
- Follows [ACID property](#) of DBMS.
- **Application:** Banking, Distributed systems, Object databases, etc.

sales

<u>trans_ID</u>	list of item_ID's
T100	I1, I3, I8, I16
...	...

Figure: transactional database for sales at AllElectronics

4) Advanced databases

- Object oriented
- Object relational
- Application oriented databases
 - Spatial
 - Temporal
 - Time-Series
 - Text
 - Multimedia databases

Multimedia Databases

- Multimedia databases consists audio, video, images and text media.
- They can be stored on Object-Oriented Databases.
- They are used to store complex information in a pre-specified formats.

- **Application:** Digital libraries, video-on demand, news-on demand, musical database, etc.

Spatial Database

- Store geographical information.
- Stores data in the form of coordinates, topology, lines, polygons, etc.
- **Application:** Maps, Global positioning, etc.

Time-series Databases

- Time series databases contains stock exchange data and user logged activities.
- Handles array of numbers indexed by time, date, etc.
- It requires real-time analysis.
- **Application:** eXtremeDB, Graphite, InfluxDB, etc.

1.3 Data Mining Functionalities—What Kinds of Patterns Can Be Mined?

- Data mining functionalities are used to specify the kind of patterns to be found in data mining tasks.
- Data mining tasks can be classified into two categories:
 - 1) Descriptive
 - Descriptive tasks derive patterns that summarizes the underlying relationship in the data. Ex: correlations, trends, clusters, trajectories and anomalies. These are in explanatory in nature.
 - 2) Predictive
 - Predictive tasks perform inference on the current data to make predictions. i.e predict the value of a particular attribute based on the values of other attributes. ex: classification, regression.

Data mining functionalities, and the kinds of patterns they can discover, are described below:

1. Characterization & Discrimination
2. Association analysis
3. Classification
4. Evolution analysis
5. Clustering
6. Outlier analysis.

1.3.1. Concept/Class Description: Characterization and Discrimination

Data can be associated with **classes or concepts**.

Ex: In the AllElectronics store,

classes of items for sale include computers and printers

concepts of customers include bigSpenders and budgetSpenders.

The summarized descriptions of class or a concept are very much useful. Such descriptions of a class or a concept are called class/concept descriptions. These descriptions can be derived via (1) data characterization 2) data discrimination, (3) both data characterization and discrimination

Data characterization is a summarization of the general characteristics or features of data.

Ex: study the characteristics of software products whose sales increased by 10% in the last year.

- Methods used for this are statistical measures, plots and OLAP operations.
- The output of data characterization can be presented in various forms.

Ex: pie charts, bar charts, curves, multidimensional data cubes, and multidimensional tables,

Data discrimination is comparison of the target class(the class under study) with one or a set of comparative classes (called the contrasting classes).

Ex: the user may like to compare the general features of software products whose sales increased by 10% in the last year with those whose sales decreased by at least 30% during the same period

- Methods used and output presentation is same as characterization although discrimination descriptions should include comparative measures that help distinguish between the target and contrasting classes

1.3.2. Association Analysis

Association analysis is the discovery of association rules showing attribute-value conditions that occur frequently together in a given set of data. This analysis is widely used for market basket or transaction data analysis.

Association rules are of the form $\mathbf{X} \Rightarrow \mathbf{Y}$, is interpreted as “database tuples that satisfy the conditions in X are also likely to satisfy the conditions in Y”.

Ex: Marketing manager of AllElectronics, would like to determine which items are frequently purchased together within the same transactions. An example of such a rule, mined from the AllElectronics transactional database

buys(X, “computer”) => buys(X, “software”) [support = 1%; confidence = 50%]

where X is a variable representing a customer. A confidence of 50% means that if a customer buys a computer, there is a 50% chance that she will buy software as well, and 1% of all of the transactions contain both computer and software were purchased together

1.3.3. Classification and Prediction

Classification is the process of finding a model that describes and distinguishes data classes or concepts. To predict the class of objects whose class label is un known. The derived model is based on the analysis of a set of training data (i.e., data objects whose class label is known).

The derived model may be represented in various forms, such as classification (IF-THEN) rules, decision trees, mathematical formulae, or neural networks

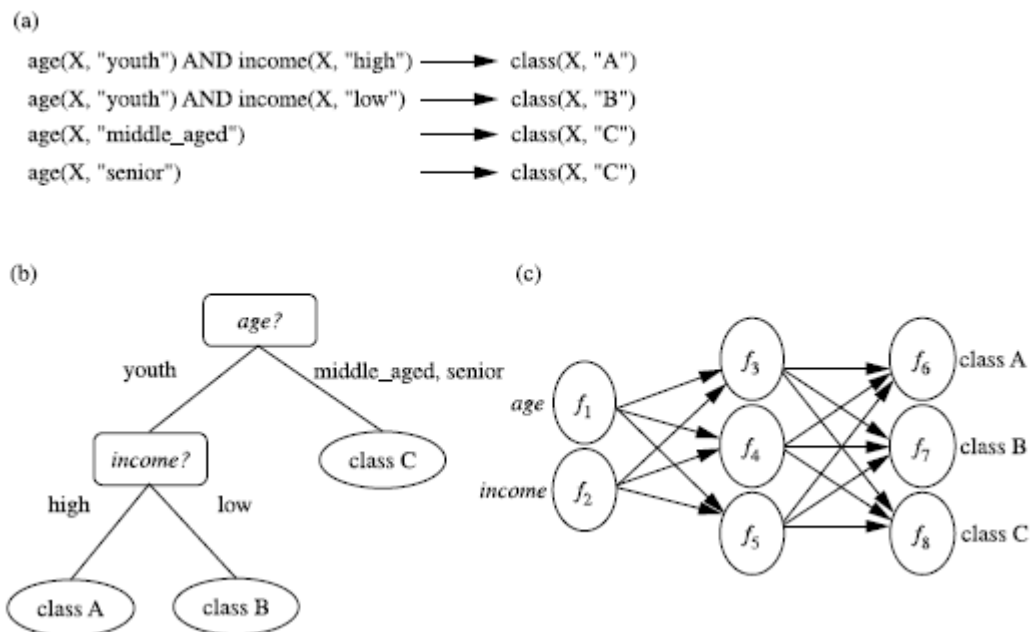


Fig: classification model can be represented in various forms, such as (a) IF-THEN rules,(b) a decision tree, or a (c) neural network.

Ex: AllElectronics, items are classified into 3 classes good response, mild response and no response. based on the descriptive features of the items based on price, brand, place made, type, and category.

Predict missing or unavailable data values are referred as Prediction.

1.3.4.Evolution Analysis

It describes and models regularities or trends for objects whose behavior changes over time, this may include characterization, discrimination, association and correlation analysis, classification, prediction, clustering.

Ex: Stock market data analysis to predict the future trends using previous years data for decision making regarding stock investments.

1.3.5. Cluster Analysis

Cluster is a group of similar data points or objects for analysis. The objects within a cluster have high similarities in comparison to one another but are very dissimilar to objects in other clusters.

Ex: Cluster AllElectronics customer data with respect to customer locations in a city. These clusters may represent individual target groups for marketing.

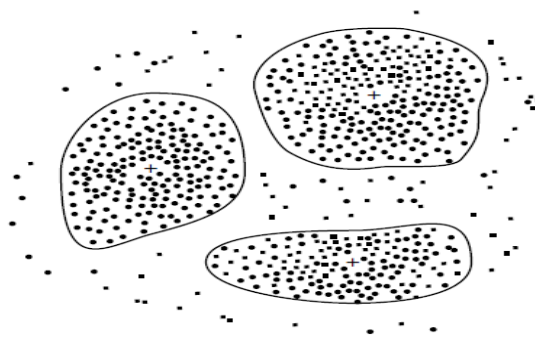


Fig : 2-D plot of customer data with respect to customer locations in a city, showing three data clusters. Each cluster “center” is marked with a “+”.

6. Outlier Analysis

A database may contain data objects that do not comply with the general behavior or model of the data. These data objects are outliers (noise in the data) outliers may be detected using statistical tests

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

UNIT-I
Assignment-Cum-Tutorial Questions
SECTION-A

Objective Questions

1. _____ is the process of discovering interesting patterns and knowledge from large amounts of data.
2. The full form of KDD is_____
3. Goal of data mining includes which of the following []
 - A. To explain some observed event or condition
 - B. To confirm that data exists
 - C. To analyze data from expected relationships
 - D. To create a new data warehouse
4. The Synonym for data mining is []

A Data warehouse B) Knowledge Discovery from Data C) ETL D) OLAP
5. Data mining tasks are classified in to _____ and _____.
6. Match the Following: []

a) Data Cleaning.	i) Multiple data sources may be combined
b) Data Transformation	ii) Remove noise and inconsistent data
c) Data Selection	iii) Data transformed into forms appropriate for mining
d) Data Integration	iv) Relevent data is retrived from database for analysis.

A. i,ii,iii,iv B. i,iii,iv,ii C. ii,iii,iv,i D. iv,ii,iii,i
7. Data mining helps in _____. []

A. inventory management.	C.sales promotion strategies
B. marketing strategies.	D.All of the above
8. Which of the following is not a data mining functionality? []

A. Characterization and Discrimination	C. Classification and regression
B. Selection and interpretation	D. Clustering and Analysis
6. Extreme values that occur infrequently are called as _____. []

A. outliers. B. rare values. C. dimensionality reduction. D. All

7. Grouping of similar objects is known as _____

8. Support and Confidence are used as a measures for Association Rule

Mining. [T/F]

9. _____ is a summarization of the general characteristics or features of a target class of data. []

A. Data Characterization

B. Data Classification

C. Data discrimination

D. Data selection

10 Match the following Issues: []

a) Mining Methodology. i) Efficiency and Scalability

b) User Interaction ii) Handling of relational and complex types of Data

c) Diverse Datatypes iii) Interactive Mining of Knowledge at multiple levels of abstraction

d) Performance iv) Mining different kinds of knowledge in databases.

A. i,ii,iii,iv

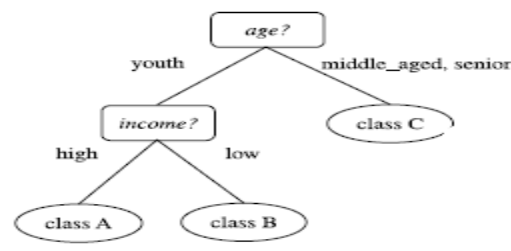
B. i,iii,iv,ii

C. ii,iii,iv,i

D. iv,iii,ii,i

11. _____ is the process of finding a model that describes and distinguishes data classes or concepts.

12. The Following diagram represents _____ Model. []



A. Classification

B. Cluster

C. Evolution

D. Association

13. _____ Analysis can be used for unlabeled dataset.

14. What mining task characterizes properties of the data in a target data set?

A) Predictive B) Descriptive C) Both D) None of the above []

SECTION-B**SUBJECTIVE QUESTIONS**

1. Write briefly about motivation of challenges for data mining.
2. Define Data Mining. Explain the steps to discover knowledge.
3. Write few disciplines where Data mining is applied.
4. Explain various kinds of databases.
5. What are advanced data base systems?
6. Differentiate operational databases and data warehousing.
7. What are Data mining Functionalities? Explain.

UNIT-II

Data Preprocessing

Major tasks in data pre-processing, Data cleaning: Missing values, noisy Data; Data reduction: Overview of data reduction strategies, principal components analysis, attribute subset selection, histograms, sampling; Data transformation: Data transformation strategies overview, data transformation by normalization.

Learning Material

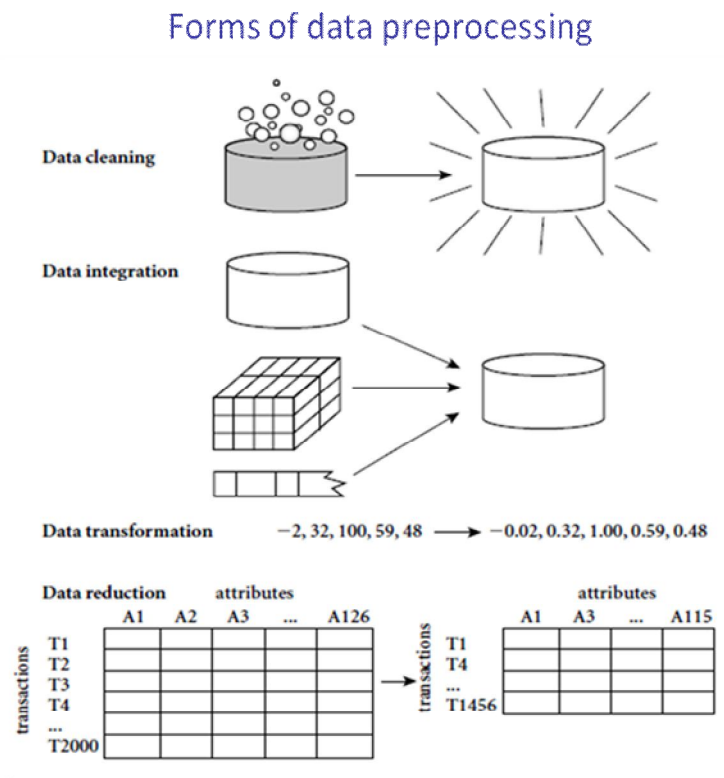
Why Data Preprocessing?

- ❖ Data in the real world is huge size, may contain
 - Incomplete: lacking attribute values, lacking certain attributes of interest, or containing only aggregate data
 - Noisy: containing errors or outliers
 - Inconsistent: containing discrepancies in codes or names
- ❖ No quality data, no quality mining results!
 - Quality decisions must be based on quality data
 - Data warehouse needs consistent integration of quality data
- ❖ Real world data tend to dirty, incomplete and inconsistent. Data preprocessing techniques can improve the quality of the data there by helping to improve the accuracy and efficiency of the subsequent mining process.

- ❖ Detecting anomalies, rectifying them early and reducing the data to be analyzed can lead to huge profit for decision making. So need preprocessing of data.

2.1 Major tasks in Data preprocessing

- **Data cleaning** : to remove noise and inconsistencies in the data.
- **Data Integration** : merge data from multiple sources.
- **Data transformation** : normalization may be applied to improve the accuracy and efficiency of the algorithms
- **Data reduction**: can reduce the data size by aggregating, eliminating redundant features or clustering.



Forms of data preprocessing.

2.2 Data Cleaning

Real world data tend to be incomplete noisy and consistent. Data cleaning routines attempts to

1. Fill in missing values
2. Smooth out noisy data while identifying outliers

2.2.1 Missing values

- Data is not always available
E.g., many tuples have no recorded value for several attributes, such as customer income in sales data
- Missing data may be due to
 - equipment malfunction
 - inconsistent with other recorded data and thus deleted
 - data not entered due to misunderstanding
 - certain data may not be considered important at the time of entry
 - not register history or changes of the data
- Missing data may need to be inferred.

Handling of Missing Data

1. Ignore the tuple: usually done when class label is missing (assuming the tasks in classification)—not effective unless the tuple contains several attributes with missing values

2. Fill in the missing value manually: Time consuming and infeasible for 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 e.g., “unknown”, or $-\infty$.
4. Use the attribute mean to fill in the missing value: Average income of AE customers is \$28,000 use this value to replace the missing value for income.
5. Use the attribute mean for all samples belonging to the same class to fill in the missing value: smarter ex: if classify customers according to credit-risk, replace the missing value with a avg 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, Bayesian formula or decision tree

2.2.2 Noisy Data

- Noise: random error or variance in a measured variable.
- Smooth out the data to remove the noise
- Incorrect attribute values may due to
 1. Faulty data collection instruments
 2. Data entry problems i.e May have been human or computer errors occurring at data entry
 3. Errors in data transmission
 4. Technology limitation ex: Limited buffer size
 5. inconsistency in naming convention

- Other data problems which requires data cleaning
 1. duplicate records
 2. incomplete data
 3. inconsistent data

Handling Noisy Data

A) Binning method: first sort data and partition into (equi-depth) bins then one can smooth by bin means, smooth by bin median, smooth by bin boundaries, etc.

B) Clustering : detect and remove outliers

C) Combined computer and human inspection : detect suspicious values and check by human

D) Regression :smooth by fitting the data into regression functions

A) Simple Discretization Methods: Binning

Binning methods smooth a sorted data value by consulting its neighborhood (value around it). The sorted values are distributed in to a number of buckets or bins.

Binning Methods for Data Smoothing

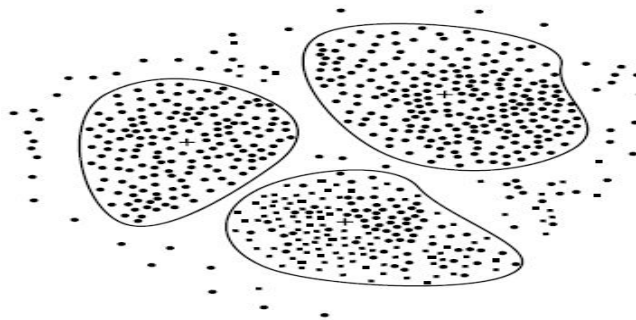
Sorted data for *price* (in dollars): 4, 8, 15, 21, 21, 24, 25, 28, 34

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

Smoothing by bin medians – In which each bin values is replaced by closest boundary values

B) Cluster Analysis

- Outliers may be detected by clustering; where similar values are organized in to groups or clusters.
- Values that fall outside of the set of clusters may be considered outliers



A 2-D plot of customer data with respect to customer locations in a city, showing three data clusters. Each cluster centroid is marked with a "+", representing the average point in space for that cluster. Outliers may be detected as values that fall outside of the sets of clusters.

C) Combined computer and human inspection

- Outliers may be identified through a combination of computer and human inspection

Ex : in one application an information-theoretic measure was used to help identify outlier patterns in a handwritten character database for classification.

- The measure's value reflected the "surprise" content of the predicted character label with respect to the known label.
- Patterns whose surprise content is above a threshold are output to a list.
- A human can then sort through the patterns in the list to identify the actual garbage ones.
- This is much faster than having to manually search through the entire database.
- Outlier patterns may be informative (ex: identifying useful data exceptions, such as different versions of the characters "0" or "7" or garbage)
- The garbage values can then be excluded from use in subsequent data mining.

2.3 Data reduction

- Warehouse may store terabytes of data: Complex data analysis/mining may take a very long time to run on the complete data set

- Data reduction technique can be applied to obtain a reduced representation of the data set that is much smaller in volume, yet closely maintains the integrity of the original data.

2.3.1 Overview of Data reduction strategies

- 1) **Dimensionality reduction** : irrelevant, weakly relevant or redundant attributes or dimensions may be detected and removed.
- 2) **Data compression** : encoding mechanisms are used to reduce the data set size.
- 3) **Numerosity reduction** : the data are replaced or estimated alternative, smaller data representations such as parametric models or nonparametric methods such as clustering, sampling and the use of histograms.

2.3.1.1 Dimensionality reduction

A) Attribute subset selection

- Data sets for analysis may contain hundreds of attributes, many of which may be irrelevant to the mining task, or redundant can slow down the mining process.
- Dimensionality reduction reduces the data set size by removing such attributes(or dimensions) from it. For this methods of attribute subset selection are applied.
- 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.

Heuristic methods of attribute subset selection include the following techniques.

1. step-wise forward selection
2. step-wise backward elimination
3. combining forward selection and backward elimination
4. decision-tree induction

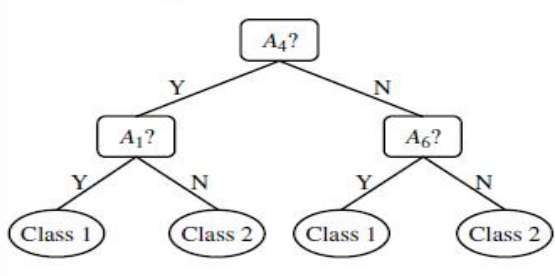
1. Stepwise forward selection: The procedure starts with an empty set of attributes as the reduced set. The best of the original attributes is determined and added to the reduced set. At each subsequent iteration or step, the best of the remaining original attributes is added to the set.

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

3. Combination of forward selection and backward elimination: The stepwise forward selection and backward elimination methods can be combined so that, at each step, the procedure selects the best attribute and removes the worst from among the remaining attributes.

4. Decision tree induction: Decision tree induction constructs a flowchart like structure where each internal (non leaf) node denotes a test on an attribute, each branch corresponds to an outcome of the test, and each external (leaf) node denotes a class prediction. At each node, the algorithm chooses the "best" attribute to partition the data into individual classes.

When decision tree induction is used for attribute subset selection, a tree is constructed from the given data. All attributes that do not appear in the tree are assumed to be irrelevant. The set of attributes appearing in the tree form the reduced subset of attributes.

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

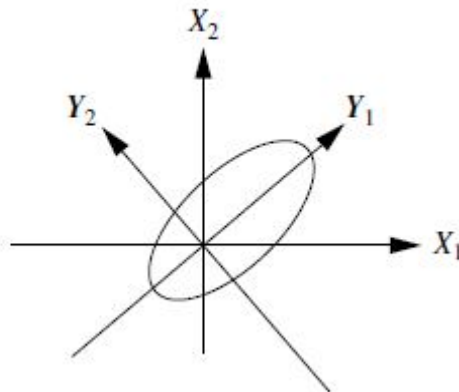
Greedy (heuristic) methods for attribute subset selection.

B) Principal Component Analysis

The principal components analysis is a method of dimensionality reduction.

The basic procedure is as follows:

1. The input data are normalized, so that each attribute falls within the same range. This step helps ensure that attributes with large domains will not dominate attributes with smaller domains.
2. PCA computes k orthonormal vectors that provide a basis for the normalized input data. These are unit vectors that each point in a direction perpendicular to the others. These vectors are referred to as the *principal components*. The input data are a linear combination of the principal components.
3. The principal components are sorted in order of decreasing “significance” or strength. The principal components essentially serve as a new set of axes for the data, providing important information about variance.



That is, the sorted axes are such that the first axis shows the most variance among the data, the second axis shows the next highest variance, and so on. For example, Figure 3.5 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.

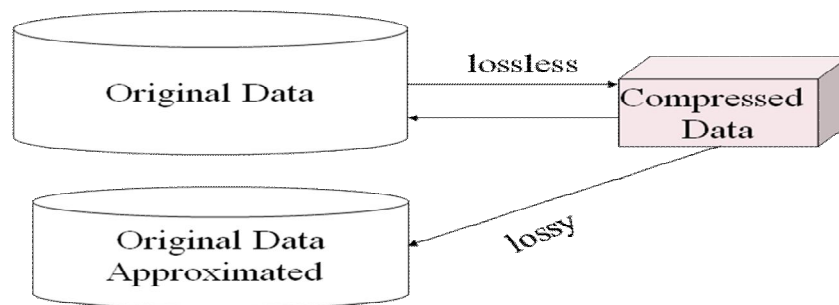
4. Because the components are sorted in decreasing order of “significance,” the data size can be reduced by eliminating the weaker components, that is, those with low variance.

Using the strongest principal components, it should be possible to reconstruct a good approximation of the original data.

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

2.3.1.2 Data Compression

- Data 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.



Two popular and effective methods of lossy data compression are wavelet transforms and principal component analysis

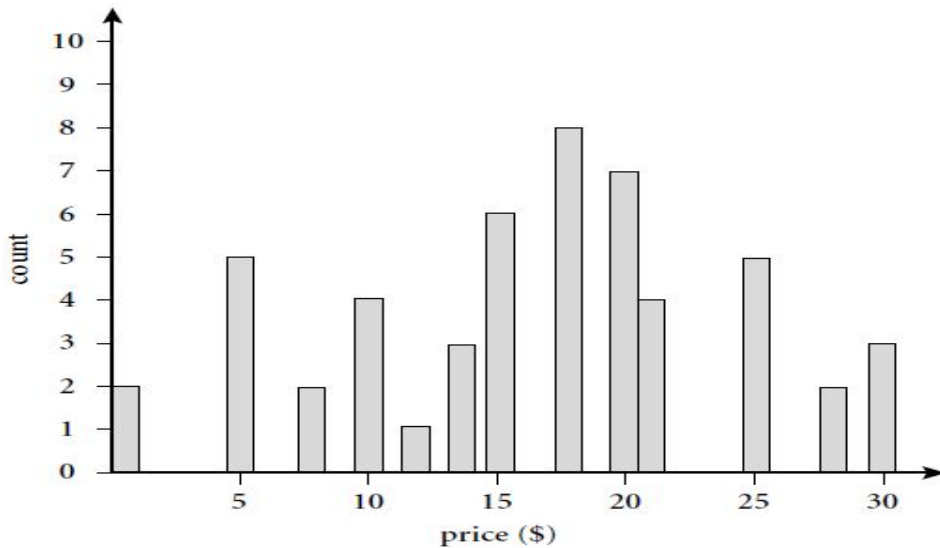
2.3.1.3 Numerosity reduction

- Can reduce the data volume by choosing alternative smaller forms of representations. Techniques may be parametric or non parametric.
- Parametric methods
 - A model is used to estimate the data, so that only the data parameters need to be stored, instead of the actual data (outliers also stored)
 - Regression and log-linear models can be used to approximate the given data. both methods used for data compression, both handle sparse & skewed data.

- Non-parametric methods
 - Do not assume models
 - histograms, clustering, sampling

A) Histograms (a popular data reduction technique)

- Histograms use binning to approximate data distributions. A histogram for an attribute, A, partitions the data distribution of A into disjoint subsets, or buckets.
- Histograms. If each bucket represents only a single attribute-value/frequency pair, the buckets are called *singleton buckets*.
- 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, 18, 20, 20, 20, 20, 20, 20, 20, 21, 21, 21, 21, 25, 25, 25, 25, 25, 28, 28, 30, 30, 30.



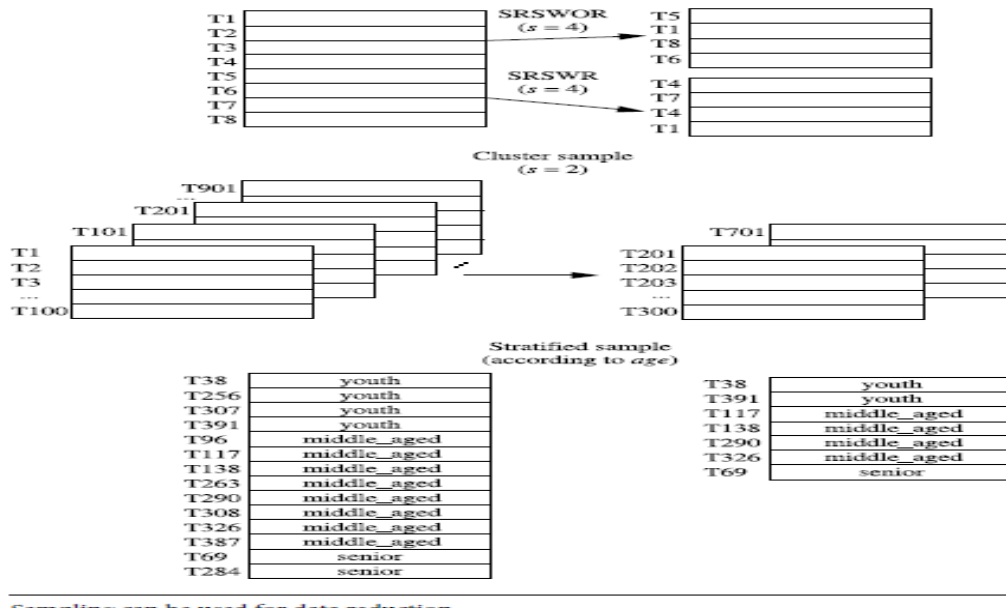
A histogram for *price* using singleton buckets—each bucket represents one price-value/frequency pair.

- Partitioning rules for the bucket:

B) 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. Let's look at the most common ways that we could sample D for data reduction are
- 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.



2.3.3 Data Transformation

Data Transformation strategies overview:

- The data are transformed or consolidated into forms appropriate for mining is called Transformation.

It involves-

- **Smoothing:** remove noise from data, techniques are binning, clustering and regression.
- **Aggregation:** summarization or aggregation operations are applied to the data. Ex: daily sales data may be aggregated. So as to compute monthly and annual total amounts. This is used in constructing data cube.

- **Generalization:** low level or primitive(raw) data are replaced by high-level concepts through the use of concept hierarchy. Ex: categorical attribute street can be generalized to higher-level concepts like city or country.
- **Normalization:** attribute data are scaled. So as to fall within a small, specified range, such as -1.0 to 1.0 or 0.0 to 1.0

2.3.4 Data Transformation by normalization:-

Normalization is useful for classification algorithms involving neural networks, or distance measurements such as nearest neighbor classification & clustering. Methods for data normalization are min-max normalization, z-score normalization and normalization by decimal scaling.

a) Min-max normalization : It performs a linear transformation on the original data. $\min A$ and $\max A$ are the minimum and maximum values of an attribute, A . Min-max normalization maps a value, v , of A to v' in the range $[\text{new min}A; \text{new max}A]$ by computing

Min-max normalization preserves the relationships among the original data values.

b) z-score normalization : (or zero-mean normalization), the values for an attribute, A , are normalized based on the mean and standard deviation of A . A value, v , of A is normalized to v' by computing

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

where \bar{A} and σ_A are the mean and standard deviation, respectively, of attribute A.

This method of normalization is useful when the actual minimum and maximum of attribute A are unknown or when there are outliers that dominate the min-max normalization

c) Normalization by decimal scaling : Normalizes by moving the decimal point of values of attribute A. The number of decimal points moved depends on the maximum absolute value of A.

A value, v , of A is normalized to v' by computing.

$$v' = \frac{v}{10^j},$$

where j is the smallest integer such that

$$\text{Max}(|v'|) < 1.$$

➤ **Attribute construction** : New attributes constructed from the given attributes and added to improve accuracy and to understanding of structure of high dimensional data.

Ex : add the attribute area based on the attributes height and width.

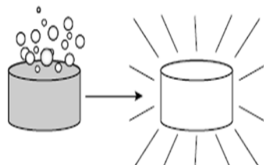
UNIT-II
Assignment-Cum-Tutorial Questions
SECTION-A

Objective Questions

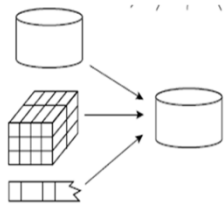
1. Real world data may contains _____ and _____ data.
2. When to apply the data preprocessing techniques for mining the data
 - A) Before mining.
 - B) During mining.
 - C) After mining.
 - D) All of the time.
3. Match the following :

$-2, 32, 100, 59, 48 \rightarrow -0.02, 0.32, 1.00, 0.59, 0.48$

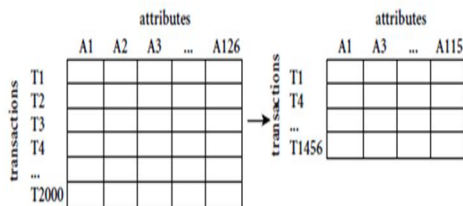
Data reduction



Data integration



Data transformation



Data cleaning

4. Use the attribute mean to fill the missing value of data []
1,2,3,4,5,6,__,7,8,9,10.
A) 2.0 B) 3.0 C) 5.5 D) 5.0
5. Data for Attendance : 50,55,60,65,70,75,80,85,90,95
Partition the above attendance data into equidepth bins of depth 5. []

- A) Bin 1:50,55,60,65,70 Bin 2: 75,80,85,90,95
 B) Bin 1:50,55,60,65 Bin 2: 70,75,80,85,90,95
 C) Bin 1:50,55,60,65,70,75 Bin 2: ,75,80,85,90,95
 D) Bin 1:50,55,60 Bin 2:65,70,75,80,85,90,95
6. For the above attendance apply bin means smoothing technique []
- A) Bin 1: 65,65,65,65,65 Bin2 : 85,85,85,85,85
 B) Bin 1: 60,60,60,60,60 Bin2 : 85,85,85,85,85
 C) Bin 1: 65,65,65,65,65 Bin2 : 80,80,80,80,80
 D) Bin 1: 75,75,75,75,75 Bin2 : 85,85,85,85,85
7. For the above attendance apply bin medians smoothing technique.
- A) Bin 1: 60,60,60,60,60 Bin2 : 85,85,85,85,85 []
 B) Bin 1: 65,65,65,65,65 Bin2 : 85,85,85,85,85
 C) Bin 1: 65,65,65,65,65 Bin2 : 80,80,80,80,80
 D) Bin 1: 75,75,75,75,75 Bin2 : 85,85,85,85,85
8. Data for Attendance : 4,8,15 Smoot by bin boundaries []
- A) 4,4,15 B) 4,15,15 C) 4,4,4 D) 15,15,15
9. Data Reduction is the process of reduced representation of data in size not in values. [T/F]
10. Reducing the number of attributes to solve the high dimensionality problem is called as _____. []
- A) Curse of dimensionality. B) Dimensionality reduction.
 C) Cleaning. D) Over fitting.
11. _____ and _____ are the popular and effective methods of lossy data compression technique.
12. ____ is the method of fitting the data values into a fixed model []
- A) Clustering. B) Regression. C. Smoothing. D) Aggregation.
13. Use min-max normalization transformation technique for finding transformed income value of \$10000 with min_income=1000,

max_income=50000 and mapping range of income [0.0,1.0] The
Transformed income value=_____.

- A) 0.225 B) 0.325 C) 0.425 D) 0.525

SECTION-B

SUBJECTIVE QUESTIONS

1. Illustrate the need for data preprocessing. List and explain various data preprocessing techniques.
2. What is data cleaning? Describe the approaches to fill missing values.
3. Define noisy data. Describe various techniques for smoothing noisy data.
4. Discuss the issues to be considered for data integration.
5. What is data normalization? Explain any two Normalization methods.
6. Outline about Data Cube Aggregation as a data reduction technique.
7. Elaborate different attribute subset selection methods with examples
8. What is a concept hierarchy? Explain different techniques used to generate concept hierarchy for categorical data.
9. Write short notes on Sampling in Numerosity Reduction.
10. Write short notes on Histograms in Numerosity Reduction.
11. Explain different sampling approaches used in data Reduction

Problems

12. Suppose that the data for analysis includes the attribute age. The age values for the data tuples are 13, 15, 16, 16, 19, 20, 20, 21, 22, 22, 25, 25, 25, 25, 30, 33, 33, 35, 35, 35, 35, 36, 40, 45, 46, 52, 70.
13. Question: Use smoothing by bin means to smooth the data, using a bin depth of 3. Illustrate your steps.
14. Apply the min-max normalization to transform the value 35 into the range [0.0, 1.0] using the data for age given in question 2.
15. Apply z-score normalization to transform the value 35 for age, where the standard deviation of age is 12.94 years. Using the data for age given in question 2.

16. Use these methods to *normalize* the following group of data:

200, 300, 400, 600,1000

- (a) min-max normalization by setting *min* D 0 and *max* D 1
- (b) z-score normalization
- (c) z-score normalization using the mean absolute deviation instead of standard deviation
- (d) normalization by decimal scaling

17. Suppose a hospital tested the age and body fat data for 18 randomly selected adults with the following result.

<i>age</i>	23	23	27	27	39	41	47	49	50
<i>%fat</i>	9.5	26.5	7.8	17.8	31.4	25.9	27.4	27.2	31.2
<i>age</i>	52	54	54	56	57	58	58	60	61
<i>%fat</i>	34.6	42.5	28.8	33.4	30.2	34.1	32.9	41.2	35.7

Calculate the mean and standard deviation of *age* and *%fat*.

UNIT – III

Data Warehouse and OLAP Technology

Objective:

To introduce the concepts of Data warehousing and Data mining.

Syllabus:

3.1 Data warehouse: Basic concepts,

3.2 OLAP vs. OLTP;

3.3 Data warehousing: A multitiered architecture; Datawarehouse modelling:

3.4 Data cube: A multidimensional data model, star, snowflake and fact constellation schemas for multidimensional data models,

3.5 Role of concept hierarchies,

3.6 Typical OLAP operations

Learning Outcomes:

At the end of the unit, students will be able to:

CO3 : Illustrate the major concepts and operations of multi dimensional data models.

Learning Material

3.1 Data warehouse:

- Data warehousing provides architectures and tools for business executives to systematically organize, understand, and use their data to make strategic decisions.
- Loosely speaking, a data warehouse refers to a data repository that is maintained separately from an organization's operational databases.
- **Definition:** According to William H. Inmon, "A data warehouse is a subject-oriented, integrated, time-variant, and nonvolatile collection of data in support of management's decision making process.
- **Subject-oriented:** A data warehouse is organized around major subjects,

such as customer, vendor, product, and sales. A data warehouses typically provide a simple and concise view around particular subject issues by excluding data that are not useful in the decision support process.

- **Integrated:** A data warehouse is usually constructed by **integrating** multiple heterogeneous sources, such as relational databases, flat files, and on-line transaction records. Data cleaning and data integration techniques are applied to ensure consistency of the data.
- **Time-variant:** Data are stored to provide information from a historical perspective (e.g., the past 5-10 years). Every key structure in the data warehouse contains, either implicitly or explicitly, an element of **time**.
- **Nonvolatile:** A data warehouse is always a physically separate store of data transformed from the application data found in the operational environment. Due to this separation, a data warehouse does not require transaction processing, recovery, and concurrency control mechanisms. It usually requires only two operations in data accessing: initial loading of data and access of data.

3.1.1. The traditional database approach to heterogeneous database integration

- 1) The traditional database approach to heterogeneous database integration is to build **wrappers** and **integrators** (or **mediators**) on top of multiple, heterogeneous databases.
- 2) When a query is posed to a client site, a metadata dictionary is used to translate the query into queries appropriate for the individual heterogeneous sites involved.
- 3) These queries are then mapped and sent to local query processors. The results returned from the different sites are integrated into a global answer set.

- 4) This **query-driven approach** requires complex information filtering and integration processes, and competes with local sites for processing resources.
- 5) It is inefficient and potentially expensive for frequent queries, especially queries requiring aggregations.

3.1.2 update driven approach:

- 1) Rather than using a query-driven approach, data warehousing employs an **updatedriven** approach in which information from multiple, heterogeneous sources is integrated in advance and stored in a warehouse for direct querying and analysis.
- 2) Unlike online transaction processing databases, data warehouses do not contain the most current information.
- 3) However, a data warehouse brings high performance to the integrated heterogeneous database system because data are copied, preprocessed, integrated, annotated, summarized, and restructured into one semantic data store.
- 4) Furthermore, query processing in data warehouses does not interfere with the processing at local sources.
- 5) Moreover, data warehouses can store and integrate historic information and support complex multidimensional queries. As a result, data warehousing has become popular in industry.

A model data warehouse of All Electronics is as follows:

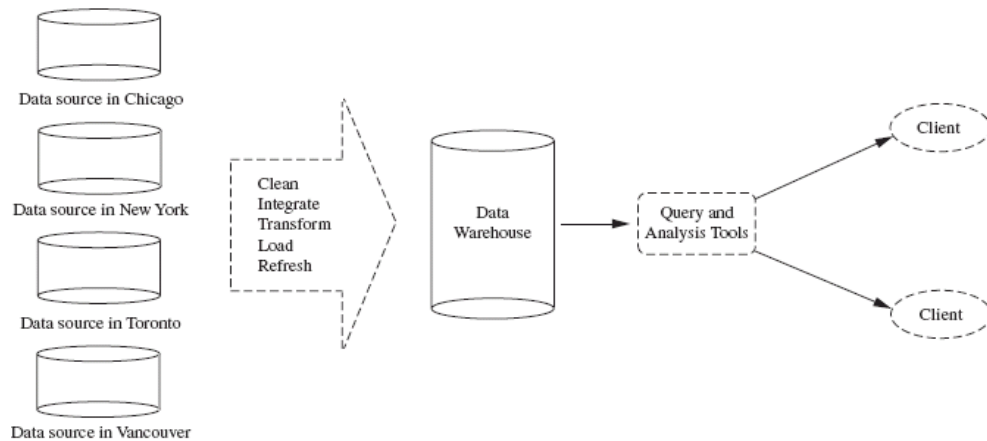


Fig : Data Warehousing and OLAP

3.2 OLAP vs. OLTP:

The major task of online operational database systems is to perform online transaction and query processing. These systems are called **online transaction processing (OLTP)** systems. 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 different users. These systems are known as **online analytical processing (OLAP)** systems.

3.2.1 Differences between OLTP and OLAP

1) Users and system orientation:

- a. An OLTP system is **customer-oriented** and is used for transaction and query processing by clerks, clients, and information technology professionals.
- b. An OLAP system is **market-oriented** and is used for data analysis by knowledge workers, including managers, executives, and analysts.

2) Data contents:

- a. An OLTP system manages current data that, typically, are too detailed to be easily used for decision making. (OLTP) systems cover most of the day-to-day operations of an organization, such as purchasing, inventory, manufacturing, banking, payroll, registration, and accounting.
- b. An OLAP system manages large amounts of historic 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 for informed decision making.

3) Database design:

- a. An OLTP system usually adopts an entity-relationship (ER) data model and an application-oriented database design.
- b. An OLAP system typically adopts either a *star* or a *snowflake* model and a subject-oriented database design.

4) View:

- a. An OLTP system focuses mainly on the current data within an enterprise or department, without referring to historic data or data in different organizations.
- b. An OLAP system often spans multiple versions of a database schema, due to the evolutionary process of an organization. OLAP systems also deal with information that originates from different organizations, integrating information from many data stores. Because of their huge volume, OLAP data are stored on multiple storage media.

5) Access patterns:

- a. The access patterns of an OLTP system consist mainly of short, atomic transactions. Such a system requires concurrency control and recovery mechanisms.

- b. Accesses to OLAP systems are mostly read-only operations (because most data warehouses store historic rather than up-to-date information), although many could be complex queries.

Comparison between OLTP and OLAP systems:

Feature	OLTP	OLAP
Characteristic	operational processing	informational processing
Orientation	transaction	analysis
User	clerk, DBA, database professional	knowledge worker (e.g., manager, executive, analyst)
Function	day-to-day operations	long-term informational requirements, decision support
DB design	ER based, application-oriented	star/snowflake, subject-oriented
Data	current; guaranteed up-to-date	historical; accuracy maintained over time
Summarization	primitive, highly detailed	summarized, consolidated
View	detailed, flat relational	summarized, multidimensional
Unit of work	short, simple transaction	complex query
Access	read/write	mostly read
Focus	data in	information out
Operations	index/hash on primary key	lots of scans
Number of records accessed	tens	millions
Number of users	thousands	hundreds
DB size	100 MB to GB	100 GB to TB
Priority	high performance, high availability	high flexibility, end-user autonomy
Metric	transaction throughput	query throughput, response time

3.2.2 Why Have a Separate Data Warehouse?

A data warehouse is kept **separate** from operational databases due to the following reasons:

- An operational database is constructed for **well-known tasks** and workloads such as searching particular records, indexing, etc. In contrast, data warehouse queries are often **complex** and they present a general form of data.

- Operational databases support concurrent processing of multiple transactions. Concurrency control and recovery mechanisms are required for operational databases to ensure robustness and consistency of the database.
- An operational database query allows to **read and modify** operations, while an OLAP query needs only **read only** access of stored data.
- A data warehouse maintains historical data whereas operational databases do not typically maintain historical data.
- Decision support requires consolidation (such as aggregation and summarization) of data from heterogeneous sources, resulting in high-quality, clean, and integrated data. In contrast, operational databases contain only detailed raw data, such as transactions, which need to be consolidated before analysis.
- Since the two systems provide quite different functionalities and require different kinds of data it is presently necessary to maintain separate databases.

3.3 DataWarehousing: A Multitiered Architecture:

1. The bottom tier is a **warehouse database server** that is almost always a relational database system.
 - a. Back-end tools and utilities are used to feed data into the bottom tier from operational databases or other external sources (e.g., customer profile information provided by external consultants).
 - b. Back-end tools and utilities perform data 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.

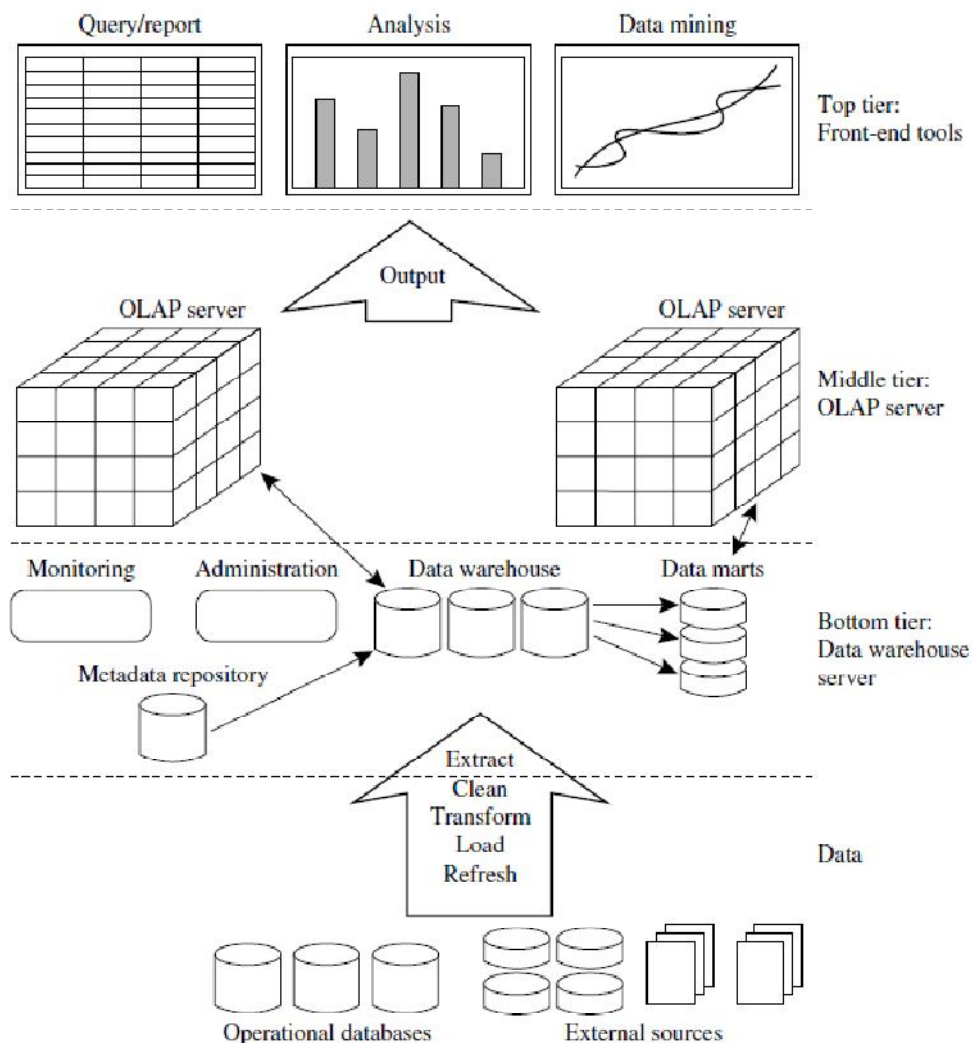


Figure 4.1 A three-tier data warehousing architecture.

- c. The data are extracted using application program interfaces known as gateways.
- d. A gateway is supported by the underlying DBMS and allows client programs to generate SQL code to be executed at a server.
- e. Examples of gateways include ODBC (Open Database Connection) and OLEDB (Object Linking and Embedding Database) by Microsoft and JDBC (Java Database Connection).

- f. This bottom tier also contains a metadata repository, which stores information about the data warehouse and its contents.
2. The middle tier is an **OLAP server** that is typically implemented using either a **relational OLAP (ROLAP)** model (i.e., an extended relational DBMS that maps operations on multidimensional data to standard relational operations); or a **multidimensional OLAP (MOLAP)** model (i.e., 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).

3.4 DataWarehouse Modeling: Data Cube and OLAP

Data warehouses and OLAP tools are based on a multidimensional data model. This model views data in the form of a data cube.

3.4.1 Data Cube: A Multidimensional Data Model

- a. "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.
- b. **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*.
- c. These dimensions allow the store to keep track of things like monthly sales of items and the branches and locations at which the items were sold.
- d. 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*.
- e. Dimension tables can be specified by users or experts, or automatically generated and adjusted based on data distributions.

- f. A multidimensional data model is typically organized around a **central theme**, such as *sales*. This theme is represented by a **fact table**.
- g. **Facts** are numeric measures. Think of them as the quantities by which we want to analyze relationships between dimensions.
- h. Examples of **facts** for a sales data warehouse include *dollars sold* (sales amount in dollars), *units sold* (number of units sold), and *amount budgeted*.
- i. The **fact table** contains the names of the *facts*, or measures, as well as keys to each of the related dimension tables.
- j. Although we usually think of cubes as 3-D geometric structures, in data warehousing the data cube is ***n*-dimensional**.

From Tables and Spreadsheets to Data Cubes

- ✓ In particular, we will look at the ***All Electronics*** sales data for items sold per quarter in the city of Vancouver.
- ✓ These data are shown in Table (a). In this 2-D representation, the sales for Vancouver are shown with respect to the *time* dimension (organized in quarters) and the *item* dimension (organized according to the types of items sold). The fact or measure displayed is *dollars sold* (in thousands).

Table(a) A 2-D view of sales data for *AllElectronics* according to the dimensions *time* and *item*, where the sales are from branches located in the city of Vancouver. The measure displayed is *dollars_sold* (in thousands).

<i>location = "Vancouver"</i>				
<i>time (quarter)</i>	<i>item (type)</i>			
	<i>home entertainment</i>	<i>computer</i>	<i>phone</i>	<i>security</i>
Q1	605	825	14	400
Q2	680	952	31	512
Q3	812	1023	30	501
Q4	927	1038	38	580

✓ Suppose that we would now like to view our sales data with a third dimension for instance, suppose we would like to view the data according to *time*, *item*, as well as *location* for the cities Vancouver, New York, Chicago, Toronto. These 3-D data shown Table(b). The Table(b) are represented as a series of 2-D tables. Conceptually, we may also represent the same data in the form of 3-D data cube, as in figure(a)

Table(b) A 3-D view of sales data for *AllElectronics*, according to the dimensions *time*, *item*, and *location*. The measure displayed is *dollars_sold* (in thousands).

<i>time</i>	<i>location = "Chicago"</i>				<i>location = "New York"</i>				<i>location = "Toronto"</i>				<i>location = "Vancouver"</i>			
	<i>Item</i>				<i>Item</i>				<i>Item</i>				<i>Item</i>			
	<i>home ent.</i>	<i>comp.</i>	<i>phone</i>	<i>sec.</i>	<i>home ent.</i>	<i>comp.</i>	<i>phone</i>	<i>sec.</i>	<i>home ent.</i>	<i>comp.</i>	<i>phone</i>	<i>sec.</i>	<i>home ent.</i>	<i>comp.</i>	<i>phone</i>	<i>sec.</i>
Q1	854	882	89	623	1087	968	38	872	818	746	43	591	605	825	14	400
Q2	943	890	64	698	1130	1024	41	925	894	769	52	682	680	952	31	512
Q3	1032	924	59	789	1034	1048	45	1002	940	795	58	728	812	1023	30	501
Q4	1129	992	63	870	1142	1091	54	984	978	864	59	784	927	1038	38	580

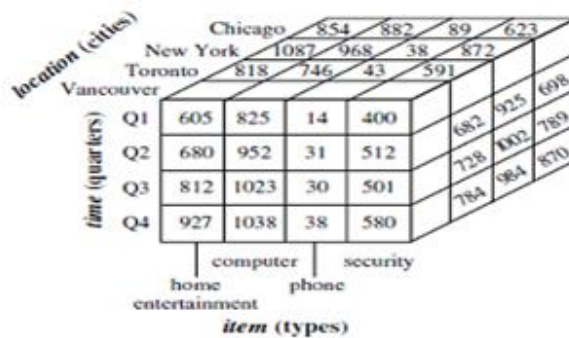
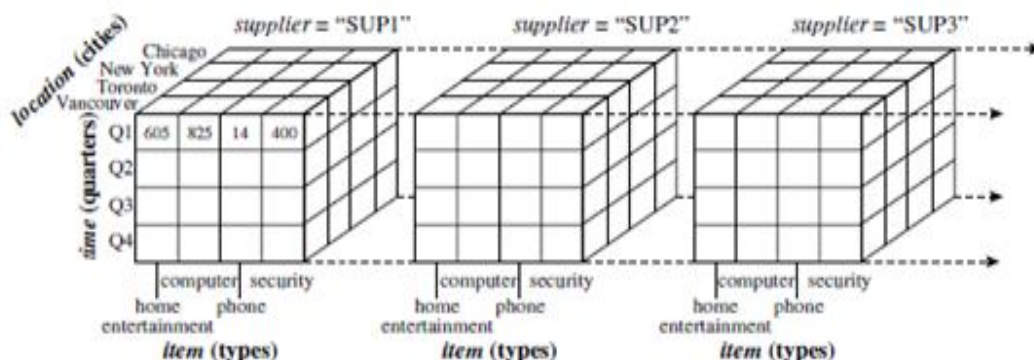


Figure (a) A 3-D data cube representation of the data in Table (b) according to the dimensions *time*, *item*, and *location*. The measure displayed is *dollars_sold* (in thousands).

- ✓ Suppose that we would now like to view our sales data with an additional fourth dimension, such as *supplier*. Viewing things in 4-D becomes tricky. However, we can think of a 4-D cube as being a series of 3-D cubes, as shown in figure(b) .
- ✓ If we continue in this way, we may display any n -D data as a series of $(n-1)$ -D "cubes."



Figure(b) A 4-D data cube representation of sales data, according to the dimensions *time*, *item*, *location*, and *supplier*. The measure displayed is *dollars_sold* (in thousands). For improved readability, only some of the cube values are shown.

- Given a set of dimensions, we can generate a cuboid for each of the possible subsets of the given dimensions. The result would form a lattice of cuboids, each showing the data at a different level of summarization, or group by. The lattice of cuboids is then referred to as a data cube. Figure (c) shows a lattice of cuboids forming a data cube for the dimensions time, item, location, and supplier.
- For example, the 4-D cuboid in Figure (b) is the base cuboid for the given time, item, location, and supplier dimensions. Figure (a) is a 3-D (non base) cuboid for time, item, and location, summarized for all suppliers
- The cuboid that holds the lowest level of summarization is called the "**base cuboid**", The 0-D cuboid, which holds the highest level of summarization, is called the "**apex cuboid**".

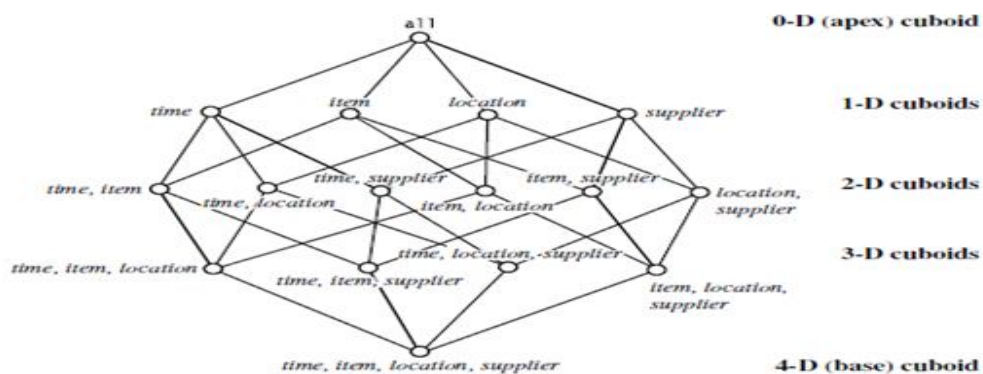


Figure (c) Lattice of cuboids, making up a 4-D data cube for the dimensions *time*, *item*, *location*, and *supplier*. Each cuboid represents a different degree of summarization.

3.4.2 Schemas for Multidimensional Databases- Stars, Snowflakes, and Fact Constellations:

The most popular data model for a data warehouse is a multidimensional model. Such a model can exist in the form of

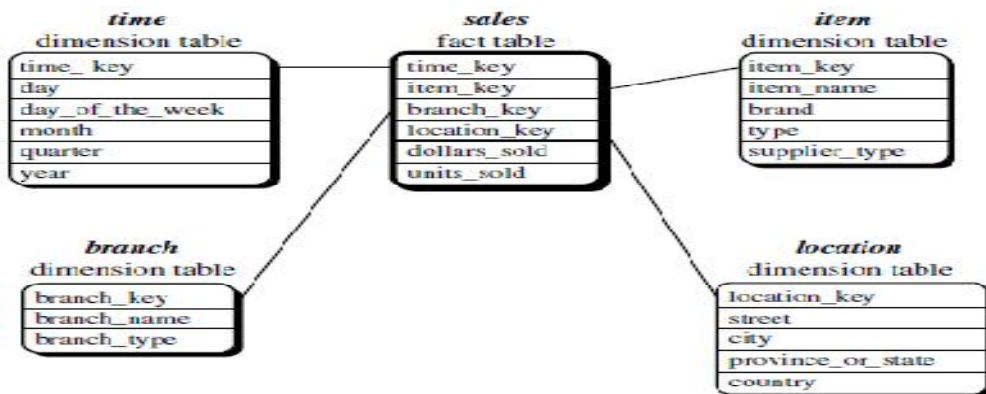
1) a star schema, 2) a snowflake schema, or 3) a fact constellation schema.

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(key and measures), with number of redundancy,
2. and a set of smaller attendant tables (dimension tables), one for each dimension. And each dimension table is joined to the fact table using primary key to foreign key join but dimension table are not joined to each other.

- ✓ It is also known as Star Join Schema.
- ✓ It is simplest style of data warehouse schema.
- ✓ It is called a Star Schema because the entity relationship diagram of schema resembles a star, with points radiating from central table.
- ✓ **Example 3.1 Star schema.** A star schema for *AllElectronics* sales is shown in Figure. Sales are considered along four dimensions: *time*, *item*, *branch*, and *location*. The schema contains 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 (e.g., *time key* and *item key*) are system-generated identifiers.



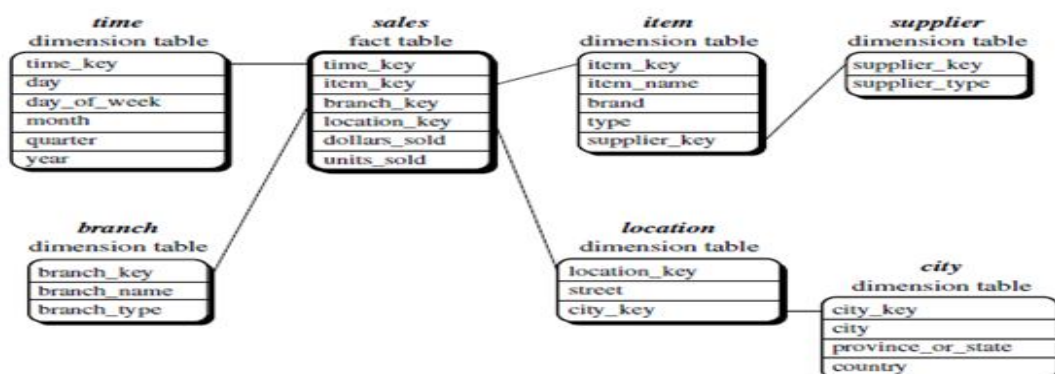
Star schema of a data warehouse for sales.

Advantages of Star Schema:

- Provide highly optimized performance for typical star queries.
- Provide a direct and intuitive mapping between the business entities being analyzed end uses and the schema design

Snowflake schema:

1. The snowflake schema is a variant of the star schema model, where some dimension tables are *normalized*, thereby further splitting the data into additional tables.
2. The resulting schema graph forms a shape similar to a snowflake..
3. The Snow flaking is only effecting the dimensional tables.

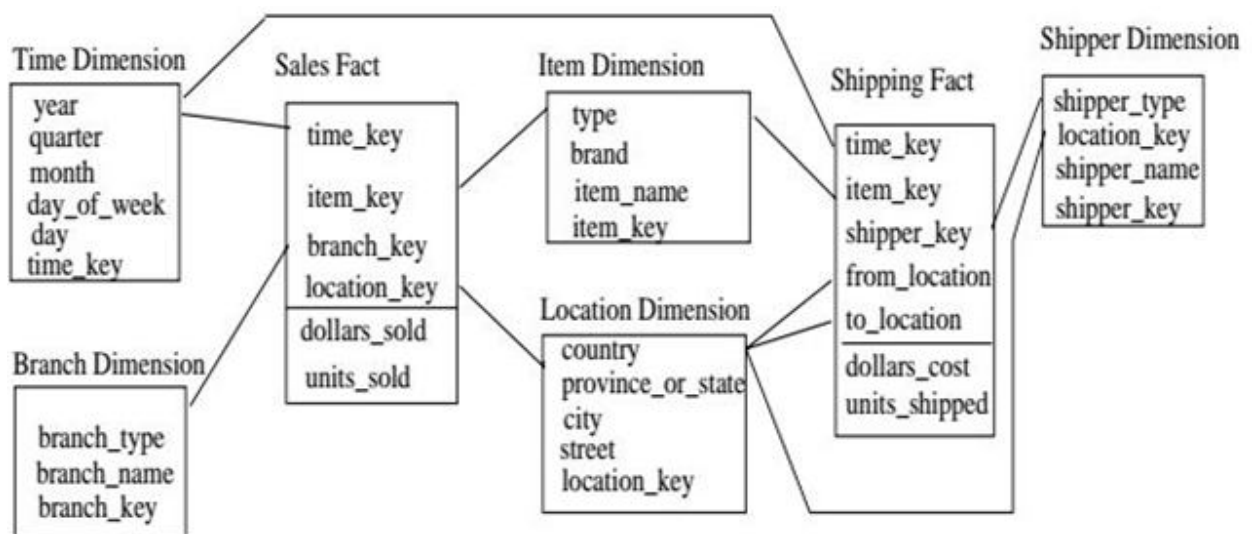


Snowflake schema of a data warehouse for sales.

The major difference between the Snowflake and Star schema models:

- The dimensional tables of the snowflake model may be kept in normalized form to reduce redundancies, which are easy to maintain and save storage a space.
- The Snowflakes structure can reduce the effectiveness of browsing since more joins will be needed to execute a query.
- Hence ,the Snow flake schema is not as popular as Star Schema in data warehouse design

Fact constellation: Sophisticated applications may require multiple fact tables to *share* dimension tables. This kind of schema can be viewed as a collection of stars, and hence is called a galaxy schema or a fact constellation.



Fact constellation schema of a data warehouse for sales and shipping.

In data warehousing, there is a distinction between a data warehouse and a data mart.

A data warehouse collects information about subjects that span the *entire organization*, such as *customers, items, sales, assets, and personnel*, and thus its scope is *enterprise-wide*.

For data warehouses, the fact constellation schema is commonly used, since it can model multiple, interrelated subjects.

A data mart, on the other hand, is a department subset of the data warehouse that focuses on selected subjects, and thus its scope is *department wide*. For data marts, the *star* or *snowflake* schema are commonly used,

Examples for defining star, snow flake, and fact constellation schemas

- Data warehouses and data marts can be defined using two language primitives, one for ***cube definition*** and one for ***dimension definition***.
- The *cube definition* statement has the following syntax:

define cube <cube name> [<dimension list>]: <measure list>

- The *dimension definition* statement has the following syntax:

define dimension <dimension name> as (<attribute or dimensions list>)

Star schema definition:

The star schema is defined in DMQL as follows:

define **cube sales star** [time, item, branch, location]:

dollars sold = sum(sales in dollars), units sold = count(*)

define dimension time as (time key, day, day of week, month, quarter,

year) define dimension item as (item key, item name, brand, type,

supplier type) define dimension branch as (branch key, branch name,

branch type)

define dimension location as (location key, street, city, province or state,

country)

The define cube statement defines a data cube called *sales star*, which corresponds to the central *sales* fact table with two measures, *dollars sold* and *units sold*.

The data cube has four dimensions, namely, *time*, *item*, *branch*, and *location*. A define dimension statement is used to define each of the dimensions.

Snowflake schema definition:

The snowflake schema is defined in DMQL as follows:

define cube sales **snowflake** [time, item, branch, location]:

dollars sold = sum(sales in dollars), units sold = count(*)

define dimension time as (time key, day, day of week, month, quarter, year)

define dimension item as (item key, item name, brand, type, supplier(supplier key, supplier type))

define dimension branch as (branch key, branch name, branch type)

define dimension location as (location key, street, city (city key, city, province or state, country))

This definition is similar to that of *sales star*, except that, here, the *item* and *location* dimension tables are normalized into two dimension tables, *item* and *supplier*.

Fact constellation schema definition:

The fact constellation schema is defined in DMQL as follows:

define cube sales [time, item, branch, location]:

dollars sold = sum(sales in dollars), units sold = count(*)

define dimension time as (time key, day, day of week, month, quarter, year)

define dimension item as (item key, item name, brand, type, supplier type)

define dimension branch as (branch key, branch name, branch type)

define dimension location as (location key, street, city, province or state, country).

Measures:

Measures can be organized into three categories based on the kind of aggregate functions used:

- Distributive,
- Algebraic,
- Holistic.

Distributive: An aggregate function is distributive if it can be computed in a distributed manner. Suppose the data are partitioned into n sets. We apply the function to each partition, resulting in n aggregate values.

For example, `count()` can be computed for a data cube by first partitioning the cube into a set of sub cubes, computing `count()` for each sub cube, and then summing up the counts obtained for each sub cube. Hence, `count()` is a distributive aggregate function.

`sum()`, `min()`, and `max()` are distributive aggregate functions.

Algebraic: An aggregate function is algebraic if it can be computed by an algebraic function with m arguments (where m is a bounded positive integer), each of which is obtained by applying a distributive aggregate function.

For example, `avg()` (average) can be computed by `sum()/count()`, where both `sum()` and `count()` are distributive aggregate functions.

Similarly, it can be shown that `min N()` and `max N()` (which find the N minimum and N maximum values, respectively, in a given set) and `standard deviation()` are algebraic aggregate functions

Holistic: An aggregate function is holistic if there is no constant bound on the storage size needed to describe

a sub aggregate. That is, there does not exist an algebraic function with m arguments (where m is a constant) that characterizes the computation.

Common examples of holistic functions include `median()`, `mode()`, and `rank()`.

Category	Examples
Distributive	<code>Sum()</code> , <code>Count()</code> , <code>Minimum()</code> , <code>Maximum()</code>
Algebraic	<code>Average()</code> , <code>StandardDeviation()</code> , <code>MaxN()</code> (N largest values), <code>MinN()</code> (N smallest values), <code>CenterOfMass()</code>
Holistic	<code>Median()</code> , <code>MostFrequent()</code> , <code>Rank()</code>

3.5 Concept Hierarchies:

A concept hierarchy defines a sequence of mappings from a set of low level concepts to higher level, more general concepts.

Consider a concept hierarchy for the dimension location. City values for location include Vancouver, Montreal, New York, and Chicago. Each city, however, can be mapped to the province or state to which it belongs.

For example, Vancouver can be mapped to British Columbia, and Chicago to Illinois.

The provinces and states can in turn be mapped to the country to which they belong, such as Canada or the USA.

These mappings form a concept hierarchy for the dimension location, mapping a set of low level concepts (i.e., cities) to higher level, more general concepts (i.e., countries). The concept hierarchy described above is illustrated in the following Figure

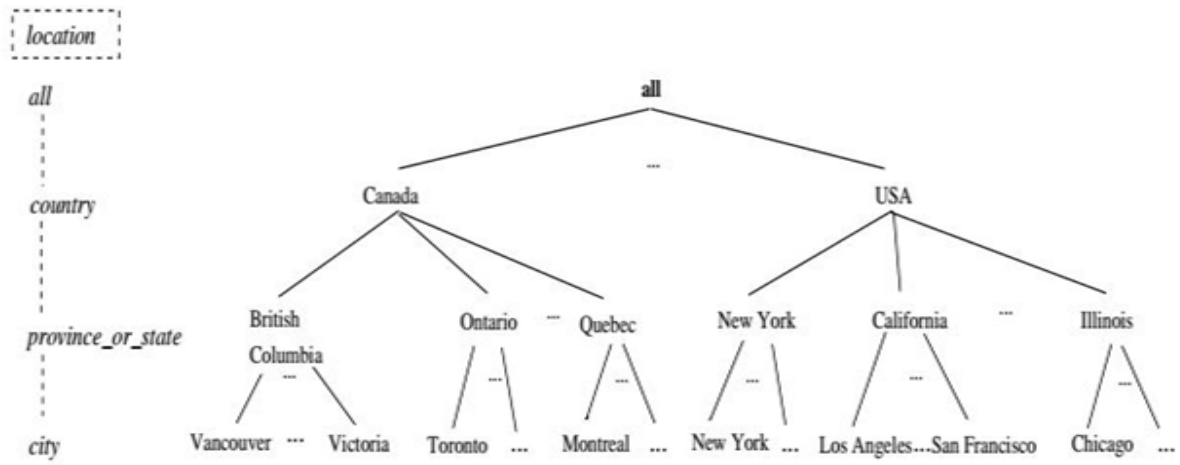
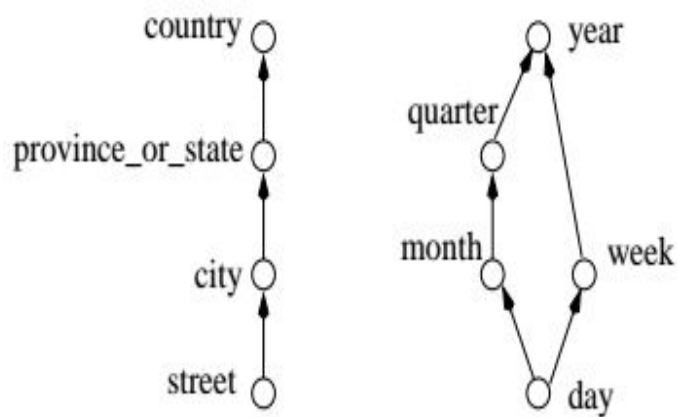


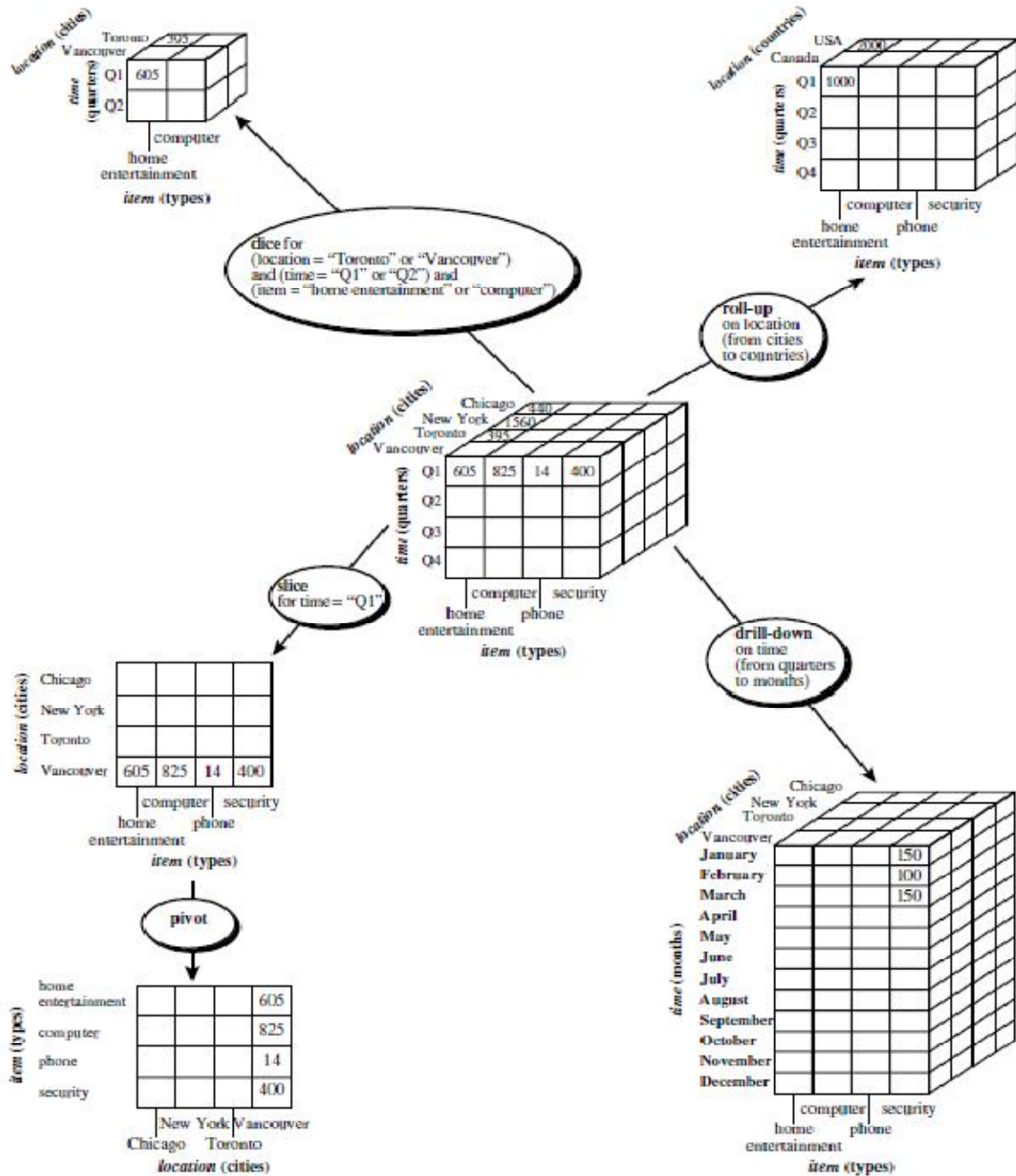
Figure 2.7 A concept hierarchy for the dimension *location*.



a) a hierarchy for *location* b) a lattice for *time*

Figure 2.8: Hierarchical and lattice structures of attributes in warehouse dimensions.

3.6 OLAP operations in the multidimensional data model:



Examples of typical OLAP operations on multidimensional data.

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.

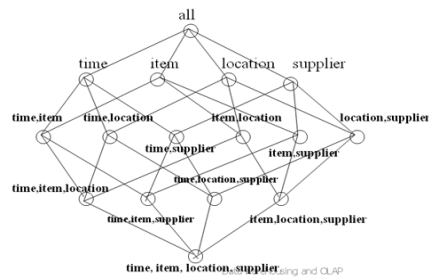
A number of OLAP data cube operations exist to materialize these different views, allowing interactive querying and analysis of the data at hand. Hence, OLAP provides a user-friendly environment for interactive data analysis.

1. **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.
2. **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.
3. **Slice and dice:** The slice operation performs a selection on one dimension of the given cube, resulting in a sub cube.
4. **Pivot (rotate):** Pivot (also called "rotate") is a visualization operation which rotates the data axes in view in order to provide an alternative presentation of the data.

- c) Slice and Dice iii) Step down Dimension
 d) Pivot iv) Climbing Up dimension []
 A) i,iii,ii,iv B) iii,iv,ii,i C) iv,iii,i,ii D) iv,iii,ii,i

11. _____ is a subset of the data warehouse and is usually oriented to a specific business line or team.

12. The following figure shows _____



SECTION-B

SUBJECTIVE QUESTIONS

1. Define Data warehouse and Write about the need of a separate Data Warehouse.
2. Differentiate between the main functionalities of OLTP and OLAP.
3. Develop various multi dimensional data model schemas.
4. Elaborate OLAP operations in multidimensional data model.
5. Describe three tier data warehouse architecture with a neat diagram.
6. Draw a concept hierarchy for dimension location, by considering the location values as Village < mandal < district < state.
7. Suppose that a data warehouse consists of the three dimensions time, doctor, and patient, and the two measures count and charge, where charge is the fee that a doctor charges a patient for a visit.
 - Draw star schema and snowflakes schema for the above data warehouse.
8. Draw a lattice of cuboids for the dimension containing five levels (including all), such as "student < major < status < university < all"?
9. Outline the implementation of data warehouse.

UNIT-IV

Mining Frequent Patterns, Association, and Correlations

Objectives:

Students should be able to

- Discovers the interesting association relationships among huge amount of data.

Syllabus:

Basic Concepts, frequent item sets, closed item sets and association rules, frequent item set mining methods : Apriori Algorithm, generations, association rules form frequent item sets, A Pattern- Growth approach for mining frequent item sets.

Outcomes:

Students should be able to

- Apply the different methods for mining Association rules.
- Understand the role of support and confidence.

Learning Material

4.1. Basic Concepts

Association Rule Mining

- Association rule mining finding frequent patterns, associations, correlations among sets of items or objects in transactional databases, traditional databases, relational databases or other information repositories.
- The discovery of interesting association relationships among huge amounts of business transaction records can help in many business decision making processes, such as market basket analysis, catalog design, cross-marketing, and loss-leader analysis.

Market Basket Analysis:

Association rule mining searches for interesting relationships among items in a given data set.

This process analyzes customer buying habits by finding associations between the different items that customers place in their “shopping baskets”. 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 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. For example, placing milk and bread within **close proximity** may further encourage the sale of these items together within single visits to the store.

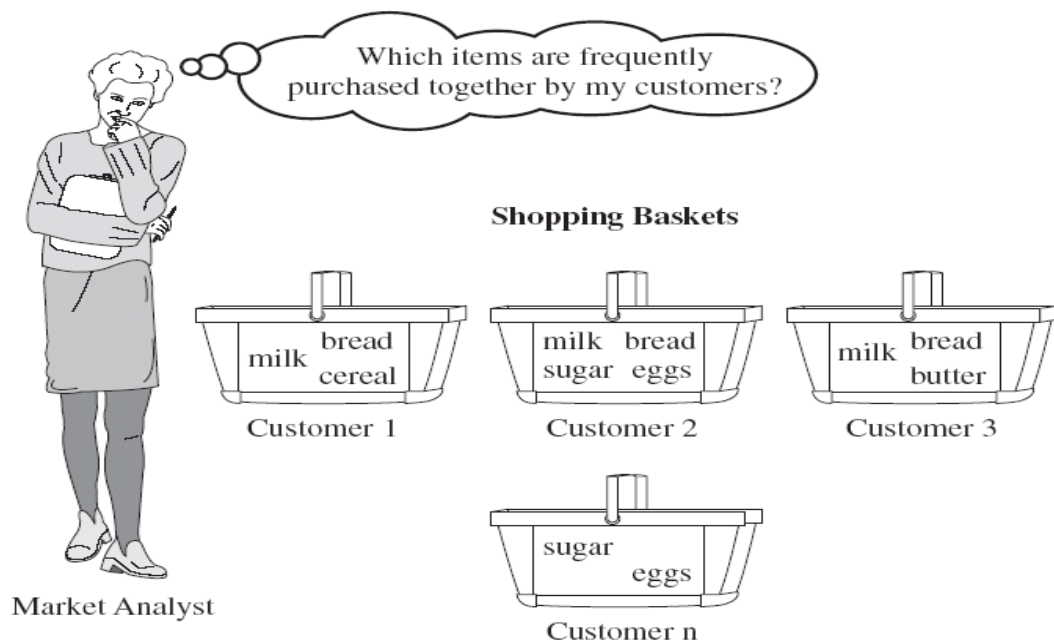


Figure 4.1 Market basket analysis.

Analyze the buying patterns that reflect items that are frequently Associated or purchased together. These patterns can be represented in the form of Association rules.

Rule form: "Body => Head [support, confidence]".

For example, the information that customers who purchase computers Also tend to buy financial management software at the same time is represented in Associated Rule below:

Computer=>financial_management_software[support=2%,confidence= 60%]

Rule **support and confidence** are two measures of rule interestingness, they respectively reflect the usefulness and certainty of discovered rules. 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 the users or domain experts.

4.1.1. Frequent Itemsets, Closed Itemsets, and Association Rules

Let I be a set of items $\{I_1, I_2, I_3, \dots, I_m\}$, Let D be a set of database transactions where each transaction T is a set of items such that $T \subseteq I$. Each transaction is associated with an identifier, called TID .

Let A be a set of items. A transaction T is said to contain A if and only if $A \subseteq T$.

An **association rule** is an implication of the form $A \Rightarrow B$ where $A \subseteq I$, $B \subseteq I$, and $A \cap B = \Phi$.

The rule $A \Rightarrow B$ holds in the transaction set D with **support s** , where **s** is **the percentage of transactions in D that contain $A \cup B$** (i.e both A and B). This is taken to be the probability, $P(A \cup B)$.

The rule $A \Rightarrow B$ has **confidence c** in the transaction set D if c is the **percentage of transactions in D containing A that also contain B** . This is taken to be the **conditional probability $P(B/A)$** . That is

$$\text{Support (A} \Rightarrow \text{B)} = P(A \cup B).$$

$$\text{Confidence (A} \Rightarrow \text{B)} = P(B/A).$$

Strong Association Rules : Rules that satisfy both a minimum support threshold (min_sup) and a minimum confidence threshold (min_conf) are called **strong**. By convention, we write support and confidence value so as to occur between 0% and 100% rather than 0 to 1.0.

Item set : A set of items is referred to as an itemset. An itemset that contains k items is a k -itemset. Ex : The set {computer, financial_management_software} is a 2-itemset.

Support count : The occurrence frequency of an itemset is the number of transactions that contain the itemset. This is also known as the **frequency, support count** or **count** of the itemset.

An itemset satisfies minimum support if the occurrence frequency of the itemset is greater than or equal to the product of min_sup and the total number of transactions in D. The number of transactions required for the itemset to satisfy minimum support is therefore referred to as the minimum support count.

Frequent itemset: if an itemset satisfies minimum support, then it is a frequent itemset .

The set of frequent K-itemsets is commonly denoted by L_K .

Association rule mining is a two-step process:

step 1: Find all frequent itemsets: By definition, each of these itemsets will occur at least as frequently as a pre-determined minimum support count.

step 2 : Generate strong association rules from the frequent itemsets: By definition ,these rules must satisfy minimum support and minimum confidence. The overall performance is determined by the First step.

Support & confidence:

Support(s) of an association rule is defined as the percentage/fraction of records that contain A U B to the total number of records in the database.

$$\text{Support (A->B)} = P(A \cup B) = \frac{\text{support_count}(A \cup B)}{\text{count}(\text{all_transactions})}$$

Confidence of an association rule is defined as the percentage/fraction of the number of transactions that contain $A \cup B$ to the total number of records that contain A .

$$\begin{aligned}\text{Confidence}(A \rightarrow B) &= P(B/A) = \text{support}(A \cup B) / \text{support}(A) \\ &= \text{support_count}(A \cup B) / \text{support_count}(A).\end{aligned}$$

Example :

Data set D

TID	Itemsets
T100	1 3 4
T200	2 3 5
T300	1 2 3 5
T400	2 5

Support count, Support and Confidence:

$$\text{Support count}(1,3)=2$$

$$|D| = 4$$

$$\text{Support}(1 \rightarrow 3) = 2/4 = 0.5$$

$$\text{Support}(3 \rightarrow 2) = 2/4 = 0.5$$

$$\text{Confidence}(3 \rightarrow 2) = \text{count}(2 \cup 3) / \text{count}(3)$$

$$= 2/3$$

$$= 0.67$$

Association Rules

Association rules can be **classified** in various ways, based on the following criteria:

- **Based on the types of values handled in the rule:**

if a rule concerns associations between the presence or absence of items ,it is a **Boolean association rule**.

Computer=>financial_management_software[support=2%,confidence= 60%]

For example, the rule above is a Boolean association rule obtained from market basket analysis.

If a rule describes associations between quantitative items or attributes, then it is a **quantitative association rule**

In these rules , quantitative values for items or attributes are partitioned into intervals. The following rule is an example of a quantitative association rule, where X is a variable representing a customer:

age(X,"30.....39")^income(X,"42K.....48")=>buys(X,highresolutionTV).

The quantitative attributes age and income ,have been discretized,

- **Based on dimensions of data involved in the rule:**

If the items or attributes in an association rule reference only one dimension, then it is a **single-dimensional** association rule.

buys(X,"computer") => buys(X,"financial_management_software")

The above rules a single dimensional association rule since it refers to only one dimension, buys.

If a rule references two or more dimensions, such as the dimensions `buys`, `time_of_transaction`, and `customer_category`, then it is a **multidimensional association rule**.

$\text{age}(X, "30\dots\dots 39") \wedge \text{income}(X, "42K\dots\dots 48") \Rightarrow \text{buys}(X, \text{high resolution TV})$.

- **Based on the levels of abstraction in the rule set:**

A method for association rule mining can find rules at differing levels of abstraction. For example, suppose that a set of association rule mined includes the following rules:

$\text{age}(x, "30\dots 39") \text{ buys}(x, \text{"laptop computer"})$.

$\text{age}(x, "30\dots 39") \text{ buys}(x, \text{"computer"})$.

In above rules the items bought are referenced at different levels of abstraction.

e.g., "computer" is a higher-level abstraction of "laptop computer".

- **Based on various extensions to association mining:**

Association mining can be extended to correlation analysis, where the absence or presence of correlated items can be identified. It can also be extended to mining max patterns (i.e., maximal frequent patterns) and frequent closed itemsets.

Closed and max pattern frequent itemsets

Closed and max pattern frequent itemsets is a frequent pattern, p , such that any proper sub pattern of p is not frequent. A frequent **closed itemset** is a frequent closed itemset where an itemset c is closed if there exists no proper superset of c , c' such that every transaction containing c also contains c' .

Maxpatterns and frequent closed itemset can be used to substantially reduce the number of frequent itemsets generated in mining.

4.2 Frequent Item set mining Methods:

It is a two step process:

Step 1: Generation of frequent itemsets.

- Apriori algorithm.
- *Frequent-pattern growth (FP-Growth)*.

Step 2: Generation of Association rules for the above frequent itemsets.

4.2.1. The Apriori algorithm : Finding Frequent itemsets by Confined candidate generation

- This algorithm uses the **prior** knowledge of frequent itemset properties.
- Uses an iterative approach known as a level-wise search, where k-itemsets are used to explore (k+1)-itemsets.
- First, the set of frequent 1-itemsets is found. This set is denoted by L_1 . 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 L_K requires **one full scan** of the database.
- **Apriori property to reduce the search space:** “ All nonempty subsets of a frequent itemset must also be frequent.”
- $P(I) < \text{min_sup} \Rightarrow I$ is not frequent.
- If an item A is added to the itemset I, then the resulting itemset (I U A) can not occur more frequently than I. Therefore, I U A is not frequent either, i.e. $P(I+A) < \text{min_sup}$
- **Anti-Monotone property** – “if a set cannot pass a test, all of its supersets will fail the same test as well “

- Using the apriori property in the algorithm:
 - Let us look at how L_{k-1} is used to find L_k , for $k \geq 2$
- Two steps:
 - **Join** : C_k is generated by joining L_{k-1} with itself.
 - **Prune**: Any $(k-1)$ -itemset that is not frequent cannot be a subset of a frequent k -itemset.
 - Scanning the database to determine the count of each candidate in C_k – heavy computation
 - To reduce the size of C_k the Apriori property is used: if any $(k-1)$ subset of a candidate k -itemset is not in L_{k-1} , then the candidate cannot be frequent either, so it can be removed from C_k . – subset testing (hash tree)

Example: Transactional data for an All Electronics branch.

TID	List of items_IDs
T100	I1,I2,I5
T200	I2,I4
T300	I2,I3
T400	I1,I2,I4
T500	I1,I3
T600	I2,I3
T700	I1,I3

T800	I1,I2,I3,I5
T900	I1,I2,I3

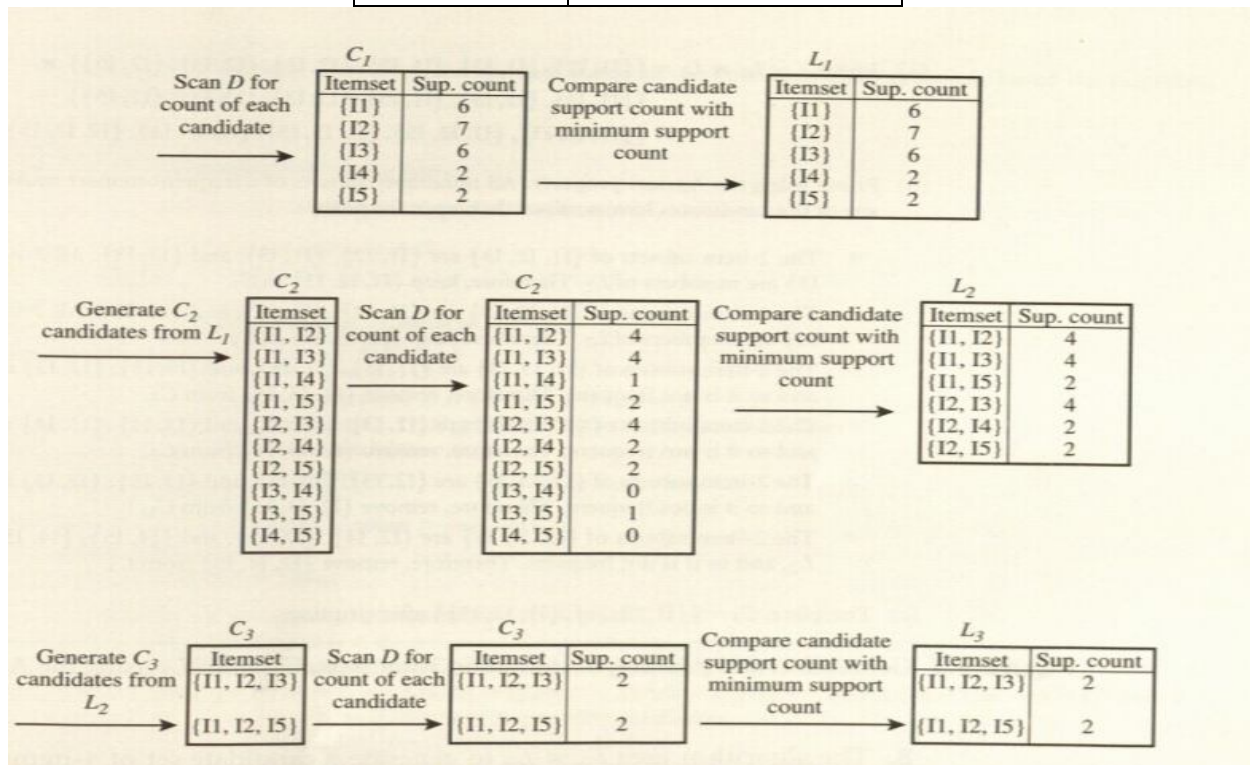


Figure: Generation of candidate itemsets and frequent itemsets, where the minimum support count is 2.

- Scan D for count of each candidate

C_1 : I1 – 6, I2 – 7, I3 – 6, I4 – 2, I5 – 2

- Compare candidate support count with minimum support count (min_sup=2)

L_1 : I1 – 6, I2 – 7, I3 – 6, I4 – 2, I5 – 2

- Generate C_2 candidates from L_1 and scan D for count of each candidate

C_2 : {I1,I2} – 4, {I1, I3} – 4, {I1, I4} – 1, ...

- Compare candidate support count with minimum support count

L2: {I1,I2} – 4, {I1, I3} – 4, {I1, I5} – 2, {I2, I3} – 4, {I2, I4} – 2, {I2, I5} – 2

- Generate C3 candidates from L2 using the join and prune steps:

Join: $C3=L2 \times L2 = \{\{I1, I2, I3\}, \{I1, I2, I5\}, \{I1, I3, I5\}, \{I2, I3, I4\}, \{I2, I3, I5\}, \{I2, I4, I5\}\}$

Prune: C3: {I1, I2, I3}, {I1, I2, I5}

- Scan D for count of each candidate

C3: {I1, I2, I3} – 2, {I1, I2, I5} – 2

- Compare candidate support count with minimum support count

L3: {I1, I2, I3} – 2, {I1, I2, I5} – 2

- Generate C4 candidates from L3

$C4=L3 \times L3 = \{I1, I2, I3, I5\}$

This itemset is pruned, because its subset $\{\{I2, I3, I5\}\}$ is not frequent => C4=null

Algorithm: Apriori. Find frequent itemsets using an iterative level-wise approach based on candidate generation.

Input:

- D , a database of transactions;
- min_sup , the minimum support count threshold.

Output: L , frequent itemsets in D .

Method:

```

(1)  $L_1 = \text{find\_frequent\_1-itemsets}(D)$ ;
(2) for ( $k = 2; L_{k-1} \neq \phi; k++$ ) {
(3)    $C_k = \text{apriori\_gen}(L_{k-1})$ ;
(4)   for each transaction  $t \in D$  { // scan  $D$  for counts
(5)      $C_t = \text{subset}(C_k, t)$ ; // get the subsets of  $t$  that are candidates
(6)     for each candidate  $c \in C_t$ 
(7)        $c.\text{count}++$ ;
(8)   }
(9)    $L_k = \{c \in C_k \mid c.\text{count} \geq min\_sup\}$ 
(10) }
(11) return  $L = \cup_k L_k$ ;

```

procedure apriori_gen(L_{k-1} :frequent ($k-1$)-itemsets)

```

(1) for each itemset  $l_1 \in L_{k-1}$ 
(2)   for each itemset  $l_2 \in L_{k-1}$ 
(3)     if ( $l_1[1] = l_2[1] \wedge l_1[2] = l_2[2] \wedge \dots \wedge l_1[k-2] = l_2[k-2] \wedge l_1[k-1] < l_2[k-1]$ ) then {
(4)        $c = l_1 \bowtie l_2$ ; // join step: generate candidates
(5)       if has_infrequent_subset( $c, L_{k-1}$ ) then
(6)         delete  $c$ ; // prune step: remove unfruitful candidate
(7)       else add  $c$  to  $C_k$ ;
(8)     }
(9) return  $C_k$ ;

```

procedure has_infrequent_subset(c : candidate k -itemset;

```

   $L_{k-1}$ : frequent ( $k-1$ )-itemsets); // use prior knowledge
(1) for each ( $k-1$ )-subset  $s$  of  $c$ 
(2)   if  $s \notin L_{k-1}$  then
(3)     return TRUE;
(4) return FALSE;

```

The Apriori algorithm for discovering frequent itemsets for mining Boolean association rules.

4.2.2 Generating association rules from frequent itemsets

- Generating strong association rules:

For each frequent itemset “l”, generate all nonempty subsets of l.

Confidence(A=>B)=P(B | A)= support_count(AUB)/support_count(A)

- support_count(AUB) – number of transactions containing the itemsets AUB
 - support_count(A) - number of transactions containing the itemsets A
- for every nonempty subset s of l, output the rule s=>(l-s) if support_count(l)/support_count(s)>=min_conf

Example: lets have l={I1, I2, I5}

The nonempty subsets are {I1, I2}, {I1, I5}, {I2, I5}, {I1}, {I2}, {I5}.

Generating association rules:

I1 and I2=>I5	conf=2/4=50%
I1 and I5=>I2	conf=2/2=100%
I2 and I5=> I1	conf=2/2=100%
I1=>I2 and I5	conf=2/6=33%
I2=>I1 and I5	conf=2/7=29%
I5=>I1 and I2	conf=2/2=100%

If `min_conf` is 70%, then only the second, third and last rules above are output.

4.3. A Pattern-Growth approach for mining frequent item sets

Two Steps:

1. Scan the transaction DB for the **first time**, find frequent items (single item patterns) and order them into a list L in frequency descending order.
 - In the format of (item-name, support)
2. For each transaction, order its frequent items according to the order in L; **Scan DB the second time**, construct FP-tree by putting each frequency ordered transaction onto it.

- **FP-Tree Definition**

- FP-tree is a **frequent pattern tree**. Formally, FP-tree is a tree structure defined below:
 1. One root labeled as "null", a set of *item prefix sub-trees* as the children of the root, and a *frequent-item header table*.
 2. Each node in *the item prefix sub-trees* has three fields:
 - item-name : register which item this node represents,
 - count, the number of transactions represented by the portion of the path reaching this node,
 - node-link that links to the next node in the FP-tree carrying the same item-name, or null if there is none.
 3. Each entry in the *frequent-item header table* has two fields,
 - item-name, and head of node-link that points to the first node in the FP-tree carrying the item-name.

- **Advantages of the FP-tree Structure**

- The most significant advantage of the FP-tree

- Scan the DB only twice and twice only.
- Completeness:
 - the FP-tree contains all the information related to mining frequent patterns (given the min-support threshold). Why?
- Compactness:
 - The size of the tree is bounded by the occurrences of frequent items
 - The height of the tree is bounded by the maximum number of items in a transaction
- **Mining Frequent Patterns Using FP-tree**
 - General idea (divide-and-conquer)
 - Recursively grow frequent patterns using the FP-tree: looking for shorter ones
Recursively and then concatenating the suffix:
 - For each frequent item, construct its conditional pattern base, and then its conditional FP-tree;
 - Repeat the process on each newly created conditional FP-tree until the resulting FP-tree is empty, or it contains only one path (single path will generate all the combinations of its sub-paths, each of which is a frequent pattern)
- **FP-Growth Method : An Example**

TID	List of items_IDs
T100	I1,I2,I5
T200	I2,I4
T300	I2,I3
T400	I1,I2,I4
T500	I1,I3
T600	I2,I3
T700	I1,I3

T800	I1,I2,I3,I5
T900	I1,I2,I3

Consider the same previous example of a database, D , consisting of 9 transactions.

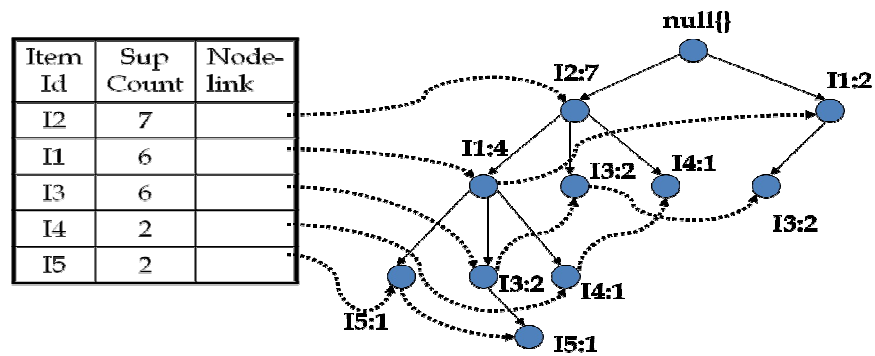
Suppose min. support count required is 2 (i.e. $\text{min_sup} = 2/9 = 22\%$)

The first scan of database is same as Apriori, which derives the set of 1-itemsets & their support counts. The set of frequent items is sorted in the order of descending support count.

The resulting set is denoted as $L = \{I2:7, I1:6, I3:6, I4:2, I5:2\}$

- **FP-Growth Method: Construction of FP-Tree**

- First, create the root of the tree, labeled with “null”.
- Scan the database D a second time. (First time we scanned it to create 1-itemset and then L).
- The items in each transaction are processed in L order (i.e. sorted order).
- A branch is created for each transaction with items having their support count separated by colon.
- Whenever the same node is encountered in another transaction, we just increment the support count of the common node or Prefix.
- 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.
- Now, The problem of mining frequent patterns in database is transformed to that of mining the FP-Tree.



An FP-Tree that registers compressed, frequent pattern information

Mining the FP-Tree by Creating Conditional (sub) pattern bases

Steps:

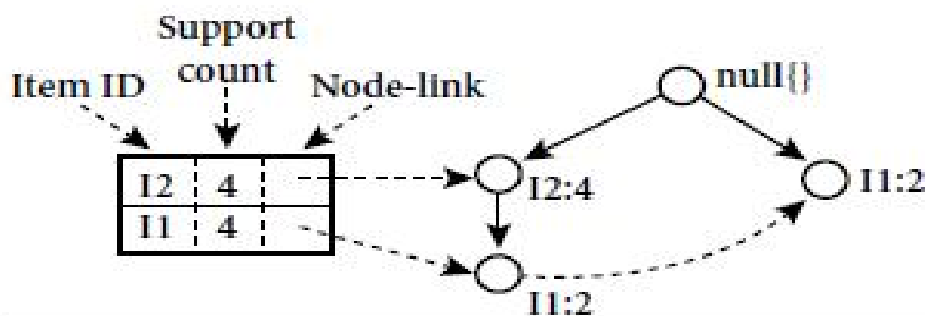
1. Start from each frequent length-1 pattern (as an initial suffix pattern).
2. Construct its conditional pattern base which consists of the set of prefix paths in the FP-Tree co-occurring with suffix pattern.
3. Then, Construct its conditional FP-Tree & perform mining on such a tree.
4. The pattern growth is achieved by concatenation of the suffix pattern with the frequent patterns generated from a conditional FP-Tree.
5. The union of all frequent patterns (generated by step 4) gives the required frequent itemset.

Item	Conditional pattern base	Conditional FP-Tree	Frequent patterns generated
I5	{(I2 I1: 1),(I2 I1 I3: 1)}	<I2:2 , I1:2>	I2 I5:2, I1 I5:2, I2 I1 I5: 2
I4	{(I2 I1: 1),(I2: 1)}	<I2: 2>	I2 I4: 2
I3	{(I2 I1: 1),(I2: 2), (I1: 2)}	<I2: 4, I1: 2>,<I1:2>	I2 I3:4, I1, I3: 2 , I2 I1 I3: 2

I2	{(I2: 4)}	<I2: 4>	I2 I1: 4
----	-----------	---------	----------

Mining the FP-Tree by creating conditional (sub) pattern bases

- Now, Following the above mentioned steps:
 - Lets start from I5. The I5 is involved in 2 branches namely {I2 I1 I5: 1} and {I2 I1 I3 I5: 1}.
 - Therefore considering I5 as suffix, its 2 corresponding prefix paths would be {I2 I1: 1} and {I2 I1 I3: 1}, which forms its conditional pattern base.
- **Properties of FP-Tree**
 - Node-link property: For any frequent item a_i , all the possible frequent patterns that contain a_i can be obtained by following a_i 's node-links, starting from a_i 's head in the FP-tree header.
 - Prefix path property : To calculate the frequent patterns for a node a_i in a path P , only the prefix sub-path of a_i in P need to be accumulated, and its frequency count should carry the same count as node a_i .



The conditional FP-tree associated with the conditional node I3.

- Out of these, Only I1 & I2 is selected in the conditional FP-Tree because I3 is not satisfying the minimum support count.
 - For I1 , support count in conditional pattern base = 1 + 1 = 2
 - For I2 , support count in conditional pattern base = 1 + 1 = 2
 - For I3, support count in conditional pattern base = 1

- Thus support count for I3 is less than required min_sup which is 2 here.
- Now , We have conditional FP-Tree with us.
- All frequent pattern corresponding to suffix I5 are generated by considering all possible combinations of I5 and conditional FP-Tree.
- The same procedure is applied to suffixes I4, I3 and I1.
- Note: I2 is not taken into consideration for suffix because it doesn't have any prefix at all.

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 $N.item-name = p.item-name$, 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 $FP_growth(Tree, \alpha)$

- (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 $\beta \cup \alpha$ with $support_count = minimum\ support\ count\ of\ nodes\ in\ \beta$;
- (4) **else for each** a_i in the header of $Tree$ **{**
- (5) generate pattern $\beta = a_i \cup \alpha$ with $support_count = a_i.support_count$;
- (6) construct β 's conditional pattern base and then β 's conditional FP_tree $Tree_\beta$;
- (7) **if** $Tree_\beta \neq \emptyset$ **then**
- (8) call $FP_growth(Tree_\beta, \beta)$; **}**

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

Reasons for the fastness of the Frequent Pattern Growth

- No candidate generation, no candidate test
- Use compact data structure
- Eliminate repeated database scan
- Basic operation is counting and FP-tree building

UNIT-IV
Assignment-Cum-Tutorial Questions
SECTION-A

Objective Questions

1. The market basket analysis is a typical example of _____
2. The interestingness measures of Association rule mining are _____ and _____.
3. Association rules are considered interesting if they satisfy_____ []
 A) Minimum support threshold. B) Minimum confidence threshold.
 C) Both A & B. D) Either A or B.
4. The formula for $\text{Support}(A \Rightarrow B) =$
5. The formula for $\text{Confidence}(A \Rightarrow B) =$
6. An association rule mining is a two step process which contains__ []
 A) Finding support and confidence.
 B) Finding all frequent itemsets.
 C) Generate strong association rules from the frequent itemsets.
 D) Both A & B. E. Both B & C.
7. All nonempty subset of a frequent itemset must also be frequent is _____ property.
8. Apriori method mines the frequent itemsets without candidate generation [T/F] []
9. For the given transactional data find the $\text{Support}(I1I2) =$ _____. []

ID	ITEMS
1	I1,I2,I4
2	I2,I4,I5
3	I1,I2
4	I1,I2,I3
5	I1,I2,I5

- A) 1 B) 2 C) 3 D) 4

10. How many number of scans were required in FP-Growth for finding frequent itemsets with 10 distinct items _____ . []

- A) 1 B) 2 C) 3 D) 100

11. The rules which involves items at different levels of abstraction are

- A) Multidimensional Association rules. B) Multilevel Association rules.
C) Rules interested at different levels. D) Predefined rules. []

12. For the given transactional database find the Support(AB)=_____

TID	A	B	C	D	E
1	1	1	1	0	1
2	0	1	0	1	1
3	1	1	1	0	1
4	0	1	0	1	0
5	1	1	0	1	1

- A) 1 B) 2 C) 3 D) 4 []

SECTION-B

SUBJECTIVE QUESTIONS

1. What is Association Rule Mining? Define Support and Confidence with example.
2. Generate frequent itemsets using the Apriori algorithm for the following data with the minimum support count 2.

TID	List of items_IDs
T100	I1,I2,I5
T200	I2,I4
T300	I2,I3
T400	I1,I2,I4
T500	I1,I3
T600	I2,I3
T700	I1,I3
T800	I1,I2,I3,I5
T900	I1,I2,I3

3. Explain how the association rules were generated from the frequent itemsets.
4. List and brief several methods to improve the efficiency of Apriori.
5. Find all frequent item sets using FP-Growth for the following data with $\min \text{sup} = 60\%$ and $\min \text{conf} = 80\%$.

TID	ITEMS_BOUGHT
T100	{K,A,D,B}
T200	{D,A,C,E,B}
T300	{C,A,B,E}
T400	{B,A,D}

6. Design the node structure for representing the FP-Tree.

UNIT -V

Classification

Objective:

- To gain knowledge on designing of Classification models to predict categorical class labels; and prediction models predict continuous valued functions.

Syllabus:

Basic Concepts, What is Classification, General approach to classification, decision tree induction, Attribute selection measures: Information gain, Bayes classification methods: Bayes' theorem, Naïve Bayesian classification.

Learning Outcomes:

At the end of the unit, students will be able to:

1. Understand the necessity of Classification and prediction models.
2. Implement classification techniques like decision tree induction, Bayesian classification.

Learning Material

5.1 Basic Concepts

Introduction

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.

Classification predicts categorical (discrete, unordered) labels, **Prediction** models continuous valued functions.

Ex: Build a classification model to **categorize bank loan applications** as either safe or risky.

Build prediction model to **predict the expenditures** in dollars of potential customers on computer equipment given their income and occupation.

Classification Techniques :

Basic classification techniques are decision tree classifiers, Bayesian classifiers, Bayesian belief networks, and rule based classifiers, Back propagation etc.,

Methods for prediction:

Linear regression, nonlinear regression etc.,

Applications :

- Detecting spam email messages.
- Target marketing
- Medical diagnosis
- Credit approval

Supervised learning:

In which the class label of each training tuple is known, and the number or set of classes to be learned known in advance.

Un Supervised learning:

In which the class label of each training tuple is not known, and the number or set of classes to be learned may not be known in advance.

Classification is supervised learning (i.e., the learning of the classifier is “supervised” in that it is told to which class each training tuple belongs)

Training data :

Consisting of records or tuples whose class labels are known must be provided. The training set is used to build a classification model

Test data:

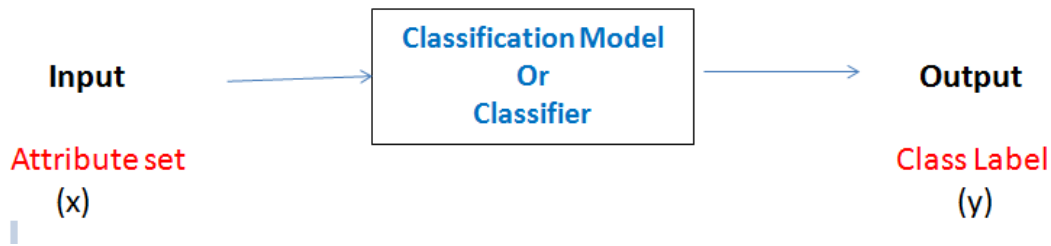
Consisting of records with unknown class labels.

Class label attribute:

The class label attribute is discrete-valued and unordered. It is *categorical in that each value serves as a category or class.*

5.1.1. What Is Classification?

Classification is one form of data analysis, 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.* **i.e.** classifying future or unknown objects



Model representation:

classification rules, decision trees, or mathematical formulae

5.1.2. General Approach to Classification

Data classification is a two-step process.

First step:(Learning)

A classifier is built describing a predetermined set of data classes or concepts. This is the **learning step** (or training phase), where a classification algorithm builds the classifier by analyzing or “learning from” a **training set**.

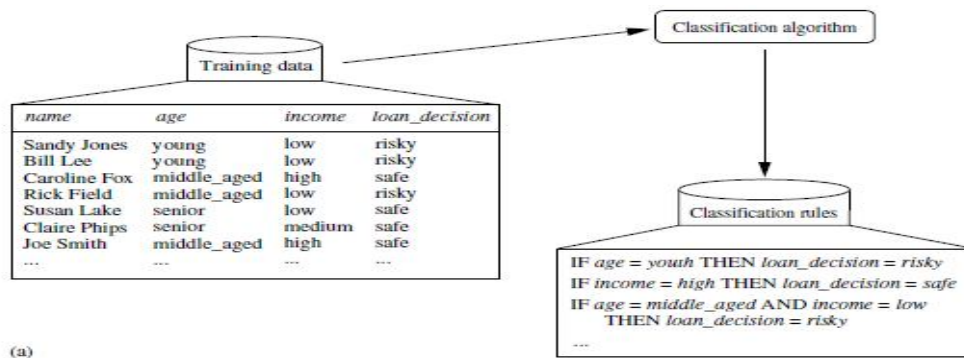
Second step :(Classification)

The model is used for classification. Use the **test set** to measure the accuracy of the classifier. If the accuracy of the classifier is considered acceptable, the classifier can be used to classify future data tuples for which the class label is not known.

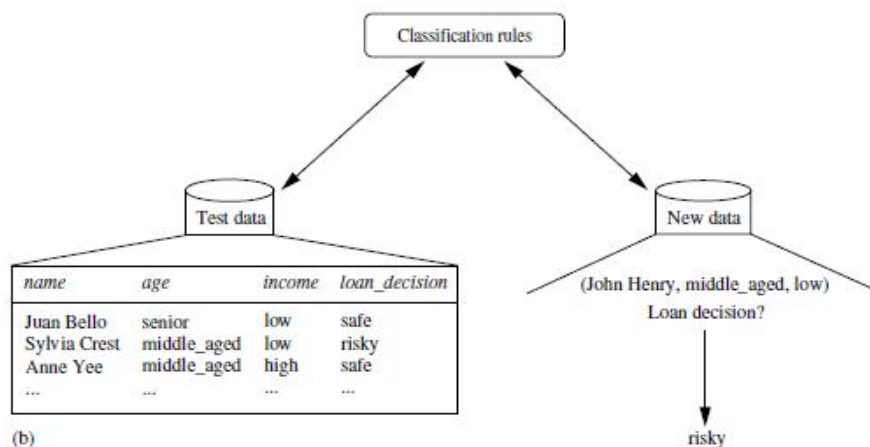
- ❖ For finding the accuracy of the model test data is used, It is made up of test tuples and their associated class labels. They are independent of the training tuples.
- ❖ The accuracy of a classifier on a given test set is the percentage of test set tuples that are correctly classified by the classifier.
- ❖ The associated class label of each test tuple is compared with the learned classifier’s class prediction for that tuple.

- ❖ If the accuracy of the classifier is considered acceptable, the classifier can be used to classify future data tuples for which the class label is not known.

Ex :A bank loans officer needs analysis of her data in order to learn which loan applicants are “safe”and whichare “risky” for the bank.



(a) **Learning:** Training data are analyzed by a classification algorithm. Here, the class label attribute is loan decision, and the learned model or classifier is represented in the form of classification rules



(b) **Classification:** Test data are used to estimate the accuracy of the classification rules. If the accuracy is considered acceptable, the rules can be applied to the classification of new data tuples.

5.2. Decision Tree Induction

- A decision tree is a flowchart-like tree structure, where each internal node (nonleaf node) denotes a **test** on an attribute, each branch represents an **outcome** of the test, and leaf nodes (or terminal node) represent **classes** or class distribution holds a class label. The topmost node in a tree is the root node.

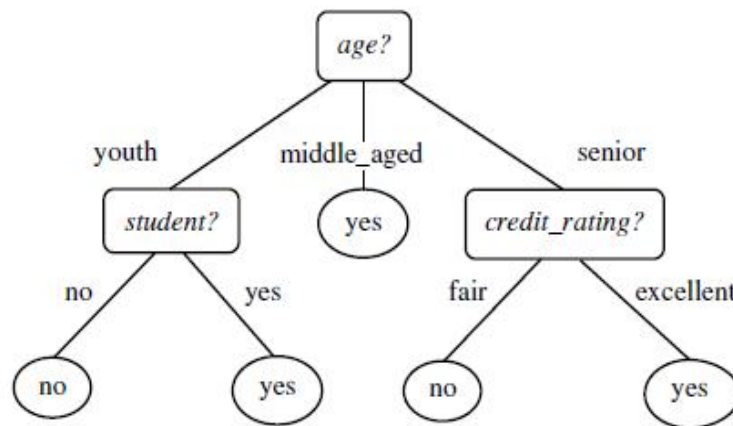


Fig : A decision tree for the concept *buys computer*, indicating whether a customer at AllElectronics is likely to purchase a computer. Each internal (nonleaf) node represents a test on an attribute. Each leaf node represents a class (either *buys computer* = *yes* or *buys computer* = *no*).

- 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 non binary trees.

Decision trees used for classification to classify an unknown sample i.e whose class label is unknown.

- 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.
- The construction of decision tree classifiers does not require any domain knowledge or parameter setting, and therefore is appropriate for exploratory knowledge discovery.
- Decision trees can handle high dimensional data.
- Decision tree induction algorithms have been used for classification in many application areas, such as medicine, manufacturing and production, financial analysis, astronomy, and molecular biology.

Algorithm: Generate_ decision_tree. Generate a decision tree from the given training data.

Input: The training samples, samples, represented by discrete-valued attributes; the set of candidate attributes , attribute_list.

Output: A decision tree.

Method:

- (1) create a node N ;
- (2) If samples 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 most common class in samples.
- (6) Select test-attribute, the attribute among attribute_list with the highest information gain.
- (7) label node N with test-attribute;
- (8) for each known value a_i of test-attribute
- (9) grow a branch from node N for the condition test-attribute = a_i ;
- (10) let s_i be the set of samples in samples for which test-attribute = a_i ;
- (11) If s_i is empty then
- (12) Attach a leaf labeled with the most common class in samples;
- (13) else attach the node returned by Generate_decision_tree(s_i , attribute-list-test-attribute);

5.2.1 Decision tree induction

- Decision tree induction is the learning of decision trees from class-labeled training tuples.
- The basic algorithm for decision tree induction is a greedy algorithm that constructs decision trees in a top-down recursive divide-and-conquer manner.
- A well-known decision tree induction algorithm is ID3(Iterative Dichotomiser).

The basic strategy for decision tree induction is

- The tree starts as a single node, N, representing the training samples(**step 1**)

- If the samples all of the same class, then the node becomes a leaf and is labeled with that class (**steps 2 and 3**).
- Otherwise, the algorithm uses an entropy-based measure known as information gain a heuristic for selecting the attribute that will best separate the samples in to individual classes (**step 6**). This attribute becomes the “test” or “decision” attribute at the node (**step 7**). All attributes are categorical, that is discrete-valued. Continuous-valued attributes must be discretized.
- A branch is created for each known value of the test attribute, and the samples are partitioned accordingly (**steps 8-10**)
- The algorithm uses the same process recursively to form a decision tree for the samples at each partition. Once an attribute has occurred at a node, it need not be considered in any of the node’s descendents(**step 13**)
- The recursive partitioning stops only when any one of the following conditions is true:
 - a) All samples for a given node belong to the same class (**steps 2 and 3**), or
 - b) There are no remaining attributes on which the samples may be further partitioned (**step 4**). In this case, majority voting is employed (**step 5**). This involves converting the given node into a leaf and labeling it with the class in majority among samples. Alternatively, the class distribution of the node samples may be stored.

- c) There are no sample for the branch test-attribute = a_i (**step 11**). In this case, a leaf is created with the majority class in samples (**step 12**).

5.2.2. Attribute Selection Measure : Information Gain

- The information gain measure is used to select the test attribute at each node in the tree. Such a measure is referred to as an attribute selection measure or a measure of the goodness of split.
- The attribute with the highest information gain (or greatest entropy reduction) is chosen as the test attribute for the current node.
- Let **S** be a set consisting of **s** data samples.
- Suppose the class label attribute has 'm' distinct values defining **m** distinct classes, **C_i**(for **i=1,...m**).
- Let **s_i** be the number of samples of **S** in class **C_i**.
- The expected information needed to classify a given sample is given by

$$I(\mathbf{s}_1, \mathbf{s}_2, \dots, \mathbf{s}_m) = - \sum_{i=1}^m P_i \log_2 (P_i)$$

- **P_i** is the probability that an arbitrary sample belongs to class **C_i** and is estimated by **s_i/s**

Entropy and Information Gain

S contains s_i tuples of class C_i for $i = \{1, \dots, m\}$

Information measures info required to classify any arbitrary tuple

$$I(s_1, s_2, \dots, s_m) = - \sum_{i=1}^m \frac{s_i}{S} \log_2 \frac{s_i}{S}$$

Entropy (weighted average) of attribute A with values $\{a_1, a_2, \dots, a_v\}$

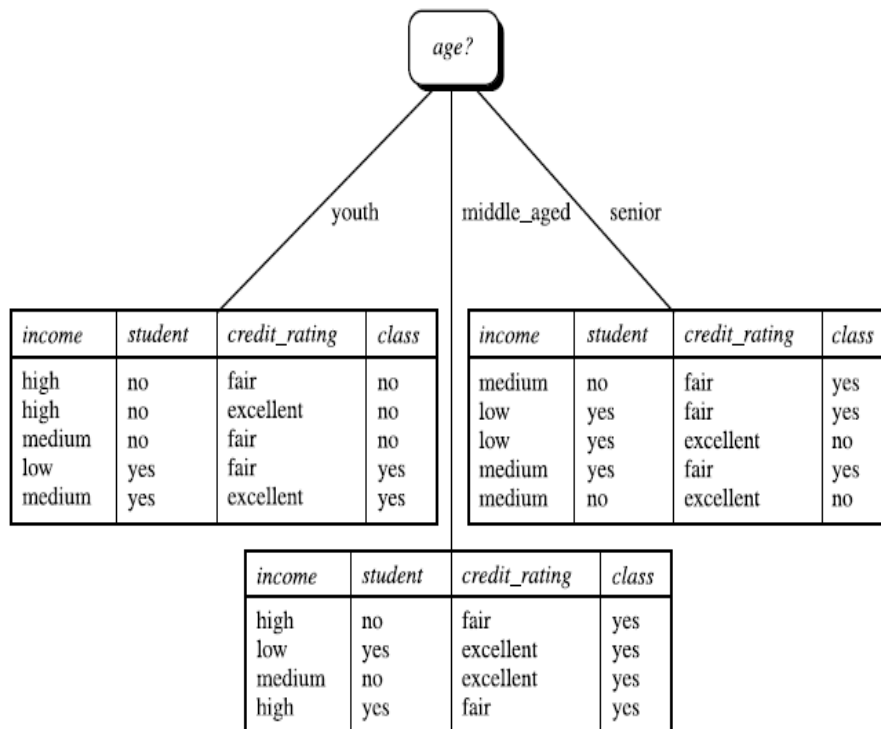
$$E(A) = \sum_{j=1}^v \frac{s_{1j} + \dots + s_{mj}}{S} I(s_{1j}, \dots, s_{mj})$$

Information gained by branching on attribute A

$$\text{Gain}(A) = I(s_1, s_2, \dots, s_m) - E(A)$$

Class-Labeled Training Tuples from the *AllElectronics* Customer Database

<i>RID</i>	<i>age</i>	<i>income</i>	<i>student</i>	<i>credit_rating</i>	<i>Class: buys_computer</i>
1	youth	high	no	fair	no
2	youth	high	no	excellent	no
3	middle_aged	high	no	fair	yes
4	senior	medium	no	fair	yes
5	senior	low	yes	fair	yes
6	senior	low	yes	excellent	no
7	middle_aged	low	yes	excellent	yes
8	youth	medium	no	fair	no
9	youth	low	yes	fair	yes
10	senior	medium	yes	fair	yes
11	youth	medium	yes	excellent	yes
12	middle_aged	medium	no	excellent	yes
13	middle_aged	high	yes	fair	yes
14	senior	medium	no	excellent	no



The attribute *age* has the highest information gain and therefore becomes the splitting attribute at the root node of the decision tree. Branches are grown for each outcome of *age*. The tuples are shown partitioned accordingly.

5.3. Bayes Classification methods

- 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
- Simple Bayesian classifier known as the naïve Bayesian classifier to be comparable in performance with decision tree and 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**

5.3.1. Bayes' Theorem

Let X be a data sample whose class label is unknown. Let H be some hypothesis, such as that the data sample X belongs to a specified class C .

For classification problems, determine $P(H/X)$, the probability that the hypothesis H holds given the observed data sample X

$P(H/X)$ is the **posterior probability**, of H conditioned on X .

Ex : Suppose the world of data samples consists of fruits, described by their color and shape.

Suppose that X is red and round, and that H is the hypothesis that X is an apple.

Then $P(H/X)$ reflects our confidence that X is an apple given that we have seen that X is red and round.

$P(H)$ is the **prior probability**, of H .

Ex: This is the probability that any given data sample is an apple, regardless of how the data sample looks.

- The posterior probability, $P(H/X)$ is based on more information (such as background knowledge) than the prior probability, $P(H)$, which is independent of X
- $P(X/H)$ is the posterior probability of X conditioned on H .

i.e It is the probability that X is red and round given that we know that it is true that X is an apple.

- $P(X)$ is the prior probability of X . it is the probability that a data sample from our set of fruits is red and round.
- $P(X)$, $P(H)$, and $P(X|H)$ may be estimated from the given data. Bayes theorem is useful in that it provides a way of calculating the posterior probability, $P(H/X)$ from $P(H)$, $P(X)$, and $P(X/H)$.
- **Bayes theorem**

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

5.3.2. 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. each tuple is represented by an n -dimensional attribute vector, $X = (x_1, x_2, \dots, x_n)$, depicting n measurements made on the tuple from n attributes, respectively, A_1, A_2, \dots, A_n .
2. Suppose that there are m classes, C_1, C_2, \dots, C_m . 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 C_i if and only if

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

Thus we maximize $P(C_i|X)$. The class C_i for which $P(C_i|X)$ is maximized is called the maximum posteriori hypothesis. By Bayes' theorem

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

3. As $P(X)$ is constant for all classes, only $P(X_j|C_i)P(C_i)$ need be maximized.
4. Given data sets with many attributes, it would be extremely computationally expensive to compute $P(X_j|C_i)$. In order to reduce computation in evaluating $P(X_j|C_i)$, the naive assumption of **class conditional independence** i.e., that there are no dependence relationships among the attributes).

$$\begin{aligned} P(X|C_i) &= \prod_{k=1}^n P(x_k|C_i) \\ &= P(x_1|C_i) \times P(x_2|C_i) \times \dots \times P(x_n|C_i). \end{aligned}$$

We can easily estimate the probabilities $P(x_1/C_i)$, $P(x_2/C_i)$, \dots , $P(x_n/C_i)$ from the training tuples. Here x_k refers to the value of attribute A_k for tuple X .

To compute $P(X/C_i)$, we consider the following:

- a) If A_k is categorical, then $P(x_k/C_i)$ is the number of tuples of class C_i in D having the value x_k for A_k , divided by, the number of tuples of class C_i in D .
- b) If A_k is continuous-valued, then the attribute is typically assumed to have a Gaussian distribution

$$g(x, \mu, \sigma) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(x-\mu)^2}{2\sigma^2}},$$

$$P(x_k|C_i) = g(x_k, \mu_{C_i}, \sigma_{C_i}).$$

5. To classify an unknown sample X , $P(X/C_i)P(C_i)$ is evaluated for each class C_i .

The classifier predicts that the class label of tuple X is the class C_i if and only if

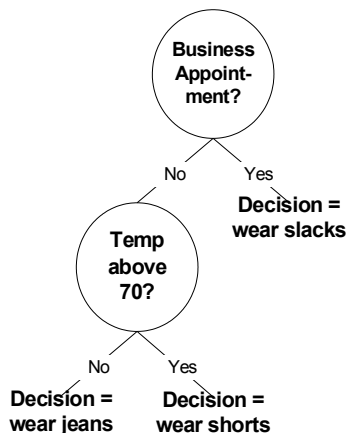
$$P(X/C_i)P(C_i) > P(X/C_j)P(C_j) \text{ for } 1 \leq j \leq m; j \neq i$$

In other words, it is assigned to the class C_i for which $P(X/C_i)P(C_i)$ is the maximum;

UNIT-V
Assignment-Cum-Tutorial Questions
SECTION-A

Objective Questions

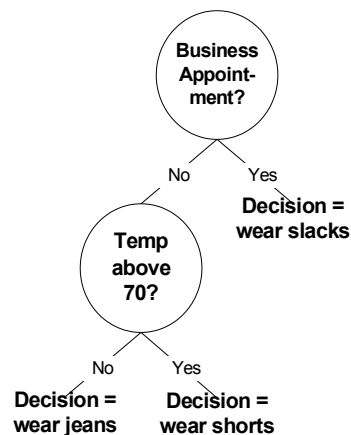
1. Data Classification process involves _____, _____.
2. Classification is a supervised learning. [T/F]
3. _____ measure is used to select the test attribute at each node in the decision tree. []
 A) Information Gain. B) Attribute Selection.
 C) Measure of the goodness of split. D) All of the above
4. Posterior probability can be calculated by _____ theorem. []
 A) Bayes. B) Apriori. C) Entropy. D) All
5. The neural network learns By adjusting the _____. []
 A) Heights. B) Weights. C) Depths. D) All
6. The process of forming general concept definitions from examples of concepts to be learned. []
 A) Deduction. B) Disjunction. C) Induction. D) Conjunction.
7. Data used to build a data mining model. []
 A) Validation Data. B) Hidden Data. C) Test Data. D) Training Data.
8. Which of the following is a valid production rule for the decision tree below?



- A) IF Business Appointment = No & Temp above 70 = No
THEN Decision = wear slacks
- B) IF Business Appointment = Yes & Temp above 70 = Yes
THEN Decision = wear shorts
- C) IF Temp above 70 = No
THEN Decision = wear shorts
- D) IF Business Appointment= No & Temp above 70 = No
THEN Decision = wear jeans

[]

9. Which of the following is a valid production rule for the decision tree below?



- A) IF Business Appointment = No & Temp above 70 = yes
THEN Decision = wear shorts.
- B) IF Business Appointment = Yes & Temp above 70 = Yes
THEN Decision = wear shorts
- C) IF Temp above 70 = No
THEN Decision = wear shorts
- D) IF Business Appointment= No & Temp above 70 = No

THEN Decision = wear slack. []

10. Decision tree is a type of _____ algorithm. []

A) Brute force approach. B) Randomized . C) Greedy D) None

SECTION-B

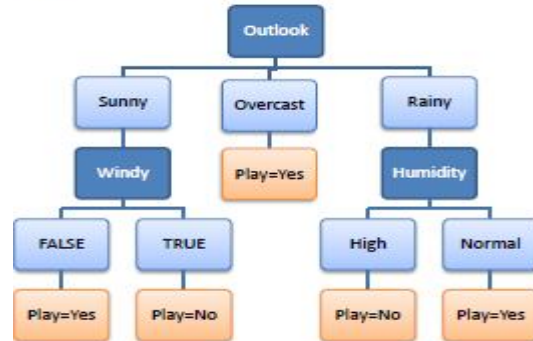
SUBJECTIVE QUESTIONS

1. With a neat diagram explain Data Classification Process.
2. Elaborate the issues regarding Classification and Prediction.
3. Illustrate the process of classification by Decision Tree Induction.
4. Build a decision tree for the concept buys_computer using the below database.

Class-labeled training tuples from the *AllElectronics* customer database.

<i>RID</i>	<i>age</i>	<i>income</i>	<i>student</i>	<i>credit_rating</i>	<i>Class: buys_computer</i>
1	youth	high	no	fair	no
2	youth	high	no	excellent	no
3	middle_aged	high	no	fair	yes
4	senior	medium	no	fair	yes
5	senior	low	yes	fair	yes
6	senior	low	yes	excellent	no
7	middle_aged	low	yes	excellent	yes
8	youth	medium	no	fair	no
9	youth	low	yes	fair	yes
10	senior	medium	yes	fair	yes
11	youth	medium	yes	excellent	yes
12	middle_aged	medium	no	excellent	yes
13	middle_aged	high	yes	fair	yes
14	senior	medium	no	excellent	no

5. What is the need for Tree Pruning?
6. Describe how classification rules are extracted from the decision tree with the following example.



7. Briefly explain about Bayesian classification.