



ALAGAPPA UNIVERSITY

[Accredited with A+⁺ Grade by NAAC (CGPA:3.64) in the Third Cycle
and Graded as Category-I University by MHRD-UGC]

KARAIKUDI – 630 003



DIRECTORATE OF DISTANCE EDUCATION

Master of Computer Applications

31535

DATA MINING AND WAREHOUSING

III - Semester



ALAGAPPA UNIVERSITY

[Accredited with 'A+' Grade by NAAC (CGPA:3.64) in the Third Cycle
and Graded as Category-I University by MHRD-UGC]

(A State University Established by the Government of Tamil Nadu)

KARAIKUDI – 630 003



Directorate of Distance Education

Master of Computer Applications

III - Semester

31535

DATA MINING AND WAREHOUSING

Authors

Prof. Subhash Krishna Chitale, *Faculty Member, IMS Engineering College, Ghaziabad*
Ms. Chandra Pushpanjali Patel, *Faculty Member, IMS Engineering College, Ghaziabad*
Units (1.0-1.2, 1.5-1.9, 2, 4.0-4.2, 4.4-4.8, 6.0-6.3, 6.5-6.9, 10.0-10.3, 10.6-10.11)

Dr. Richa Bhargava, *Adjunct Faculty, ICFAI Business School (IBS), Gurgaon*
Units (1.3-1.4, 4.3, 10.4-10.5, 12-14)

Dr. Gauri Jain, *Adjunct Faculty in IBS, Gurgaon*
Units (3, 5, 8)

Evelyn Learning Systems (P) Ltd.
Units (4.2.1, 6.4)

Shelly Gupta, *Assistant Professor, Inderprastha Engineering College, Ghaziabad*
Units (7, 9.2.5-9.8, 11)

Dr. Kuldeep Singh Kaswan, *Associate Professor, School of Computing Science & Engineering, Galgotias University, Greater Noida, U.P.*

Dr. Shreddha Sagar, *Associate Professor, School of Computing Science & Engineering, Galgotias University, Greater Noida, U.P.*
Unit (9.0-9.2.4)

"The copyright shall be vested with Alagappa University"

All rights reserved. No part of this publication which is material protected by this copyright notice may be reproduced or transmitted or utilized or stored in any form or by any means now known or hereinafter invented, electronic, digital or mechanical, including photocopying, scanning, recording or by any information storage or retrieval system, without prior written permission from the Alagappa University, Karaikudi, Tamil Nadu.

Information contained in this book has been published by VIKAS® Publishing House Pvt. Ltd. and has been obtained by its Authors from sources believed to be reliable and are correct to the best of their knowledge. However, the Alagappa University, Publisher and its Authors shall in no event be liable for any errors, omissions or damages arising out of use of this information and specifically disclaim any implied warranties or merchantability or fitness for any particular use.



Vikas® is the registered trademark of Vikas® Publishing House Pvt. Ltd.

VIKAS® PUBLISHING HOUSE PVT. LTD.

E-28, Sector-8, Noida - 201301 (UP)

Phone: 0120-4078900 • Fax: 0120-4078999

Regd. Office: 7361, Ravindra Mansion, Ram Nagar, New Delhi 110 055

• Website: www.vikaspublishing.com • Email: helpline@vikaspublishing.com

Work Order No. AU/DDE/DE1-291/Preparation and Printing of Course Materials/2018 Dated 19.11.2018 Copies - 500

SYLLABI-BOOK MAPPING TABLE

Data Mining and Warehousing

Syllabi	Mapping in Book
BLOCK - I: INTRODUCTION TO DATA MINING AND WAREHOUSING Unit 1: Data Warehousing Introduction - Definition - Architecture - Warehouse Schema - Warehouse Server - OLAP Operations. Data Warehouse Technology - Hardware and Operating System Unit 2: Data Mining - Definition - DM Techniques - Current Trends in Data Mining - Different forms of Knowledge - Data Selection, Cleaning, Integration, Transformation, Reduction and Enrichment. Unit 3: Data: Types of Data - Data Quality - Data Preprocessing - Measures of Similarity and Dissimilarity. Exploration: Summary Statistics - Visualization.	Unit 1: Introduction to Data Warehousing (Pages 1-23) Unit 2: Data Mining (Pages 24-74) Unit 3: Data (Pages 75-90)
BLOCK - II: ASSOCIATION RULE MINING AND CLASSIFICATION Unit 4: Association Rules: Introduction - Methods to Discover Association Rule - Apriori Algorithm Partition Algorithm Unit 5: AR Algorithms: Pincher Search Algorithm - Dynamic Item Set Algorithm - FP Tree Growth Algorithm. Unit 6: Classification: Decision Tree Classification - Bayesian Classification - Classification by Back Propagation.	Unit 4: Association Rules (Pages 91-117) Unit 5: AR Algorithms (Pages 118-131) Unit 6: Classification (Pages 132-154)
BLOCK - III: CLUSTERING TECHNIQUES AND MACHINE LEARNING Unit 7: Introduction: - Clustering Paradigms - Partitioning Algorithms - K Means & K Medoid Algorithms - CLARA - CLARANS - Hierarchical Clustering - DBSCAN - BIRCH - Categorical Clustering Algorithms - STIRR - ROCK - CACTUS. Unit 8: Introduction to Machine Learning: - Supervised Learning - Unsupervised Learning - Machine Learning and Data Mining. Unit 9: Neural Networks: Introduction - Use of NN - Working of NN Genetic Algorithm: Introduction - Working of GA.	Unit 7: Introduction to Clustering (Pages 155-173) Unit 8: Introduction to Machine Learning (Pages 174-186) Unit 9: Neural Networks (Pages 187-198)
BLOCK - IV: WEB MINING AND VISUAL DATA MINING Unit 10: Introduction: Web Content Mining - Web Structure Mining - Web Usage Mining - Text Mining - Text Clustering, Temporal Mining - Spatial Mining Unit 11: Visual Data Mining: Knowledge Mining - Various Tools and Techniques for Implementation using Weka, Rapidminer and MATLAB.	Unit 10: Introduction to Web Mining (Pages 199-227) Unit 11: Visual Data Mining (Pages 228-240)
BLOCK - V : INTRODUCTION TO BIG DATA ANALYTICS Unit 12: Big Data Characteristics: Types of Big Data - Traditional Versus Big Data Approach Unit 13: Technologies: Available for Big Data Unit 14: Hadoop: Introduction - What is Hadoop? - Core Hadoop Components - Hadoop Ecosystem - Physical Architecture - Hadoop Limitations	Unit 12: Big Data Characteristics (Pages 241-2252) Unit 13: Technologies (Pages 253-258) Unit 14: Hadoop (Pages 259-284)

CONTENTS

INTRODUCTION

BLOCK I: INTRODUCTION TO DATA MINING AND WAREHOUSING

UNIT 1 INTRODUCTION TO DATA WAREHOUSING 1-23

- 1.0 Introduction
- 1.1 Objectives
- 1.2 Data Warehousing (DWH)
 - 1.2.1 Schemas; 1.2.2 Data Warehouse Architecture
 - 1.2.3 ROLAP, MOLAP and HOLAP; 1.2.4 OLTP and OLAP
- 1.3 Warehouse Server – OLAP Operations
- 1.4 Data Warehouse Technology-Hardware and Operating System
- 1.5 Answers to Check Your Progress Questions
- 1.6 Summary
- 1.7 Key Words
- 1.8 Self Assessment Questions and Exercises
- 1.9 Further Readings

UNIT 2 DATA MINING 24-74

- 2.0 Introduction
- 2.1 Objectives
- 2.2 Data Mining
 - 2.2.1 Data Mining and Knowledge Discovery
- 2.3 Data Mining Functionalities and Techniques
- 2.4 Data Mining Task Primitives
- 2.5 Integration of Data Mining Systems
- 2.6 Data Reduction
- 2.7 Data Mining Applications
- 2.8 Trends in Data Mining
- 2.9 Answers to Check Your Progress Questions
- 2.10 Summary
- 2.11 Key Words
- 2.12 Self Assessment Questions and Exercises
- 2.13 Further Readings

UNIT 3 DATA 75-90

- 3.0 Introduction
- 3.1 Objectives
- 3.2 Types of Data
- 3.3 Need of Preprocessing and Data Quality
 - 3.3.1 Data Preprocessing
- 3.4 Measure of Similarity and Dissimilarity
- 3.5 Exploration: Summary Statistics - Visualization
- 3.6 Answers to Check Your Progress Questions
- 3.7 Summary
- 3.8 Key Words
- 3.9 Self Assessment Questions and Exercises
- 3.10 Further Readings

BLOCK II: ASSOCIATION RULE MINING AND CLASSIFICATION

UNIT 4 ASSOCIATION RULES 91-117

- 4.0 Introduction
- 4.1 Objectives
- 4.2 Association Rules and Apriori Algorithm
 - 4.2.1 Association Rules from Relational Databases and Data Warehouses
- 4.3 Partition Algorithm
- 4.4 Answers to Check Your Progress Questions
- 4.5 Summary
- 4.6 Key Words
- 4.7 Self Assessment Questions and Exercises
- 4.8 Further Readings

UNIT 5 AR ALGORITHMS 118-131

- 5.0 Introduction
- 5.1 Objectives
- 5.2 Pincer Search Algorithm
- 5.3 Dynamic Item Set Algorithm
- 5.4 FP Tree Growth Algorithm
- 5.5 Answers to Check Your Progress Questions
- 5.6 Summary
- 5.7 Key Words
- 5.8 Self Assessment Questions and Exercises
- 5.9 Further Readings

UNIT 6 CLASSIFICATION 132-154

- 6.0 Introduction
- 6.1 Objectives
- 6.2 Decision Tree Classification
- 6.3 Bayesian Classification
- 6.4 Classification by Back Propagation
- 6.5 Answers to Check Your Progress Questions
- 6.6 Summary
- 6.7 Key Words
- 6.8 Self Assessment Questions and Exercises
- 6.9 Further Readings

BLOCK III: CLUSTERING TECHNIQUES AND MACHINE LEARNING

UNIT 7 INTRODUCTION TO CLUSTERING 155-173

- 7.0 Introduction
- 7.1 Objectives
- 7.2 Clustering: Definition
- 7.3 Clustering Methods and Algorithms
 - 7.3.1 Partitioning Methods; 7.3.2 Hierarchical Methods
 - 7.3.3 Density-Based Methods; 7.3.4 Grid-Based Methods
 - 7.3.5 K-means; 7.3.6 K-Medoid; 7.3.7 CLARA (Clustering for Large Applications)
 - 7.3.8 CLARANS (Clustering Large Application Based on Randomized Search)
 - 7.3.9 BIRCH (Balanced Iterative Reducing and Clustering using Hierarchies)
 - 7.3.10 CACTUS – Clustering Categorical Data Using Summaries
 - 7.3.11 ROCK - Robust Clustering Algorithm for Categorical Attributes
 - 7.3.12 DBSCAN (Density Based SCAN Clustering Algorithm)
 - 7.3.13 STIRR (Sieving Through Iterated Relational Reinforcement)

- 7.4 Answers to Check Your Progress Questions
- 7.5 Summary
- 7.6 Key Words
- 7.7 Self Assessment Questions and Exercises
- 7.8 Further Readings

UNIT 8 INTRODUCTION TO MACHINE LEARNING **174-186**

- 8.0 Introduction
- 8.1 Objectives
- 8.2 Machine Learning: An Overview
 - 8.2.1 Steps Involved in a Machine Learning Process
 - 8.2.2 Types of Machine Learning
 - 8.2.3 Machine Learning Applications
- 8.3 Machine Learning and Data Mining
- 8.4 Answers to Check Your Progress Questions
- 8.5 Summary
- 8.6 Key Words
- 8.7 Self Assessment Questions and Exercises
- 8.8 Further Readings

UNIT 9 NEURAL NETWORKS **187-198**

- 9.0 Introduction
- 9.1 Objectives
- 9.2 Introduction to Neural Systems and Networks
 - 9.2.1 A Brief History of ANN; 9.2.2 Biological Neuron
 - 9.2.3 ANN versus BNN; 9.2.4 Model of Artificial Neural Network
 - 9.2.5 Uses of Neural networks
- 9.3 Genetic Algorithm
 - 9.3.1 GA Operators
 - 9.3.2 Genetic Algorithm Framework
- 9.4 Answers to Check Your Progress Questions
- 9.5 Summary
- 9.6 Key Words
- 9.7 Self Assessment Questions and Exercises
- 9.8 Further Readings

BLOCK IV: WEB MINING AND VISUAL DATA MINING

UNIT 10 INTRODUCTION TO WEB MINING **199-227**

- 10.0 Introduction
- 10.1 Objectives
- 10.2 Web Content Mining
 - 10.2.1 Mining the Web Page Layout Structure
 - 10.2.2 Web Page Mining
- 10.3 Text Mining
- 10.4 Text Clustering
- 10.5 Temporal Mining
- 10.6 Spatial Data Mining
- 10.7 Answers to Check Your Progress Questions
- 10.8 Summary
- 10.9 Key Words
- 10.10 Self Assessment Questions and Exercises
- 10.11 Further Readings

UNIT 11 VISUAL DATA MINING **228-240**

- 11.0 Introduction
- 11.1 Objectives
- 11.2 Data Visualization
- 11.3 Visual Data Mining
 - 11.3.1 Weka; 11.3.2 RapidMiner
 - 11.3.3 MATLAB (Matrix Laboratory)
 - 11.3.4 Visualization based Clustering in Weka- A Practical Approach
- 11.4 Answers to Check Your Progress Questions
- 11.5 Summary
- 11.6 Key Words
- 11.7 Self Assessment Questions and Exercises
- 11.8 Further Readings

BLOCK V: INTRODUCTION TO BIG DATA ANALYTICS

UNIT 12 BIG DATA CHARACTERISTICS **241-2252**

- 12.0 Introduction
- 12.1 Objectives
- 12.2 Introduction to Big Data Platform
 - 12.2.1 Nature of Data
- 12.3 Traditional vs Big Data Approach
- 12.4 Answers to Check Your Progress Questions
- 12.5 Summary
- 12.6 Key Words
- 12.7 Self Assessment Questions and Exercises
- 12.8 Further Readings

UNIT 13 TECHNOLOGIES **253-258**

- 13.0 Introduction
- 13.1 Objectives
- 13.2 Technologies Available for Big Data
- 13.3 Answers to Check Your Progress Questions
- 13.4 Summary
- 13.5 Key Words
- 13.6 Self Assessment Questions and Exercises
- 13.7 Further Readings

UNIT 14 HADOOP **259-284**

- 14.0 Introduction
- 14.1 Objectives
- 14.2 Introduction to Hadoop
- 14.3 Core Hadoop Components
- 14.4 Hadoop Ecosystem and Architecture
 - 14.4.1 Physical Architecture of Hadoop
- 14.5 Hadoop Limitations
- 14.6 Answers to Check Your Progress Questions
- 14.7 Summary
- 14.8 Key Words
- 14.9 Self Assessment Questions and Exercises
- 14.10 Further Readings

INTRODUCTION

NOTES

An organization generates large amount of data every year during the course of business. The organizational data can be used to generate useful information for furthering the business. The process of analysing data to arrive at patterns which in turn can be used to generate useful information is referred to as data mining.

The concept of data mining, even though new, is being put to use in a large number of organizations. This is because on an average, the data in an organization doubles in every three years. Data mining software is used by most companies to arrive at a relationship between internal factors, such as price and product-mix and between external factors such as economic indicators, competition and customer demographics.

Data warehousing is defined as the process of centralized data management and retrieval. Data warehouse is a repository of an organization's electronically stored data. It functions as the store house of data procured from various operational systems in use in the organization.

Data pre-processing is a data mining method that comprises converting raw data into a logical format. Real-world data is frequently inadequate, unreliable, and/or deficient in certain actions or drifts, and is expected to comprise numerous blunders. Data pre-processing is a verified method of determining such issues. Data pre-processing fixes raw data for added processing. Data pre-processing is used database-driven applications, such as customer relationship management and rule-based applications.

This book, *Data Mining and Warehousing*, follows the SIM format or the self-instructional mode wherein each unit begins with an 'Introduction' to the topic followed by an outline of the 'Objectives'. The detailed content is then presented in a simple and organized manner, interspersed with 'Check Your Progress' questions to test the understanding of the students. A 'Summary' along with a list of 'Key Words' and a set of 'Self Assessment Questions and Exercises' is also provided at the end of each unit for effective recapitulation.

BLOCK - I
INTRODUCTION TO DATA MINING
AND WAREHOUSING

*Introduction to Data
Warehousing*

NOTES

**UNIT 1 INTRODUCTION TO DATA
WAREHOUSING**

Structure

- 1.0 Introduction
- 1.1 Objectives
- 1.2 Data Warehousing (DWH)
 - 1.2.1 Schemas
 - 1.2.2 Data Warehouse Architecture
 - 1.2.3 ROLAP, MOLAP and HOLAP
 - 1.2.4 OLTP and OLAP
- 1.3 Warehouse Server – OLAP Operations
- 1.4 Data Warehouse Technology-Hardware and Operating System
- 1.5 Answers to Check Your Progress Questions
- 1.6 Summary
- 1.7 Key Words
- 1.8 Self Assessment Questions and Exercises
- 1.9 Further Readings

1.0 INTRODUCTION

In this unit, you will learn about the basic concepts of data warehouse and OLAP technology. A data warehouse is a subject oriented, integrated, non-volatile and time variant collection of data in support of management's decisions.' Subject oriented means that database is organized in a data warehouse in a subject-wise manner even at the expense of redundancy. Thus, every manager would have access to the desired information in the shortest possible time notwithstanding the extra space occupied by it. OnLine Analytical Processing (OLAP) is a methodology that provides end-users, fast access to large amounts of data in an intuitive manner for assisting deductions which is based on investigative reasoning.

1.1 OBJECTIVES

After going through this unit, you will be able to:

- Understand and explain data warehousing
- Discuss warehouse server and OLAP operations
- Explain the data warehouse technology, hardware and operating system

1.2 DATA WAREHOUSING (DWH)

NOTES

According to W.H. Inmon, known as the father of the data warehouse concept, 'A data warehouse is a subject oriented, integrated, non-volatile and time-variant collection of data in support of management's decisions.'

Subject oriented means that database is organized in a data warehouse in a subject-wise manner even at the expense of redundancy. Thus, every manager would have access to the desired information in the shortest possible time not withstanding the extra space occupied by it.

Integrated implies related database tables created in the form of fact and dimension tables that can be linked to each other and are not stored as standalone data resources.

Non-volatile means storage of data on a permanent, non-destructible basis. It can be purged or removed only as an exception for an organizational need.

To be time variant, it requires all data to be entered in the data warehouse to be time stamped or associated with its time of entry. The time element introduced may not be the actual time when data was entered in the operational system.

Figure 1.1 shows a data warehouse system.

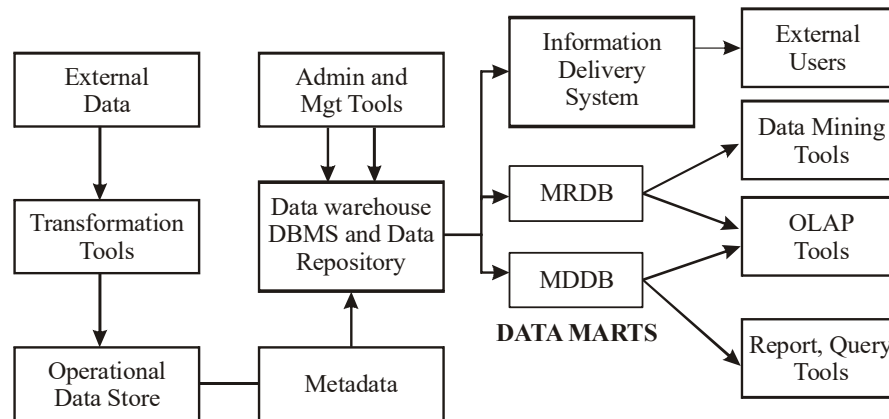


Fig. 1.1 Data Warehouse System

Data Warehouse Building Blocks

The following are the building blocks of data warehouse:

Data Pre-Processing Tools

Data pre-processing means sourcing, acquisition, cleaning and transformation of data prior to its entry into a data warehouse repository. Data is received from legacy systems, the Web or other external sources. The data and database from where it is received will be heterogeneous and requires:

- (i) Removing unwanted data.
- (ii) Converting to common data and definition names:
 - (a) Summarizing the data.
 - (b) Completing missing data.

NOTES

Operational Data Store

Data is transformed and loaded into the Operational Data Store (ODS) in real-time frame. From the ODS it is loaded into the data warehouse after extraction and cleaning operations at regular intervals, but not as and when received from external sources. As such a time of entry is attached with it. The data, thus, available is loaded under the control of metadata.

Metadata

Metadata is data about data and keeps information as:

- (i) **Technical Metadata:** It contains the sources of access data, data structure, transformation description, rules specified during data processing, access authorizations and backup history.
- (ii) **Business Metadata:** It contains information about the subject areas, information object types, the Internet home pages, details of information delivery system, that is, when to dispatch information and to whom, data warehouse operational information and ownership details.

Data Warehouse Database

It is the central database consisting of data warehouse RDBMS, a large repository and supporting databases such as multi-relational database, multidimensional database and data marts.

Data Mart

Data mart is another important component of data warehouse and is a data store that is subsidiary to a data warehouse. It is created to meet specific information needs of different functional area managers. Data marts are a part of the data warehouse database and cannot be taken as an alternative for data warehouse.

Management and Administration Tools

They are provided for:

- (i) Managing and updating metadata.
- (ii) Providing backup and recovery.
- (iii) Removing unwanted data.
- (iv) Providing security and assigning priorities.
- (v) Checking quality.
- (vi) Distributing data.

NOTES

Access Tools

They are categorized as:

- (i) **Query and Reporting Tools:** A querying and reporting tool helps you run regular reports, create organized listing and perform cross-tabular reporting and querying.
- (ii) **Application Tools:** To meet specific user requirements.
- (iii) **Data Mining Tools:** To discover knowledge, data visualization and for correcting data when the input data is incomplete.
- (iv) **OLAP Tools:** These are associated with multidimensional databases to provide elaborate, complex views for analysis.

Information Delivery System

It provides an external interface to provide data warehouse reports, information objects to external users as per a specified schedule.

Granularity of Data

Granularity of data refers to the level of detail or summarization at which data is stored in a data warehouse. Larger the granularity less will be the detail available for those data items and vice versa also holds good. A data warehouse manager is required to identify the granularity of data for his organization so that reports of the requisite detail are available.

An example is the maintenance of details of all calls made by a mobile user by the telecom operator to provide a high level of detail (low level of granularity) to meet legal requirements at a later stage.

Granular data offers the advantage of reusability of data by other users and also helps in optimizing the storage space.

Multidimensional Data Models and Schemas

Data warehouses and OLAP tools are based on what is known as a multidimensional model. Data is visualized as a data cube in such a model identified by fact and dimension tables.

Facts are the numerical measures of a central theme, for example a student. The measures may be Marks_obtained, Division scored.

Dimensions are the entities with respect to which an organization keeps its records; for example, teacher, subject, class, college, university, etc.

Concept Hierarchies

It is a method of defining a sequence of identifying levels for each entity, for example, city, district, state and country.

1.2.1 Schemas

While Entity-Relationship Model (E-R Model) was found adequate in the design of relational databases, a data warehouse requires a subject-oriented schema for better analysis and handling of more complex queries.

Three schemas are, therefore, created to meet the data warehouse requirements. These are as follows:

- (i) **Star Schema:** It is the most common model. In the star schema there is a large central fact table containing numerical data without duplication or redundancy. A large number of dimension tables are referred by it. Each of these handles a dimension. Figure 1.2 is an example of a star schema.

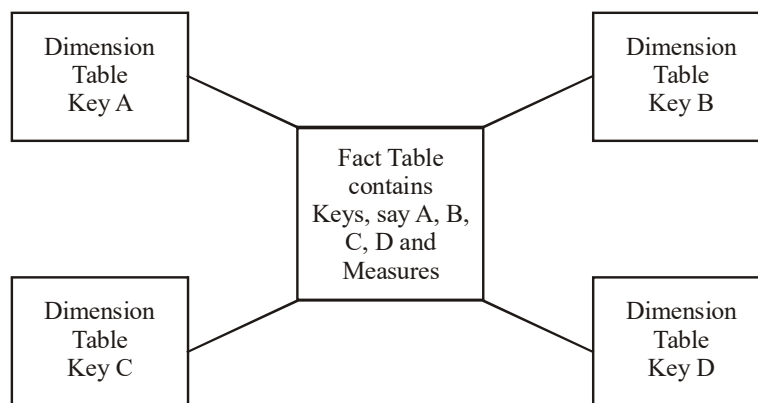


Fig. 1.2 An Example of a Star Schema

- (ii) **Snowflake Schema:** It is an extension of star schema. The dimension tables are further normalized and extra tables are added. Figure 1.3 shows an example of a snowflake schema.

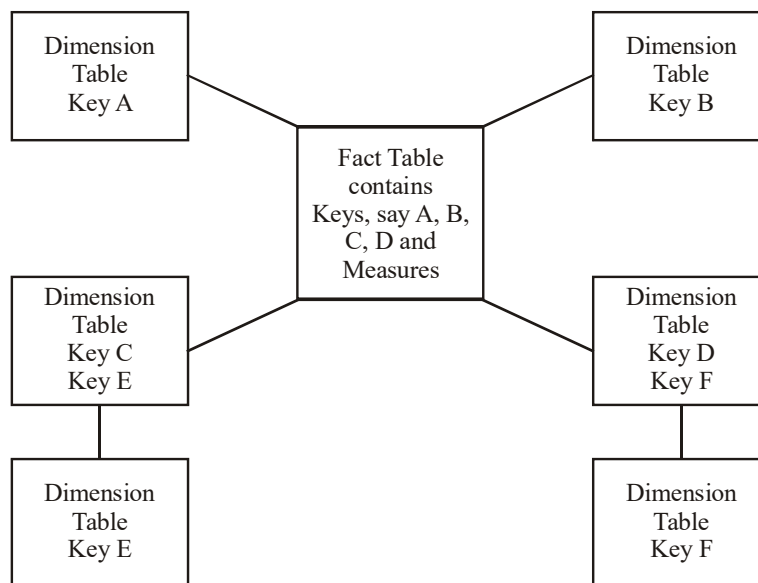


Fig. 1.3 An Example of a Snowflake Schema

NOTES

NOTES

- (iii) **Fact Constellation Schema:** It has multiple fact tables to meet the requirements of more advanced applications. Fact tables are permitted to share dimension tables. Figure 1.4 shows an example of a fact constellation schema.

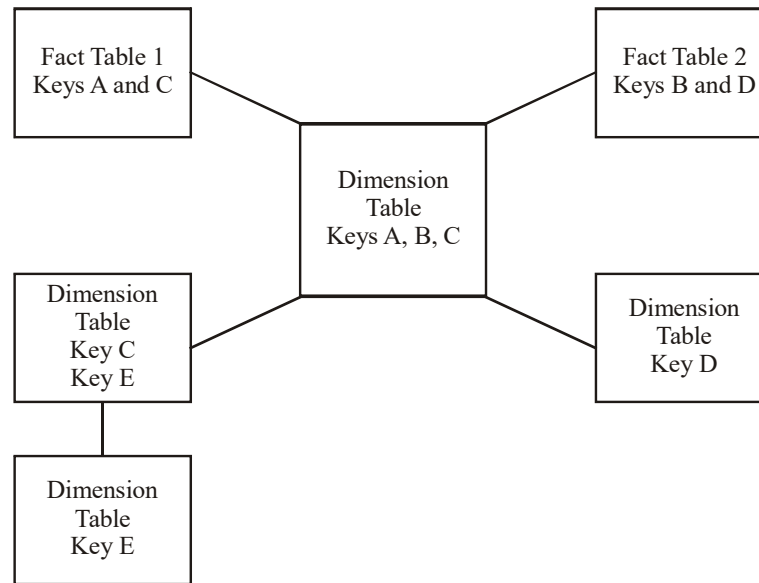


Fig. 1.4 An Example of a Fact Constellation Schema

Data Warehouse Design

A data warehouse design involves:

- (i) Choosing a business process to model, for example orders, invoices, shipments, etc.
- (ii) Choosing a DWH for a large organization while selecting a data mart for departmental implementation.
- (iii) Choosing the grain of the business—the fundamental, atomic level of data to be represented in the fact table.
- (iv) Choosing the dimensions to be applied to each fact table.
- (v) Choosing the measures that will populate each fact table record, for example, Units_sold, Rs_sold.

Based on these five principles, a nine-step method is evolved as follows:

- (i) Choosing the subject matter.
- (ii) Deciding what the fact table represents.
- (iii) Identifying and conforming the dimensions.

- (iv) Choosing the facts.
- (v) Storing pre-calculations in the fact table.
- (vi) Rounding out the dimension tables.
- (vii) Choosing the duration of the databases.
- (viii) Tracking the slowly changing dimensions.
- (ix) Deciding the query priorities.

NOTES

1.2.2 Data Warehouse Architecture

Data warehouse architecture is based on an RDBMS system server. It has a massive central repository for storage of data, subsidiary databases and front-end tools.

The architecture consists of:

- (i) **Bottom Tier:** An RDBMS and a DWH server
- (ii) **Middle Tier:** An OLAP server
- (iii) **Top Tier:** Front-end tools

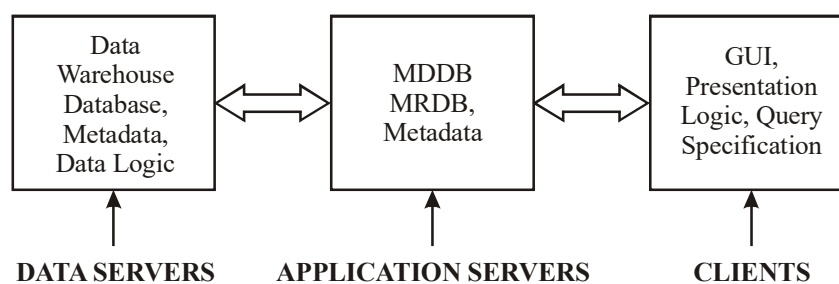


Fig. 1.5 Data Warehouse Architecture

Virtual Warehouse

Another commonly used term is the virtual server. It is a set of views over operational databases. For efficient query processing, some of the possible summary views are materialized. It is easy to build but requires excess capacity of operational database servers.

Developing a Data Warehouse

It involves:

- (i) Defining a high-level corporate data model.
- (ii) Developing an enterprise data warehouse and continuing to refine it to meet user requirements.
- (iii) Developing parallel data marts and refining these models.

1.2.3 ROLAP, MOLAP and HOLAP

These tools utilize specialized data structures to organize, navigate and analyse data, typically in an aggregated form. They require a tight coupling with the application and presentation layer.

NOTES

1. Multidimensional Online Analytical Processing or MOLAP architecture (see Figure 1.6) creates a data structure to store data in a way it will finally be utilized to enhance its performance. It is particularly well suited for iterative and time series analysis. It provides tools to access data maintained in the DWH repository (RDBMS) and permits its access when the MDDDB does not have the desired data. It is used for providing the user a high performance and better understanding through specialized indexing and storage optimizations. It requires less space due to usage of compression.

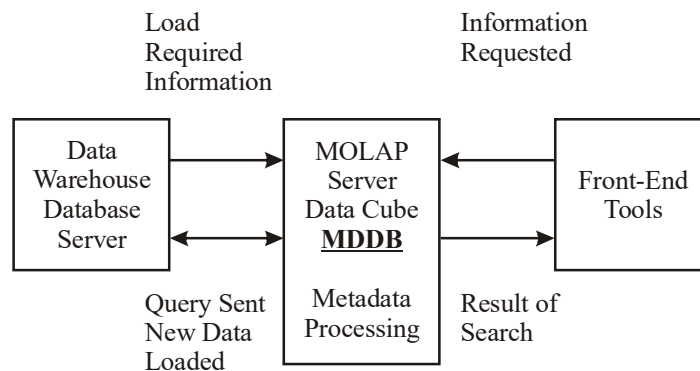


Fig. 1.6 MOLAP Architecture

2. Relational Online Analytical Processing or ROLAP works directly with RDBMS and is more scalable. It depends on databases for calculations, and therefore its performance suffers. ROLAP servers contain both numeric and textual data and serve broader needs. They support large databases supporting parallel processing, good security and employs known technologies. Figure 1.7 shows the architecture of ROLAP.

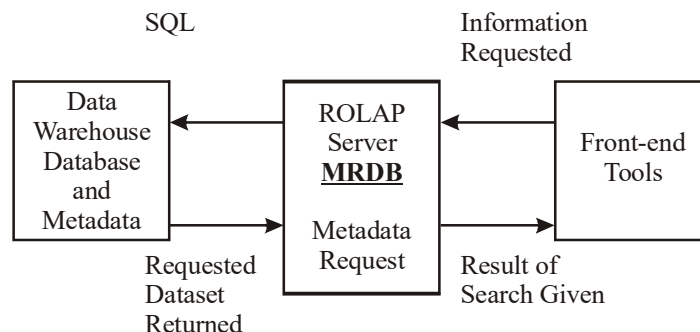


Fig. 1.7 ROLAP Architecture

3. Hybrid Online Analytical Processing or HOLAP uses the best features of both, i.e., the flexibility of ROLAP RDBMS and the optimized multidimensional structure of MOLAP. Users are provided the ability to perform limited analysis capability, either against RDBMS products or by introducing an intermediate MOLAP server. A user can send a query to select data from the DBMS which then delivers the requested data to the desktop where it is placed in a data cube. The desired information is maintained locally and need not be created each time a query is given.

The salient differences among MOLAP, ROLAP and HOLAP are as follows:

- (i) In MOLAP there is no query directly given by the user to the DWH server. The desired multidimensional data is positioned in the MOLAP server after an SQL sent by the MOLAP server is sent to the DWH server.
- (ii) The ROLAP server does not store the intermediate result in a cube but in a relational table. The user gets his query serviced by the ROLAP server.
- (iii) In HOLAP, an SQL is sent by the user to the DWH server. Then either the result is received by it directly or an intermediate MOLAP server data cube is created and accessed by the user.

NOTES

1.2.4 OLTP and OLAP

After understanding, the database requirements of OLTP and DSS, let us now differentiate the query systems associated with each. These are OLTP and OLAP. OLTP fulfils the requirements of OIS well, as the queries are simple in nature. OLAP, on the other hand, addresses the needs of defining more complex queries and requires novel databases in the form of Multi-Dimensional and Multi-Relational Databases (MDDDB and MRDB, respectively) to provide back-end support.

Table 1.1 shows the comparison of the features of OLAP and OLTP.

Table 1.1 Differences between OLAP and OLTP Features

<i>Feature</i>	<i>OLTP</i>	<i>OLAP</i>
1. Meant for	OIS	MIS/DSS
2. Purpose	Supports transaction	For analysis
3. End-user	Operations level, DB specialists	Knowledge worker
4. Function	Daily operations	Long-term needs
5. DB design	ER based, application oriented	Star/snowflake schemas, Subject oriented
6. Data	Current, up-to-date	Historical
7. Summarization	Primitive, highly detailed	Aggregated
8. View	Relational	Multidimensional, multi-relational

NOTES

9. Work unit	Short, simple transaction	Complex query
10. Access mode	Both read/write	Mostly read
11. Based on	Data inputs	Derived information
12. Operations	Operation on primary key	Multiple scans
13. Number of records accessed	Few	Many
14. Number of users	Large number	Selected
15. DB size	In MB/GB	In over 100GB to TB
16. Priority	High performance and availability	High flexibility, End-user autonomy
17. Measure	Transactions throughput	Specific query

Online Transaction Processing (OLTP)

Databases can be split into various categories on the basis of their application and requirements. The most popular method is OLTP which stands for Online Transaction Processing. Other methods include Decision Support Systems (DSS), Data Warehouses and Data Marts.

OLTP databases handle real-time transactions having some inherent and special requirements. A person managing a store should ensure efficient updating alongside purchases tables, and customer tables. OLTP databases should have atomicity in operation. Transactions must be atomic in nature which means that an entire transaction should be completed or aborted, and each data should be consistent, isolated and durable. All these factors are essential for running a successful OLTP database.

OLTP databases act like front line warriors. A database should be extremely robust and scalable to meet growing needs. An undersized database for DSS database might shift your schedule before time but due to an undersized database in OLTP you will loose customers. If OLTP database is unable to update within 15 seconds, order bookings cannot be done from an online book store.

OLTP have features of 'row-level locking'. A given record in a table may be locked from updates by other processes till transactions on that record is complete. It is similar to mutex locks in POSIX threading.

Online Analytical Processing (OLAP)

This is a methodology that provides end-users, fast access to large amounts of data in an intuitive manner for assisting deductions which is based on investigative reasoning.

OLAP Systems for Decision Support

People in IT sector face challenges in meeting the growing demand of delivering system for business owners to make decisions strategic as well as tactical based on corporate information. Decision support systems of this type come under OnLine Analytical Processing (OLAP) systems. They enable knowledge workers to

manipulate operational data fast with flexibility and use familiar business terms, providing analytical insight.

OLAP systems need to:

- (i) Provide support for the requirement of complex analysis for decision-makers.
- (ii) Analyse data from different angles.
- (iii) Provide support for complex analysis against large data set as input.

Dr E.F. Codd's Guidelines for OLAP

OLAP is an essential ingredient of data mining. It is, therefore, essential to understand the relevance of Dr E.F. Codd's (a well-known authority on Relational Database Management System (RDBMS)) guidelines. The following is an interpretation of each of them, relating them to the issues involved:

- (i) **Multidimensional Conceptual View:** Business problems are complex and can be solved only through a multidimensional concept as normal queries cannot address them effectively. As such, multidimensional schemas are essential to create relevant databases.
- (ii) **Transparency:** An end-user must present a cohesive, unambiguous version of data and must not be exposed to complexity and diversity of data sources.
- (iii) **Accessibility:** Essential data must be identified and accessed.
- (iv) **Consistent Reporting Performance:** Reporting must remain dependable and reliable even with an increase in database size.
- (v) **Client/Server Architecture:** Data mining and warehousing systems are created to meet the growing business needs and financial constraints.
- (vi) **Generic Dimensionality:** Every data dimension has the same importance.
- (vii) **Dynamic Sparse Matrix Handling:** Provides capability to keep the database size within limits by adopting suitable methods of handling sparse matrices.
- (viii) **Multi-User Support:** The system must permit access to a large number of users at the same time.
- (ix) **Unrestricted Cross-Dimensional Operations:** Multidimensional schemas must be well understood, designed and permit cross-references.
- (x) **Intuitive Data Manipulation:** OLAP is created for decision-makers to make intuitive decisions. They are not computer experts and must be provided with a user-friendly, uncomplicated access to generate queries.
- (xi) **Flexible Reporting:** The system must be capable of providing reports desired by the end- users.
- (xii) **Unlimited Dimensions and Aggregation Levels:** The system must remain flexible/ expandable for adding extra dimensions and permitting additional aggregations.

NOTES

NOTES

Data Mining and Warehousing Challenges

Major Issues/Challenges to Data Mining

The major issues and challenges to data mining are as follows:

- (i) Mining different types of knowledge in databases.
- (ii) Interactive mining of knowledge at multiple levels of abstraction.
- (iii) Incorporation of background knowledge.
- (iv) Data mining query languages and ad hoc data mining.
- (v) Presentation and visualization of data mining results.
- (vi) Handling noisy or incomplete data.

Data Warehouse—Open Issues and Research Problems

Data warehousing being an evolving system has many issues which are open for further study and research. These are in the areas of the following:

- (i) **Security:** Data warehouse contains full details of organizational information, some of which are highly confidential. A data warehouse provides access to many employees for accessing its contents. Special efforts are required to maintain the safety, security and confidentiality of this valuable and sensitive information residing in it.
- (ii) **Performance:** Very large databases require their storage in multiprocessors and not in conventional computers. These fall under the category of MPP and SMP. Both are upgradeable and permit very large database to be partitioned and accessed at fast speeds. Features, such as shared memory protection and dynamic load balancing are in-built in them. Designing scalable systems is an important factor.
- (iii) **Functionality:** Modifying accessing methods is a major challenge to data warehouse managers. Developing novel data mining algorithms and OLAP for meeting users' latest requirements are essential features of extracting interesting data from the warehouse.
- (iv) **Presentation:** Data visualization techniques have to be constantly improved for end-users to utilize the full capability of the system, draw inferences and analyse the results.

Check Your Progress

1. What is data warehouse database?
2. What is star schema?

1.3 WAREHOUSE SERVER – OLAP OPERATIONS

Data warehouse is an environment that provides an architectural construct of information system by collecting and managing data from multiple sources to provide meaningful business insight. In today's business environment, the data warehousing and its complementary technologies are providing the organizations with a sustainable competitive advantage. The technology helps organizations to make informed and relational decisions about different business aspects such as customer retention, sales and customer service, marketing, risk assessment and fraud detection etc.

The architecture of data warehouse is based on a relational database management system server and acts as a central repository for informational data. The difference is that the operational data and processing is completely separate from data warehouse processing. Of course, the operational applications are the source of data for the warehouse, where the data is transformed into an integrated structure and format as soon as the operational data is entered into the warehouse. All these significant data collected by various business activities are stored on a physical storage of the data warehouse system known as data warehouse servers. The data warehouse must be capable of storing and managing large volumes of data as well as different types of data structures as data warehouse contains a large historical components (covering data from 5 to 10 years also). The data warehouse server is responsible for performing data logic, data services, file services and maintaining metadata. For any organization, the configuration of a warehouse server is very critical. It is also very critical for optimization and querying of the data. The processing power of this server should be capable of handling multiple queries at the same time along with the ability to store, manage and secure the historical as well as new data. The report generation based on several queries without any interruption in other ongoing processes is also a characteristic of warehouse server as interruption may have a domino effect on the other server units present in the data warehouse system. This means that the data warehouse design and data warehouse software installed in the data warehouse system must be optimized to run as a unit. Moreover, the data warehouse system must be secure and responsive while data consolidation, which means that reliability of data warehouse server is also an integral part of the data warehouse architecture.

The above discussion shows that, for any business enterprise choice of a right kind of data warehouse server is important for fast and efficient data processing.

As discussed, the data warehouse server is the repository to store historical data and can be used for analysis. The data stored in data warehouse is analysed and evaluated using analytical queries. This analysis and evaluation of data is being done using a computing method called as Online Analytical Processing (OLAP). OLAP system helps data warehouse to analyse the data effectively. It allows users to analyse the information from multiple database systems at the same time. It extracts the data and enables users to view business data from different points of

NOTES

NOTES

view. The data coming from various resources sources get stored in data warehouse. This data is organized in warehouse with the help of OLAP tool using multidimensional model. As OLAP operations is a multidimensional data models, these operations are performed over a data cube, and is a core concept of OLAP optimized for quick and consistent data analysis. The OLAP cube, also called as hypercube, is a data structure consisting of numeric facts called measures which are categorized by dimensions of business intelligence such as relational database, data mining and report writing.

Generally, OLAP applications include sales report, marketing, business process management (BPM), forecasting, budgeting, creating finance report etc.

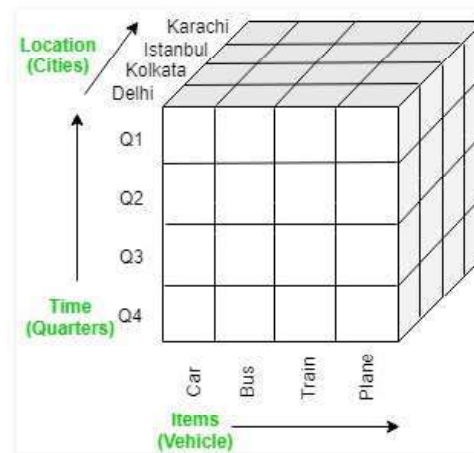


Fig. 1.8 Data cube

There are four types of OLAP servers which are discussed earlier.

- **Relational OLAP (ROLAP):** This is a form of OLAP that manages and stores the data in the form of relational data base, placed between relational back-end server and client front-end tools. It has no limitations on data size and can handle large amounts of data
- **Multidimensional OLAP (MOLAP):** This uses array based multidimensional cube to store the data and gives multidimensional views of data. To handle low storage utilization due to dense and sparse data sets, MOLAP uses two levels of data storage. Since MOLAP cubes are built for fast retrieval, it performs excellently and is optimal for slicing and dicing operations. It can also perform complex calculations.
- **Hybrid OLAP (HOLAP):** This is a combination of both ROLAP and MOLAP and provides the properties of both types of servers. It provides the higher scalability of ROLAP and faster computation of MOLAP. It allows to store the large data volumes of detailed information and leverages cube technology for faster performance.

- **Specialized SQL Servers:** This is a specialized support for SQL queries over star and snowflake schemas in a read-only environment that provides advanced query language and query processing.

Since OLAP servers are based on multidimensional view of data, OLAP techniques are applied to retrieve the information from the data warehouse in the form of OLAP multidimensional databases. Multidimensional model has two types of tables:

1. Dimension tables: contains the attributes of dimensions
2. Fact tables: consists of the facts or measures.

Following are the OLAP operations:

- **Roll-Up**

Since multidimensional databases have hierarchies with respect to dimensions, this operation performs aggregation on the data relationship with respect to one or more dimensions. The aggregation is performed in any of the following ways:

- o By climbing up a concept hierarchy for a dimension
- o By dimension reduction

Roll-up is a dimension reduction technique on a given data cube that can be done by combining similar dimension across any axis of the data cube using notion of concept hierarchy.

Consider for example the sale of various items in different locations and in different quarters. This is shown in Figure 1.9. The aggregate function (roll-up) is performed on the data shown in Figure 1.8.

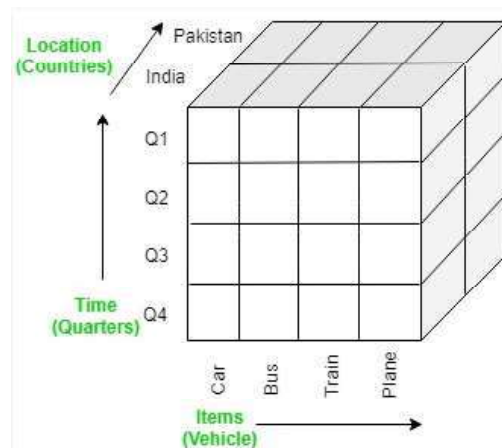


Fig. 1.9 Roll-Up operation

In the above illustration, it can be seen that the cities Karachi, Istanbul, Kolkata and Delhi are rolled up into the countries Pakistan and India respectively. That is, the data is grouped into countries rather than cities. When rolled-up, at least one or more dimensions need to be removed.

NOTES

NOTES

• Drill-Down

Drill-down is exactly the reverse process of Roll-up. This process expands dimension or add new dimension by fragmenting the data into smaller parts. That is, less detailed data is converted into highly detailed data. The expansion is done in the following ways:

- o By moving down a concept hierarchy for a dimension
- o By adding a new dimension

Figure 1.10 illustrates the drill-down operation performed on the Figure 1.8 by moving down a concept hierarchy Time dimension from quarter to months.

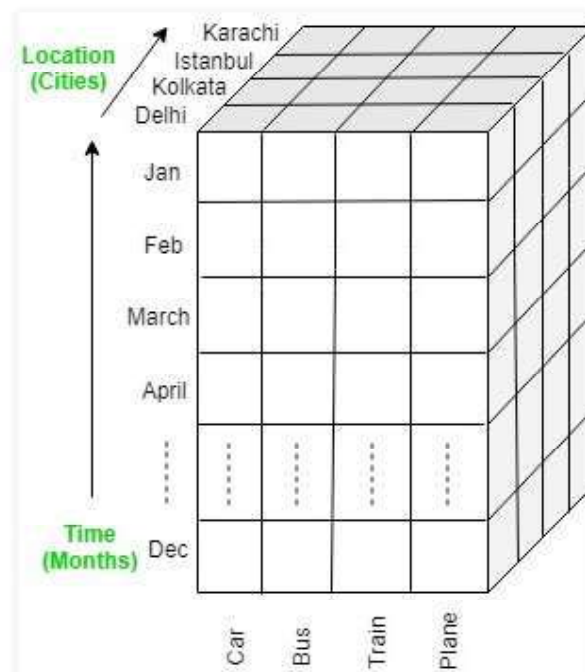


Fig. 1.10 Drill-Down operation

The time dimension is expanded from the quarter level to month level on drilling down. During the drill-down process, one or more dimensions are added from the data cube that navigates the less detailed data to a highly detailed data.

• Slice

In this operation, one particular dimension is selected from the cube and a new sub-cube is created. Figure 1.11 illustrates the slice operation performed on the Figure 1.8 by selecting the dimension Time using the criterion time = "Q1" as filter. All together a new cube is being created.

NOTES



Fig. 1.11 Slice operation

- **Dice**

This operations works by creating a new sub-cube by selecting two or more dimensions from the data cube. In Figure 1.12, a new sub-cube is created by selecting the following dimension criterion:

Time = “Q1” or “Q2”

Item = “Car” or “Bus”

Location = “Delhi” or “Kolkata”

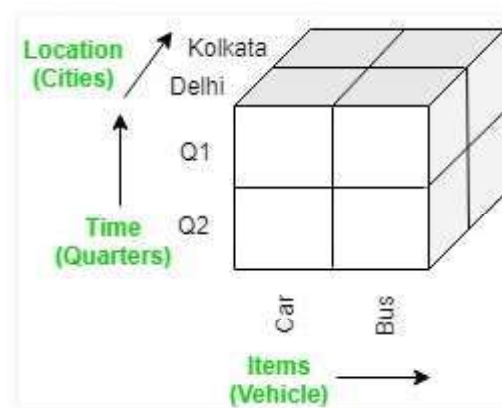


Fig. 1.12 Dice operation

- **Pivot**

Also known as rotation operation, it rotates the data axes to provide alternate view or presentation of data to the users. Figure 1.13 represents the sub-cube generated after applying the pivot operation on the data cube presented in Figure 1.8.

NOTES

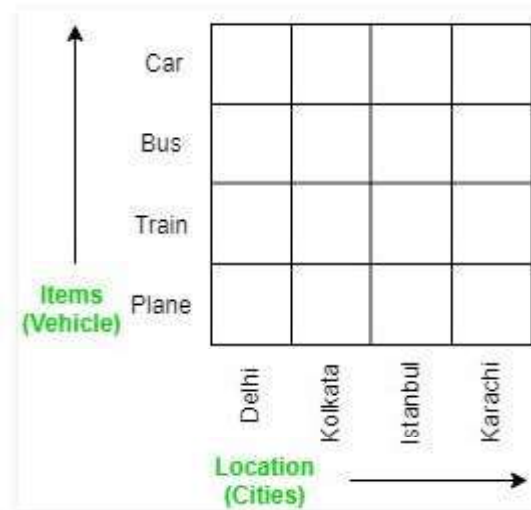


Fig. 1.13 Pivot operation

Check Your Progress

3. What is drill-down?
4. Discuss multidimensional OLAP.

1.4 DATA WAREHOUSE TECHNOLOGY- HARDWARE AND OPERATING SYSTEM

The terms hardware and operating systems are the server platforms and operating systems that support the computing environment of the data warehouse. The various jobs such as data extraction, transformation, integration, and staging jobs run on the hardware selected with the help of operating system chosen. The challenge for the companies to implement data warehouse environment is to select an appropriate platform (hardware, software, networking etc.).

In making the selection for the hardware platform to implement data warehouse, there are certain factors that need to be considered carefully. These factors are:

- **Scalability:** The data warehouse platform should be scalable to be able to handle the system as the data grows. The proposed platform should be able to handle large sets of data, number of concurrent users and complex queries in a most efficient and consistent way. The data warehouse should also be able to understand the additional hardware and software required for each of the incremental uses.
- **Powerful:** The data warehouse platform should be designed to support complex decision making activity in a multi-user, mixed-workload

environment. It should also be able to support all types of queries with good performance and should be able to determine the best execution plan when data demographics changes.

- **Manageable:** The platform requires minimal support or minimal intervention from the system administration and should simplify the task by providing a single point of control. The platform should also be capable of providing a robust set of features and functions that may include monitoring utilities, locking scheme and other security mechanism, remote maintenance capabilities and user chargeback functionalities.
- **Extensible:** The platform data warehouse should be extensible (i.e. provide flexible database design and system architecture) that can keep pace with and can accommodate the evolving business requirements.

Apart from above discussed factors, the platform should also be available, interoperable and affordable. Moreover, the vendor viability and stability should also be considered.

NOTES

Check Your Progress

5. What is the relevance of hardware and operating systems in data warehousing?
6. What are the challenges faced by companies to implement data warehousing?

1.5 ANSWERS TO CHECK YOUR PROGRESS QUESTIONS

1. Data warehouse database is the central database consisting of data warehouse RDBMS, a large repository and supporting databases such as multi-relational database, multidimensional database and data marts.
2. Star schema is the most common ER model. In the star schema there is a large central fact table containing numerical data without duplication or redundancy. A large number of dimension tables are referred by it. Each of these handles a dimension.
3. Drill-down process expands dimension or add new dimension by fragmenting the data into smaller parts. That is, less detailed data is converted into highly detailed data.
4. Multidimensional OLAP (MOLAP) uses array based multidimensional cube to store the data and gives multidimensional views of data. To handle low storage utilization due to dense and sparse data sets, MOLAP uses two levels of data storage.

NOTES

5. The terms hardware and operating systems are the server platforms and operating systems that support the computing environment of the data warehouse. The various jobs such as data extraction, transformation, integration, and staging jobs run on the hardware selected with the help of operating system chosen.
6. The challenge for the companies to implement data warehouse environment is to select an appropriate platform (hardware, software, networking etc.).

1.6 SUMMARY

- A data warehouse is a subject oriented, integrated, non-volatile and time variant collection of data in support of management's decisions.
- Data is transformed and loaded into the Operational Data Store (ODS) in real time frame. From the ODS it is loaded into the data warehouse after extraction and cleaning operations at regular intervals, but not as and when received from external sources.
- Metadata is data about data and keeps information as:
 - (i) Technical Metadata: It contains the sources of access data, data structure, transformation description, rules specified during data processing, access authorizations and backup history.
 - (ii) Business Metadata: It contains information about the subject areas, information object types, the Internet home pages, details of information delivery system.
- Data Warehouse Database - It is the central database consisting of data warehouse RDBMS, a large repository and supporting databases such as multi-relational database, multidimensional database and data marts.
- Data mart is another important component of data warehouse and is a data store that is subsidiary to a data warehouse. It is created to meet specific information needs of different functional area managers.
- Information Delivery System - It provides an external interface to provide data warehouse reports, information objects to external users as per a specified schedule.
- Granularity of data refers to the level of detail or summarization at which data is stored in a data warehouse. Larger the granularity less will be the detail available for those data items and vice versa also holds good.
- Data warehouses and OLAP tools are based on what is known as a multidimensional model.
- Schemas - While Entity-Relationship Model (E-R Model) was found adequate in the design of relational databases, a data warehouse requires a subject-oriented schema for better analysis and handling of more complex queries.

- Three types of schemas are:
 - (i) **Star Schema:** It is the most common model. In the star schema there is a large central fact table containing numerical data without duplication or redundancy.
 - (ii) **Snowflake Schema:** It is an extension of star schema. The dimension tables are further normalized and extra tables are added.
 - (iii) **Fact Constellation Schema:** It has multiple fact tables to meet the requirements of more advanced applications.
- Data warehouse architecture is based on an RDBMS system server. It has a massive central repository for storage of data, subsidiary databases and front-end tools.
- Multidimensional Online Analytical Processing or MOLAP architecture creates a data structure to store data in a way it will finally be utilized to enhance its performance.
- Relational Online Analytical Processing or ROLAP works directly with RDBMS and is more scalable. It depends on databases for calculations, and therefore its performance suffers.
- Hybrid Online Analytical Processing or HOLAP uses the best features of both, i.e., the flexibility of ROLAP RDBMS and the optimized multidimensional structure of MOLAP.
- OLTP databases handle real-time transactions having some inherent and special requirements.
- Data warehouse is an environment that provides an architectural construct of Information system by collecting and managing data from multiple sources to provide meaningful business insight.
- The architecture of data warehouse is based on a relational database management system server and acts as a central repository for informational data.
- The data stored in data warehouse is analysed and evaluated using analytical queries. This analysis and evaluation of data is being done using a computing method called as Online Analytical Processing (OLAP).
- There are four types of OLAP servers:
 - (i) Relational OLAP (ROLAP)
 - (ii) Multidimensional OLAP (MOLAP)
 - (iii) Hybrid OLAP (HOLAP)
 - (iv) Specialized SQL Servers
- Following is the list of OLAP operations:
 - (i) Roll-Up: Roll-up is a dimension reduction technique on a given data cube that can be done by combining similar dimension across any axis of the data cube using notion of concept hierarchy.

NOTES

NOTES

- (ii) Drill-Down: Drill-down is exactly the reverse process of Roll-up. This process expands dimension or add new dimension by fragmenting the data into smaller parts.
 - (iii) Slice: In this operation, one particular dimension is selected from the cube and a new sub-cube is created.
 - (iv) Dice: This operations works by creating a new sub-cube by selecting two or more dimensions from the data cube.
 - (v) Pivot: Also known as rotation operation, it rotates the data axes to provide alternate view or presentation of data to the users.
- The terms hardware and operating systems are the server platforms and operating systems that support the computing environment of the data warehouse.
 - In making the selection for the hardware platform to implement data warehouse, there are certain factors that need to be considered carefully. These factors are:
 - (i) Scalability
 - (ii) Powerful
 - (iii) Manageable
 - (iv) Extensible

1.7 KEY WORDS

- **Data Warehouse:** A subject oriented, integrated, non-volatile and time-variant collection of data in support of management's decisions.
- **Technical Metadata:** It contains the sources of access data, data structure, transformation description, rules specified during data processing, access authorizations and backup history.
- **Business Metadata:** It contains information about the subject areas, information object types, the Internet home pages, details of information delivery system.

1.8 SELF ASSESSMENT QUESTIONS AND EXERCISES

Short Answer Questions

1. What is OLAP?
2. What are data pre – processing tools?
3. What are specialized SQL servers?
4. List the types of access tools in data warehousing.

Long Answer Questions

1. What is data warehousing? Discuss the data warehouse building blocks.
2. What is OLAP? Describe its types and its operations.
3. What the factors considered for the selection of the hardware platform to implement data warehouse?
4. Explain the different types of schemas.
5. Explain ROLAP, MOLAP and HOLAP.

NOTES

1.9 FURTHER READINGS

Han, Jiawei, Micheline Kamber and Jian Pei. 2011. *Data Mining: Concepts and Techniques*, 2nd edition. Amsterdam: Elsevier.

Pujari, Arun K. 2010. *Data Mining Techniques*, 2nd edition. United States: Universities Press.

Anahory, Sam and Dennis Murray. 1997. *Data Warehousing in the Real World: A Practical Guide for Building Decision Support Systems*, 1st edition. Boston: Addison Wesley.

Witten, I. H. and Eibe Frank. 2005. *Data Mining: Practical Machine Learning Tools and Techniques*, 2nd edition. Amsterdam: Elsevier.

Soman, K. P., Shyam Diwakar and V. Ajay. 2006. *Insight Into Data Mining: Theory and Practice*. New Delhi: PHI.

UNIT 2 DATA MINING

NOTES

Structure

- 2.0 Introduction
- 2.1 Objectives
- 2.2 Data Mining
 - 2.2.1 Data Mining and Knowledge Discovery
- 2.3 Data Mining Functionalities and Techniques
- 2.4 Data Mining Task Primitives
- 2.5 Integration of Data Mining Systems
- 2.6 Data Reduction
- 2.7 Data Mining Applications
- 2.8 Trends in Data Mining
- 2.9 Answers to Check Your Progress Questions
- 2.10 Summary
- 2.11 Key Words
- 2.12 Self Assessment Questions and Exercises
- 2.13 Further Readings

2.0 INTRODUCTION

Data mining is the practice of discovering patterns in huge data sets involving methods at the crossroads of machine learning, statistics, and database systems. Data mining is an interdisciplinary subfield of computer science and statistics with an overall objective to dig out information (with intelligent methods) from a data set and transform the information into a understandable structure for further use.

Data mining is the analysis step of the “knowledge discovery in databases” process, or KDD. Aside from the raw analysis step, it also involves database and data management aspects, data pre-processing, model and inference considerations, interestingness metrics, complexity considerations, post-processing of discovered structures, visualization, and online updating. The difference between data analysis and data mining is that data analysis is used to test models and hypotheses on the dataset, e.g., analyzing the efficiency of a marketing campaign, regardless of the amount of data; in contrast, data mining uses machine-learning and statistical models to discover surreptitious or hidden patterns in a large volume of data.

2.1 OBJECTIVES

After going through this unit, you will be able to:

- Explain data mining
- Understand data mining functionalities and techniques

- Discuss data mining task primitives
- Understand Integration of data mining systems
- Define data reduction
- Explain data mining applications
- Discuss the trends in data mining

NOTES

2.2 DATA MINING

Data mining means locating, identifying and finding unforeseen information from a large database. The information is one which is interesting to the end-user. It can also be understood as a data analysis based on searching or learning dependent on deduction.

A data pattern discovered through a database search is considered interesting, if it is easily understood, is valid on a new or test data with some degree of uncertainty, potentially useful and is novel. Interesting patterns are identified by objective parameters which are combined with subjective requirements to reflect the needs and interests of a particular user.

Data mining software is an analytical tool used for data analysis. Users can analyse data from different angles, classify/categorize it, and then summarize the identified relationships. Technologically, data mining follows the process of finding correlations or patterns among many fields having large relational databases.

Data mining technology is not new but the term ‘data mining’ is new. Companies in the past have handled large databases using powerful computers for market research. Continuous advancement in processing power of computer and statistical software are enabling more use of accurate analysis at a lower cost.

2.2.1 Data Mining and Knowledge Discovery

Data mining is devoted specifically to the processes involved in the extraction of useful information by applying specific techniques based on certain knowledge domains. These are based on statistics, artificial intelligence, and so on. However, knowledge discovery is a wide term and is the entire range of activities right from deciding business objectives, capturing desired data, preparing, processing, arranging them, applying predefined techniques and then presenting them in an understandable form to the user. To say specifically, knowledge discovery can be subdivided into five specific steps which are performed repetitively till the desired result is reached, and one of them is data mining.

- (i) Data processing comprising data selection, data cleaning and data integration.
- (ii) Data transformation and organization in a form ready for fast access.
- (iii) Data Mining (DM) engine and other techniques, such as OLAP or Online Transaction Processing (OLTP) for searching and extraction.

- (iv) Knowledge presentation methods through Graphical User Interface (GUI).
- (v) Analysing results and assimilating them in a knowledge domain.

Figure 2.1 shows the steps in knowledge discovery.

NOTES

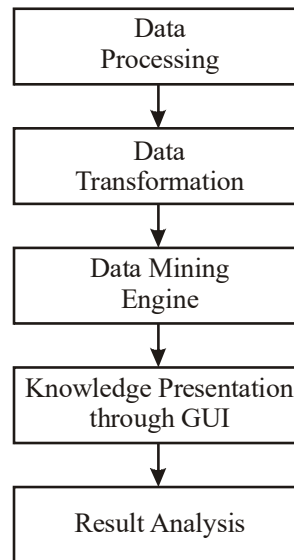


Fig. 2.1 Steps in Knowledge Discovery

We can, thus, consider data mining as a subset of knowledge discovery.

Nature of Data to be Mined—Operational and Analytical

Data mining is an essential step towards the creation of information systems. These are operational information systems, such as Enterprise Resource Planning (ERP) or management information systems including Decision Support System (DSS). DSS assists managers in taking decisions based on available unstructured data and to validate their intuitive judgements. OIS and DSS have their own requirements of data structures and databases.

The data in turn is categorized as operational data which is dynamic in nature and meets short-term goals. Another category is analytical data that has a longer time span and supports intuitive decisions. Operational database supports transaction processing through Online Transaction Processing (OLTP) queries. Analytical database meets the Online Analytical Processing (OLAP) requirements of DSS.

Table 2.1 shows the differences in database requirements for OLTP and DSS.

Table 2.1 Differences in Database (DB) Requirements for OLTP and DSS

Characteristic	DB for OLTP	DB for OLAP Needs
1. Nature of content	Dynamic	Static
2. Time span	Current	Historical
3. Time measured	Implicit, implied	Explicit and mentioned
4. Level of detail/granularity	Primitive/detailed	Detailed and derived
5. Update cycle	Real time	Periodic, planned
6. Tasks	Known pattern, Repetitive	Unpredictable
7. Response	Time bound	Flexible

NOTES

Data Mining—A Multidisciplinary Area

Data mining is a combination of multiple disciplines. Some of the disciplines are:

- (i) Information Science
- (ii) Database Technology
- (iii) Statistics
- (iv) Machine Learning
- (v) Visualization
- (vi) Other Sciences

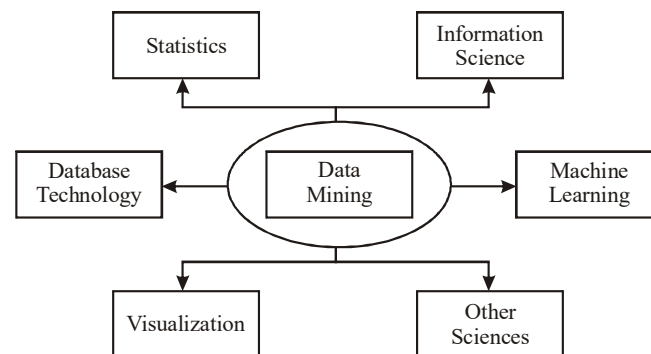


Fig. 2.2 Various Disciplines Contributing to Data Mining

Successful development of a data mining system would thus require joint efforts from experts of different domains.

Classification of Data Mining Systems

Data mining means developing special algorithms to answer the queries of various users. The procedure is to evolve a number of models and to match one of them to

data stored in the database. Three steps involved in this process are: creating a model, finding out the criteria to give preference to a model over others and identifying the search technique.

NOTES

Data mining models being mathematical in nature are categorized as predictive and descriptive.

- (i) A predictive model spells out in advance the values a data may assume based on known results from other data stored in the database. This model performs data mining tasks of classification, time series analysis, regression and prediction.
- (ii) A descriptive model is based on identification and relationships in data. This model aims at discovering rather than predicting the properties of data.

A descriptive model performs data mining tasks comprising clustering, summarization, association rules and sequence discovery.

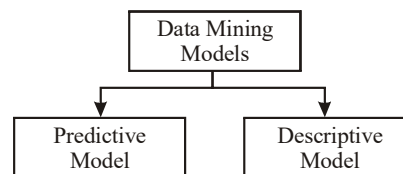


Fig. 2.3 Types of Data Mining Models

Data Mining Tasks

The basic tasks under predictive and descriptive models are:

(i) Predictive Model

- (a) **Classification:** Data is mapped into predefined groups or classes. It is also termed as supervised learning as classes are established prior to examination of data.
- (b) **Regression:** It means mapping of data items into known types of functions. These may be linear, logistic functions, etc.
- (c) **Time Series Analysis:** The value of an attribute is examined at evenly spaced times, as it varies with time.
- (d) **Prediction:** It means foretelling future data states based on past and current data.

(ii) Descriptive Model

- (a) **Clustering:** It is referred to as unsupervised learning or segmentation/partitioning. In clustering, groups are not predefined.
- (b) **Summarization:** Data is mapped into subsets with simple descriptions. It is also termed as characterization or generalization.

- (c) **Sequence Discovery:** Sequential analysis or sequence discovery is utilized to find out sequential patterns in data. It is similar to association, but the relationship is based on time.
- (d) **Association Rules:** A model which identifies specific types of data associations.

NOTES

Data Mining Primitives

A data mining task is expressed in the form of a Data Mining Query Language (DMQL) statement and requires certain primitives to be stated. The primitives are:

- (i) **Task-Relevant Data:** It mentions the part of the database to be examined.
- (ii) **Nature of Knowledge to be Mined:** It defines the tasks or functions to be performed on the data. Examples are characterization, association and clustering.
- (iii) **Background Knowledge:** It means the concept hierarchy as it indicates the level of abstraction at which data is to be mined.
- (iv) **Interestingness Measures:** These are defined for the task or function to be performed. For example, for association rules, the support and confidence factors are measured corresponding to the threshold levels specified by users as a measure of interestingness.
- (v) **Presentation and Visualization of Discovered Patterns:** They refer to the ways in which the result obtained can be displayed for the convenience of the user.

Data Mining Query Language (DMQL)

Data mining systems are required to support the ad hoc and interactive requirements of knowledge discovery from relational database and multiple levels of abstraction. Data mining languages are designed to meet this requirement. They help in formulating a query to define data mining task primitives.

The primitives require:

- (i) Set of task-relevant data to be mined.
- (ii) Nature of knowledge to be mined.
- (iii) Background knowledge required for the discovery.
- (iv) Measures of interestingness.
- (v) Visualization representation.

DMQL follows a Structured Query Language (SQL) like syntax which is amenable for linking with relational query languages and it simplifies a user's task of knowledge extraction.

NOTES

Different data mining algorithms extract different patterns from data. For supporting knowledge discovery process, integrated systems are not needed that are capable of dealing with either data patterns. Knowledge discovery processes are querying processes and query languages are needed for this. To support declarative specification of pattern and data manipulation for rule mining, query languages are to be designed.

Integration of Data Mining Systems

A diverse number of data mining tools may be available in an organization. It is essential to identify and categorize them to understand which model and tasks they support. Also, it is necessary to find out if some data mining tools are being developed in-house. They should be displayed on the GUI of client desktop to select the right data mining tool for the problem in hand.

Data mining concepts boost the ability to analyse data and these techniques are based on statistics and machine learning. Data analysis should be consolidated at the warehouse for data integrity and management concerns. It is of utmost need to integrate data mining technology seamlessly under the framework of traditional database systems.

Significant progress in research has been made for streamlining data mining algorithms. There are huge requirements for scaling data mining techniques for working with large databases. Data mining implementations are ‘disk-aware’ as well as ‘SQL aware’. Implementations use functionalities provided by SQL Engine and the API.

Advancement in data mining can implement several data mining operations on large databases, very efficiently. The final goal of data mining is construction of data mining models from their databases, and use of models for a variety of predictive and analytic tasks, and sharing these models with other applications. This type of integration is essential to make data mining successful in the database environment.

Although a data mining model can be derived by use of an SQL application, for implementing a training algorithm database management system will not be aware about the semantics of the mining models. This is because these mining models have no explicit representation in database. But, such explicit representation is required to enable database management system to share, reuse and manage mining models in an efficient way.

Profitable Applications

Many companies have used data mining applications successfully. Early users were more information-intensive. But the technology may be used by any company that looks to use large data warehouse for managing their customer relationships in a better way. Data mining needs a large and well-integrated data warehouse and well-defined understanding of the business process for applying data mining.

Successful application areas of data mining are as follows:

- Analysis of recent activities and results gained by the sales force to find marketing activities having greatest impact in the coming months. Data on activities of competitors and information on existing local health care systems are needed. These results require distribution to sales personnel using wide area network enabling representatives to review the recommendations.
- Analysis by credit card companies to use vast data warehouse on customer transaction for identification of potential customers for new product. A test mail can be sent to know the attributes of customers that may have an affinity for the product.
- A transportation company whose business is spread over many areas and has a large direct sales force, can use data mining for identification of the best prospects about its services.
- A company dealing in consumer package goods may use data mining for improving its sales process for the satisfaction of retailers. If the company has many units at different locations it may take data from the activities of competitors, list of present consumers, weekly or monthly enquiries, details of shipments, etc., can be applied to understand the reasons for brand and store switching.

NOTES

2.3 DATA MINING FUNCTIONALITIES AND TECHNIQUES

We have observed various types of databases and information repositories on which data mining can be performed. Let us now examine the kinds of data patterns that can be mined.

Data mining functionalities are used to specify the kind of patterns to be found in data mining tasks. In general, data mining tasks can be classified into two categories: descriptive and predictive. Descriptive mining tasks characterize the general properties of the data in the database. Predictive mining tasks perform inference on the current data in order to make predictions.

In some cases, users may have no idea regarding what kinds of patterns in their data may be interesting, and hence may like to search for several different kinds of patterns in parallel. Thus, it is important to have a data mining system that can mine multiple kinds of patterns to accommodate different user expectations or applications. Furthermore, data mining systems should be able to discover patterns at various granularities (i.e., different levels of abstraction). Data mining systems should also allow users to specify hints to guide or focus the search for interesting patterns. Because some patterns may not hold for all of the data in the database, a measure of certainty or “trustworthiness” is usually associated with each discovered pattern.

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

Concept/Class Description: Characterization and Discrimination

NOTES

Data can be associated with classes or concepts. For example, in the AllElectronics store, classes of items for sale include computers and printers, and concepts of customers include bigSpenders and budgetSpenders. It can be useful to describe individual classes and concepts in summarized, concise, and yet precise terms. Such descriptions of a class or a concept are called class/concept descriptions. These descriptions can be derived via (1) data characterization, by summarizing the data of the class under study (often called the target class) in general terms, or (2) data discrimination, by comparison of the target class with one or a set of comparative classes (often called the contrasting classes), or (3) both data characterization and discrimination.

Data characterization is a summarization of the general characteristics or features of a target class of data. The data corresponding to the user-specified class are typically collected by a database query. For example, to study the characteristics of software products whose sales increased by 10% in the last year, the data related to such products can be collected by executing an SQL query.

There are several methods for effective data summarization and characterization. The data cube-based OLAP roll-up operation can be used to perform user-controlled data summarization along a specified dimension. The output of data characterization can be presented in various forms. Examples include pie charts, bar charts, curves, multidimensional data cubes, and multidimensional tables, including crosstabs. The resulting descriptions can also be presented as generalized relations or in rule form (called characteristic rules).

Example 2.1 Data Characterization

A data mining system should be able to produce a description summarizing the characteristics of customers who spend more than \$1,000 a year at AllElectronics. The result could be a general profile of the customers, such as they are 40–50 years old, employed, and have excellent credit ratings. The system should allow users to drill down on any dimension, such as on occupation in order to view these customers according to their type of employment.

Data discrimination is a comparison of the general features of target class data objects with the general features of objects from one or a set of contrasting classes. The target and contrasting classes can be specified by the user, and the corresponding data objects retrieved through database queries. For example, 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. The methods used for data discrimination are similar to those used for data characterization.

“How are discrimination descriptions output?” The forms of output presentation are similar to those for characteristic descriptions, although discrimination descriptions should include comparative measures that help distinguish between the target and contrasting classes. Discrimination descriptions expressed in rule form are referred to as discriminant rules.

Example 2.2 Data Discrimination

A data mining system should be able to compare two groups of AllElectronics customers, such as those who shop for computer products regularly (more than two times a month) versus those who rarely shop for such products (i.e., less than three times a year). The resulting description provides a general comparative profile of the customers, such as 80% of the customers who frequently purchase computer products are between 20 and 40 years old and have a university education, whereas 60% of the customers who infrequently buy such products are either seniors or youths, and have no university degree. Drilling down on a dimension, such as occupation, or adding new dimensions, such as income level, may help in finding even more discriminative features between the two classes.

Mining Frequent Patterns, Associations and Correlations

Frequent patterns, as the name suggests, are patterns that occur frequently in data. There are many kinds of frequent patterns, including itemsets, subsequences and sub structures. A frequent itemset typically refers to a set of items that frequently appear together in a transactional data set, such as milk and bread. A frequently occurring subsequence, such as the pattern that customers tend to purchase first a PC, followed by a digital camera and then a memory card, is a (frequent) sequential pattern. A substructure can refer to different structural forms, such as graphs, trees or lattices, which may be combined with itemsets or subsequences. If a substructure occurs frequently, it is called a (frequent) structured pattern. Mining frequent patterns leads to the discovery of interesting associations and correlations within data.

Example 2.3 Association Analysis

Suppose, as a marketing manager of AllElectronics, you would like to determine which items are frequently purchased together within the same transactions. An example of such a rule, mined from the AllElectronics transactional database, is

$$\text{buys}(X, \text{“computer”}) \Rightarrow \text{buys}(X, \text{“software”}) \text{ [support} = 1\%, \text{ confidence} = 50\%]$$

Where, X is a variable representing a customer. A confidence, or certainty, of 50% means that if a customer buys a computer, there is a 50% chance that she will buy software as well. A 1% support means that 1% of all of the transactions under analysis showed that computer and software were purchased together. This association rule involves a single attribute or predicate (i.e., buys) that repeats. Association rules that contain a single predicate are referred to as single-dimensional

NOTES

NOTES

association rules. Dropping the predicate notation, the above rule can be written simply as “computer \Rightarrow software [1%, 50%]”.

Suppose, instead, that we are given the AllElectronics relational database relating to purchases. A data mining system may find association rules like:

$$\text{age}(X, \text{“20...29”}) \wedge \text{income}(X, \text{“20K...29K”}) \Rightarrow \text{buys}(X, \text{“CD player”})$$

[support = 2%, confidence = 60%]

The rule indicates that of the AllElectronics customers under study, 2% are 20 to 29 years of age with an income of 20,000 to 29,000 and have purchased a CD player at AllElectronics. There is a 60% probability that a customer in this age and income group will purchase a CD player. Note that this is an association between more than one attribute, and predicate (i.e., age, income, and buys). Adopting the terminology used in multidimensional databases, where each attribute is referred to as a dimension, the above rule can be referred to as a multidimensional association rule.

Typically, association rules are discarded as uninteresting if they do not satisfy both a minimum support threshold and a minimum confidence threshold. Additional analysis can be performed to uncover interesting statistical correlations between associated attribute-value pairs.

Frequent itemset mining is the simplest form of frequent pattern mining. Sequential pattern mining and structured pattern mining are considered advanced topics.

Classification and Prediction

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

“How is the derived model presented?” The derived model may be represented in various forms, such as classification (IF-THEN) rules, decision trees, mathematical formulae or neural networks. A decision tree is a flow chart like tree structure, where each node denotes a test on an attribute value, each branch represents an outcome of the test, and tree leaves represent classes or class distributions. Decision trees can easily be converted to classification rules. A neural network, when used for classification, is typically a collection of neuron-like processing units with weighted connections between the units. There are many other methods for constructing classification models, such as naïve Bayesian classification, support vector machines, and k-nearest neighbor classification.

Whereas, classification predicts categorical (discrete, unordered) labels, prediction models continuous-valued functions. That is, it is used to predict missing or unavailable numerical data values rather than class labels. Although the term

prediction may refer to both numeric prediction and class label prediction, this is use it to refer primarily to numeric prediction in subsequent sections. Regression analysis is a statistical methodology that is most often used for numeric prediction, although other methods exist as well. Prediction also encompasses the identification of distribution trends based on the available data.

Classification and prediction may need to be preceded by relevance analysis, which attempts to identify attributes that do not contribute to the classification or prediction process. These attributes can then be excluded.

Example 2.4 Classification and Prediction

Suppose, as sales manager of AllElectronics, you would like to classify a large set of items in the store, based on three kinds of responses to a sales campaign: good response, mild response, and no response. You would like to derive a model for each of these three classes based on the descriptive features of the items, such as price, brand, place made, type, and category. The resulting classification should maximally distinguish each class from the others, presenting an organized picture of the data set. Suppose that the resulting classification is expressed in the form of a decision tree. The decision tree, for instance, may identify price as being the single factor that best distinguishes the three classes. The tree may reveal that, after price, other features that help further distinguish objects of each class from another include brand and place made. Such a decision tree may help you understand the impact of the given sales campaign and design a more effective campaign for the future. The classification model can be represented in various forms, such as IF-THEN rules, decision tree and neural network (see Figure 2.4).

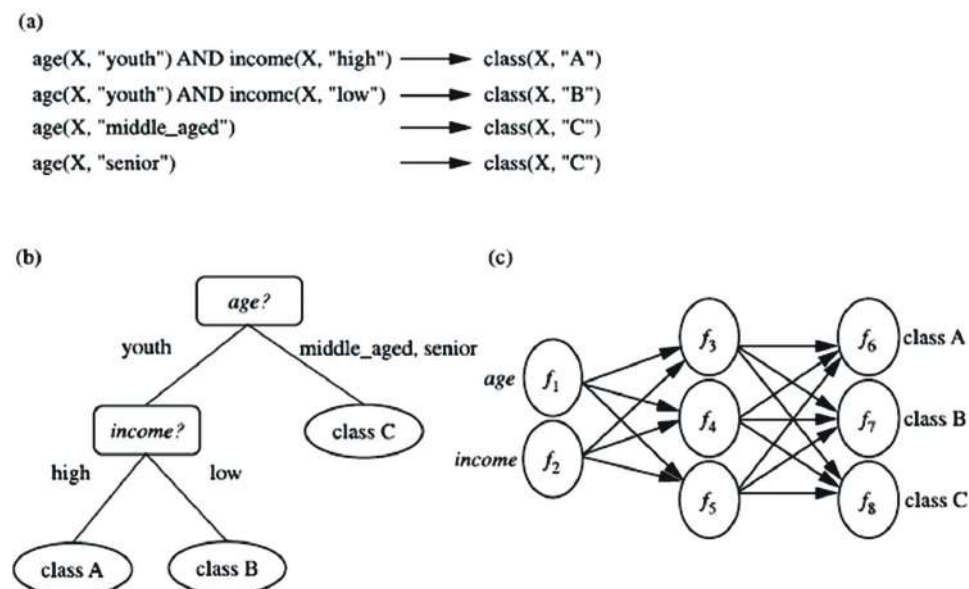


Fig. 2.4 A Classification Model can be Represented in Various forms, such as (a) IF-THEN Rules, (b) a Decision Tree, or a (c) Neural Network

NOTES

NOTES

Suppose instead, that rather than predicting categorical response labels for each store item, you would like to predict the amount of revenue that each item will generate during an upcoming sale at AllElectronics, based on previous sales data. This is an example of (numeric) prediction because the model constructed will predict a continuous-valued function, or ordered value.

Cluster Analysis

“What is cluster analysis?” Unlike classification and prediction, which analyse class-labeled data objects, clustering analyses data objects without consulting a known class label.

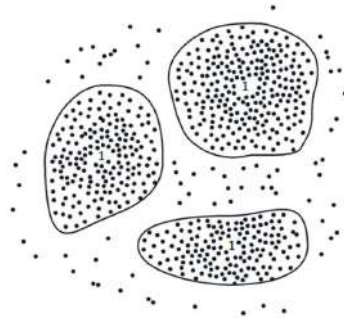


Fig. 2.5 Data Cluster

A 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 “+” (see Figure 2.5). In general, the class labels are not present in the training data simply because they are not known to begin with. Clustering can be used to generate such labels. The objects are clustered or grouped based on the principle of maximizing the intraclass similarity and minimizing the interclass similarity. That is, clusters of objects are formed so that objects within a cluster have high similarity in comparison to one another, but are very dissimilar to objects in other clusters. Each cluster that is formed can be viewed as a class of objects, from which rules can be derived. Clustering can also facilitate taxonomy formation, that is, the organization of observations into a hierarchy of classes that group similar events together.

Example 2.5 Cluster Analysis

Cluster analysis can be performed on AllElectronics customer data in order to identify homogeneous subpopulations of customers. These clusters may represent individual target groups for marketing. Figure 2.5 shows a 2-D plot of customers with respect to customer locations in a city. Three clusters of data points are evident.

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. Most data mining methods discard outliers as noise or exceptions. However, in some applications such as

fraud detection, the rare events can be more interesting than the more regularly occurring ones. The analysis of outlier data is referred to as outlier mining.

Outliers may be detected using statistical tests that assume a distribution or probability model for the data, or using distance measures where objects that are a substantial distance from any other cluster are considered outliers. Rather than using statistical or distance measures, deviation-based methods identify outliers by examining differences in the main characteristics of objects in a group.

Example 2.6 Outlier Analysis

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. Outlier values may also be detected with respect to the location and type of purchase or the purchase frequency.

Evolution Analysis

Data evolution analysis describes and models regularities or trends for objects whose behavior changes over time. Although this may include characterization, discrimination, association and correlation analysis, classification, prediction, or clustering of time-related data, distinct features of such analysis include time series data analysis, sequence or periodicity pattern matching, and similarity-based data analysis.

Example 2.7 Evolution Analysis

Suppose that you have the major stock market (time series) data of the last several years available from the New York Stock Exchange and you would like to invest in shares of high-tech industrial companies. A data mining study of stock exchange data may identify stock evolution regularities for overall stocks and for the stocks of particular companies. Such regularities may help predict future trends in stock market prices, contributing to your decision making regarding stock investments.

Check Your Progress

1. What is data mining software?
2. What is data characterization?

2.4 DATA MINING TASK PRIMITIVES

Each user will have a data mining task in mind, that is, some form of data analysis that he or she would like to have performed. A data mining task can be specified in the form of a data mining query, which is input to the data mining system. A data mining query is defined in terms of data mining task primitives. These primitives allow the user to interactively communicate with the data mining system during discovery in order to direct the mining process, or examine the findings from different

NOTES

NOTES

angles or depths. The data mining primitives specify the following, as illustrated in Figure 2.6.

- **The set of task-relevant data to be mined:** This specifies the portions of the database or the set of data in which the user is interested. This includes the database attributes or data warehouse dimensions of interest (referred to as the relevant attributes or dimensions).
- **The kind of knowledge to be mined:** This specifies the data mining functions to be performed, such as characterization, discrimination, association or correlation analysis, classification, prediction, clustering, outlier analysis, or evolution analysis.

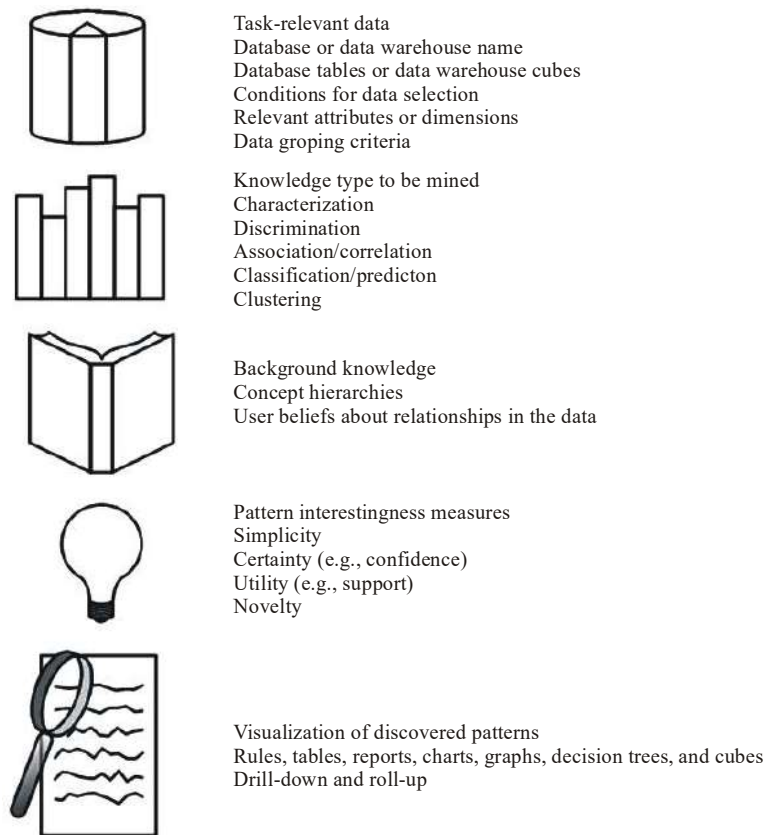


Fig. 2.6 Primitives for Specifying a Data Mining Task

- **The background knowledge to be used in the discovery process:** This knowledge about the domain to be mined is useful for guiding the knowledge discovery process and for evaluating the patterns found. Concept hierarchies are a popular form of background knowledge, which allow data to be mined at multiple levels of abstraction. An example of a concept hierarchy for the attribute (or dimension) age is shown in Figure 2.7. User beliefs regarding relationships in the data are another form of background knowledge. A concept hierarchy for the attribute (or dimension) age. The root node represents the most general abstraction level, denoted as all.

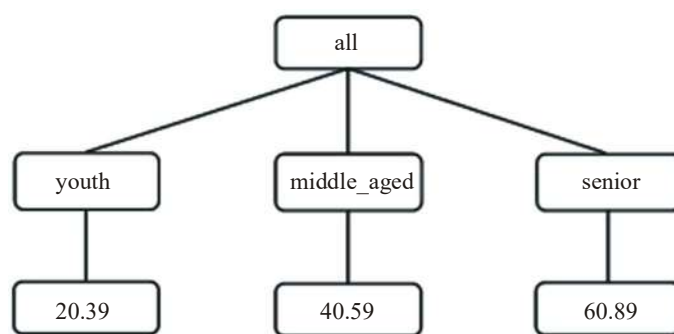


Fig. 2.7 Concept Hierarchy for the Attribute

NOTES

- The interestingness measures and thresholds for pattern evaluation:**
 They may be used to guide the mining process or, after discovery, to evaluate the discovered patterns. Different kinds of knowledge may have different interestingness measures. For example, interestingness measures for association rules include support and confidence. Rules whose support and confidence values are below user-specified thresholds are considered uninteresting.
- The expected representation for visualizing the discovered patterns:**
 This refers to the form in which discovered patterns are to be displayed, which may include rules, tables, charts, graphs, decision trees and cubes.

A data mining query language can be designed to incorporate these primitives, allowing users to flexibly interact with data mining systems. Having a data mining query language provides a foundation on which user-friendly graphical interfaces can be built.

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

Designing a comprehensive data mining language is challenging because data mining covers a wide spectrum of tasks, from data characterization to evolution analysis. Each task has different requirements. The design of an effective data mining query language requires a deep understanding of the power, limitation and underlying mechanisms of the various kinds of data mining tasks.

There are several proposals on data mining languages and standards. In this book, we use a data mining query language known as DMQL (Data Mining Query Language), which was designed as a teaching tool, based on the above primitives. Examples of its use to specify data mining queries appear throughout this book. The language adopts an SQL-like syntax, so that it can easily be integrated with the relational query language, SQL. Let's look at how it can be used to specify a data mining task.

2.5 INTEGRATION OF DATA MINING SYSTEMS

NOTES

Good system architecture will facilitate the data mining system to make best use of the software environment, accomplish data mining tasks in an efficient and timely manner, interoperate and exchange information with other information systems, be adaptable to users' diverse requirements and evolve with time.

A critical question in the design of a Data Mining (DM) system is how to integrate or couple the DM system with a database (DB) system and/or a Data Warehouse (DW) system. If a DM system works as a stand-alone system or is embedded in an application program, there are no DB or DW systems with which it has to communicate. This simple scheme is called no coupling, where the main focus of the DM design rests on developing effective and efficient algorithms for mining the available data sets. However, when a DM system works in an environment that requires it to communicate with other information system components, such as DB and DW systems, possible integration schemes include no coupling, loose coupling, semi tight coupling, and tight coupling. We examine each of these schemes, as follows:

No Coupling: No coupling means that a DM system will not utilize any function of a DB or DW system. It may fetch data from a particular source (such as, a file system), process data using some data mining algorithms and then store the mining results in another file.

Such a system, though simple, suffers from several drawbacks. First, a DB system provides a great deal of flexibility and efficiency at storing, organizing, accessing, and processing data. Without using a DB/DW system, a DM system may spend a substantial amount of time finding, collecting, cleaning, and transforming data. In DB and/or DW systems, data tend to be well organized, indexed, cleaned, integrated, or consolidated, so that finding the task-relevant, high-quality data becomes an easy task. Second, there are many tested, scalable algorithms and data structures implemented in DB and DW systems. It is feasible to realize efficient, scalable implementations using such systems. Moreover, most data have been or will be stored in DB/DW systems. Without any coupling of such systems, a DM system will need to use other tools to extract data, making it difficult to integrate such a system into an information processing environment. Thus, no coupling represents a poor design.

Loose Coupling: Loose coupling means that a DM system will use some facilities of a DB or DW system, fetching data from a data repository managed by these systems, performing data mining, and then storing the mining results either in a file or in a designated place in a database or data warehouse.

Loose coupling is better than no coupling because it can fetch any portion of data stored in databases or data warehouses by using query processing, indexing, and other system facilities. It incurs some advantages of the flexibility, efficiency,

and other features provided by such systems. However, many loosely coupled mining systems are main memory-based. Because mining does not explore data structures and query optimization methods provided by DB or DW systems, it is difficult for loose coupling to achieve high scalability and good performance with large data sets.

Semitight Coupling: Semitight coupling means that besides linking a DM system to a DB/DW system, efficient implementations of a few essential data mining primitives (identified by the analysis of frequently encountered data mining functions) can be provided in the DB/DW system. These primitives can include sorting, indexing, aggregation, histogram analysis, multiway join, and precomputation of some essential statistical measures, such as sum, count, max, min, standard deviation, and so on. Moreover, some frequently used intermediate mining results can be precomputed and stored in the DB/DW system. Because these intermediate mining results are either precomputed or can be computed efficiently, this design will enhance the performance of a DM system.

Tight Coupling: Tight coupling means that a DM system is smoothly integrated into the DB/DW system. The data mining subsystem is treated as one functional component of an information system. Data mining queries and functions are optimized based on mining query analysis, data structures, indexing schemes and query processing methods of a DB or DW system. With further technology advances, DM, DB, and DW systems will evolve and integrate together as one information system with multiple functionalities. This will provide a uniform information processing environment.

This approach is highly desirable because it facilitates efficient implementations of data mining functions, high system performance, and an integrated information processing environment.

With this analysis, it is easy to see that a data mining system should be coupled with a DB/DW system. Loose coupling, though not efficient, is better than no coupling because it uses both data and system facilities of a DB/DW system. Tight coupling is highly desirable, but its implementation is nontrivial and more research is needed in this area. Semitight coupling is a compromise between loose and tight coupling. It is important to identify commonly used data mining primitives and provide efficient implementations of such primitives in DB or DW systems.

Major Issues in Data Mining

There are major issues in data mining including mining methodology, user interaction, performance, and diverse data types. These issues are introduced below:

- **Mining Methodology and User Interaction Issues:** This includes the kinds of knowledge mined the ability to mine knowledge at multiple

NOTES

NOTES

granularities, the use of domain knowledge, ad hoc mining and knowledge visualization.

- o **Mining Different Kinds of Knowledge in Databases:** Because different users can be interested in different kinds of knowledge, data mining should cover a wide spectrum of data analysis and knowledge discovery tasks, including data characterization, discrimination, association and correlation analysis, classification, prediction, clustering, outlier analysis, and evolution analysis (which includes trend and similarity analysis). These tasks may use the same database in different ways and require the development of numerous data mining techniques.
- o **Interactive Mining of Knowledge at Multiple Levels of Abstraction:** Because it is difficult to know exactly what can be discovered within a database, the data mining process should be interactive. For databases containing a huge amount of data, appropriate sampling techniques can first be applied to facilitate interactive data exploration. Interactive mining allows users to focus the search for patterns, providing and refining data mining requests based on returned results. Specifically, knowledge should be mined by drilling down, rolling up, and pivoting through the data space and knowledge space interactively, similar to what OLAP can do on data cubes. In this way, the user can interact with the data mining system to view data and discovered patterns at multiple granularities and from different angles.
- o **Incorporation of Background Knowledge:** Background knowledge, or information regarding the domain under study, may be used to guide the discovery process and allow discovered patterns to be expressed in concise terms and at different levels of abstraction. Domain knowledge related to databases, such as integrity constraints and deduction rules, can help focus and speed up a data mining process or judge the interestingness of discovered patterns.
- o **Data Mining Query Languages and Ad Hoc Data Mining:** Relational query languages (such as, SQL) allow users to pose ad hoc queries for data retrieval. In a similar vein, high-level data mining query languages need to be developed to allow users to describe ad hoc data mining tasks by facilitating the specification of the relevant sets of data for analysis, the domain knowledge, the kinds of knowledge to be mined, and the conditions and constraints to be enforced on the discovered patterns. Such a language should be integrated with a database or data warehouse query language and optimized for efficient and flexible data mining.
- o **Presentation and Visualization of Data Mining Results:** Discovered knowledge should be expressed in high-level languages,

visual representations, or other expressive forms so that the knowledge can be easily understood and directly usable by humans. This is especially crucial if the data mining system is to be interactive. This requires the system to adopt expressive knowledge representation techniques, such as trees, tables, rules, graphs, charts, crosstabs, matrices or curves.

- o **Handling Noisy or Incomplete Data:** The data stored in a database may reflect noise, exceptional cases, or incomplete data objects. When mining data regularities, these objects may confuse the process, causing the knowledge model constructed to over fit the data. As a result, the accuracy of the discovered patterns can be poor. Data cleaning methods and data analysis methods that can handle noise are required, as well as outlier mining methods for the discovery and analysis of exceptional cases.
- o **Pattern Evaluation:** A data mining system can uncover thousands of patterns. Many of the patterns discovered may be uninteresting to the given user, either because they represent common knowledge or lack novelty. Several challenges remain regarding the development of techniques to assess the interestingness of discovered patterns, particularly with regard to subjective measures that estimate the value of patterns with respect to a given user class, based on user beliefs or expectations. The use of interestingness measures or user-specified constraints to guide the discovery process and reduce the search space is another active area of research.

Performance Issues

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

To effectively extract information from a huge amount of data in databases, data mining algorithms must be efficient and scalable. In other words, the running time of a data mining algorithm must be predictable and acceptable in large databases. From a database perspective on knowledge discovery, efficiency and scalability are key issues in the implementation of data mining systems. Many of the issues discussed above under mining methodology and user interaction must also consider efficiency and scalability. Parallel, distributed, and incremental mining algorithms: The huge size of many databases, the wide distribution of data, and the computational complexity of some data mining methods are factors motivating the development of parallel and distributed data mining algorithms. Such algorithms divide the data into partitions, which are processed in parallel. The results from the partitions are then merged. Moreover, the high cost of some data mining processes promotes the need for incremental data mining algorithms that incorporate database updates without having to mine the entire data again “from scratch.” Such algorithms perform knowledge modification incrementally to amend and strengthen what was previously discovered.

NOTES

NOTES

Issues Relating to the Diversity of Database Types

- **Handling of Relational and Complex Types of Data:** Because relational databases and data warehouses are widely used, the development of efficient and effective data mining systems for such data is important. However, other databases may contain complex data objects, hypertext and multimedia data, spatial data, temporal data, or transaction data. It is unrealistic to expect one system to mine all kinds of data, given the diversity of data types and different goals of data mining. Specific data mining systems should be constructed for mining specific kinds of data. Therefore, one may expect to have different data mining systems for different kinds of data.
- **Mining Information from Heterogeneous Databases and Global Information Systems:** Local- and wide-area computer networks (such as, the Internet) connect many sources of data, forming huge, distributed, and heterogeneous databases. The discovery of knowledge from different sources of structured, semi structured, or unstructured data with diverse data semantics poses great challenges to data mining. Data mining may help disclose high-level data regularities in multiple heterogeneous databases that are unlikely to be discovered by simple query systems and may improve information exchange and interoperability in heterogeneous databases. Web mining, which uncovers interesting knowledge about Web contents, Web structures, Web usage, and Web dynamics, becomes a very challenging and fast-evolving field in data mining.

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

Data Integration and Transformation

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

Data Integration

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

There are a number of issues to consider during data integration. Schema integration and object matching can be tricky. How can equivalent real-world entities from multiple data sources be matched up? This is referred to as the entity identification problem. For example, how can the data analyst or the computer be sure that customer id in one database and customer number in another refers to

the same attribute? Examples of metadata for each attribute include the name, meaning, data type and range of values permitted for the attribute, and null rules for handling blank, zero, or null values. Such metadata can be used to help avoid errors in schema integration. The metadata may also be used to help transform the data (e.g., where data codes for pay type in one database may be “H” and “S”, and 1 and 2 in another). Hence, this step also relates to data cleaning, as described earlier.

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

Some redundancies can be detected by correlation analysis. Given two attributes, such analysis can measure how strongly one attribute implies the other, based on the available data. For numerical attributes, we can evaluate the correlation between two attributes, A and B , by computing the correlation coefficient (also known as Pearson’s product moment coefficient, named after its inventor, Karl Pearson). This is where N is the number of tuples, a_i and b_i are the respective values of A and B in tuple i , \bar{A} and \bar{B} are the respective mean values of A and B , σ_A and σ_B are the respective standard deviations of A and B , and $\Sigma(a_i b_i)$ is the sum of the AB cross-product (that is, for each tuple, the value for A is multiplied by the value for B in that tuple). Note that $-1 \geq r_{A,B} \leq +1$. If $r_{A,B}$ is greater than 0, then A and B are positively correlated, meaning that the values of A increase as the values of B increase. The higher the value, the stronger the correlation (i.e., the more each attribute implies the other). Hence, a higher value may indicate that A (or B) may be removed as a redundancy. If the resulting value is equal to 0, then A and B are independent and there is no correlation between them. If the resulting value is less than 0, then A and B are negatively correlated, where the values of one attribute increase as the values of the other attribute decrease. This means that each attribute discourages the other. Scatter plots can also be used to view correlations between attributes.

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

Note that correlation does not imply causality. That is, if A and B are correlated, this does not necessarily imply that A causes B or that B causes A . For example, in analysing a demographic database, we may find that attributes representing the number of hospitals and the number of car thefts in a region are correlated. This does not mean that one causes the other. Both are actually causally linked to a third attribute, namely, population.

For categorical (discrete) data, a correlation relationship between two attributes, A and B , can be discovered by a χ^2 (chi-square) test. Suppose A has c

NOTES

NOTES

distinct values, namely a_1, a_2, \dots, a_c . B has r distinct values, namely b_1, b_2, \dots, b_r . The data tuples described by A and B can be shown as a contingency table, with the c values of A making up the columns and the r values of B making up the rows. Let (A_i, B_j) denote the event that attribute A takes on value a_i and attribute B takes on value b_j , that is, where $(A = a_i, B = b_j)$. Each and every possible (A_i, B_j) joint event has its own cell (or slot) in the table. The χ^2 value (also known as the Pearson χ^2 statistic) is computed as:

$$\chi^2 = \sum_{i=1}^c \sum_{j=1}^r \frac{(o_{ij} - e_{ij})^2}{e_{ij}}, \quad (2.1)$$

where o_{ij} is the observed frequency (i.e., actual count) of the joint event (A_i, B_j) and e_{ij} is the expected frequency of (A_i, B_j) , which can be computed as

$$e_{ij} = \frac{\text{count}(A = a_i) \times \text{count}(B = b_j)}{N}, \quad (2.2)$$

where N is the number of data tuples, $\text{count}(A = a_i)$ is the number of tuples having value a_i for A , and $\text{count}(B = b_j)$ is the number of tuples having value b_j for B . The sum in Equation (2.1) is computed over all of the $r \times c$ cells. Note that the cells that contribute the most to the χ^2 value are those whose actual count is very different from that expected.

Table 2.2 $A 2 \times 2$ contingency table are gender and preferred Reading correlated?

	Male	Female	Total
Fiction	250 (90)	200 (360)	450
Non-fiction	50 (210)	1000 (840)	1050
Total	300	1200	1500

The χ^2 statistic tests the hypothesis that A and B are independent. The test is based on a significance level, with $(r-1) \times (c-1)$ degrees of freedom. We will illustrate the use of this statistic in an example below. If the hypothesis can be rejected, then we say that A and B are statistically related or associated.

In addition to detecting redundancies between attributes, duplication should also be detected at the tuple level (e.g., where there are two or more identical tuples for a given unique data entry case). The use of denormalized tables (often done to improve performance by avoiding joins) is another source of data redundancy. Inconsistencies often arise between various duplicates, due to inaccurate data entry or updating some but not all of the occurrences of the data. For example, if a purchase order database contains attributes for the purchaser's name and address instead of a key to this information in a purchaser database, discrepancies can occur, such as the same purchaser's name appearing with different addresses within the purchase order database.

A third important issue in data integration is the detection and resolution of data value conflicts. For example, for the same real-world entity, attribute values from different sources may differ. This may be due to differences in representation, scaling, or encoding. For instance, a weight attribute may be stored in metric units in one system and British imperial units in another. For a hotel chain, the price of rooms in different cities may involve not only different currencies but also different services (such as free breakfast) and taxes. An attribute in one system may be recorded at a lower level of abstraction than the “same” attribute in another. For example, the total sales in one database may refer to one branch of All Electronics, while an attribute of the same name in another database may refer to the total sales for All Electronics stores in a given region.

When matching attributes from one database to another during integration, special attention must be paid to the structure of the data. This is to ensure that any attribute functional dependencies and referential constraints in the source system match those in the target system. For example, in one system, a discount may be applied to the order, whereas in another system it is applied to each individual line item within the order. If this is not caught before integration, items in the target system may be improperly discounted.

The semantic heterogeneity and structure of data pose great challenges in data integration. Careful integration of the data from multiple sources can help reduce and avoid redundancies and inconsistencies in the resulting data set. This can help improve the accuracy and speed of the subsequent mining process.

Data Transformation

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

- Smoothing, which works to remove noise from the data? Such techniques include binning, regression and clustering.
- Aggregation, where summary or aggregation operations are applied to the data. For example, the daily sales data may be aggregated so as to compute monthly and annual total amounts. This step is typically used in constructing a data cube for analysis of the data at multiple granularities.
- Generalization of the data, where low-level or “primitive” (raw) data are replaced by higher-level concepts through the use of concept hierarchies. For example, categorical attributes, like street, can be generalized to higher-level concepts, like city or country. Similarly, values for numerical attributes, like age, may be mapped to higher-level concepts, like youth, middle-aged, and senior.
- Normalization, where the attribute data are scaled so as to fall within a small specified range, such as -1.0 to 1.0 , or 0.0 to 1.0 .

NOTES

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

NOTES

An attribute is normalized by scaling its values so that they fall within a small specified range, such as 0.0 to 1.0. Normalization is particularly useful for classification algorithms involving neural networks, or distance measurements, such as nearest-neighbor classification and clustering. If using the neural network back propagation algorithm for classification mining, normalizing the input values for each attribute measured in the training tuples will help speed up the learning phase. For distance-based methods, normalization helps prevent attributes with initially large ranges (e.g., income) from out-weighting attributes with initially smaller ranges (e.g., binary attributes). There are many methods for data normalization. We study three: min-max normalization, Z score normalization and normalization by decimal scaling.

Min-max normalization performs a linear transformation on the original data. Suppose that 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 $[new_min_A, new_max_A]$ by computing

$$v' = \frac{v - min_A}{max_A - min_A} (new_max_A - new_min_A) + new_min_A.$$

Min-max normalization preserves the relationships among the original data values. It will encounter an “out-of-bounds” error if a future input case for normalization falls outside of the original data range for A .

2.6 DATA REDUCTION

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

Strategies for data reduction include the following:

1. Data cube aggregation, where aggregation operations are applied to the data in the construction of a data cube.
2. Attribute subset selection, where irrelevant, weakly relevant or redundant attributes or dimensions may be detected and removed.
3. Dimensionality reduction, where encoding mechanisms are used to reduce the data set size.
4. Numerosity reduction, where the data are replaced or estimated by alternative, smaller data representations, such as parametric models (which need store only the model parameters instead of the actual data) or non-parametric methods, such as clustering, sampling, and the use of histograms.

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

NOTES

Data Cube Aggregation

Imagine that you have collected the data for your analysis. These data consist of the *AllElectronics* sales per quarter, for the years 2002 to 2004. You are, however, interested in the annual sales (total per year), rather than the total per quarter. Thus the data can be aggregated so that the resulting data summarize the total sales per year instead of per quarter. This aggregation is illustrated in Figure 2.8. The resulting data set is smaller in volume, without loss of information necessary for the analysis task.

Year 2002	
Quarter	Sales
Q1	\$224,000
Q2	\$408,000
Q3	\$350,000
Q4	\$586,000

Year 2003	
Quarter	Sales
Q1	0
Q2	0
Q3	0
Q4	0

Year 2004	
Quarter	Sales
Q1	0
Q2	0
Q3	0
Q4	0

Year	Sales
2002	\$1,568,000
2003	\$2,356,000
2004	\$3,594,000

Fig. 2.8 Sales Data for a Given Branch of *AllElectronics* for the Years 2002 to 2004. On the Left, the Sales are Shown Per Quarter. On the Right, the Data are Aggregated to Provide the Annual Sales

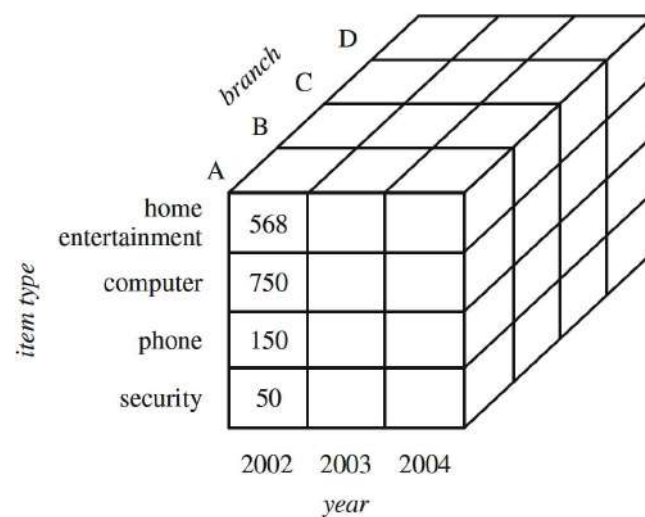


Fig. 2.9 A Data Cube for Sales at *AllElectronics*

NOTES

Hierarchies may exist for each attribute, allowing the analysis of data at multiple levels of abstraction. For example, a hierarchy for branch could allow branches to be grouped into regions, based on their address. Data cubes provide fast access to precomputed, summarized data, thereby benefiting OnLine Analytical Processing or OLAP as well as data mining.

The cube created at the lowest level of abstraction is referred to as the base cuboid. The base cuboid should correspond to an individual entity of interest, such as sales or customer. In other words, the lowest level should be usable, or useful for the analysis. A cube at the highest level of abstraction is the apex cuboid. For the sales data of Figure 2.9, the apex cuboid would give one total—the total sales for all three years, for all item types, and for all branches. Data cubes created for varying levels of abstraction are often referred to as cuboids, so that a data cube may instead refer to a lattice of cuboids. Each higher level of abstraction further reduces the resulting data size. When replying to data mining requests, the smallest available cuboid relevant to the given task should be used.

Attribute Subset Selection

Data sets for analysis may contain hundreds of attributes, many of which may be irrelevant to the mining task or redundant. For example, if the task is to classify customers as to whether or not they are likely to purchase a popular new CD at AllElectronics when notified of a sale, attributes, such as the customer's telephone number are likely to be irrelevant, unlike attribute, such as age or music taste. Although it may be possible for a domain expert to pick out some of the useful attributes, this can be a difficult and time-consuming task, especially when the behavior of the data is not well known (hence, a reason behind its analysis!). Leaving out relevant attributes or keeping irrelevant attributes may be detrimental, causing confusion for the mining algorithm employed. This can result in discovered patterns of poor quality. In addition, the added volume of irrelevant or redundant attributes can slow down the mining process.

Attribute subset selection reduces the data set size by removing irrelevant or redundant attributes (or dimensions). The goal of attribute subset selection is to find a minimum set of attributes such that the resulting probability distribution of the data classes is as close as possible to the original distribution obtained using all attributes. Mining on a reduced set of attributes has an additional benefit. It reduces the number of attributes appearing in the discovered patterns, helping to make the patterns easier to understand.

“How can we find a ‘good’ subset of the original attributes?” For n attributes, there are 2^n possible subsets. An exhaustive search for the optimal subset of attributes can be prohibitively expensive, especially as n and the number of data classes increase. Therefore, heuristic methods that explore a reduced search space are commonly used for attribute subset selection. These methods are typically greedy in that, while searching through attribute space, they always make what looks to be the best choice at the time. Their strategy is to make a locally optimal choice in the hope

that this will lead to a globally optimal solution. Such greedy methods are effective in practice and may come close to estimating an optimal solution.

The “best” and “worst” attributes are typically determined using tests of statistical significance, which assume that the attributes are independent of one another. Many other attribute evaluation measures can be used, such as the information gain measure used in building decision trees for classification.

Basic heuristic methods of attribute subset selection include the following techniques, some of which are illustrated in Figure 2.10.

NOTES

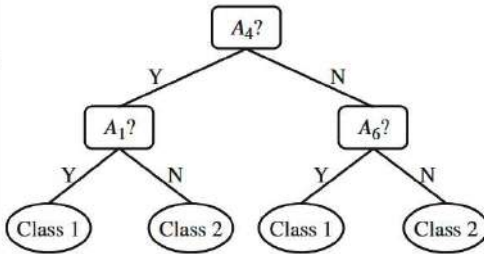
Forward selection	Backward elimination	Decision tree induction
Initial attribute set: $\{A_1, A_2, A_3, A_4, A_5, A_6\}$ Initial reduced set: $\{\}$ $\Rightarrow \{A_1\}$ $\Rightarrow \{A_1, A_4\}$ \Rightarrow Reduced attribute set: $\{A_1, A_4, A_6\}$	Initial attribute set: $\{A_1, A_2, A_3, A_4, A_5,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

Fig. 2.10 Greedy (Heuristic) Methods for Attribute Subset Selection

1. **Stepwise Forward Selection:** The procedure starts with an empty set of attributes as the reduced set. The best of the original attributes is determined and added to the reduced set. At each subsequent iteration or step, the best of the remaining original attributes is added to the set.
2. **Stepwise Backward Elimination:** The procedure starts with the full set of attributes. At each step, it removes the worst attribute remaining in the set.
3. **Combination of Forward Selection and Backward Elimination:** The stepwise forward selection and backward elimination methods can be combined so that, at each step, the procedure selects the best attribute and removes the worst from among the remaining attributes.
4. **Decision Tree Induction:** Decision tree algorithms, such as ID3, C4.5, and CART, were originally intended for classification. 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

NOTES

assumed to be irrelevant. The set of attributes appearing in the tree form the reduced subset of attributes.

The stopping criteria for the methods may vary. The procedure may employ a threshold on the measure used to determine when to stop the attribute selection process.

Dimensionality Reduction

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

Wavelet Transforms

The Discrete Wavelet Transform (DWT) is a linear signal processing technique that, when applied to a data vector X , transforms it to a numerically different vector, X' , of wavelet coefficients. The two vectors are of the same length. When applying this technique to data reduction, we consider each tuple as an n -dimensional data vector, that is, $X = (x_1, x_2, \dots, x_n)$, depicting n measurements made on the tuple from n database attributes. “How can this technique be useful for data reduction if the wavelet transformed data are of the same length as the original data?” The usefulness lies in the fact that the wavelet transformed data can be truncated. A compressed approximation of the data can be retained by storing only a small fraction of the strongest of the wavelet coefficients. For example, all wavelet coefficients larger than some user-specified threshold can be retained. All other coefficients are set to 0. The resulting data representation is therefore very sparse, so that operations that can take advantage of data sparsely are computationally very fast if performed in wavelet space. The technique also works to remove noise without smoothing out the main features of the data, making it effective for data cleaning as well. Given a set of coefficients, an approximation of the original data can be constructed by applying the inverse of the DWT used.

The DWT is closely related to the Discrete Fourier Transform (DFT), a signal processing technique involving sines and cosines. In general, however, the DWT achieves better lossy compression. That is, if the same number of coefficients is retained for a DWT and a DFT of a given data vector, the DWT version will provide a more accurate approximation of the original data. Hence, for an equivalent approximation, the DWT requires less space than the DFT. Unlike the DFT, wavelets are quite localized in space, contributing to the conservation of local detail.

There is only one DFT, yet there are several families of DWTs. Figure 2.11 shows some wavelet families. Popular wavelet transforms include the Haar-2, Daubechies-4, and Daubechies-6 transforms. The general procedure for applying a discrete wavelet transform uses a hierarchical pyramid algorithm that halves the data at each iteration, resulting in fast computational speed. The method is as follows:

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

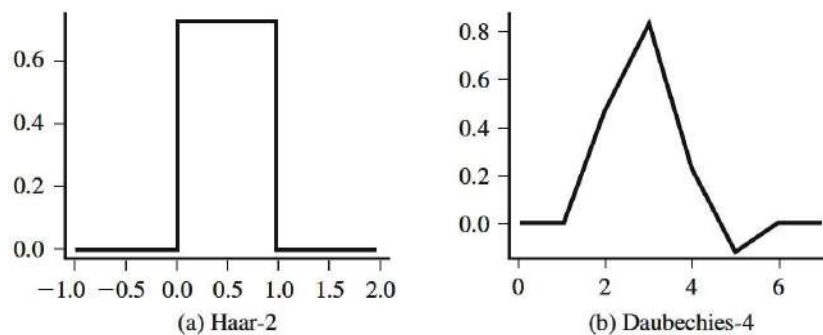


Fig. 2.11 Examples of Wavelet Families. The Number Next to a Wavelet Name is the Number of Vanishing Moments of the Wavelet. This is a Set of Mathematical Relationships that the Coefficients must Satisfy and is Related to the Number of Coefficients.

Equivalently, a matrix multiplication can be applied to the input data in order to obtain the wavelet coefficients, where the matrix used depends on the given DWT. The matrix must be orthonormal, meaning that the columns are unit vectors and are mutually orthogonal, so that the matrix inverse is just its transpose. This property allows the reconstruction of the data from the smooth and smooth-difference data sets. By factoring the matrix used into a product of a few sparse matrices, the resulting “fast DWT” algorithm has a complexity of $O(n)$ for an input vector of length n .

NOTES

NOTES

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

Principal Components Analysis

In this subsection, we provide an intuitive introduction to principal components analysis as a method of dimensionality reduction. A detailed theoretical explanation is beyond the scope of this book.

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

The basic procedure is as follows:

1. The input data are normalized, so that each attribute falls within the same range. This step helps ensure that attributes with large domains will not dominate attributes with smaller domains.
2. PCA computes k orthonormal vectors that provide a basis for the normalized input data. These are unit vectors that each point in a direction perpendicular to the others. These vectors are referred to as the principal components. The input data are a linear combination of the principal components.
3. The principal components are sorted in order of decreasing “significance” or strength. The principal components essentially serve as a new set of axes for the data, providing important information about variance. That is, the sorted axes are such that the first axis shows the most variance among the data, the second axis shows the next highest variance, and so on. For example, Figure 2.12 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. Y_1 and Y_2 are the first two principal components for the given data.

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

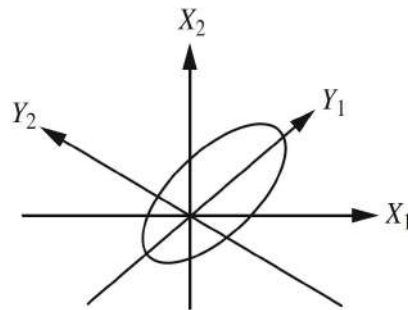


Fig. 2.12 Principal Components Analysis

PCA is computationally inexpensive, can be applied to ordered and unordered attributes, and can handle sparse data and skewed data. Multidimensional data of more than two dimensions can be handled by reducing the problem to two dimensions. Principal components may be used as inputs to multiple regression and cluster analysis. In comparison with wavelet transforms, PCA tends to be better at handling sparse data, whereas wavelet transforms are more suitable for data of high dimensionality.

Numerosity Reduction

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

Let us look at each of the numerosity reduction techniques mentioned above.

Regression and Log-Linear Models

Regression and log-linear models can be used to approximate the given data. In (simple) linear regression, the data are modeled to fit a straight line. For example, a random variable, y (called a response variable), can be modeled as a linear function of another random variable, x (called a predictor variable), with the equation

$$y = wx + b, \quad (2.3)$$

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

NOTES

NOTES

coefficients), specify the slope of the line and the Y -intercept, respectively. These coefficients can be solved for by the method of least squares, which minimizes the error between the actual line separating the data and the estimate of the line. Multiple linear regressions is an extension of (simple) linear regression, which allows a response variable, y , to be modeled as a linear function of two or more predictor variables.

Log-linear models approximate discrete multidimensional probability distributions. Given a set of tuples in n dimensions (e.g., described by n attributes), we can consider each tuple as a point in an n -dimensional space. Log-linear models can be used to estimate the probability of each point in a multidimensional space for a set of discretized attributes, based on a smaller subset of dimensional combinations. This allows a higher-dimensional data space to be constructed from lower-dimensional spaces. Log-linear models are therefore also useful for dimensionality reduction (since the lower-dimensional points together typically occupy less space than the original data points) and data smoothing (since aggregate estimates in the lower-dimensional space are less subject to sampling variations than the estimates in the higher-dimensional space).

Regression and log-linear models can both be used on sparse data, although their application may be limited. While both methods can handle skewed data, regression does exceptionally well. Regression can be computationally intensive when applied to high-dimensional data, whereas log-linear models show good scalability for up to 10 or so dimensions.

Histograms

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

Check Your Progress

3. What is the meaning of No coupling?
4. What do you understand by redundancy?
5. What is dimensionality reduction?

2.7 DATA MINING APPLICATIONS

Data mining has emerged as a powerful tool for the decision-makers. Rapid developments in hardware and software resources, continuous improvement in their price-performance ratio and their availability are some of the reasons for encouraging organizations to invest in their data mining infrastructure. Affordability is a strong reason for managers to consider applying data mining in every possible domain.

The applications can, therefore, be many, a few of which are elaborated in the following subsections:

- Retail
- Telecom
- Life science
- Other scientific applications
- Intrusion detection

Data Mining in Financial Data Applications

Banks and financial institutions offer different services to their customers including maintenance of accounts, offering ATM facilities, credits and investment facilities.

- **Designing of Data Warehouse for Multidimensional Analysis:** Multidimensional cubes are created to provide different OLAP reports. Such reports can generate reports for the region, sector, etc.
- **Loan Payment and Customer Credit:** It helps to carry out loan payment prediction and customers' credit status to ascertain the creditworthiness of customers.
- **Classification and Clustering:** They may be employed to identify customer groups and for target marketing.
- **Financial Irregularities:** It requires collecting data from various sources and then integrating it. Data can then be analysed by clustering tools for grouping, by classification tools for removing unrelated attributes and by outlier analysis for detecting unusual transactions.
- **Stock Forecasting:** There are so many software applications in the market that uses data mining techniques for stock predictions. One of the application is shown in the following Figure 2.13.



Fig. 2.13 Application of Stock Forecasting

NOTES

In the above Figure 2.13, there are two lines, representing real and predicted stock values. One of the stock prediction applications is NetProphet. This application uses neural network.

NOTES

In banking system, data mining technology is used for fraud detection. The banks that use data mining are: Bank of America, First USA Bank, Headlands Mortgage Company, FCC National Bank, Federal Home Loan Mortgage Corporation, Wells Fargo Bank, Nations- Bank Services, Mellon Bank N.A., Advanta Mortgage Corporation, Chemical Bank, Chevy Chase Bank, U.S. Bancorp, and USAA Federal Savings Bank.

Data Mining in the Retail Industry

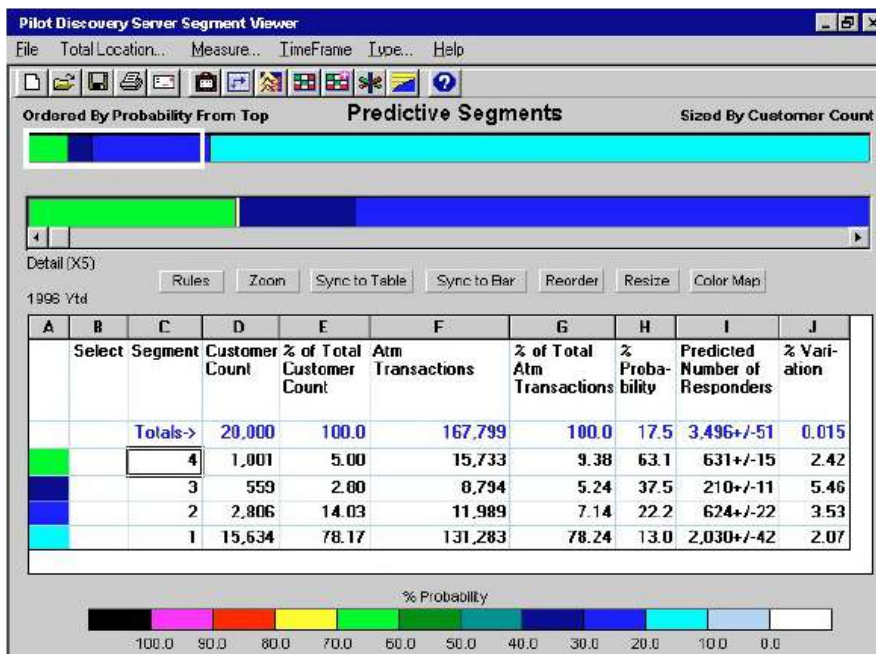
The retail industry has attracted a large number of data mining applications as most transactions are recorded and are available for useful analysis. The useful areas of interest are customers' buying patterns, customer retention and satisfaction.

- **Data on Sales:** It requires the level of detail and dimensions to be identified.
- **Design and Construction of Data Warehouse:** A data cube should be constructed to support multidimensional (OLAP) analysis.
- **Analysis of Sales Campaigns:** In this analysis, effectiveness of sales campaigns can be carried out.
- **Customer Retention:** An effective customer relationship management scheme can be built and customer loyalty attained.
- **Purchase Recommendation:** Association rules can be applied for identifying the requirement of items to be procured.

Other Data Mining Applications in Retail Industry

Retailers improve the decision-support processes which lead directly to improved efficiency in inventory management and financial forecasting. The recent adoption of data warehousing by retailers has given them a better opportunity to take advantage of data mining. It stores vast amounts of point-of-sale data that is information rich. This application directly focuses on marketing. Generally, most retailers including cataloguers, consumer retail chains, publishers, business-to-business marketers and packaged good manufacturers use direct good marketing. Today, more than 500 companies rely on data mining in direct marketing.

Direct marketers are often concerned about customer segmentation, which is a clustering problem in data mining. Many vendors offer customer segmentation packages like the one shown in the following Figure 2.14.



NOTES

Fig. 2.14 Customer Segmentation Software Courtesy: Pilot Software

Pilot software uses the customer segmentation program. It helps in direct-mailing campaigns, as shown following (see Figure 2.15).

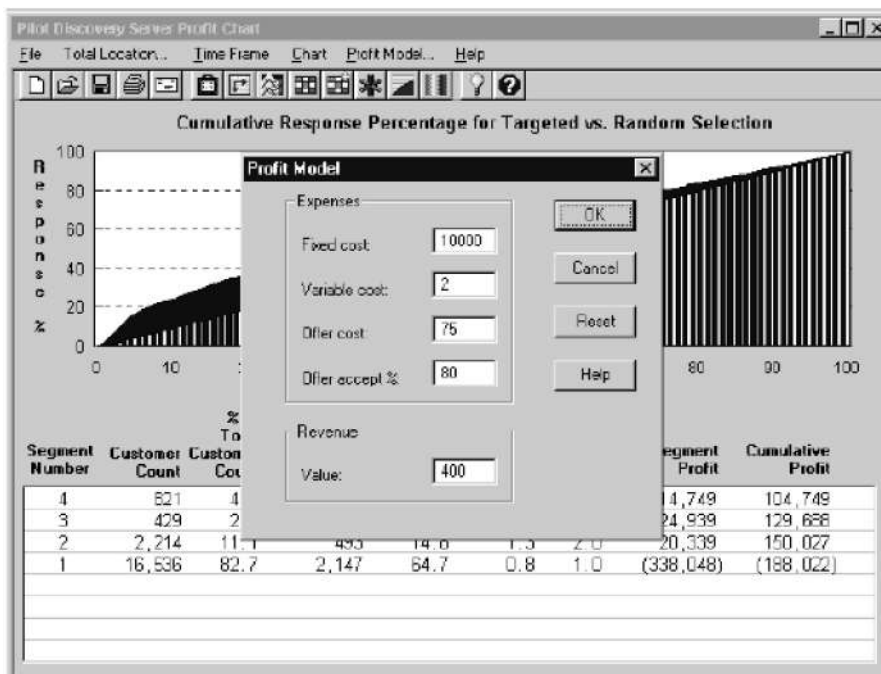


Fig. 2.15 An Application for Direct Marketing Campaign Courtesy: Pilot Software

NOTES

Other Types of Studies Related to Retail Data Mining

Retailers keep an interest on different types of data mining studies in the area of marketing and direct marketing campaign. Retailers seek answers to the following questions from information generated from various data mining models:

- Number of potential customers capable of spending over long durations.
- Customer purchasing behaviour related to frequency of purchase.
- Most effective way of advertising to cover certain segments.
- Most effective media of reaching customers.
- The correct time to send promotional mails.

In the case of merchandisers, they focus on the following questions:

- What types of customers are buying specific products?
- What determines the best product mix to sell on a regional level?
- What are the latest product trends?
- When is a merchandise department saturated?
- When is a customer most likely to buy?
- What types of products can be sold together?

Basic Principles in Data Mining in Retail

The VAL (Value, Activity and Loyalty) method uses transactional data to measure the probability of activeness of customers. It calculates how many active customers reside in the customer database, determines the customer-base attrition rate and forecasts revenues from the currently active customers. It also extracts useful information, such as average customer lifetime (how long customers are expected to remain active), and average repurchase rates. This methodology is better than the classic RFM (Recency, Frequency and Monetary) analysis because it is forward-looking, rather than backward-looking, and produces more actionable results. For example, segmenting the customer base by activity and value can suggest marketing strategies to stimulate those customers who are marginally active, but who have high expected value. Churn rates in conjunction with revenue forecasts can be used to determine what customer acquisition rate is required for meeting revenue or profitability goals.

Churn rates over time can also be used to identify and counteract seasonal periods that might trigger inactivity.

Today, transactional data is a very valuable asset to retailers because of the actionable information it can generate if it is analysed and mined carefully. These days, due to the level of computing power and affordability, mining transactional data is no longer reserved for large retailers.

Data Mining in the Telecommunications Industry

Local and long distance services through mobile and landlines by telecom-service providers have attained astronomical proportion. Along with it has risen the need to provide the best possible service to their customers, have effective customer relationship management and prevent their attrition to other competitors. It has also become a legal necessity for service providers to keep transaction details of each customer and if required, identify calls made or received by them. Data mining provides opportunities for better business understanding, identifying patterns, better utilization of resources and quality improvement.

NOTES

- Multidimensional Analysis of Telecom Data:** Systems may be built based on data cubes for OLAP and Online Analytical Mining (OLAM) operations. OLAP enabled performing standard operations, such as slicing, dicing, drilling and pivoting on data held in MDDB. In addition, OLAM combines the features of OLAP with data mining tasks and provides online interaction with the user (see Figure 2.16).

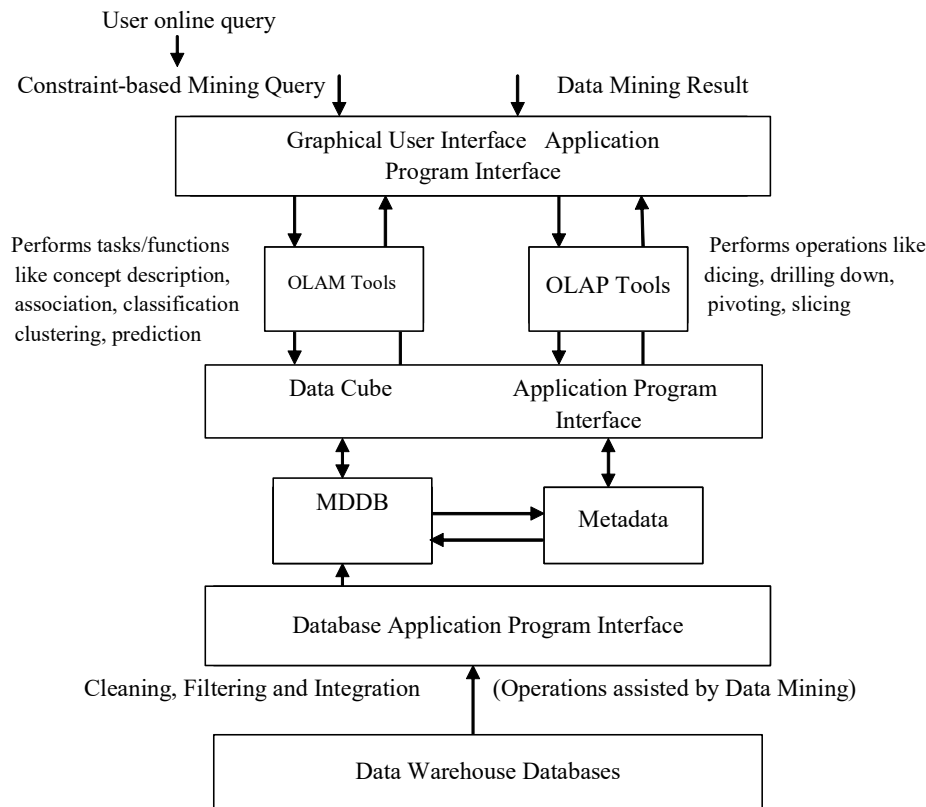


Fig. 2.16 Multidimensional Analysis of Telecom Data

- Fraudulent Calls:** Potentially fraudulent customers are identified through outlier analysis for adopting wrong practices, such as accessing other customers' accounts.

NOTES

- **Multidimensional Association and Sequential Pattern Analysis:** These are available through drilling down and drilling up operations and provide an approach to promote telecom services to identified groups.
- **Visualization Tools:** These provide an effective way of data analysis of telecom information.

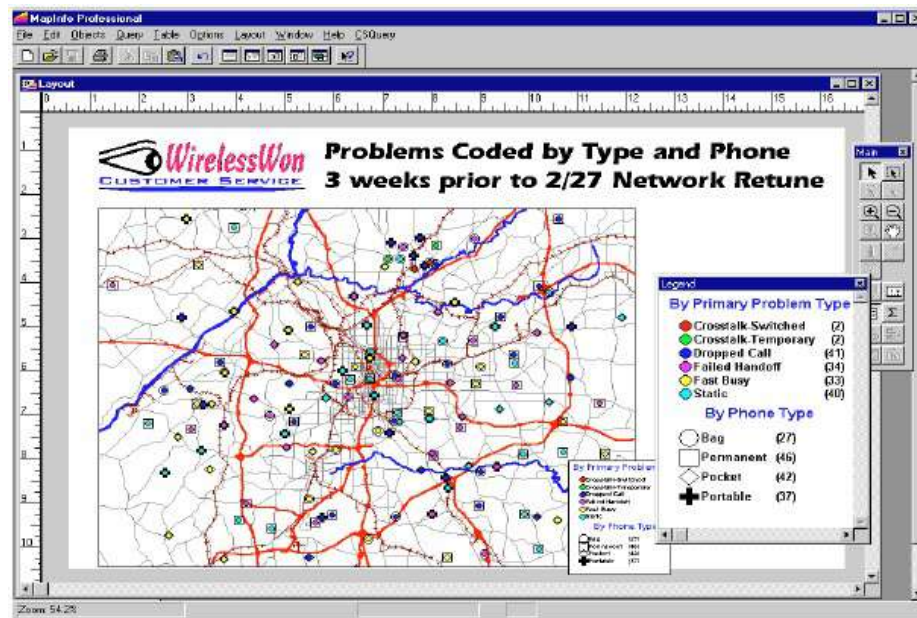


Fig. 2.17 A Map of Wireless Telecommunications Network Pinpointing Dropped Calls

Types of Studies in Telecommunications

The telecommunications industry applies data mining principles for the following:

- Recognizing and predicting cellular frauds.
- Developing techniques of creating and retaining loyal customers such that they do not leave even if reduced rates and special offers are made by competitors.
- Finding the spending time of customers.
- Focussing on the characteristics of a customer to determine whether he will be profitable or unprofitable.
- Predicting whether customers will buy additional products, such as cellular service, call waiting or basic services.
- Finding out the characteristics that make a customer decide on high-risk investments, such as investing in new fiber optic lines.
- Finding products and services which yield the highest amount of profit.
- Finding the characteristics that differentiate products from those of competitors.

- Finding the set of characteristics which indicate companies or customers who will increase their line usage.

Data Mining in Biological Data Analysis

Biomedical research has been growing at a rapid pace, developing new drugs. A study of human genome by identifying the sequencing patterns is another area of interest requiring DNA analysis leading to discovery of genetic disorders. As biological databases are extremely large, data mining is the apt technology to draw inferences and reach results. The essential features of such exploration are:

- Biological databases are heterogeneous in nature and are available through diverse sources. Data mining and warehousing technology integrates them and presents the data for further investigation and research.
- Comparison of DNA sequences can be done effectively to differentiate between healthy and diseased tissues.
- Association analysis helps in identifying and comparing genes occurring together.
- Path analysis permits establishing the connection between genes and diseases.
- Visualization tools such as graphs, trees, chains and cuboids help in presenting and understanding complex structures in a visual form.

Biclustering Algorithm in Biological Data Analysis

The term ‘Cluster’ refers to similar type of data in a set. As an example in the marketing field, sales executives deal differently with different types of customers. For example, they pay more attention to customers who are influenced by the pricing and other features of a product. Such customers are grouped together. Some customers do not give much importance to the price of products; they are referred to as ‘cool customers’ (see Figure 2.18).

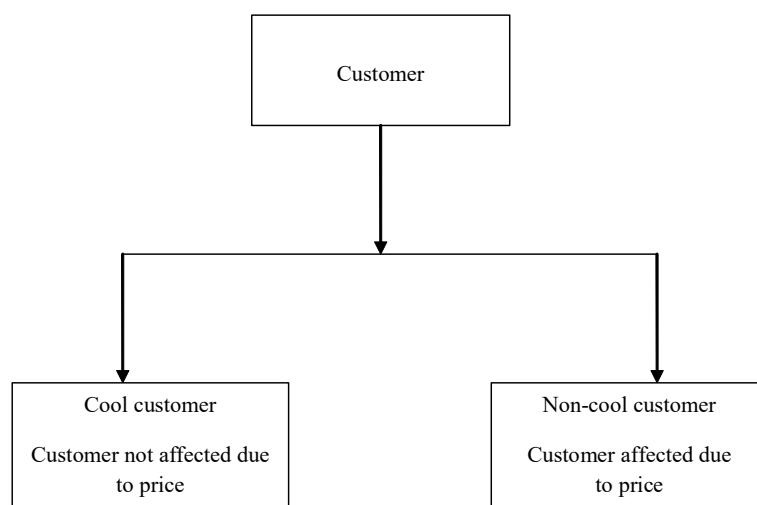


Fig. 2.18 Grouping of Customers Based on Their Preferences

NOTES

NOTES

Now, standard clustering algorithm uses all variables for other clusters but it is unable to detect the structure of data. We can do this with the help of biclustering. Biclustering is the simultaneous clustering of columns and rows in a data set. Each cluster is defined by a different subset of variables; these subsets can of course be overlapping. It contains a comprehensive collection of bicluster algorithms, preprocessing methods, and validation and visualization techniques for bicluster results.

Biclustering is a two-mode clustering technique of data mining which shows simultaneous clustering of elements of a matrix. Let there be an $m \times n$ matrix having m rows and n columns; then, a biclustering algorithm generates biclusters containing a subset of rows and exhibiting similar behaviour across a subset of columns or vice versa.

Types of Biclusters:

In general, there are four types, which show (see Figure 2.19):

- (i) Constant values (m)
- (ii) Constant values, either on rows or on columns (m, n)
- (iii) Coherent values (p, q)
- (iv) Bicluster with coherent evaluation

Description of Biclustering Algorithm: Text mining or classification uses biclustering. This is called co-clustering. Here, vector form as a Matrix D is used to represent text in rows and words in columns. Matrix elements D_{ij} signify the occurrence of word j in document i . To discover blocks corresponding to a group of documents (rows) in the matrix, the co-clustering algorithms are applied which are:

2.0	2.0	2.0	2.0
2.0	2.0	2.0	2.0
2.0	2.0	2.0	2.0
2.0	2.0	2.0	2.0

(1) Constant Bicluster

1.0	1.0	1.0	1.0
2.0	2.0	2.0	2.0
4.0	4.0	4.0	4.0
5.0	5.0	5.0	5.0

(2) Constant Rows

1.0	1.0	1.0	1.0
2.0	2.0	2.0	2.0
4.0	4.0	4.0	4.0
5.0	5.0	5.0	5.0

(3) Constant Columns

1	2	3	4
5	3	4	2
5	6	3	2
5	1	2	3

(4) Coherent Values (Additive)

1.0	3.0	0.4	0.3
2.0	1.0	4.0	2.0
3.0	2.0	1.0	0.1
1.0	2.0	1.0	5.0

(5) Coherent Values (Multiplication)

Fig. 2.19 Examples of Different Type of Clusters

Complexity of Biclusters: This is dependent on the exact problem formulation. Complexity depends particularly on the merit function that is used for evaluation of quality of a given bicluster. NP-complete problems are its variants that require heuristics to find a short-cut for calculating large computational values.

Biclustering Application

This technique finds application when data is available in the form of a real-valued matrix A , with set of values a_{ij} representing a relation between rows i and columns j . Gene expression matrices can be given as example. These require data modelled as weighted bipartite graph. This technique can be used to identify sub-matrices formed using a subset of rows and a subset of columns having coherence properties.

Clinical samples create large data sets and are ideal for use in biclustering techniques. Biclustering is used in other areas, such as in information retrieval, text mining, target marketing, database research, and so on.

Data Mining in Other Scientific Applications

Scientific, engineering and social science applications require well-established statistical techniques for data analysis of available numeric data. The data is predominantly prevalent in these disciplines and requires the following techniques to be applied based on the requirement:

- Regression
- Linear models
- Regression trees
- Variance models
- Factor analysis
- **Discriminant Analysis:** It is a statistical technique used for classifying dependent variables between two or more categories. It has a regression technique, too, used to predict the value of categorical dependent variables. In such an analysis, values of two categories are predicted. When dependent variables have more than two categories, it is taken as an extension of simple discriminant analysis and referred to as multiple discriminant analysis. Multiple discriminant analysis is considered similar to MANOVA, as many assumptions and tests are similar in both these tests. The Wilks' lambda or F test is used to test the significance of discriminant model as a whole. If the F test shows the overall significance of the model, then the individual variables are accessed to see which variable will move the significance from the group mean. Discriminant analysis also assumes several assumptions, such as multiple linear regressions, linear relationships, homoscedastic relationships, untruncated interval data, and so on. Logistic regression is an alternative technique and is frequently used in place of discriminant analysis when the data does not meet the assumptions.

NOTES

- Time series
- Survival analysis
- Quality control

NOTES

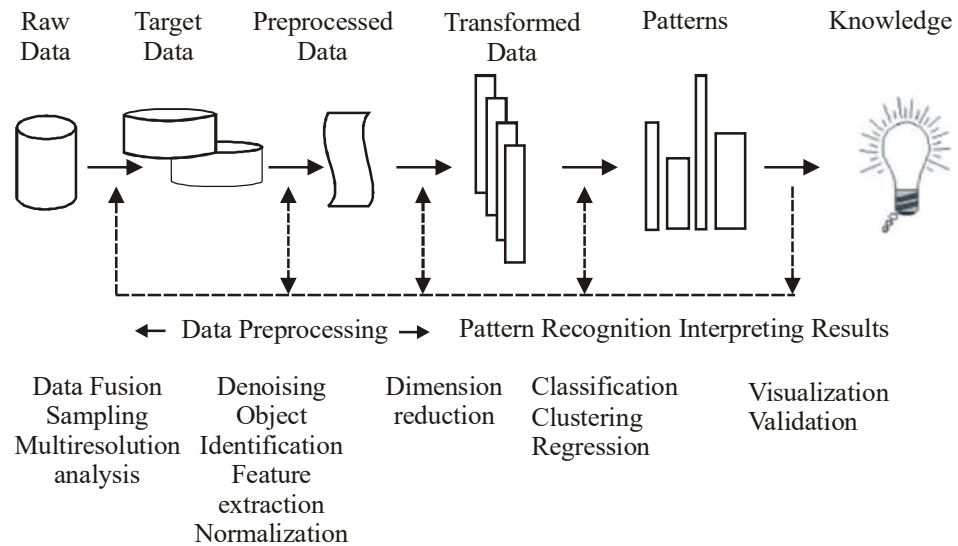


Fig. 2.20 End-to-End Scientific Data Mining Process

The following tables show various analytical information technologies and application examples in data mining technologies:

Table 1: Analytical Information Technologies

<p><u>Data Mining Technologies</u></p> <ul style="list-style-type: none"> • Association, correlation, clustering, classification, regression, database knowledge discovery • Signal and image processing, Nonlinear systems analysis, time series and spatial statistics, time and frequency domain analysis • Expert systems, Case-based reasoning, System dynamics • Econometrics, Management Science <p><u>Decision Support Systems</u></p> <ul style="list-style-type: none"> • Automated Analysis and Modeling <ul style="list-style-type: none"> ◦ Operations Research ◦ Data Assimilation, Estimation and Tracking • Human Computer Interaction <ul style="list-style-type: none"> ◦ Multidimensional OLAP and spreadsheets ◦ Allocation and consolidation engine, alerts ◦ Business workflows and data sharing

Table 2: Application Examples

<p><u>Science and Engineering</u></p> <ul style="list-style-type: none"> • Bio-Informatics • Genomics • Hydrology, Hydrometeorology • Weather Prediction • Climate Change Science • Remote Sensing • Smart Infrastructures • Sensor Technologies • Land-use, Urban Planning • Materials Science <p><u>Business and Economics</u></p> <ul style="list-style-type: none"> • Financial Planning • Risk Analysis • Supply Chain Planning • Marketing Plans • Text and Video Mining • Handwriting/Speech Recognition • Image and Pattern Recognition • Long-range Economic Planning • Homeland Security

Data Mining in Intrusion Detection

Intrusion may be either physical intrusion or unauthorized access of a database by a person.

In the first case, the problem falls into predictive model and classification task or function. Classification maps data into certain groups or classes. It is also considered as supervised learning, as the classes are defined prior to examining the data. In the case of physical intrusion, the solution lies in pattern recognition which is an extension of classification. The face of each person is scanned at the entry point and its basic characteristics—the shape of the head or mouth, the distance between eyes are recorded. These are matched with the pre-recorded patterns of known offenders and potential criminals or identified terrorists.

In the case of unauthorized net access, an outlier analysis is carried out to detect fraudulent transactions.

Network Intrusion Detection

Intrusion detection monitors the system regularly. It uses a computer network with instrumentation for data collection. Pattern-based software monitors the system. It senses the network traffic, looking for a match with the saved pattern, raising ‘alarms’ whenever matches are found. Such alarms are to be identified for response. It has to be responded when a serious event occurs. The response may be different in different cases. This may be to shut down a network partly or just noting unusual traffic for future reference. Such detection systems work well in small networks. But in organizations with large, complex networks, the number of alarms is very high and requires review.

R.L. Grossman defines data mining, that is, concerned with uncovering patterns, anomalies, associations, other significant data structures and events related to data. At times, we use this term as knowledge discovery in data.

Now, we come to the point ‘How do current Intrusion Detection System or IDS detect intrusions?’ It means how data mining helps in intrusion detection. The different approaches to intrusion detection are:

- Detection on misuse
- Detection for anomaly

Detection on Misuse: This detection is based on a known pattern for malicious activities, called signatures.

Detection for Anomaly: This approach identifies malicious activities by noting deviations from the normal patterns of network traffic.

NOTES

NOTES

Nowadays, signatures are used to detect intrusions.

- **Variants:** Signatures are developed to counter new vulnerabilities. For success, a signature must be unique. One limitation is that these codes can be easily changed.
- **False Positives:** There are cases of generating false positives by IDS. A critical problem arises on the extent to which filtering is possible without missing an attack.
- **False Negatives:** This is detection of attacks when known signatures are absent. This gives false negatives and the IDS gives no alert even if there is an intrusion.
- **Data Overload:** This is extremely important to decide how much data can be effectively and efficiently analysed. The possibility for logs to access millions of records everyday depends on the detection tools employed by a company and its size.

Data mining provides better intrusion detection since it focuses on anomaly detection. Data mining can distinguish an activity related to attack from common everyday traffic on the network.

2.8 TRENDS IN DATA MINING

Data mining and data warehousing are evolving technologies and are constantly being upgraded or improved. Their potential and availability as powerful decision-making tools have become a common knowledge to all decision-makers. Their deep impact on the industry and the society as a whole suggests the emergence of new trends. Predicting these trends and finding or exploring solutions for them would give a certain edge to individuals or organizations who are bound to get an early-bird advantage over others.

The trends manifested, thus, and the challenges they pose before the technical specialists and potential users are explained below:

- **Application Exploration:** Use of data mining systems is gaining popularity and is now going beyond the realm of providing a competitive advantage to the organization. Applications are, therefore, going beyond the procurement of generic systems for common use, to custom-made applications.
- **Scalable Data Mining Products:** As the data being processed is growing rapidly, designing scalable algorithms has become a necessity. The overall efficiency is also increased by providing users a control over constraints and optimizing searches. This is termed as constraint-based mining.
- **Integration of Components:** Data mining and warehousing systems obtain data from diverse sources. While the data is integrated and stored in the

data warehouse database, the data mining and other query systems have to be integrated too as a unified framework.

- **Standardization of Languages:** It is desirable to provide a standard data mining language to improve interoperability and collaborative development of data mining solutions.
- **Visual Data Mining:** This utilizes the techniques including graphical, geometric, icon based, pixel-based hierarchical and hybrid methods for discovering knowledge using great amounts of data.
- **Mining Complex Data:** Complex types of data including geospatial, multimedia, text and time series do not have adequate data mining techniques and constitute a research frontier.
- **Web Mining:** Considering the vast information available on the Web, new data mining techniques are required to be identified for its full utilization.
- **Privacy Protection and Information Security:** As the data warehouses supporting data mining are open to access through network, it is essential to build additional safeguards to doubly ensure privacy of data stored and make information better secured .

NOTES

Profitable Applications in Data Mining

Nowadays, a wide range of companies has deployed successful applications of data mining. Early adopters of this technology have tended to be in information-intensive industries, such as financial services and direct mail marketing. This technology is applicable to any company looking to leverage a large data warehouse to better manage their customer relationships.

There are two critical factors for success with data mining:

- (i) A large, well-integrated data warehouse.
- (ii) A well-defined understanding of the business process within which data mining is to be applied (such as, customer prospecting, retention, campaign management, and so on).

Some successful application areas include the following:

- A pharmaceutical company investigates its recent sales force activity and the results to improve targeting of high-value physicians and determine which marketing activities will have the greatest impact in the next few months. The data analysis identifies competitor market activities as well as information about the local health care systems. Based on the results, the company can distribute the sales force via a wide-area network that enables the representatives to review the recommendations from the perspective of the key attributes in the decision process. The ongoing, dynamic analysis of the data warehouse allows best practices from throughout the organization to be applied in specific sales situations.

NOTES

- Data mining also works on credit card assessment. A credit card company can leverage its vast warehouse of customer transaction data to identify customers most likely to be interested in a new credit product. With the help of small test mailing, the attributes of customers with an affinity for the product can be identified. Currently, projects have indicated more than a twenty-fold decrease in costs for targeted mailing campaigns over conventional approaches.
- A diversified transportation company with a large direct sales force can apply data mining to identify the best prospects for its services. Using data mining to analyse its own customer experience, this company can build a unique segmentation identifying the attributes of high-value prospects. Applying this segmentation to a general business database can yield a prioritized list of prospects by region.
- A large consumer package goods company use data mining approaches to improve its sales process to retailers. Data which comes from consumer panels, shipments and competitor activity can be applied to understand the reasons for brand and store switching. Based on this analysis, the manufacturer can select promotional strategies that best reach their target customer segments.

Each of these examples has a clear common ground. They leverage the knowledge about customers implicit in a data warehouse to reduce costs and improve the value of customer relationships. These organizations can then focus their efforts on the most important (profitable) customers and prospects, and design targeted marketing strategies to best reach them.

Check Your Progress

6. What is Biclustering?
7. How can diversified transportation company benefit from data mining?

2.9 ANSWERS TO CHECK YOUR PROGRESS QUESTIONS

1. Data mining software is an analytical tool used for data analysis. Users can analyse data from different angles, classify/categorize it, and then summarize the identified relationships.
2. Data characterization is a summarization of the general characteristics or features of a target class of data. The data corresponding to the user-specified class are typically collected by a database query.
3. No coupling means that a DM system will not utilize any function of a DB or DW system. It may fetch data from a particular source (such as, a file system), process data using some data mining algorithms and then store the mining results in another file.

4. An attribute (such as annual revenue, for instance) may be redundant if it can be “derived” from another attribute or set of attributes. Inconsistencies in attribute or dimension naming can also cause redundancies in the resulting data set.
5. In dimensionality reduction, data encoding or transformations are applied so as to obtain a reduced or “compressed” representation of the original data. If the original data can be reconstructed from the compressed data without any loss of information, the data reduction is called lossless.
6. Biclustering is a two-mode clustering technique of data mining which shows simultaneous clustering of elements of a matrix.
7. A diversified transportation company with a large direct sales force can apply data mining to identify the best prospects for its services. Using data mining to analyse its own customer experience, this company can build a unique segmentation identifying the attributes of high-value prospects

NOTES

2.10 SUMMARY

- Data mining means locating, identifying and finding unforeseen information from a large database. The information is one which is interesting to the end-user.
- A data pattern discovered through a database search is considered interesting, if it is easily understood, is valid on a new or test data with some degree of uncertainty, potentially useful and is novel.
- Data mining is devoted specifically to the processes involved in the extraction of useful information by applying specific techniques based on certain knowledge domains.
- Knowledge discovery can be subdivided into five specific steps which are performed repetitively till the desired result is reached, and one of them is data mining.
 - (i) Data processing comprising data selection, data cleaning and data integration.
 - (ii) Data transformation and organization in a form ready for fast access.
 - (iii) Data Mining (DM) engine and other techniques, such as OLAP or Online Transaction Processing (OLTP) for searching and extraction.
 - (iv) Knowledge presentation methods through Graphical User Interface (GUI).
 - (v) Analyzing results and assimilating them in a knowledge domain.
- Operational database supports transaction processing through Online Transaction Processing (OLTP) queries. Analytical database meets the Online Analytical Processing (OLAP) requirements of DSS.

NOTES

- Data mining means developing special algorithms to answer the queries of various users. The procedure is to evolve a number of models and to match one of them to data stored in the database.
- Data mining models being mathematical in nature are categorized as predictive and descriptive.
 - (i) A predictive model spells out in advance the values a data may assume based on known results from other data stored in the database. This model performs data mining tasks of classification, time series analysis, regression and prediction.
 - (ii) A descriptive model is based on identification and relationships in data. This model aims at discovering rather than predicting the properties of data.
- Data mining systems are required to support the ad hoc and interactive requirements of knowledge discovery from relational database and multiple levels of abstraction. Data mining languages are designed to meet this requirement. They help in formulating a query to define data mining task primitives.
- Data mining functionalities are used to specify the kind of patterns to be found in data mining tasks. In general, data mining tasks can be classified into two categories: descriptive and predictive.
- Classification is the process of finding a model (or function) that describes and distinguishes data classes or concepts, for the purpose of being able to use the model to predict the class of objects whose class label is unknown.
- Prediction may refer to both numeric prediction and class label prediction; this is use it to refer primarily to numeric prediction in subsequent sections.
- Data mining often requires data integration—the merging of data from multiple data stores. The data may also need to be transformed into forms appropriate for mining.
- In data transformation, the data are transformed or consolidated into forms appropriate for mining. Data transformation can involve the following:
 - (i) Smoothing, which works to remove noise from the data? Such techniques include binning, regression and clustering.
 - (ii) Aggregation, where summary or aggregation operations are applied to the data. For example, the daily sales data may be aggregated so as to compute monthly and annual total amounts. This step is typically used in constructing a data cube for analysis of the data at multiple granularities.

- (iii) Generalization of the data, where low-level or “primitive” (raw) data are replaced by higher-level concepts through the use of concept hierarchies.
- (iv) Normalization, where the attribute data are scaled so as to fall within a small specified range, such as -1.0 to 1.0 , or 0.0 to 1.0 .
- Data reduction techniques can be applied to obtain a reduced representation of the data set that is much smaller in volume, yet closely maintains the integrity of the original data.
- Data mining has emerged as a powerful tool for the decision-makers. Rapid developments in hardware and software resources, continuous improvement in their price–performance ratio and their availability are some of the reasons for encouraging organizations to invest in their data mining infrastructure.

NOTES

2.11 KEY WORDS

- **Histogram:** A diagram consisting of rectangles whose area is proportional to the frequency of a variable and whose width is equal to the class interval.
- **Correlation:** A mutual relationship or connection between two or more things.

2.12 SELF ASSESSMENT QUESTIONS AND EXERCISES

Short Answer Questions

1. How to integrate or couple the DM system with a database (DB) system and/or a Data Warehouse (DW) system?
2. Give the division of knowledge discovery in five steps.
3. What is cluster analysis?
4. What are few techniques of data reduction?

Long Answer Questions

1. Discuss data mining in the telecommunications industry, at length
2. Write an example for data characterization and data discrimination.
3. Discuss and explain the types of coupling.
4. Explain Biclustering and its types.
5. What are the recent trends in data mining?

2.13 FURTHER READINGS

NOTES

- Han, Jiawei, Micheline Kamber and Jian Pei. 2011. *Data Mining: Concepts and Techniques*, 2nd edition. Amsterdam: Elsevier.
- Pujari, Arun K. 2010. *Data Mining Techniques*, 2nd edition. United States: Universities Press.
- Anahory, Sam and Dennis Murray. 1997. *Data Warehousing in the Real World: A Practical Guide for Building Decision Support Systems*, 1st edition. Boston: Addison Wesley.
- Witten, I. H. and Eibe Frank. 2005. *Data Mining: Practical Machine Learning Tools and Techniques*, 2nd edition. Amsterdam: Elsevier.
- Soman, K. P., Shyam Diwakar and V. Ajay. 2006. *Insight Into Data Mining: Theory and Practice*. New Delhi: PHI.

UNIT 3 DATA

Structure

- 3.0 Introduction
- 3.1 Objectives
- 3.2 Types of Data
- 3.3 Need of Preprocessing and Data Quality
 - 3.3.1 Data Preprocessing
- 3.4 Measure of Similarity and Dissimilarity
- 3.5 Exploration: Summary Statistics - Visualization
- 3.6 Answers to Check Your Progress Questions
- 3.7 Summary
- 3.8 Key Words
- 3.9 Self Assessment Questions and Exercises
- 3.10 Further Readings

NOTES

3.0 INTRODUCTION

Data is a collection of facts and figures. In a real world it represents an entity that can be seen and experienced. For e.g. the items in the departmental store or the students studying in the university, they all represent an entity of similar type. These are known as the instance of the entity or the data objects. Each entity has some various unique characteristics which is used to identify the objects and these characteristics are known as the attributes of the object. For e.g. the entity item is characterized by the attributes: item id, item name, category and price etc. Similarly, the student attributes may consists of enrollment no., name, grade, DOB, and address etc. The attributes are the data that is stored and transformed for various purposes like analysis, reporting etc.

3.1 OBJECTIVES

After going through this unit, you will be able to:

- Explain the data and it's types
- Understand the need of pre-processing and data quality
- Explain the measure of similarity and dissimilarity

3.2 TYPES OF DATA

These data are of different types and fall into two basic categories: **Categorical** and **Numerical** data. The categorical data is further sub divided into **nominal** and **ordinal** data. Similarly, numerical data is further subdivided into **interval** and **ratio**.

NOTES

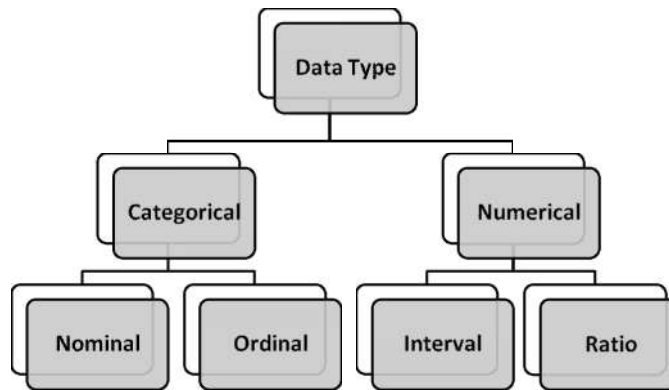


Fig. 3.1 Types of Data

1. **Nominal Data:** The nominal data represents labels for the attributes or they represent some name of the characteristics. For e.g. the gender values like male or female, marital status can have values like married, unmarried, widowed or divorced. These can also be considered as categories and are also known as categorical but they do not have any order between the values. Although the nominal values are names but the name of the categories can also be represented with the numbers. For e.g. the skin colors can be assigned the numerical values like very fair – 01, fair – 02, wheatish – 03 and dark – 04. These numbers do not represent quantities but they represent the color of the skin.
2. **Ordinal:** They also represent labels or names of the attribute but they have a meaningful order or have a ranking between them. However, the difference between the consecutive values is not defined or is not equal. For e.g. pizza sizes can be regular, medium or large. These sizes are in increasing order it can't be said with surety that the difference between the medium and regular is equal to the difference between large and medium. Another example can be student's grade A+, A, B+, B, etc. The grades are in decreasing order.
3. **Interval:** This type of data has numerically measured values and the difference between two successive data values is equal. There is no absolute zero associated with this type of data. For e.g. the temperature measured in degree Celsius or Fahrenheit. Though, we can measure the temperature of the environment but there is no true zero associated with the measured temperature. This means that there is no such temperature as any temperature in Celsius and Fahrenheit. We can add and subtract this temperature but because of the absence of true zero, we cannot multiply or divide.
4. **Ratio:** The ratio data is similar to interval data i.e. it is also measured but unlike interval data it has a true or absolute zero value. Therefore, if a measured quantity is ratio scaled, then the values are the multiple of other values. For e.g. the temperature in Kelvin is a ratio data as it has a meaningful zero i.e. the temperature at which the particles of the matter has no kinetic energy left. The other example of ratio data can be height, weight and price

etc, all of these can hold zero value. The summary of the above data type is shown in the Table 3.1.

Table 3.1 Summary of data type

Scale	True Zero	Equal Intervals	Order	Category	Example
Nominal	No	No	No	Yes	Marital Status
Ordinal	No	No	Yes	Yes	Student grade
Interval	No	Yes	Yes	Yes	Temperature in Fahrenheit
Ratio	Yes	Yes	Yes	Yes	Age, Weight

NOTES

The data can also be divided into 2 different categories based on the type of values. These are Discrete and Continuous data.

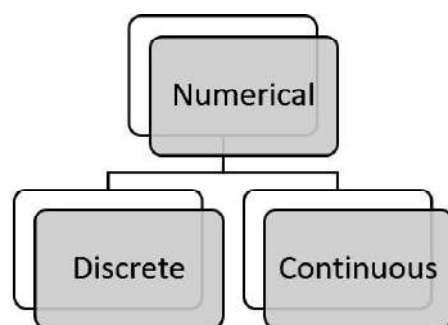


Fig. 3.2 Categorical division of Data based on type of values

- 1. Discrete:** This type of data has finite or infinite counted values. For e.g. the number of children in the class is a value that is a whole number, the value can't be a number like 40.5. Similarly, zip codes are also a set of discrete values that can be a set of natural numbers.
- 2. Continuous:** The continuous values are the measured values in the form of real numbers that can be further refined. For e.g. the distance between the two places is a measured value that is in kilometers but it can be converted into meters and centimeters etc.

Check Your Progress

1. What are the two basic categories of data?
2. What does nominal data represents?
3. Give the two different categories of data based on the type of values.

3.3 NEED OF PREPROCESSING AND DATA QUALITY

Data is said to be good quality data if it serves the purpose of its intended usage. For example, in order to analyze the sales data for a retail store, the sales record for that store is needed by the manager performing the analysis. While analyzing

NOTES

the data, the manager found that many of the attribute values were missing from the attributes like item no., quantity sold and price etc. and some items had even two item codes. Furthermore, he suspects that some of the values might not be correct since they contain unusual values like the price of milk recorded as Rs. 420/lt. Therefore, the data we wished to analyze was inaccurate, incomplete and inconsistent. These are the few of factors that define the data quality. Some of the parameters on which the quality of data depends are:

- **Accuracy:** Inaccurate data means incorrect data value. This might be due to many reasons such as human error due to manual data entry, submitting the wrong data on purpose, fault in the instrument recording the data, inconsistent data entry formats or inconsistent units of data entry etc.
- **Completeness:** Incomplete data means, some of the attribute values that holds much importance in the analysis might be missing. This may happen due to a number of reasons such as inability to understand the importance of the data attribute while recording, fault in recording instruments, or deletion of records by mistake.
- **Consistent:** Since many systems may be used to collect the data from the same source. Regardless of the source used to collect the data, there values should not be contradicting. There must be a stable and steady mechanism that collects and stores the data without contradiction or unwarranted variance.
- **Timeliness:** It refers to the impact of date and time of data collection on their quality. At times this factor really becomes important for the data. For e.g. the previous month sales could have direct effect on this month sales, therefore they have to be updated on time.
- **Noise and Outliers:** Noise refers to some modification in the original data value and outliers means the considerable difference in the data value when compared with the rest of the data. These two quality parameters sometime impacts the results of the analysis considerably especially when they do not represent the dataset and should be ignored while data analysis.

3.3.1 Data Preprocessing

Data preprocessing is the first step in the data mining process which processes the raw data into a form which is easier to use by the data mining algorithms. It is the most important step in data mining process, though at times it is neglected due to many reasons like lack of time, lack of patience or lack of realization of the importance of this step. Analyzing the data that has not been preprocessed properly results incorrect and misleading results. The preprocessing step makes sure the good quality data is used during the mining process. Detecting data anomalies, rectifying them early, and reducing the data to be analyzed can lead to huge payoffs while decision making. The preprocessing steps followed in the data mining process are as follows:

1. Data Cleaning
2. Data Integration
3. Data Transformation
4. Data Reduction

NOTES

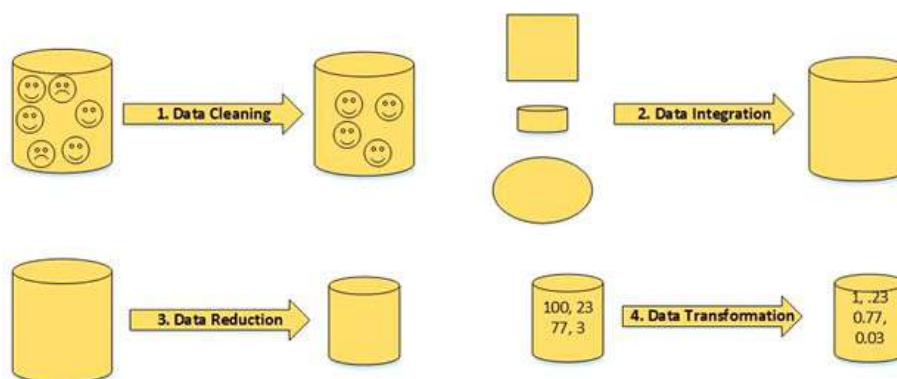


Fig. 3.3 Preprocessing steps in data mining

1. Data Cleaning

The real world data tend to have missing values, can be noisy, and/or are inconsistent. This step helps in data cleaning by filling the missing values, handling the noise and detecting and correcting the inconsistent data. Therefore, various approaches to data cleaning are:

- **Missing values** – Various ways in which missing values can be handled are:
 - o Removing the data tuple – This method is effective when the amount of data is very large and multiple values are missing within a tuple.
 - o Filling the missing values – The missing values can be filled by the most commonly occurring value of other attributes like mean.
- **Noisy Data** – There are various ways to handle noise in the data such as:
 - o Binning method – This method first sorts the data and divides it into various equal size clusters or bins. It then performs local smoothing using various techniques like mean, or replacement of values by boundary values.
 - o Regression – It is used to find a linear or nonlinear function that finds the relationship between different variables or attributes of the dataset. Therefore, data smoothing can be done based on this function.
 - o Outlier Analysis – One of the way to detect outliers is by clustering method where the similar data values are grouped together in a cluster leaving behind the outliers.

NOTES

2. Data Integration

Data often comes from varied sources which need to be integrated for the purpose of reducing data redundancies and data inconsistencies. Data can be integrated in various ways and each face some challenges while data integration.

- **Entity Identification Problem** – Data from various sources come together suffers from the entity identification problem which occurs when the same entities from different sources have different names. For e.g. *customer_id* in one source may be named as *cust_id*. Therefore, there needs to be confirmation that they both represent the same attribute of the same entity.
- **Redundancy** – A derived attribute that is included while integration can lead to redundancy in data. This type of redundancy can be detected with the help of correlation analysis, which helps to establish relationship between the two or more attributes.
- **Tuple Duplication** – It means the data in the tuple is repeated from some other source that is integrated with the data. The repeated tuple is the source of redundancy and often leads to inconsistent data if not identified and removed.
- **Data Value Conflict Detection and Resolution** – Data value conflict means different data value for a data from the different sources. This may be due to difference in measurement units and scaling etc. Therefore, while preprocessing the data such data conflicts should be identified and resolved.

3. Data Reduction

Data mining technique involves working with a huge amount of data but in some cases the analysis of data and results become difficult as the data increases beyond a limit. This preprocessing technique aims to obtain a smaller subset of data in terms of volume and that represents closely the original data. This is done so as to perform the data mining task more efficiently yet producing the same result. Various data reduction techniques are:

- **Dimensionality Reduction:** It refers to removal of irrelevant, weak attribute, or redundant attribute from the data. This helps in reducing the number of random parameters under consideration while applying the data mining techniques. The most commonly used methods for dimensionality reduction are Wavelet Transforms and Principal Component Analysis (PCA).
- **Numerosity Reduction:** In this type of data reduction technique, the reduction is done by replacing the original data with an alternative smaller datasets. Parametric and Non parametric models are the two models for numerosity reduction. Parametric is one which does not require the actual data to be replaced but it is replaced by a model that produces the required representations while the later reduction includes techniques like clustering, sampling and the use of histogram.

- **Data Compression:** Original data can be reduced by compression which can be done with the help of encoding or some other technique. There are two types of reduction: one in which the original data can be produced back without any loss of information, known as lossless compression, The other in which only the approximate data is produced from the reduced data, known as lossy compression.

NOTES

4. Data Transformation

At times the original data, on which data mining technique has to be applied, is not in an appropriate form for applying the technique. Therefore, it has to be transformed to a form so that it becomes suitable for data mining task. This process is known as data transformation. For e.g. for some data mining task, the distance in meter would be more appropriate than in kilometers. Therefore the distance has to be converted into meter. There are various techniques for data transformation like:

- **Normalization:** Normalizing the data is required when the data is expressed in different units and these units can affect the outcome of analysis. Therefore, data normalization is carried out so that each attribute value has an equal weight. It can also be perform to altogether eliminate the units of measurement. For e.g. the number of banking frauds per month can be normalized to the percentage of frauds per month which gives a more clear comparison of banking frauds in a month. Data can also be transformed to a certain range of values like *tanh* function squashes any input value between [0, 1].
- **Smoothing:** This technique aims at reducing noise from the data. Various smoothing techniques are: binning, clustering and regression which already defined in the data cleaning step.
- **Data Aggregation:** It is a step in which data is summarized to give a more meaningful analysis. For example, daily sales closing may be aggregated to monthly or yearly closing to see if the company achieved its annual target or not. This step is useful when we want to aggregate the data related to many parameters at a time like in data cube.
- **Data Generalization:** Generalization is the concept where the fine data is replaced by the high level or abstract data. For e.g. instead of individual class of the student, it might be replaced by concepts such as preschoolers, toddlers, primary, middle or high school.

Check Your Progress

4. Give four parameters on which the quality of data depends.
5. What is data pre-processing?
6. What is the relevance of data reduction in data mining?

3.4 MEASURE OF SIMILARITY AND DISSIMILARITY

NOTES

Two data distributions can be compared for similarity and dissimilarity as comparison is very important for many data mining problems like classification, outlier analysis and clustering. This type information can be useful for business organizations like retail stores where they want to group the customers based on their shopping experience or item preferences etc. There are various similarity and dissimilarity measures, referred to as proximity measures. The similarity between two objects is a numeral measure of the degree to which the two objects are alike. Therefore, similarities are higher for pairs of objects that are more alike. They are usually non-negative and are often between 0 (no similarity) and 1 (complete similarity). The dissimilarity between two objects is the numerical measure of the degree to which the two objects are different. Dissimilarity is lower for more similar pairs of objects. The term distance is used to define dissimilarity. It may fall in the interval $[0, \infty]$.

- Similarity and dissimilarity between different types of attributes:**
 The proximity of object having a number of attributes is given by combining the proximities of individual attributes. These are different for different data type. For e.g. the dissimilarity for numerical data type is given by absolute difference between their values whereas for nominal values, if the attribute values are equal then the data objects are similar else they are dissimilar.

Table 3.2 Similarity and dissimilarity of data based on their attribute type

Attribute Type	Dissimilarity	Similarity
Interval or Ratio	$d = x - y $	$s = -d, s = \frac{1}{1+d}, s = e^{-d},$ $s = 1 - \frac{d - \min_d}{\max_d - \min_d}$
Nominal	$d = \begin{cases} 0 & \text{if } x = y \\ 1 & \text{if } x \neq y \end{cases}$	$s = \begin{cases} 1 & \text{if } x = y \\ 0 & \text{if } x \neq y \end{cases}$
Ordinal	$d = \frac{ x - y }{(n - 1)}$ (values are mapped to integers 0 and n-1 where n is the number of values)	$s = 1 - d$

The dissimilarity between two numeric objects is defined by the measure of distance between them. The distance can be calculated in various manners such as:

- Euclidian Distance:** It is the most popular and the straight line distance between two data objects and is given by:

$$d(i, j) = \sqrt{(x_{i1} - x_{j1})^2 + (x_{i2} - x_{j2})^2 + \dots + (x_{in} - x_{jn})^2}$$

Where x_i and x_j are the two objects having n numeric attributes.

- **Manhattan Distance:** The Manhattan distance is the distance that would be traveled to get from one data point to the other if a grid-like path is followed. The Manhattan distance between two items is the sum of the differences of their corresponding components. It is given by:

$$d(i, j) = |x_{i1} - x_{j1}| + |x_{i2} - x_{j2}| + \dots + |x_{in} - x_{jn}|$$

- **Minkowski Distance:** It is the generalization of Euclidian and Manhattan distance and is given by:

$$d(i, j) = \sqrt[h]{|x_{i1} - x_{j1}|^h + |x_{i2} - x_{j2}|^h + \dots + |x_{in} - x_{jn}|^h}$$

Where h is a real number having value $e \geq 1$. The above equation represents Manhattan distance with $h=1$ and Euclidian distance with $h=2$.

Check Your Progress

7. What are proximity measures?
8. What is Euclidian distance?
9. Define similarity.

3.5 EXPLORATION: SUMMARY STATISTICS - VISUALIZATION

Exploratory Data Analysis (EDA) – EDA is an approach to summarizing, visualizing, and becoming intimately familiar with the important characteristics of a data set. It refers to a set of techniques to display data in such a way that interesting features becomes apparent. It provides the initial pointer towards various learning techniques rather than beginning with an assumed model for the data and taking the research further. It also confirms to certain level of certainty that the future results would be valid, correctly interpreted and applied to desired business context. Exploration is carried out for structures and may indicate the relationship between various variables. The two aspects of EDA are numerical summarization and data visualization.

1. Numerical Summarization

Numerical summarization is the overall picture of the data under consideration. These are the basic statistical descriptions that can be used to identify certain properties of the data and helps in highlighting the important data and the data that can be treated as noise or outliers. There are several measures of statistical descriptions.

NOTES

NOTES

- **Measure of Central Tendency** –It is also known as measure of location since it aims to find the location of the middle or center of distribution. There are three measures of central tendency: **Mean, Median** and **Mode**.

Mean or arithmetic mean is the most common numeric measure of the center of the data. It is calculated by adding all the data values and dividing the result by the number of values. For e.g. if the n data values are represented as x_1, x_2, \dots, x_n , then the mean is given by:

$$\bar{x} = \frac{x_1 + x_2 + \dots + x_n}{n}$$

Median is the central data value from the list of sorted data values. The

location of median from the list of n sorted values is given by $\frac{n}{2}$ if n is odd

and mean of the data at $\frac{n}{2}$ and $\frac{n+1}{2}$ if n is even.

Mode is another measure that gives the most frequently occurring value from the dataset. For finding the mode the frequency of each data value has to be recorded and the data value having maximum frequency is the mode.

- o **Measure of Dispersion:** The dispersion or the spread of data is another important statistical measure to understand the variability of data. Some of the measures of dispersion are: Variance, Standard Deviation, Range and Interquartile Range.

Variance and Standard Deviation (SD) gives an idea about the distribution of data. Variance gives information about the distance of data point from the mean. It is defined as the average of the squared difference between the mean and individual data value. SD is the square root of variance and is defined as average distance between mean and individual data value. Range is defined as the difference between the maximum and the minimum value of the sample. Quartiles are points taken at regular intervals of a data distribution, dividing it into equal size consecutive sets. For e.g. 2 - quartile is the point that divides the data distribution into 2 equal halves. It is also the median of the dataset. Similarly, 4-quartile is the 3 points that divide the distribution into 4 equal halves. The difference between the first and the third quartile is known as the Interquartile range (IQR) which gives the distribution of middle 50% of the data.

- o **Measure of Skewness:** It tells about the shape of distribution. The distribution is said to be well behaved if it is symmetrical and the mean and the median coincides. The symmetry fails if there exists a tail on the either side.

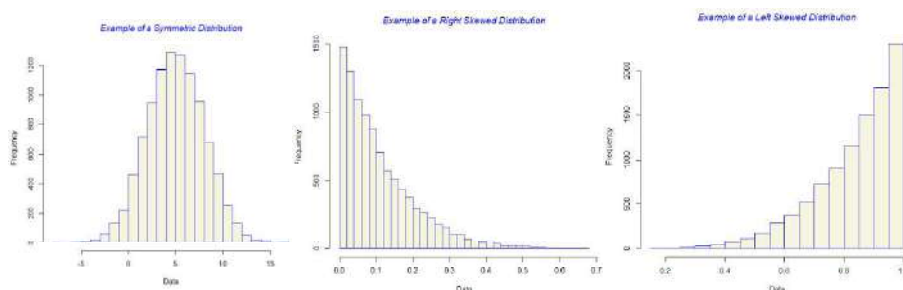


Fig. 3.4 Example of Symmetric, Left skewed and Right Skewed Distribution

It describes the degree of linear relationship between two variables. The correlation coefficient is always between $[-1, 0]$ for negative linear relationship and $[0, +1]$ for positive linear relationship. If the correlation coefficient is 0, then there is no linear relationship between the two variables.

2. Visualization

Data visualization is a most effect way to understand and analyze the large amount of data. Visualization is the use of graphs and charts for revealing the hidden patterns in the data. It helps to explore and analyze the relationship between different attributes of the data object by creating the visuals in a number of dimensions. Visualization techniques depend on the type of variables. It is different for different data type, visualizations for nominal data type are different from those for continuous data type. There are various tools for visualization like:

- o **Histogram:** Histogram is the most effective visualization tool to represent frequency in continuous data or discrete data. The horizontal axis is used to plot the samples and the vertical axis the frequencies or relative frequencies of each class is plotted. The histogram shows the wage plot for 3000 workers which are almost symmetrical.

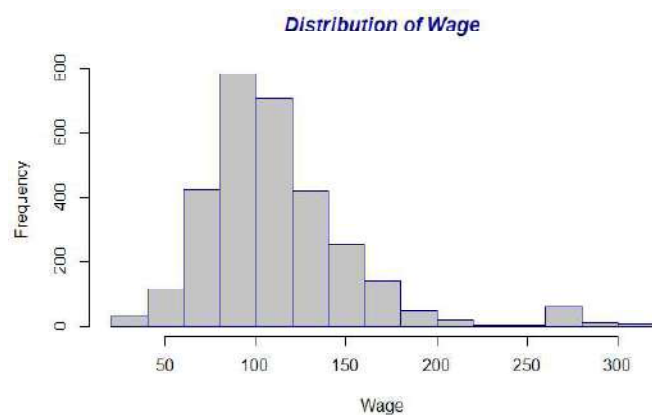


Fig. 3.5 Example of Histogram representing frequency in continuous data or discrete data

NOTES

NOTES

- **Boxplot:** Boxplot is used to analyze the data distribution and to identify outliers in the data. An observation is considered to be an outlier if it is either less than $Q_1 - 1.5 \text{ IQR}$ or greater than $Q_3 + 1.5 \text{ IQR}$ as shown in the Figure 3.6, where IQR is the inter-quartile range defined as $Q_3 - Q_1$. This rule is conservative and often too many points are identified as outliers. Hence sometimes only those points outside of $[Q_1 - 3 \text{ IQR}, Q_3 + 3 \text{ IQR}]$ are only identified as outliers.

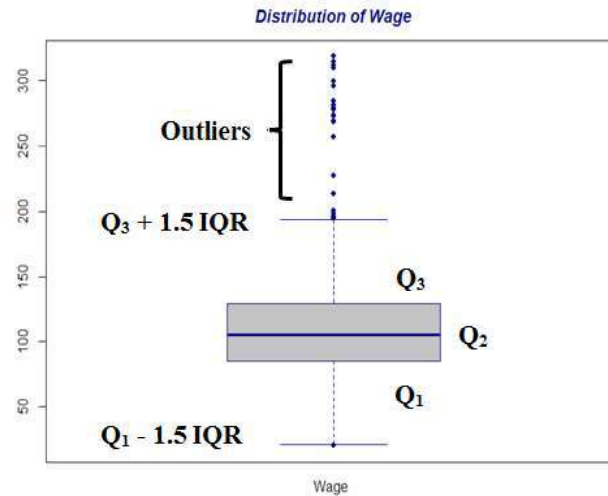


Fig. 3.6 Example of Box plot identifying outliers in the data

- **Scatterplot:** Scatterplot is a visualization technique to find the relationship between two variables. It shows the direction and strength of association between two variables but not in terms of quantity. Scatterplots also help to identify unusual observations. The scatterplot shows that the weight is directly dependent on height. From the plot we can infer that as the height increases, the weight also increases.

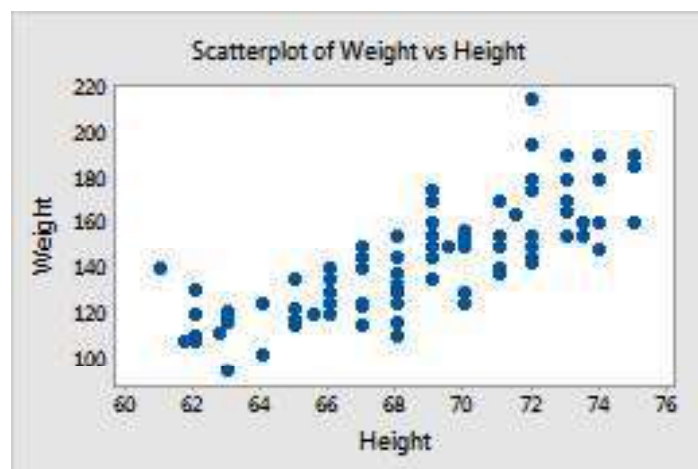


Fig. 3.7 Example of Scatterplot showing the direction and strength of association between two variables

- **Scatterplot Matrix:** Displaying more than two variables on a single scatterplot is not possible. Scatterplot matrix is one possible visualization, of three or more continuous variables, taken two at a time. The scatterplot matrix given below shows the relationship between the various attributes of the college.

NOTES

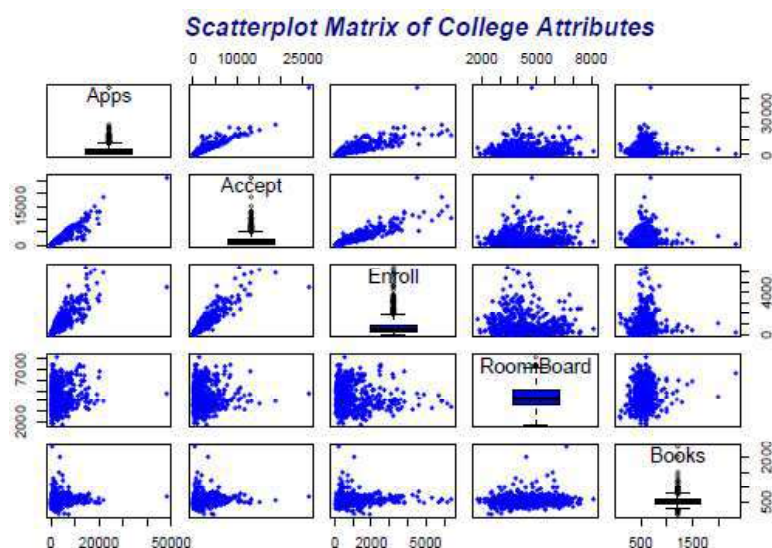


Fig. 3.8 Example of Scatterplot Matrix of three or more continuous variables, taken two at a time

Check Your Progress

10. What are the two aspects of EDA?
11. What are the measures of central tendency?
12. What is variance?

3.6 ANSWERS TO CHECK YOUR PROGRESS QUESTIONS

1. The two basic categories of data are Categorical and Numerical.
2. The nominal data represents labels for the attributes or they represent some name of the characteristics.
3. The two different categories of data based on the type of values are discrete and continuous.
4. The parameters on which the quality of data depends are: Accuracy, Completeness, Consistent and Timeliness.
5. Data pre-processing is the first step in the data mining process which processes the raw data into a form which is easier to use by the data mining algorithms.

NOTES

6. Data reduction pre-processing technique aims to obtain a smaller subset of data in terms of volume. This is done so as to perform the data mining task more efficiently yet producing the same result.
7. There are various similarity and dissimilarity measures that are referred to as proximity measures.
8. Euclidian Distance is the most popular and the straight line distance between two data objects and is given by:

$$d(i, j) = \sqrt{(x_{i1} - x_{j1})^2 + (x_{i2} - x_{j2})^2 + \dots + (x_{in} - x_{jn})^2}$$

where x_i and x_j are two objects having n numeric attributes.

9. The similarity between two objects is a numeral measure of the degree to which the two objects are alike.
10. The two aspects of EDA are Numerical summarization and data visualization.
11. There are three measures of central tendency: Mean, Media and Mode.
- 12) Variance gives information about the distance of data point from the mean. It is defined as the average of the squared difference between the mean and individual data value.

3.7 SUMMARY

- Data is a collection of facts and figures. In a real world it represents an entity that can be seen and experienced.
- The attributes are the data that is stored and transformed for various purposes like analysis, reporting etc.
- These data are of different types and fall into two basic categories: Categorical and Numerical.
- The categorical data is further sub divided into nominal and ordinal data. Similarly, numerical data is further subdivided into interval and ratio.
- The nominal data represents labels for the attributes or they represent some name of the characteristics.
- Ordinal data also represent labels or names of the attribute but they have a meaningful order or have a ranking between them.
- Interval data has numerically measured values and the difference between two successive data values is equal. There is no absolute zero associated with this type of data.
- The ratio data is similar to interval data i.e. it is also measured but unlike interval data it has a true or absolute zero value.
- Data is said to be good quality data if it serves the purpose of its intended usage.

- The parameters on which the quality of data depends are: Accuracy, Completeness, Consistent, Timeliness, Noise and Outliers.
- Data preprocessing is the first step in the data mining process which processes the raw data into a form which is easier to use by the data mining algorithms.
- Analyzing the data that has not been preprocessed properly results incorrect and misleading results. The preprocessing step makes sure the good quality data is used during the mining process.
- The preprocessing steps followed in the data mining process are – data cleaning, data integration, data transformation and data reduction.
- Two data distributions can be compared for similarity and dissimilarity as comparison is very important for many data mining problems like classification, outlier analysis and clustering.
- The similarity between two objects is a numeral measure of the degree to which the two objects are alike.
- The dissimilarity between two objects is the numerical measure of the degree to which the two objects are different.
- The dissimilarity between two numeric objects is defined by the measure of distance between them. The distance can be calculated in various manners such as: Euclidian distance, Manhattan distance and Minkowski distance.
- Exploratory Data Analysis (EDA) is an approach to summarizing, visualizing, and becoming intimately familiar with the important characteristics of a data set.
- The two aspects of EDA are numerical summarization and data visualization.

NOTES

3.8 KEY WORDS

- **Tuple:** A data structure consisting of multiple parts.
- **Anomalies:** Problems that can occur in poorly planned, un-normalized databases.
- **Data Cube:** The cube is used to represent data along some measure of interest.

3.9 SELF ASSESSMENT QUESTIONS AND EXERCISES

Short Answer Questions

1. Write the equation for Manhattan distance.
2. What is the relevance of data preprocessing?

3. What is BoxPlot?
4. What is noise and outliers?
5. What is data transformation?

NOTES

Long Answer Questions

1. Describe the types of data and its categorization with the help of figures.
2. What are the parameters of Data quality?
3. What are the steps followed in Data Preprocessing? Explain.
4. Define Histogram. Explain the types of Histograms with diagrams.
5. What do you understand by Exploratory Data Analysis? Explain.

3.10 FURTHER READINGS

Han, Jiawei, Micheline Kamber and Jian Pei. 2011. *Data Mining: Concepts and Techniques*, 2nd edition. Amsterdam: Elsevier.

Pujari, Arun K. 2010. *Data Mining Techniques*, 2nd edition. United States: Universities Press.

Anahory, Sam and Dennis Murray. 1997. *Data Warehousing in the Real World: A Practical Guide for Building Decision Support Systems*, 1st edition. Boston: Addison Wesley.

Witten, I. H. and Eibe Frank. 2005. *Data Mining: Practical Machine Learning Tools and Techniques*, 2nd edition. Amsterdam: Elsevier.

Soman, K. P., Shyam Diwakar and V. Ajay. 2006. *Insight Into Data Mining: Theory and Practice*. New Delhi: PHI.

BLOCK - II

ASSOCIATION RULE MINING AND CLASSIFICATION

NOTES

UNIT 4 ASSOCIATION RULES

Structure

- 4.0 Introduction
- 4.1 Objectives
- 4.2 Association Rules and Apriori Algorithm
 - 4.2.1 Association Rules from Relational Databases and Data Warehouses
- 4.3 Partition Algorithm
- 4.4 Answers to Check Your Progress Questions
- 4.5 Summary
- 4.6 Key Words
- 4.7 Self Assessment Questions and Exercises
- 4.8 Further Readings

4.0 INTRODUCTION

Association rules are if-then statements that lend a hand to explain the probability of associations between data items inside large data sets in a variety of databases. Association rule mining has a number of applications and is widely used to help discover sales correlations in transactional data or in medical data sets. Association rule mining, at a basic level, involves the use of machine learning models to analyze data for patterns, or co-occurrence, in a database. It identifies frequent if-then relations, which are called *association rules*.

Association rules are considered from *itemsets*, which are made up of two or more items. If rules are built from analyzing all the possible itemsets, there could be so many rules that the rules hold little meaning. With that, association rules are typically created from rules well-represented in data.

4.1 OBJECTIVES

After going through this unit, you will be able to:

- Explain the significance of association rules
- Describe the significance of different types of association rules
- Discuss the process of mining frequent itemsets without candidate generation
- Explain the association rules from relational database and data warehouses
- Discuss partition algorithm

4.2 ASSOCIATION RULES AND APRIORI ALGORITHM

NOTES

Association rules are applied for data and text mining in huge databases and are associated with correlation relationships. Association rules are used in data mining to find the rule:

If X, then Y

Here, X and Y are single values, items or rules.

For example, (e.g., if (Car=HondaCity and Gender=Male and Age<25) then (Risk=High and Insurance=High)).

Association rules are tagged with data mining providing correlation relationships if a large data set of data items is provided. In this, market-basket analysis is most popularly used. Today, bar-code based data is collected from malls and supermarkets. These types of market baskets refer to large databases. These databases consist of large number of transaction records. If a customer purchases on each record list, the concerned manager would know which particular items are popular. To arrive at a conclusion regarding customer preferences, the manager can rely on store layout, cross-selling, promotions, catalogue designs, etc. With the aid of the data collected from these, customer segments are identified based on the buying trends. The **'if-then'** statement is used for association rule providing data computation. The **'if'** part is taken as antecedent and **'then'** part is the consequent phase of association rule. This rule has two numbers which support the degree of uncertainty. For this, itemsets are used to represent the antecedent and consequent sets refer to disjoint property. Disjoint property does not consider common items. The first number among two given numbers is known as **'support'** that supports the number of transactions including the antecedent and consequent sets. Support is expressed in terms of percentage which is the total number of records in the specified database. The second part is known as **'confidence'** of the rule. Confidence represents the ratio of number of transactions. It includes all items in consequent (confidence) as well as in antecedent (support). For example, the database of a supermarket has one lakh transaction records (in point-of-sale) in which 2000 items are of Item **A** and Item **B** and 800 are of Item **C**. The possibility of implementing the association rule is as follows:

If A and B are purchased then C can also be purchased on the same trip.

This statement supports that 800 transactions have been made for which support is calculated as 0.8%, i.e., $800/100000$ and confidence as 40% ($800/2000$).

Here there is a possibility of random selection transaction from the database. The database includes all items of antecedent and consequent. Confidence refers

to the conditional probability depending on the random selecting transaction. The items in consequent include all the specified items that are included in the transaction referring to the antecedent. The term, '**Lift** (confidence ratio)', included in association analysis calculates the confidence ratio to the expected confidence. The term expected confidence can be defined as follows:

'Expected confidence, refers to the fact that buying Items A and B do not enhance the probability of buying Item C.'

The total number of transactions includes the consequent which is divided by the number of transactions. This can be explained as follows:

Assume that the possible number of transactions for Item C is 5000. The expected confidence can be calculated as $5000/100000=5\%$. But, for the supermarket **Lift** is calculated as follows:

$$\begin{aligned} \text{Lift} &= \text{Confidence} / \text{Expected Confidence} \\ &= 40\% / 5\% \\ &= 8 \end{aligned}$$

Therefore, it is clear that **Lift** value increases the probability of '**then**' representing consequent followed by '**if**' antecedent part.

There are two types of association rules, known as single association rule and double association rule.

Single Association Rule

It is a rule that references only one dimension. The following example of single association rule predicts one-dimensional association as follows:

`buys(X, "milk") ⇒ buys(X, "bread")`

The single dimension is buys.

Double Association Rule

It is a set of rules that reference different levels of abstraction. The following example of two-dimensional association rule predicts two-dimensional association as follows:

`age(X, "19-25") ⇒ buys(X, "popcorn") ⇒ buys(X, "coke")`

The transformation into two-dimensional rules predicts the value pairs of items as follows:

`customer(X, [age, "19-25"]) ⇒ customer(X, [buys, "popcorn"]) ⇒ customer(X, [buys, "coke"])`

The simplified notation for two-dimensional rules is written as follows:

`{milk} ⇒ {bread} {[age, "19-25"], [buys, "popcorn"]} ⇒ {[buys, "coke"]}`

NOTES

NOTES

Some basic notations are used in association rules of data mining. These notations are known as ‘TISC’ representing Transaction, Itemset, Support and Confidence. A transaction consists of an itemset and a transaction identifier. A set of items is known as itemset which collectively works with transactions; for example, purchased items. Support refers to the total number of transactions satisfying the rules of association, whereas confidence refers to the probability with which tagged rule is satisfied.

Terminology and Notation
 Set of all items I , subset of I is called itemset
 Transaction (tid, T), $T \subseteq I$ itemset, transaction identifier tid
 Set of all transactions D (database), Transaction $T \in D$

Definition of Association Rules $A \Rightarrow B [s, c]$
 A, B itemsets ($A, B \subseteq I$)
 $A \cap B$ empty
 supports $S =$ probability that a transaction contains $A \cup B$
 $= P(A \cup B)$
 confidence $c =$ conditional probability that a transaction having A also contains B
 $= P(B|A)$

Fig. 4.1 Definitions of Association Rules

In Figure 4.1, the association rules are referenced with the following examples:

Item $I = \{apple, beer, eggs, milk\}$

Transaction (2000, {beer, diaper, milk})

Association rule {beer} \Rightarrow {diaper} [0.5, 0.66]

In Figure 4.2, assume that the support for $A \cup B$ as high. The pictorial representation shows the possibilities for support and confidence in association rules.

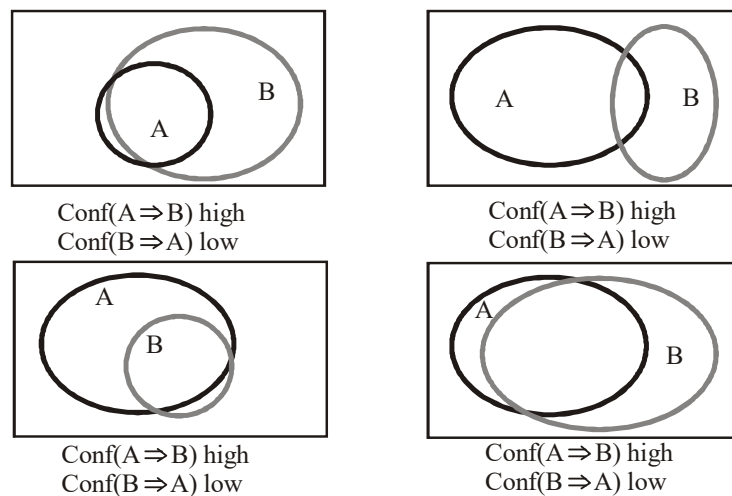


Fig. 4.2 Possible Situations for Support and Confidence

Basic Concepts

Let $I = \{I_1, I_2, \dots, I_m\}$ be a set of items. Let D , the task-relevant data, 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 $A \subset I$, $B \subset I$, and $A \cap B = \emptyset$. 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., the union of sets A and B , or say, 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 , where c is the percentage of transactions in D containing A that also contain B . This is taken to be the conditional probability, $P(B|A)$. That is,

$$\text{support}(A \Rightarrow B) = P(A \cup B) \quad (4.1)$$

$$\text{confidence}(A \Rightarrow B) = P(B|A). \quad (4.2)$$

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 values so as to occur between 0% and 100%, rather than 0 to 1.0.

A set of items is referred to as an itemset. An itemset that contains k items is a k -itemset. The set {computer, antivirus software} is a 2-itemset. The occurrence frequency of an itemset is the number of transactions that contain the itemset. This is also known, simply, as the frequency, support count, or count of the itemset. Note that the itemset support defined in Equation (4.1) is sometimes referred to as relative support, whereas the occurrence frequency is called the absolute support. If the relative support of an itemset I satisfies a prespecified minimum support threshold (i.e., the absolute support of I satisfies the corresponding minimum support count threshold), then I is a frequent itemset. The set of frequent k -itemsets is commonly denoted by L_k .

From Equation (4.2), we have

$$\text{confidence}(A \Rightarrow B) = P(B|A) = \frac{\text{support}(A \cup B)}{\text{support}(A)} = \frac{\text{support_count}(A \cup B)}{\text{support_count}(A)}. \quad (4.3)$$

Equation (4.3) shows that the confidence of rule $A \Rightarrow B$ can be easily derived from the support counts of A and $A \cup B$. That is, once the support counts of A , B , and $A \cup B$ are found, it is straightforward to derive the corresponding association rules $A \Rightarrow B$ and $B \Rightarrow A$ and check whether they are strong. Thus the problem of mining association rules can be reduced to that of mining frequent itemsets.

In general, association rule mining can be viewed as a two-step process:

1. **Find all frequent itemsets:** By definition, each of these itemsets will occur at least as frequently as a predetermined minimum support count, min sup.

NOTES

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

NOTES

Apriori Algorithm

The Apriori algorithm finds frequent itemsets using candidate generation. It was proposed by R. Agarwal and R. Srikant in 1994 for mining frequent itemsets for Boolean association rules.

The Apriori algorithm is the classical algorithm for data mining association rules. This algorithm is basically designed for databases containing transactions; examples include purchasing of a collection of items and details of Website searching. This algorithm finds the subsets commonly known as minimum number C. This term is commonly referred to as the cut-off or cut-off threshold of itemsets. The following are the features of Apriori algorithm:

- It uses ‘bottom-up’ approach that is basically an extended subset of one given item at a time. This step is referred to as itemset of candidate generation.
- If successful extensions are not found, the algorithm is terminated.
- This algorithm uses the technique of breadth-first search and structure of hash tree. These two mechanisms are used to count the itemsets of candidate generation.
- This algorithm is used to find all the frequent itemsets.
- This algorithm is used to implement the searching technique using frequent item property.
- This algorithm is used to optimize the dataset conditionally.
- This algorithm has various types of measures to maintain the association rules.

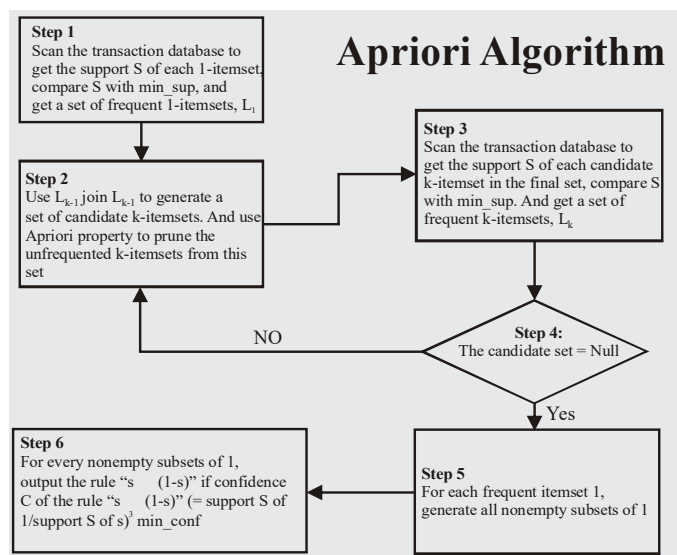


Fig. 4.3 Flowchart of an Apriori Algorithm

Figure 4.3 shows that six steps are involved in an Apriori algorithm. Step 1 scans the transaction and proceeds to Step 2. Step 2 generates the candidate itemsets using Apriori property to prune the unfrequented itemsets. Step 3 scans the transaction database to get the support S . S belongs to candidate itemsets. The control flows to Step 4 to check whether candidate set is equal to NULL or not. If condition is satisfied, the control proceeds to Step 5; otherwise, it returns to Step 3. Step 5 checks frequently those itemsets which generates all non-empty subsets. It sends the control to Step 6 which finally decides the confidence rule.

NOTES

Apriori Algorithm Examples

The following problem decomposition is considered as an example of Apriori Algorithm:

Transaction ID	Items Bought
1	Shoes, Shirt, Jacket
2	Shoes, Jacket
3	Shoes, Jeans
4	Shirt, Sweatshirt

If the minimum support is 50%, then {Shoes, Jacket} is the only itemset that satisfies the minimum support.

Frequent Itemset	Support
{Shoes}	75%
{Shirt}	50%
{Jacket}	50%
{Shoes, Jacket}	50%

$$Confidence(A \Rightarrow B) = \frac{\#_tuples_containing_both_A_and_B}{\#_tuples_containing_A}$$

If the minimum confidence is 50%, then the only two rules are generated from the itemset and the confidence will be greater than 50%. These are represented with the help of an example as follows:

Shoes \Rightarrow Jacket Support=50%, Confidence=66%

Jacket \Rightarrow Shoes Support=50%, Confidence=100%

$$Support(A \Rightarrow B) = \frac{\#_tuples_containing_both_A_and_B}{total_#_of_tuples}$$

NOTES

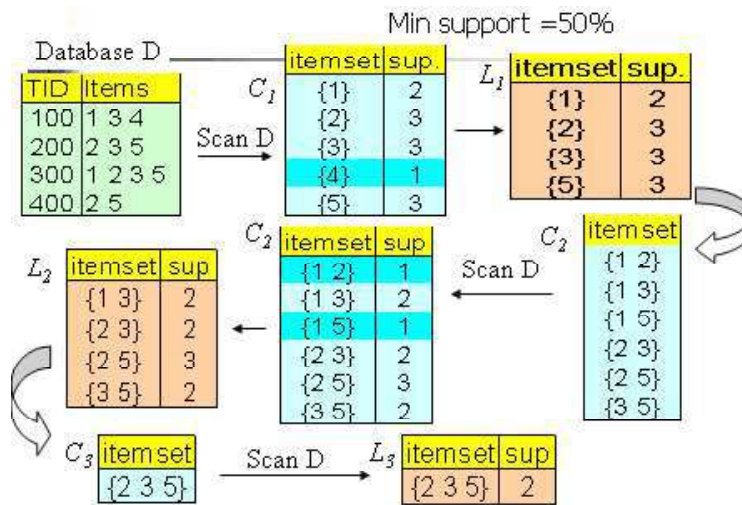


Fig. 4.4 Steps Involved in an Apriori Algorithm

Figure 4.4 shows the steps involved in an Apriori algorithm.

The pseudocode implemented in *Apriori algorithm* is as follows:

Apriori is designed to operate on databases containing transactions (for example, collections of items bought by customers, or details of a Website frequentation). Other algorithms are designed for finding association rules in data having no transactions (Winepi and Minepi), or having no timestamps (DNA sequencing). Each transaction is seen as a set of items (an *itemset*). Given a threshold C , the Apriori algorithm identifies the item sets which are subsets of at least C transactions in the database.

Apriori uses a ‘bottom up’ approach, where frequent subsets are extended one item at a time (a step known as *candidate generation*), and groups of candidates are tested against the data. The algorithm terminates when no further successful extensions are found.

Apriori uses breadth-first search and a Hash tree structure to count candidate item sets efficiently. It generates candidate item sets of length k from item sets of length $k - 1$. Then it prunes the candidates which have an infrequent sub pattern. According to the downward closure lemma, the candidate set contains all frequent-length item sets. After that, it scans the transaction database to determine frequent item sets among the candidates.

The pseudo code for the algorithm is given below for a transaction database T , and a support threshold of ϵ . Usual set theoretic notation is employed; though note that T is a multiset. C_k is the candidate set for level k . Generate () algorithm is assumed to generate the candidate sets from the large item sets of the preceding level, heeding the downward closure lemma. *count* [c] accesses a field of the data structure that represents candidate set c , which is initially assumed to be zero.

Many details are omitted below, usually the most important part of the implementation is the data structure used for storing the candidate sets, and counting their frequencies.

```

Apriori( $T, \epsilon$ )
 $L_1 \leftarrow \{\text{large 1-itemsets}\}$ 
 $k \leftarrow 2$ 
while  $L_{k-1} \neq \emptyset$ 
     $C_k \leftarrow \{a \cup \{b\} \mid a \in L_{k-1} \wedge b \in \bigcup L_{k-1} \wedge b \notin a\}$ 
    for transactions  $t \in T$ 
         $C_t \leftarrow \{c \mid c \in C_k \wedge c \subseteq t\}$ 
        for candidates  $c \in C_t$ 
             $\text{count}[c] \leftarrow \text{count}[c] + 1$ 
         $L_k \leftarrow \{c \mid c \in C_k \wedge \text{count}[c] \geq \epsilon\}$ 
         $k \leftarrow k + 1$ 
return  $\bigcup_k L_k$ 

```

NOTES

Advantages and Disadvantages of Apriori Algorithm

The advantages of Apriori algorithm are as follows:

- It uses large itemset property.
- It can be easily parallelized.
- It is easy to implement.

The disadvantages of Apriori algorithm are as follows:

- It maintains a transaction database which is basically memory resident.
- It scans database as per the requirement.

Apriori is an influential algorithm for mining frequent itemsets for Boolean association rules. This algorithm uses prior knowledge of frequent itemset properties. This algorithm iteratively finds all possible itemsets that have support greater or equal to a given minimum support value. First, the set of frequent 1-itemsets is found by scanning the database to accumulate the count for each item, and collecting those items that satisfy minimum support. This resulting set is denoted as L_1 . Next L_1 is used to find L_2 , the set of frequent 2-itemsets, which is used to find L_3 , and so on, until no more frequent k -itemsets can be found. The finding of each L_k requires one full scan of the database. The output of the Apriori algorithm consists of a set of k -itemsets ($k=1,2,\dots$) that have support greater or equal to a given minimum support value.

The working mechanism of Apriori algorithm is as follows:

1. First, the 1-itemset C_1 is obtained. Based on C_1 , we count the frequency of occurrence of each itemset in all the transactions, prune the itemsets that do not exceed the support threshold and get the frequent 1-itemset L_1 .

NOTES

2. For all $k, k=2, \dots$, use L_k to generate C_{k+1} by L_k join L_k . For each of these $(k+1)$ -itemsets, check whether all its k subset is in the frequent k -itemset L_k . If the answer is yes, keep it in C_{k+1} ; otherwise, prune it. After getting C_{k+1} , scan the database once again to count the support for all the itemsets in C_{k+1} , prune those do not exceed the support threshold and get L_{k+1} .
3. Repeat the above Step 2 until no itemset could be generated for C_{k+1} .

Apriori algorithm needs to scan the database multiple times. When mining a huge database, multiple database scans are costly. One feasible strategy to improve the efficiency of Apriori algorithm is to reduce the number of database scans.

To improve the efficiency of the level-wise generation of frequent itemsets, an important property called the Apriori property is used to reduce the search space.

Apriori Property: *All nonempty subsets of a frequent itemset must also be frequent.*

By definition, if an itemset I does not satisfy the minimum support threshold, $min\ sup$, then I is not frequent; that is, $P(I) < min\ sup$. If an item A is added to the itemset I , then the resulting itemset (i.e., $I \cup A$) cannot occur more frequently than I . Therefore, $I \cup A$ is not frequent either; that is, $P(I \cup A) < min\ sup$.

This property belongs to a special category of properties called anti-monotone in the sense that if a set cannot pass a test, all of its supersets will fail the same test as well. It is called anti-monotone because the property is monotonic in the context of failing a test.

‘How is the Apriori property used in the algorithm?’ To understand this, let us look at how L_{k+1} is used to find L_k for $k \geq 2$. A two-step process is followed, consisting of join and prune actions.

1. **The Join Step:** To find L_k , a set of candidate k -itemsets is generated by joining L_{k+1} with itself. This set of candidates is denoted C_k . Let l_1 and l_2 be itemsets in L_{k+1} . The notation $l_i[j]$ refers to the j th item in l_i (e.g., $l_1[k-2]$ refers to the second to the last item in l_1). By convention, Apriori assumes that items within a transaction or itemset are sorted in lexicographic order. For the $(k-1)$ -itemset, l_i , this means that the items are sorted such that $l_i[1] < l_i[2] < \dots < l_i[k-1]$. The join, $L_{k+1} \bowtie L_{k+1}$, is performed, where members of L_{k+1} are joinable if their first $(k-2)$ items are in common. That is, members l_1 and l_2 of L_{k+1} are joined if $(l_1[1] = l_2[1]) \wedge (l_1[2] = l_2[2]) \wedge \dots \wedge (l_1[k-2] = l_2[k-2]) \wedge (l_1[k-1] < l_2[k-1])$. The condition $l_1[k-1] < l_2[k-1]$ simply ensures that no duplicates are generated. The resulting itemset formed by joining l_1 and l_2 is $l_1[1], l_1[2], \dots, l_1[k-2], l_1[k-1], l_2[k-1]$.
2. **The prune step:** C_k is a superset of L_k , that is, its members may or may not be frequent, but all of the frequent k -itemsets are included in C_k . A scan of the database to determine the count of each candidate in C_k would result in

the determination of L_k (i.e., all candidates having a count no less than the minimum support count are frequent by definition, and therefore belong to L_k). C_k , however, can be huge, and so this could involve heavy computation. To reduce the size of C_k , the Apriori property is used as follows. Any $(k-1)$ -itemset that is not frequent cannot be a subset of a frequent k -itemset. Hence, if any $(k-1)$ -subset of a candidate k -itemset is not in L_{k-1} , then the candidate cannot be frequent either and so can be removed from C_k . This subset testing can be done quickly by maintaining a hash tree of all frequent itemsets.

NOTES

Generating Association Rules from Frequent Itemsets

Once the frequent itemsets from transactions in a database D have been found, it is straightforward to generate strong association rules from them (where strong association rules satisfy both minimum support and minimum confidence). This can be done using Equation (4.3) for confidence, i.e., :

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

The conditional probability is expressed in terms of itemset support count, where $\text{support_count}(A \cup B)$ is the number of transactions containing the itemsets $A \cup B$, and $\text{support_count}(A)$ is the number of transactions containing the itemset A . Based on this equation, association rules can be generated as follows:

- For each frequent itemset l , generate all nonempty subsets of l .
- For every nonempty subset s of l , output the rule

$$"s \Rightarrow (l - s)" \text{ if } \frac{\text{support_count}(l)}{\text{support_count}(s)} \geq \text{min_conf},$$

where min_conf is the minimum confidence threshold.

Because the rules are generated from frequent itemsets, each one automatically satisfies minimum support. Frequent itemsets can be stored ahead of time in hash tables along with their counts so that they can be accessed quickly.

Improving the Efficiency of Apriori

There are various methods involved to improve an Apriori algorithm. These methods are explained as follows:

Hash-Based Technique: This technique can be used to reduce the size of the candidate k -itemsets (C_k) for $k > 1$. The frequent 1-itemsets L_1 is generated from the candidate 1-itemsets in C_1 by scanning each transformation in the database. Now, L_1 generates L_2 , L_2 generates L_3 , and so on. These are stored in separate buckets defined in a hash table structure and increase the counts of the corresponding bucket. This scenario is explained in Table 4.1.

Table 4.1 Hash Table for Candidate 2 – Itemsets

Bucket Address	0	1	2	3	4	5
Bucket Count	2	2	4	2	2	4
Bucket Contents	{I1 , I4} {I3 , I5}	{I1 , I5} {I1 , I5}	{I2 , I3} {I2 , I3} {I2 , I3}	{I2 , I4} {I2 , I4}	{I2 , I5} {I2 , I5}	{I1 , I2} {I1 , I2} {I1 , I2}

NOTES

A hash table H_2 is created using hash function $h(x,y) = ((\text{order of } x) \times 10 + (\text{order of } y)) \bmod 6$. This hash table is produced by scanning the transactions table. If the support threshold is high than the bucket count in the hash table, then that item should be removed from the candidate set.

Transaction Reduction: This reduces the number of transactions scanned in future iterations. A transaction which does not contain any frequent k – itemsets cannot contain any $(k + 1)$ itemsets and such transaction is marked as removed.

Partitioning: Database scans for two frequent itemsets as shown in Figure 4.5.

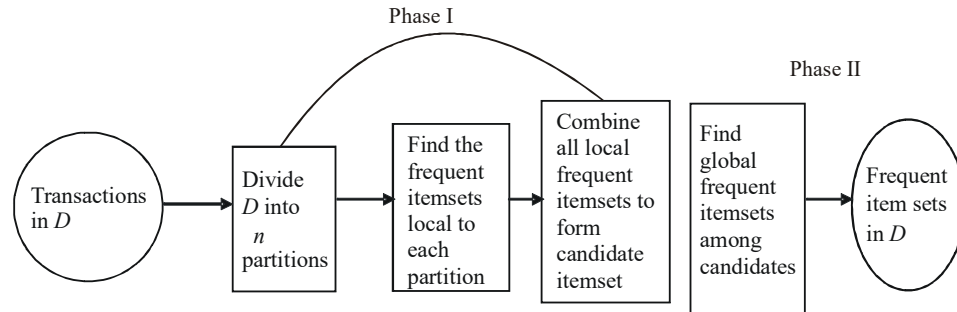


Fig. 4.5 Mining by Partitioning Data

It consists of two phases. In Phase I, the algorithm subdivides the transactions of D into n non-overlapping partitions. If the minimum support threshold for transactions in D is min_sup , then the minimum support count for a partition is $min_sup \times \text{the number of transactions}$ in that partition. For each partition, all frequent itemsets within the partition are found. These are referred to as local frequent itemsets.

The procedure employs a special data structure that, for each itemset, records the TIDs of the transactions containing the items in the itemset. This allows it to find all of the local frequent k -itemsets, for $k = 1, 2, \dots, n$, in just one scan of the database. A local frequent itemset may or may not be frequent with respect to the entire database, D . Any itemset that is potentially frequent with respect to D must occur as a frequent itemset in at least one of the partitions. Therefore, all local frequent itemsets are candidate itemsets with respect to D . The collection of frequent itemsets from all partitions forms the global candidate itemsets with respect to D . In Phase II, a second scan of D is conducted in which the actual support of each candidate is assessed in order to determine the global frequent itemsets. Partition size and the number of partitions are set so that each partition can fit into main memory and therefore be read only once in each phase.

Sampling: In this approach, a random sample s is taken for given data D . It searches frequent itemsets in s and does not consider D . A degree of accuracy is attempted against efficiency. This search is done in the main memory. Since the searching is done in s itemsets instead of D some of the global frequent itemsets are missed. A lower support threshold is used to overcome the missing values that find desired frequent itemsets to $s(L_s)$.

The rest part of database is computed the actual frequencies of each itemset specified L_s . Only one scan is required for D if L_s contains all the frequent itemsets. If one scan is not possible, then a second pass is done. The second pass finds the frequent itemsets. These were missed during processing. This approach gives the efficiency in computation.

Dynamic Itemset Counting

In this approach, blocks are available after partitioning the database. These blocks are marked by the starting points. At any of the starting point, new candidate itemsets can be inserted.

Different Types of Association Rules

In this section, we will focus on additional applications including multilevel association rules, quantitative association rules and multidimensional association rules in relational databases, data warehouses and transactional databases. Multilevel association rules work with different levels of abstraction. Multidimensional association rules involve more than one predicate or dimension. Quantitative association rule involves numeric attributes that have implicit ordering among values.

Mining Multilevel Association Rules

For many applications, it is not easy task to implement the strong association in data items at primitive or low levels due to sparsity of data at those levels. Strong association occurs during high levels of abstraction, which can be represented as commonsense knowledge. Commonsense may differ from one user to another. Thus, data mining provides mining association rules at the different of multiple levels involved in abstraction.

Let us consider a relevant set of transactional data be taken as specified in Table 4.2.

Table 4.2 Transactional Data

TID	Items Purchased
T100	Norton Antivirus
T200	Logitech - MX700 - Cordless Mouse
T300	MS - Office XP
T400	HP - Photomart - 7660
T500	IBM - ThinkPad - T40/2373

NOTES

NOTES

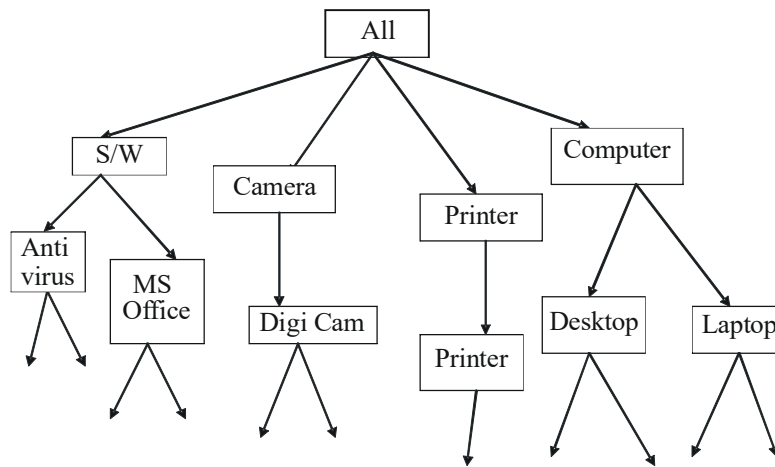


Fig. 4.6 Concept Hierarchy for Computer Items

With the help of Table 4.2, we can draw a tree-like structure from the set of low-level concepts that goes to higher levels. Data is generalized using low-level technique within the data specified by high-level concepts. This concept hierarchy is shown in Figure 4.6. It starts from Level 1 to Level 3. Here, Level 1 includes all, Level 2 includes S/W, camera, printer, computer, and so on. Numerical attributes involve concept hierarchies which can be generated with the help of clustered or discretization techniques. From Figure 4.6, it is not easy to find an interesting purchase pattern using primitive or raw-level data. Suppose Norton Anti virus or IBM ThinkPad T40 / 2373 involves transactions. At this level, strong association rules are not easily found. But it is easy to find strong association rules in the case of an IBM desktop or IBM laptop.

Association rules that are generated from mining data at multiple levels of abstraction are known as multiple-level or multilevel association rules. Using concept hierarchies, the multilevel association rules can be efficiently mined based on the support-confidence framework. Generally, a top-down strategy is employed, where counts are accumulated for the calculation of frequent itemsets at each concept level, starting at the concept Level 1 and working downward in the hierarchy toward the more specific concept levels, until no more frequent itemsets can be found.

For each level, Apriori algorithm or its variations may be used for discovering frequent itemsets. A number of variations to this approach are described below, where each variation involves “playing” with the support threshold in a slightly different way. The variations are illustrated in Figures 4.7 and 4.8, where nodes indicate an item or itemset that has been examined, and nodes with thick borders indicate that an examined item or itemset is frequent.

Using Uniform Minimum Support for All Levels

It is also referred to as uniform support. In uniform support, the same minimum support threshold is used when mining at each level of abstraction. For example,

in Figure 4.7, a minimum support threshold of 5% is used throughout (e.g., for mining from “computer” down to “laptop computer”). Both “computer” and “laptop computer” are found to be frequent, while “desktop computer” is not. When a uniform minimum support threshold is used, the search procedure is simplified. The method is also simple in that users are required to specify only one minimum support threshold. An Apriori-like optimization technique can be adopted, based on the knowledge that an ancestor is a superset of its descendants: The search avoids examining itemsets containing any item whose ancestors do not have minimum support.

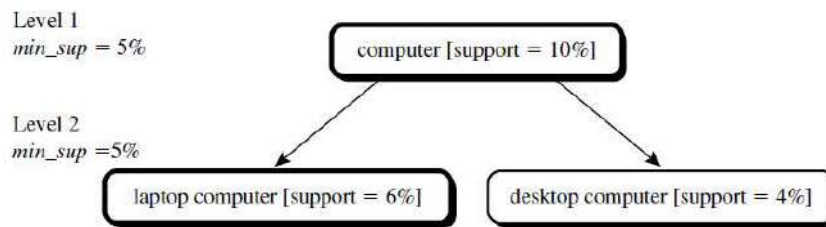


Fig. 4.7 Multilevel Mining with Uniform Support

Using Reduced Minimum Support at Lower Levels

It is also referred to as reduced support. Each level of abstraction has its own minimum support threshold. The deeper the level of abstraction, the smaller the corresponding threshold is. For example, in Figure 4.8, the minimum support thresholds for levels 1 and 2 are 5% and 3%, respectively. In this way, “computer,” “laptop computer,” and “desktop computer” are all considered frequent.

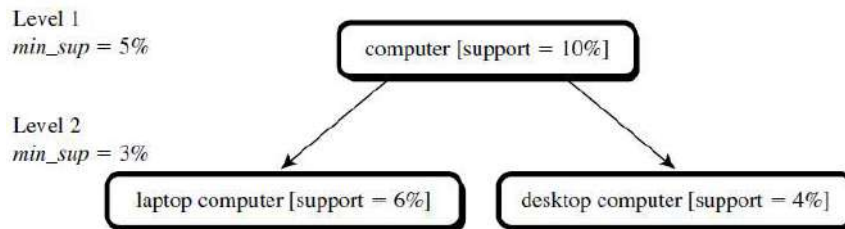


Fig. 4.8 Multilevel Mining with Reduced Support

Using Item or Group-Based Minimum Support

It is also referred to as group-based support. Because users or experts often have insight as to which groups are more important than others, it is sometimes more desirable to set up user-specific, item, or group based minimal support thresholds when mining multilevel rules. For example, a user could set up the minimum support thresholds based on product price, or on items of interest, such as by setting particularly low support thresholds for laptop computers and flash drives in order to pay particular attention to the association patterns containing items in these categories.

NOTES

NOTES

Mining Multidimensional Association Rules using Static Discretization of Quantitative Attributes

Quantitative attributes can be clustered or discretized before using the predefined concept hierarchies in mining. In this technique, interval labels replace the numeric values. In case of task relevant data in a relational table, itemset mining algorithms are used that modify easily for finding all frequent predicate sets. This technique does not involve frequent itemsets. The transformed multidimensional data constructs a data cube. These data cubes are suitable to mine the multidimensional association rules. Multidimensional space is required to compute the support and confidence of multidimensional association rules.

The quantitative association rules are mined by two quantitative rules. One is considered as the left hand side rule and the other is considered as categorical attribute based on the right hand side rule.

The quantitative association rules are mapped to Boolean association rules in which all attributes represent the categorical attribute or quantitative attribute. This mapping is applied in data mining. For example, in the table of database, instead of taking one field for each attribute, many fields can be taken for the attribute values. The value residing in the Boolean field representing (attribute_i, value_i) would be 1 if the attribute had value *i* in the original record; otherwise, the value will be 0. If a large domain of values is obtained, an obvious approach is to first partition values into intervals. It then maps each (attribute, interval) pair to a Boolean attribute. This algorithm is used to find Boolean association rules as well as quantitative association rules.

Consider a case of mapping for the non-key attributes of a table in which 'age' is partitioned into two intervals: 20..29 and 30..39. The categorical attribute 'married' has two Boolean attributes 'Married: Yes' and 'Married: No'.

This partition process is known as binning in which intervals represent bins. It supports the following binning strategies:

- **Equal Width Binning:** Each bin contains the same interval size.
- **Equal Frequency Binning:** Same number of tuples is assigned to each bin.
- **Clustering-Based Binning:** Clustering is performed on the quantitative attributes to group neighbouring points into the same bin.

Mining Frequent Itemsets without Candidate Generation

Frequent set played the essential functions in many data mining tasks. It attempt to find interesting patterns from databases, such as association rules, correlations, sequences, episodes, classifiers and clusters. The mining of association rules is one of the most popular problems. The identification of sets of items, products, symptoms and characteristics, which often occur together in the given database, can be seen as one of the most basic tasks in data mining. Though the Apriori

candidate generate-and-test method significantly reduces the size of candidate sets and lead to good performance gain, but it can suffer from two non-trivial costs which are mentioned as follows:

- It may need to generate a huge number of candidate sets.
- It may need to repeatedly scan the database and check a large set of candidates by pattern matching.

This problem can be solved by Frequency Pattern growth or FP growth. FP growth mines the complete set of frequent itemsets without candidate generation. Divide and conquer strategy is adopted by this method. It compresses the database which represents frequent items into Frequent Pattern tree, or FP tree, in which the itemset association information are retained. The compressed database is then divided into a set of conditional database, each associated with one frequent item or 'pattern fragment' and each such database are mined separately.

Let us create the FP tree for the following table:

Transaction ID	List of Item_ID
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

The first scan of the database is the same as Apriori, which derives the set of frequent items (1-itemsets) and their support counts (frequencies). Let the minimum support count be 2. The set of frequent items is sorted in the order of descending support count. This resulting set or list is denoted L. Thus, we have $L = \{\{I2: 7\}, \{I1: 6\}, \{I3: 6\}, \{I4: 2\}, \{I5: 2\}\}$.

An FP-tree is then constructed as follow:

First, create the root of the tree, labeled with "null." Scan database D a second time. The items in each transaction are processed in L order (i.e., sorted according to descending support count), and a branch is created for each transaction. For example, the scan of the first transaction, "T100: I1, I2, I5," which contains three items (I2, I1, I5 in L order), leads to the construction of the first branch of the tree with three nodes, $\{I2: 1\}$, $\{I1: 1\}$, and $\{I5: 1\}$, where I2 is linked as a child of the root, I1 is linked to I2, and I5 is linked to I1.

The second transaction, T200, contains the items I2 and I4 in L order, which would result in a branch where I2 is linked to the root and I4 is linked to I2. However, this branch would share a common prefix, I2, with the existing path for

NOTES

NOTES

T100. Therefore, we instead increment the count of the I2 node by 1, and create a newnode, {I4: 1}, which is linked as a child of {I2: 2}. In general, when considering the branch to be added for a transaction, the count of each node along a common prefix is incremented by 1, and nodes for the items following the prefix are created and linked accordingly.

To facilitate tree traversal, an item header table is built so that each item points to its occurrences in the tree via a chain of node-links. The tree obtained after scanning all of the transactions is shown in Figure 4.9 with the associated node-links. In this way, the problem of mining frequent patterns in databases is transformed to that of mining the FP-tree.

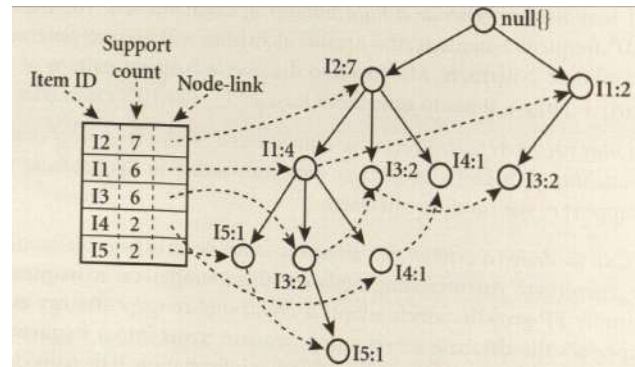


Fig. 4.9 The FP-Tree

The FP-tree is mined as follows. Start from each frequent length-1 pattern (as an initial suffix pattern), construct its conditional pattern base (a “sub-database,” which consists of the set of prefix paths in the FP-tree co-occurring with the suffix pattern), then construct its (conditional) FP-tree, and perform mining recursively on such a tree. The pattern growth is achieved by the concatenation of the suffix pattern with the frequent patterns generated from a conditional FP-tree.

Mining of the FP-tree is summarized in Table 4.3. We first consider I5, which is the last item in L, rather than the first. The reason for starting at the end of the list will become apparent as we explain the FP-tree mining process. I5 occurs in two branches of the FP-tree of Figure 4.9. The occurrences of I5 can easily be found by following its chain of node-links. The paths formed by these branches are {I2, I1, I5: 1} and {I2, I1, I3, I5: 1}. Therefore, considering I5 as a suffix, its corresponding two prefix paths are {I2, I1: 1} and {I2, I1, I3: 1}, which form its conditional pattern base. Its conditional FP-tree contains only a single path, {I2: 2, I1: 2}; I3 is not included because its support count of 1 is less than the minimum support count. The single path generates all the combinations of frequent patterns: {I2, I5: 2}, {I1, I5: 2}, {I2, I1, I5: 2}.

The FP-growth method transforms the problem of finding long frequent patterns to searching for shorter ones recursively and then concatenating the suffix. It uses the least frequent items as a suffix, offering good selectivity. The method substantially reduces the search costs.

When the database is large, it is sometimes unrealistic to construct a main memory based FP-tree. An interesting alternative is to first partition the database into a set of projected databases, and then construct an FP-tree and mine it in each projected database. Such a process can be recursively applied to any projected database if its FP-tree still cannot fit in main memory.

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

The frequent pattern generated for each node is shown in Table 4.3.

Table 4.3 Frequency Pattern Generated for each node

Item	Conditional Pattern Base	Conditional FP-Tree	Frequent Pattern 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, I1:2}, {I2:1}}	(I2:2)	{I2, I4:2}
I3	{I2, I1:2}, {I2:2}, {I1:2}}	(I2:4, I1:2), (I1:2), (I2:4)	{I2, I3:4}, {I1, I3:4}, {I2, I1, I3:2}, {I2, I1:4}
I1	{I2:4}}	(I2:4)	{I2, I1:4}

4.2.1 Association Rules from Relational Databases and Data Warehouses

In the relational database and data warehouse, data is related to sales and other. These data stored are in multidimensional database. In a relational database, record related items purchased while sale transaction is record and other attributes related to item like number of item purchased, price of the item purchased or the location of the sales area are recorded. A relational database also contain information related to customer about his purchases items, age of the customer, credit rating, occupation and the customer address.

According to the above considered database attribute or data warehouse dimension association rules contain multiple predicates as:

$$age(X, "25.....32) \wedge occupation(A, "customer") \Rightarrow buys(A, "item")$$

Association rule includes two or more than two predicates or dimensions which are termed as multidimensional association rule and these predicates, such as in above example (occupation, age, buys) occurs only one time in the rule. And the multidimensional rules which does not occurs no more than one time predicates are known as inter dimension association rule. The association rules where predicates occurs more than one time are known as mine multidimensional

NOTES

association rules and these rules contain hybrid dimension association rule, for example:

$$\text{age}(X, "25\dots32) \wedge \text{occupation}(A, "customer") \Rightarrow \text{buys}(A, "name of brand \text{ and name of item}")$$

NOTES

Attributes of the relational database and data warehouse are quantitative and categorical. Attributes which are categorical includes finite number of values which is possible and the attributes does not contain any order between the values, such as (brand, occupation). Categorical attributes are also termed as nominal attributes because values related to them are defined according to names of the things. Attributes which are quantitative includes numeric value and ordering between these values are implicit such as (price, age, earning).

The following are the different approaches used in quantitative attribute for mining the association rules:

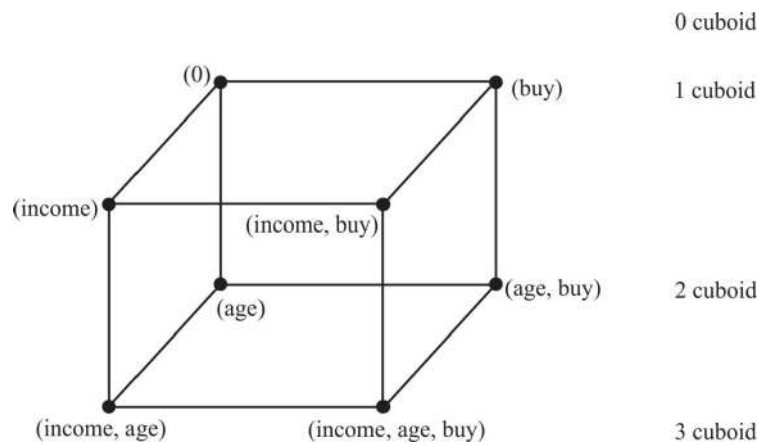
1. In first approach, quantitative attributes are discretized using the already defined hierarchy concepts. For example, the hierarchy concept for income is to replace the number with the interval like 0...10k, 11...20k, and so on. In this case, the discretization is predetermined and static. It is also called as static discretization of quantitative attribute for mining association rules.
2. In second approach, quantitative attributes are discretized or clustered into 'bins' according to the distributed data. These bins are again combined with other bins in different process of mining. This is why it is called as dynamic and established to fulfill the criteria of mining like increasing the confidence of the mined rule. This approach treats numeric attributes as a number or quantities instead of range.

Static Discretization of Quantitative Attribute for Mining Association Rules

In static discretization of quantitative attributes, the numeric attributes are discretized before mining using concept hierarchy and numeric values are changed into interval. When it is required the conceptual values can also generalized to next level. After that, store all relevant data into relation table. Whenever it is required use any relevant mining algorithm and then modify to get predicate sets then item sets. Instead of getting only one attribute, search all relevant attribute whereas as each attribute pair is called as itemset.

One more alternative is to store data into data cube. A data cube is best to determine the multidimensional association rule. It stores the aggregate (intersection) in multidimensional space which is necessary for providing support to multidimensional association rule.

Consider the following cuboid showing the data cube for dimension income, age and buys. When cuboid is n -dimensional then it is used to support count of n -predicates set.



The base cuboid displays all relevant data for dimension income, age and buys whereas the 2-D cuboid display the aggregate of age and income and similarly goes on.

Dynamic Discretization of Quantitative Attribute for Mining Association Rules

It can be simply called as mining quantitative association rules. In this rule, numeric values are discretized dynamically in order to satisfy minimum criteria and maximizing rule confidence. In this part, we focus on how to mine or get rule which has two quantitative attribute on left side and one categorical attribute on right side.

$$P_{quant1} \wedge P_{quant2} \Rightarrow P_{cate}$$

In above P_{quant1} and P_{quant2} are two quantitative attribute where P_{cate} is categorical attribute having relevant data.

Quantitative attribute have large variety of values. There are three binning strategies which are as follows:

- **Equal-Width Binning:** When interval sizes of each quantitative attribute are same then this binning is used.
- **Equal – Frequency Binning:** When each bin has same number of tuple then this binning is used.
- **Clustering –Based Binning:** When quantitative attribute are clustered to group into one bins then this binning is used.

Finding Frequent Predicate Set

When 2-D array have count distribution for each category then traverse that array to get the predicate set which satisfy the minimum confidence. Now apply rule generation algorithm on predicate set to determine strong association rule.

NOTES

Clustering the Association Rule

The strong association rule determine in previous step are mapped into 2-D grid. Suppose there are quantitative attribute income and age. The 2-D grid is used to determine right hand side of 2-D quantitative association rule. Suppose there are four following rules:

NOTES

$$age(Y, 24) \wedge income(Y, "21K...30K") \Rightarrow buys(Y, "LED")$$

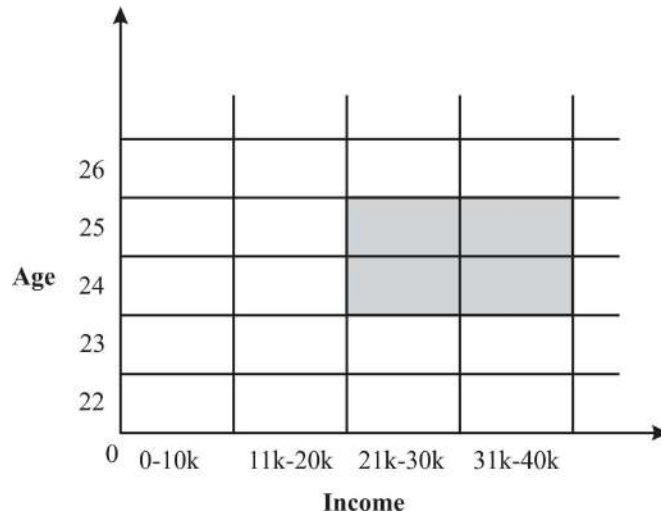
$$age(Y, 25) \wedge income(Y, "21K...30K") \Rightarrow buys(Y, "LED")$$

$$age(Y, 24) \wedge income(Y, "31K...40K") \Rightarrow buys(Y, "LED")$$

$$age(Y, 25) \wedge income(Y, "31K...40K") \Rightarrow buys(Y, "LED")$$

The above four rule looks similar so it is also possible to combine all rules into one simple rule as follows:

$$age(Y, 24...25) \wedge income(Y, "21K...40K") \Rightarrow buys(Y, "LED")$$



Now, ARCS algorithm is used for clustering purpose. This algorithm traverses the whole grid to find rectangular cluster rule. In this way, rules in 2D-grid combines each other to get resultant rule. Hence, this lead to further discretization of quantitative attributes.

Data mining is developed for the interactive mining which constitute multilevel information in the large relational database and data warehouse which includes following distinct features:

- OLAP technology and data cube are the techniques which include attribute oriented induction, progressive deepening related to multi-level mining rules, statistical analysis and meta rule guided knowledge mining.
- Some data mining functions, such as comparison, characterization, classification, association, clustering and prediction are also implemented.
- On a particular user-defined set of data, mining is done at the multilevel abstraction in the relational database and data warehouse by using database

query language, such as SQL. In this way, user adjust some thresholds perform roll-up or drill-down process, control process related to data mining at the multilevel abstraction includes various types of output.

- By implementing various types of techniques, relational database and data warehouse are integrated smoothly.
- Data mining process also include some expert defined set of group which is adjusted dynamically based on the data distribution and are generated automatically for the numerical attributes.
- For data mining, system also constitute client-server architecture which runs on the Windows and NT operating system which helps to communicate different database system by using technology ODBC.

Rule for relational database and data warehouse is related to the extracting frequent patterns, association or casual structure, interesting correlations between the set of items related to transactional database

For example – If a user wants to buy a computer science book through online book store then before purchasing there are some tips available for taking a look to the book to be purchase. Such that if the user wants to purchase a book name “Data mining concepts and techniques” when he clicked on that book then a list of the book get also displayed which will recommend user for further purchasing. These recommended books have 45% Database System and 20% Data Warehouse books.

- In the above example association rule are those rules related to list of recommended books 45% Database System and 20% Data Warehouse books at the time of purchasing the “Data mining concepts and techniques” book.
- These rules are used to fetch data (books name) from transaction database (books names which are already purchased by other user) to display list of book names which are useful for user and also makes the rules further stronger if user also purchases any recommended book.
- These rules are also helpful for making the market strategies related to the promotion of the specified book which will increase the sales of also other two recommended book

Association rule is mostly used in various areas, such as in market and risk management, telecommunication networks and inventory control. The data stored in the relational database are termed as bigger source for mining objects because of its highly structured data arrangement. In this, data is described by using set of attributes and it is stored in the form of table.

Check Your Progress

1. How are association rules tagged with data mining?
2. Define Apriori algorithm.

NOTES

NOTES

4.3 PARTITION ALGORITHM

Association rules are considered to be one of the major technologies in data mining. The collection of transactions each involving a set of items is regarded as the database. These rules are generally used to generate rules to perform market basket analysis (A market basket analysis corresponds to the sets of items that are purchased by the consumer during one visit to a super market). Here the *item set* is the set of items purchased by the customers.

Partition algorithm is one of the algorithms used to generate association rule. Here, the partitioning divides the database into non-overlapping subsets, which are individually considered as separate databases and all large item sets for that partition, called *local frequent itemsets*, are generated in one pass. Then apriori algorithm can be used efficiently on each partition if it fits entirely in main memory. Partitions are chosen in such a way that each partition can be accommodated in main memory. The only caveat with the partition method is that the minimum support used for each partition has a slightly different meaning from the original value. The minimum support is based on the size of the partition rather than the size of the database for determining local frequent (large) item sets. The actual support threshold value is the same as given earlier, but the support is computed only for a partition. At the end of pass one, we take the union of all frequent item sets from each partition. This forms the global candidate frequent item sets for the entire database. When these lists are merged, they may contain some false positives. That is, some of the item sets that are frequent (large) in one partition may not qualify in several other partitions and hence may not exceed the minimum support when the original database is considered. Note that there are no false negatives; no large item sets will be missed. The global candidate large item sets identified in pass one are verified in pass two; that is, their actual support is measured for the entire database. At the end of phase two, all global large item sets are identified. The partition algorithm lends itself naturally to a parallel or distributed implementation for better efficiency.

Check Your Progress

3. What is a database?
4. How are local frequent itemsets generated?

4.4 ANSWERS TO CHECK YOUR PROGRESS QUESTIONS

1. Association rules are tagged with data mining providing correlation relationships if a large data set of data items is provided.

2. The Apriori algorithm is the classical algorithm for data mining association rules. This algorithm is basically designed for databases containing transactions; examples include purchasing of a collection of items and details of Website searching.
3. The collection of transactions each involving a set of items is regarded as the database.
4. When the partitioning divides the database into non-overlapping subsets, which are individually considered as separate databases and all large itemsets for that partition, called *local frequent itemsets*, are generated in one pass.

NOTES

4.5 SUMMARY

- Association rules are applied for data and text mining in huge databases and are associated with correlation relationships.
- Association rules are tagged with data mining providing correlation relationships if a large data set of data items is provided.
- Some basic notations are used in association rules of data mining. These notations are known as ‘TISC’ representing Transaction, Itemset, Support and Confidence.
- A transaction consists of an itemset and a transaction identifier. A set of items is known as itemset which collectively works with transactions;
- A set of items is referred to as an itemset. An itemset that contains k items is a k-itemset. The set {computer, antivirus software} is a 2-itemset. The occurrence frequency of an itemset is the number of transactions that contain the itemset.
- The Apriori algorithm finds frequent itemsets using candidate generation. It was proposed by R. Agarwal and R. Srikant in 1994 for mining frequent itemsets for Boolean association rules.
- The Apriori algorithm is the classical algorithm for data mining association rules. This algorithm is basically designed for databases containing transactions; examples include purchasing of a collection of items and details of Website searching.
- Apriori uses a ‘bottom up’ approach, where frequent subsets are extended one item at a time (a step known as candidate generation), and groups of candidates are tested against the data. The algorithm terminates when no further successful extensions are found.
- Apriori is an influential algorithm for mining frequent itemsets for Boolean association rules. This algorithm uses prior knowledge of frequent itemset properties. This algorithm iteratively finds all possible itemsets that have support greater or equal to a given minimum support value.

NOTES

- Multilevel association rules work with different levels of abstraction. Multidimensional association rules involve more than one predicate or dimension. Quantitative association rule involves numeric attributes that have implicit ordering among values.
- Association rules that are generated from mining data at multiple levels of abstraction are known as multiple-level or multilevel association rules. Using concept hierarchies, the multilevel association rules can be efficiently mined based on the support-confidence framework.
- Quantitative attributes can be clustered or discretized before using the predefined concept hierarchies in mining. In this technique, interval labels replace the numeric values. In case of task relevant data in a relational table, itemset mining algorithms are used that modify easily for finding all frequent predicate sets.
- Frequent set played the essential functions in many data mining tasks. It attempt to find interesting patterns from databases, such as association rules, correlations, sequences, episodes, classifiers and clusters.
- The Frequency Pattern or FP-growth method transforms the problem of finding long frequent patterns to searching for shorter ones recursively and then concatenating the suffix. It uses the least frequent items as a suffix, offering good selectivity. The method substantially reduces the search costs.
- In static discretization of quantitative attributes, the numeric attributes are discretized before mining using concept hierarchy and numeric values are changed into interval. When it is required the conceptual values can also generalized to next level.
- Partition algorithm is one of the algorithms used to generate association rule. Here, the partitioning divides the database into non-overlapping subsets, which are individually considered as separate databases and all large item sets for that partition, called *local frequent itemsets*, are generated in one pass.

4.6 KEY WORDS

- **Apriori Algorithm:** An influential algorithm for mining frequent itemsets for Boolean association rules.
- **Transaction Reduction:** This reduces the number of transactions scanned in future iterations, a transaction which does not contain any frequent k -itemsets cannot contain any $(k + 1)$ itemsets and such transaction is marked as removed.

4.7 SELF ASSESSMENT QUESTIONS AND EXERCISES

Short Answer Questions

1. What are association rules?
2. What are the benefits of the Apriori algorithm?
3. How can you improve the efficiency of Apriori algorithm?
4. Define the term sampling.
5. Name the different types of association rules.
6. Which binning strategies are used for dynamic discretization of quantitative attribute for mining association rules?

Long Answer Questions

1. What are the association rules? Describe its types with the help of diagram.
2. Describe the Apriori algorithm with the help of an example.
3. Discuss the pseudocode implemented in an Apriori algorithm.
4. Explain the different types of association rules with the help of examples.
5. How the frequent set plays the essential functions in many data mining tasks? Explain with the help of an example.
6. Describe the different approaches used in quantitative attribute for mining the association rules.

4.8 FURTHER READINGS

- Han, Jiawei, Micheline Kamber and Jian Pei. 2011. *Data Mining: Concepts and Techniques*, 2nd edition. Amsterdam: Elsevier.
- Pujari, Arun K. 2010. *Data Mining Techniques*, 2nd edition. United States: Universities Press.
- Anahory, Sam and Dennis Murray. 1997. *Data Warehousing in the Real World: A Practical Guide for Building Decision Support Systems*, 1st edition. Boston: Addison Wesley.
- Witten, I. H. and Eibe Frank. 2005. *Data Mining: Practical Machine Learning Tools and Techniques*, 2nd edition. Amsterdam: Elsevier.
- Soman, K. P., Shyam Diwakar and V. Ajay. 2006. *Insight Into Data Mining: Theory and Practice*. New Delhi: PHI.

NOTES

UNIT 5 AR ALGORITHMS

NOTES

Structure

- 5.0 Introduction
- 5.1 Objectives
- 5.2 Pincer Search Algorithm
- 5.3 Dynamic Item Set Algorithm
- 5.4 FP Tree Growth Algorithm
- 5.5 Answers to Check Your Progress Questions
- 5.6 Summary
- 5.7 Key Words
- 5.8 Self Assessment Questions and Exercises
- 5.9 Further Readings

5.0 INTRODUCTION

Association rule mining is the one of the most important technique of the data mining. Its aim is to extract interesting correlations, frequent patterns and association among set of items in the transaction database. In the previous chapter, we have talked about the Apriori association rule mining and it was the first algorithm for finding frequent item sets and strong association rules. Discovering frequent item sets is a key problem in finding associations amongst the transactions. Apriori algorithm operates in a bottom-up breadth-first search direction. The computation starts from frequent 1-item sets and continues until the maximal (length) frequent item sets are found. During the execution, every frequent item set is explicitly considered. Such algorithms perform reasonably well when all maximal frequent item sets are short. However, performance drastically decreases when some of the maximal frequent item sets are relatively long. Therefore, some of the drawbacks it suffers are:

Disadvantages of Apriori Algorithm

- The number of candidate sets generated are very high. For e.g. if there are 10^4 frequent 1-itemset, the number of frequent 2-itemset generated would be around 10^7 .
- The repeated number of scans of dataset are needed to determine the support of candidate item sets which prove to be very costly.
- The candidate item sets are matched with the help of pattern matching which is again very expensive.

To cater these drawbacks several other algorithms are suggested such as Pincer Search Algorithm, Dynamic Item set Algorithm and FP Tree Growth Algorithm.

5.1 OBJECTIVES

After going through this unit, you will be able to:

- Explain Pincer search algorithm and its advantages
 - Discuss dynamic item set algorithm
 - Understand FP tree growth algorithm
-

5.2 PINCER SEARCH ALGORITHM

As opposed to Apriori algorithm, Pincer search algorithm works on both bottom-up and top-down direction. It attempts to find the frequent item sets in a bottom – up manner but, at the same time, it maintains a list of maximal frequent item sets. While making a database pass, it also counts the support of these candidate maximal frequent item sets to see if any one of these is actually frequent. In that event, it can conclude that all the subsets of these frequent sets are going to be frequent and, hence, they are not verified for the support count in the next pass and due to this we may discover a very large maximal frequent item set very early in the algorithm. If this set subsumes all the candidate sets of level k , then we need not proceed further and thus we save many database passes. Clearly, the pincer – search has an advantage over a priori algorithm when the largest frequent item set is long.

Algorithm

In this algorithm, in each pass, in addition to counting the supports of the candidate in the bottom – up direction, it also counts the supports of the item sets of some item sets using a top – down approach. These are called the Maximal Frequent Candidate Set (MFCS). This process helps in pruning the candidate sets very early on in the algorithm. If we find a maximal frequent set in this process, then it is recorded in the MFCS.

Consider a pass k , during which item sets of size k are to be classified. If some item set that is an element of the MFCS, say X , of cardinality greater than k is found to be frequent in this pass, then all its subsets must be frequent. Therefore all of its subsets of cardinality k can be pruned from the set of candidate sets considered in the bottom – up direction in the pass. This subset and their super sets would not be candidates throughout the rest of the execution therefore improving the performance. Similarly, when a new infrequent item set is found in the bottom – up direction, the algorithm will use it to update the MFCS. The subsets of the MFCS should not contain this in frequent item sets.

The MFCS initially contains a single element, the item set of cardinality n containing all the elements of the database. If some $m - 1$ – item sets are infrequent after the first pass (after reading the database once), the MFCS will have one element of cardinality $n - m$. removing the m infrequent items from the initial element

NOTES

of the MFCS, generates this item set. In this case, the top – down search goes down m levels in one pass. Unlike the search in the bottom – up direction which goes in one – pass, the top – down search can go down many levels in one pass. This is because we may discover a maximal frequent set very early in the algorithm.

NOTES

Algorithm: MFCS-gen

Input: Old MFCS and the infrequent set S_k found in pass k

Output: New MFCS

1. for all item sets $s \in S_k$
2. for all item sets $m \in \text{MFCS}$
3. if s is a subset of m
4. $\text{MFCS} := \text{MFCS} \setminus \{m\}$
5. for all items $e \in \text{item set } s$
6. if $m \setminus \{e\}$ is not a subset of any item set in the MFCS
7. $\text{MFCS} := \text{MFCS} \cup \{m \setminus \{e\}\}$
8. return MFCS

Algorithm: The recovery procedure

Input: C_{k+1} from join procedure, L_k , and current MFS

Output: a complete candidate set C_{k+1}

1. for all item sets l in L_k
2. for all item sets m in MFS
3. if the first $k - 1$ items in l are also in m
4. /* suppose $m.\text{item}_j = l.\text{item}_{k-1}$ */
5. for i from $j + 1$ to $|m|$
6. $C_{k+1} := C_{k+1} \cup \{l.\text{item}_1, l.\text{item}_2, \dots, l.\text{item}_k, m.\text{item}_i\}$

Algorithm: New prune procedure

Input: current MFCS and C_{k+1} after join and recovery procedures

Output: final candidate set C_{k+1}

1. for all itemsets c in C_{k+1}
2. if c is not a subset of any itemset in the current MFCS
3. delete c from C_{k+1}

Algorithm: The Pincer-Search algorithm

Input: a database and a user-defined minimum support

Output: MFS which contains all maximal frequent item sets

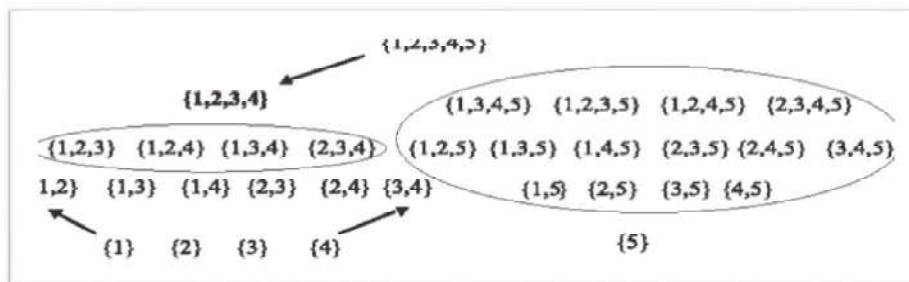
1. $L_0 := \emptyset$; $k := 1$; $C_1 := \{\{i\} \mid i \in I\}$
2. $\text{MFCS} := \{\{1, 2, \dots, n\}\}$; $\text{MFS} := \emptyset$
3. while $C_k \neq \emptyset$
4. read database and count supports for C_k and MFCS
5. remove frequent item sets from MFCS and add them to MFS
6. $L_k := \{\text{frequent item sets in } C_k\} \setminus \{\text{subsets of MFS}\}$
7. $S_k := \{\text{infrequent item sets in } C_k\}$
8. call the MFCS-gen algorithm if $S_k \neq \emptyset$
9. call the join procedure to generate C_{k+1}
10. if any frequent item set in C_k is removed in line 6
11. call recovery procedure to recover candidates to C_{k+1}
12. call new prune procedure to prune candidates in C_{k+1}
13. $k := k + 1$
14. end-while
15. return MFS

Consider a database containing five distinct items, 1, 2, 3, 4, and 5. There are four transactions in this database: {1,2,3,4,5}, {1,3}, {1,2}, and {1,2,3,4}. The minimum support is set to 0.5.

Transaction	Items
1	{1,2,3,4,5}
2	{1,3}
3	{1,2}
4	{1,2,3,4}

NOTES

In the first pass, all five 1-item sets are the candidates for the bottom-up search and the 5-itemset {1, 2, 3, 4, 5} is the candidate for the top-down search. After the support counting phase, infrequent item set {5} is discovered by the bottom-up search and this information is shared with the top-down search. This infrequent item set {5} not only allows the bottom-up search to eliminate its supersets as candidates but also allows the top-down search to eliminate its supersets as candidates in the second pass. In the second pass, the candidates for the bottom-up search are {1,2}, {1,3}, {1,4}, {2,3}, {2,4}, and {3,4}. Item sets {1, 5}, {2,5}, {3,5}, and {4,5} are not candidates, since they are supersets of {5}. The only candidate for the top-down search in the second pass is {1, 2, 3, 4}, since all the other 4-subsets of {1, 2, 3, 4, 5} are supersets of {5}. After the second support counting phase, {1, 2, 3, 4} is discovered to be frequent by the top-down search. This information is shared with the bottom-up search. All of its subsets are frequent and need not be examined. In this example, item sets {1,2,3}, {1,2,4}, {1,3,4}, and {2,3,4} will not be candidates for our bottom-up or top-down searches. After that, the program can terminate, since there are no candidates for either bottom-up or top-down searches.



Advantages of Pincer Search Algorithm

- It reduces the number of times the database is read.
- It also reduces the number of candidates considered for frequent item set.
- The performance is improved especially when some maximal frequent item sets are long.

NOTES

Check Your Progress

1. How does a Pincer search algorithm work?
2. What is a maximal frequent candidate set?

5.3 DYNAMIC ITEM SET ALGORITHM

Dynamic Item set Counting (DIC) is an algorithm which reduces the number of passes made over the data while keeping the number of item sets which are counted in any pass relatively low. It works in a way such that in the first M transactions the algorithm starts counting the 1-itemsets. After M transactions for a given minimum support threshold, if any of the item sets exceeds the minimum support threshold M transactions, then start counting the 2-itemsets count before waiting for a complete scan of database. In this way, it keeps on increasing items in the item set after every M transactions.

For example, if we are mining 40,000 transactions and $M = 10000$, we will count all the 1-itemsets in the first 40,000 transactions. However, we will begin counting 2-itemsets after the first 10,000 transactions have been read. We will begin counting 3-itemsets after 20,000 transactions and so on. Once we get to the end of the file, we will stop counting the 1-itemsets and go back to the start of the file to count the 2 and 3-itemsets. After the first 10,000 transactions, we will finish counting the 2-itemsets and after 20,000 transactions, we will finish counting the 3-itemsets. In total, we have made 1.5 passes over the data instead of the 3 passes a level-wise algorithm would make.

Algorithm

The DIC algorithm, described here, marks item sets in four different possible ways:

1. **Solid box** - confirmed large item set - an item set we have finished counting that exceeds the support threshold.
2. **Solid circle** - confirmed small item set - an item set we have finished counting that is below the support threshold.
3. **Dashed box** - suspected large item set - an item set we are still counting that exceeds the support threshold.
4. **Dashed circle** - suspected small item set - an item set we are still counting that is below the support threshold.

The DIC algorithm works as follows:

Step 1: The empty item set is marked with a solid box. All the 1-itemsets are marked with dashed circles. All other item sets are unmarked as shown in Figure 5.1(a).

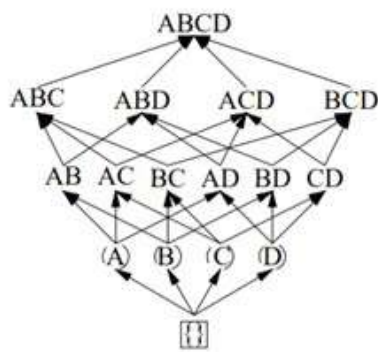


Fig.5.1(a) Start of DIC algorithm

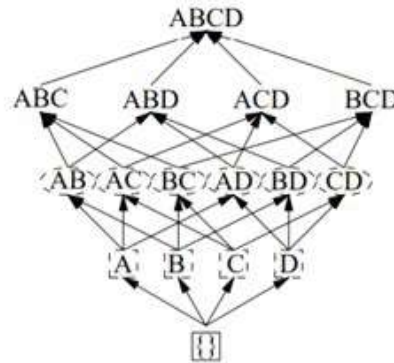


Fig. 5.1(b) After M transactions

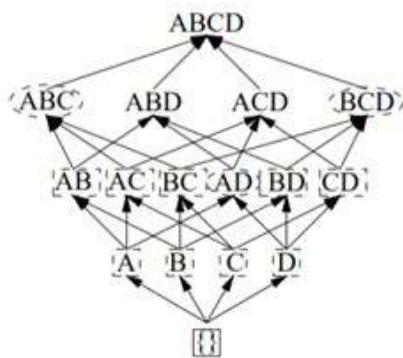


Fig. 5.1 (c) After 2M transactions

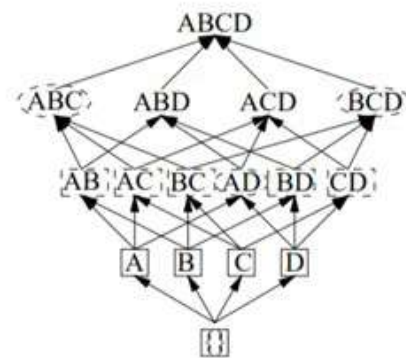


Fig. 5.1(d) After one pass

Step 2: Read M transactions. We experimented with values of M ranging from 100 to 10,000. For each transaction, increment the respective counters for the item sets marked with dashes.

Step 3: If a dashed circle has a count that exceeds the support threshold, turn it into a dashed square. If any immediate superset of it has all of its subsets as solid or dashed squares, add a new counter for it and make it a dashed circle. (See Figures 5.1(c) and 5.1(d))

Step 4: If a dashed item set has been counted through all the transactions, make it solid and stop counting it.

Step 5: If we are at the end of the transaction file, rewind to the beginning.

NOTES

NOTES**Dynamic Item set Algorithm**

```

SS = E // solid square (frequent)
SC = E // solid circle (infrequent)
DS = E // dashed square (suspected frequent)
DC = {all 1-itemsets} // dashed circle (suspected infrequent)
while (DS!= 0) or (DC != 0) do begin
  read M transactions from database into T

  for all transactions t ∈ T do begin
    //increment the respective counters of the item sets
    marked with dash
    for each item set c in DS or DC do begin
      If (c ∈ t) then
        c.counter++;
    for each item set c in DC
      if(c.counter ≥ threshold ) then
        move c from DC to DS ;
        if ( any immediate superset sc of c has
          all of its subsets in SS or DS ) then
          add a new item set sc in DC ;
    end
  for each item set c in DS
    if(c has been counted through all transactions
    ) then
      move it into SS ;
  for each item set c in DC
    if(c has been counted through all transactions
    ) then
      move it into SC ;
  end
end
end

```

This way DIC starts counting just the 1-itemsets and then quickly adds counters 2, 3, 4, ..., k-item sets. After just a few passes over the data (usually less than two for small values of M) it finishes counting all the item sets. Ideally, we would want M to be as small as possible so we can start counting item sets very early in step 3. However, steps 3 and 4 incur considerable overhead so we do not reduce M below 100.

Advantages of DIC Algorithm

- The number of passes is less if data is homogenous
- Has the flexibility of adding & deleting datasets on the fly
- This algorithm can be extended to parallel version

Check Your Progress

3. What is DIC algorithm?
4. What are the ways in which the DIC algorithm marks item sets?
5. Give an advantage of DIC algorithm.

5.4 FP TREE GROWTH ALGORITHM

The FP-Tree Growth Algorithm is an alternative algorithm used to find frequent item sets. It is different from the Apriori Algorithm explained in previous sections in a way that it uses a FP-tree to encode the data set and then extract the frequent item sets from this tree. Like Apriori, the FP-growth algorithm begins by counting the number of times individual items (i.e., attribute–value pairs) occur in the dataset. The first pass is followed by a second pass where tree structure is created. Initially, the tree is empty and it grows as each instance in the dataset is inserted into it. For the tree to be compact, the item sets need to be sorted in a descending order of their frequency of their occurrence in the dataset which was recorded in the first pass. This tree in turn is processed to find the large item sets. Individual items in each instance that do not meet the minimum support threshold are not inserted into the tree, effectively removing them from the dataset. There might be many instances of the items that occur most frequently individually, resulting in a high degree of compression close to the tree’s root.

NOTES

Algorithm

The illustration given below is of the FP Tree Growth Algorithm, followed by the algorithm itself. In this example the transaction database D. There are nine transactions in the database. We want to find the associations between these items taking the minimum support count of 2. The following steps are followed for the FP Tree Growth algorithm:

TID	Item ID
T1	I1, I2, I5
T2	I2, I4
T3	I2, I3
T4	I1, I2, I4
T5	I1, I3
T6	I2, I3
T7	I1, I3
T8	I1, I2, I3, I5
T9	I1, I2, I3

Step 1: The first pass consists of finding the frequency of individual attribute for all the possible values.

Table 5.1 Individual item with frequency

Attribute value	Frequency
I1	6
I2	7
I3	6
I4	2
I5	2

NOTES**Table 5.2** Individual item with frequency sorted by frequency

Attribute value	Frequency
I2	7
I1	6
I3	6
I4	2
I5	2

Step 2: The table 5.1 shows the individual item with their frequency and the table 5.2 is the sorted table by frequency. Next step is to find the data instances with the items in each instance sorted into descending frequency order.

TID	Item ID
T1	I2=7, I1=6, I5=2
T2	I2=7, I4=2
T3	I2=7, I3=6
T4	I2=7, I1=6, I4=2
T5	I1=6, I3=6
T6	I2=7, I3=6
T7	I1=6, I3=6
T8	I2=7, I1=6, I3=6, I5=2
T9	I2=7, I1=6, I3=6

Step 3: Create a FP tree as follows: An FP-tree is then constructed as follows. The construction of FP Tree starts with the creation of the root of the tree labeled as “null”. The database D is scanned a second time. A branch for each transaction is created with items in L order (i.e., sorted in descending support count). For e.g. first transaction, “T1: I1, I2, I5,” containing three items (I2, I1, I5 in L order), leads to the construction of the first branch of the tree with three nodes, [I2: 1], [I1: 1], and [I5: 1], where I2 is linked as a child to the root, I1 is linked to I2, and I5 is linked to I1. Similarly, the second transaction, T2, contains the items I2 and I4 in L order, which would result in a branch where I2 is linked to the root and I4 is linked to I2. However, this branch would share a common prefix, I2, with the existing path for T1. Therefore, increment the count of the I2 node by 1, and create a new node, [I4: 1], which is linked as a child to [I2: 2]. To facilitate tree traversal, an item header table is built so that each item points to its occurrences in the tree via a chain of node-links. The tree obtained after scanning all the transactions is shown in Figure 5.2 with the associated node-links.

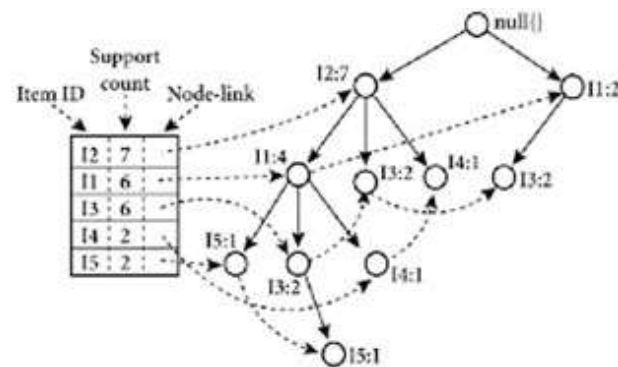


Fig. 5.2 Tree after scanning all the transactions

Step 4: For finding the frequent patterns in databases, FP-tree needs to be mined which is done as follows. Start from each frequent length-1 pattern (as an initial suffix pattern), construct its conditional pattern base (a “sub-database,” which consists of the set of prefix paths in the FP-tree co-occurring with the suffix pattern), then construct its (conditional)FP-tree, and perform mining recursively on the tree. The pattern growth is achieved by the concatenation of the suffix pattern with the frequent patterns generated from a conditional FP-tree. Mining of the FP-tree is summarized in Table 5.3 and detailed as follows.

Table 5.3 Summary of mining of the FP tree

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: 2}, {I2: 2}, {I1: 2} }	[2: 4, I1: 2], [I1: 2]	{I2, I3: 4}, {I1, I3: 4}, {I2, I1, I3: 2}
I1	{ {I2: 4} }	[I2: 4]	{I2, I1: 4}

We first consider I5, which is the last item in L, rather than the first. I5 occurs in two FP-tree branches of Figure above. (The occurrences of I5 can easily be found by following its chain of node-links.) The paths formed by these branches are [I2, I1, I5: 1] and [I2, I1, I3, I5: 1]. Therefore, considering I5 as a suffix, its corresponding two prefix paths are [I2, I1: 1] and [I2, I1, I3: 1], which form its conditional pattern base. Using this conditional pattern base as a transaction database, we build an I5-conditional FP-tree, which contains only a single path, [I2: 2, I1: 2]; I3 is not included because its support count of 1 is less than the minimum support count. The single path generates all the combinations of frequent patterns: {I2, I5: 2}, {I1, I5: 2}, {I2, I1, I5: 2}. For I4, its two prefix paths form the conditional pattern base, { {I2, I1: 1}, {I2: 1} }, which generates a single-node conditional FP-tree, [I2: 2], and derives one frequent pattern, {I2, I4: 2}. Similar to the preceding analysis, I3’s conditional pattern base is { {I2, I1: 2}, {I2: 2}, {I1: 2} }. Its conditional FP-tree has two branches, [I2: 4, I1: 2] and [I1: 2], as shown in Figure 5.3 which generates the set of patterns { {I2, I3: 4}, {I1, I3: 4}, {I2, I1, I3: 2} }.

NOTES

NOTES

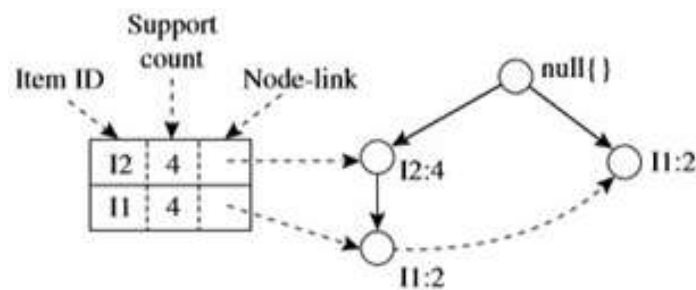


Fig. 5.3 Conditional FP-tree

Finally, I1's conditional pattern base is $\{\{I2: 4\}\}$, with an FP-tree that contains only one node, [I2: 4], which generates one frequent pattern, {I2, I1: 4}.

Advantages of FP Growth Algorithm

- It is much faster than Apriori algorithm.
- In this algorithm no candidate generation is required.
- It requires only two passes over dataset.

Check Your Progress

6. How is FP – tree algorithm different from Apriori?
7. Give the advantages of FP growth algorithm.

5.5 ANSWERS TO CHECK YOUR PROGRESS QUESTIONS

1. Pincer search algorithm works on both bottom-up and top-down direction. It attempts to find the frequent item sets in a bottom – up manner but, at the same time, it maintains a list of maximal frequent item sets.
2. Maximal Frequent Candidate Set (MFCS) is the process that helps in pruning the candidate sets very early on in the algorithm. If we find a maximal frequent set in this process, then it is recorded in the MFCS.
3. Dynamic Item set Counting (DIC) is an algorithm which reduces the number of passes made over the data while keeping the number of item sets which are counted in any pass relatively low.
4. The DIC algorithm marks item sets in four different possible ways i.e. solid box, solid circle, dashed box and dashed circle.
5. An advantage of DIC Algorithm is that it has the flexibility of adding & deleting datasets on the fly.

6. FP-tree algorithm is different from the Apriori Algorithm in a way that it uses a FP-tree to encode the data set and then extract the frequent item sets from the tree.
7. Advantages of FP growth algorithm are as follows:
 - It is much faster than Apriori algorithm.
 - In this algorithm no candidate generation is required.
 - It requires only two passes over dataset.

NOTES

5.6 SUMMARY

- Association rule mining is the one of the most important technique of the data mining. Its aim is to extract interesting correlations, frequent patterns and association among set of items in the transaction database.
- Opposed to Apriori algorithm, Pincer search algorithm works on both bottom-up and top-down direction. It attempts to find the frequent item sets in a bottom – up manner but, at the same time, it maintains a list of maximal frequent item sets.
- The pincer search has an advantage over a priori algorithm when the largest frequent item set is long.
- In pincer search algorithm, in each pass, in addition to counting the supports of the candidate in the bottom – up direction, it also counts the supports of the item sets of some item sets using a top – down approach. These are called the Maximal Frequent Candidate Set (MFCS).
- When a new infrequent item set is found in the bottom – up direction, the algorithm will use it to update the MFCS. The subsets of the MFCS should not contain this in frequent item sets.
- The MFCS initially contains a single element, the item set of cardinality n containing all the elements of the database.
- Advantages of Pincer Search Algorithm are:
 - i. It reduces the number of times the database is read.
 - ii. It also reduces the number of candidates considered for frequent item set.
 - iii. The performance is improved especially when some maximal frequent item sets are long.
- Dynamic Item set Counting (DIC) is an algorithm which reduces the number of passes made over the data while keeping the number of item sets which are counted in any pass relatively low.

NOTES

- Advantages of DIC Algorithm are:
 - i. The number of passes is less if data is homogenous
 - ii. Has the flexibility of adding & deleting datasets on the fly
 - iii. This algorithm can be extended to parallel version
- The FP-Tree Growth Algorithm is an alternative algorithm used to find frequent item sets.
- Advantages of FP growth algorithm are:
 - i. It is much faster than Apriori algorithm.
 - ii. In this algorithm no candidate generation is required.
 - iii. It requires only two passes over dataset.

5.7 KEY WORDS

- **Algorithm:** A process or set of rules to be followed in calculations or other problem-solving operations, especially by a computer.
- **Suffix:** A morpheme added at the end of a word to form a derivative.
- **Cardinality:** The number of elements in a set or other grouping, as a property of that grouping.

5.8 SELF ASSESSMENT QUESTIONS AND EXERCISES

Short Answer Questions

1. What are the disadvantages of Apriori Algorithm?
2. Define the FP Tree growth Algorithm.
3. Give the advantages of Pincer Search Algorithm?
4. Discuss the working of DIC algorithm with steps.

Long Answer Questions

1. Describe the ways in which the DIC algorithm marks item sets.
2. Explain DIC algorithm with an example.
3. Explain the step by step working of Pincer search algorithm. Give its advantages.
4. Give an illustration of the FP tree growth algorithm along with the algorithm.

5.9 FURTHER READINGS

- Han, Jiawei, Micheline Kamber and Jian Pei. 2011. *Data Mining: Concepts and Techniques*, 2nd edition. Amsterdam: Elsevier.
- Pujari, Arun K. 2010. *Data Mining Techniques*, 2nd edition. United States: Universities Press.
- Anahory, Sam and Dennis Murray. 1997. *Data Warehousing in the Real World: A Practical Guide for Building Decision Support Systems*, 1st edition. Boston: Addison Wesley.
- Witten, I. H. and Eibe Frank. 2005. *Data Mining: Practical Machine Learning Tools and Techniques*, 2nd edition. Amsterdam: Elsevier.
- Soman, K. P., Shyam Diwakar and V. Ajay. 2006. *Insight Into Data Mining: Theory and Practice*. New Delhi: PHI.

NOTES

UNIT 6 CLASSIFICATION

NOTES

Structure

- 6.0 Introduction
- 6.1 Objectives
- 6.2 Decision Tree Classification
- 6.3 Bayesian Classification
- 6.4 Classification by Back Propagation
- 6.5 Answers to Check Your Progress Questions
- 6.6 Summary
- 6.7 Key Words
- 6.8 Self Assessment Questions and Exercises
- 6.9 Further Readings

6.0 INTRODUCTION

Classification can be performed on structured or unstructured data. Classification is a method where we classify data into a given number of classes. The main objective of a classification problem is to classify the category/class to which a new data will fall under.

Few of the terminologies encountered in machine learning classification are:

- **Classifier:** An algorithm that maps the input data to a specific category.
- **Classification model:** A classification model tries to draw some conclusion from the input values given for training. It will predict the class labels/categories for the new data.
- **Feature:** A feature is an individual measurable property of a phenomenon being observed.
- **Binary Classification:** Classification task with two possible outcomes. Eg: Gender classification (Male / Female)
- **Multi class classification:** Classification with more than two classes. In multi class classification each sample is assigned to one and only one target label. Eg: An animal can be cat or dog but not both at the same time
- **Multi label classification:** Classification task where each sample is mapped to a set of target labels (more than one class). Eg: A news article can be about sports, a person, and location at the same time.

6.1 OBJECTIVES

After going through this unit, you will be able to:

- Understand the decision tree classification

- Explain the Bayesian classification
- Discuss the classification by back propagation

6.2 DECISION TREE CLASSIFICATION

NOTES

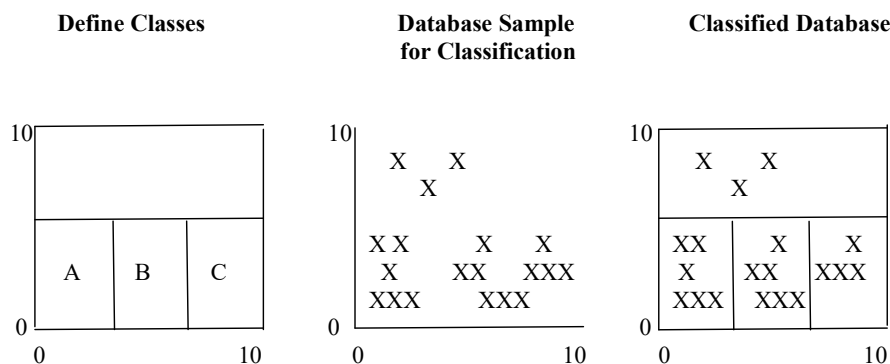
Classification and prediction are used for data analysis. They are used to define models describing important data classes or to predict future data trends. Such analysis gives good results or we may say, better understanding of large-scale data. Classification predicts categorical labels.

Suppose that one of the private hospitals provides three types of treatment, based on allopathy, homeopathy and ayurvedic principles and practices. Then a ‘classification model’ can be built to categorize each type of treatment.

Prediction may be considered as a type of classification where an attribute is bracketed in a class itself. Thus, if we focus on any one of the treatment categories, then it is categorized as a ‘prediction model’.

Steps in Database Classification

The following classification is for tuples in the range of 0 to 10:



Many classification and prediction methods have been proposed by researchers in machine learning, pattern recognition and statistics.

Classification and prediction have numerous applications including fraud detection, performance prediction, manufacturing and medical diagnosis.

Basic Concepts

Classification and prediction are two different approaches of data analysis that are used to identify models to categorize data classes or to predict future directions of data trends. While classification deals with categories of labels, prediction deals with continuous valued functions.

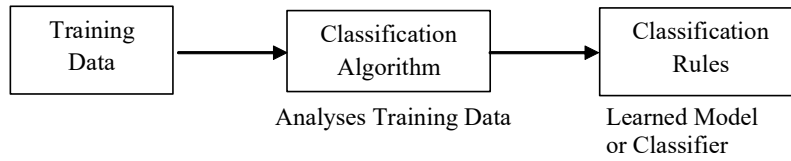
Classification

Classification can be explained by a two-step process given in the following block diagram.

Step 1: Model Construction

Build a model taking data from training set.

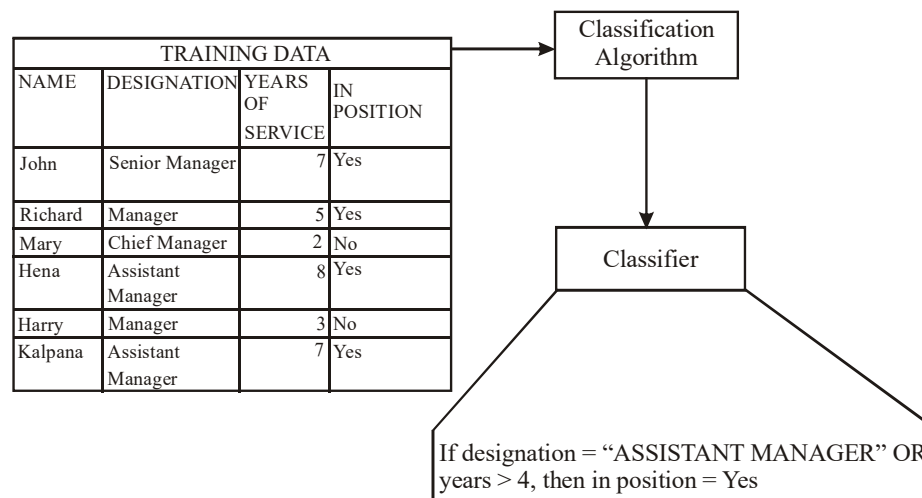
NOTES



In constructing a model, first gather training data. Gathered training data is tabulated for the following data.

Name	Designation	Years of Services	In Position
John	Senior Manager	7	Yes
Richard	Manager	5	Yes
Mary	Chief Manager	2	No
Hena	Assistant Manager	8	Yes
Harry	Manager	3	No
Kalpana	Assistant Manager	7	Yes

Each tuple or sample belongs to a predefined class. Class is determined by class label attribute. The set of training data is used to construct the model. The model is a classifier and this has a set of classification rules. This may be a decision tree or mathematical formulae.

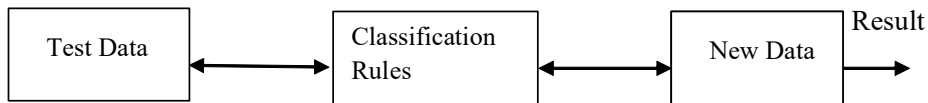


Class labels are also known as ‘supervised learning’. It is different from ‘unsupervised learning’ or clustering where class labels are not known.

Once a classifier is created, it should be evaluated for accuracy.

Step 2: Model Uses

Evaluate the model for accuracy and use it for classifying new data.



Test data verifies the accuracy of the classification rules.

If the constructed model is correct, then it can be used to classify future data that may be unknown. This needs estimation on accuracy of the model. For this, a known class of the test tuple is compared against the result produced by the model.

Accuracy is measured according to the following equation:

Accuracy rate = Percentage of the tests tuples accurately classified by the model

Prediction

As prediction deals with continuous values, one example may be that of a linear regression model, in which a large number of samples available may be approximated to a straight line and any future prediction made. In Figure 6.1, the pattern shows a linear relationship:

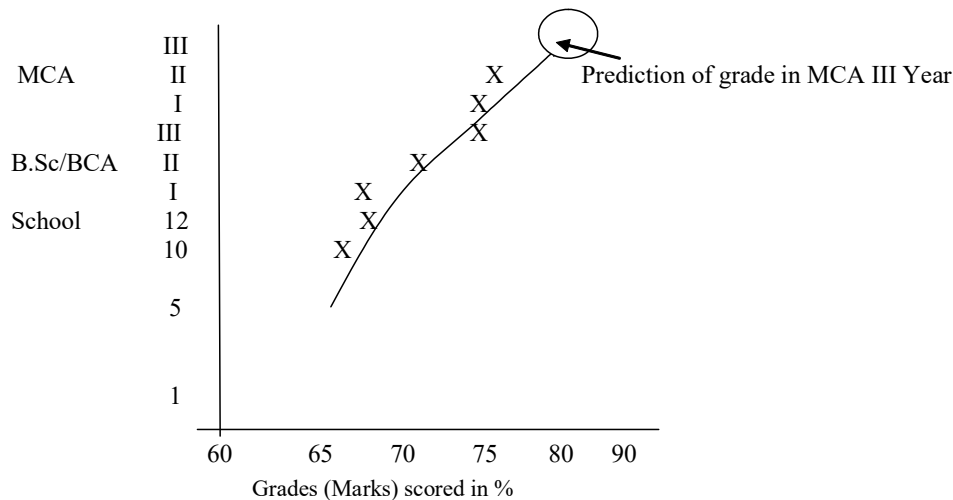


Fig. 6.1 An Example of Prediction

NOTES

NOTES

Issues regarding Classification and Prediction

- (i) **Data Cleaning:** Data cleaning removes or reduces noisy data by applying smoothing techniques. There are many algorithms to handle noisy or missing data. Classification and prediction methods are applied for predicting missing values and filling them with the most commonly occurring values for that attribute.
- (ii) **Relevance Analysis:** This analysis handles redundant data. It exists in the form of correlation analysis and attributes subset selection. Correlation analysis is used to identify whether any two given attributes are statistically related. Attribute subset selection finds a reduced set of attributes, such that the resulting probability distribution of the data classes is nearer to the original distribution. Relevance can be used to detect attributes' similarities and relationship. It helps in finding whether or not they contribute to the classification or prediction task.
- (iii) **Data Transformation and Reduction:** Data is transformed by normalization. Normalization scales all values for a given attribute. The data can also be transformed by generalizing to higher level concepts. These are particularly useful for continuous valued attributes.
- (iv) **Accuracy:** It is the quality of being near to the true value. In the fields of science, engineering, industry and statistics, accuracy is the degree of closeness of a measured or calculated quantity to its actual (true) value. Accuracy can be estimated using one or more test sets that are independent of the training set. The accuracy of a 'classifier' means the prediction of new or previously unseen data. In case of 'predictor', accuracy refers to how well the predictor can guess the value of the predicted attribute for new or previously unseen data.
- (v) **Speed:** It calculates the computational cost of classification and prediction.
- (vi) **Robustness:** It refers to correct prediction on noisy data and data with missing values.
- (vii) **Scalability:** It enhances the efficiency of classification and prediction on large scale of data.

Classification by Decision Tree Induction**Decision Tree Induction**

A decision tree is one in which each branch node represents a choice between a number of alternatives, and each leaf node represents a classification or decision. The topmost node in a tree is the root node. A decision tree is a predictive model, i.e., mapping from observations about an item to reach the conclusion about its target value.

In decision tree induction, we study the decision trees from class-labelled training tuples. The machine learning technique for inducting a decision tree from data is called decision tree learning.

In data mining, trees have three additional descriptive categories/names:

- **Classification Tree Analysis:** When the predicted outcome is the class to which the data belongs.
- **Regression Tree Analysis:** When the predicted outcome can be considered a real number, e.g., the price of a house, or a patient's length of stay in a hospital.
- **Classification and Regression Tree (CART) Analysis:** When both the above procedures are referred.

NOTES

Construction of a Simple Decision Tree

The following data pertains to a fictitious marketing strategy. A company has sent some promotional matters to various houses and few factual data about each house was recorded with their response, whether nil or positive.

District	House Type	Income	Previous	Outcome
			Customer	
Rural	Independent	High	No	Positive
Urban	Joint	Low	Yes	Nil
Suburban	Pent house	High	No	Nil
Urban	Pent house	Low	No	Positive
Rural	Pent house	High	Yes	Positive
Urban	Pent house	High	Yes	Nil
Suburban	Independent	High	No	Nil
Urban	Joint	High	No	Positive
Suburban	Joint	Low	No	Positive
Rural	Independent	Low	No	Positive
Rural	Joint	Low	Yes	Positive
Suburban	Independent	High	Yes	Nil
Suburban	Pent house	Low	Yes	Positive
Urban	Joint	Low	No	Positive
Urban	Pent house	Low	Yes	Nil

Although only 15 have been shown here, we assume it as if there are thousands of such records on this context. When we have multiple instances of such data, it is quite normal and reasonable to expect some patterns in it. We may also discover some kind of underlying relationships between few attributes. It is wise to know about factors that influence someone to respond in the positive. This will also reveal factors that affect the response most strongly to the promotional efforts made.

The data reveals the fact that all rural households responded in the positive. This example illustrates the basic facts just at a glance, but we may form a general idea if we collect more data to build some kind of classifier for examining the

NOTES

underlying relationships. After building the classifier, some kind of future predictions can be made on the type of householders who may respond in the positive. By gathering more data and creating a model to automatically build the decision trees, it is possible to make generalizations and predictions.

A decision tree can be constructed to generate a set of rules. This has a top-down structure originating from a root node, partitioning data into subsets containing instances having similar values. The resultant tree of the data in this example can be shown in Figure 6.2

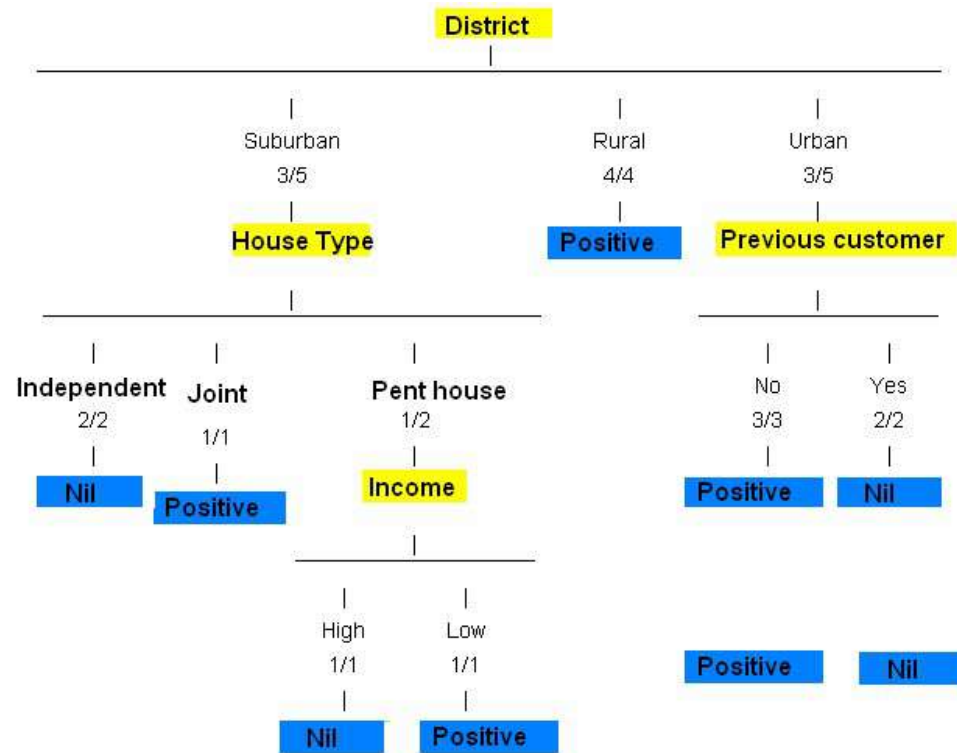


Fig. 6.2 The Resultant Decision Tree

Note: If a software implementation is made for this, then by clicking the dark grey boxes such as ‘Positive’ in leaf nodes of the tree, an option highlighting the matches in the data will be retrieved. Selection of this option will highlight rows covered by rules of the tree ending in this leaf node.

Nodes in light grey show the attributes. At every node, the dataset is split into subsets based on the value of the attribute at the node. Here, at the root node, dataset is divided into three subsets. One has only instances (rows, tuples, whatever) for the value ‘Suburban’ for the ‘District’ attribute, another contains only instances of District attribute as ‘Urban’, and the third in which all the instances are ‘Rural’ for that attribute.

The numbers on the branches refer to the number of instances in each subset that have only one value for the target attribute (‘Outcome’).

We can extract some rules from this simple decision tree very easily by reading the paths of all the leaf nodes. This is given below, going from left to right:

- ```
#
=> (Outcome = Nil)
(District=Suburban) AND (House Type=Pent house)
 AND (Income=Low) => (Outcome = Positive)
• (District=Suburban) AND (House Type=Independent) =>
 (Outcome = Nil) (Outcome = Positive)
• (District=Suburban) AND (House Type=pent house) AND
 (Income=High) => (Outcome = Nil)
• (District=Suburban) AND (House Type=Pent house) AND
 (Income=Low) => (Outcome = Positive)
• (District=Urban) AND (Previous Customer=No) => (Outcome
= Positive)
• (District=Urban) AND (Previous Customer=Yes) =>
 (Outcome = Nil)
• (District=Rural) => (Outcome = Positive)
```

## NOTES

This model may be used for predicting some of the outcomes of an effort. We may like to predict the outcomes by sending a promotional mail to a certain house. This illustrates the basic idea on Decision Tree Learning. The sample is very small but if we had thousands of such records of data for a particular concept with more additional attributes, it will not be possible to analyse by just looking at it. Construction of a decision tree helps in such cases.

Many decision trees are possible with the same dataset. We may even change the root node and instead of 'District', 'Income' might be selected as the root node.

### Decision Tree-Based Algorithms

- (i) Construct a tree to model classification.
- (ii) Apply the tree to the database.

### Advantages of Decision Trees

- (i) They are easy to use.
- (ii) The generated rules are easy to understand.
- (iii) They are amenable to scaling and database size.

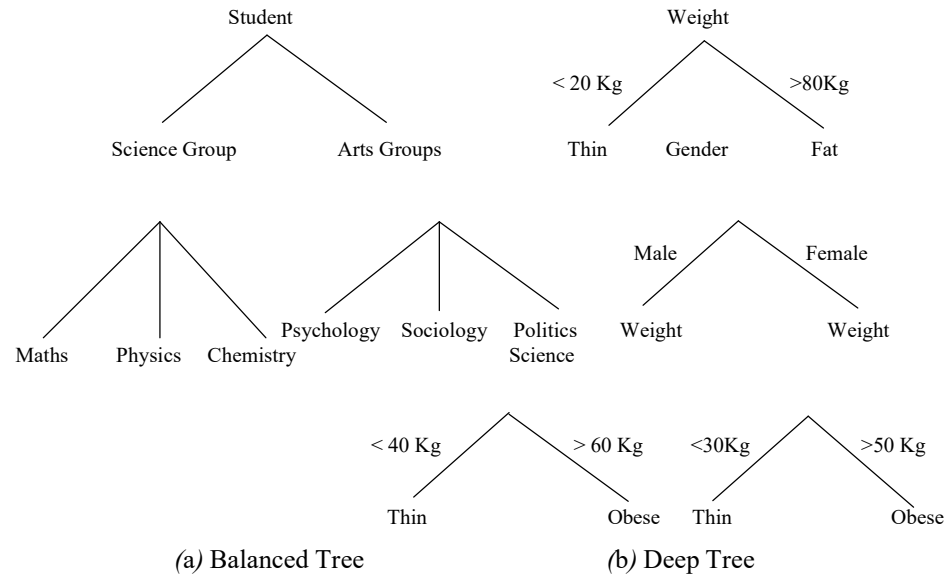
### Disadvantages of Decision Trees

- (i) They cannot handle continuous data.
- (ii) They are incapable of handling many problems which cannot be divided into attribute domains.
- (iii) They can lead to over-fitting as trees are constructed from training data.

## Decision Tree Algorithms and Their Main Issues

- (i) **Tree Structure:** Select a tree structure such as a balanced tree for improving performance. Examples of trees are balanced tree and deep tree.

### NOTES



**Fig.6.3** Structure of Trees

- (ii) **Training Data:** Structure of a tree depends on training data. Selecting adequate data prevents either the tree to over-fit or makes it good enough to work on a general data.
- (iii) **Stopping Criteria:** Construction of a tree stops on a stopping criterion. It is essential to achieve a balance between too early or late to create a tree with the right level.
- (iv) **Pruning:** After constructing a tree, modify it to remove duplication or subtrees.
- (v) **Splitting:** Selection of the best splitting attribute and size of the training set are important factors in creating a decision tree algorithm. For example, splitting attributes in the case of students may be gender, marks scored and electives chosen. The order in which splitting attributes are chosen is important for avoiding redundancy and unnecessary comparisons at different levels.

## Decision Tree Algorithms

J. Ross Quinlan, a researcher in machine learning developed a decision tree algorithm known as ID3 (Iterative Dichotomizer). In 1984, a group of statisticians described a generation of binary decision trees known as Classification and Regression Trees (CART).

ID3 and CART follow non-backtracking approach in which decision trees are constructed in a top-down recursive 'divide and conquer' manner. Most of

the algorithms in decision tree induction follow the top-down approach. This approach starts with a training set of tuples and their associated class labels. The training set is recursively partitioned into smaller subsets as the tree is being built. The following is a basic tree algorithm:

- (i) create a node N;
  - (ii) if tuples in D are all of the same class C then
  - (iii) return N as a leaf node labeled with the class C;
  - (iv) if attribute\_list is empty then
  - (v) return N as a leaf node labeled with the majority class in D;
  - (vi) apply Attribute\_selection\_method(D, attribute\_list) to find the ‘best’ splitting\_criterion; label node N with with splitting\_criterion;
  - (vii) if splitting\_attribute is discrete-valued and multiway splits are allowed then //not restricted to binary trees attribute\_list (arrow mark) attribute\_list - splitting\_attribute;
  - (viii) for each outcome j of splitting\_criterion
  - (ix) let (symbol) be the set of data tuples in D satisfying outcome j; // partition
  - (x) if (symbol) is empty then attach a leaf labeled with the majority class in D to node N;
  - (xi) else attach the node returned by Generate\_decision\_tree(symbol, attribute\_list) to node N;
- endfor  
return N;

**Explanation:** The above algorithm has three parameters—D, attribute\_list and attribute\_selection\_method. D is the data partition. It is a set of training tuples and their associated class labels. Attribute\_list contains a list of attributes describing the tuples.

Now, the tree starts as a single node N. It represents the training tuples in D. If the tuples in D are all of the same class, then node N is considered a leaf. It is labelled with that class. It occurs in Step 2 and Step 3. Steps 4 and 5 are terminating conditions. If this condition does not follow then algorithm calls Attribute\_selection\_method to determine the splitting criterion. This criterion determines the best way to partition the tuples in D into individual classes (Step 6). Step 7 serves as a test at the node. In Step 10 and Step 11, tuples in D are partitioned.

### Tree Pruning

When a decision tree is constructed, many types of unusual data are reflected in the training data as noise and outliers. Tree pruning handles the problem of overfitting the data using statistical measures to remove the least reliable branches. Figure 6.4 shows an unpruned tree and a pruned tree.

### NOTES

## NOTES

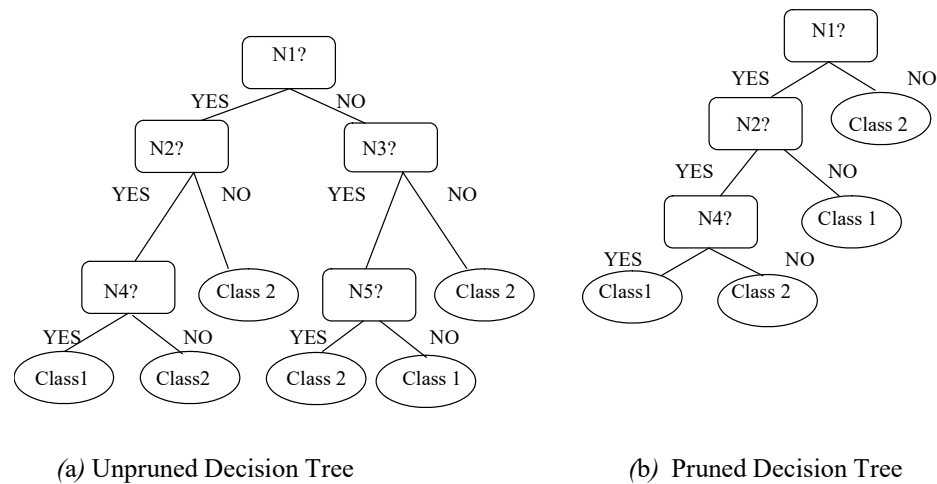


Fig. 6.4 Tree Pruning

There are two approaches to tree pruning.

- Pre-pruning
- Post-pruning
- **Pre-pruning:** In this approach, a tree is ‘pruned’ by stopping its construction much earlier. On halting, that node becomes a leaf. As partitioning has occurred at the pre-specified threshold, further partitioning of the given subset is halted.
- **Post-pruning:** This approach removes subtrees from a fully grown tree. This is done by removing its branches and replacing it with a leaf. The cost complexity pruning algorithm is used in CART. It adopts a post-pruning approach. In this approach, we calculate the cost of each internal node  $N$  and then compute the cost complexity of the subtree at  $N$ . Then the two values are compared. If it gives a smaller cost complexity, then the subtree is pruned.

We can also prune a tree with the help of a number of bits required to encode them. This method can be followed through the Minimum Description Length (MDL) principle. Post-pruning requires more computation than pre-pruning. Post-pruning gives a better decision tree. Pruned trees tend to be more compact than unpruned ones. One of the drawbacks of decision trees is the problem of repetition and replication. In replication, duplicate trees do not give an accurate value. This problem can be prevented through multi-variant splits.

### Scalability and Decision Tree Induction

Scalability increases the efficiency of the existing system. In this subsection, we will discuss how scalable is decision tree induction, i.e., the efficiency of the existing decision tree algorithm (ID3, C4.5 and CART). This efficiency becomes an issue of concern when these algorithms are applied to mining of very large real-world databases.

In data mining, applications and amount of data increase. As data increases very large training sets of millions of tuples become common. Most often, this training data does not fit into the memory. At this condition, decision tree construction becomes inefficient due to swapping of the training tuples in and out of the main and cache memories. Now a scalable approach is required to handle training data that are too large to fit into the memory. One of the earlier strategies to handle this approach is to save space. It includes discretizing continuous valued attributes and sampling data at each node.

**NOTES**

Recent decision tree algorithms that address the scalability issue include Supervised Learning in Quest (SLIQ) and Scalable Parallelizable Induction of Decision Tree (SPRINT). Both handle categorical and continuous valued attributes. Both algorithms propose presenting techniques on disk-resident datasets that are too large to fit into the memory. Both define the new data structures for decision tree construction.

**Supervised Learning in Quest (SLIQ)**

It employs disk-resident attribute lists and a single memory-resident class list. We may draw the attribute lists and class lists from the tuple data of Table 3.1. Each attribute has an associate attribute list indexed by the Record Identifier (RID). Each tuple is represented by a linkage of one entry from each attribute list to an entry in the class list. The class list exists in the memory because it is often accessed and modified in the building and pruning phases. The size of the class list grows proportionally with the number of tuples in the training set. If a class list cannot fit into the memory, the performance of SLIQ decreases.

*Table 6.1 Tuple Data*

| RID | Credit Rating | Age | Buys Computer |
|-----|---------------|-----|---------------|
| 1   | Good          | 28  | No            |
| 2   | Good          | 30  | Yes           |
| 3   | Excellent     | 32  | Yes           |
| 4   | Good          | 36  | No            |

**Disk-Resident Attribute List**

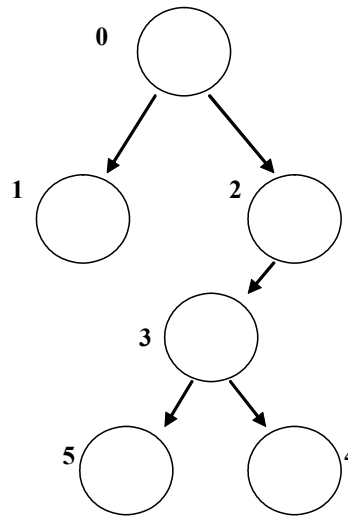
| Credit Rating | RID |
|---------------|-----|
| Good          | 1   |
| Good          | 2   |
| Good          | 4   |
| Excellent     | 3   |

**Memory-Resident Class List**

| Age | RID | RID | Node | Buys |
|-----|-----|-----|------|------|
| 28  | 1   | 1   | 5    | No   |
| 30  | 2   | 2   | 2    | Yes  |
| 32  | 3   | 3   | 3    | Yes  |
| 36  | 4   | 4   | 6    | No   |



**NOTES**



*Fig. 6.5 Attribute List and Class List Data Structure*

**Scalable Parallelizable Induction of Decision Tree (SPRINT)**

It holds the class and RID information as shown in Table 6.2.

*Table 6.2 Attribute List Data Structure used in SPRINT for the Tuple Data of Table 6.1*

| Credit Rating | RID | Buys_Computer | Age | RID | Buys_Computer |
|---------------|-----|---------------|-----|-----|---------------|
| Excellent     | 3   | Yes           | 28  | 1   | No            |
| Good          | 1   | No            | 30  | 2   | Yes           |
| Good          | 2   | Yes           | 32  | 3   | Yes           |
| Good          | 4   | No            | 36  | 4   | No            |

In SPRINT, when a node is split, the attribute lists are partitioned and distributed among the child nodes. When a list is partitioned, the order of the record in the list is maintained. Thus, SPRINT is designed for the enhancement of scalability and also for easy parallelizing. SLIQ and SPRINT both handle large data sets, i.e., disk-resident data sets. The scalability of SLIQ is limited by the use of memory-resident data structures, while SPRINT removes all memory restrictions yet requires the use of a hash tree, proportional in size to the training set, enhancing the cost as the training set size grows.

**Check Your Progress**

1. What is a prediction model?
2. What is relevance analysis?

## 6.3 BAYESIAN CLASSIFICATION

### Bayes' Theorem

Reverend Thomas Bayes stated Bayes' Theorem which is a theory of probability. It is way of understanding how the probability is affected by a new piece of evidence. Bayesian theorem are used in a wide variety of contexts, ranging from marine biology to the development of 'Bayesian' spam blockers for e-mail systems. It has been used in the philosophy of science in order to clarify the relationship between theory and evidence. Bayes theorem can be used in many insights in the philosophy of science involving confirmation, falsification, the relation between science and pseudoscience and other topics can be made more precise, and sometimes extended or corrected. The terminology involved in this theorem is given below:

$$P(T | E) = \frac{P(E | T) \times P(T)}{P(E | T) \times P(T) + P(E | -T) \times P(-T)}$$

A theory or hypothesis that we are interested in testing is represented by T in this formula and E represents a new piece of evidence that seems to confirm or disconfirm the theory. For any proposition S, we used P(S) to stand for our degree of belief, or 'subjective probability,' that S is true. In particular, P (T) represents our best estimate of the probability of the theory we are considering, prior to consideration of the new piece of evidence. It is known as the *prior probability* of T.

What we want to discover is the probability that T is true supposing that our new piece of evidence is true. This is a *conditional probability*, the probability that one proposition is true provided that another proposition is true. For instance, suppose you draw a card from a deck of 52, without showing it to me. Assuming the deck has been well shuffled, I should believe that the probability that the card is a jack, P (J), is 4/52, or 1/13, since there are four jacks in the deck. But now suppose you tell me that the card is a face card. The probability that the card is a jack, given that it is a face card, is 4/12, or 1/3, since there are 12 face cards in the deck. We represent this conditional probability as P (J|F), meaning the probability that the card is a jack *given that* it is a face card.

We don't need to take conditional probability as a primitive notion; we can define it in terms of absolute probabilities:  $P (A|B) = P (A \text{ and } B) / P(B)$ , that is, the probability that A and B are both true divided by the probability that B is true.

Using this idea of conditional probability to express what we want to use Bayes' Theorem to discover, we say that P (T|E), the probability that T is true given that E is true, is the *posterior probability* of T. The idea is that P (T|E) represents the probability assigned to T *after* taking into account the new piece of evidence, E. To calculate this we need, in addition to the prior probability P (T),

### NOTES

## NOTES

two further conditional probabilities indicating how probable our piece of evidence is depending on whether our theory is or is not true. We can represent these as  $P(E|T)$  and  $P(E|\sim T)$ , where  $\sim T$  is the *negation* of  $T$ , i.e. the proposition that  $T$  is false.

### Naïve Bayesian Classification

Naïve Bayesian classification is based on the Bayesian theorem. It is mainly suited when the dimensionality of the inputs is high. Parameter estimation for naive Bayes models uses the method of maximum likelihood. In spite over-simplified assumptions, it often performs better in many complex real-world situations. Advantage of Naïve Bayesian classification is that it requires a small amount of training data to estimate the parameters.

### Uses of Naive Bayes Classification

1. The Bayesian classification is used as a probabilistic learning method. Naive Bayes classifiers are among the most successful known algorithms for learning to classify text documents.
2. Spam filtering is the best known use of Naive Bayesian text classification. It makes use of a naive Bayes classifier to identify spam e-mail. Bayesian spam filtering has become a popular mechanism to distinguish illegitimate spam email from legitimate email sometimes called “ham” or “bacn”). Many modern mail clients implement Bayesian spam filtering. Users can also install separate email filtering programs. Server-side email filters, such as DSPAM, SpamAssassin, SpamBayes, Bogofilter and ASSP, make use of Bayesian spam filtering techniques, and the functionality is sometimes embedded within mail server software itself.

Hybrid Recommender System uses Naive Bayes Classifier and Collaborative Filtering. Recommender Systems apply machine learning and data mining techniques for filtering unseen information and can predict whether a user would like a given resource. It is proposed a unique switching hybrid recommendation approach by combining a Naive Bayes classification approach with the collaborative filtering. Experimental results on two different data sets, show that the proposed algorithm is scalable and provide better performance—in terms of accuracy and coverage—than other algorithms while at the same time eliminates some recorded problems with the recommender systems.

### Check Your Progress

3. What is Bayes’ theorem?
4. What is Naïve Bayesian classification?

---

## 6.4 CLASSIFICATION BY BACK PROPAGATION

---

Back propagation is defined as a neural network learning algorithm. A neural network is defined as a set of connected input/output units where weight is associated with each connection. Neural network learning can also be called as connectionist learning because there are connections between input/output units. Long training time is involved into the neural network. Various parameters are required by the neural network like network topology.

There are various kinds of neural networks and the neural network algorithms. The most popular neural network algorithm is the back propagation, which is grown repute in 1980s. The main reason of criticism of neural network is poor interpretability because human beings are not able to interpret the symbolic meaning of weight of connections. Neural networks have high tolerance to the noisy data. The patterns on which the neural networks are not trained can be classified by it. The classification by back propagation includes the following:

- Multilayer feed-forward neural network.
- Network topology
- Back propagation algorithm

### Multilayer Feed-Forward Neural Network

Multilayer feed-forward neural network is performed by the back propagation algorithm. Input is measured for each training set corresponding to the attributes. Each and every layer is designed by the units. Input layer is made by feeding input into the layer simultaneously. The units in the input layer are called input units. The weighted output from these units is now feed to the second layer which is called hidden layer.

The output of the hidden layer is given as input to another layer, and so on. This will emit the network prediction for the given sample. The units in output layer and hidden layers are called as neuroses because of symbolic biological basis. Multilayer feed-forward neural network can approximate any function for linear threshold function.

### Network Topology

The network topology must be decided by the user before the beginning of the training and also specify: Number of units in input layer, Number of units in hidden layer and Number of unit in output layer.

- Input value normalization for each attribute will help in speeding up the leaning phase. Input values are normalized in such a manner so that its value falls between 0.0. to 1.0. Discrete-valued attributes are encoded in such a manner that there is only one input unit per domain value.

## NOTES

## NOTES

For example, three input units are used for representing an attribute  $X$  if the domain of that attribute is  $(x_0, x_1, \text{ and } x_2)$ . In the beginning, each unit is initialized to zero. If  $X=x_0$  then it is set to 0. If it is  $X=x_1$  then it is set to 1, and so on.

- For the hidden layer unit, there is no clear rule particularly. Network design is a trial and error process which affects the accuracy of the resulting trained network. The resulting accuracy is affected by initial value of weights. Once the network has been taught and its accuracy is not measured for acceptable, then there is a common need to repeat the training process with a different set of initial weights or a different network topology. The Cross-validation techniques for accuracy estimation can be used to help decide when an acceptable network has been found.
- Numbers of the automated techniques have been planned that search for 'good' network structure. And these naturally use the hill-climbing approach that starts with the initial structure that is selectively modified. Two classes are used for representing one output where value 1 represent one class and value 0 represent another class. One output unit is used per class if there are more than two classes.

### Back Propagation Algorithm

Back propagation learns by processing the set of training samples iteratively and the network prediction is compared with the actual known class label. For minimizing the mean squared error between the actual class and network prediction, the weight of each training sample has to be modified. The modification is done in the backward direction it means from output layer to each hidden layer and hence it is known as back propagation. Small random numbers are used for initializing the weights which is in the range of -1.0 to 1.0 or -0.5 to 0.5. There is a bias associated with each unit. As in weights, small random numbers are used for initializing bias. Any training sample  $X$  is processed in following steps:

1. Propagate the input forward
2. Back propagate the errors
3. Terminating condition.

### Propagate the Input Forward

In this step, the net input and output of output layers and hidden layers are calculated. First of all the training sample is fed into the input layer. The output is equal to input layer for the unit into the input layers and it is computed as a combination of input and output unit in previous layer.

For computing the net input for the unit, each input has to be multiplied by its corresponding weight. The net input to unit for given a unit in output or hidden layer is:

$$i_j = \sum w_{ij} O_i + \theta_j$$

Where,

$W_{ij}$  = Weight of connection from unit  $i$  to unit  $j$ .

$O_i$  = Output of unit  $i$  from previous layer.

$\theta_j$  = Bias of the unit  $j$ .

The bias serves for varying the activity of unit by acting as threshold. Every unit in output and hidden layer takes its net input and then activation function is applied to the net input. The activation of the neuron is symbolized by the neuron which is represented by the unit.

The sigmoid or logistic function is used which is also called as squashing function because a large input domain is mapped in a small range from 0 to 1. The squashing function is differentiable and non-linear which allow back propagation algorithm to model classification problem.

### Back Propagate the Error

The error is propagated in backward direction by updating the weights. Error is also biased by reflecting the error of network prediction. For output layer unit  $j$ , error  $Err_j$  is:

$$Err_j = O_j (1 - O_j) (T_j - O_j)$$

For computing the error of hidden layer unit  $j$ , the weighted sum of the errors in next layer is considered for the units connected to unit  $j$ . The error of a hidden layer unit  $j$  is:

$$Err_i = O_j (1 - O_j) \sum_k Err_k W_{jk}$$

Where:

$W_{jk}$  = Weight of the connection from unit  $j$  to unit  $k$  in the next higher layer.

$Err_k$  = Error of unit  $k$ .

The biases and weights are updated for reflecting propagated the errors. Following equation is used for updating the weight,

$$\Delta W_{ij} = (L) Err_j O_i$$

$$W_{ij} = W_{ij} + \Delta W_{ij}$$

Here,

$\Delta W_{ij}$  is the change in weight  $W_{ij}$ .

$L$  = learning rate and its value lies between 0.0 to 1.0

A method of gradient decent for searching set of weight is used for learning in back propagation. This method helps in modelling given classification problem which tends to minimize mean squared distance between class label and network class prediction.

### NOTES

**NOTES**

The learning rate helps in avoiding to find local minimum in decision space and helps in encouraging global optimum. Learning will occur at slow speed if learning rate is too small and the oscillation between inadequate solutions may occur if learning rate is too large. For setting the learning rate to  $1/f$ , a thumb rule has to be set. For updating the biases, following equation has to be used:

$$\Delta\theta_j = (L) \text{Err } j$$

$$\theta_j = \theta_j + \Delta\theta_j$$

Where,

$\Delta\theta_j$  = change in Bias  $\theta_j$

Err  $j$  = error in unit  $j$

After the presentation of each sample, the biases and weight needs to be updated. This is called as case updating. The increments of bias and weight are accumulated in variables so that both are updated after the presentation of all samples present in the training set. This process is called as epoch updating.

In epoch updating, epoch is iteration through training set. As more accurate results are yielded by the case updating hence it is more commonly used than the epoch updating.

**Termination Condition**

The training will get stopped in the following condition:

- For previous epoch, all the  $\Delta W_{ij}$  are too small so that they are below the given threshold.
- For previous epoch, the percentage of misclassified samples is below some specific threshold.
- A pre-specified number of epochs have been expired.

**Check Your Progress**

5. Define a neural network.
6. What is the main reason of criticism of neural network?

**6.5 ANSWERS TO CHECK YOUR PROGRESS QUESTIONS**

1. Prediction may be considered as a type of classification where an attribute is bracketed in a class itself. Thus, if we focus on any one of the treatment categories, then it is categorized as a 'prediction model'.
2. Relevance analysis handles redundant data. It exists in the form of correlation analysis and attributes subset selection. Correlation analysis is used to identify whether any two given attributes are statistically related.

3. Bayes' Theorem is a theory of probability. It is way of understanding how the probability is affected by a new piece of evidence. Bayesian theorem are used in a wide variety of contexts, ranging from marine biology Classification to the development of 'Bayesian' spam blockers for e-mail systems.
4. Naïve Bayesian classification is based on the Bayesian theorem. It is mainly suited when the dimensionality of the inputs is high. Parameter estimation for naive Bayes' models uses the method of maximum likelihood. In spite over-simplified assumptions, it often performs better in many complex real-world situations.
5. A neural network is defined as a set of connected input/output units where weight is associated with each connection.
6. The main reason of criticism of neural network is poor interpretability because human beings are not able to interpret the symbolic meaning of weight of connections.

## NOTES

---

## 6.6 SUMMARY

---

- Classification and prediction are used for data analysis. They are used to define models describing important data classes or to predict future data trends.
- Classification and prediction are two different approaches of data analysis that are used to identify models to categorize data classes or to predict future directions of data trends.
- As prediction deals with continuous values, one example may be that of a linear regression model, in which a large number of samples available may be approximated to a straight line and any future prediction made.
- Issues regarding Classification and Prediction are:
  - (i) Data Cleaning: Data cleaning removes or reduces noisy data by applying smoothing techniques.
  - (ii) Relevance Analysis: This analysis handles redundant data. It exists in the form of correlation analysis and attributes subset selection.
  - (iii) Data Transformation and Reduction: Data is transformed by normalization
  - (iv) Accuracy: It is the quality of being near to the true value.
  - (v) Speed: It calculates the computational cost of classification and prediction.
  - (vi) Robustness: It refers to correct prediction on noisy data and data with missing values.



## NOTES

- (vii) Scalability: It enhances the efficiency of classification and prediction on large scale of data.
- A decision tree is one in which each branch node represents a choice between a number of alternatives, and each leaf node represents a classification or decision. The topmost node in a tree is the root node.
- Decision tree-based algorithms:
  - (i) Construct a tree to model classification.
  - (ii) Apply the tree to the database.
- Advantages of decision trees are:
  - (i) They are easy to use.
  - (ii) The generated rules are easy to understand.
  - (iii) They are amenable to scaling and database size.
- Disadvantages of decision trees are:
  - (i) They cannot handle continuous data.
  - (ii) They are incapable of handling many problems which cannot be divided into attribute domains.
  - (iii) They can lead to over-fitting as trees are constructed from training data.
- J. Ross Quinlan, a researcher in machine learning developed a decision tree algorithm known as ID3 (Iterative Dichotomizer). In 1984, a group of statisticians described a generation of binary decision trees known as Classification and Regression Trees (CART).
- ID3 and CART follow non-backtracking approach in which decision trees are constructed in a top- down recursive ‘divide and conquer’ manner.
- When a decision tree is constructed, many types of unusual data are reflected in the training data as noise and outliers. Tree pruning handles the problem of over fitting the data using statistical measures to remove the least reliable branches.
- There are two approaches to tree pruning.
  - (i) Pre-pruning
  - (ii) Post-pruning
- In SPRINT, when a node is split, the attribute lists are partitioned and distributed among the child nodes. When a list is partitioned, the order of the record in the list is maintained.
- Reverend Thomas Bayes stated Bayes’ Theorem which is a theory of probability. It is way of understanding how the probability is affected by a new piece of evidence.

- Naïve Bayesian classification is based on the Bayesian theorem. It is mainly suited when the dimensionality of the inputs is high. Parameter estimation for naive Bayes models uses the method of maximum likelihood. In spite of over-simplified assumptions, it often performs better in many complex real-world situations.
- The Bayesian classification is used as a probabilistic learning method. Naive Bayes classifiers are among the most successful known algorithms for learning to classify text documents.
- Back propagation is defined as a neural network learning algorithm. A neural network is defined as a set of connected input/output units where weight is associated with each connection.
- Multilayer feed-forward neural network is performed by the back propagation algorithm. Input is measured for each training set corresponding to the attributes.
- The network topology must be decided by the user before the beginning of the training and also specify: Number of units in input layer, Number of units in hidden layer and Number of unit in output layer.
- Back propagation learns by processing the set of training samples iteratively and the network prediction is compared with the actual known class label. For minimizing the mean squared error between the actual class and network prediction, the weight of each training sample has to be modified.

## NOTES

---

### 6.7 KEY WORDS

---

- **Accuracy:** The degree of closeness of a measured or calculated quantity to its actual (true) value.
- **Robustness:** Correct prediction on noisy data and data with missing values.
- **Decision Tree:** A predictive model in which each branch node represents a choice between a number of alternatives, and each leaf node represents a classification or decision.
- **Decision Tree Learning:** The machine learning technique for inducting a decision tree from data.
- **Pruning:** Modifying a tree to remove duplication or subtrees.

---

### 6.8 SELF ASSESSMENT QUESTIONS AND EXERCISES

---

#### Short Answer Questions

1. Define back propagation.
2. Give the terminology involved in Bayes' theorem.

3. What are the two approaches to tree pruning?
4. List the advantages of decision trees.

## NOTES

### Long Answer Questions

1. What is the difference between classification and prediction? What are the issues regarding classification and prediction?
2. Explain the construction of a simple decision tree.
3. Explain Supervised Learning in Quest (SLIQ) with the help of diagrams.
4. What is Naïve Bayesian Classification? Discuss its uses.
5. Explain the classifications by back propagation.

---

## 6.9 FURTHER READINGS

---

Han, Jiawei, Micheline Kamber and Jian Pei. 2011. *Data Mining: Concepts and Techniques*, 2nd edition. Amsterdam: Elsevier.

Pujari, Arun K. 2010. *Data Mining Techniques*, 2nd edition. United States: Universities Press.

Anahory, Sam and Dennis Murray. 1997. *Data Warehousing in the Real World: A Practical Guide for Building Decision Support Systems*, 1st edition. Boston: Addison Wesley.

Witten, I. H. and Eibe Frank. 2005. *Data Mining: Practical Machine Learning Tools and Techniques*, 2nd edition. Amsterdam: Elsevier.

Soman, K. P., Shyam Diwakar and V. Ajay. 2006. *Insight Into Data Mining: Theory and Practice*. New Delhi: PHI.

---

**BLOCK - III**  
**CLUSTERING TECHNIQUES AND**  
**MACHINE LEARNING**

---

*Introduction to  
Clustering*

**NOTES**

---

**UNIT 7 INTRODUCTION TO  
CLUSTERING**

---

**Structure**

- 7.0 Introduction
- 7.1 Objectives
- 7.2 Clustering: Definition
- 7.3 Clustering Methods and Algorithms
  - 7.3.1 Partitioning Methods
  - 7.3.2 Hierarchical Methods
  - 7.3.3 Density-Based Methods
  - 7.3.4 Grid-Based Methods
  - 7.3.5 K- means
  - 7.3.6 K- Medoid
  - 7.3.7 CLARA (Clustering for Large Applications)
  - 7.3.8 CLARANS (Clustering Large Application Based on Randomized Search)
  - 7.3.9 BIRCH (Balanced Iterative Reducing and Clustering using Hierarchies)
  - 7.3.10 CACTUS – Clustering Categorical Data Using Summaries
  - 7.3.11 ROCK - Robust Clustering Algorithm for Categorical Attributes
  - 7.3.12 DBSCAN (Density Based SCAN Clustering Algorithm)
  - 7.3.13 STIRR (Sieving Through Iterated Relational Reinforcement)
- 7.4 Answers to Check Your Progress Questions
- 7.5 Summary
- 7.6 Key Words
- 7.7 Self Assessment Questions and Exercises
- 7.8 Further Readings

---

**7.0 INTRODUCTION**

---

Clustering may be considered one of the most important *unsupervised learning* problem; so, just like every other problem of this kind, it involves finding a *structure* in a collection of unlabeled data.

A loose definition of clustering could be “the process of organizing objects into groups whose members are similar in some way”. A *cluster* is thus, a collection of objects which are “similar” between them and are “dissimilar” to the objects of the other clusters.

The objective of clustering is to understand and define the intrinsic grouping in a set of unlabeled data. But how do we decide what constitutes a good clustering?

## NOTES

There is no absolute “best” criterion which would be independent of the final aim of the clustering. Thus, it is the user who must supply the criterion, in a way that the result of the clustering suits their needs.

For instance, the user might be interested in finding representatives for homogeneous groups (*data reduction*), in finding “natural clusters” and describing their unknown properties (“*natural*” *data types*), in finding useful and suitable groupings (“*useful*” *data classes*) or in finding unusual data objects (*outlier detection*).

---

### 7.1 OBJECTIVES

---

After going through this unit, you will be able to:

- Define clustering
- Understand the requirements of clustering in data mining
- Explain the various methods and algorithms of clustering

---

### 7.2 CLUSTERING: DEFINITION

---

Clustering is the process of making group of abstract objects into classes of similar objects. A cluster of data objects can be treated as a one group. While doing the cluster analysis, we first partition the set of data into groups based on data similarity and then assign the label to the groups. In other words the similar object are grouped in one cluster and dissimilar are grouped in other cluster.

#### Major Requirements of Clustering in Data Mining

- **Scalability:** We need highly scalable clustering algorithms to deal with large databases.
- **Ability to deal with different kind of attributes:** Algorithms should be capable to be applied on any kind of data such as interval based (numerical) data, categorical, binary data.
- **Discovery of clusters with arbitrary shape:** The clustering algorithm should be capable of detect cluster of arbitrary shape. They should not be bounded to only distance measures that tend to find spherical cluster of small size.
- **High dimensionality:** The clustering algorithm should not only be able to handle low- dimensional data but also the high dimensional space.
- **Ability to deal with noisy data:** Databases contain noisy, missing or erroneous data. Some algorithms are sensitive to such data and may lead to poor quality clusters.
- **Interpretability:** The clustering results should be interpretable, comprehensible and usable.

- **Constraint-based clustering:** A challenging task is to find data groups with good clustering behavior that satisfy specified constraints.
- **Requirements for domain knowledge to determine input parameters:** Many clustering algorithms require users to provide domain knowledge in the form of input parameters such as the desired number of clusters. Consequently, the clustering results may be sensitive to such parameters.
- **Incremental clustering and insensitivity to input order:** Clustering algorithms may return dramatically different clustering depending on the order in which the objects are presented. Incremental clustering algorithms and algorithms that are insensitive to the input order are needed.

Clustering is useful in understanding the hidden structure in data, automatically organizing data and preprocessing for further analysis. Clustering Analysis is broadly used in many applications such as pattern recognition, data analysis, and image processing. Various applications of cluster analysis are listed below:

- **Biology:** it can be used to derive plant and animal taxonomies, categorize genes with similar functionality and gain insight into structures inherent in populations.
- **Web:** Clustering also helps in classifying documents on the web for information discovery.
- **Marketing:** Customer segmentation based on a database of customer data containing their properties and past buying records.
- **Education:** Improves the student's performance and enhances the academic planners to monitor the performance and progression level of each student.
- **Social Science:** Helps to make predictions by identifying the number of active suspects based on the graphical data.
- **Outlier Detection:** Clustering is also used in detection of credit card fraud.

#### Check Your Progress

1. Define clustering.
2. Give few applications of cluster analysis.

### 7.3 CLUSTERING METHODS AND ALGORITHMS

The clustering methods can be classified into following categories:

#### 7.3.1 Partitioning Methods

Suppose we are given a database of  $n$  objects, the partitioning method constructs  $k$  partition of data. Each partition will represent a cluster and  $k \leq n$ . It means that it will classify the data into  $k$  groups, where each group contains at least one object

#### NOTES

## NOTES

and each object must belong to exactly one group. In this type of clustering methods for a given number of partitions (say  $k$ ), the partitioning method will create an initial partitioning and then it uses the iterative relocation technique to improve the partitioning by moving objects from one group to other.

### 7.3.2 Hierarchical Methods

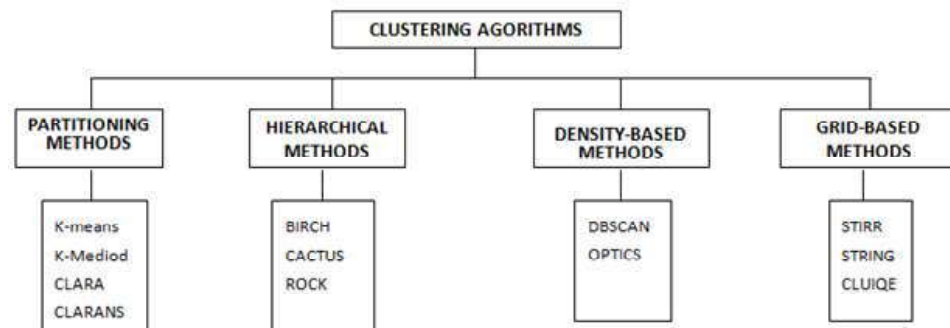
There are two approaches to perform Hierarchical clustering techniques Agglomerative (top-bottom) and Divisive (bottom-top). In Agglomerative approach, initially one object is selected and successively merges the neighbor objects based on the distance as minimum, maximum and average. The process is continuous until a desired cluster is formed. The Divisive approach deals with set of objects as single cluster and divides the cluster into further clusters until desired number of clusters is formed.

### 7.3.3 Density-Based Methods

Data objects are categorized into core points, order points and noise points. All the core points are connected together based on the densities to form cluster. Arbitrary shaped clusters are formed by various clustering algorithms. These methods are based on the notion of density. The basic idea is to continue growing the given cluster as long as the density in the neighborhood exceeds some threshold i.e. for each data point within a given cluster; the radius of a given cluster has to contain at least a minimum number of points.

### 7.3.4 Grid-Based Methods

Here, the objects together from a grid. The object space is quantized into finite number of cells that form a grid structure. To form clusters Grid algorithm uses subspace and hierarchical clustering techniques.



*Fig. 7.1 An Overview of Clustering Methods*

**Table 7.1** Overview of Clustering Methods

| Methods               | General characteristics                                                                                                                                                                                                                                                                                                               |
|-----------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Partitioning Methods  | <ul style="list-style-type: none"> <li>– Find mutually exclusive clusters of spherical shape</li> <li>– Distance-based</li> <li>– May use mean or medoid (etc.) to represent cluster center</li> <li>– Effective for small- to medium-size data sets</li> </ul>                                                                       |
| Hierarchical Methods  | <ul style="list-style-type: none"> <li>– Clustering is a hierarchical decomposition (i.e., multiple levels)</li> <li>– Cannot correct erroneous merges or splits</li> <li>– May incorporate other techniques like microclustering or consider object “linkages”</li> </ul>                                                            |
| Density-based Methods | <ul style="list-style-type: none"> <li>– Can find arbitrarily shaped clusters</li> <li>– Clusters are dense regions of objects in space that are separated by low-density regions</li> <li>– Cluster density: Each point must have a minimum number of points within its “neighborhood”</li> <li>– May filter out outliers</li> </ul> |
| Grid-based Methods    | <ul style="list-style-type: none"> <li>– Use a multi resolution grid data structure</li> <li>– Fast processing time (typically independent of the number of data objects, yet dependent on grid size)</li> </ul>                                                                                                                      |

**NOTES**

**Table 7.2** Advantages and Disadvantages of various types of clustering Algorithms

| Clustering type | Advantages                                                                                                                                                                                                                                    | Disadvantages                                                                                                                                                                                                                                                                                                                      |
|-----------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Partitioning    | <ul style="list-style-type: none"> <li>• Relatively scalable and simple.</li> <li>• Suitable for datasets with compact spherical clusters that are well-separated.</li> </ul>                                                                 | <ul style="list-style-type: none"> <li>• Degradation in high dimensional spaces.</li> <li>• Poor cluster descriptors</li> <li>• High sensitivity to initialization phase, noise and outliers</li> </ul>                                                                                                                            |
| Hierarchical    | <ul style="list-style-type: none"> <li>• Embedded flexibility regarding the level of granularity.</li> <li>• Well suited for problems involving point linkages, e.g. taxonomy trees.</li> <li>• Application to any attribute types</li> </ul> | <ul style="list-style-type: none"> <li>• Inability to make corrections once the splitting/merging decision is made.</li> <li>• Lack of interpretability regarding the cluster descriptors.</li> <li>• Vagueness of termination criterion.</li> <li>• Prohibitively expensive for high dimensional and massive datasets.</li> </ul> |
| Density Based   | <ul style="list-style-type: none"> <li>• Discovery of arbitrary shaped clusters with varying size</li> <li>• Resistance to noise and outliers</li> </ul>                                                                                      | <ul style="list-style-type: none"> <li>• High sensitivity to the setting of input parameters</li> <li>➤ Poor cluster descriptors</li> <li>• Unsuitable for high dimensional datasets</li> </ul>                                                                                                                                    |
| Grid            | <ul style="list-style-type: none"> <li>• Discovery of arbitrary shaped clusters with varying size</li> <li>• able to bear noisy data and outlier in the dataset</li> </ul>                                                                    | <ul style="list-style-type: none"> <li>• high time complexity</li> <li>• optimal grid size identification</li> </ul>                                                                                                                                                                                                               |



## NOTES

### 7.3.5 K- means

k-means is one of the simplest unsupervised learning algorithms that solve the well known clustering problem in partitions. The procedure follows a simple and easy way to classify a given data set through a certain number of clusters (assume  $c$  clusters) fixed prior.

#### Algorithmic Steps for k-means Clustering

Let  $X = \{x_1, x_2, x_3, \dots, x_n\}$  be the set of data points

- 1) Randomly select ' $c$ ' cluster centers.
- 2) Calculate the distance between each data point and cluster centers.
- 3) Assign the data point to the cluster center whose distance from the cluster center is minimum of all the cluster centers..
- 4) Recalculate the new cluster center using:

$$v_i = (1/c_i) \sum_{j=1}^{c_i} x_j$$

where, ' $c_i$ ' represents the number of data points in  $i^{th}$  cluster.

- 5) Recalculate the distance between each data point and new obtained cluster centers.
- 6) If no data point was reassigned then stop, otherwise repeat from step 3).

#### Advantages of K-Means

- 1) Fast, robust and easier to understand.
- 2) Relatively efficient:  $O(knd)$ , where  $n$  is number of objects,  $k$  is number of clusters,  $d$  is number of dimension of each object, and  $t$  is number of iterations. Normally,  $k, t, d \ll n$ .
- 3) Gives best result when data set are distinct or well separated from each other.

#### Disadvantages of K- Means

- 1) The learning algorithm requires prior specification of the number of cluster centers.
- 2) If there are two highly overlapping data then k-means will not be able to resolve that there are two clusters.
- 3) The learning algorithm is not invariant to non-linear transformations i.e. with different representation of data we get different results.
- 4) Randomly choosing of the cluster center cannot lead us to the fruitful result.
- 5) Applicable only when mean is defined i.e. fails for categorical data.
- 6) Unable to handle noisy data and outliers.
- 7) Algorithm fails for non-linear data set.

### 7.3.6 K- Medoid

The ***k*-medoids or PAM algorithm** is a clustering algorithm reminiscent to the *k*-means algorithm. Both the *k*-means and *k*-medoids algorithms are partitional (breaking the dataset up into groups) and both attempt to minimize the distance between points labeled to be in a cluster and a point designated as the center of that cluster. In contrast to the *k*-means algorithm, *k*-medoids chooses data points as centers (medoids or exemplars) and can be used with arbitrary distances, while in *k*-means the centre of a clusters is not necessarily one of the input data points (it is the average between the points in the cluster). *k*-medoid is a classical partitioning technique of clustering, which clusters the data set of *n* objects into *k* clusters, with the number *k* of clusters assumed known *a priori* (which implies that the programmer must specify *k* before the execution of the algorithm). The “goodness” of the given value of *k* can be assessed with methods such as silhouette.

It is more robust to noise and outliers as compared to *k*-means because it minimizes a sum of pair wise dissimilarities instead of a sum of squared Euclidean distances.

A medoid can be defined as the object of a cluster whose average dissimilarity to all the objects in the cluster is minimal, that is, it is a most centrally located point in the cluster.

#### Algorithmic steps for k-Medoids clustering

1. Initialize: select *k* of the *n* data points as the medoids
2. Associate each data point to the closest medoid.
3. While the cost of the configuration decreases: For each medoid *m*, for each non-medoid data point *o*:
4. Swap *m* and *o*, associate each data point to the closest medoid, recompute the cost (sum of distances of points to their medoid)
5. If the total cost of the configuration increased in the previous step, undo the swap

#### Advantages of K-Medoid

- 1) It is simple to understand and easy to implement.
- 2) K-Medoid Algorithm is fast and converges in a fixed number of steps.
- 3) Partition Around Medoid (PAM) algorithm is less sensitive to outliers than other partitioning algorithms.

#### Disadvantages of K-Medoid

- 1) K-Medoids is more costly than K-Means Method because of its time complexity.
- 2) It does not scale well for large datasets.
- 3) Results and total run time depends upon initial partitions.

## NOTES

## NOTES

### 7.3.7 CLARA (Clustering for Large Applications)

CLARA means clustering large applications and has been developed by Kaufman and Rousseeuw in 1990. This partitioning algorithm has come into effect to solve the problem of Partition Around Medoids (PAM). CLARA extends their K-Medoids approach for large number of object. This technique selects arbitrarily the data using PAM. According to Raymond T. Ng and Jiawei Han the following steps are performed in case of CLARA.

- 1) Draw a sample of  $40+2k$  objects randomly from the entire data set, and call Algorithm PAM to find  $k$  medoid of the sample.
- 2) For each of the object determine the specific  $K$  medoid which is similar to the given object ( $O_j$ ).
- 3) Calculate the average dissimilarity of the clustering thus obtained. If the value thus obtained is less than the present minimum we can use it and retained the  $K$ -Medoid found in the second step as best of medoid.
- 4) We can repeat the steps for 'next iteration'.

#### Advantage of CLARA

- 1) CLARA Algorithm deals with larger data sets than PAM (Partition Around Medoids).

#### Disadvantages of CLARA

- 1) The efficient performance of CLARA depends upon the size of dataset.
- 2) A biased sample data may result into misleading and poor clustering of whole datasets.

### 7.3.8 CLARANS (Clustering large Application Based on Randomized Search)

K-Medoid algorithm does not work effectively for large data sets. Therefore CLARA has been improved and modified so as to used large databases. CLARANS has been developed by Ng and Han in 1994. To overcome the limitations of K-Medoid algorithm CLARANS is introduced. CLARANS (Clustering large Application Based on Randomized Search) is partitioning method used for large database. It is more efficient and scalable than both PAM and CLARA. As in case of CLARANS the following steps to be performed:

- 1) Input parameters  $numlocal$  and  $maxneighbour$ .
- 2) Select  $K$  object from the database object  $D$  randomly.
- 3) Mark these  $K$  object as selected  $S_i$  and all other as non-selected  $S_i$ .
- 4) Calculate the cost  $T$  for selected  $S_i$ .
- 5) If  $T$  is negative update medoid set. Otherwise selected medoid is chosen as local optimum.

- 6) Restart the selection of another set of medoid and find another local optimum.
- 7) CLARANS stops until returns the best.

CLARANS uses two parameters – numlocal and maxneighbour. Numlocal means number local minima obtained and maxneighbour means maximum number of neighbour examined. The higher the value of latter, the closer will be CLARANS to PAM and longer will each search of local minima be. This is an advantage because the quality of local minima is higher and less number of local minima are to be found out.

## NOTES

### Advantages of CLARANS

- 1) It is easy to handle outliers.
- 2) CLARANS result is more the effective as compare PAM and CLARA.

### Disadvantages of CLARANS

- 1) It does not guarantee to give search to a localized area.
- 2) It uses randomize samples for neighbors.
- 3) It is not much efficient for large datasets.

### 7.3.9 BIRCH (Balanced Iterative Reducing and Clustering using Hierarchies)

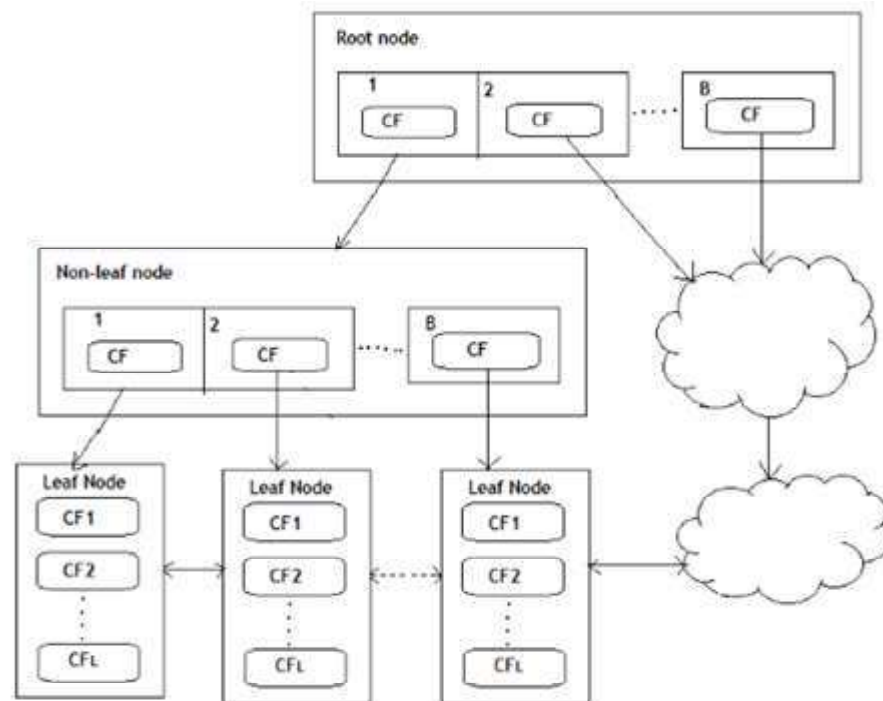
It is an agglomerative hierarchical algorithm which uses a Clustering Feature (CF-Tree) and incrementally adjusts the quality of sub clusters. CF tree is a height balanced tree that stores the clustering features for a hierarchical clustering. Cluster of data points is represented by a triple of numbers (N, LS, SS)

Where,     N= Number of items in the sub cluster  
              LS=Linear sum of the points  
              SS=sum of the squared of the points

A CF Tree structure is given as below

- Each non-leaf node has at most B entries.
- Each leaf node has at most L CF entries which satisfy threshold T, a maximum diameter of radius
- P(page size in bytes) is the maximum size of a node
- Compact: each leaf node is a subcluster, not a data point

## NOTES



*Fig. 7.2 A CF tree structure*

The algorithm is as follows:

1. **Load data into memory:** CF Tree is constructed with one scan of the data. Subsequent phases become fast, accurate and less order sensitive.
2. **Condense data:** Rebuild the CF tree with larger T.
3. **Global Clustering:** Use the existing clustering algorithm on CF leaves.
4. **Cluster refining-**Do additional passes over the dataset and reassign data points to the closest centroids from above step.
5. The process continuous until to form k no. of clusters.

### **Advantage of BIRCH**

Finds a good clustering with a single scan and improves the quality with a few additional scans

### **Disadvantage of BIRCH**

Handles only numeric data

### **7.3.10 CACTUS – Clustering Categorical Data Using Summaries**

It is a very fast and scalable algorithm for finding the clusters. A hierarchy structure is used to generate maximum segments or clusters. The algorithm includes majorly three phases:

- **Summarization Phase:** Compute the summary information.
- **Clustering Phase:** Discover a set of candidate clusters.
- **Validation Phase:** Determine the actual set of clusters.

A two step procedure deals with the description of algorithm as follows:

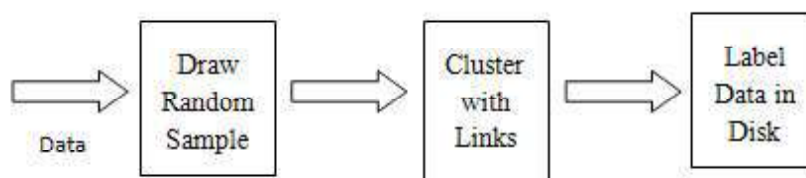
1. Attributes are strongly connected if the data points are having larger frequency.
2. Clusters are formed based on the co-occurrences of attribute value pairs.
3. A cluster is formed if any segment is having no of elements  $\alpha$  times greater than elements of other.

#### Advantages of CACTUS

- 1) Discovers clusters consisting of overlapping cluster-projections on any attribute
- 2) Discovers clusters where two or more clusters share the same cluster projection
- 3) CACTUS is very fast and scalable (only two scans of the dataset)

#### 7.3.11 ROCK - Robust Clustering Algorithm for Categorical Attributes

ROCK a robust hierarchical-clustering algorithm is an agglomerative hierarchical clustering based on the notion of links. It is appropriate for handling large data sets. ROCK combines, from a conceptual point of view, nearest neighbor, relocation, and hierarchical agglomerative methods. In this algorithm, cluster similarity is based on the number of points from different clusters that have neighbors in common. The steps involved in clustering using ROCK are described in Figure 7.3.



*Fig. 7.3 ROCK Process*

The algorithm is as follow:

1. Obtaining a random sample of the data.
2. Performing clustering on the data using the link agglomerative approach. A goodness measure is used to determine which pair of points is merged at each step.
3. Build a heap and maintain heap for each cluster.

#### NOTES

4. A goodness measure based on the criterion function will be calculated between pairs of clusters.
5. Merge the clusters which have maximum value of criteria function.

## NOTES

### Advantage of ROCK

ROCK performs well on real categorical data, and respectably on time-series data.

### Disadvantage of ROCK

High worst case complexity

### 7.3.12 DBSCAN (Density Based SCAN clustering algorithm)

It is a density-based clustering non-parametric algorithm: given a set of points in some space, it groups together points that are closely packed together (points with many nearby neighbors), marking as outliers points that lie alone in low-density regions (whose nearest neighbors are too far away). It is a connectivity based algorithm which consists of three points namely core, reachable and noise.

- A point  $p$  is a *core point* if at least  $\text{minPts}$  points are within distance  $\hat{a}$  (a parameter specifying the radius of a neighborhood with respect to some point) of it (including  $p$ ).
- A point  $q$  is *directly reachable* from  $p$  if point  $q$  is within distance  $\hat{a}$  from core point  $p$ . Points are only said to be directly reachable from core points.
- A point  $q$  is *reachable* from  $p$  if there is a path  $p_1 \dots p_n$  with  $p_1 = p$  and  $p_n = q$ , where each  $p_{i+1}$  is directly reachable from  $p_i$ . Note that this implies that all points on the path must be core points, with the possible exception of  $q$ .
- All points not reachable from any other point are *outliers* or *noise points*.

Two points,  $p$  and  $q$  are density-connected if there is a point  $o$  such that both  $p$  and  $q$  are reachable from  $o$ . Density-connectedness is symmetric. A cluster then satisfies two properties that all points within the cluster are mutually density-connected. And if a point is density-reachable from any point of the cluster, it is part of the cluster as well.

The algorithm is as follows:

1. Set of points to be considered to form a graph.
2. Create an edge from each point  $c$  to the other point in the neighborhood of  $c$ .
3. If set of nodes  $N$  not contain any core points then terminate  $N$ .
4. Select a node  $X$  that must be reached from  $c$ .
5. Repeat the procedure until all core points forms a cluster.

### Advantages of DBSCAN

1. DBSCAN does not require one to specify the number of clusters in the data a priori, as opposed to k-means.
2. DBSCAN can find arbitrarily shaped clusters. It can even find a cluster completely surrounded by (but not connected to) a different cluster. Due to the MinPts parameter, the so-called single-link effect (different clusters being connected by a thin line of points) is reduced.
3. DBSCAN has a notion of noise, and is robust to outliers.

### Disadvantages of DBSCAN

1. DBSCAN is not entirely deterministic: border points that are reachable from more than one cluster can be part of either cluster, depending on the order the data are processed.
2. DBSCAN cannot cluster data sets well with large differences in densities, since the minPts- $\epsilon$  combination cannot then be chosen appropriately for all clusters.
3. If the data and scale are not well understood, choosing a meaningful distance threshold  $\epsilon$  can be difficult.

OPTICS (Ordering Points to Identify the Clustering Structure) is an extension of DBSCAN algorithm which is also based on the same parameters as DBSCAN algorithm. The run time of OPTICS is 1.6 times greater than DBSCAN algorithm.

### 7.3.13 STIRR (Sieving Through Iterated Relational Reinforcement)

STIRR (Sieving through Iterated Relational Reinforcement), proposed by Gibson, Kleinberg and Raghavan, is an iterative algorithm based on non-linear dynamical systems. The salient features of STIRR are the following.

1. The database is represented as a graph, where each distinct value in the domain of each attribute is represented by a weighted node. Thus, if there are  $N$  attributes and the domain size of the  $l$ th attribute is  $d_l$ , then the number of nodes in the graph is " $d_l$ ".
2. For each tuple in the database, an edge represents a set of nodes which participate in that tuple. Thus, a tuple is represented as a collection of nodes, one from each attribute type.
3. We assign a weight to each node. The set of weights of all the nodes define the configuration of this structure.
4. The algorithm proceeds iteratively to update the weight of each node, based on the weights of other nodes to which it is connected. Thus, it moves from one configuration to the other till it reaches a stable point.

### NOTES



- The updating of the weights depends on a combiner function that combines the weights of the nodes participating in a given tuple. Their convergence is dependent on the combiner function.

## NOTES

**Initial configuration:** The initial weights of nodes can be either assigned uniformly, or randomly, or by a focusing technique. In the uniform initialization, all weights are set to 1. In the random initialization, each weight is an independently selected random value in the interval  $[0,1]$ . We can focus a portion of the graph by initializing the portion to 1 and the remaining part of the graph to 0.

**Weight Update:** STIRR iteratively changes the configuration by updating the weight of any single node. The new weight of any node is calculated, based on a combiner function. Typically, a combiner function combines the weights of other nodes participating in any tuple with the given node for which the weight is to be updated. A combiner function can simply add all the weights, or can multiply all the weights, or may combine them in some other way.

### Advantages of STIRR

- Gives arbitrary shaped clusters.
- STIRR is fast and scalable.

### Disadvantages of STIRR

- clusters consisting of overlapping cluster-projections on any attribute
- clusters where two or more clusters share the same cluster projection

*Table 7.3 Comparison of Various clustering methods*

| Algorithm Type | Algorithm Name | Dataset Size | High Dimensionality | Avoid Outliers | Dataset Type            | Cluster Shape     |
|----------------|----------------|--------------|---------------------|----------------|-------------------------|-------------------|
| Partition      | K-Means        | Large        | No                  | No             | Numerical               | Non convex        |
|                | K-Mediod       | small        | Yes                 | Yes            | Categorical             | Non convex        |
|                | CLARA          | Large        | No                  | No             | Numerical               | Non convex        |
|                | CLARANS        | Large        | No                  | No             | Numerical               | Non convex        |
| Hierarchical   | BIRCH          | Large        | No                  | No             | Numerical               | Non convex        |
|                | CACTUS         | Small        | NO                  | No             | Categorical             | Hyper rectangular |
|                | ROCK           | Large        | No                  | No             | Numerical & Categorical | Arbitrary         |
| Density Based  | DBSCAN         | Large        | No                  | No             | Numerical               | Arbitrary         |
| Grid           | STIRR          | Large        | No                  | No             | Categorical             | Arbitrary         |

## Cluster Evaluation

**Clustering evaluation** assesses the feasibility of clustering analysis on a data set and the quality of the results generated by a clustering method. The tasks include assessing clustering tendency, determining the number of clusters, and measuring clustering quality.

In **Assessing clustering tendency** task, for a given data set, we assess whether a nonrandom structure exists in the data. Blindly applying a clustering method on a data set will return clusters; however, the clusters mined may be misleading. Clustering analysis on a data set is meaningful only when there is a nonrandom structure in the data.

**Determining the number of clusters in a data set:** A few algorithms, such as  $k$ -means, require the number of clusters in a data set as a parameter. Moreover, the number of clusters can be regarded as an interesting and important summary statistic of a data set. Therefore, it is desirable to estimate this number even before a clustering algorithm is used to derive detailed clusters.

**Measuring clustering quality:** After applying a clustering method on a data set, we want to assess how good the resulting clusters are. A number of measures can be used. Some methods measure how well the clusters fit the data set, while others measure how well the clusters match the ground truth, if such truth is available. There are also measures that score clustering and thus can compare two sets of clustering results on the same data set.

## Cluster Validation

In Cluster validation the reliability and quality of clusters generated from clustering process is determined. The main problem of prediction of correct number of clusters with clustering is overcome by using cluster validity indices. A cluster is selected as best cluster among all which make the best or most effective use of a validity index. The silhouette index, Dunn's index and Davies and Bouldin index are the three main indices for optimal partitioning of data objects.

The **silhouette index** value for any data item of a cluster is given as the difference between the minimum value of the average distance of this data item to data in any other cluster and the average distance between this data item to the other data in the same cluster divided by the maximum value of these two average distance values. The range of the silhouette value is from -1 to 1. If the silhouette value for a data item is great than 0 then it is similar to other values in the given cluster and the cluster is valid and if it is less than 0 then it shows dissimilarity and the cluster is invalid. Whereas, **Dunn's index** for cluster validation is to identify the compactness of the clusters from inside and how well they are separated. **Davies and Bouldin index** is just like the Dunn's index. It also does not depend upon the number of clusters. It gives the average similarity between the clusters and a minimum value of it with the number of clusters indicates that the clustering scheme is good.

## NOTES

**Table 7.4** Difference between Supervised and Unsupervised Learning

| <b>Basis For Comparison</b>     | <b>Supervised Learning</b>        | <b>Unsupervised Learning</b>           |
|---------------------------------|-----------------------------------|----------------------------------------|
| <b>input data</b>               | uses known and labeled input data | uses unknown input data                |
| <b>computational complexity</b> | very complex in computation       | less computational complexity          |
| <b>real time</b>                | uses off-line analysis            | uses real time analysis of data        |
| <b>number of classes</b>        | number of classes is known        | number of classes is not known         |
| <b>accuracy of results</b>      | accurate and reliable results     | moderate accurate and reliable results |

**NOTES**

**Table 7.5** Difference between Clustering and Classification

| <b>Basis for Comparison</b> | <b>Classification</b>                                                                          | <b>Clustering</b>                                                                                                                             |
|-----------------------------|------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------|
| Basic                       | This model function classifies the data into one of numerous already defined definite classes. | This function maps the data into one of the multiple clusters where the arrangement of data items is relies on the similarities between them. |
| Involved in                 | Supervised learning                                                                            | Unsupervised learning                                                                                                                         |
| Training sample             | Provided                                                                                       | Not provided                                                                                                                                  |

**Check Your Progress**

3. What are the classifications of clustering methods?
4. What is K – Means?
5. Define K – Mediod.
6. What does CACTUS stand for?
7. What is cluster evaluation?
8. Define medoid.

**7.4 ANSWERS TO CHECK YOUR PROGRESS QUESTIONS**

1. Clustering is the process of making group of abstract objects into classes of similar objects. A cluster of data objects can be treated as a one group.
2. The applications of cluster analysis are: Biology, Web, Marketing, Education and Social Sciences.
3. The clustering methods can be classified into following categories: Partitioning Method, Hierarchical Method, Density-based Method and Grid-Based Method

4.  $k$ -means is one of the simplest unsupervised learning algorithms that solve the well known clustering problem in partitions.
5. The  $k$ -medoids or PAM algorithm is a clustering algorithm reminiscent to the  $k$ -means algorithm.
6. CACTUS stands for Clustering Categorical Data Using Summaries.
7. Clustering evaluation assesses the feasibility of clustering analysis on a data set and the quality of the results generated by a clustering method.
8. A medoid can be defined as the object of a cluster whose average dissimilarity to all the objects in the cluster is minimal, that is, it is a most centrally located point in the cluster.

## NOTES

---

### 7.5 SUMMARY

---

- Clustering is the process of making group of abstract objects into classes of similar objects. A cluster of data objects can be treated as a one group.
- Clustering is useful in understanding the hidden structure in data, automatically organizing data and pre-processing for further analysis.
- The applications of cluster analysis are: Biology, Web, Marketing, Education, Social Science and Outliers detection.
- The clustering methods can be classified into following categories:
  - i. Partitioning Method
  - ii. Hierarchical Method
  - iii. Density-based Method
  - iv. Grid-Based Method
- $k$ -means is one of the simplest unsupervised learning algorithms that solve the well known clustering problem in partitions. The procedure follows a simple and easy way to classify a given data set through a certain number of clusters (assume  $c$  clusters) fixed prior.
- The  $k$ -medoids or PAM algorithm is a clustering algorithm reminiscent to the  $k$ -means algorithm.
- CLARA means clustering large applications and has been developed by Kaufman and Rousseuw in 1990. This partitioning algorithm has come into effect to solve the problem of Partition Around Medoids (PAM). CLARA extends their  $K$ -Medoids approach for large number of object.
- CLARA has been improved and modified so as to use large databases. CLARANS has been developed by Ng and Han in 1994. Clarans (Clustering large Application Based on Randomized Search) is partitioning method used for large database.

## NOTES

- BIRCH (Balanced Iterative Reducing and Clustering using Hierarchies) is an agglomerative hierarchical algorithm which uses a Clustering Feature (CF-Tree) and incrementally adjusts the quality of sub clusters.
- CACTUS – Clustering Categorical Data Using Summaries is a very fast and scalable algorithm for finding the clusters. The algorithm includes majorly three Phases: Summarization Phase, Clustering Phase and Validation Phase.
- ROCK - Robust Clustering algorithm for Categorical attributes is a robust hierarchical-clustering algorithm is an agglomerative hierarchical clustering based on the notion of links. It is appropriate for handling large data sets.
- DBSCAN (Density Based SCAN clustering algorithm) is a density-based clustering non-parametric algorithm: given a set of points in some space, it groups together points that are closely packed together (points with many nearby neighbors), marking as outliers points that lie alone in low-density regions (whose nearest neighbors are too far away).
- STIRR (Sieving through Iterated Relational Reinforcement), proposed by Gibson, Kleinberg and Raghavan, is an iterative algorithm based on non-linear dynamical systems.
- Clustering evaluation assesses the feasibility of clustering analysis on a data set and the quality of the results generated by a clustering method. The tasks include assessing clustering tendency, determining the number of clusters, and measuring clustering quality.
- In Cluster validation the reliability and quality of clusters generated from clustering process is determined.

---

## 7.6 KEY WORDS

---

- **Outlier:** A data point on a graph or in a set of results that is very much bigger or smaller than the next nearest data point.
- **Robust:** It is the ability to withstand or overcome adverse conditions.
- **ROCK:** It is a robust hierarchical-clustering algorithm is an agglomerative hierarchical clustering based on the notion of links.

---

## 7.7 SELF ASSESSMENT QUESTIONS AND EXERCISES

---

### Short Answer Questions

1. What the three major phases in the algorithm of CACTUS?
2. Define silhouette index.
3. What are the advantages and disadvantages of BIRCH?

4. List the Algorithmic steps for k-medoids clustering.
5. Discuss the hierarchical methods for clustering.
6. What is CLARANS?

### Long Answer Questions

1. Discuss K – Means at length. Also, discuss its advantages and disadvantages.
2. Explain the 4 different categories of clustering methods.
3. What is BIRCH? Discuss the structure of a CF tree with the help of a diagram.
4. Explain DBSCAN and its algorithm. List the advantages and disadvantages of DBSCAN.
5. Explain the salient features of STIRR. What are its advantages and disadvantages?
6. What are the differences between Supervised and Unsupervised Learning?
7. Differentiate between Clustering and Classification.

### NOTES

---

## 7.8 FURTHER READINGS

---

- Han, Jiawei, Micheline Kamber and Jian Pei. 2011. *Data Mining: Concepts and Techniques*, 2nd edition. Amsterdam: Elsevier.
- Pujari, Arun K. 2010. *Data Mining Techniques*, 2nd edition. United States: Universities Press.
- Anahory, Sam and Dennis Murray. 1997. *Data Warehousing in the Real World: A Practical Guide for Building Decision Support Systems*, 1st edition. Boston: Addison Wesley.
- Witten, I. H. and Eibe Frank. 2005. *Data Mining: Practical Machine Learning Tools and Techniques*, 2nd edition. Amsterdam: Elsevier.
- Soman, K. P., Shyam Diwakar and V. Ajay. 2006. *Insight Into Data Mining: Theory and Practice*. New Delhi: PHI.

---

## UNIT 8 INTRODUCTION TO MACHINE LEARNING

---

### NOTES

#### Structure

- 8.0 Introduction
- 8.1 Objectives
- 8.2 Machine Learning: An Overview
  - 8.2.1 Steps Involved in a Machine Learning Process
  - 8.2.2 Types of Machine Learning
  - 8.2.3 Machine Learning Applications
- 8.3 Machine Learning and Data Mining
- 8.4 Answers to Check Your Progress Questions
- 8.5 Summary
- 8.6 Key Words
- 8.7 Self Assessment Questions and Exercises
- 8.8 Further Readings

---

### 8.0 INTRODUCTION

---

In the previous chapters, we have gone through the various data mining concepts along with several types of classification and clustering algorithms. These techniques were able to give insights from the data and thus help in making decisions. Now days, most of the work is done through computers/IT systems as these are the task that are too tedious to be performed by an individual. Therefore, it is very important that machines learn to do the work as though it is performed by humans.

---

### 8.1 OBJECTIVES

---

After going through this unit, you will be able to:

- Understand the machine learning process and its types
- Discuss the steps involved in machine learning process
- Discuss the differences between machine learning and data mining

---

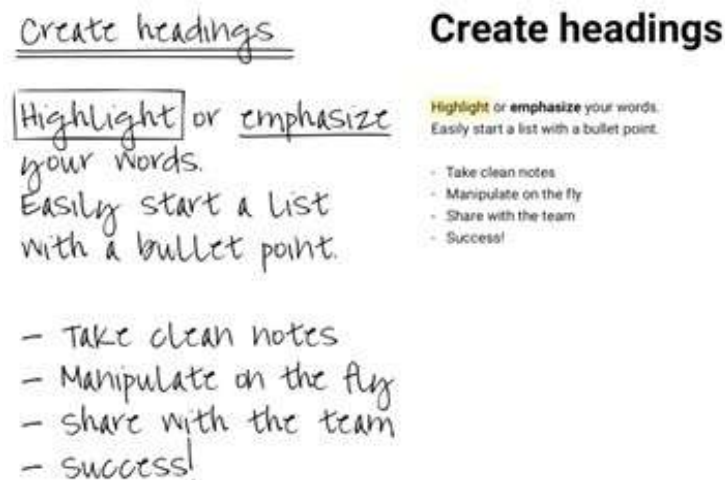
### 8.2 MACHINE LEARNING: AN OVERVIEW

---

One such example is recognizing the image of the handwritten script so that it can be converted into digital format. Many of the older handwritten scripts were written during the time when the computers were not invented. Therefore, to preserve these scripts, they have to be converted in the digital form. This task was very time taking and error prone when done by the humans. But if the machines are trained

to recognize the hand written characters and convert them into a digital copy then it becomes an easy, fast and accurate task. This requires the machines to learn from examples so that they can perform the task autonomously. They must also be able to correct themselves if they find any error in their judgment. Therefore, the ability of the machines to learn to perform such task are said to have an artificial intelligence and this type of learning is known as machine learning. An example of conversion of digitally converted text is shown in Figure 8.1.

## NOTES



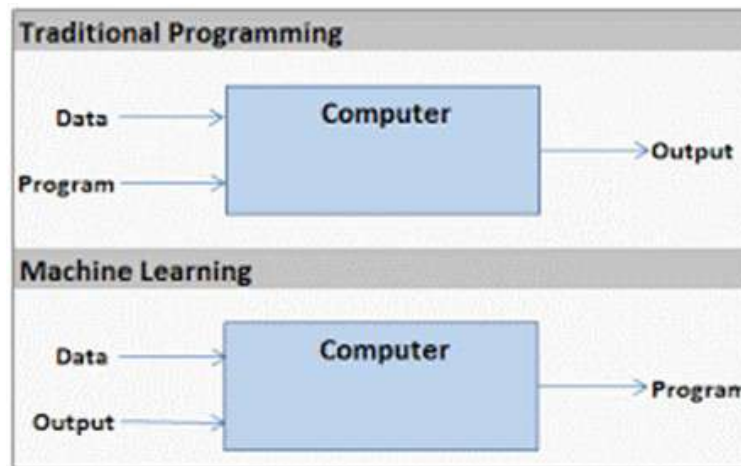
*Fig. 8.1 Conversion of handwritten script to digital script*

Formally, we can say that machine learning is an area of artificial intelligence that has computer algorithms which enables the computer to learn without being explicitly programmed. Machine learning (ML) also helps you use already existing data to make better future decisions. They also construct mathematical models and discover patterns in data. For example, one possible application of a machine learning model would be to predict how likely a customer is to come back to purchase more products based on his/her feedback.

The algorithms in the machine learning are different from any traditional program in the sense that in a traditional algorithm, the data and inputs are provided to get the desired output. In case of machine learning, the data and corresponding outputs are provided to give a resultant program or model and which can now run in the form of traditional program. Thus the model created needs a set of test input which tells the effectiveness of the trained model. For e.g. a traditional program may be a system which takes the customer transactions as inputs and gives the total amount to be paid as output. On the other hand, if you want to predict who will pay the bills late, first step is to identify the inputs like customer details and the bills and the output whether bill is paid late or not. Using these two, the machine learning use creates a model that in future is used to predict whether a new customer will pay the bill on time or not.



## NOTES



*Fig. 8.2 Process of Traditional programming and Machine learning*

The question arises how exactly machines are taught to learn. The answer is teaching a machine can be done by giving them experience of what they should remember. For example, if we want a machine to learn to differentiate between a dog and a cat then machine learning model would be taught how to differentiate between a dog and a cat by showing different set of their images. This would help the machine to extract features from the images based on which it can learn the differences between a dog and a cat like for e.g. dogs have longer nose than cat. So, if the machine comes across the picture where the nose is longer than usual then the chances of the picture being of the dog increases.

### 8.2.1 Steps Involved in a Machine Learning Process

The machine learning process is divided into two phases: Training Phase and Testing Phase. The various steps involve in these two phases are described below:

- 1. Data collection:** This step consists of choosing and gathering of the relevant data from various sources and the data can be in the form of a text, excel or a database file. This is the most important step as quality and the volume of data is the base for the efficient machine learning model.
- 2. Data Preprocessing:** The effectiveness of the machine learning depends on the quality of the data we are feeding it for learning purpose. Therefore, this step requires considerable work and effort so as to bring the data in the correct form. The data acquired in the first step is often raw and unstructured with some missing values, duplicate values or the noise which would interfere in getting good results. The preprocessing step brings all the data from various sources in the form which can be readily used by the machine learning algorithm for creating a model.
- 3. Choose a model:** The data from the pre-processing step is now ready to be used for the creation of machine learning model. This pre-processed data is in structured form and now can be used as an input for the machine

learning algorithm. The algorithm to be chosen depends on the type of prediction problem and the type of data. If the prediction problem is a type of classification problem then a classification algorithm is chosen or if it is type of regression problem, we would choose a suitable regression technique. Some algorithms are more suited for text data while other works best in case of image data. This task of can be an iterative one, and different algorithms can be worked upon until the best results are achieved. The result of the algorithm when applied on data is a model which is then used on the new set of data to analyze its effectiveness.

## NOTES

- 4. Train the model:** Before the training process of the model starts, the acquired data needs to be divided into two different sets: training set and testing set. The training set is used by the model as the samples to learn from and testing set evaluates the effectiveness the model. The size and quality of the training set also determines how accurate our model is. The larger the training set, the better model is created. Also, it should represent the future test set as a whole. Generally, the optimum ratio of training and testing set varies between 70/30 to 80/20. The training data may or may not contain the correct answer or the target value depending upon the type of learning. The learning algorithm finds patterns in the training data that maps the input data attributes to the target, and it outputs a machine learning model that captures these patterns. This model can be used to get the predictions on the test data for which the expected value is unknown.

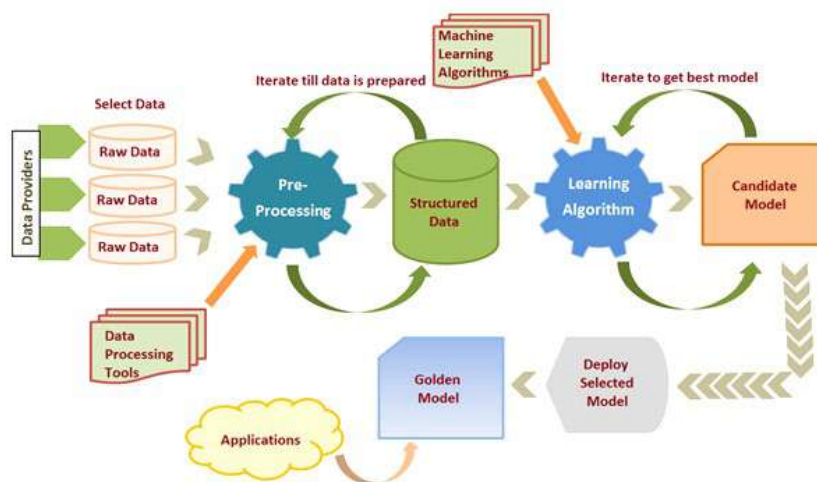


Fig. 8.3 Machine Learning Process

- 5. Evaluate the model:** After the training phase, the model is evaluated to see if it is efficient in predicting the value for the task which it is expected to. This is the time where the test dataset is used. Therefore, the test data, which is never used to train the machine learning model, is run on the model to get the predictions or the outcome. This allows us to evaluate the real performance of the model on the data never seen before and hence decide

## NOTES

whether the model is doing justice while solving the problem or not. The efficiency of the model depends on the volume of data. The more the training data, the better the model performs.

**6. Parameter Tuning:** The evaluation step is followed by parameter tuning. The parameters are the constraints or certain values that need to be set in order for learning algorithm to work. The value of these parameters depends on many factors and there are few parameters which are implicitly assumed when we start the training and these assumptions can be tested once we train the model. Therefore, the goal is to set the parameter values to optimum so that the learning is improved and the performance of the model is maximized. This is known as parameter tuning. One example of parameter is the number of times the training dataset is run during the training step i.e. the training dataset can be fed into the model a number of times. This can sometimes result in increased accuracy of the model.

**7. Prediction:** Machine learning models are used to provide the answer to the questions for which the model is built. Therefore, prediction, or inference, is the step where the answers to the questions are finally predicted. This is the point where result of all the earlier steps of machine learning is realized.

Therefore, to summarize it can be said that the power of machine learning is to predict and decide based on the model rather than applying manual rules or human judgment.

### 8.2.2 Types of Machine Learning

There are different ways in making a machine to learn. Three of most important learning methods are as follows:

**(a) Supervised Learning:** Machine learning means the computer can automatically learn to recognize hidden pattern in the given data and therefore helps to make better decisions. This learning method as the name suggest learns in the presence of some supervision. The data used for learning in this case is already mapped with the correct values. These set of values are used to iteratively train and correct the model based on the matching of the predicted values and the correct values. Learning is said to be done when the model achieves an acceptable level of performance.

In other words, if we consider the mapping function to be “ $f$ ” and the set of input variables as “ $X$ ” and the set of output variable as “ $Y$ ”, then learning can be defined as:

$$Y = f(X)$$

Where we try to construct a trained model by finding a function ‘ $f$ ’ that maps input objects ( $X$ ) to the output objects ( $Y$ ) in the training data. This function is fine – tuned by comparing the inferred output with the actual or desired output and error are determined. Whenever a new input set is fed to the function, it is able to predict the values for  $Y$  with the acceptable level of accuracy.

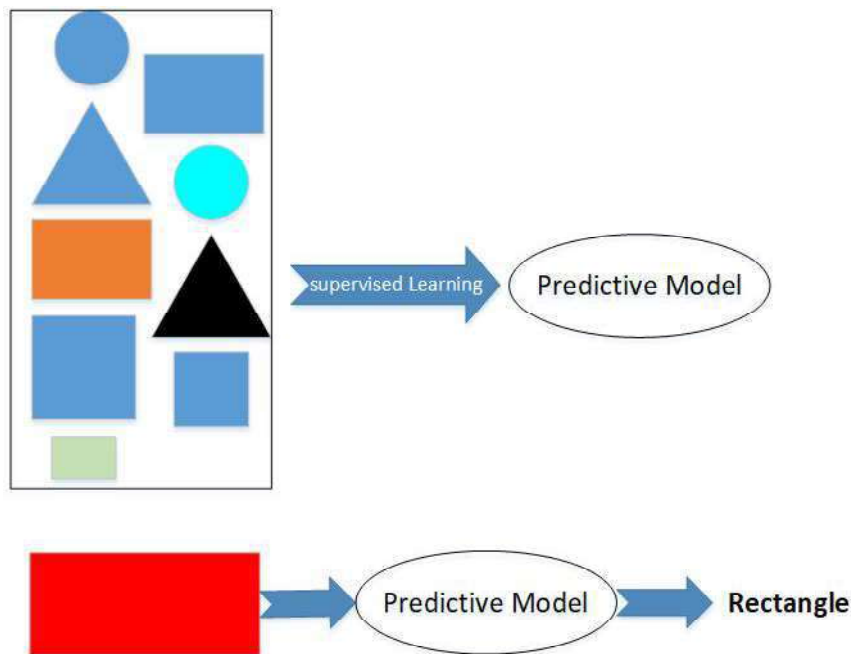


Fig. 8.4 Supervised learning process

One of the examples of supervised learning can be identifying the 2D shape of the object from the set of given objects. Now the first step is to train the machine by showing different types of shapes one by one.

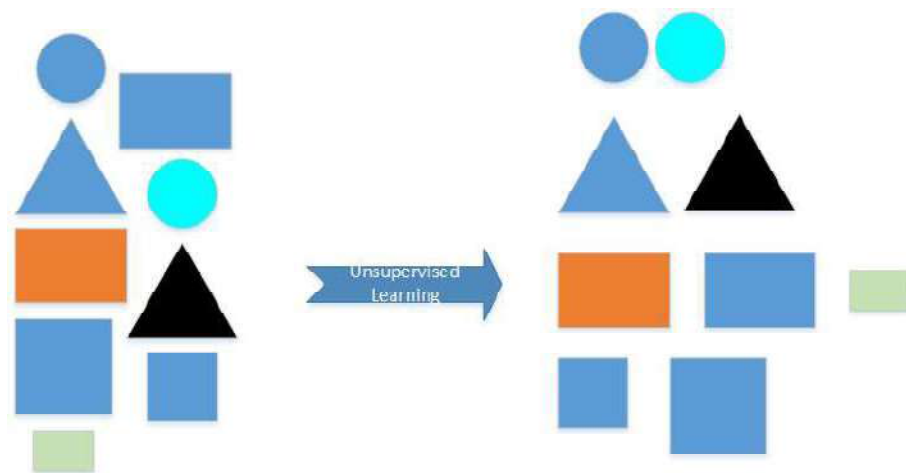
- If the shape has no edges, then it will be labeled as circle
- If the shape has 3 edges, it will be a triangle.
- If the shape has 4 edges, where set of two edges are of same length, then it is a rectangle.
- If the shape has 4 edges, where all edges are of same length, then it is a square.

Now suppose after the training, a new shape circle is given to the machine to identify it, therefore since machine has already been trained from the training data, it would be able to identify it as a CIRCLE.

**(b) Unsupervised Learning:** Unsupervised learning, as the name suggests, is the training the machine without any supervision. In this type of training, no information about the class or the labels is provided, only the training data is passed for the training of machine. Here the task of machine is to group the information according to similarities, patterns and differences without any prior training of data. Unlike supervised learning, no teacher is provided that means no training will be given to the machine. Therefore machine is restricted to find the hidden structure in unlabeled data by our-self.

## NOTES

NOTES



**Fig. 8.5** Unsupervised Learning Processes

For instance, suppose the machine is given an image having different types of shapes that it has not seen ever. Thus the machine has no idea about the features of different shapes like circle, triangle, rectangle or square so it can't categorize it in these categories. But it can categorize them according to their properties, similarities, patterns, and differences i.e., the machine can categorize the above picture into 4 parts, each part having different types of shapes without learning beforehand. From the above example it can be seen that the target classes are not specified while training. The model is left to infer a function by discovering and learning the structure and features of the unlabelled data.

- (c) **Reinforcement Learning:** Reinforcement learning instead of working on the labeled training data works interactively with the environment in which it is working. Therefore, it helps the machine to learn based on its progress. At each phase of learning a feedback is provided taking in account the current context and based on this feedback the machine gets a reward and decides the next step to be taken. The decision regarding the next best step is taken by an agent based on the award or the punishment that is given for the particular step or decision. The goal is to maximize the awards and thus the performance of the machine.

To summarize the three types of learning methodologies, we can say that in supervised learning, there is an external supervisor that that helps the machine to learn but this supervisor is not the part of unsupervised learning. Here, the machine has the ability to interpret patterns and learning on its own with the help of input data. However, reinforcement learning the machine decides its next action based on the reward in the current step which is always maximized.

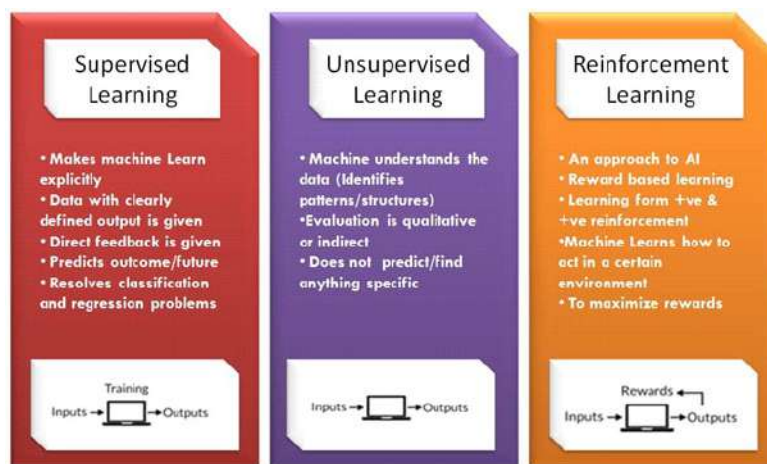


Fig. 8.6 Types of Machine learning at a glance

## NOTES

### 8.2.3 Machine Learning Applications

The area or the scope of machine learning is vast and it is increasing day by day. Here are some of the applications that would help in appreciating its scope in modern world. Though the applications are not limited to these but they would give a fair idea of the importance of machine learning.

- **Image Recognition**– It is one of the most common and useful machine learning applications. It helps to identify or classify the given image or even provide a suitable caption for the same. Another example of image recognition is face recognition which is used to act as the authentication process in many places.
- **Speech Recognition**– is an application that converts the spoken text into written text. This is also used to decode the spoken words for the purpose of giving automated commands, conversion of live speech into the transcripts for later usage etc. It is also a boon for person having vision related disabilities.
- **Medical Diagnosis**–Machine Learning offers vast number of methods, techniques, and tools that has helped the medical domain especially in the diagnosing various diseases. It is also used in recording the clinical symptoms along with the patient details and then giving out the prediction of the disease progression. It is also used in analyzing the medical data, looking for irregularities and improve the quality and efficiency of medical care.
- **Making Recommendations** – With the help of machine learning, various product recommender systems can be developed, which according to customer behavior, suggests the product they might be interested. Over the period of time, this recommender system learns as the data that is exposed to these system increases and helps the user discover more meaningful and relevant products.
- **Fraud Detection**–Machine learning, through many of its complex algorithms, is able to detect and recognize thousands of online fraud in the banking domains

like loans, credit cards etc. It is able to find the patterns of anomaly in day to day transactions done by the people. Not only that, it is also able to predict, who is like to commit fraud and raise an alarm in such situations.

## NOTES

### Check Your Progress

1. How are the algorithms in machine learning different from traditional program?
2. How does reinforcement learning work?

## 8.3 MACHINE LEARNING AND DATA MINING

In the last few chapters, various techniques like classification, clustering, association rule mining and clustering were introduced. While we can say data mining refers to extracting knowledge which is not inherently visible from huge amount of data and that knowledge can be in the form of new trends and patterns. It is an iterative process of creating a predictive or descriptive model with the help of historic data and thus helps in making improved decision making. It may use the techniques from machine learning but the basis of data mining is the real world data and the goal is to leverage the power of various pattern recognition techniques of machine learning.

Machine learning on the other hand involves the use of algorithms that automatically improves through the experience based the data being exposed. Basically it is an approach to develop human like intelligence in the machines also known as artificial intelligence. This allows computer to learn without being explicitly programmed. However, these two fields are related and most of the techniques and algorithms that are used in data mining are developed within the field of Machine Learning. Data mining also aspires to learn but in a more practical sense that it is more related to finding structural patterns in data and as a result give predictions with a certain level of accuracy.

Data mining can be used for varied purposes like prediction financial health of the organizations given the financial data for past few years and data outside the organization's environment. This might be useful for bigger organizations who want to make the funding related decision by comparing these predictions. The company might also carry out the trend analysis for the business organizations so as to better inform all the related departments like supply chain, marketing, finance etc. to gear up for the future sales.

Machine learning on the other hand apart from predicting the future trends and decisions, helps in the process of learning by applying new algorithms. It provides a foundation necessary for a machine to teach itself. This helps in improving the accuracy of the results over the period of time. For e.g. after placing an order for an item from Amazon, the website quickly takes us to the page where it recommends other item related to the ordered products. Fraud detection is another area where banks and other related financial institutions are investing heavily so that their investments are more secure.

**Table 8.1** Comparison between data mining and machine learning

| Factors                    | Data Mining                                                                                                                                                                                                                                                                                        | Machine Learning                                                                                                                                                                                                                                                                                             |
|----------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Scope</b>               | Data Mining is used to find out how different attributes of a data set are related to each other through patterns and data visualization techniques.<br>The goal of data mining is to find out relationship between 2 or more attributes of a dataset and use this to predict outcomes or actions. | Machine Learning is used for making predictions of the outcome such as price estimate or time duration approximation.<br>It automatically learns the model with experience over time. It provides real time feedback                                                                                         |
| <b>Working</b>             | Data Mining is the technique of digging deep into data to take out useful information                                                                                                                                                                                                              | Machine Learning is method of improving complex algorithms to make machines near to perfect by iteratively feeding it with trained dataset.                                                                                                                                                                  |
| <b>Uses</b>                | Data Mining is more often used in research field such as web mining, text mining, fraud detection                                                                                                                                                                                                  | Machine learning has more uses in making recommendations of products, prices, estimating the time required for delivery etc.                                                                                                                                                                                 |
| <b>Concept</b>             | The concept behind mining is to extract information using techniques and find out the trends and patterns.                                                                                                                                                                                         | Machine Learning runs on the concept that machines learns from existing data and learns and improves by itself. Machine learning uses data mining methods and algorithms to build models on logic behind data which predict the future outcome. The algorithms are built on Math's and programming languages |
| <b>Nature</b>              | Data mining requires human intervention for applying techniques to extract information.                                                                                                                                                                                                            | Machine Learning is different from Data Mining as machine learning learns automatically.                                                                                                                                                                                                                     |
| <b>Learning Capability</b> | Data Mining requires the analysis to be initiated by human thus it is a manual technique.                                                                                                                                                                                                          | Machine Learning is a step ahead of data mining as it uses the same techniques used by data mining to automatically learn and adapt to changes. It is more accurate then data mining.                                                                                                                        |
| <b>Examples</b>            | At places where data mining is used is in identifying sales patterns or trends, by cellular companies for customer retention and so on.                                                                                                                                                            | Machine learning is used in running marketing campaigns, for medical diagnosis, image recognition etc.                                                                                                                                                                                                       |

**NOTES**

**Check Your Progress**

3. On what concept does machine learning run?
4. What is the concept of data mining?



## NOTES

---

### 8.3 ANSWERS TO CHECK YOUR PROGRESS QUESTIONS

---

1. The algorithms in the machine learning are different from any traditional program in the sense that in a traditional algorithm, the data and inputs are provided to get the desired output.
2. Reinforcement learning instead of working on the labelled training data works interactively with the environment in which it is working. Therefore, it helps the machine to learn based on its progress.
3. Machine learning runs on the concept that machines learn from existing data and learn and improve by itself.
4. Data mining is used to find out how different attributes of a data set are related to each other through patterns and data visualization techniques.

---

### 8.4 SUMMARY

---

- The ability of the machines to learn to perform such task are said to have an artificial intelligence and this type of learning is known as machine learning.
- The algorithms in the machine learning are different from any traditional program in the sense that in a traditional algorithm, the data and inputs are provided to get the desired output.
- The machine learning process is divided into two phases: training phase and testing phase.
- Steps involved in a machine learning process:
  - (i) Data collection
  - (ii) Data Pre – processing
  - (iii) Choose a model
  - (iv) Train the model
  - (v) Evaluate the model
  - (vi) Parameter Tuning
  - (vii) Prediction
- There are different ways in making a machine to learn. Three of the most important ways are:
  - (i) **Supervised Learning:** Machine learning means the computer can automatically learn to recognize hidden pattern in the given data and therefore helps to make better decisions.
  - (ii) **Unsupervised Learning:** Unsupervised learning, as the name suggests, is the training the machine without any supervision. In this

type of training, no information about the class or the labels is provided, only the training data is passed for the training of machine.

**(iii) Reinforcement Learning:** Reinforcement learning instead of working on the labeled training data works interactively with the environment in which it is working. Therefore, it helps the machine to learn based on its progress.

- In supervised learning, there is an external supervisor that helps the machine to learn but this supervisor is not the part of unsupervised learning. Here, the machine has the ability to interpret patterns and learning on its own with the help of input data.
- In reinforcement learning the machine decides its next action based on the reward in the current step which is always maximized.
- The area or the scope of machine learning is vast and it is increasing day by day. Some of the applications that would help in appreciating its scope in modern world are as follows:
  - (i) Image Recognition
  - (ii) Speech Recognition
  - (iii) Medical Diagnosis
  - (iv) Making Recommendations
  - (v) Fraud Detection
- Data mining can be used for varied purposes like prediction financial health of the organizations given the financial data for past few years and data outside the organization's environment. This might be useful for bigger organizations who want to make the funding related decision by comparing these predictions.
- Machine learning on the other hand apart from predicting the future trends and decisions, helps in the process of learning by applying new algorithms. It provides a foundation necessary for a machine to teach itself.
- Data mining requires the analysis to be initiated by human thus it is a manual technique.
- Machine learning is a step ahead of data mining as it uses the same techniques used by data mining to automatically learn and adapt to changes. It is more accurate than data mining.

## NOTES

---

### 8.5 KEY WORDS

---

- **Machine Learning:** It is an application of artificial intelligence (AI) that provides systems the ability to automatically learn and improve from experience without being explicitly programmed.

## NOTES

- **Supervised Learning:** It is the data mining task of inferring a function from labeled training data. The training data consist of a set of training examples.
- **Inference:** A conclusion reached on the basis of evidence and reasoning.
- **Iteratively:** A procedure in which repetition of a sequence of operations yields results successively closer to a desired results.

---

## 8.6 SELF ASSESSMENT QUESTIONS AND EXERCISES

---

### Short Answer Questions

1. What is artificial intelligence?
2. How are machines taught to learn?
3. Which are the two phases of machine learning process?
4. List the various steps involved in machine learning process.
5. What do you understand by unsupervised learning?

### Long Answer Questions

1. Explain the conversion of digitally converted text with the help of an example.
2. Describe the Steps involved in a machine learning process.
3. What are the types of machine learning processes? Explain each process, using diagrams wherever required.
4. What are the most commonly used machine learning applications?
5. Differentiate between machine learning and data mining.

---

## 8.7 FURTHER READINGS

---

Han, Jiawei, Micheline Kamber and Jian Pei. 2011. *Data Mining: Concepts and Techniques*, 2nd edition. Amsterdam: Elsevier.

Pujari, Arun K. 2010. *Data Mining Techniques*, 2nd edition. United States: Universities Press.

Anahory, Sam and Dennis Murray. 1997. *Data Warehousing in the Real World: A Practical Guide for Building Decision Support Systems*, 1st edition. Boston: Addison Wesley.

Witten, I. H. and Eibe Frank. 2005. *Data Mining: Practical Machine Learning Tools and Techniques*, 2nd edition. Amsterdam: Elsevier.

Soman, K. P., Shyam Diwakar and V. Ajay. 2006. *Insight Into Data Mining: Theory and Practice*. New Delhi: PHI.

---

## UNIT 9 NEURAL NETWORKS

---

### Structure

- 9.0 Introduction
- 9.1 Objectives
- 9.2 Introduction to Neural Systems and Networks
  - 9.2.1 A Brief History of ANN
  - 9.2.2 Biological Neuron
  - 9.2.3 ANN versus BNN
  - 9.2.4 Model of Artificial Neural Network
  - 9.2.5 Uses of Neural networks
- 9.3 Genetic Algorithm
  - 9.3.1 GA Operators
  - 9.3.2 Genetic Algorithm Framework
- 9.4 Answers to Check Your Progress Questions
- 9.5 Summary
- 9.6 Key Words
- 9.7 Self Assessment Questions and Exercises
- 9.8 Further Readings

### NOTES

---

## 9.0 INTRODUCTION

---

The power and speed of present day advanced PCs is genuinely surprising. No human can ever plan to figure a million tasks every second. On the other hand, there are a few undertakings for which even the most dominant PCs can't fight over the human brain, maybe not even with the aptitude of an earthworm. Estimate the intensity of the machine which has the capacities of the two PCs and humans. It would be the most surprising thing ever. And all people can live cheerfully ever after (or will they?). This is the point of artificial as a rule.

Neural Networks approaches this issue by proving to represent the structure and part of our nervous system. Numerous scientists trust that AI (Artificial Intelligence) and neural systems are direct inverse in their methodology. Regular AI depends on the image system hypothesis. Freely, an image system consists of unbreakable elements called images, which can shape more complex elements, by straightforward rules. The theory at that point expresses that such a system is able to do and is important for intelligence.

---

## 9.1 OBJECTIVES

---

After going through this unit, you will be able to:

- Understand the neural systems and networks
- Discuss the history of ANN

- Explain genetic algorithm
- Explain the genetic algorithm framework

## NOTES

---

## 9.2 INTRODUCTION TO NEURAL SYSTEMS AND NETWORKS

---

Neural systems are parallel calculating gadgets, which is fundamentally an experiment to make a PC model of the mind. The main aim is to build up a framework to perform different computational problems quicker than the traditional frameworks. These problems include pattern recognition and grouping, guess, improvement, and information clustering.

Artificial Neural Network (ANN) is an effective computing system whose focal topic is obtained from the similarity of biological neural systems. ANNs are additionally named as “artificial neural frameworks,” or “parallel distributed processing frameworks,” or “connectionist frameworks.” ANN obtains a huge gathering of units that are interconnected in some pattern to permit correspondence between the units. These units, likewise referred to as hubs or neurons, are basic processors which work in parallel.

Each neuron is connected with other neuron through a connection interface. Every connection interface is related with a weight that has data about the information signal. This is the most helpful data for neurons to take care of a specific problem because the weight normally energizes or represses the signal that is being conveyed. Every neuron has an inside state, which is called an initiation signal. Output signals, which are created subsequent to joining the input signals and activation rule, might be sent to different units.

### 9.2.1 A Brief History of ANN

The historical backdrop of ANN can be partitioned into the following three areas:

#### ANN during 1940s to 1960s

Some key advancements of this time are as follows:

- **1943:** It has been expected that the idea of neural system began with crafted by physiologist, Warren McCulloch, and mathematician, Walter Pitts, when in 1943 they displayed a straightforward neural system utilizing electrical circuits so as to depict how neurons in the brain may function.
- **1949:** Donald Hebb’s book, *The Organization of Behavior*, set forth the way that repeated initiation of one neuron by another expands its quality each time they are used.
- **1956:** An acquainted memory network was presented by Taylor.
- **1958:** A learning strategy for McCulloch and Pitts neuron model named Perceptron was designed by Rosenblatt.

- **1960:** Bernard Widrow and Marcian Hoff created models called “ADALINE” and “MADALINE.”

### ANN during 1960s to 1980s

Some key advancements of this period are as follows:

- **1961:** Rosenblatt made a unsuccessful trial however proposed the “backpropagation” scheme for multilayer systems.
- **1964:** Taylor built a victor take-all circuit with hindrances among output units.
- **1969:** Multilayer perceptron (MLP) was developed by Minsky and Papert.
- **1971:** Kohonen created Associative awareness.
- **1976:** Stephen Grossberg and Gail Carpenter created Adaptive resonance hypothesis.

### ANN from 1980s- Till Present

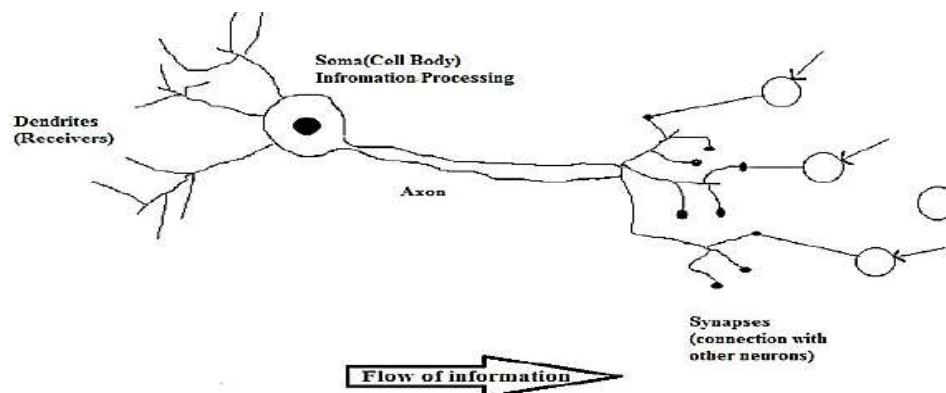
Some key advancements of this time are as follows:

- **1982:** The significant development was Hopfield’s Energy approach.
- **1985:** Boltzmann machine was created by Ackley, Hinton, and Sejnowski.
- **1986:** Rumelhart, Hinton, and Williams presented Generalized Delta Rule.
- **1988:** Kosko created Binary Associative Memory (BAM) and furthermore gave the idea of Fuzzy Logic in ANN.

The historical survey demonstrates that critical advancement has been made in this field. Neural system based chips are rising and applications to complex issues are being created. Without a doubt, today is a time of progress for neural system technology.

#### 9.2.2 Biological Neuron

A nerve cell (neuron) is an uncommon natural cell that processes data. As indicated by estimation, there are large numbers of neurons, roughly  $10^{11}$  with various interconnections, around  $10^{15}$ .



*Fig. 9.1 Working of a Biological Neuron*

### NOTES

**NOTES**

As appeared in the above figure, a typical neuron comprises of the following four sections with the help of which we can clarify its working.

- **Dendrites:** They are tree-like branches, obligated for accepting the data from different neurons it is associated with. In other sense, we can say that they resemble the ears of neuron.
- **Soma:** It is the cell body of the neuron and is obligated for preparing of data, they have gotten from dendrites.
- **Axon:** It is much the same as a link through which neurons send the data.
- **Synapses:** It is the association between the axon and other neuron dendrites.

**9.2.3 ANN versus BNN**

Prior to investigating the contrasts between Artificial Neural Network (ANN) and Biological Neural Network (BNN), let us investigate the similarities dependent on the terminology between these two.

*Table 9.1 Similarities between BNN and ANN*

| Biological Neural Network (BNN) | Artificial Neural Network (ANN) |
|---------------------------------|---------------------------------|
| Soma                            | Node                            |
| Dendrites                       | Input                           |
| Synapse                         | Weights or Interconnections     |
| Axon                            | Output                          |

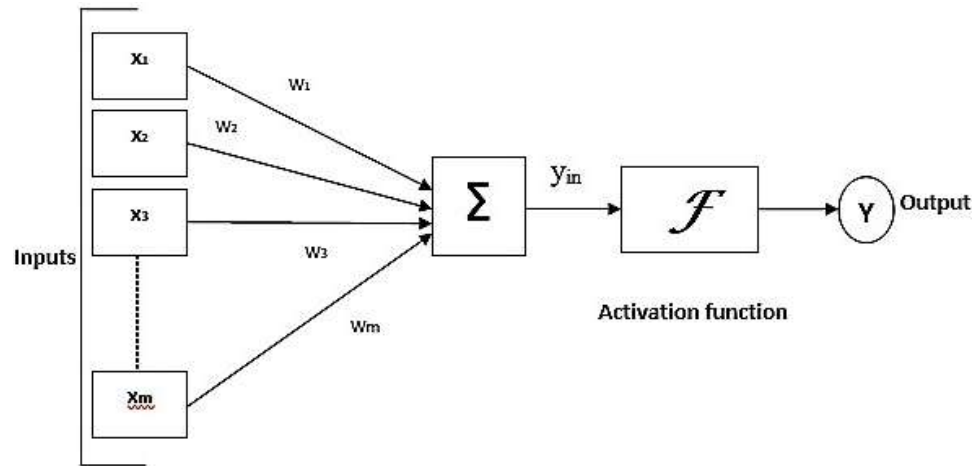
The following table shows the comparison between ANN and BNN based on some criteria mentioned.

*Table 9.2 Criteria Based Differentiation between BNN and ANN*

| Criteria                | BNN                                              | ANN                                                                                     |
|-------------------------|--------------------------------------------------|-----------------------------------------------------------------------------------------|
| <b>Processing</b>       | Massively parallel, slow but superior than ANN   | Massively parallel, fast but inferior than BNN                                          |
| <b>Size</b>             | $10^{11}$ neurons and $10^{15}$ interconnections | $10^2$ to $10^4$ nodes (mainly depends on the type of application and network designer) |
| <b>Learning</b>         | These type of network can tolerate ambiguity     | Very precise, structured and formatted data is required to tolerate ambiguity           |
| <b>Fault tolerance</b>  | Performance degrades with even partial damage    | It is capable of robust performance, hence has the potential to be fault tolerant       |
| <b>Storage capacity</b> | Stores the information in the synapse            | Stores the information in continuous memory locations                                   |

## 9.2.4 Model of Artificial Neural Network

The following diagram shows the general model of ANN followed by its processing.



*Fig. 9.2 General Model of ANN*

For the above general model of artificial neural network, the net input can be calculated as follows:

$$y_{in} = x_1 \cdot w_1 + x_2 \cdot w_2 + x_3 \cdot w_3 \dots x_m \cdot w_m$$

$$\text{i.e., Net input } y_{in} = \sum_i^m x_i \cdot w_i$$

The output can be calculated by applying the activation function over the net input.

$$Y = F(y_{in})$$

Output = function (net input calculated)

The general conviction is that Neural Networks is a sub-representative science. Before images themselves are perceived, something must be done as such that traditional AI would then be able to control those images. To make this point confirm, consider images, for example, dairy animals, grass, house and so on. When these images and the “straightforward principles” which govern them are known, traditional AI can perform wonders. Yet, to find that something is a cow isn’t minor. It should maybe be possible utilizing traditional AI and images, for example, - white, legs, and so forth. In any case, it would be tedious and absolutely, when you see a dairy animals, you immediately remember it to be in this way, without counting its legs.

In any case, this conviction - that AI and Neural Networks are direct inverse, isn’t substantial in light of the fact that, notwithstanding when you perceive a cow, it is a result of specific properties which you watch, that you infer that it is a dairy animals. This happens in a split second on the grounds that different pieces of the brain work in parallel. Every one of the properties which you watch is “summed

## NOTES



## NOTES

up”. Absolutely there are images here and rules - “summing up”. The main contrast is that in AI, images are carefully inseparable, though here, the images (properties) may happen with changing degrees or powers.

Advancement here can be made just by breaking this line of refinement among AI and Neural Networks, and combining the outcomes acquired in both, towards a bound together structure.

### 9.2.5 Uses of Neural networks

The neural networks are used in many ways because of their some of wonderful properties. The major applications of neural networks in generalization, predictions, forecasting, feature extraction, pattern recognition etc. are discussed below:

#### Generalization, Prediction and Forecasting

The neural networks learn from the initial inputs and their relationships. Hence, they can infer unseen relationships on unseen data. The generated model can now generalize and predict on unseen data. Also not like other prediction models, they have the ability to learn hidden relationships in the data without imposing any fixed relationships in the data. Due to this property they are very useful in time series forecasting where data volatility is very high.

From above, it is very clear that they can be used in:

- 1. Financial Time Series Forecasting-** In stock market there are many factors responsible in whether a given stock will go up or down on any given day. Since neural networks can examine much information very fast and can provide solution to that.
- 2. Traveling Salesman Problem-** neural networks can solve the traveling salesman problem, but only to a certain degree of approximation.
- 3. Loan Applications** -These are some applications, with the acceptance of a neural network that will decide whether or not to grant a loan etc.

#### Feature Extraction and Pattern Recognition

The neural networks have the ability to take in a lot of inputs, process them to infer hidden as well as complex, non-linear relationships. Due to which they play big role in feature extraction and pattern recognition.

From above, it is very clear that they can be used in:

- 1. Character Recognition** – Neural networks pattern recognition ability helps in character recognition and handwriting identification has lot of applications in fraud detection (e.g. bank fraud) and even in national security assessments
- 2. Image Processing and Recognition** - Image recognition is feasible because of neural networks which helps in facial recognition in social media,

Fingers print detection, cancer detection in medicine and satellite imagery processing for agricultural and defense usage.

3. **Speech Recognition** - application in national security, armature prevention from wrong use.
4. **Miscellaneous Applications** - computer vision, natural language processing etc. example- self driving cars.

## NOTES

### Check Your Progress

1. What is a nerve cell?
2. What are dendrites?

## 9.3 GENETIC ALGORITHM

There is no known polynomial time to solve many real worlds' optimization problems making them hard to solve. A number of heuristics solve these problems which are providing sub optimal but acceptable solution in a computational time. Genetic algorithm is one of the computing paradigms which are basically used for solving optimization problem. It is frequently used to find optimal or near-optimal solutions to difficult problems which otherwise would take a lifetime to solve. It is frequently used to solve optimization problems, in research, and in machine learning.

When the two biological processes namely genetics and evolution are used to solve optimization problem in the form of evolutionary algorithm is called the genetic algorithm. Based on this concept, we can say that a genetic algorithm is a population based probabilistic search and optimization technique, which works based on the mechanism of natural genetics and natural evaluation.

### Optimization Problem Solving with GA

For optimization problem identify the following:

- **Objective Functions:** an optimization problem is specified by means of objective function and Objective function in fact, defined in terms of some input parameters and these are the parameters whose values decides the value of the objective function.
- **Constraints:** an optimization problem is specified by means of objective function, and then constraints. The constraints are basically the requirement, by which all parameters values should satisfy so, that the optimization function will get it is value.
- **Fitness Evaluation:** for every solution we have to calculate some fitness value. That means, if say suppose solution is the optimum solution one or global solution one, then it should have the highest fitness value.

## NOTES

- **Encoding:** It is nothing but representation of chromosome for a solution, and one solution we can consider an individual. Now chromosome is basically encoded form; that means some symbolic representation.
- **Decoding:** It is the reverse process of encoding where the actual values are decoded from the encoded chromosomes.

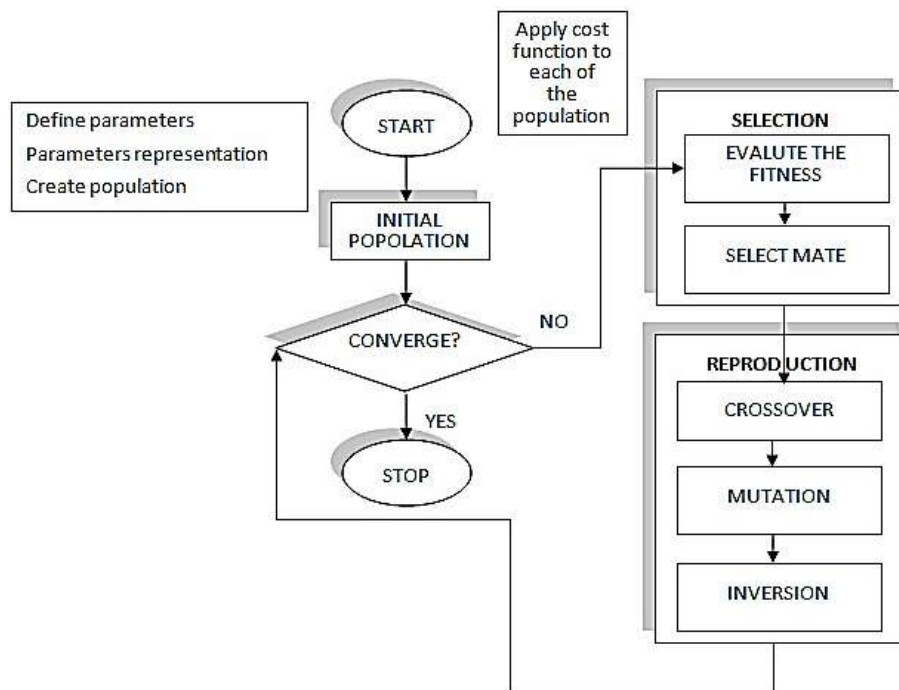
### 9.3.1 GA Operators

A GA implementation is involved with the realization of the following operations:

- **Encoding:** It is used to represent a solution to fit with the GA framework
- **Convergence:** it is used to decide the termination criterion. It is basically if we get a population or a generation, it will check whether we have reached to the termination condition or not, then you have to stop it or you have to continue the process of the genetic algorithm.
- **Mating pool:** It is a process for generating the next solutions. It is a function or it is an operator, by which input is a population and it produced the output as the pools.
- **Fitness evaluation:** it is basically function or procedure to evaluate the solution.
- **Crossover:** It is basically about to make the diverse set of solutions. If we pass two chromosomes to this operator, it will produce two or more offspring. Crossover is usually applied in a GA with a high probability.
- **Mutations:** It is used to explore other solutions. It is a small random tweak in the chromosome, to get a new solution. It is used to maintain and introduce diversity in the genetic population and is usually applied with a low probability. If the probability is very high, the GA gets reduced to a random search. It has been observed that mutation is essential to the convergence of the GA.
- **Inversion:** It is basically to jump from one optimum value to another optimum value.

### 9.3.2 Genetic Algorithm Framework

Genetic Algorithms (GA) are adaptive algorithms derived from Darwin's principal of survival of the fittest in natural genetics and concept of genetics proposed by Gregory Johan Mandala. GA maintains a population of potential solutions of the candidate problem termed as individuals and by manipulating these individuals using GA operators like selection, crossover and mutation, a GA evolves better solutions over a number of generations. The working framework of a GA is shown as flowchart in Figure 9.3.



*Fig. 9.3 Flow Chart of a Genetic Algorithm*

A pseudo-code for a GA is explained below:

#### GA ()

- (1) Initialize population
- (2) Find fitness of population
- (3) While (termination criteria is reached) do
  - Parent selection
  - Crossover with probability  $p_c$
  - Mutation with probability  $p_m$
  - Decode and fitness calculation
  - Survivor selection
  - Find best
- (4) Return best

#### Check Your Progress

3. Define genetic algorithm.
4. What is fitness value in GA?

#### NOTES

---

## 9.4 ANSWERS TO CHECK YOUR PROGRESS QUESTIONS

---

### NOTES

1. A nerve cell (neuron) is an uncommon natural cell that processes data. As indicated by estimation, there are large numbers of neurons, roughly  $10^{11}$  with various interconnections, around  $10^{15}$ .
2. Dendrites are tree-like branches, obligated for accepting the data from different neurons it is associated with. In other sense, we can say that they resemble the ears of neuron.
3. Genetic algorithm is one of the computing paradigms which are basically used for solving optimization problem.
4. For every solution we have to calculate some fitness value. That means, if say suppose solution is the optimum solution one or global solution one, then it should have the highest fitness value.

---

## 9.5 SUMMARY

---

- Neural systems are parallel calculating gadgets, which is fundamentally an experiment to make a PC model of the mind. The main aim is to build up a framework to perform different computational problems quicker than the traditional frameworks.
- Artificial Neural Network (ANN) is an effective computing system whose focal topic is obtained from the similarity of biological neural systems. ANNs are additionally named as “artificial neural frameworks,” or “parallel distributed processing frameworks,” or “connectionist frameworks.”
- Each neuron is connected with other neuron through a connection interface. Every connection interface is related with a weight that has data about the information signal.
- The historical survey demonstrates that critical advancement has been made in this field. Neural system based chips are rising and applications to complex issues are being created.
- A typical neuron comprises of the following four sections with the help of which we can clarify its working:
  - (i) Dendrites
  - (ii) Soma
  - (iii) Axon
  - (iv) Synapse

- For general model of artificial neural network, the net input can be calculated as follows:

$$y_{in} = x_1 \cdot w_1 + x_2 \cdot w_2 + x_3 \cdot w_3 \dots x_m \cdot w_m$$

i.e., Net input  $y_{in} = \sum_i^m x_i \cdot w_i$

The output can be calculated by applying the activation function over the net input.

$$Y = F(y_{in})$$

Output = function (net input calculated)

- Uses of Neural Networks:
  - A. Generalization, Prediction, Forecasting:
    - (i) Financial Time Series Forecasting-
    - (ii) Traveling Salesman Problem
    - (iii) Loan Applications
  - B. Feature Extraction, Pattern Recognition:
    - (i) Character recognition
    - (ii) Image processing and recognition
    - (iii) Speech recognition
    - (iv) Miscellaneous Applications
- Genetic algorithm is one of the computing paradigms which are basically used for solving optimization problem. It is frequently used to find optimal or near-optimal solutions to difficult problems which otherwise would take a lifetime to solve.
- For optimization problem identify the following:
  - (i) Objective Functions
  - (ii) Constraints
  - (iii) Fitness Evaluation
  - (iv) Encoding
  - (v) Decoding
- Genetic Algorithms (GA) are adaptive algorithms derived from Darwin's principal of survival of the fittest in natural genetics and concept of genetics proposed by Gregory Johan Mandala.

## NOTES

---

## 9.6 KEY WORDS

---

- **Artificial Neural Networks or Connectionist Systems:** These are computing systems that are inspired by, but not necessarily identical to, the biological neural networks that constitute animal brains.

- **Pseudo-Code:** A notation resembling a simplified programming language, used in program design.

## NOTES

---

### 9.7 SELF ASSESSMENT QUESTIONS AND EXERCISES

---

#### Short Answer Questions

1. What are neural systems?
2. What is ANN?
3. How can we calculate the net input for a general model of artificial neural network?
4. Write a pseudo-code for a GA.

#### Long Answer Questions

1. What are the factors that need to be identified for optimization problem?
2. Explain the genetic algorithm framework with the help of a diagram.
3. List and explain the GA operators.
4. Discuss the history of ANN from 1940's to present.

---

### 9.8 FURTHER READINGS

---

Han, Jiawei, Micheline Kamber and Jian Pei. 2011. *Data Mining: Concepts and Techniques*, 2nd edition. Amsterdam: Elsevier.

Pujari, Arun K. 2010. *Data Mining Techniques*, 2nd edition. United States: Universities Press.

Anahory, Sam and Dennis Murray. 1997. *Data Warehousing in the Real World: A Practical Guide for Building Decision Support Systems*, 1st edition. Boston: Addison Wesley.

Witten, I. H. and Eibe Frank. 2005. *Data Mining: Practical Machine Learning Tools and Techniques*, 2nd edition. Amsterdam: Elsevier.

Soman, K. P., Shyam Diwakar and V. Ajay. 2006. *Insight Into Data Mining: Theory and Practice*. New Delhi: PHI.

---

**BLOCK - IV**  
**WEB MINING AND VISUAL DATA MINING**

---

*Introduction to Web  
Mining*

---

**UNIT 10 INTRODUCTION TO WEB  
MINING**

---

**NOTES**

**Structure**

- 10.0 Introduction
- 10.1 Objectives
- 10.2 Web Content Mining
  - 10.2.1 Mining the Web Page Layout Structure
  - 10.2.2 Web Page Mining
- 10.3 Text Mining
- 10.4 Text Clustering
- 10.5 Temporal Mining
- 10.6 Spatial Data Mining
- 10.7 Answers to Check Your Progress Questions
- 10.8 Summary
- 10.9 Key Words
- 10.10 Self Assessment Questions and Exercises
- 10.11 Further Readings

---

**10.0 INTRODUCTION**

---

In this unit, you will learn about the web mining, text mining, temporal mining and data mining. Web mining is the application of data mining techniques to discover patterns from the World Wide Web. Text mining describes data in text format rather than in the structural form. It extracts patterns and associations from large databases. Moreover, you will learn about problems which occur while in mining complex data. Spatial data mining is the branch of data mining that deals with spatial, i.e., geographical and geo-referenced data.

---

**10.1 OBJECTIVES**

---

After going through this unit, you will be able to:

- Explain the different types of web mining
- Discuss the pros and cons of web mining
- Identify the problems in mining complex data
- Define text and spatial data mining
- Explain text clustering and temporal mining
- Explain spatial clustering methods



## NOTES

---

## 10.2 WEB CONTENT MINING

---

The World Wide Web serves as a huge, widely distributed, global information service center for news, advertisements, consumer information, financial management, education, government, e-commerce, and many other information services. The Web also contains a rich and dynamic collection of hyperlink information and Web page access and usage information, providing rich sources for data mining. However, based on the following observations, the Web also poses great challenges for effective resource and knowledge discovery.

- The Web seems to be too huge for effective data warehousing and data mining. The size of the Web is in the order of hundreds of terabytes and is still growing rapidly. Many organizations and societies place most of their public-accessible information on the Web. It is barely possible to set up a data warehouse to replicate, store, or integrate all of the data on the Web.
- The complexity of Web pages is far greater than that of any traditional text document collection. Web pages lack a unifying structure. They contain far more authoring style and content variations than any set of books or other traditional text-based documents. The Web is considered a huge digital library; however, the tremendous number of documents in this library is not arranged according to any particular sorted order. There is no index by category, nor by title, author, cover page, table of contents, and so on. It can be very challenging to search for the information you desire in such a library!
- The Web is a highly dynamic information source. Not only does the Web grow rapidly, but its information is also constantly updated. News, stock markets, weather, sports, shopping, company advertisements, and numerous other Web pages are updated regularly on the Web. Linkage information and access records are also updated frequently.
- The Web serves a broad diversity of user communities. The Internet currently connects more than 100 million workstations, and its user community is still rapidly expanding. Users may have very different backgrounds, interests, and usage purposes. Most users may not have good knowledge of the structure of the information network and may not be aware of the heavy cost of a particular search. They can easily get lost by groping in the “darkness” of the network, or become bored by taking many access “hops” and waiting impatiently for a piece of information.
- Only a small portion of the information on the Web is truly relevant or useful. It is said that 99% of the Web information is useless to 99% of Web users. Although this may not seem obvious, it is true that a particular person is generally interested in only a tiny portion of the Web, while the rest of the Web contains information that is uninteresting to the user and may swamp desired search results.

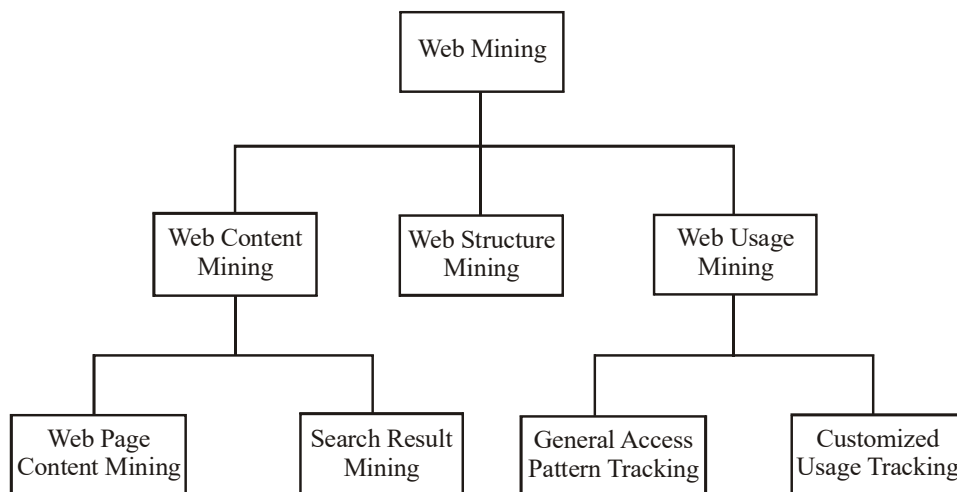
These challenges have promoted research into efficient and effective discovery and use of resources on the Internet.

The Web is accessed for:

- (i) Finding relevant information.
- (ii) Discovering new knowledge.
- (iii) Creating personalized Web page.
- (iv) Learning about individual users.

Web mining can be categorized as shown in the following Figure 10.1.

- (i) Web content mining, which describes the discovery of useful information from the Web contents.
- (ii) Web structure mining, which is concerned with discovering the model underlying the link structures of the Web.
- (iii) Web usage mining which deals with the study of data generated by the users and their usage characteristics.



*Fig. 10.1 Types of Web Mining*

### 10.2.1 Mining the Web Page Layout Structure

Web page mining is the mining of data related to the World Wide Web and stored in the Web pages. The data referred is the one present in the Web pages or related to the Web activity. It can be classified as:

- Content of actual Web pages.
- Intrapage structure including HTML (Hypertext Markup Language) or XML (eXtensible Markup Language) code for the page.
- Interpage structure providing a link between the Web pages.
- User profiles.

## NOTES

## NOTES

### Mining the Web's Link Structure to Identify Authoritative Web Pages

Web structure mining is concerned with discovering the model underlying the link structure of the Web. This model helps us in categorizing Web pages and generating information about similarities and relationships between different Websites. It is used to find authority Websites and overview sites for the subjects that point to many authorities. It explores the structure within a document (intra-document structure) and studies a structure of documents within the Web itself.

The concept of PageRank has been introduced to measure the back links or count the citations of a given document. It indicates the importance or quality of the document. A PageRank is defined by the number of links going out of the page.

Links have also been categorized as a transverse link if it is between the pages with different domain names and an intrinsic link if it is between the pages with the same domain name. Domain name here refers to the first level in the URL string associated with a page.

#### 10.2.2 Web Page Mining

Web page mining can be considered as a Web page content mining and an extension of the work done by the search engines. There are different techniques employed to search the Internet. While most of the search engines are keyword based, they can in general be divided into either agent based or database-oriented search engines. In agent-based systems, software systems perform the data mining operation. The other approach considers the Web data as belonging to a database. Content mining is also similar to text mining in nature.

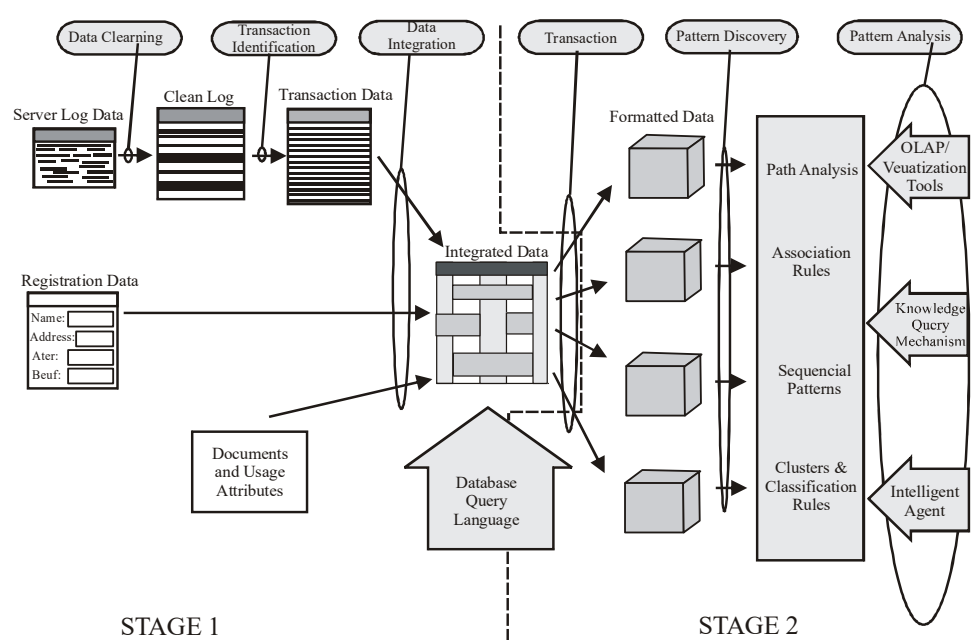
**Web Mining:** It is the application of data mining techniques to discover patterns from the Web. According to analysis targets, Web mining can be divided into three different types, which are Web usage mining, Web content mining and Web structure mining.

**Web Content Mining:** This type of mining takes text, image, audio or video data in the Web for discovering useful information. This is also called Web text mining. This is so because text content is the widest area of research. Natural Language Processing and Information Retrieval are the latest technologies, normally used in Web content mining.

#### *Web Mining Pros and Cons*

Web mining offers many advantages, making this technology attractive for big corporates as well as government agencies. Web mining enables personalized marketing, by introducing e-commerce that has resulted in higher trade volumes. Government agencies use it for classifying threats to fight terrorism. This has enabled companies to establish better customer relationship. Companies can explore avenues for expanding their business, attract and retain customers. They can make better analysis on the purchasing habits of customers and their satisfaction level.

**Cons:** Due to invasion of privacy, Web mining raises an ethical issue. Privacy is lost, since information related to an individual can be easily obtained and may be misused. The information so obtained can be analysed and then clustered to create profiles.



**NOTES**

*Fig. 10.2 Web Mining Architecture*

**Problems in Mining Complex Data**

Data mining, undoubtedly, is extremely useful. But it has its own challenges. Challenging problems related to mining of complex data are listed as follows:

- (i) Using complex data for mining complex knowledge.
- (ii) Mining distributed and multi-agent data.
- (iii) Scaling up high-dimensional data and high-speed data streams.
- (iv) Development of unifying theory of data mining.
- (v) Handling non-static, unbalanced and cost-sensitive data.
- (vi) Mining of time series and sequence data.
- (vii) Data mining in networks.
- (viii) Using data mining techniques for biological and environmental problems.
- (ix) Problems related to the process of data mining.
- (x) Security, privacy and data integrity.

**Check Your Progress**

1. Define web page mining.
2. What is web structure mining?

### 10.3 TEXT MINING

#### NOTES

Text mining describes data in text format rather than in the structural form. It extracts patterns and associations from large databases. Another feature is that text mining and information retrieval are different; text mining refers to data mining, whereas information retrieval refers to database management system.

Nowadays, there are many techniques for data mining work and tools developed for working with relational databases. Currently, there is no data mining tool which is applicable for text data. The following are different methods for mining unstructured data:

- Tagging techniques are used for extracting data from unstructured databases. On extraction, the data is in a structured database with the help of data mining tools its is converted to unstructured data (see Figure 10.3).

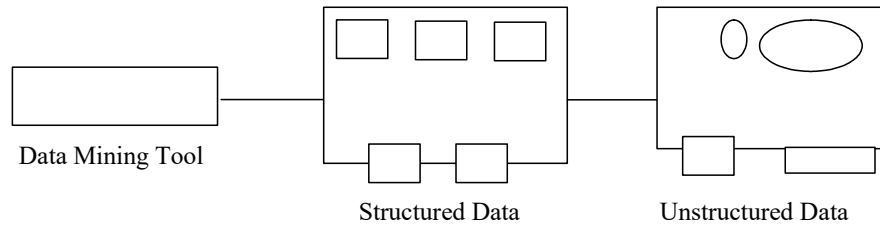


Fig. 10.3 Converting Unstructured Data to Structured Data

- Data preprocessing technique that merges the data from multiple heterogenous data sources into a coherent data store.

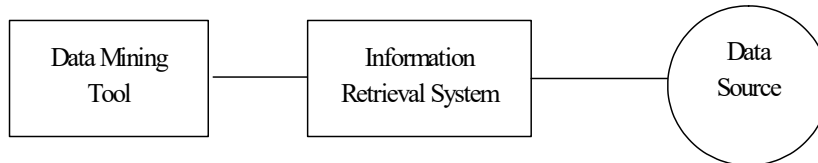


Fig. 10.4 Information Retrieval System

- We develop data mining tools in such a way that we may operate it directly on unstructured database (see Figure 10.5).

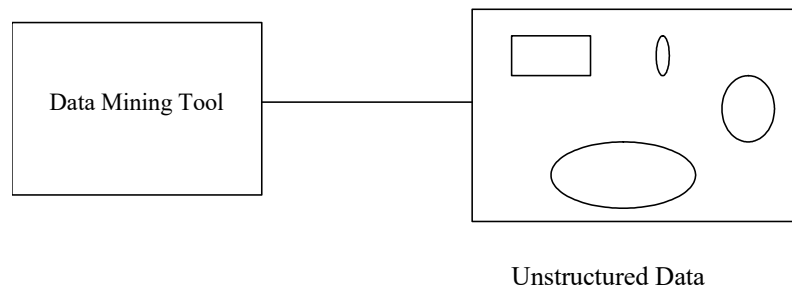


Fig. 10.5 Mining Unstructured Data

As we change the text data to relational data, we concentrate on the actual information, i.e., there is no loss of key information.

As we have discussed that data mining is applicable to structured relational and transactional data, the maximum textual data available pertains to semi-structured data which is available as text data from different sources, such as electronic publications and the World Wide Web. It may partially contain structured data, such as Title and Author's Name on one hand and unstructured text like Abstracts and Content (see Figure 10.6).

## NOTES

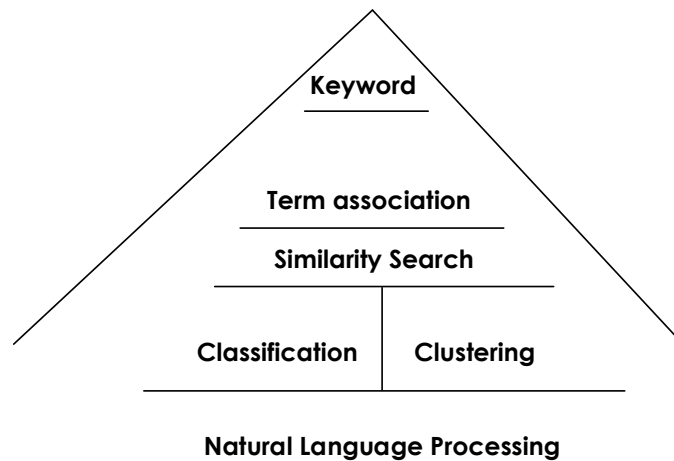


Fig. 10.6 Mining Semi-Structured Data

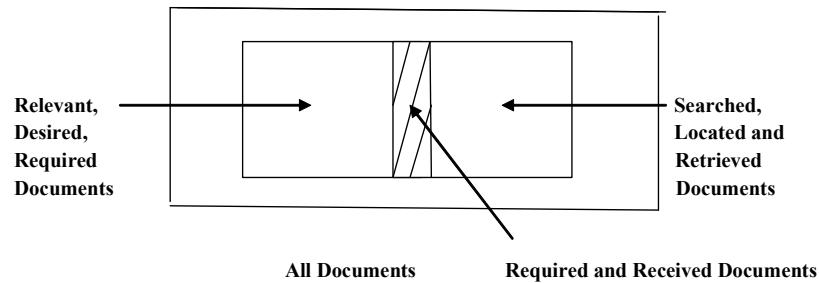
### Text Data Analysis and Information Retrieval

Information Retrieval (IR) is a field that has been developing in parallel with database systems for many years. Unlike the field of database systems, which has focused on query and transaction processing of structured data, information retrieval is concerned with the organization and retrieval of information from a large number of text-based documents. Since information retrieval and database systems each handle different kinds of data, some database system problems are usually not present in information retrieval systems, such as concurrency control, recovery, transaction management, and update. Also, some common information retrieval problems are usually not encountered in traditional database systems, such as unstructured documents, approximate search based on keywords, and the notion of relevance.

Due to the abundance of text information, information retrieval has found many applications. There exist many information retrieval systems, such as on-line library catalog systems, on-line document management systems, and the more recently developed Web search engines. A typical information retrieval problem is to locate relevant documents in a document collection based on a user's query, which is often some keywords describing an information need, although it could also be an example relevant document. In such a search problem, a user takes the

**NOTES**

initiative to “pull” the relevant information out from the collection; this is most appropriate when a user has some ad hoc (i.e., short-term) information need, such as finding information to buy a used car. When a user has a long-term information need (e.g., a researcher’s interests), a retrieval system may also take the initiative to “push” any newly arrived information item to a user if the item is judged as being relevant to the user’s information need. Such an information access process is called information filtering, and the corresponding systems are often called filtering systems or recommender systems. From a technical viewpoint, however, search and filtering share many common techniques.



**Fig. 10.7** Text Data Analysis

During the study of database systems the entire focus is on query and transaction systems based on OLTP. In the case of information retrieval, the intention is to organize information in the form of text-based documents and then retrieve them as per our requirement. Thus, information stored is searched for keywords or information documents. In information management, unstructured documents are encountered, while problems, such as transaction management and updating database systems are not encountered (see Figure 10.7).

**Measures for Text Retrieval**

**Precision:** This is the percentage of retrieved documents that are in fact relevant to the query (i.e., “correct” responses). It is formally defined as

$$\text{precision} = \frac{|\{\text{Relevant}\} \cap \{\text{Retrieved}\}|}{|\{\text{Retrieved}\}|}$$

**Recall:** This is the percentage of documents that are relevant to the query and were, in fact, retrieved. It is formally defined as

$$\text{Recall} = \frac{|\{\text{Relevant}\} \cap \{\text{Retrieved}\}|}{|\{\text{Relevant}\}|}$$

An information retrieval system often needs to trade off recall for precision or vice versa. One commonly used trade-off is the F-score, which is defined as the harmonic mean of recall and precision:

$$F\_score = \frac{\text{recall} \times \text{precision}}{(\text{recall} + \text{precision}) / 2}$$

The harmonic mean discourages a system that sacrifices one measure for another too drastically.

Precision, recall, and F-score are the basic measures of a retrieved set of documents. These three measures are not directly useful for comparing two ranked lists of documents because they are not sensitive to the internal ranking of the documents in a retrieved set. In order to measure the quality of a ranked list of documents, it is common to compute an average of precisions at all the ranks where a new relevant document is returned. It is also common to plot a graph of precisions at many different levels of recall; a higher curve represents a better-quality information retrieval system.

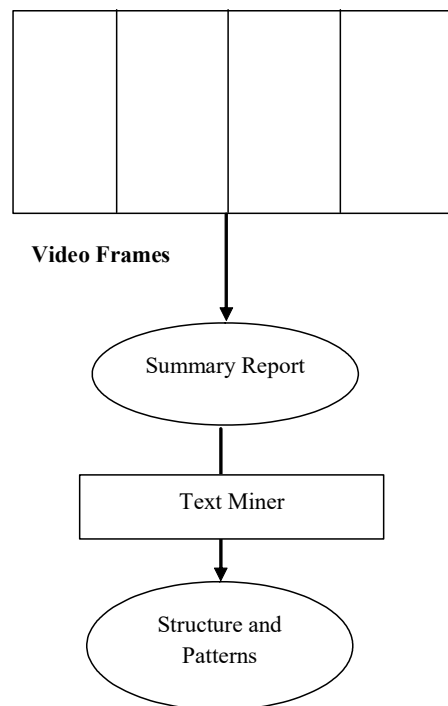
## NOTES

### Video Mining

In video mining, we find correlations and patterns from a large video database. In this mining, we analyse a video clipping or multiple video clippings and capture the text in video mining and this text data behaves as video data.

The goal of data mining is to discover interesting patterns. In video mining, we examine the patterns matching in a video database.

In video mining, we discover patterns in audio visual content through principal cast detection and also through location of significant parts of video sequence (see Figure 10.8).



**Fig. 10.8** Text Mining Extraction from Video Database



## NOTES

### Audio Mining

Audio mining is a technique for auto-analysing and searching for the content of an audio signal, used in automatic speech recognition. Musical audio mining is the identification of perception-based characteristics of a musical piece. This may be having a melody, or a rhythmic structure (see Figure 10.9).

**Inside Audio Mining:** Audio mining is also known as audio search. This takes queries to locate the term in an audio file. This work is done through audio indexing. It makes use of speech recognition for analysing the entire file so as to produce an index of content with words and their locations that can be searched.

**Research on Audio Mining:** Audio mining research work started in the 1970s at technical institutes in USA. Some of these institutions are, Carnegie Mellon University, Columbia University, The Georgia Institute of Technology and the University of Texas. By integrating audio mining products, larger systems have been created.

**Audio Mining Approaches:** In audio mining, two main approaches are adopted.

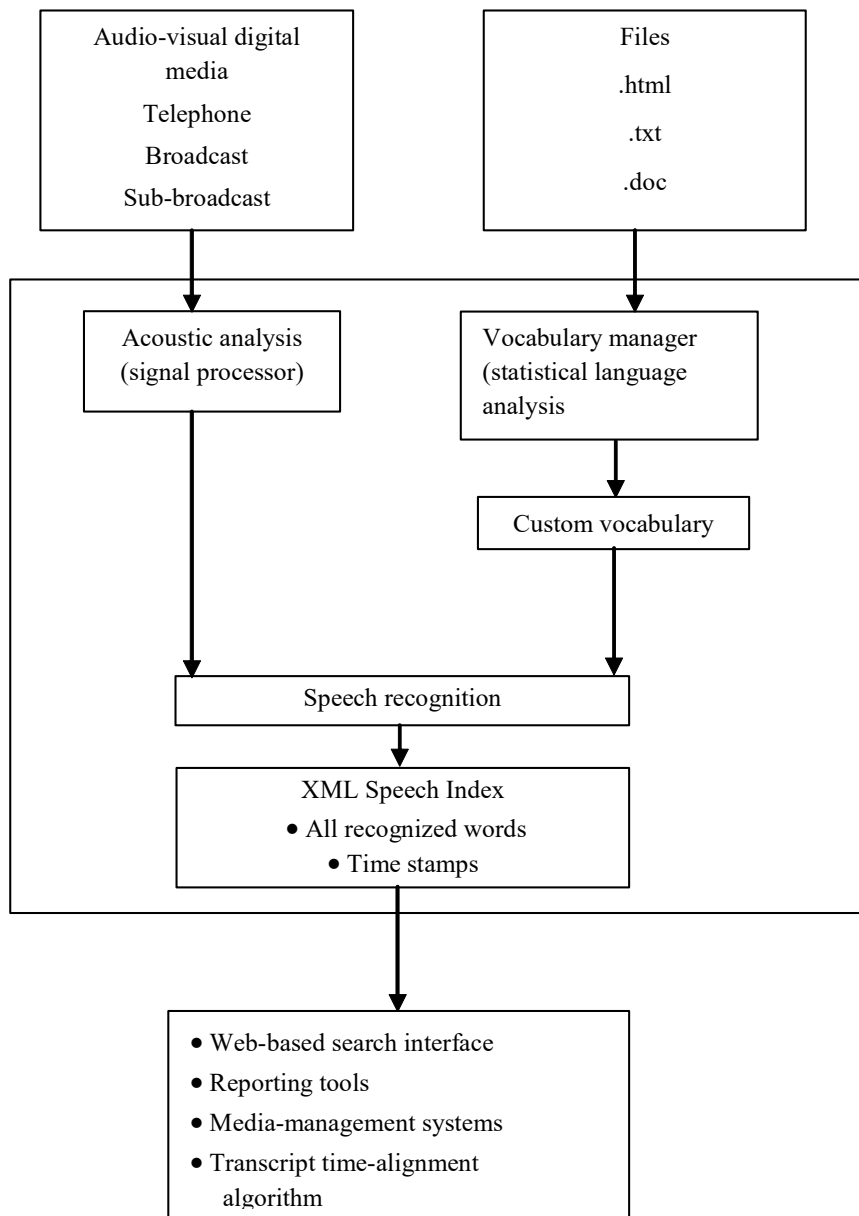
- (i) **Text Mining Indexing:** This mining does the work of conversion from speech to text, identifying words in a dictionary that may contain thousands of entries and is known as LVCSR (Large Vocabulary Continuous Speech Recognition). In case a word is not found in the dictionary, LVCSR system chooses a similar word. Language understanding is used by the system users.
- (ii) **Phoneme-Based Indexing:** Phoneme is the smallest unit of speech. This indexing works with sound. It does not convert speech to text. This uses sound and creates an index which is based on phonetics. It uses a dictionary containing many phonemes for converting the search term into the phoneme form. Phonetic systems need proprietary search tools to put the term as phonetic and then find a match for the existing string. This is a very complex operation.

Text-based approaches are better than phoneme-based approach because phoneme-based approach gives more false results.

### Working of Text and Video Approaches

Systems based on text and phonemes operate almost in the same way, but the difference is in the use of a text-based dictionary in the case of the former approach and a phonetic dictionary for the latter.

NOTES



**Fig. 10.9** Working of an Audio Mining System

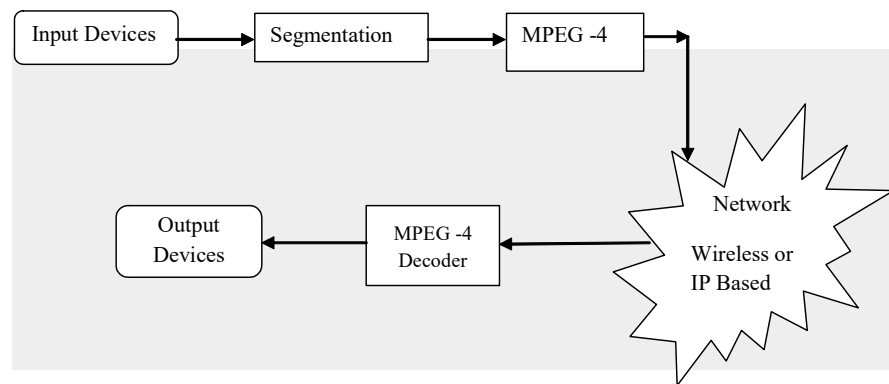
Generally, audio mining uses speech recognition. Speech recognition software is made up of acoustic models containing representation of all phonemes. These also contain a statistical language model indicating the way words follow each other in a language. In this technology, a speech signal of unknown content is taken and converted into a series of words referring to the program's dictionary. But with highly inflected languages, such as Chinese, this is not easy. Some audio mining dictionaries are domain specific, such as those for law and medicine.

## NOTES

### Performance

Nowadays, by using large processors, fast memories and high-computational algorithms, audio mining technology gives high levels of performance. For example, the latest technology adopted by Fast-Talk is capable of indexing an audio file of one-hour duration in only five minutes. It is capable of processing 30 hours of content in one second using a search query of 10-phoneme in systems running with 2.53-GHz Pentium CPU.

### Audio Video Processing



*Fig. 10.10 Audio Video Processing*

Most audio/video processing technology revolves around industry standards (see Figure 10.10). The most popular digital video compression is the Motion Picture Encoding Group (MPEG). Its digital video compression standards are:

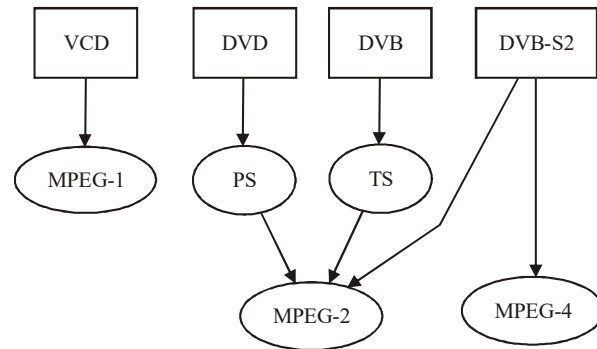
- MPEG – 1
- MPEG – 2
- MPEG – 4
- MPEG – 7

**MPEG -1:** It works on CDROM- based applications. Nowadays, it is the most widely compatible lossy audio/video format in the world and is used in a large number of products and technologies. One of the best examples of MPEG-1 standard is the MP3 audio format.

MPEG -1 standard contains five parts:

- (i) Systems (storage and synchronization of video, audio and other data together).
- (ii) Video (compressed video content).
- (iii) Audio (compressed audio content).
- (iv) Conformance testing (testing the correctness of implementations of the standard).
- (v) Reference software (example, software showing how to encode and decode according to the standard).

**MPEG-2:** It is a standard for ‘the generic coding of moving pictures and associated audio information’. It describes a combination of lossy video compression and lossy audio data compression methods which permit storage and transmission of movies using currently available storage media and transmission bandwidth (see Figure 10.11).



**Fig. 10.11** MPEG-2 used in Digital Video Broadcast

**MPEG-4:** It is a video compression standard that addresses object-based processing, blended of synthetic and natural videos. It involves low-bit rate transmission and was developed in 1998. It is named as Moving Picture Expert Group. It includes voice (Telephone, Videophone) and television- based applications. MPEG-4 Codec technology for encoder optimization and video segmentation also been developed. We also have a system development effort for wireless video streaming using the MPEG-4 standard.

The characteristics of MPEG -4 are as follows:

- Improved coding efficiency.
- Ability to encode mixed media data (video, audio, speech).
- Error resilience to enable robust transmission.
- Ability to interact with the audio-visual scenes generated at the receiver.

**MPEG-7:** It addresses standardization of multimedia content descriptions so as to enable applications, such as content-based remote and local browsing. It is also a multimedia content description standard. MPEG-7 is formally called *Multimedia Content Description Interface*. Thus, it is *not* a standard which deals with the actual encoding of moving pictures and audio, such as MPEG 1, MPEG 2 and MPEG 3. Research efforts include technology for video browsing, indexing, and summarization, as well as creating new MPEG-7-enabled product and service concepts.

### Dimensionality Reduction for Text

Dimensionality reduction means obtaining compressed or reduced data from the original data. In this process, data may be lost or remain lossless. If we reconstruct

## NOTES

## NOTES

the original data from the compressed data without any loss of information, the data reduction is called lossless and if we obtain the original data approximate to the original values then it is called lossy.

There are two popular methods of lossy dimensionality reduction. They are as follows:

- Wavelet transforms
- Principal components analysis

**Wavelet Transforms:** The Discrete Wavelet Transforms (DWT) is a linear signal processing technique. It is applied to a vector  $X$  which transforms to a numerically different vector  $X'$  of wavelet coefficients. The two vectors are of the same length. The DWT is closely related to Discrete Fourier Transform (DFT). It is a signal processing technique involving sines and cosines. Generally, DWT achieves better lossy compression.

Wavelet transforms can be applied to multidimensional data, such as a data cube. They are applied to each dimension, one at a time.

**Principal Components Analysis:** It searches for  $k$ ,  $n$ -dimensional orthogonal vectors that can be used to represent the data, where  $k \leq n$ . The original data is thus, projected into a small space resulting in dimensionality reduction. In other words, the initial data is projected into a smaller set of variables.

### Text Mining Approaches

There are two approaches to text mining.

- (i) **Keyword-Based Analysis:** Collect a set of keywords that occur together and then find the association or correlation between them.

The procedure consists of pre-processing, i.e., removing unwanted data, for example, removing parsing. Only after this essential step, association mining algorithms are applied.

A text database may be in the following form:

{document\_id, a\_set\_of\_words}

Association mining is applied to these sets of words. It also helps in identifying compound associations, such as 'perambur rail coach factory' or non-compound associations, such as 'Rupees, Total Cost'.

- (ii) **Document Classification Analysis:** Automated document classification helps us in organizing documents into classes to simplify their retrieval and analysis at a later stage. The steps involved are: take a pre-classified set as a training set, then analyse it to derive a classification scheme. The classification scheme thus, obtained is further utilized to classify other documents in an online mode.

### Check Your Progress

3. What do you understand by text mining?
4. What are the two approaches of text mining?

### NOTES

## 10.4 TEXT CLUSTERING

Clustering is a technique or task of finding a group of unlabelled objects so that the objects in a group will be related or similar to each other and will be unrelated or different to the objects in other groups. Alternately, clustering is like unsupervised classification where the algorithm models the similarities instead of boundaries.

Text clustering is one of the application of cluster analysis that uses machine learning and Natural Language Processing (NLP) to understand and categorized unstructured and textual data. That is, it is a task of regrouping similar text units within a collection of documents which is used in mining any text-based resource. Unlike text classification, this is an unsupervised task as we have no prior idea about the categories.

Depending upon the use case, clustering can be performed on either documents or sentences.

### Document Clustering

To start off, there are three requirements that need to be met:

- (i) A distance measure to identify if two texts are similar
- (ii) A criterion function to compute the quality of the clusters and
- (iii) An algorithm to optimize this criterion.

The distance measure helps in identifying the proximity of two points in our dataset. The distance is large if the two documents are different and is small if the two documents are similar. Criterion function will help us to identify the best cluster and stop the pipeline. And lastly an algorithm is required to optimize this criterion function. There could be many stages in the algorithm, but the common approach is the greedy approach that consists of two steps: initial clustering and refinement.

As a first step of text clustering, sets of words or descriptors (that describe topic matter) are extracted from the document (or text), which are then analysed for the frequency they occur in the document as compared to other terms. Concept and category models are built on the data that is mined. After this, the clusters are identified for the descriptors where they can be auto tagged.

Various data mining techniques such as clustering, classification, and predictive modelling are used to discover relationship between the concepts. Google search engine is probably the most widely known example of text clustering that

## NOTES

pulls up the pages containing the terms users are searching for. Text clustering helps google to analyse billions of web pages to deliver an accurate and fast results by breaking down unstructured data from web pages and turns into a mix model, tagging pages with keywords that are then used in search results.

---

### 10.5 TEMPORAL MINING

---

Temporal mining is an important extension of the data mining. Temporal data mining is defined as process of knowledge discovery used for extraction of implicit, non-trivial, and potentially useful abstract information that is previously unrecorded, from large collection of temporal data (data that represents a state in time or a sequences of a primary data type most commonly numerical or categorical values and multivariate or composite information sometimes). Temporal mining helps to identify temporal patterns, unexpected trends or other hidden relations in the large sequential data (data that is ordered with respect to some index) consisting of sequence of nominal symbols from a particular alphabet, usually called a temporal sequence and a continuous, real valued elements known as a time series, using a combination of techniques from machine-learning, statistics and database technologies.

The temporal data often involve duration of time, i.e., a start time and an end time. There are various types of temporal data:

- **Fully temporal:** This type of data is completely dependent on time. Ex. Transactional data in databases
- **Time Stamped:** This data has explicit information related to time. It is possible to find temporal distance between data elements. Inferences made can be either temporal or non-temporal. Example includes stock exchange data or inventory management.
- **Time Series:** This is a special case of time stamped where events are separated uniformly in time dimension.
- **Sequences:** In this type, events are ordered with or without a concrete notion of time. Ex. Customer shopping sequences, biological sequences.

The eventual goal of temporal mining is to uncover the hidden relations between sequences and subsequence of events. There are three steps in identifying relations between sequences of events:

- Representation and modelling of data sequence in a suitable form
- Defining similarity measures between sequences
- Application of models and representations to the actual mining problem.

The approach to solve the problem through mining may be different and depends on the nature of the event.

There are various techniques used for the temporal data mining that can be broadly categorised as:

- **Prediction:** is an important task in data mining to predict future values of the time series based on its past samples. A predictive model for the data needs to be built to predict the future. Prediction has huge application in fields like medicine, finance, environment, and engineering.
- **Classification:** is a supervised learning, where the unknown set of attributes are classified into predefined classes. In temporal classification, temporal sequence is assumed to belong to one of the predefined classes which can also be called as training database. Using this training database, it can be determined, which class the given input belongs to. Classification has a huge application in gesture recognition, speech recognition, handwriting recognition, signature verification etc.
- **Clustering:** is an unsupervised learning that is concerned with grouping a collection of time series (or sequence) based on their similarity. Clustering provides a mechanism to automatically find patterns in large datasets that would otherwise be difficult to identify. Manhattan distance, Euclidean distance are some of the basis for similarity measures to group the data. The well-known clustering techniques are K-means, K-medoids. Clustering has an application in web activity logs where clusters can indicate navigation patterns of different user groups, in financial data, where the stocks that exhibit similar trends in price movement can be grouped in same cluster etc.
- **Search and Retrieval:** searching and retrieval is another important task in temporal data mining and plays an important role in interactive exploration of large sequential database. The goal is to locate subsequences in large database of sequences in a single sequence efficiently. While the problem of locating exact matches of substring efficiently is a trivial problem, looking for approximate match and handle it efficiently is a difficult task. It is an approximate match, we are interested in when it is an application like content-based retrieval.
- **Pattern discovery/Association Rule:** unlike in search and retrieval applications, there is no specific query in hand in pattern discovery with which to search the database. The aim is to uncover all the patterns of interest. Pattern discovery has its origin in data mining itself whereas the other temporal data mining techniques such as prediction, classification, clustering and matching has their origins other disciplines like estimation theory, machine learning or pattern recognition.

## NOTES

---

## 10.6 SPATIAL DATA MINING

---

Data mining contains the automatic procedure for discovering data patterns. This set-ups correlation between datasets of different types that are not predicted; for



## NOTES

example, supermarket chains. These chains serve as the main model of entities that makes use of techniques of data mining to increase their sales volume. They carry out analysis of correlations in customer trade practices. On-Line Transaction Processing (OLTP) serves as a traditional model in processing enterprise-oriented data. OLTP tackles transactions related to input, updating and retrieval of data. On-Line Analytical Processing (OLAP) applications analyse, collate and summarize contents. Data mining applies techniques of artificial intelligence and machine learning in finding relationships in the data that was previously not known or were undiscovered. This is not like normal analytical techniques, where the main goal is proving or disproving an existing hypothesis.

### What is Spatial Data Mining?

Spatial data is data that has a spatial or location component. It is viewed as a data about objects that are located in space. Such data is implemented with location attributes like latitude and longitude on a geographical map. It may be accessed using queries containing operators, such as north, south, east and west. Spatial data objects and their non-spatial data are stored in a spatial database.

Spatial data mining is the branch of data mining that deals with spatial, i.e., geographical and geo-referenced data.

Consider a city with a large number of residential buildings spread across it. Each point in a cluster represents a building. The clusters are themselves created based on their characteristics, such as built-up area, stories of construction and commercial value.

### Spatial Data Queries

Mining of spatial data is useful if queries can be made from the database. Spatial database contains complex data. Some kind of software is needed to handle complex data that comprises of geographical objects. PostGIS is one such open source software program adding support for geographic objects to a database. PostGIS is capable of storing spatial data in an enterprise-level relational database. Data is retrieved using standard SQL queries that ensure availability of data to carry out analysis with another program without importing or exporting data into proprietary data formats. It is capable of using spatially aware PostGIS functions for performing analysis in database and can use SQL. PostGIS has many operators and functions of different types for handling spatial data by creating, managing and manipulating. It is also capable of computing geometric buffers, unions, intersections, re-projection in coordinate systems, and so on. Examples shown here put a focus on loading the spatial data and retrieving it.

To find data in the table 'countyp010', use any basic SQL query to make a request for listing. The following SQL query makes a list of all the data in the table and includes very lengthy geometry definitions.

```
SELECT * FROM states010;
```

In this entire database, you will get a list of more than 5,000 countries which is of no use. A more useful query is given in the following example that gives result in a group having limited number of objects.

**Example 10.1:** A basic SELECT DISTINCT query to list the states in India

```
SELECT DISTINCT state FROM states010 WHERE country =
'INDIA';

 States

J & K
Himachal Pradesh
Utter Pradesh
Bihar
Jharkhand
West Bengal
Assam
Arunachal pradesh
Meghalaya
Orissa
Nagaland
Gujarat
Maharashtra
Rajasthan
Punjab
Haryana
...
(28 rows)
```

## NOTES

A **geo-database** is designed for storing, querying and manipulating spatial data and geographic information. This is a **spatial database**. In such a database, spatial data is like any other data type. There may be vector data that is stored as point, line or polygon that may have a spatial reference system associated with it. A record in a geo-database can make use of a geometrical data type for representing the location of an object in the real world along with a standard data type in another database for storing associated attributes of the object. Some geo-databases have support for storing raster data. Such databases are used by ESRI as ArcGIS. Many geo-databases contain custom functions allowing spatial data that can be manipulated and retrieved by the use of SQL. For example, we may like to find a list of residents in an area that is exposed to a potential hazard, either by environmental consideration or due to storage of hazardous chemicals. There are some geo-databases in which spatial data can be accessed by the use of specialized software in a client's computer.

## NOTES

Spatial database is one of the components in a Geographic Information System (GIS). This database stores and manipulates data. A complete system has client software for viewing and editing stored data inside the database. Geodatabases are also capable of serving data directly to the server of Web Map software. Internet Map Server, Google's mapping API and ESRI's ArcGIS MapServer are examples of such software.

The primary advantage that spatial databases offer is their support to GIS in adding Relational Database Management Systems (RDBMS) capabilities to the existing databases. These provide support for SQL and are capable of generating complex geospatial queries.

The objective of data mining is to find patterns in geographical data, following similar functions as in data mining.

**Data Mining Techniques:** Machine learning techniques are of four major categories. These are given below.

- Classification
- Association
- Clustering
- Numeric prediction

### Features of Spatial Databases

Database systems make use of indexes to process queries for retrieving data values, but these are not optimal for spatial queries. Spatial databases utilize spatial index to carry out database operations. A typical SQL query uses `SELECT` statement, but spatial databases, in addition to this, are capable of performing various other spatial operations. Open Geospatial Consortium supports the following types of queries along with other types:

- **Spatial Measurements:** Computes line length, polygon area, the distance between geometries, etc.
- **Spatial Functions:** Modify existing features to create new ones, for example by providing a buffer around them, intersecting features, etc.
- **Spatial Predicates:** Allows true/false queries about spatial relationships between geometries. Examples include “do two polygons overlap” or ‘is there a residence located within a mile of the area we are planning to build the landfill?’
- **Geometry Constructors:** Creates new geometries, usually by specifying the vertices (points or nodes) which define the shape.
- **Observer Functions:** Queries which return specific information about a feature such as the location of the center of a circle

Some databases support only simplified or modified sets of these operations, especially in cases of NoSQL systems like MongoDB and CouchDB.

Observer functions: Return specific information of a feature in response to a query; for example, finding the centre of a circle when parameters are given in the query.

Every spatial database may not support such types of queries.

### Spatial Database Systems

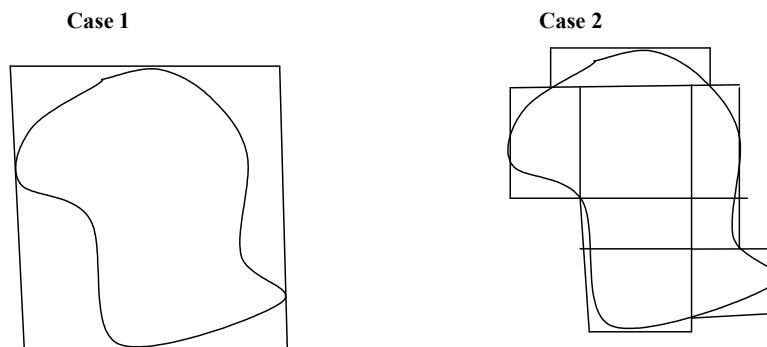
The following are spatial databases:

- All OpenGIS specification compliant products.
- Boeing Spatial Query Server spatially enables Sybase ASE.
- Oracle Spatial.
- Microsoft SQL Server since version 2008 supports spatial types.
- PostgreSQL, DBMS (Database Management System), which utilizes spatial extension of PostGIS for implementing standardized data type geometry and corresponding functions.
- MySQL DBMS that implements data type geometry and some spatial functions, implemented as per OpenGIS specifications. MySQL versions 5.5 and earlier supports spatial data in tables of MyISAM. Spatial features are supported by MySQL 5.0.16, InnoDB, NDB, BDB and ARCHIVE.
- Spatial Databox is a spatial engine that is fast and provides spatial queries of the type nearest neighbour, containment and tile-based over and above a guaranteed real-time response to queries in supporting interactive map mashups.

### Spatial Data Structures

Due to the unique features of spatial data, specific data structures have been designed to store or index them. For example, Minimum Bounding Rectangle (MBR) (see Figure 10.12) in which an irregular shaped two-dimensional object is enclosed:

- A rectangle touching its sides (Case 1).
- A number of rectangles encompassed by it (Case 2).



**Fig. 10.12** Minimum Bounding Rectangle

### NOTES

## NOTES

### Spatial Data Mining Primitives

There are many operations which support spatial data mining in space and define their relationships. Few of these operations are disjoint, overlaps, equals, covered by or inside and covers or contains.

As we are dealing with objects in space, the distance between objects can be quantified as minimum, maximum, average or centre.

### Generalization

Concept hierarchy, seen earlier in the case of multidimensional analysis, also finds an application in spatial data examples. Concept hierarchies permit development of rules and relationships at each level of hierarchy. Generalization means obtaining information at a higher level from information found at lower levels.

Figure 10.13 shows an example of generalization.

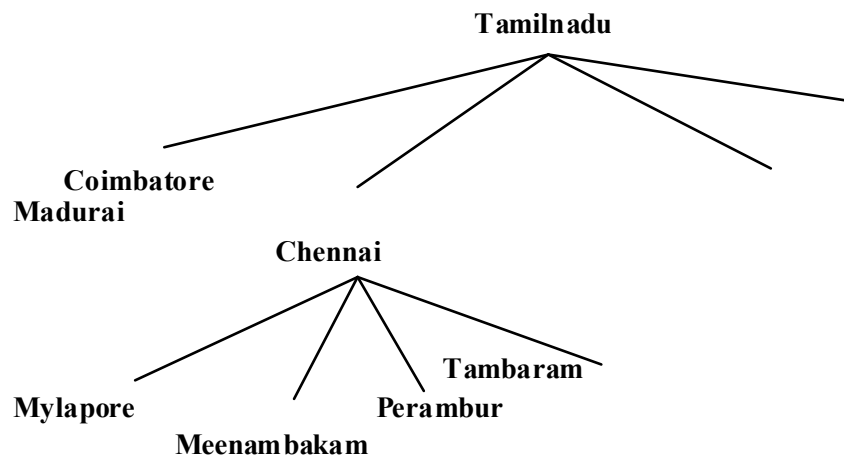


Fig. 10.13 An Example of Generalization

### Spatial Data Cube Construction and Spatial OLAP

A spatial data warehouse is a subject-oriented, integrated, time-variant and non-volatile collection of both spatial and non-spatial data, in support of spatial data mining and spatial data related decision-making process.

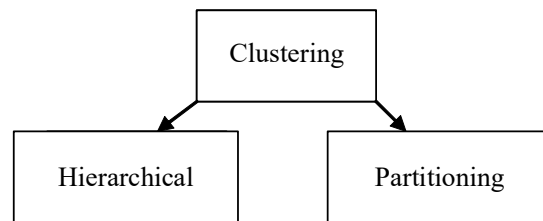
### Mining Spatial Association and Co-location Patterns

Similar to the mining of association rules in transactional and relational databases, spatial association rules can be mined in spatial databases. A spatial association rule is of the form  $A \Rightarrow B [s\%, c\%]$ , where  $A$  and  $B$  are sets of spatial or nonspatial predicates,  $s\%$  is the support of the rule, and  $c\%$  is the confidence of the rule. For example, the following is a spatial association rule:

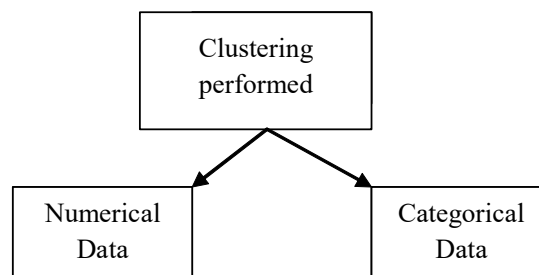
*is a*( $X$ , "school")  $\wedge$  *close to*( $X$ , "sports center")  $\Rightarrow$  *close to*( $X$ , "park") [0.5%, 80%]. This rule states that 80% of schools that are close to sports centers are also close to parks, and 0.5% of the data belongs to such a case.

## Spatial Clustering Methods

Spatial clustering methods have evolved by either applying the same algorithms or by modifying the three clustering methods. These are the partitioning methods, hierarchical methods and grid-based methods (see Figure 10.14).



*Fig. 10.14 Approaches to Clustering*



*Fig. 10.15 Types of Data for Clustering*

## Spatial Classification and Spatial Trend Analysis

Spatial classification analyses spatial objects to derive classification schemes in relevance to certain spatial properties, such as the neighbourhood of a district, highway or river. For example, we have to classify regions into rich, poor and average.

Spatial trend analysis deals with another issue: the detection of changes and trends along a spatial dimension. Trend analysis detects changes with time, whereas spatial trend analysis replaces time with space and studies the trend of non-spatial or spatial data changing with space. We often apply regression and correlation analysis methods for spatial trend, for example, economic situation of a city, climate changes, etc.

There are many applications where patterns change with space and time; for example, traffic flows on highways and in cities are both time and space related.

## Spatial Outliers

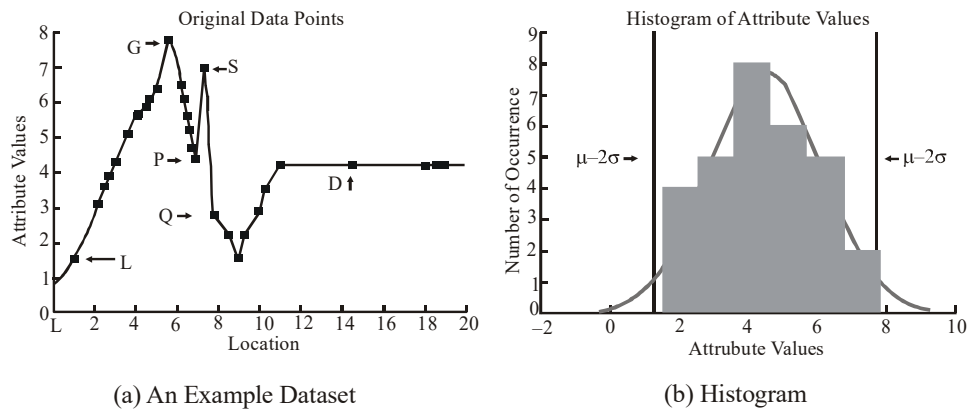
A spatial outlier is an object that is spatially referenced and its non-spatial attribute values differ significantly from that of its neighbourhood. Identification of such outliers leads to interesting and useful spatial patterns that need to be further analysed. Previous work in detecting such outliers focussed on outliers having a

## NOTES

**NOTES**

single attribute. Outliers, on identification, lead to discovery of implicit knowledge. The existing methods detect spatial outliers in multidimensional geometric data sets with distance metric.

Outliers maintain the data over time that appears inconsistent with the remaining part of the data set. One of the outlier is global outlier. This outlier enables discovering unexpected information. It finds practical applications in dealing with credit card frauds, performance analysis of athletes, checking irregularities in voting and in weather forecasting. This outlier gives the observations on data inconsistency. It applies geographic information systems and spatial databases in transportation, public safety, ecology, etc., and location-based services. A spatial outlier is an object referenced spatially in which values of non-spatial attributes have significant differences from those referenced spatially in its neighborhood. We may explain global and spatial outlier detection with the help of an example. In Figure 10.16(a), the location of data points is along X-axis and their attribute values are on Y-axis. The outlier, thus detected is point G having a very high value of 7.9 that exceeds the threshold value of 7.71 that is shown in Figure 10.16(b). For this test, a normal distribution of attribute values is considered. Here, S, a spatial outlier, shows an observed value, significantly different from its neighbours, P and Q.



**Fig. 10.16** A Data Set for Outlier Detection

**Tests for Detecting Spatial Outliers**

Initially, we separate the non-spatial from the spatial attributes. Spatial attributes are used to characterize location, neighbourhood and distance. Dimensions of non-spatial attributes compare spatially-referenced object in its neighbourhood. There are two types of bi-partite multidimensional tests, namely graphical and quantitative. Graphical tests visualize spatial data highlighting spatial outliers. Quantitative methods carry precise test for distinguishing spatial outliers from the remainder of data.

**Spatial Co-Location Rules**

Co-location patterns are subsets of the Boolean spatial features. Instances of co-location pattern are mostly found in close proximity. Such rules are models which

draw inference on the presence of Boolean spatial features in the neighbourhood, with instances of other Boolean spatial features. One of the co-location patterns is ('+', '£') and ('o', '×'). Co-location rule is a process to identify co-location patterns from large spatial data sets having a large number of Boolean features (see Figure 10.17).

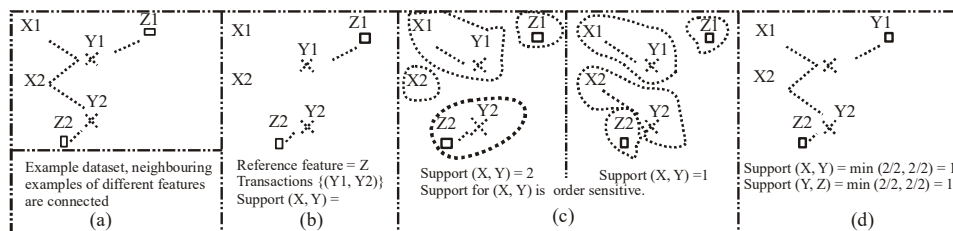
**NOTES**

**Co-Location and Association Rule**

Location rule discovery problem looks similar but it is different from the association rule-mining problem. This problem crops up due to lack of transactions. These rules have been derived from associations that have support values more than the threshold values. Mining of these rules is done for identifying frequent item sets to plan marketing campaigns or store layouts.

Location rule has three categories: (i) Association rules, (ii) Spatial statistics, and (iii) Event-centric approach. Spatial statistics finds spatial correlation, characterizing relationships between different types of spatial features by using K function.

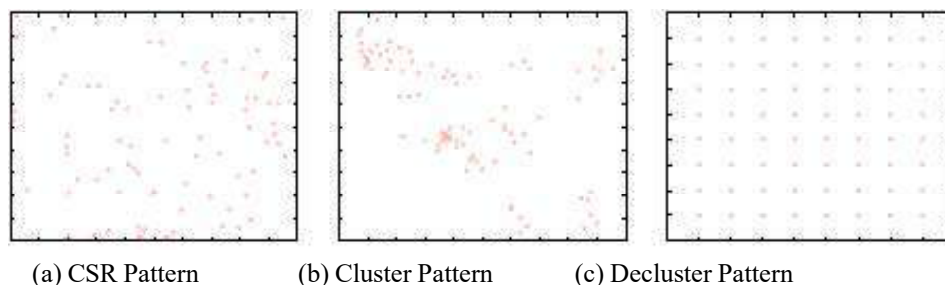
It also measures all possible co-location patterns.



**Fig. 10.17** Spatial Co-Location Patterns

**Spatial Clustering**

It is the process of making clusters or groups from a set of spatial objects into clusters such that objects within a cluster show high similarity when compared to one another, but not so with objects that belong to other clusters. For example, clustering determines 'hot spots' in crime analysis and tracking of diseases. Agencies involved in criminal investigation and justice use computer technologies for identifying the hot spots of criminal activities (see Figure 10.18).



**Fig. 10.18** Spatial Clustering Patterns



## NOTES

### Mining Raster Databases

In these databases, we usually handle vector data that consists of points, lines, polygons and their composition. Generally, data include maps, design graphs and 3D representations of items. We also handle huge amount of space-related data such as digital raster (image) forms and remote sensing data.

#### Check Your Progress

5. What is text clustering?
6. Define spatial data mining.

---

### 10.7 ANSWERS TO CHECK YOUR PROGRESS QUESTIONS

---

1. Web page mining is the mining of data related to the World Wide Web and stored in the Web pages. The data referred is the one present in the Web pages or related to the Web activity.
2. Web structure mining is concerned with discovering the model underlying the link structure of the Web. This model helps us in categorizing Web pages and generating information about similarities and relationships between different Websites.
3. Text mining describes data in text format rather than in the structural form. It extracts patterns and associations from large databases.
4. There are two approaches of text mining:
  - (i) Keyword-based analysis
  - (ii) Document classification analysis
5. Clustering is a technique or task of finding a group of unlabelled objects so that the objects in a group will be related or similar to each other and will be unrelated or different to the objects in other groups.
6. Spatial data mining is the branch of data mining that deals with spatial, i.e., geographical and geo-referenced data.

---

### 10.8 SUMMARY

---

- The World Wide Web serves as a huge, widely distributed, global information service center for news, advertisements, consumer information, financial management, education, government, e-commerce, and many other information services.
- Web content mining describes the discovery of useful information from the Web contents.

- Web structure mining is concerned with discovering the model underlying the link structures of the Web.
- Web usage mining which deals with the study of data generated by the users and their usage characteristics.
- Web page mining is the mining of data related to the World Wide Web and stored in the Web pages. The data referred is the one present in the Web pages or related to the Web activity.
- Web mining enables personalized marketing, by introducing e-commerce that has resulted in higher trade volumes. Government agencies use it for classifying threats to fight terrorism.
- Text mining describes data in text format rather than in the structural form. It extracts patterns and associations from large databases. Another feature is that text mining and information retrieval are different; text mining refers to data mining, whereas information retrieval refers to database management system.
- Information Retrieval (IR) is a field that has been developing in parallel with database systems for many years. Unlike the field of database systems, which has focused on query and transaction processing of structured data, information retrieval is concerned with the organization and retrieval of information from a large number of text-based documents.
- In video mining, we find correlations and patterns from a large video database. In this mining, we analyse a video clipping or multiple video clippings and capture the text in video mining and this text data behaves as video data.
- Audio mining is a technique for auto-analysing and searching for the content of an audio signal, used in automatic speech recognition. Musical audio mining is the identification of perception-based characteristics of a musical piece.
- Dimensionality reduction means obtaining compressed or reduced data from the original data. In this process, data may be lost or remain lossless.
- Clustering is a technique or task of finding a group of unlabelled objects so that the objects in a group will be related or similar to each other and will be unrelated or different to the objects in other groups.
- Temporal data mining is defined as process of knowledge discovery used for extraction of implicit, non-trivial, and potentially useful abstract information that is previously unrecorded, from large collection of temporal data (data that represents a state in time or a sequences of a primary data type most commonly numerical or categorical values and multivariate or composite information sometimes).
- Spatial data mining is the branch of data mining that deals with spatial, i.e., geographical and geo-referenced data.

## NOTES

## NOTES

- Mining of spatial data is useful if queries can be made from the database.
- A spatial data warehouse is a subject-oriented, integrated, time-variant and non-volatile collection of both spatial and non-spatial data, in support of spatial data mining and spatial data related decision-making process.
- A spatial outlier is an object that is spatially referenced and its non-spatial attribute values differ significantly from that of its neighbourhood. Identification of such outliers leads to interesting and useful spatial patterns that need to be further analysed.

---

### 10.9 KEY WORDS

---

- **Spatial Data:** Data that have a spatial or location component.
- **Generalization:** Obtaining information at a higher level from information found at lower levels.
- **Similarity Search:** Finding identical data description or data content.
- **Degree of Relevance:** A measure of closeness of words.
- **Web Page Mining:** Mining of data related to the World Wide Web and stored in the Web pages.
- **Outlier Analysis:** A type of data analysis that determines and reports records in the database that is significantly different from the expectations.

---

### 10.10 SELF ASSESSMENT QUESTIONS AND EXERCISES

---

#### Short Answer Questions

1. What are the different types of web mining?
2. Discuss the pros and cons of web mining.
3. What do you understand by web content mining?
4. Write a note on video and audio mining?
5. What is dimensionality reduction for text?
6. What is text clustering?
7. What are the ways by which spatial association mining is implemented?
8. What are the spatial clustering methods?

#### Long Answer Questions

1. What are the problems in mining the complex data?
2. What are different methods for mining unstructured data?

3. Explain the measures of text retrieval?
4. What are the approaches of text mining? Explain.
5. What are the techniques used for the temporal data mining?
6. Write a detailed note on spatial outliners.

## NOTES

---

### 10.11 FURTHER READINGS

---

- Han, Jiawei, Micheline Kamber and Jian Pei. 2011. *Data Mining: Concepts and Techniques*, 2nd edition. Amsterdam: Elsevier.
- Pujari, Arun K. 2010. *Data Mining Techniques*, 2nd edition. United States: Universities Press.
- Anahory, Sam and Dennis Murray. 1997. *Data Warehousing in the Real World: A Practical Guide for Building Decision Support Systems*, 1st edition. Boston: Addison Wesley.
- Witten, I. H. and Eibe Frank. 2005. *Data Mining: Practical Machine Learning Tools and Techniques*, 2nd edition. Amsterdam: Elsevier.
- Soman, K. P., Shyam Diwakar and V. Ajay. 2006. *Insight Into Data Mining: Theory and Practice*. New Delhi: PHI.

---

## UNIT 11 VISUAL DATA MINING

---

### NOTES

#### Structure

- 11.0 Introduction
- 11.1 Objectives
- 11.2 Data Visualization
- 11.3 Visual Data Mining
  - 11.3.1 Weka
  - 11.3.2 RapidMiner
  - 11.3.3 MATLAB (Matrix Laboratory)
  - 11.3.4 Visualization based Clustering in Weka- A Practical Approach
- 11.4 Answers to Check Your Progress Questions
- 11.5 Summary
- 11.6 Key Words
- 11.7 Self Assessment Questions and Exercises
- 11.8 Further Readings

---

### 11.0 INTRODUCTION

---

In today's world, visual data mining serves as a technique that increasingly provides a competitive advantage to those who are willing to produce insights from their data to increase their efficiency, spot trends, and get a better Rate of Interest on business efforts. Visual data mining allows the user to view the impact of different factors on data, aiding companies with future decision making.

This technique takes data mining to a much broader and new level. It combines data mining with high-performance visuals, intuitive reporting, and interactive dashboards. With the use of these features, users can simultaneously derive information from various sources using data mash up technology while analyzing latest and upcoming trends, and discovering correlations among the analyzed trends.

---

### 11.1 OBJECTIVES

---

After going through this unit, you will be able to:

- Understand data visualization and its purpose
- Explain visual data mining
- Explain the usage of visual data mining tools

---

### 11.2 DATA VISUALIZATION

---

Data visualization aims to communicate data clearly and effectively through graphical representation. Data visualization has been used extensively in many applications—

for example, at work for reporting, managing business operations, and tracking progress of tasks. More popularly, we can take advantage of visualization techniques to discover data relationships that are otherwise not easily observable by looking at the raw data. The advantage of visual data exploration is that the user is directly involved in the data mining process. There are a large number of information visualization techniques that have been developed over the last two decades to support the exploration of large data sets.

### **Purpose of Visualization**

- Gain insight into an information space by mapping data onto graphical primitives
- Provide qualitative overview of large data sets
- Search for patterns, trends, structure, irregularities, relationships among data
- Help find interesting regions and suitable parameters for further quantitative analysis
- Provide a visual proof of computer representations derived

The main advantages of visual data exploration over automatic data mining techniques are:

- visual data exploration can easily deal with highly non-homogeneous and noisy data
- Visual data exploration is intuitive and requires no understanding of complex mathematical or statistical algorithms or parameters.
- Visualization can provide a qualitative overview of the data, allowing data phenomena to be isolated for further quantitative analysis.

### **The data type to be visualized may be**

- One-dimensional data, such as temporal (time-series) data
- Two-dimensional data, such as geographical maps
- Multidimensional data, such as relational tables
- Text and hypertext, such as news articles and Web documents
- Hierarchies and graphs, such as telephone calls and Web documents
- Algorithms and software, such as debugging operations

The visualization technique used may be classified as:

- Standard 2D/3D displays, such as bar charts and x-y plots
- Geometrically transformed displays, such as landscapes and parallel coordinates
- Icon-based displays, such as needle icons and star icons
- Dense pixel displays, such as the recursive pattern and circle segments
- Stacked displays, such as tree maps and dimensional stacking

## **NOTES**

NOTES

**Check Your Progress**

1. What is data visualization?
2. Give an advantage of visual data exploration.
3. What are the classifications for data visualization techniques?

---

### 11.3 VISUAL DATA MINING

---

The presence of large quantities of data requires interactive analysis for decision-making. Interactive decision support system (DSS) based on knowledge discovery in databases (KDD) process proves to be useful. Data visualization techniques are used in the KDD stages to increase the user participation as well as its confidence in the result in order to improve the decision support quality. Applying visual representation in the KDD process aims to facilitate the understanding over its results. As visual data mining “discovers implicit and useful knowledge from large datasets using data and/or knowledge visualization techniques”; in other words we can say that it is the combination of data visualization and data mining.

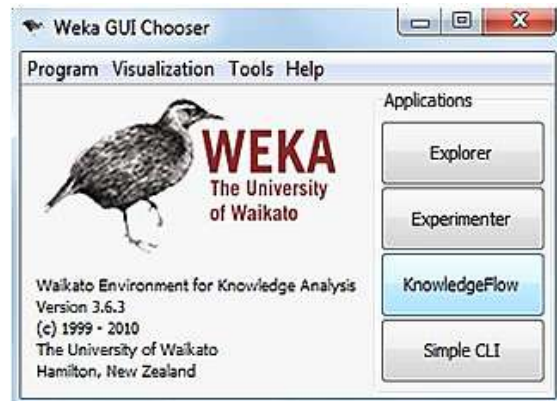
Thus, visualization techniques can be integrated into the process of KDD in three different ways:

- to preview the data to be analyzed
- to help in understanding the results of data mining
- to understand the partial results of the iterations inherent in the process of extracting knowledge

The three major data mining software tools are described below which solve the above mentioned purpose of visual data mining for effective data analysis in knowledge mining.

#### 11.3.1 Weka

Waikato Environment for Knowledge Analysis was developed in 1992. Weka is collection of different machine learning algorithms which can be used for data mining. It is written in Java and is especially used for educational purposes. Weka is a platform independent, open source, easy to use, data processing tool, flexible for scripting experiments and 3 graphical user interface tool. Weka contains different tools and algorithms for visualization, forecasting, regression, classification, pre-processing and clustering. It is supportable on different platforms such as Mac OS, Linux and Windows. When dealing with large data sets, it is best to use a CL based approach as Explorer tries to load the whole data set into the main memory causing performance issues.



*Fig. 11.1 Weka Interface*

## NOTES

### General Features

- Weka is a Java based open source tool data mining tool which is a collection of many data mining and machine learning algorithms, including pre-processing on data, classification, clustering, and association rule extraction
- Weka provides three graphical user interfaces i.e. the Explorer for exploratory data analysis to support preprocessing, attribute selection, learning, visualization, the Experimenter that provides experimental environment for testing and evaluating machine learning algorithms, and the Knowledge Flow for new process model inspired interface for visual design of KDD process. A simple Command-line explorer which is a simple interface for typing commands is also provided by Weka.

### Specialization:

- Weka is best suited for mining association rules.
- Stronger in machine learning techniques.
- Suited for machine Learning.

### Advantages

- It is also suitable for developing new machine learning schemes.
- Results visualization is good for further analysis
- Weka loads data file in formats of ARFF, CSV, binary.
- Though it is open source, Free, Extensible, hence can be integrated into other java packages.

### Limitation

- It lacks proper and adequate documentations and suffers from “Kitchen Sink Syndrome” where systems are updated constantly.
- Worse connectivity to Excel spreadsheet and non-Java based databases.



## NOTES

- CSV reader is not as robust as in RapidMiner.
- Weka is much weaker in classical statistics.
- Does not have the facility to save parameters for scaling to apply to future datasets.
- Does not have automatic facility for Parameter optimization of machine learning/statistical methods.

### 11.3.2 RapidMiner

RapidMiner also called Yet another Learning Environment, was developed in 2001, written in java. RapidMiner provides an integrated environment for data preparation, machine learning, deep learning, text mining, and predictive analytics. It is available as both free and commercial editions. It is one of the most used predictive analytic tools. Gartner recognized RapidMiner and Knife as leaders in the magic quadrant for advanced analytic platforms in 2016. It is available for Mac OS, Linux and Windows. Its user friendly and rich library of data science algorithms and machine learning algorithms makes it first choice for enterprises to implement predictive analysis in their business processes. Its unique features are repeatable work flows, built in templates, visualization and integration with different languages like Weka, SPSS, Python and R which helps in rapid prototyping. RapidMiner is mainly used in educational and research fields for data exploration and visualization, data mining, financial forecasting, segmentation image mining can be integrated with Weka.

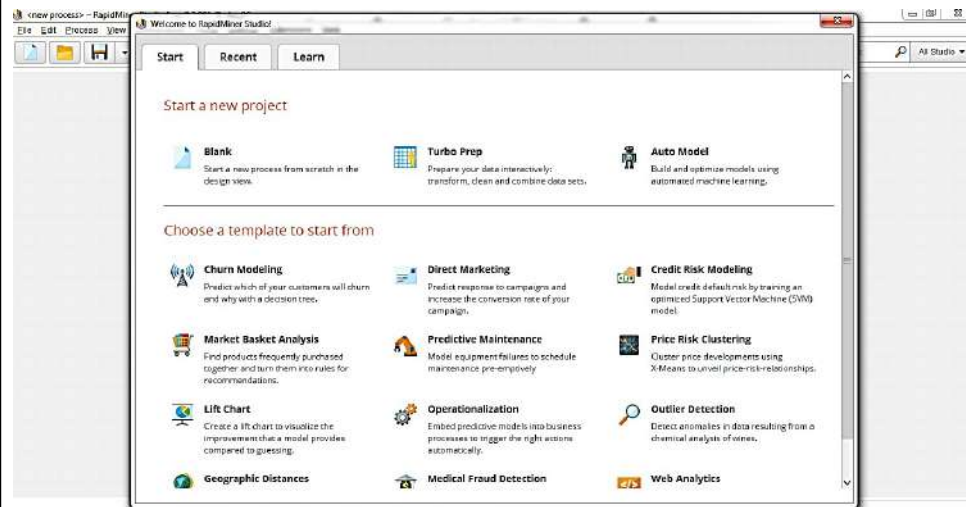


Fig. 11.2 RapidMiner Studio Interface

### General Features

- RapidMiner is an environment for machine learning and data mining processes.

- It represents a new approach to design even very complicated problems by using a modular operator concept which allows design of complex nested operator chains for huge number of learning problems.
- RapidMiner uses XML to describe the operator trees modeling knowledge discovery process.
- It has flexible operators for data input and output file formats.
- It contains more than 100 learning schemes for regression classification and clustering analysis.
- RapidMiner supports about twenty two file formats.
- RapidMiner has a lot of functionality, is polished and has good connectivity.
- RapidMiner includes many learning algorithms from Weka.
- It easily reads and writes Excel files and different databases.
- You program by piping components together in a graphic ETL work flows.
- If you set up an illegal work flows RapidMiner suggest quick fixes to make it legal.

## NOTES

### Specialization

- RapidMiner provides support for most types of databases, which means that users can import information from a variety of database sources to be examined and analyzed within the application.
- Specialized for Business solutions that include predictive analysis and statistical computing.

### Advantages

- It has the full facility for model evaluation using cross validation and independent validation sets.
- Over 1,500 methods for data integration, data transformation, analysis and, modeling as well as visualization – no other solution on the market offers more procedures and therefore more possibilities of defining the optimal analysis processes.
- RapidMiner offers numerous procedures, especially in the area of attribute selection and for outlier detection, which no other solution offers.

### Limitation

- RapidMiner is the data mining software package that is most suited for people who are accustomed to working with database files, such as in academic settings or in business settings. The reason for this is that the software requires the ability to manipulate SQL statements and files.

## NOTES

### 11.3.3 MATLAB (Matrix Laboratory)

MATLAB is a high-performance multi-paradigm environment for numerical computing. It is developed by MathWorks organization and first released in 1984. It mixes computation, visualization and programming in a very easy way. Its programming environment is very easy to use where problems and solutions are represented in easy mathematical form. When doing data mining, a large part of the work is to manipulate data. The part of coding the algorithm can be quite short since MATLAB has a lot of powerful toolboxes for data mining. And when manipulating data, MATLAB is definitely better. It is normal since it is done to work with matrices (MATrix LABoratory). MATLAB is available for all OS. It is primarily written in c, C++ and java and is mainly used for numerical computing. Typical uses include: data analysis, exploration, and visualization.

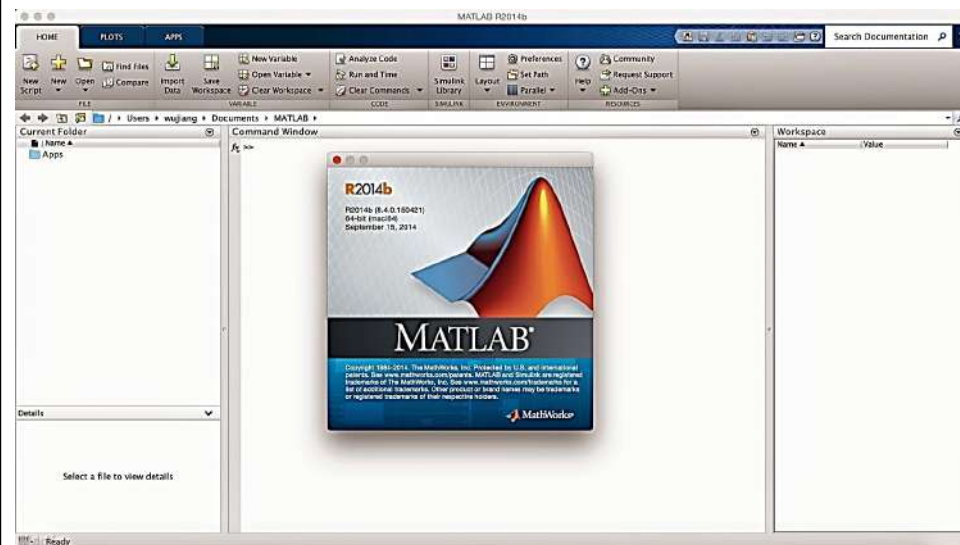


Fig. 11.3 MATLAB Interface

#### General features

- It is a high-level language for numerical computation, visualization and application development.
- It also provides an interactive environment for iterative exploration, design and problem solving.
- It provides vast library of mathematical functions for linear algebra, statistics, Fourier analysis, filtering, optimization, numerical integration and solving ordinary differential equations.
- It provides built-in graphics for visualizing data and tools for creating custom plots.
- MATLAB's programming interface gives development tools for improving code quality maintainability and maximizing performance.

- It provides tools for building applications with custom graphical interfaces.
- It provides functions for integrating MATLAB based algorithms with external applications and languages such as C, Java, .NET and Microsoft Excel.

## Specialization

MATLAB is widely used as a computational tool in science and engineering encompassing the fields of physics, chemistry, math and all engineering streams. It is used in a range of applications including signal processing and communications, image and video processing, control systems, test and measurement, computational finance and computational biology.

## Advantages

- Give visualize results without the need for complicated and time consuming programming.
- MATLAB provides its user accurate solution of the problems and produce code easily.
- MATLAB is interpreted language, errors are easier to fix.

## Disadvantages

- It's an Interpreted language so works slow.
- It's a licensed tool need to be purchased.

### 11.3.4 Visualization based Clustering in Weka - A Practical Approach

The objective of this section is to show how the visualization helped in improving the clusters quality based on clustering results visualization. Figure 11.4 shows the clusters generated on the iris Weka dataset using K-means clustering algorithms in Weka.

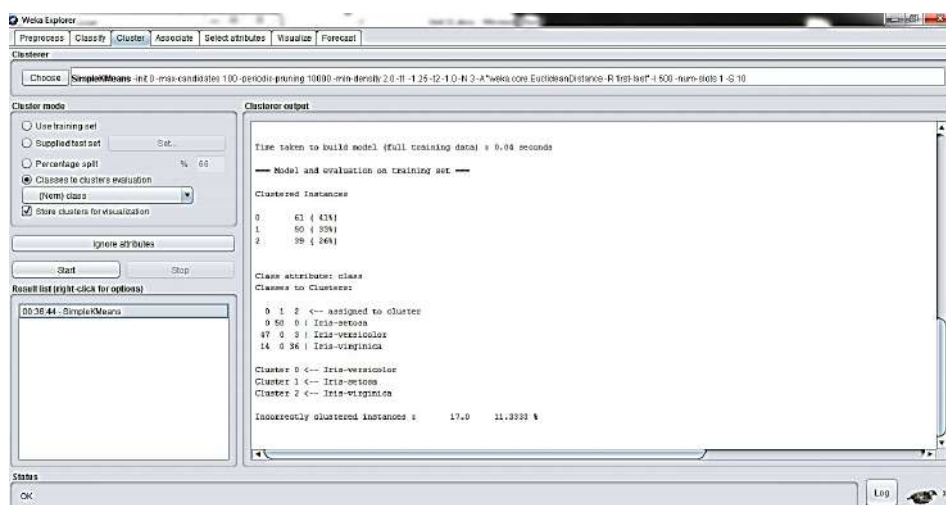


Fig. 11.4 K-means Clustering on Iris Weka dataset

## NOTES

NOTES

From above it is observed that the percentage of incorrectly clustered instances is 11.33%. Now, focusing on to reduce this percentage we work with the visualization of clustered instances with respect to every attribute in the given dataset parameters. Figure 11.5 shows that the distribution of instances between sepal length and sepal width attributes is not much distinguishable. The cluster-1 and cluster-2 instances are coming in approximately same area space for these attributes.

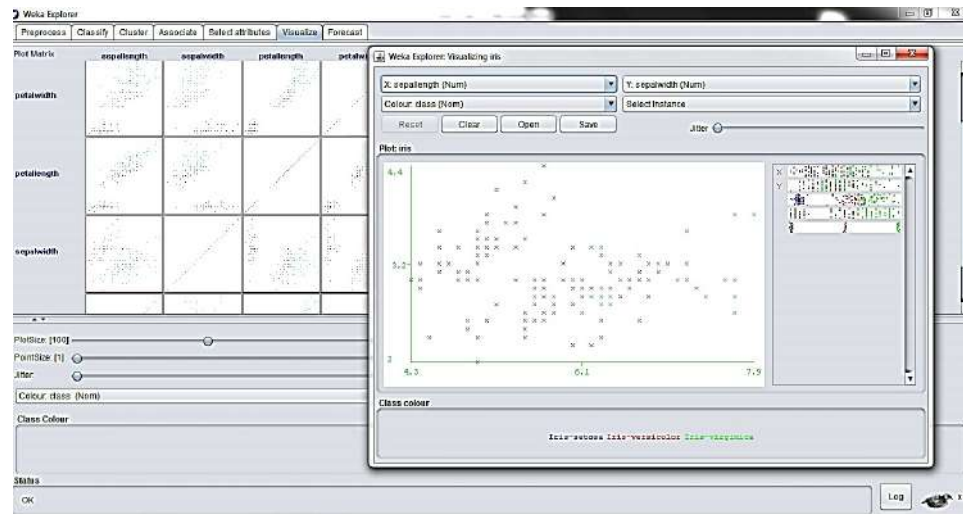


Fig. 11.5 Attribute Wise Visualization of Clustering Results

From above visualization it is observed that removal of any one of the attribute may improve the cluster quality. Figure 11.6 shows the clustering results after removal of sepal length parameter from the given dataset. Now, the percentage of incorrectly clustered instances observed is 5.33%.

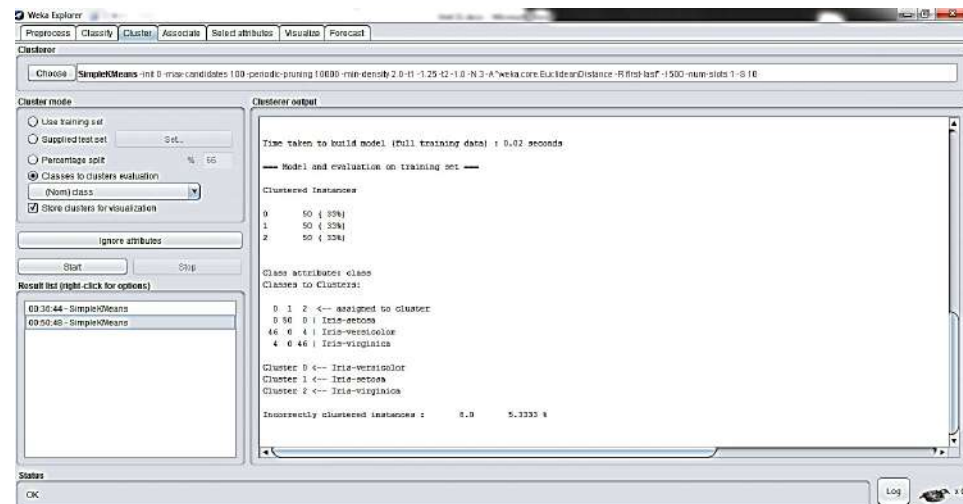


Fig. 11.6 K-means Clustering on Iris Weka Dataset after Removal of Sepal Length Parameter

On comparing the both percentages of incorrectly clustered instances, it is observed that the cluster quality is improved approx by 6% based on visualization of results.

The key summary of the above practical approach is that based on visualization of data and data mining results more effective data analysis can be performed in more effective and accurate way.

## NOTES

### Check Your Progress

4. How can visualization techniques be integrated into the process of KDD?
5. Which are the three major data mining software tools?
6. Give a specialization of RapidMiner.
7. Give an advantage of MATLAB.

## 11.4 ANSWERS TO CHECK YOUR PROGRESS QUESTIONS

1. Data visualization aims to communicate data clearly and effectively through graphical representation.
2. Visual data exploration can easily deal with highly non-homogeneous and noisy data
3. The visualization technique used may be classified as - standard 2D/3D displays, geometrically transformed displays, icon-based displays, dense pixel displays and stacked displays.
4. Visualization techniques can be integrated into the process of KDD in three different ways: to preview the data to be analyzed, to help in understanding the results of data mining and to understand the partial results of the iterations inherent in the process of extracting knowledge
5. The three major data mining software tools are: Weka, RapidMiner and MATLAB.
6. RapidMiner is specialized for business solutions that include predictive analysis and statistical computing.
7. MATLAB provides its user accurate solution of the problems and produce code easily.

## 11.5 SUMMARY

- Data visualization aims to communicate data clearly and effectively through graphical representation.

## NOTES

- Data visualization has been used extensively in many applications—for example, at work for reporting, managing business operations, and tracking progress of tasks. More popularly, we can take advantage of visualization techniques to discover data relationships that are otherwise not easily observable by looking at the raw data.
- The advantage of visual data exploration is that the user is directly involved in the data mining process.
- Visualization can provide a qualitative overview of the data, allowing data phenomena to be isolated for further quantitative analysis.
- The presence of large quantities of data requires interactive analysis for decision-making. Interactive decision support system (DSS) based on knowledge discovery in databases (KDD) process proves to be useful.
- Applying visual representation in the KDD process aims to facilitate the understanding over its results. As visual data mining “discovers implicit and useful knowledge from large datasets using data and/or knowledge visualization techniques”
- Waikato Environment for Knowledge Analysis was developed in 1992. Weka is collection of different machine learning algorithms which can be used for data mining. It is written in Java and is especially used for educational purposes.
- Weka contains different tools and algorithms for visualization, forecasting, regression, classification, pre-processing and clustering. It is supportable on different platforms such as Mac OS, Linux and Windows.
- Weka is best suited for mining association rules, is stronger in machine learning techniques and suited for machine learning.
- RapidMiner also called Yet another Learning Environment, was developed in 2001, written in java.
- RapidMiner provides an integrated environment for data preparation, machine learning, deep learning, text mining, and predictive analytics. It is available as both free and commercial editions.
- RapidMiner provides support for most types of databases, which means that users can import information from a variety of database sources to be examined and analyzed within the application.
- RapidMiner is specialized for business solutions that include predictive analysis and statistical computing.
- MATLAB is a high-performance multi-paradigm environment for numerical computing. It is developed by Mathworks organization and first released in 1984. It mixes computation, visualization and programming in a very easy way.

- MATLAB is widely used as a computational tool in science and engineering encompassing the fields of physics, chemistry, math and all engineering streams.
- MATLAB may also be used in a range of applications including signal processing and communications, image and video processing, control systems, test and measurement, computational finance and computational biology.

## NOTES

---

### 11.6 KEY WORDS

---

- **Weka:** It is a collection of machine learning algorithms for data mining tasks. It contains tools for data preparation, classification, regression, clustering, association rules mining, and visualization.
- **RapidMiner:** It is a data science platform for teams that unites data preparation, machine learning, and predictive model deployment.
- **MATLAB:** It is a multi-paradigm numerical computing environment and proprietary programming language developed by MathWorks.

---

### 11.7 SELF ASSESSMENT QUESTIONS AND EXERCISES

---

#### Short Answer Questions

1. What are the ways in which techniques can be integrated into the process of KDD?
2. List the advantages of data exploration over automatic data mining techniques.
3. What are the advantages of Weka?
4. What are the specializations of RapidMiner?

#### Long Answer Questions

1. What is data visualization? Discuss its purpose and classification of visualization techniques.
2. Explain the three major data mining software tools.
3. What are the general features, specialization, advantages and limitations of Weka?
4. Write a detailed note on MATLAB. What are its advantages and limitations?



---

## 11.8 FURTHER READINGS

---

### NOTES

Han, Jiawei, Micheline Kamber and Jian Pei. 2011. *Data Mining: Concepts and Techniques*, 2nd edition. Amsterdam: Elsevier.

Pujari, Arun K. 2010. *Data Mining Techniques*, 2nd edition. United States: Universities Press.

Anahory, Sam and Dennis Murray. 1997. *Data Warehousing in the Real World: A Practical Guide for Building Decision Support Systems*, 1st edition. Boston: Addison Wesley.

Witten, I. H. and Eibe Frank. 2005. *Data Mining: Practical Machine Learning Tools and Techniques*, 2nd edition. Amsterdam: Elsevier.

Soman, K. P., Shyam Diwakar and V. Ajay. 2006. *Insight Into Data Mining: Theory and Practice*. New Delhi: PHI.

---

**BLOCK - V**  
**INTRODUCTION TO BIG DATA ANALYTICS**

---

---

**UNIT 12 BIG DATA**  
**CHARACTERISTICS**

---

**NOTES**

**Structure**

- 12.0 Introduction
- 12.1 Objectives
- 12.2 Introduction to Big Data Platform
  - 12.2.1 Nature of Data
- 12.3 Traditional vs Big Data Approach
- 12.4 Answers to Check Your Progress Questions
- 12.5 Summary
- 12.6 Key Words
- 12.7 Self Assessment Questions and Exercises
- 12.8 Further Readings

---

**12.0 INTRODUCTION**

---

Big data is a blanket term for the non-traditional strategies and technologies needed to gather, organize, process, and gather insights from large datasets. While the problem of working with data that exceeds the computing power or storage of a single computer is not new, the pervasiveness, scale, and value of this type of computing has greatly expanded in recent years.

Data drives the modern organizations of the world and hence making sense of this data and unravelling the various patterns and revealing unseen connections within the vast sea of data becomes critical and a hugely rewarding endeavour indeed. In this unit, you will learn about the various types of data and significance of Big data approach.

---

**12.1 OBJECTIVES**

---

After going through this unit, you will be able to:

- Explain the three Vs of Big data
- Differentiate between structured, unstructured and semi structured data
- Discuss the benefits of Big data approach over traditional approach

---

## 12.2 INTRODUCTION TO BIG DATA PLATFORM

---

### NOTES

Data has always offered greatest benefits, but managing, organizing and analyzing the data is always a challenge to the organizations across all industries, no matter what the size of an organization is. Let us first take a look at the definition of data as defined by Oxford dictionary. It defines data as:

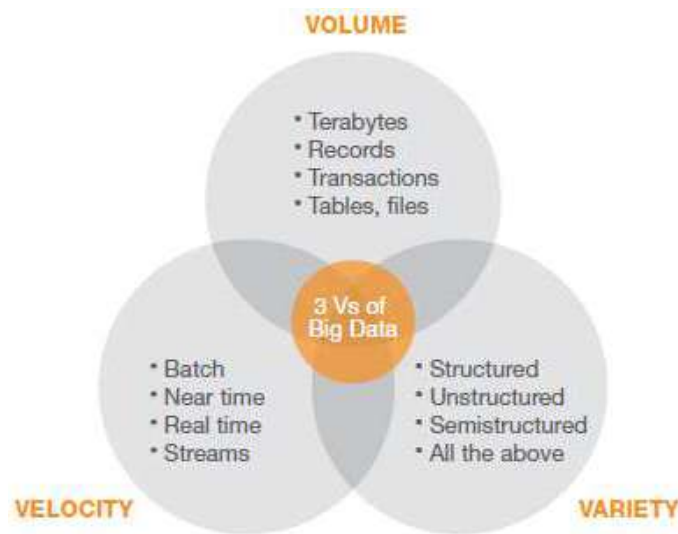
‘The quantities, characters, or symbols on which operations are performed by a computer, which may be stored and transmitted in the form of electrical signals and recorded on magnetic, optical, or mechanical recording media.’

It has always been a struggle for businesses to capture information about their products and services and of course their customers in a rational way. Earlier, it was easy and simple to manage this data when the company had a handful of customers to buy the same product all in a same way. With time, more products were added by the companies and the delivery pattern was also diversified in order to gain competitive advantage. The struggle with data was not limited to only businesses and companies but also with the Research and Development (R&D) organizations to get computing resources to run the models or to process images and other sources of data.

In fact, data has become the most complex part of businesses. With the success of Internet, generation of data is very high. The source of generation of data is not only laptops or computers, but by machines such as sensors and the more powerful mobile devices with ubiquitous access to global networks.

After Internet, big data is another buzz word that has received so much hype as broadly and as quickly. Few years ago, it was completely an unknown term, which is now the most talked about thing. Thanks to technology, a huge amount of data is being processed by the systems every day. This data is generated across the globe at an exceptional rate. Alone, social media activities generate astounding number of data every minute. If we talk about the most popular and active social network, Facebook, which generates most of the social media data adds up to 250 million posts every hour which means more than four million posts every minute. The data which is accumulating so fast is termed what we call as ‘*Big Data*’. The word Big Data has become a part of our lives, just like the Internet. From online shopping to searching videos on demand, every activity is generating data which plays an important role in our lives. This also shows that Big Data is also growing very fast. According to IBM, 90% of the data has been generated in the past two years.

When it comes to big data, the size of the data is the most talked about. But the other attributes which form the basis of big data are data variety and data velocity. The important terms in Big Data are the three Vs which form the comprehensive definition of the term and broken the myth about big data as just the size (or Volume). These three Vs are: Volume, Variety and Velocity.



*Fig. 12.1 The Three Vs of Big Data*

**Volume:** In simple terms, volume implies ‘How much square’. The volume or size is an important attribute of big data. The data so generated reaches an incomprehensible proportion. For example, consider a simple instance of storing photographs by the Facebook users. The statement has no impact until we realize that Facebook users are more than the population of China and hence Facebook is storing almost 250 billion of images. That means, if we talk about volume in big data we are actually talking about insanely huge amount of data which is still growing as we are adding connected sensors to almost everything. Also, the use of smartphones and hence a world of connected apps is adding a lot of data to store. For example, an app, ‘Todoist’ is an app for managers to manage their to-do-list. According to Android Play, it has around 10 million active installs. Apart from these sources of data, there is an internally collected enterprise data ranging from security to energy industry to healthcare and many more. The data is being generated and stored by all these industries and that is the volume vector of big data.

**Velocity:** This term defines the rate at which the data is being generated. Velocity also underscores how fast the data is being processed. The Internet and mobile era has changed the way we do the business, i.e. the way the product is delivered, the way the product is consumed and the way the services are provided. The data flows back to the providers and enables them to aggregate the history of interaction and every moment of customer’s action. Even the social media is playing an important role in increasing the amount of data every second. As a matter of fact, the users on Facebook alone are uploading more than 900 million photos each

## NOTES

## NOTES

day and these photos are being handled and processed by Facebook at that rate. Consider another example, where the campaign is run through an Internet to understand the feeling of the people about your candidate. This results in a data stream coming at a great speed and the duration of act on the basis of these data stream is very low.

Even the new technology, Internet of Things (IoT) has allowed more sensors to be connected across the globe, which are transmitting tiny bits of data at a near constant rate. This IoT is converting into IoE (Internet of Everything) which is also increasing the number of units and hence the flow of data.

**Variety:** In the above two Vs, we have talked about likes, photograph images, sensors and so on. The forms of each type of data are different from the other. This data unlike conventional data (in the form of rows and columns and database) differ from application to application and is in an unstructured form. This means these types of data cannot be fit into excel spreadsheet or traditional database. The sources of these data are different. The data can be originated internally or externally and can come from different sources such as log data and transaction, images, videos and audios from various applications as unstructured data, database table as structured data, XML data as semi-structured data. The formats of data are incredibly varied as there is a shift from completely structured data to increasingly more unstructured data.

To process the three Vs of data, it would take library of books for practitioners to define various methods. So we may say that when we consider the data which goes beyond basic baskets and when data is in epic quantity with insane flow and wide range, this means we are talking about big data.

To conclude, we may say that big data is a term used to represent data which is large in size with various forms, which is beyond the capacity of traditional database to store, manage or process. The data is captured from various sources such as sensors, devices, video, audio, social media, log files and the list is on, and is real time in nature.

The fourth V is also gradually emerging as the big data is gaining its momentum and this is known as Value.

Thus, we can summarize the 4 Vs of big data as it is important to process the required volume of data with appropriate velocity by the system and should be capable enough to handle wide variety of data formats. If all these elements are put inline, there is an opportunity to extract real value from this data after analyzing it. It may provide a competitive gain to the business if companies are able to manage all these four Vs effectively and efficiently.



**Fig. 12.2** Data Generated Every Minute

**Source** 1: <http://www.zdnet.com/article/volume-velocity-and-variety-understanding-the-three-vs-of-big-data/>

2: <http://wersm.com/how-much-data-is-generated-every-minute-on-social-media/>

### 12.2.1 Nature of Data

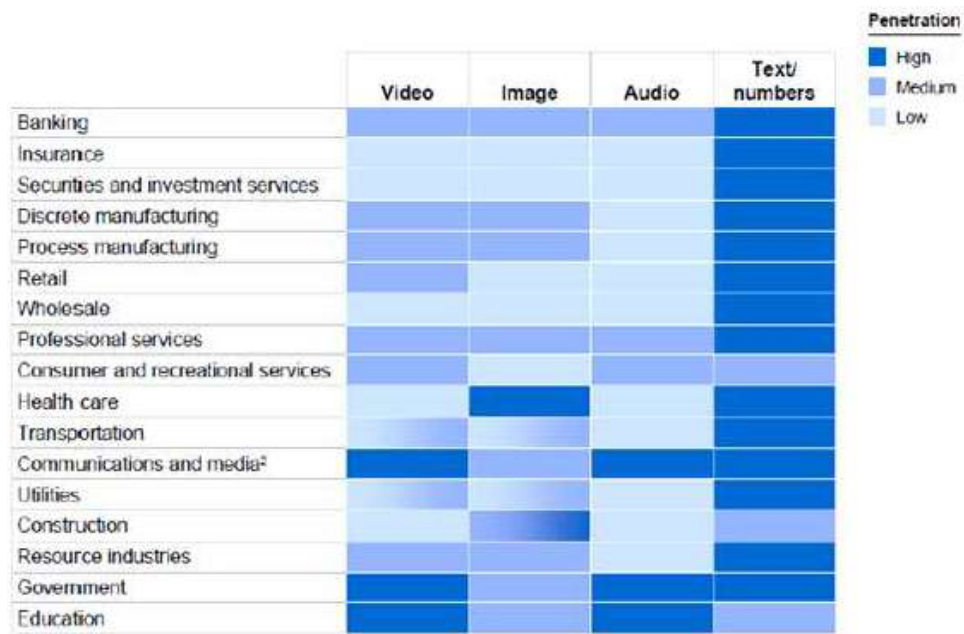
We have talked about data in the above sections and have learnt about the size of data available today. We will now understand that the size of the data is huge and it is still growing exponentially with time. We have seen above some of the statistics and have realized that Facebook alone is generating almost 500+ terabytes of data every day. Moreover big data has many sources. For instance a click of a mouse on any website may capture data in the form of Web log files. Machines such as smart meters are also generating data that stream data about electricity, water, gas consumptions details of customers. There is a huge amount of data generated by geospatial (e.g. GPS) created by smart phones to know the location of anything. Data is also generated from “Internet of Things”. This means the data so generated has many forms; it may be in the form of a comment, photo, video, likes, message exchange, etc. If we look at the figure below, we can conclude that various industries are generating data in various formats.

This data is knowledge oriented and makes sense in driving the modern organizations of the world. The data so generated over Internet across the globe can be classified in three different formats viz.

1. Structure
2. Unstructured and
3. Semi Structured

## NOTES

NOTES



**Structured Data**

Generally, the term structured refers to organized. In case of data also, the term has the same meaning. It reflects the data or information with a degree of organization. This data which is stored in databases in an ordered form for a seamless and easy search by simple, straightforward algorithms is known as structured data. That means any form of data that can be stored in relational database (traditional row database structure is termed as *structured data*. This data can be accessed and processed in a fixed format. This data has a high degree of organization and is readily searched by simple and straight forward search engine algorithms. Structured data is relatively easier to enter, store, query and even analyze and is defined in terms of field name and types. This type of data can be analyzed simply with the help of excel spreadsheet or Structure Query Language (SQL).

This type of data accounts almost 20% of the total existing data now days. The structured data is mostly useful in programming and computer related activities.

The sources of structured data are either human beings or machines. Example of machine generated data is sensors, web logs and financial systems including GPS data, medical devices, data generated by the usage statistics by servers, etc. The data generated by humans includes entering information (such as name, or any other personal information) in a computer system. Every move of human on any website (click on any link in any website, any move in a game) is a data generation activity and is being captured by the companies to identify the behavior of the customer that is useful in taking any decision. Here the data generated is a human generated data which is in a structured format. Structured data is always a code data that is helpful in a timely compilation of information.

## Unstructured Data

Whereas structured data can be presented in the traditional row-column format, there is some data which has no clear format or pre-defined data model and is not possible to be stored in row-column format. That means, unstructured data is everything else which has no pre-defined models. There is an immense amount of data and material which cannot be fit into a firm's organization of information through structured data. Until recently, this type of data was stored in the form of paper to supplement the structured data. But with the advancement of computing processing, lowered data storage cost and the new data format, the unstructured data concept evolved. This data has its own internal structure which is not confine to only database or spreadsheet. This type of is also increasingly at the faster pace and is available in the form of complex data sources such as email, multimedia content, sales automation, social media and customer service interaction. Indeed, many of the data generated today is unstructured in nature rather than structured data. Until recently most of the data was presented in a structured format, but now unstructured data makes almost 80% of the data generated toady and is growing at the rate of 65% per year. Earlier, nothing much could be done with this type of data except for storing it or for manual analysis. Like structured data, unstructured data can also have two sources of generation. One is human-generated and another is machine-machine. All the images from satellites, data generated from scientific experiments, digital surveillance, sensor data (traffic, weather etc.) and data captured through radar using various technologies are all considered to be unstructured data. Unstructured data generated by humans contributes to the major source of unstructured data. This includes data generated on social media, data generated over mobiles, communication and the various types of contents present over websites. That means the photo or image uploaded, comments and likes made on Facebook, the tweet we make on Twitter, the videos uploaded on You tube, the various forms of messages we exchange through different apps present in the mobile devices are all contributing to a huge heap of unstructured data.

So the basic difference between the two types of data can be presented in the following table:

| Feature         | Structured Data                                                                                                                                                         | Unstructured Data                                                                                                                                                                                                                                   |
|-----------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Characteristics | <ul style="list-style-type: none"> <li>• Pre-Defined Data models</li> <li>• Generally the data is of text form only</li> <li>• Easy and convenient to search</li> </ul> | <ul style="list-style-type: none"> <li>• No pre-defined data model</li> <li>• The format of data is largely images, sounds, video, or some other forms. It can be text also</li> <li>• It is a time consuming process to search anything</li> </ul> |

## NOTES



**NOTES**

|                                   |                                                                                                                                                                     |                                                                                                                                                                                   |
|-----------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Storage                           | <ul style="list-style-type: none"> <li>• RDBMS (Relational Databases)</li> <li>• Data warehouses</li> </ul>                                                         | <ul style="list-style-type: none"> <li>• Applications</li> <li>• NoSQL database</li> <li>• Data warehouses</li> <li>• Data lakes</li> </ul>                                       |
| Source of generation              | <ul style="list-style-type: none"> <li>• Machine and human, both</li> </ul>                                                                                         | <ul style="list-style-type: none"> <li>• Machine and human, both</li> </ul>                                                                                                       |
| Applications from where generated | <ul style="list-style-type: none"> <li>• Reservation systems (Airlines, Railways etc.)</li> <li>• CRM</li> <li>• ERP</li> <li>• Inventory control</li> </ul>        | <ul style="list-style-type: none"> <li>• Emails</li> <li>• Social media</li> <li>• Word processing</li> </ul>                                                                     |
| Examples                          | <ul style="list-style-type: none"> <li>• Dates</li> <li>• Phone numbers</li> <li>• Credit card numbers</li> <li>• Customer names</li> <li>• Transactions</li> </ul> | <ul style="list-style-type: none"> <li>• Text files</li> <li>• Reports</li> <li>• Emails</li> <li>• Images</li> <li>• Videos</li> <li>• Audios</li> <li>• Surveillance</li> </ul> |

**Semi-Structured Data**

With invent in technology, the data generated in not in rigorously structured. Such data is prevailing day by day. This type of data can be called as semi-structured data. Semi-structured data lies in between the structured and unstructured data. It has characteristics of both structured and unstructured data. This type of data may not have highly organized structure and hence does not have sophisticated access for analysis purpose, but consists of information associated with it such as metadata tagging that helps elements contained to be identified and addressed. For example, consider any X-ray that consists largely of many pixels which is highly unstructured data. The way this X-ray can be searched and analyzed is not same as compared to the data contained in the relational database, since the search is based on identifying pixels within the image. However, some information in the form of keywords, known as metadata, can be added that helps analyzing unstructured data. Metadata is a file that contains information about data and represents the content of the document that makes the data to be traceable when searched for those terms. This makes the nature of the data as semi-structured. The information generally consists of how data was created, the purpose of creating it, the file size, the author, the sender and receiver, etc. Thus most of the semi-structured data can be searched through their metadata. Since, the document still does not have any organized structure of a database, it cannot be considered as structured data. The base line is that the nature of data generated today is semi-structured which is going to stay cannot be ignored as it provides a potential value to any business that can handle and analyze such data well.

In real sense, today there is an overlap between the boundaries of the three types and is collectively called as data continuum. The percentage of unstructured

and semi-structured data is going to increase after the concepts like artificial intelligence (AI), machine learning (ML) and internet of things (IoT) gain real thrust in the marketplace.

---

## 12.3 TRADITIONAL VS BIG DATA APPROACH

---

## NOTES

Data is important and is collected and analyzed to create information for decision making. ‘Analytics’ has always been an important tool required for business in order to gain in depth insight of data for different views of information to critically monitor and analyse the heap of data. Traditionally, Relational Database Management System (RDBMS) and data warehouses were used by business and organizations to store and analyze the data. These systems were designed primarily for structured data. The features of the structured data in conventional system are:

- Records are stored in the form of table. The table has row, column form with clearly defined fields with name and relationship among fields.
- Schema-on-write, where data can only be written to the disk once it has been validated against a schema. This means, storing data in a clearly defined structure requires a significant amount of analysis, design and effort which can increase the time of business value that is realized from the data.
- A design that loads the data from the disk into the memory to be processed by applications.
- Use of Structured Query Language (SQL), to manage and accessing the data.
- Use of 8K or 16k block sizes relational and warehouse database systems which reads and loads the data into memory in block sizes to process the data through applications.

Since the data is growing at an insane pace, data being in semi-structure and unstructured format which do not have pre-defined data model or order, it becomes challenging to unearth all the hidden pattern and information through a vast amount of data. The main challenges involve:

- Since the analysis is done on well understood and known data, this may not work on unstructured data in an efficient way.
- As discussed above, traditional way of storing data is in the row and column form with clearly mentioned fields and relationships created inside the system. Analysis is also done based on it, which might not be adequate for a vast amount of data and hence not efficient for analysis.
- Since parallel processing is required in big data, this is achieved through costly hardware like Massive Parallel Processing (MPP) systems in traditional analytic systems.

## NOTES

- Also, relational and warehouse database system is used, reading the data in block size is extremely inefficient to process large volumes of data.
- Scheme-on-write features may increase the analysis time in case of big data.

Apart from the above mentioned challenges, there might be some other challenges such as:

- *Data Challenges* that includes three Vs, data discovery and comprehensive and scalability.
- *Process Challenges* that includes data capturing, synchronization of data from different resources, transformation of data into compatible form of data analysis, understanding and visualizing data analysis results on mobile devices.
- *Management Challenges* that includes privacy, security, ethics and governance

### Check Your Progress

1. What are the three Vs of Big data?
2. What are the three formats of data generated over internet?

---

## 12.4 ANSWERS TO CHECK YOUR PROGRESS QUESTIONS

---

1. The three Vs of Big data are: Volume, Variety and Velocity.
2. The data so generated over Internet across the globe can be classified in three different formats viz.
  - (i) Structure
  - (ii) Unstructured and
  - (iii) Semi Structured

---

## 12.5 SUMMARY

---

- Data has always offered greatest benefits, but managing, organizing and analyzing the data is always a challenge to the organizations across all industries, no matter what the size of an organization is.
- The data which is accumulating so fast is termed what we call as '*BigData*'. The word Big Data has become a part of our lives, just like the Internet. From online shopping to searching videos on demand, every activity is generating data which plays an important role in our lives.

- The volume or size is an important attribute of big data. The data so generated reaches an incomprehensible proportion.
- Velocity defines the rate at which the data is being generated. Velocity also underscores how fast the data is being processed. The Internet and mobile era has changed the way we do the business, i.e. the way the product is delivered, the way the product is consumed and the way the services are provided.
- Generally, the term structured refers to organized. In case of data also, the term has the same meaning. It reflects the data or information with a degree of organization. The data which is stored in databases in an ordered form for a seamless and easy search by simple, straightforward algorithms is known as structured data.
- Structured data can be presented in the traditional row-column format, there is some data which has no clear format or pre-defined data model and is not possible to be stored in row-column format. That means, unstructured data is everything else which has no pre-defined models.
- Semi-structured data lies in between the structured and unstructured data. It has characteristics of both structured and unstructured data. This type of data may not have highly organized structure and hence does not have sophisticated access for analysis purpose, but consists of information associated with it such as metadata tagging that helps elements contained to be identified and addressed.

## NOTES

---

### 12.6 KEY WORDS

---

- **Internet of Everything (IoE):** The Internet of Everything (IoE) is a broad term that refers to devices and consumer products connected to the Internet and outfitted with expanded digital features.
- **Relational Database Management System (RDBMS):** RDBMS stands for Relational Database Management System. An RDBMS is a particular type of DBMS that uses a relational model for its databases.

---

### 12.7 SELF ASSESSMENT QUESTIONS AND EXERCISES

---

#### Short Answer Questions

1. What do you understand by the term Big data?
2. Discuss the significance of big data over traditional approach.

### Long Answer Questions

1. Explain the three Vs of big data.
2. What are the different forms of data? Explain.
3. Differentiate between structured and un-structured data.

### NOTES

---

### 12.8 FURTHER READINGS

---

Han, Jiawei, Micheline Kamber and Jian Pei. 2011. *Data Mining: Concepts and Techniques*, 2nd edition. Amsterdam: Elsevier.

Pujari, Arun K. 2010. *Data Mining Techniques*, 2nd edition. United States: Universities Press.

Anahory, Sam and Dennis Murray. 1997. *Data Warehousing in the Real World: A Practical Guide for Building Decision Support Systems*, 1st edition. Boston: Addison Wesley.

Witten, I. H. and Eibe Frank. 2005. *Data Mining: Practical Machine Learning Tools and Techniques*, 2nd edition. Amsterdam: Elsevier.

Soman, K. P., Shyam Diwakar and V. Ajay. 2006. *Insight Into Data Mining: Theory and Practice*. New Delhi: PHI.

---

## UNIT 13 TECHNOLOGIES

---

### Structure

- 13.0 Introduction
- 13.1 Objectives
- 13.2 Technologies Available for Big Data
- 13.3 Answers to Check Your Progress Questions
- 13.4 Summary
- 13.5 Key Words
- 13.6 Self Assessment Questions and Exercises
- 13.7 Further Readings

### NOTES

---

### 13.0 INTRODUCTION

---

In the previous unit, you have learnt about the Big data and its types. In this unit, you will learn about the various technologies of Big data such as R, python, MNIME etc.

---

### 13.1 OBJECTIVES

---

After going through this unit, you will be able to:

- Explain the various technologies of Big data and their features

---

### 13.2 TECHNOLOGIES AVAILABLE FOR BIG DATA

---

For business organizations, the data is an asset and need to be analyzed. The data analytics tools are being used to automatically collect, clean and analyze data, to predict, improve accuracy of prediction and even refine the models. There are a number of analytical tools available which are powerful, and also some of them are free and open source that helps enhance business.

Let us first look at some of the analytical tools.

#### R

R is the most popular analytical tool in the industry today. R is a software programming environment for statistical computing and graphics supported by the R Foundation for Statistical Computing. It compiles and runs on variety of operating system platforms such as UNIX, Windows, MacOS.

R was first implemented by Robert Gentleman and Ross Ihaka in the early 90s. They both were working as faculty members at the University of Auckland. The first release of R came in February 2000. R project has been managed by the

## NOTES

R Core Group. R is a programming language which allows you to perform data analysis by writing functions and scripts. R has become robust over the years that can handle huge data set in a much better way.

R provides a wide variety of statistical techniques such as linear and nonlinear modeling, time series analysis, classification, classification statistical tests, clustering etc. and graphical techniques. It is easy to use and is versatile.

There are many R packages available which are the collection of data, compiled codes and R functions. There are many default packages that come with R and many other packages can be installed to extend the capabilities of R. The total number of packages available in R is over 8000 with 1800 new packages introduced between April 2015 and April 2016.

R is integrated software that includes manipulation, calculation and graphical display of data with the following characteristics of R environment:

- Effective data handling and storage facility.
- Different operators for calculations on array, matrices in particular.
- Collection of intermediate tools for data analysis purpose.
- Programming language which is well-developed, simple and effective that includes conditional, loops, user-defined recursive functions and input-output facilities.

R is primarily used when data analysis or computing is required to be performed standalone on individual servers. R can become even part of Big Data Analytics. To start with R one needs to download RStudio IDE.

### Python

Python is not a new programming language and has been a favourite programming language for long. It is an easy to use language and with the development of statistical and analytical libraries like pandas, matplotlib and numpy, etc. it has developed into a powerful analytical tool. There are ample mathematical and statistical functions that are being offered in python. Due to this programmers and technological people are moving into analytics using python and hence becomes an obvious choice.

Python was introduced in the year 1991 by Guido Van Rossem and the main focus was on productivity and code readability. It is known for its flexibility. Like R, Python also has packages to support data analytics. Python has become a choice for data science and the expectation is growing.

Unlike R, Python is used when data analysis needs to be incorporated in web based apps or when statistical code needs to be combined with production database.

## KNIME

KNIME (pronounced as “nime”) stands for the Konstanz Information Miner, is another data analytical tool for data-driven innovation, which is a visual programming tool, which allows manipulation, analyzing and modeling of data in an intuitive way and is supported by KNIME.com. It was developed at University of Konstanz in Germany. It allows discovering the potential hidden in data, mining for insights and even predicts the future. KNIME is fast to deploy and easy to scale tool integrating various components for data mining and machine learning. Drag and drop connection points between activities are available that avoids the writing blocks of code. KNIME analytic platform is a perfect toolbox for data scientists with 1500 modules available, hundreds of which are ready-to-run examples and wide range of advance algorithms.

KNIME analytical platform is written in Java and built on Eclipse to leverage the Eclipse’s module extension capabilities with the help of plug-ins and connectors. KNIME has integrated with the power of Apache Hadoop and Apache Spark in KNIME Big Data Extension to provide an easy-to-use KNIME Analytical Platform and KNIME Server. It has also integrated with many tools including R/Python (legacy Scripting/ code) that allows the reusability of code (expertise), graphically documented and shared among data scientists.

KNIME features include:

- Sophisticated data handling i.e. automatic caching of data done intelligently in the background while maximizing throughput performance for scalability
- Intuitive user interface
- Import/export of workflows
- Well-defined API for plugin extensions to support high and simple extensibility.
- Parallel execution on multi-core systems

The KNIME resources are available at:

- Web pages (documentation): [www.knime.org](http://www.knime.org), [tech.knime.org](http://tech.knime.org), [tech.knime.org/installation-0](http://tech.knime.org/installation-0)
- Downloads: [knime.org/download-desktop](http://knime.org/download-desktop)
- Community forum: [tech.knime.org/forum](http://tech.knime.org/forum)
- Books and White papers: [knime.org/node/33079](http://knime.org/node/33079)

## Tableau Public

Tableau is a simple and intuitive analytical tool for data visualization and is considered to be exceptionally powerful tool for business intelligence. It is an easy learning tool to explore data with great visualization and dashboards. It also allows you to

## NOTES



## NOTES

understand the real time data, lets you share the work with others and hence imparts colorful life to data.

The power of Tableau for Business Intelligence can be considered on the following factors:

- **Technical knowledge:** To realize the power of Tableau there is no need of prior technical knowledge.
- **Answer to any problem:** Any business problem can be answered using the available innumerable Tableau features
- **Data visualization:** Strong data visualization capabilities are the biggest strength making Tableau most powerful tool for business.
- **Sharing insights:** Tableau provides capability to gain business insights on any device.

Although, Tableau is a proprietary tool and is not an open source, and the cost associated is not low as well, there is available free version of Tableau called Tableau Public. There are two products available: Tableau Public Server and Tableau Public Desktop.

### Pig and Hive

Other analytical platforms for managing large set of data are Apache Pig and Apache Hive which are high level programming languages. These programming languages are integrated in Hadoop ecosystem and are used to handle data which includes data processing, data analysis and infer the business decision from the analysis. We will learn about Hadoop in the later units, which is important for Big Data Analytics.

Apache Pig is a high level platform that uses a scripting language called *Pig Latin* to create programs that can run on Hadoop. Apache Hive is a data warehouse system that is used to query and analyze large dataset stored in Hadoop files. We will look at these later in the units.

### RapidMiner

RapidMiner is an analytical tool that provides data visualization and processing, statistical modeling deployment and evaluation and specifically predictive analysis used in machine learning procedures and data mining techniques. This tool is considered to be in the top list of Big Data Analytics tool that is written in Java programming language. Models and algorithm from WEKA and R scripts makes it more powerful in providing learning schemes.

### SAS

Created by Anthony James Barr at North Carolina State University, SAS (Statistical Analysis System) is another analytical tool for business intelligence for reporting, data mining, predictive modeling and analyzing using powerful visualization and interactive dashboard. SAS makes understanding of complex data easy. It also

helps in identifying unprecedented pattern, trend analysis, correlations, spotting outliers, predicting future trends etc.

SAS visual analytics provides a powerful insight and analyze the huge data set in an easy manner. Output reliability can be ensured through SAS and the data sources can be traced back through the results as well.

## NOTES

### Check Your Progress

1. What does R provides?
2. Who introduced the Python?

## 13.3 ANSWERS TO CHECK YOUR PROGRESS QUESTIONS

1. R provides a wide variety of statistical techniques such as linear and nonlinear modelling, time series analysis, classification, classification statistical tests, clustering etc. and graphical techniques.
2. Python was introduced in the year 1991 by Guido Van Rossem and the main focus was on productivity and code readability.

## 13.4 SUMMARY

- R is the most popular analytical tool in the industry today. R is a software programming environment for statistical computing and graphics supported by the R Foundation for Statistical Computing.
- R provides a wide variety of statistical techniques such as linear and nonlinear modelling, time series analysis, classification, classification statistical tests, clustering etc. and graphical techniques.
- R is primarily used when data analysis or computing is required to be performed standalone on individual servers.
- Unlike R, Python is used when data analysis needs to be incorporated in web based apps or when statistical code needs to be combined with production database.
- KNIME analytical platform is written in Java and built on Eclipse to leverage the Eclipse's module extension capabilities with the help of plug-ins and connectors.
- Apache Pig is a high level platform that uses a scripting language called *Pig Latin* to create programs that can run on Hadoop.
- RapidMiner is an analytical tool that provides data visualization and processing, statistical modelling deployment and evaluation and specifically predictive analysis used in machine learning procedures and data mining techniques.

NOTES

---

## 13.5 KEY WORDS

---

- **R:** It is a leading programming language of data science, consisting of powerful functions to tackle all problems related to Big Data processing.
- **Apache Pig:** It is a high-level platform for creating programs that run on Apache Hadoop.

---

## 13.6 SELF ASSESSMENT QUESTIONS AND EXERCISES

---

### Short Answer Questions

1. What are the characteristics of R?
2. What is RapidMiner?

### Long Answer Questions

1. What are the features of KNIME?
2. Write a note on the following:
  - (i) R
  - (ii) KNIME
  - (iii) SAS
  - (iv) Pig and Hive

---

## 13.7 FURTHER READINGS

---

Han, Jiawei, Micheline Kamber and Jian Pei. 2011. *Data Mining: Concepts and Techniques*, 2nd edition. Amsterdam: Elsevier.

Pujari, Arun K. 2010. *Data Mining Techniques*, 2nd edition. United States: Universities Press.

Anahory, Sam and Dennis Murray. 1997. *Data Warehousing in the Real World: A Practical Guide for Building Decision Support Systems*, 1st edition. Boston: Addison Wesley.

Witten, I. H. and Eibe Frank. 2005. *Data Mining: Practical Machine Learning Tools and Techniques*, 2nd edition. Amsterdam: Elsevier.

Soman, K. P., Shyam Diwakar and V. Ajay. 2006. *Insight Into Data Mining: Theory and Practice*. New Delhi: PHI.

---

# UNIT 14 HADOOP

---

## Structure

- 14.0 Introduction
- 14.1 Objectives
- 14.2 Introduction to Hadoop
- 14.3 Core Hadoop Components
- 14.4 Hadoop Ecosystem and Architecture
  - 14.4.1 Physical Architecture of Hadoop
- 14.5 Hadoop Limitations
- 14.6 Answers to Check Your Progress Questions
- 14.7 Summary
- 14.8 Key Words
- 14.9 Self Assessment Questions and Exercises
- 14.10 Further Readings

## NOTES

---

### 14.0 INTRODUCTION

---

Apache Hadoop is an open-source software framework used for distributed storage and processing of datasets of big data using the MapReduce programming model. It consists of computer clusters built from commodity hardware.

Hadoop was created to handle and enhance the capability of big data. Big Data can be used as an opportunity with the evolution of technology. It was never expected of using commodity hardware to store and manage the data, until the big data came into existence. This option is reliable and feasible as compare to the other costly servers. With time, benefits associated with Big Data Analytics came into picture and organizations started realizing it. However, the challenge that was associated with Big Data Analytics was accessing and processing speed along with the volume (huge data) and variety (different formats of data).

Apache Hadoop is an important framework to work with big data, which is scalable and can upgrade itself from working on a single node to thousands of nodes in a seamless manner, without facing any issues. It allows handling a large set of data by running the applications on the basis of MapReduce to process the data parallel in order to accomplish the statistical analysis on a large set of data. The framework of Hadoop is based on Java programming. Thus, Hadoop is a framework that allows us to store big data in a distributed environment so that data can be processed parallel.

## NOTES

---

## 14.1 OBJECTIVES

---

After going through this unit, you will be able to:

- Explain the Hadoop distributed file system
  - Describe the process of scaling out in Hadoop
  - Discuss the three main components of HDFS
  - Explain Hadoop ecosystem
  - Discuss the limitations of Hadoop
- 

## 14.2 INTRODUCTION TO HADOOP

---

Apache Hadoop was created to solve the major issues and to enhance the usage of big data. The story of Hadoop begins in the year 1997 when a Yahoo! Employee, Doug Cutting started writing the first version of *Lucene*, a text search library used for the fast search of web pages. Even after years, he experience ‘Dead Code Syndrome’ and he open sourced to Source Forge. Lucene was converted to Apache Lucene in 2001, which was focusing on indexing the web pages. Then a graduate of Washington University, Mike Cafarella, joined him to index the entire web as a result of a new sub-project of Lucene came up with the name Apache Nutch. Meanwhile, Cutting and Cafarella were facing some issues with the existing file system. These issues are listed below:

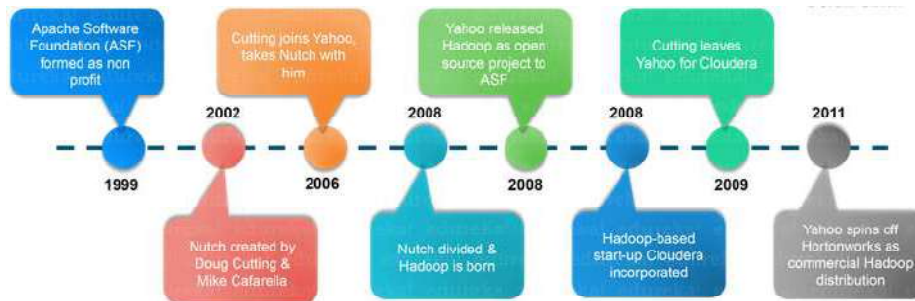
- The existing file system had no tables and columns, i.e., it was schema less
- It was not durable (once written should never lost)
- The system was not capable of handling any failure of components (CPU, Network, Memory)
- The system was not capable of re-balancing (disk space consumption) automatically.

The duo was astonished, when Google published a paper on the Google File System (GFS), where they found the solutions to the problem they were facing. Using this paper on GFS and integrating Java, they developed their own file system and called it as Nutch Distributed File System (NDFS). But for this file system also, the problem of durability and fault tolerance still persist and hence they came up with the idea of distributed processing. They divided the file system into chunks of 64MB and stored each element on 3 different nodes (replicate the data). Now, they wanted an algorithm for NDFS so as to integrate parallel processing, i.e., they wanted to run multi nodes at the same time. Again in 2004, Google published a paper titled ‘Map Reduce – Simple Data Processing on Large Clusters’ which solved the problem of parallelism, distribution and fault-tolerance.

Again in the year 2005, Cutting reported the integration of Map Reduce into Nutch and in year 2006, he separated GFS and Map Reduce from the Nutch code base and called it as *Hadoop*. The HDFS, Map Reduce and Hadoop Common (Core Libraries) were included in Hadoop. Gradually, companies like Twitter, Facebook, and LinkedIn also started using Hadoop. However, Hadoop

was still a sub-project of Lucence till the year 2008. So, it was separated from Lucence by Cutting and licensed it under Apache Software Foundation. When different other companies also started facing problems with their file system, they started experimenting with Hadoop and some sub-projects like Hive, PIG, HBase, Zookeeper, etc., were also created.

## NOTES



*Fig. 14.1 Evolution of Hadoop*

The Hadoop also has an interesting story to get its name. Cutting's son, who was two years old, just started speaking and called one of his stuffed yellow elephants as 'Hadoop'.

'Being a guy in the software business, we're always looking for names,' Cutting said. 'I'd been saving it for the right time.'

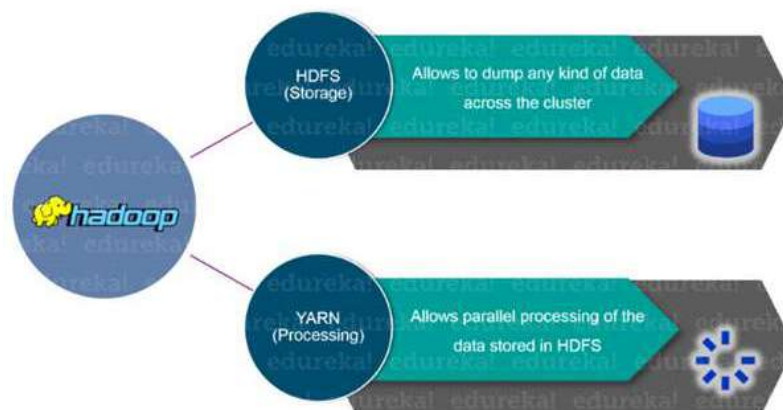
There are broadly two components of Hadoop:

1. **Hadoop Distributed File System (HDFS):** This is actually storage for the system and can store data of various formats across a cluster. We will discuss this component in detail in the next section.
2. **YARN:** This acts as a processing unit of Hadoop that is capable of parallel processing of data stored across the HDFS.

### The Hadoop Distributed File System (HDFS)

Before discussing HDFS, let's look at some of the statistics related to HDFS:

- In 2012, Facebook stored 21 *petabytes* of data, claiming to have one of the largest HDFS cluster.



*Fig. 14.2 Hadoop Components and Framework*

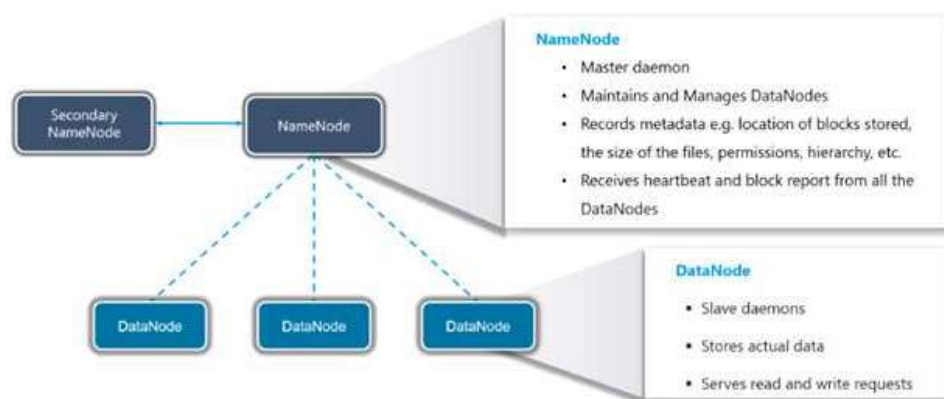
## NOTES

- Yahoo! has 40,000 servers with more than 100,000 CPUs running Hadoop. Yahoo! stores 455 PB of data in HDFS with the biggest Hadoop cluster running 4,500 nodes.
- Most of the big names in Fortune 50 companies started using Hadoop by 2013.

We will discuss about HDFS, but before that let's take a quick run to understand Distributed File System (DFS).

DFS is a way of managing data (files and folders) across multiple servers (computers). It is a system that allows data to be stored over multiple machines (called nodes) in a cluster. This also allows multiple users to access the data. Thus, the basic principle is same as that of the file system that is available in your personal machines (for example, New Technology File System, NTFS, for windows or Hierarchical File System, HFS, for Mac systems), with the only difference that DFS allows you to store data at multiple machines at different locations rather than storing at a single machine. Although, in this architecture (DFS), the files are distributed across the network, the data are organized and displayed in such a way that the user feels like all the data are being stored on a single machine. The underlying concept of HDFS is similar to DFS.

Hadoop is ideal for storing huge amount of data in terabytes and petabytes. The Hadoop Distributed File System (HDFS) is the primary storage system used in Hadoop application that is designed to run on commodity hardware and to store very large datasets reliably and to stream that data at high bandwidth for user applications. The HDFS connects the commodity hardware or personal computers connected over a cluster where the data files are stored in a distributed manner. These are known as nodes Hadoop. Precisely, HDFS creates an abstraction of resources, i.e., the data is being stored across multiple nodes in a distributed manner, but virtually we can see HDFS logically as a single unit for storing data. This system allows access and storage of data as one seamless file system. The access to the files is done in streaming manner, i.e., applications or commands are directly executed using MapReduce (discussed later) processing models. Also, HDFS provides high throughput access to the datasets and is a fault tolerant. HDFS holds a huge amount of data and still provides easier access. The files are stored across multiple machines to accommodate such a huge data. It has master-slave architecture.



## NOTES

**Fig. 14.3** The Architecture of HDFS

The above figure depicts the architecture of HDFS. Here, there is a master node called as *NameNode* and slave nodes identified by *DataNode*. Both (*NameNode* and *DataNode*) are java processes that run on the machine when Hadoop software is installed.

The master node, *NameNode*, is responsible for storing the metadata about the HDFS files, i.e., the information about HDFS file such as file name, file permissions, file size, block, where the file is stored and the information about the replica of data nodes are kept. There are primarily two files associated with the metadata:

- **FSImage:** FSImage that consists of the complete state of the file system namespace since the start of the *NameNode*
- **EditLogs:** EditLogs which consist of all the modifications made to the file system in the most recent FSImage.

It also takes care of the other file operations such as opening, closing, renaming of the files or directories. The *DataNode* gets instruction from the *NameNode* about where to write data. The *NameNode* is also responsible for recording all the changes that happen to file system metadata, i.e., if any file is deleted in HDFS, the *NameNode* will record this change in EditLog file immediately.

In case of *DataNode* failure, the new *DataNodes* are chosen by *NameNode* for new replicas, balance disk usage and communicate about the traffic to the *DataNode*. The *NameNode* also receives all the block reports from all the *DataNodes* in the cluster in order to ensure that the *DataNodes* are alive.

In return, *DataNode* reports back to the *NameNode* about the block and the drive where it has written the data. This helps in creating a central repository. This information is duplicated into secondary *NameNode*. The *DataNodes* are actually slave nodes in HDFS and actually stores the data. The file is divided into blocks before it gets stored in HDFS. By default the block size is 64 MB. The data blocks present in *DataNode* are also replicated, default replication factor is 3, which can be configured based on the requirement. The failure rate of commodity



hardware is high which is being used by HDFS. So, if the data block is lost due to failure of any DataNode, HDFS has still a copy of this lost data block. This is the reason of duplicating the data block.

## NOTES

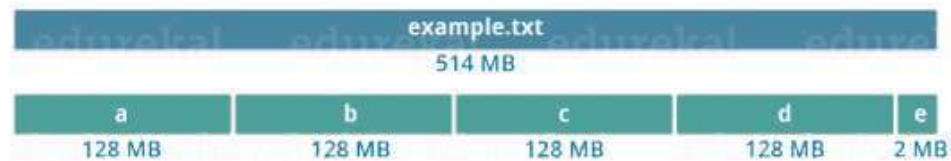
Apart from these two daemons, HDFS has a third process known as Secondary NameNode, which works concurrently with NameNode as a helper. The secondary NameNode is responsible for reading all the file systems and metadata from the NameNode RAM and writes back into the hard disk or the file system. It also combines the EditLogs with FsImage from the NameNode. This daemon is also responsible for downloading the EditLogs from the NameNode at regular intervals and applies to FsImage. The new NameNode is copied back to NameNode for the use whenever the NameNode is started next time. This daemon is also called as CheckpointNode as it performs regular checkpoints in HDFS.

### ***What are Blocks in HDFS?***

Since we have used the term blocks as the data is scattered in HDFS in blocks across the DataNodes, we will take a closer look at blocks and how they are formed?

The concept of blocks is similar to any other file system on your laptops or PCs, where data is stored as a collection of block. Blocks are the continuous location where the data is stored on the hard drive. In a similar fashion, each file is stored in blocks scattered throughout the Hadoop cluster in HDFS. The default size of each block is 64 MB in Apache Hadoop 1.x and 128MB in Apache Hadoop 2.x. This size can be configured according to one's requirement.

For example, a file of size 514 MB to be stored in a default configuration of block size 128MB. Then the file is stored in 4 blocks of 128 MB each, and the last block will be of size 2 MB only. That means the whole file is stored in total 5 blocks which indicates that it is not necessary to store each file in exact multiple of the configured block size, i.e., of size 128 MB or 256 MB. The reason for having such a huge size of blocks is obvious as HDFS talks about huge data sets in terabytes or petabytes of data. So, if we have small size of blocks then there are too many blocks and hence too much of the metadata and it is difficult to manage these numbers of blocks and their metadata.

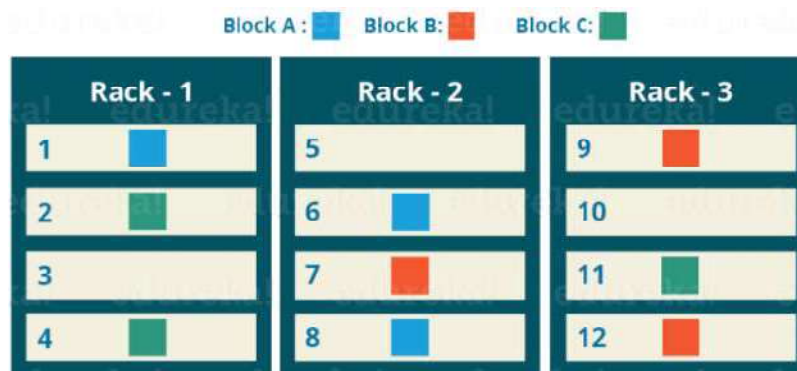


### ***Rack Awareness***

Now that we are aware of the fact that HDFS stores the file in replication. Let us see how HDFS does replication and what is rack awareness? Since HDFS stores the files in duplicate to avoid Fault-tolerance, NameNode makes sure not to store

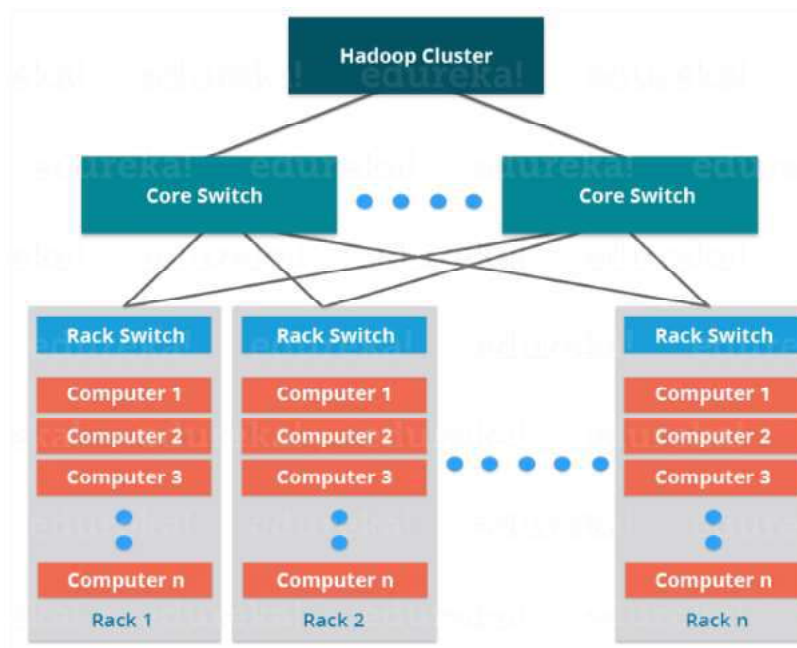
all the replicas on the single or same rack. Instead, a built in Rack Awareness Algorithm is followed to provide fault tolerance and to reduce latency. So, with default replication factor as 3, then this algorithm will store the first replica of a block on a local rack and the next two replicas will be stored in different rack and in different DataNodes within that Rack. If the replication factor is more than 3 then the replicas will be stored randomly on different DataNodes provided the same rack does not have more than two replicas, if possible. The following figure will explain the concept further. In the figure, there are three different racks Rack 1, Rack 2 and Rack 3 with four different DataNodes on each rack.

## NOTES



*Fig. 14.4 Rack Awareness Algorithm*

The following figure depicts the actual Hadoop production cluster.



*Fig. 14.5 The Hadoop Production Cluster*

## NOTES

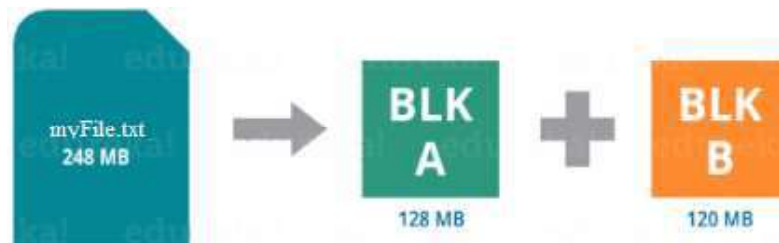
The advantages of using Rack awareness algorithm are as follows:

- **For improved network performance:** Since, the network bandwidth is greater between nodes in the same rack rather than nodes residing in different racks, this algorithm helps to reduce the write traffic in between the racks and hence improve the write performance. Also, the reading bandwidth is increased as the bandwidth of multiple racks is being used. The communication is directed through switches between the nodes on different racks.
- **Reliability:** The data is not lost even if the entire rack is failed due to power failure or switch failure as the algorithm suggests storing the replicas on different racks.

### HDFS Read/Write Architecture

HDFS follows write once and read many approach for the files stored in HDFS platform. Also, it does not allow editing the files once written. However, it allows to append the file by reopening the file.

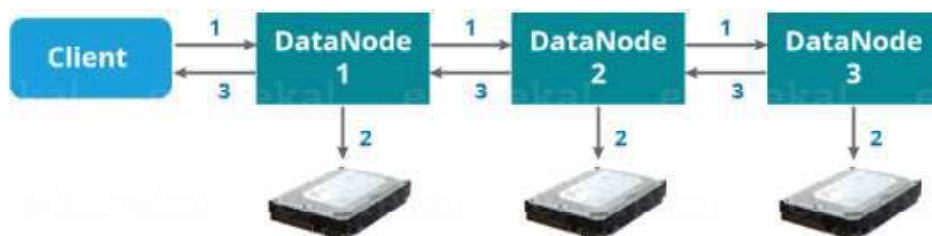
- **HDFS Write Architecture:** Let us understand how a file will be stored on HDFS system. Assume that there is a file named 'myFile.txt' which an HDFS client wants to write/store in the HDFS system. Also, assume that the size of the file is 248 MB. Also, assume that the block size configured for the HDFS system is 128 MB which is a default value. Thus, the file of the client will be divided into two blocks: Block A of 128 MB and Block B of 120 MB.



The protocol that will be followed in writing the file into HDFS is:

1. The HDFS client that wants to write the file will contact NameNode with a write request for writing two blocks: Block A and Block B
2. The permission will be granted to DataNode from NameNode with IP addresses of the DataNodes where the client is supposed to write the file eventually. The IP address is selected randomly based on replication factor, rack awareness algorithm and of course the availability of space.
3. If assuming the default replication factor as 3, the NameNode has to provide a list of 3 IP addresses of DataNode to the client. Obviously, the list has to be unique for each block.

4. The data blocks will be copied to three different DataNodes to maintain the consistent replication factor throughout the cluster.
5. In writing the whole data into blocks the following steps will take place.



## NOTES

- o Setup of pipeline
- o Data streaming and replication
- o Shutdown of pipeline or acknowledgement.
- **Setup of pipeline:** The client needs to confirm the readiness of the DataNodes present in each of the list of IP address, i.e., to confirm if the DataNodes are ready to receive the data. In doing so, a pipeline is created for each block by the client to connect the individual DataNode in the respective list for that block. The following steps are performed in order to create a pipeline (for Block A, for Block B it is similar):
  - o The client will select the first IP DataNode (for Block A, assume the list is DataNode1, DataNode 4 and DataNode6 and for Block B the list is DataNode 3, DataNode 7 and DataNode 9) and will establish a connection (TCP/IP).
  - o The first DataNode in the list will be informed by client to get ready to receive the block. The DataNode 1 will also be informed about the IPs of other two DataNodes (DataNode 4 and 6) where the block needs to be replicated.
  - o The DataNode 1 will connect and inform the next DataNode (DataNode 4) to be ready to receive the block. Also, DataNode 1 will give the IP address of the DataNode 6. The DataNode in turn inform DataNode 6 about the receiving of the block and hence to get ready for receiving the data. The acknowledgement of the readiness will traverse in the reverse direction, i.e., from DataNode 6 to 4 and then to 1.
  - o The DataNode 1 will inform the client about the readiness of all the DataNodes and finally a pipeline will be created between client and all the DataNodes to copy or streaming process of data.

NOTES

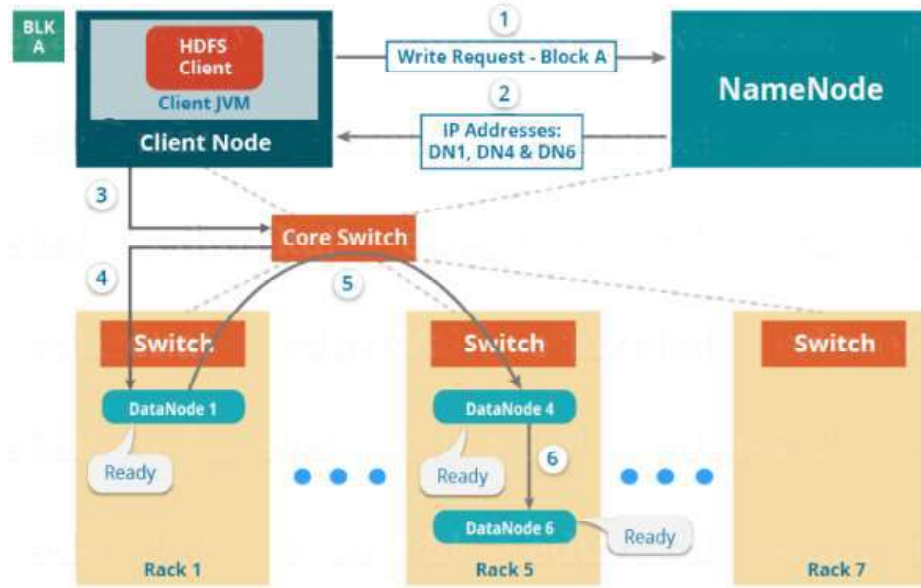


Fig. 14.6 Setting Up HDFS: Write Pipeline

- Data Streaming:** As soon as the pipeline is created, the data is pushed into the pipeline by the client. The block (A) will be copied to DataNode1 (only), and the replication will be done by DataNodes sequentially. That means, the client will write the block on DataNode 1. The DataNode 1 in turn will connect to DataNode 4 and the DataNode 1 will push the block in the pipeline and a copy of data is created on DataNode 4. Again, in similar fashion, DataNode 4 will copy the last replica of the block on to the DataNode 6.

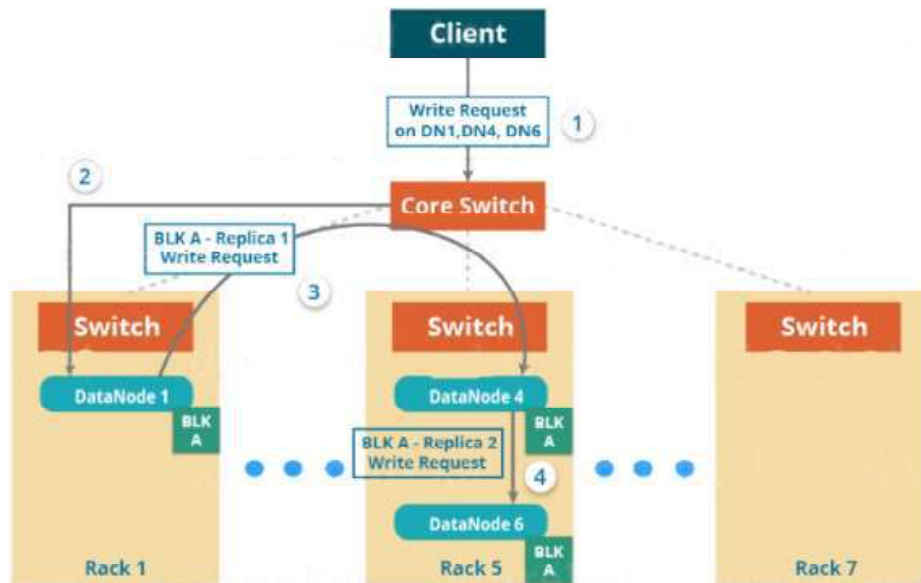
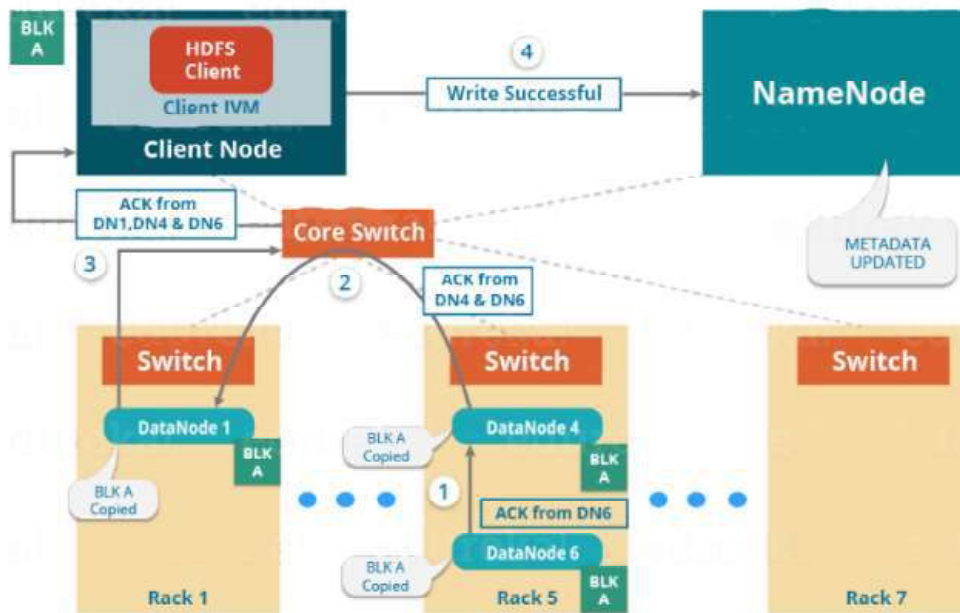


Fig. 14.7 HDFS: Write Pipeline

- Acknowledgement Stage (or Shutdown of Pipeline):** Once the block has been stored in all the DataNodes, an acknowledgement will happen to ensure client and NameNode about the successful writing of the data, after which the client will close the pipeline. The acknowledgement will happen in the reverse sequence. That is, acknowledgment will be sent from DataNode 6 to DataNode 4 and then to DataNode 1. The DataNode 1 will push the three acknowledgements (including its own) into the pipeline to reach to client. The client in turn informs the NameNode which then updates its metadata and finally the pipeline will be shut down.

## NOTES



*Fig. 14.8 Acknowledgment in HDF*

Similarly, Block B will also be copied into the HDFS cluster in parallel. The copy of Block A and Block B to their first DataNode respectively will be done simultaneously by the client. Thus, in this case, two pipelines will be created for each block and the process will happen parallel in the two different pipelines.

## NOTES

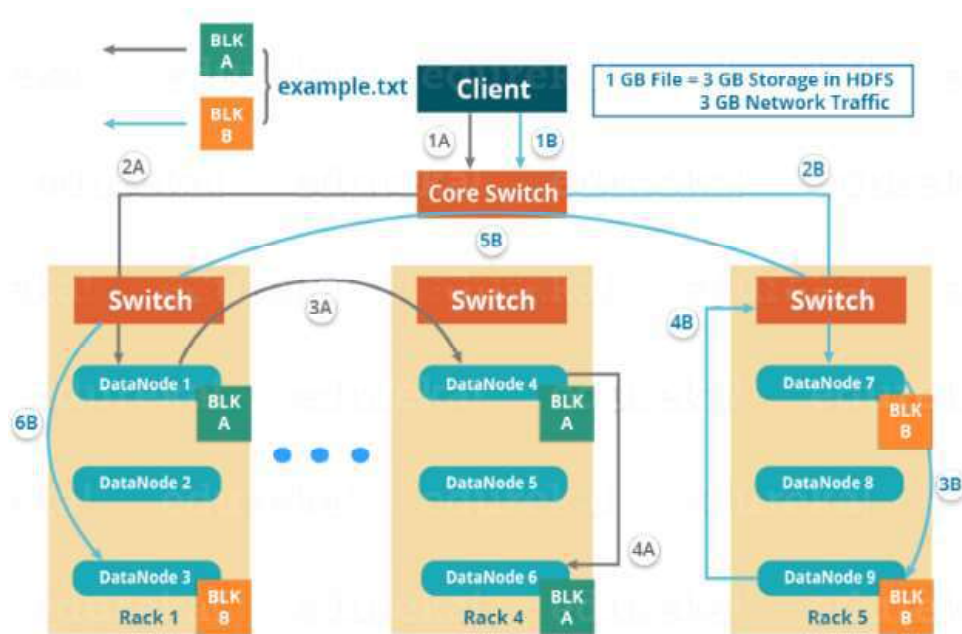


Fig. 14.9 HDFS Multi: Block Write Pipeline

In the above figure, there are two pipelines that are created for each block A and B. Following is the flow of operations in the pipelines:

For Block A: 1A -> 2A -> 3A -> 4A

For Block B: 1B -> 2B -> 3B -> 4B -> 5B -> 6B

- **HDFS Read Architecture:** This is comparatively easier architecture to understand. To read a file from HDFS system, say for example, the file myFile.txt, the following steps are performed:
  - o The client will connect with NameNode with a request for the block metadata for the file 'myFile.txt'.
  - o The NameNode will provide the information of the list of DataNodes where the blocks A and B of the file are stored.
  - o On getting this information, the client will reach to the DataNodes where the blocks are stored and start reading the data parallel from the DataNodes (Block A from DataNode 1 and Block B from DataNode 3).
  - o After getting all the required blocks of the file, the client will combine these blocks to form a file.
  - o Also note that while serving for the read request, the closest replica of the data is selected by the HDFS so as to reduce the read latency and the bandwidth consumption. Thus, the replica residing on the same rack as that of the reader node is selected, if at all it is possible.

## Advantages of using HDFS

- **Distributed storage:** HDFS has a distributed storage system, i.e., whenever we store a large file (say 10 TB) in HDFS, then it can be stored distributed on any of machines (say 10 machines) in the Hadoop cluster such that the size of the file is not limited to the physical boundaries of each individual machine. That means a single large file of 10 TB is distributed over the ten machines of 1 TB each.
- **Distributed and parallel computation:** Due to the file distributed over several machines, the computation can be done in parallel. Let's take an example to understand. Assume that a single machine can process a 1 TB file in 40 minutes. If the file is stored on HDFS system, i.e., stored on 10 different machines in the cluster, then the time taken to process 1 TB file would be 4 minutes as each of the node is working with a part of the 1 TB file in parallel. Thus, the entire file will be processed in 4 minutes instead of 40 minutes as the file is divided over 10 machines.
- **Horizontal scalability:** There are two types of scaling in any system: Horizontal scaling and vertical scaling. In vertical scaling (or scaling up), the capacity of hardware of the system is increased, i.e., we can add more RAM or CPU to the existing system to make it more powerful and robust. However, there are some challenges associated with this scalability. The system has some hardware capacity limit, and it is not possible to add RAM or CPU to any extent in the system. Also, in vertical scaling, to increase the hardware capacity of the system to make it more powerful or robust, the machine needs to be stopped first and needs to be restart once the scaling of the system is done. This down time is also a challenge.

In horizontal scaling or scaling out, more nodes are added to the existing cluster to increase the hardware capacity of the machine. Even more machine can be added to the cluster. There is no need to stop the system and hence the down time. HDFS adopts the scaling out (or horizontal scaling) system.

## Salient Features of HDFS

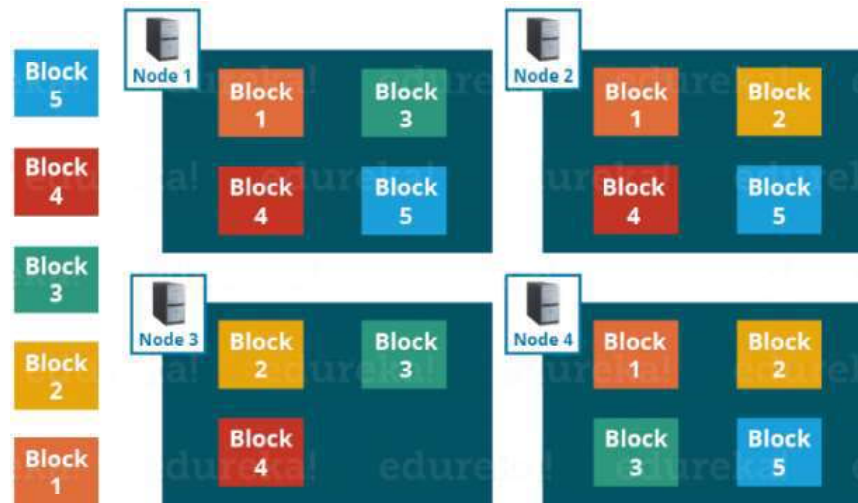
- **Fault tolerance and reliability:** HDFS provides highly resilient, since the data stored on HDFS is divided internally into different data blocks, which are stored in a distributed way across the Hadoop cluster and the data is also replicated, i.e., maintains the multiple copies of the data which makes HDFS fault tolerant and reliable. That means, even failure of any node will transfer the workload immediately on to another node. As mentioned above, the data is replicated by a factor of 3, a 1 GB of data stored on HDFS will occupy 3GB of space eventually.

## NOTES



## NOTES

- **Throughput:** HDFS provides high throughput even for a huge amount of data. Throughput is the work done in a given amount of time. This talks about the performance of the system, i.e., how quickly the data can be accessed from this file system. As observed above that the data is stored in a distributed way and hence through parallel processing the processing time can be reduced, increasing the throughput time and hence increasing the performance of the system.



*Fig. 14.10 Blocks Replication*

- **Cost:** Since, the commodity hardware are deployed in the HDFS system, it is economical in terms of the cost and scaling out (adding more nodes or machines) is more cost effective. Also, the computation is moved to the place where data resides instead otherwise in HDFS system, this makes it much cheaper, faster and improves overall throughput.
- **Volume and variety of data:** HDFS system has the capability of storing a gigantic data (Terabytes/ Petabytes) and different kind of data (structured, unstructured and semi-structured).

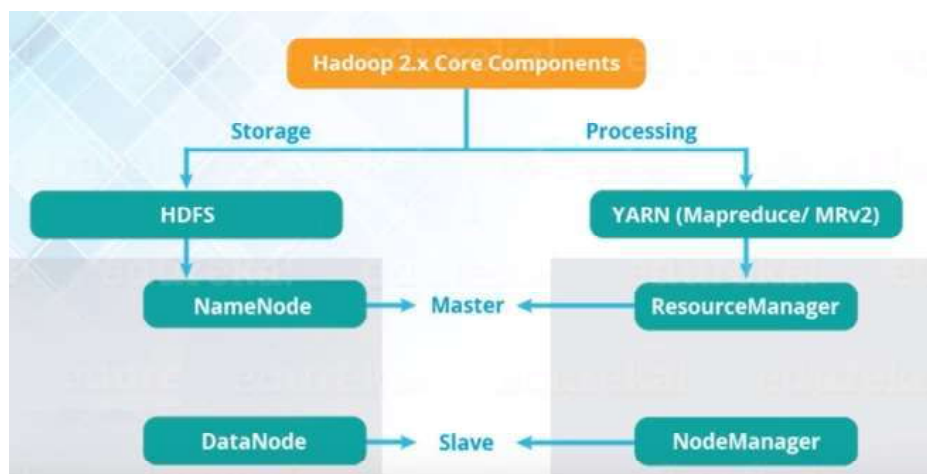
#### Check Your Progress

1. Define the term Apache Hadoop.
2. Name the two components of Hadoop.
3. What is Distributed File System (DFS)?
4. What is the use of the master node, NameNode?
5. List the advantages of using HDFS.

## 14.3 CORE HADOOP COMPONENTS

As discussed earlier, the core components of Hadoop are HDFS (For storing purpose) and YARN (for processing purpose).

### NOTES



*Fig. 14.11 Two Components of HDFS*

Following are the different components of Hadoop that forms the Hadoop Ecosystem as a whole. The first three form the basic Hadoop framework. The rest are other components that have emerged, to enrich Hadoop ecosystem to make it faster, to integrate it with other databases in a better way, and to build new capabilities.

- HDFS (Hadoop Distributed File System)
- YARN (Yet Another Resource Negotiator)
- MapReduce (Data Processing using programming)
- Spark (In-Memory Data Processing)
- PIG, HIVE (Data processing services using Query like SQL)
- HBase (NoSQL Database)
- Mahout, Spark MLlib (Machine Learning)
- Apache Drill (SQL on Hadoop)
- Zookeeper (Managing Cluster)
- Oozie (Job Scheduling)
- Flume, Sqoop (Data Integrating Services)
- Solr and Lucence (Searching and Indexing)
- Ambari (To manage and monitor Clusters)

NOTES

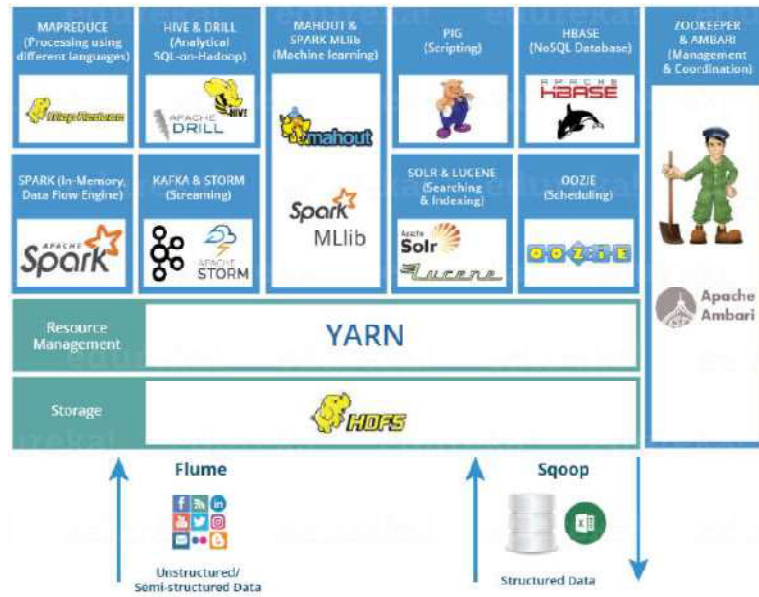


Fig. 14.12 Different components of Hadoop

We have already discussed HDFS in detail in the previous section. Let us go through the rest of the components briefly.

**YARN (Yet another Resource Negotiator)**

As discussed earlier, YARN acts as a processing unit of Hadoop that is capable of parallel processing of data stored across the HDFS. It acts as a resource management layer to allow multiple data processing engines to run and process data stored in HDFS. This means YARN acts as a brain of the Hadoop ecosystem. The responsibility of YARN is to process the activities by allocating resources and scheduling tasks.

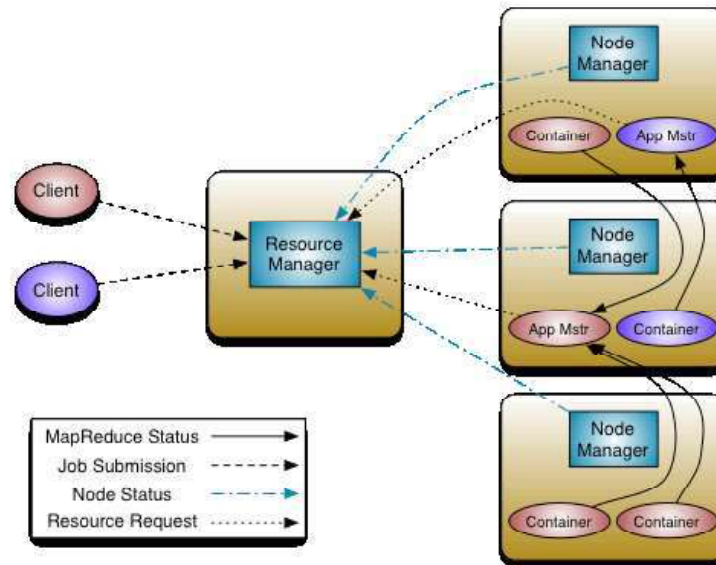


Fig. 14.13 Two components of YARN

There are two components of YARN: ResourceManager (RM) and NodeManager (NM). Both form the computation framework for YARN.

- **ResourceManager:** This is a master node, per cluster service, that receives the processing requests. These requests are then diverted in parts to the corresponding NodeManagers for actual processing. This is the actual authority to arbitrate resources (scheduling computing resources to applications) among all the applications in the system to help manage the distributed applications running on the YARN system. It is primarily a *pure scheduler*. It tracks about the live nodes and resources that are available on the cluster. It coordinates with the applications to decide which application submitted by the user should get which resource and when. It works in association with the per-node NodeManager and the per-application ApplicationMasters. The ResourceManager also has two components: Scheduler and ApplicationManager which are described later in this section.
- **NodeManager:** This is a per-machine agent framework that is installed on every DataNode that takes instructions from ResourceManager to create application's execution container, manage and monitor the usage of the resources available on the single node and reports to the ResourceManager.
- **ApplicationMasters:** This is a per-application framework specific library that is responsible for negotiating resources from ResourceManager. It works with the NodeManager to start the containers and to execute and monitor the tasks.

When an application is submitted by a user, the ApplicationMaster, an instance of a light weight process is started in order to coordinate the execution of all tasks such as monitoring task, restarting failed tasks, speculatively running slow tasks and calculating total values of application counters within the application. The ApplicationMaster and the tasks that belong to its application run in the container that is controlled by the NodeManager.

The ResourceManager has two main components:

- **Scheduler:** The scheduler performs scheduling algorithms to allocate the resources to different running applications based on the application resource requirements subject to familiar constraints of capacities, queues, application priority, data locality, etc. Scheduler does not monitor the status of the application and does not guarantee about restarting failed tasks due to either application or hardware failure. The scheduling function is performed according to the application resource requirement based on the abstract notion of a resource Container, which incorporates elements such as memory, CPU, disk, network, etc. It is a pure pluggable component responsible for partitioning the cluster resources among various queues, applications, etc.
- **ApplicationManager:** ApplicationManager is responsible for accepting job submission, negotiating the container (Data node environment where

## NOTES

process executes) for executing the application specific application master and monitoring the progress. It also provides service for restarting the ApplicationMaster container, in case of failure

## NOTES

### MapReduce (Model and it's working)

MapReduce is a core component (heart) for processing in Hadoop ecosystem. It is a software framework that provides the logic of processing for large datasets (multi-terabyte data-set) using distributed and parallel algorithms in the Hadoop environment by writing applications. It basically takes care of the tasks related to data processing and distributes the tasks across the nodes. The MapReduce program consists of two phases (functions): Map () and Reduce ().

**Map ():** It performs actions like filtering, grouping and sorting that converts a dataset into another dataset, where each element is broken down into tuples of key/ value pair (K, V), which acts as an input to the Reduce function. The input for this function is generally the files and directories that are stored in HDFS. The input is passed through map function line by line and converts the input into several small chunks of data.

**Reduce ():** Reduce () function takes the input from the Map () function and integrate the data tuples into smaller set of tuples to summarize and aggregate the result. The output so generated through this function is stored back to HDFS.

As the sequence in the name, the reduce task is always performed after the map job.

Consider an example to understand it further.

Assume a set of students with their respective departments as shown in the table below.

| Student   | Department | Count | (Key, Value),<br>Pair |
|-----------|------------|-------|-----------------------|
| Student 1 | D1         | 1     | (D1, 1)               |
| Student 2 | D1         | 1     | (D1, 1)               |
| Student 3 | D1         | 1     | (D1, 1)               |
| Student 4 | D2         | 1     | (D2, 1)               |
| Student 5 | D2         | 1     | (D2, 1)               |
| Student 6 | D3         | 1     | (D3, 1)               |
| Student 7 | D3         | 1     | (D3, 1)               |

The problem is to identify the number of students in each department. Through MapReduce framework, the Map () function will be executed to produce the key/ value pair, which is an input to the Reduce function. This function will then

aggregate each department and hence calculating the total number of students in each department.

Hadoop

| Department | Total Student |
|------------|---------------|
| D1         | 3             |
| D2         | 2             |
| D3         | 2             |

## NOTES

The Hadoop sends the Map and Reduce tasks to the appropriate servers in the cluster during MapReduce processing. The MapReduce framework is also responsible for managing all the data-passing details such as issue of task, task completion verification and copying data between the nodes around the cluster.

The objective of using MapReduce is to reduce the processing task of a huge datasets. The strength of this framework is its scalability, i.e., scale the work over a cluster which has thousands of nodes within it, once the MapReduce program is written.

### Apache PIG

Pig was initially developed by Yahoo! Apache PIG is an abstraction over MapReduce that is used to analyze larger sets of data representing them as data flows. All the data manipulation operations are performed using PIG in Hadoop. Pig has two components: Pig Latin which provides a high level language and has SQL like command structure and the Pig Runtime which provides an execution environment that accepts Pig Latin script as an input and converts this into MapReduce jobs for data analysis purpose. Pig is used to load the data, apply required filters and convert the data into required format.

Apache Pig has the following features:

- It provides various operators function such as join, filter, sort, etc.
- Pig Latin has a SQL like structure and it is easy to write Pig Script.
- Execution optimization is done automatically and hence programmers can focus on the semantics of the language.
- Users can develop their own functions to read, process and write data using existing operators.
- All kind of data, i.e., both structured and unstructured data can be analyzed using Pig. The result is stored in HDFS.

In general, Pig provides a platform for data ETL (Extract, Transform and Load), processing and analyzing huge data sets where the load command is used to load the data. Then various functions like grouping, filtering, joining, sorting, etc., which can either be dumped on screen or can be stored to HDFS.

## NOTES

**Apache HIVE**

Hive was initially created by Facebook which is a data warehousing component that performs reading, writing, and managing large data sets in a distributed environment using SQL-like interface. The query language of Hive is called as HQL (Hive Query Language) similar to SQL. There are two basic components: Hive Command Line and JDBC/ ODBC driver. The HQL command is executed using Hive command line interface and JDBC (Java Database Connectivity) and ODBC (Object Database Connectivity) is used to establish connection from data storage. Also, Hive is highly scalable which can serve the purposes of processing large dataset (Batch query processing) and real-time processing (Interactive query processing).

**Check Your Progress**

6. What is the responsibility of YARN?
7. What is the objective of using MapReduce?

**14.4 HADOOP ECOSYSTEM AND ARCHITECTURE**

Hadoop Ecosystem is a platform or a framework that provides solutions to the big data problem. It is neither a programming language nor a service. Hadoop ecosystem consists of different components that makes the Hadoop so powerful and provides several Hadoop job roles (or services) like ingesting, storing, analysing and maintaining. Figure 14.12 shows different components of Hadoop that collectively forms a Hadoop ecosystem.

HDFS (Hadoop Distributed File System), Yarn, Mapreduce, Apache Pig, Apache Hive, HBase have been discussed in above sections. Let's take a look at the remaining components in brief.

- **Mahout:** It is an open source framework known to perform machine learning algorithms and data mining library that provides an environment for creating machine learning applications which are scalable. It provides the data science tools to find meaningful patterns automatically from the big data sets that are stored in HDFS once. The various algorithms that it can perform are clustering, classification, filtering and frequent item set missing.
- **Apache Spark:** It is a framework written in Scala provided for real time data analytics in a distributed computing environment. To increase speed of data processing over Map-Reduce it executes in-memory computations. Since, it is faster than hadoop it requires high processing power than MapReduce. Spark includes high-level libraries including support for R, SQL, Python, Scala, Java etc. Many companies use Spark and Hadoop together for processing and analysing their Big Data stored in HDFS.

- **Apache Drill:** This is the first distributed SQL query engine to drill into any kind of data. It is an open source application that works with distributed environment to analyse large data sets including structured and semi structured data. It is capable of combining a variety of data stored just by using a single query. It follows ANSI SQL with powerful scalability factor in supporting millions of users and serve their query requests over large scale data. Drill works well with Hive by allowing developers to reuse their existing Hive deployment.
- **Apache Zookeeper:** It is a centralized service for maintaining configuration information, naming, providing distributed synchronization and providing group services. It acts as a coordinator between any Hadoop jobs that requires a combination of various services in Hadoop. Zookeeper is fast with workloads where reading data is more common than writing. The ideal read/write ratio is 10:1. It is ordered maintaining records of all transactions.
- **Oozie:** It is server based system that schedules and manages jobs in Hadoop ecosystem by combining multiple jobs sequentially into one logical unit of work. There are two basic types of Oozie jobs, Oozie workflow and Oozie coordinator.
- **Apache Flume:** Ingesting is an important part of Hadoop ecosystem. Flume is a service that helps in ingesting (collecting, aggregating, and moving) large amount of data (unstructured and semi-structured) from its origin to HDFS. In other words, it is a reliable and fault tolerance mechanism that allows the data to flow from the source to Hadoop environment. It help in ingesting online data streaming from different sources such as social media, email messages, log files, network traffic etc. into HDFS environment. The flume agent has 3 components: source (accepts the data from the incoming streamline to store the data into the channel), channel ( temporary storage) and sink (collects the data from the channel and writes it in the HDFS)
- **Apache Sqoop:** It is another ingesting service of Hadoop ecosystem that imports data from external sources into HDFS, Hbase or Hive. Unlike flume, sqoop can import as well as export structured data from RDBMS. Whenever, a sqoop command is submitted the task gets divided into sub tasks and are handled by individual map tasks internally. The whole data is imported by various Map Tasks which imports part of data to the Hadoop ecosystem. Export also works in a similar manner.
- **Ambari:** It is another Hadoop Ecosystem that provides highly intuitive, interacting, easy-to-use Hadoop management web UI to keep track of running applications and their status. It is deployed on the top of the Hadoop cluster to manage the platform for provisioning, managing, monitoring and securing Apache Hadoop Cluster. Provisioning provides step by step process for installing Hadoop services across a number of hosts and also handles configuration of Hadoop services over a cluster. It also enables central

## NOTES



management service for starting, stopping and re-configuring Hadoop services across the cluster. It also provides dashboard to notify users whenever attention is needed. For example, it alerts users on low disk space of a node or when node goes down.

## NOTES

### 14.4.1 Physical Architecture of Hadoop

Apache Hadoop offers a scalable, flexible, and reliable distributed computing big data framework to store and process unstructured data by leveraging commodity hardware. Hadoop follows master/slave architecture to store, process and analyse large set of data using Hadoop MapReduce paradigm. In Hadoop environment, the data files are stored on multiple nodes called as data nodes. The Hadoop architecture consists of 3 major components:

- Hadoop Distributed File System (HDFS) – Patterned after the UNIX file system
- Hadoop MapReduce
- Yet Another Resource Negotiator (YARN)

These components have been discussed the above sections.

---

## 14.5 HADOOP LIMITATIONS

---

Hadoop has emerged as a solution to handle Big Data. It is an open source with distributed storage and processing. It has many advantages, but everything comes with limitations. Here, few limitations have been discussed:

- **Unfit to handle small files:** HDFS lacks the ability to support the random reading of small files as the objective behind the Hadoop concept was to divide the large files into many small files. Infact, the small files may have size quite less than the block size (default 128 MB) in Hadoop. Storing large number of small files may overload the NameNode and HDFS might not be able to handle those lot of small files. Also, retrieving small files would be inefficient in Hadoop as it may cause many disks to seek and hopping between different DataNodes and incurs a lot of time.
- **(Slow) Processing Speed:** Hadoop process a large set of data using parallel and distributed algorithm by breaking the process into phases: Map and Reduce. This requires a lot of time to perform tasks thereby increasing latency and hence reducing processing speed.
- **Lack of Real-time processing:** Hadoop framework supports only batch processing and is unable to process the real-time data, thus effecting the overall performance.
- **No Iterative Processing:** Core Hadoop is not efficient for iterative processing. Iterative processing requires a cyclic data flow where output of previous stage acts as an input to next stage. Hadoop works on batch

processing which works on write-once-read-many and has no capability for iterative processing.

- **Security problem:** Hadoop does not implement encryption-decryption at the storage as well as at network level which is a major concern. It adopts Kerberos authentication which is difficult to manage.
- **Increased latency:** Since Hadoop (MapReduce supports different structure and format of data and the amount of data is also huge, it is slower. In the MapReduce process, data is converted into another set of data in Map phase where an individual element is broken down into a key-value pair. The output from Map phase is taken into Reduce phase to process further. Hence a lot of time is required to perform these task there by increasing latency.

## NOTES

### Check Your Progress

8. What is Hadoop ecosystem?
9. What is Apache zookeeper?

## 14.6 ANSWERS TO CHECK YOUR PROGRESS QUESTIONS

1. Apache Hadoop is an open-source software framework used for distributed storage and processing of datasets of big data using the MapReduce programming model. It consists of computer clusters built from commodity hardware.
2. There are broadly two components of Hadoop:
  - (i) **Hadoop Distributed File System (HDFS):** This is actually storage for the system and can store data of various formats across a cluster. We will discuss this component in detail in the next section.
  - (ii) **YARN:** This acts as a processing unit of Hadoop that is capable of parallel processing of data stored across the HDFS.
3. Distributed File System (DFS) is a way of managing data (files and folders) across multiple servers (computers). It is a system that allows data to be stored over multiple machines (called nodes) in a cluster. This also allows multiple users to access the data.
4. The master node, NameNode, is responsible for storing the metadata about the HDFS files, i.e., the information about HDFS file such as file name, file permissions, file size, block, where the file is stored and the information about the replica of data nodes are kept.

## NOTES

5. The following are advantages of using HDFS:
  - Distributed storage
  - Distributed and parallel computation
  - Horizontal scalability
6. The responsibility of YARN is to process the activities by allocating resources and scheduling tasks.
7. The objective of using MapReduce is to reduce the processing task of a huge datasets. The strength of this framework is its scalability, i.e., scale the work over a cluster which has thousands of nodes within it, once the MapReduce program is written.
8. Hadoop Ecosystem is a platform or a framework that provides solutions to the big data problem.
9. Apache Zookeeper is a centralized service for maintaining configuration information, naming, providing distributed synchronization and providing group services.

---

## 14.7 SUMMARY

---

- Apache Hadoop is an open-source software framework used for distributed storage and processing of datasets of big data using the MapReduce programming model. It consists of computer clusters built from commodity hardware.
- Apache Hadoop is an important framework to work with big data, which is scalable and can upgrade itself from working on a single node to thousands of nodes in a seamless manner, without facing any issues.
- DFS is a way of managing data (files and folders) across multiple servers (computers). It is a system that allows data to be stored over multiple machines (called nodes) in a cluster.
- The concept of blocks is similar to any other file system on your laptops or PCs, where data is stored as a collection of block.
- Blocks are the continuous location where the data is stored on the hard drive.
- HDFS follows write once and read many approach for the files stored in HDFS platform. Also, it does not allow editing the files once written.
- HDFS has a distributed storage system, i.e., whenever we store a large file (say 10 TB) in HDFS, then it can be stored distributed on any of machines (say 10 machines) in the Hadoop cluster such that the size of the file is not limited to the physical boundaries of each individual machine.

- HDFS provides highly resilient, since the data stored on HDFS is divided internally into different data blocks, which are stored in a distributed way across the Hadoop cluster and the data is also replicated, i.e., maintains the multiple copies of the data which makes HDFS fault tolerant and reliable.
- YARN acts as a processing unit of Hadoop that is capable of parallel processing of data stored across the HDFS.
- MapReduce is a core component (heart) for processing in Hadoop ecosystem. It is a software framework that provides the logic of processing for large datasets (multi-terabyte data-set) using distributed and parallel algorithms in the Hadoop environment by writing applications.
- Pig was initially developed by Yahoo! Apache PIG is an abstraction over MapReduce that is used to analyze larger sets of data representing them as data flows. All the data manipulation operations are performed using PIG in Hadoop.
- Hive was initially created by Facebook which is a data warehousing component that performs reading, writing, and managing large data sets in a distributed environment using SQL-like interface.
- Hadoop ecosystem consists of different components that makes the Hadoop so powerful and provides several Hadoop job roles (or services) like ingesting, storing, analysing and maintaining.
- Apache Hadoop offers a scalable, flexible, and reliable distributed computing big data framework to store and process unstructured data by leveraging commodity hardware. Hadoop follows master/slave architecture to store, process and analyse large set of data using Hadoop MapReduce paradigm.

## NOTES

---

### 14.8 KEY WORDS

---

- **YARN (Yet Another Resource Negotiator):** YARN acts as a processing unit of Hadoop that is capable of parallel processing of data stored across the HDFS. It acts as a resource management layer to allow multiple data processing engines to run and process data stored in HDFS.
- **ResourceManager:** This is a master node, per cluster service, that receives the processing requests. These requests are then diverted in parts to the corresponding NodeManagers for actual processing.
- **NodeManager:** This is a per-machine agent framework that is installed on every DataNode that takes instructions from ResourceManager to create application's execution container, manage and monitor the usage of the resources available on the single node and reports to the ResourceManager.

---

## 14.9 SELF ASSESSMENT QUESTIONS AND EXERCISES

---

### NOTES

#### Short Answer Questions

1. Discuss the evolution of Hadoop.
2. What are the two broad components of Hadoop?
3. What are Blocks in HDFS?
4. What are the advantages of using Rack awareness algorithm?
5. What are the protocol that will be followed in writing the file into HDFS?
6. What are the two components of YARN?

#### Long Answer Questions

1. Explain the architecture of HDFS.
2. What are the advantages and salient features of HDFS?
3. Explain the various core Hadoop components.
4. Write a note on Hadoop ecosystem.
5. What are the limitations of Hadoop?

---

## 14.10 FURTHER READINGS

---

Han, Jiawei, Micheline Kamber and Jian Pei. 2011. *Data Mining: Concepts and Techniques*, 2nd edition. Amsterdam: Elsevier.

Pujari, Arun K. 2010. *Data Mining Techniques*, 2nd edition. United States: Universities Press.

Anahory, Sam and Dennis Murray. 1997. *Data Warehousing in the Real World: A Practical Guide for Building Decision Support Systems*, 1st edition. Boston: Addison Wesley.

Witten, I. H. and Eibe Frank. 2005. *Data Mining: Practical Machine Learning Tools and Techniques*, 2nd edition. Amsterdam: Elsevier.

Soman, K. P., Shyam Diwakar and V. Ajay. 2006. *Insight Into Data Mining: Theory and Practice*. New Delhi: PHI.

# Master of Computer Applications 31535 DATA MINING AND WAREHOUSING

III - Semester



## ALAGAPPA UNIVERSITY

[Accredited with A+' Grade by NAAC (CGPA:3.64) in the Third Cycle  
and Graded as Category-I University by MHRD-UGC]

**KARAIKUDI – 630 003**

**DIRECTORATE OF DISTANCE EDUCATION**

