SUBJECT CODE : CS8075

Strictly as per Revised Syllabus of

ANNA UNIVERSITY

Choice Based Credit System (CBCS)
Semester - VI (CSE) Professional Elective - I

DATA WAREHOUSING & DATA MINING

Pranjali Deshpande

M.E. (Comp. Engg.), Ph.D (Pursuing), DoT, SPPU.
Assistant Professor,Department of Computer Engineering,
MKSSS's Cummins College of Engineering for Women,
Pune

Soudamini Patil

M.E. (Computer Science and Engineering)
Formerly Assistant Professor,
Department of Computer Engineering,
MKSSS's Cummins College of Engineering for Women, Pune
Formerly Principal, Education and Research, Infosys Ltd.
Pune

TECHNICAL®
PUBLICATIONS
SINCE 1993
An Up-Thrust for Knowledge

DATA WAREHOUSING & DATA MINING

Subject Code : CS8075

Semester - VI (Computer Science & Engineering) Professional Elective - I

First Edition : January 2020

Published by :
TECHNICAL® PUBLICATIONS
SINCE 1993 An Up-Thrust for Knowledge
Amit Residency, Office No.1, 412, Shaniwar Peth, Pune - 411030, M.S. INDIA
Ph.: +91-020-24495496/97, Telefax : +91-020-24495497
Email : sales@technicalpublications.org Website : www.technicalpublications.org

Printer :
Yogiraj Printers & Binders
Sr.No. 10/1A,
Ghule Industrial Estate, Nanded Village Road,
Tal. - Haveli, Dist. - Pune - 411041.

Price : ₹ 295/-
ISBN 978-93-89420-92-0
9 789389 420920

AU 17

PREFACE

The importance of **Data Warehousing and Data Mining** is well known in various engineering fields. Overwhelming response to our books on various subjects inspired us to write this book. The book is structured to cover the key aspects of the subject **Data Warehousing and Data Mining**.

The book uses plain, lucid language to explain fundamentals of this subject. The book provides logical method of explaining various complicated concepts and stepwise methods to explain the important topics. Each chapter is well supported with necessary illustrations, practical examples and solved problems. All chapters in this book are arranged in a proper sequence that permits each topic to build upon earlier studies. All care has been taken to make students comfortable in understanding the basic concepts of this subject.

Representative questions have been added at the end of each section to help the students in picking important points from that section.

The book not only covers the entire scope of the subject but explains the philosophy of the subject. This makes the understanding of this subject more clear and makes it more interesting. The book will be very useful not only to the students but also to the subject teachers. The students have to omit nothing and possibly have to cover nothing more.

We wish to express our profound thanks to all those who helped in making this book a reality. Much needed moral support and encouragement is provided on numerous occasions by our whole family. We wish to thank the **Publisher** and the entire team of **Technical Publications** who have taken immense pain to get this book in time with quality printing.

Any suggestion for the improvement of the book will be acknowledged and well appreciated.

Authors

Soudamini Patil

Pranjali Deshpande

SYLLABUS

Data Warehousing and Data Mining [CS8075]

UNIT I Data Warehousing, Business Analysis and On - Line Analytical Processing (OLAP)

Basic Concepts - Data Warehousing Components - Building a Data Warehouse - Database Architectures for Parallel Processing - Parallel DBMS Vendors - Multidimensional Data Model - Data Warehouse Schemas for Decision Support, Concept Hierarchies - Characteristics of OLAP Systems - Typical OLAP Operations, OLAP and OLTP. **(Chapter - 1)**

UNIT II Data Mining - Introduction

Introduction to Data Mining Systems - Knowledge Discovery Process - Data Mining Techniques - Issues - applications - Data Objects and attribute types, Statistical description of data, Data Preprocessing - Cleaning, Integration, Reduction, Transformation and Discretization, Data Visualization, Data similarity and dissimilarity measures. **(Chapter - 2)**

UNIT III Data Mining - Frequent Pattern Analysis

Mining Frequent Patterns, Associations and Correlations – Mining Methods - Pattern Evaluation Method – Pattern Mining in Multilevel, Multi Dimensional Space – Constraint Based Frequent Pattern Mining, Classification using Frequent Patterns. **(Chapter - 3)**

UNIT IV Classification and Clustering

Decision Tree Induction, Bayesian Classification - Rule Based Classification - Classification by Back Propagation - Support Vector Machines - Lazy Learners - Model Evaluation and Selection - Techniques to improve Classification Accuracy.

Clustering Techniques - Cluster analysis - Partitioning Methods - Hierarchical Methods - Density Based Methods - Grid Based Methods - Evaluation of clustering - Clustering high dimensional data - Clustering with constraints, Outlier analysis - Outlier detection methods. **(Chapter - 4)**

UNIT V WEKA Tool

Datasets - Introduction, Iris plants database, Breast cancer database, Auto imports database - Introduction to WEKA, The Explorer - Getting started, Exploring the explorer, Learning algorithms, Clustering algorithms, Association - rule learners. **(Chapter - 5)**

Table of Contents

Unit - I

Chapter - 1 Data Warehousing, Business Analysis and On-Line Analytical Processing (OLAP) (1 - 1) to (1 - 40)

Unit - II

Unit - III

Chapter - 3 Data Mining - Frequent Pattern Analysis (3 - 1) to (3 - 30)

Unit - IV

Chapter - 4 Classification and Clustering

(4 - 1) to (4 - 70)

Unit - V

Unit - I

1 Data Warehousing, Business Analysis and On-Line Analytical Processing (OLAP)

Syllabus

Basic Concepts - Data Warehousing Components – Building a Data Warehouse – Database Architectures for Parallel Processing – Parallel DBMS Vendors - Multidimensional Data Model – Data Warehouse Schemas for Decision Support, Concept Hierarchies -Characteristics of OLAP Systems – Typical OLAP Operations, OLAP and OLTP.

Contents

1.1 Data Warehousing Components

1.1.1 Overall Architecture

- To understand the architecture of data warehouse let's first understand the difference between operational system and informational system.
- Operational system (OLTP) maintain records of daily business transactions and is typically optimized for quick inserts and updates and look ups of single row or small collection of rows.
- An information system is an organised system that affects the interplay between people, processes and technology in an organisation.
- Relational database management system server functions as the central repository for informational data.
- The overall architecture of data warehouse is based on this Relational database management system server.
- The architecture separates operational data processing from data warehouse processing.
- The key components shown in Fig. 1.1.1 are responsible for management and accessibility of operational system which source data into warehouse and end user query and analysis tools.
- Let's discuss the various components of data warehouse architecture shown in Fig. 1.1.1.

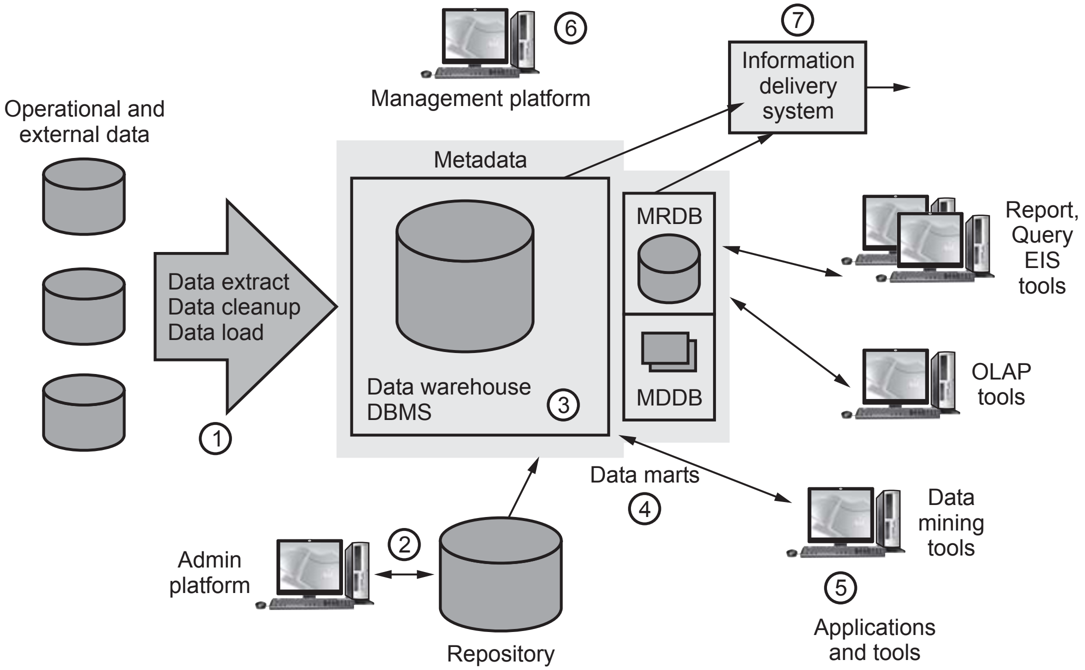


Fig. 1.1.1 Data warehouse architecture

1.1.2 Data Warehouse Database

- In the Fig. 1.1.1 number ② indicates the data warehouse database.
- Data warehouse database is usually implemented using RDBMS technology.
- The constraint faced by traditional RDBMS system is, the implementation is optimized for transactional database processing.
- To handle various attributes of data warehouse, like very large database size, adhoc query processing and to provide flexible user view creation like aggregates, multiple joins, various approaches can be adapted.
- Some of the approaches are :
  - Parallel relational design using parallel computing platforms for symmetric multiprocessors (SMPs) massively parallel processors (MPPs).
  - Use of new index structures for by passing relational table scans.
  - Use of multidimensional databases (MDDBs) to overcome the emulation posed by nature of relational data model.

1.1.3 Sourcing, Acquisition, Cleanup and Transformation Tools

- In the Fig. 1.1.1 number ① indicates the data sourcing, cleanup, transformation and transformation tools.
- These tools are responsible for performing all the conversions, summarizations, key changes, structural changes and condensational for transforming disparate data to information which can be used by decision support tool.
- The main functions are :
  - Remove unwanted data from operational database.
  - Calculate summary.
  - Missing data handling, etc.
- The challenges may be faced due to database heterogeneity and data heterogeneity.

1.1.4 Metadata

- To understand the concept of Metadata let's consider the example to yellow pages.
- Think of metadata as the Yellow Pages® of your town. If you need information about the stores in your town, where they are, what their names are and what products they specialize in ? You go to the Yellow Pages.
- The Yellow Pages is a directory with data about the institutions in you town. Almost in the same manner, the metadata component serves as a directory of the contents of your data warehouse.

- The following diagram shows the roles of metadata.

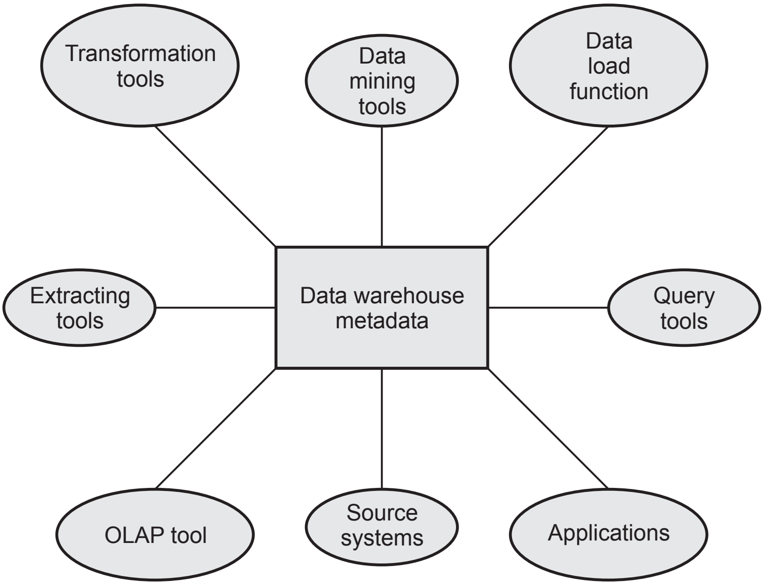


Fig. 1.1.2 Roles of metadata

- Metadata can be classified into

  Technical metadata

  Business metadata

- Technical Metadata

Technical metadata is meant for the IT staff responsible for the development and administration of the data warehouse.

The Technical metadata documents include :

- ○ Data source information
- ○ Transformation descriptions
- ○ The rules for performing data cleanup and data enhancement, etc.

- Business Metadata

  Business metadata connects your business users to your data warehouse.

  Business users need to know what is available in the data warehouse from a perspective different from that of IT professionals.

  Business metadata is like a roadmap or an easy-to-use information directory showing the contents and how to get there.

  It is like a tour guide for executives and a route map for managers and business analysis.

The Business Metadata documents include :

- Queries, reports, images, videos, etc.
- Internet home pages.
- Supporting information for data warehousing components.
- Operational information like data history, ownership, usage data, etc.

- Data warehousing and business intelligence metadata is best managed through a combination of people, process and tools.
- Metadata can be managed through individual tools :
  - Metadata manager/repository
  - Metadata extract tools
  - Data modelling
  - ETL
  - BI Reporting
- Metadata Manager/Repository
  - Metadata can be managed through a shared repository that combines information from multiple sources.

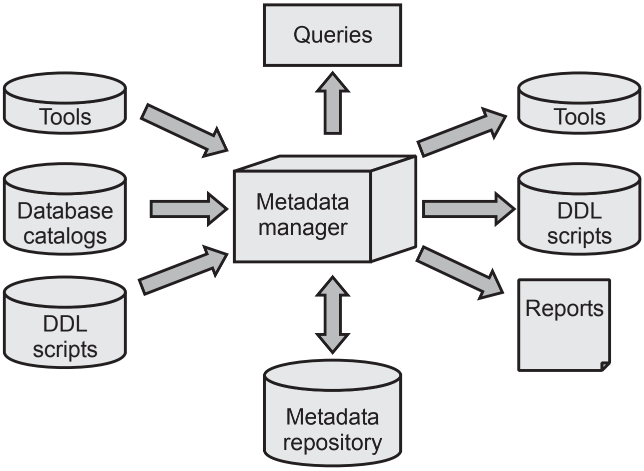


Fig. 1.1.3 Metadata management

- The metadata manager can be purchased as a software package or built as "home grown" system. Many organization start with a spreadsheet containing data definitions and then grow to a more sophisticated approach.

1.1.5 Access Tools

- Data warehouse is used business users to perform strategic decisions.
- For this uses interaction data warehouse is needed which is facilitated by different front end tools.

- Apart from adhoc requests, regular reports and custom applications one more type of report is focused on known as alerts.
- Alerts are used to notify the occurrence of contain event to the user.
- For example : The data warehouse designed to access risk of currency trading, alert can be generated when currency rate drops.
- For handling all such applications various tools are used, which are divided in five groups -
  1. Data energy and reporting tools
  2. Application development tools
  3. Executive information tools
  4. On time analytical tools
  5. Data mining tools

Query and reporting tools :

- These are further divided into two groups :
  1. Reporting tools : These are divided in
     a. Production reporting tools : To generate various operational reports,
     b. Report writers : Inexpensive tools for end users.

OLAP (on-line analytical processing) tools :

- These tools are based on concept of multi-dimensional databases.
- They facilitate users to analyse the data using elaborate, multidimensional and complex views.

Data Mining :

- Data mining is a process of discovering meaningful new correlations, patterns and trends by digging into large amounts of data stored in warehouses, using artificial intelligence and statistical and mathematical techniques.
- Data mining is majorly used by organizations to :
  - **Discover knowledge :**
    - Hidden relationships, patterns are determined in this.
    - Segmentation, classification, association and preferencing can be performed using data mining.
  - **Visualize data :**
    - It deals with proper and effective ways to display humongous data present in corporate databases.

- **Correct data :**
  - Various techniques are used to identify inconsistency and incompleteness in the data.

Data visualization :

- It is not separate class of tools, instead it is a method to represent outputs of all the different tools.
- For data visualization complex techniques are used to display complex relationships. For example : Use of different shapes, colours, sound and virtual reality which help users to see and feel the problems.

1.1.6 Data Warehouse Administration and Management

- As data warehouse is almost 4 times large as compared to other operational databases, managing it is a crucial task.
- Following factors should be considered in management of warehouse
  - Security and priority management
  - Monitoring updates from multiple sources
  - Data quality checking
  - Metadata management and updation
  - Audit of usage and status
  - Replication and distribution of data
  - Backup and recovery
  - Storage management.

1.1.7 Information Delivery System

- It involves subscription of data warehouse information and its delivery to different destinations.
- Connection with other data warehouses is needed in this process.
- Delivery turning can be scheduled, either it will be done on some fixed time of day or after complection of certain event.

1.2 Building a Data Warehouse

- For survival and success in business following factors are important :
  - Quick decisions using all available data.
  - The fact that users are not computer experts need to be considered.
  - Rapid growth of data.

  - Competition due to adaption of business intelligence techniques.

- Data warehouse must handle the incompatibility of informational and operational system.
- At the same time is has to handle ever changing IT infrastructure.

1.2.1 Business Considerations

- According to requirement of business, the organization may choose to build the separate data warehouses for different departments.
- Individual warehouse is called as data mart.
- For development of warehouse two approaches can be taken :

  1 **Top down approach** : In this enterprise data model is developed first considering various business requirements. Later warehouse is built using data mart.

  2. **Bottom-up approach** : In this individual data marts are built first, which are then integrated ratio enterprise data warehouse.

Organizational Issues :

- Generally organizations can built operational systems efficiently but to built a data warehouse there are different requirements.
- The data from operational systems as well as from outside need to be considered.
- Data warehouse building is not just a technical issue but case should be taken to establish information requirements.

1.2.2 Design Considerations

- For designing data warehouse, designer must consider all data warehouse components, all possible data sources and all known usage requirements.
- The data is consolidated from multiple heterogeneous sources into query database.
- Heterogeneity of data resources, use of historical data and tendency of growth of database are the main consideration factors.

Data content :

- Data warehouse need detailed data but the data need to be cleaned and transformed to fit in the warehouse model.
- Content and structure of data warehouse can be seen in its data model.
- Data model in a template that describes how information will be organized in data warehouse framework.

Metadata :

Refer section 1.1.4 for detailed description of metadata.

Data distribution :

- As data grows rapidly, it becomes necessary to distribute it to multiple servers.
- In this process of data distribution by subject area, location or time should be considered.

Tools :

- To implement data warehouse various tools are available.
- These tools are used for data movement, end user query, reporting, data analysis, etc.
- Each tool maintains our metadata stored in proprietary metadata repository.
- The care must be taken to ensure that the selected tools are compatible with data warehouse environment.

Performance considerations :

- The data warehouse should support rapid query processing.

Nine decisions in the design of a data warehouse :

- The management expects precise and quick response on processing of enterprise data.
- It's responsibility of designer is to provide answers to all the questions to all the questions by management but still have a simple design.
- To facilitate this the design methodology by Ralph kimball and can be used, which is known as "nine step method".
- The nine steps are listed in the Table 1.2.1 below.

1.	Choosing the subject matter
2.	Deciding what a fact table represents
3.	Identifying and conforming the dimensions
4.	Choosing the facts
5.	Storing precalculations in the fact table
6.	Rounding out the dimension tables
7.	Choosing the duration of the database
8.	The need to track slowly changing dimensions.
9.	Deciding the query priorities and the query modes.

- In both top-down view or in bottom up view, the data ware house designer should follow following steps.

1. Choosing the subject matter of a particular data mart :
   - The designed data mart should answer important business questions and also it should be accessible for data extraction.
   - As per the kimball the process can be started by building a data mart consisting of customer invoices and monthly statements.
2. Deciding exactly what a fact table represents :
   - A fact table is the central table in design that has multipart key.
   - Multipart key components acts as foreign key to an individual dimension table.
   - After deciding fact table representation, dimensions of data marts fact table is decided.
3. Identifying and conforming the dimensions :
   - Dimensions are very important part of data mart.
   - They make data mart understandable and easy to use.
   - While deciding dimensions, long range data warehouse should be considered.
   - If a dimension occur in two data marts it should be same or mathematical subset of each other.
   - Such type of dimension is called as conformed dimension.

1.2.3 Technical Considerations

- Various technical issues need to be considered while building a warehouse.
- Some of them are :
  - The hardware platform
  - Supporting DBMS
  - Communication infrastructure
  - Hardware and software support for metadata repository.
  - The system management framework for earlier environment.

Hardware platforms :

- The following hardware platform considerations should be taken care of while designing data warehouse.
  - Capacity of handling range volume of data for decision support applications.
  - Data warehouse server should be specialized to handle tasks related to date warehouse mainframe can be used as data warehouse server.

- The balance need to be maintained between computing components like number of processors and I/O bandwidth.
- For this disk I/O rates and processor capability need to be analyzed.
- Non uniform distribution of data or data skew will have effect on scalability, which may overpower best data layout for parallel execution.

Data warehouse and DBMS specialization :

- Catering to large size of databases performance, throughput and scalability are the important requirements for data warehouse DBMS.
- The relational DBMS systems like DB2, Oracle, Informix or Sybase are used to fulfil the requirement of data warehouse.
- Some move specialized databases include Red brick warehouse from Red brick systems.

Communications Infrastructure :

- To access the corporate data from the desktop require cost and efforts.
- Typically large bandwidth is required to interact with data warehouse.

1.2.4 Implementation Considerations

- Implementation of data warehouse needs integration of many products within a warehouse.
- To build a data warehouse following logical steps need to be taken :
  - Business requirement collection and analysis.
  - Data model and physical design for warehouse.
  - Define data sources.
  - Database technology and platform selection for warehouse.
  - Data extraction, transformation, cleaning and loading into database.
  - Access and reporting tool selection for the database.
  - Selection of database correctively software.
  - Selection and data analysis and presentation software.
  - Data warehouse updation.

Data extraction, clean up, transformation and migration.

- Data extraction in a critical factor for making successful data warehouse architecture.
- Following selection criteria related to transformation, consolidation, integration, repairing of data should be considered :

- Identification of data in the data source environment is important.
- Flat files, indexed files and legacy DBMS should be supported as most of the data is still stored in these formats.
- Merging data from different data stores is important.
- Datatype and character set translation is needed.
- Summarization, aggregation, etc. capabilities are needed.
- Evaluation of vendor stability is needed.

Vendor solutions :

- The vendors described below provide more focused solutions to fulfil the requirements for data warehouse implementation.

1. **Prism solutions :**
   - ○ Prism warehouse manager extracts data from multiple source environments like DB2, IDMS, IMS, VSAM, etc.
   - ○ Target databases are Oracle, Sybase and Informix.

2. **Carleton's PASSPORT :**

- It consists of two components :
  1. First component collects the file record table layouts and converts into Passport Data Language (PDL)
  2. Second component is used to create the metadata directory which is used to build COBOL programs to create the extracts.

3. **Information builders Inc. :**

- These products provide SQL access and uniform relational view of relational and non relational data in 60 different databases and 35 different platforms.

4. **SAS institute Inc. :**

- SAS system tools are used for all data warehousing functions.

Metadata :

Refer section 1.1.4 for the details.

1.2.5 Integrated Solutions

- Vendors provide suite of services and products for establishment of data warehouse :
- Some of the vendors are as follows :

1. **Digital equipment corp :**

- They use :

  Prism warehouse manager → for data modeling extraction and cleansing capabilities and ACCESS WORKS → as database servers to provide users the ability to build and use information warehouses.

2. **Hewlett-Packard :**

- They give single source support for full HP open warehouse solution.
- HP open warehouse consists of data management architecture,
  the HP-UX operating system, HP 9000 computers, warehouse management tools, an Allbase/SQL relational database and HP information access query tool

3. **IBM :**

- IBM information warehouse includes
  - Data management tools
  - OS/2, AIX and MVS OS
  - Hardware platforms
  - Relational database
- Other components are :
  - Data Guide/2 - catalog of shared data and information objects
  - Data propagation
  - Data refresher
  - Data hub
  - Application system and personal application system
  - Query management facility → IBM flow mark

Sequent :

- Sequent computer systems Inc. has a decision point program for delivering of data warehouses.
- It has sequent symmetric multiprocessing (SMP) architectural with client/server products and services such as UNIX-based sequent symmetry 2000 series, Red Brick warehouse for Red Brick systems and clear access query tool from clear access corp.

1.2.6 Benefits of Data Warehousing

- There are two major benefits of data warehouse architecture

1. The availability of business intelligence data is increased.
2. Business decisions can be made more effectively considering the timeline constraints.

- The benefits can be categorized as
  1. Tangible benefits
  2. Intangible benefits

1. Tangible benefits :

- One of the major benefit is out of stock conditions can be improved
- Some additional benefits are :
  - ○ It provides big picture of purchasing and inventory patterns which facilitates cost saving.
  - ○ Business intelligence can be enhanced by proper market analysis.
  - ○ Cost effective decisions can be made by spectrum of ad-hoc query processing and operational databases.
  - ○ Target market selection is improved resulting in decrease in cost of product introduction. Also improvement product inventory turnover is observed.

2. Intangible benefits :

- Following are the intangible benefits of data warehouse :
  - ○ As all the required data can be kept at a single location productivity is improved.
  - ○ Overlap in decision support applications is reduced by reduction in redundant processing.
  - ○ Customer relations are enhanced as individual requirements and trends can be better understood.
  - ○ Useful insights are provided into work processing, through which the processes can be re-engineered by inclusion of innovative ideas.

1.3 Database Architectures for Parallel Processing

- Parallel architectures include parallel hardware on which software parallelism can be exploited along with parallel operating system.
- The use of suitable parallel database software architecture is required to take advantage of shared memory and distributed memory parallel environments.
- Use of parallel software database architecture decides scalability of the solution.
- Three main DBMS software architecture are
  1. Shared-everything architecture
  2. Shared disk architecture
  3. Shared nothing architecture.

1.3.1 Shared Everything Architecture (Shared Memory Architecture)

- The parallel platform in which all the processors access the common data space is called as shared memory platform.
- Processors interact with each other by accessing and modifying the data elements stored in the shared address space.
- It is a traditional approach to implement RDBMS on SMP hardware.
- As shown in Fig. 1.3.1 single system enrage is provided to the used.

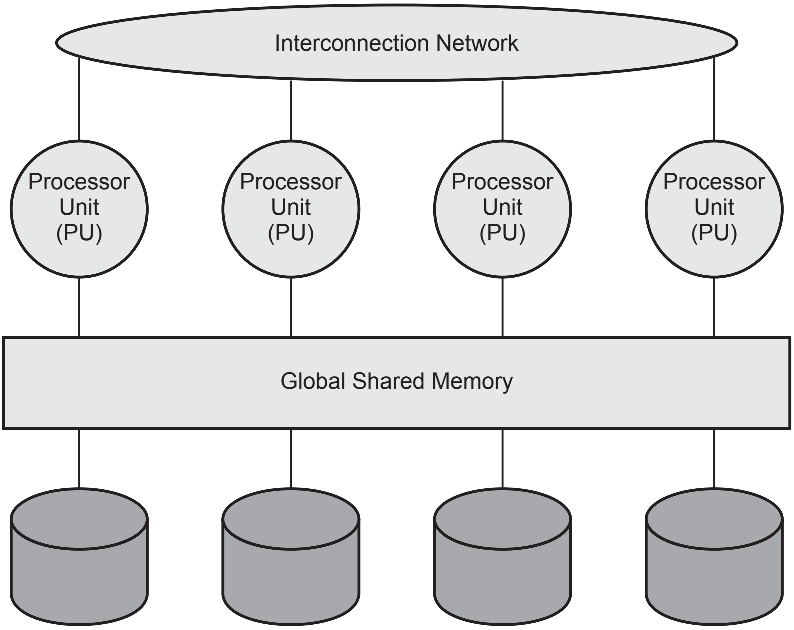


Fig. 1.3.1 Shared memory architecture

- All the processors, memory and entire database is utilized by single RDBMS sever.
- The SQL statements executed by multiple database components are communicated to each other by exchanging the messages.
- The data is partitioned in local disks which can be accessed by all the processor.
- The scalability is dependent on the design process

1. Process based implementation :
   - ○ Exploited in oracle 7.x running on UNIX platform
2. Thread based implementation :
   - ○ RDBMS implement its own threads
     e.g. SYBASE SQL server

  - OR it used OS threads
    eg. Microsoft SQL server running on NT.
- The threads based architecture provides better scalability due to better utilization and fast context switching.
- If the threads are too tightly coupled it results in limited RDBMS portability.
- The disadvantages of this architecture are
  - Scalability is limited.
  - Throughput is limited as it is based on processor and system bus speed.

1.3.2 Shared Disk Architecture

- As shown in Fig. 1.3.2 this architecture uses concept of distributed memory system.

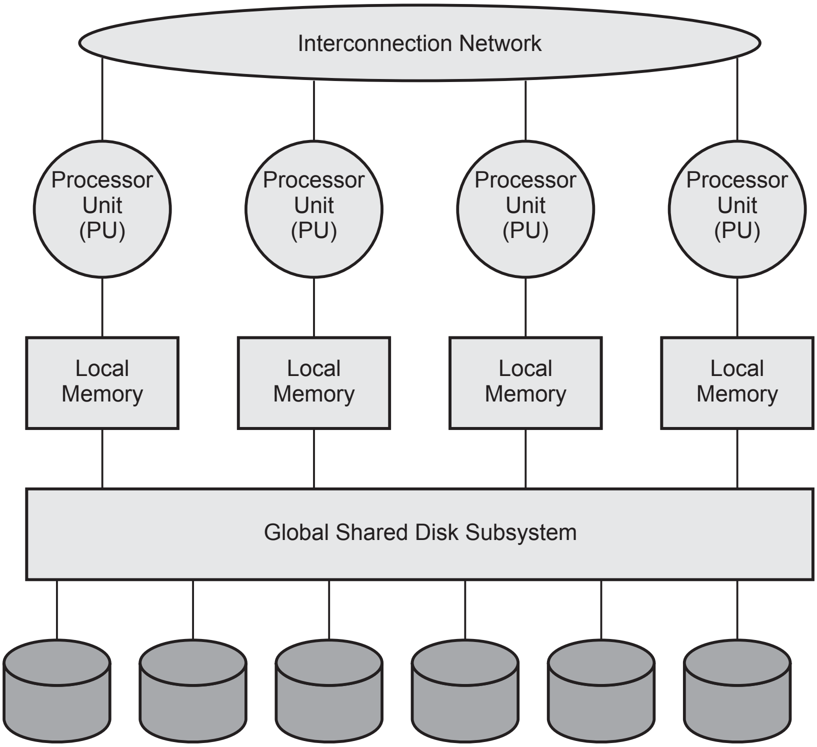


Fig. 1.3.2 Distributed - memory shared - disk architecture

- RDBMS servers shares the entire database running on the nodes.
- The records are read, written, updated and deleted by the each RDBMS server.
- Distributed lock manager (DLM) concept is used for coordination.
- Single system image is provided by hiding the DLM components found in hardware, OS in software layers etc.

- This scenario poses the challenge of synchronization as if the servers are reading and updating the same data, resources are wasted in synchronization.
- One more drawback is : if utilization of RDBMS servers are more, DLM can witness the bottleneck.
- Some of the advantages of this architecture are :
  - System availability is increased as bottleneck due to uneven distribution of data is reduced.
  - DBMS dependency on data partitioning is reduced due to reduction in memory access bottleneck.
- The example of this architecture are :
  - Oracle parallel server and DB2/MVS running in IBM's parallel sysplex.

1.3.3 Shared-nothing Architecture

- As shown in Fig. 1.3.3 in shared nothing architecture each nodes the disk and data is partitioned into these disks.
- DBMS is also partitioned into co-owners which resend on these disks.
- SQL query in executed occurs the nodes parallelly.
- This architecture in suitable for MPP and cluster systems.
- It is the difficult architecture for implementation due to need of new compiler and specified programming languages.

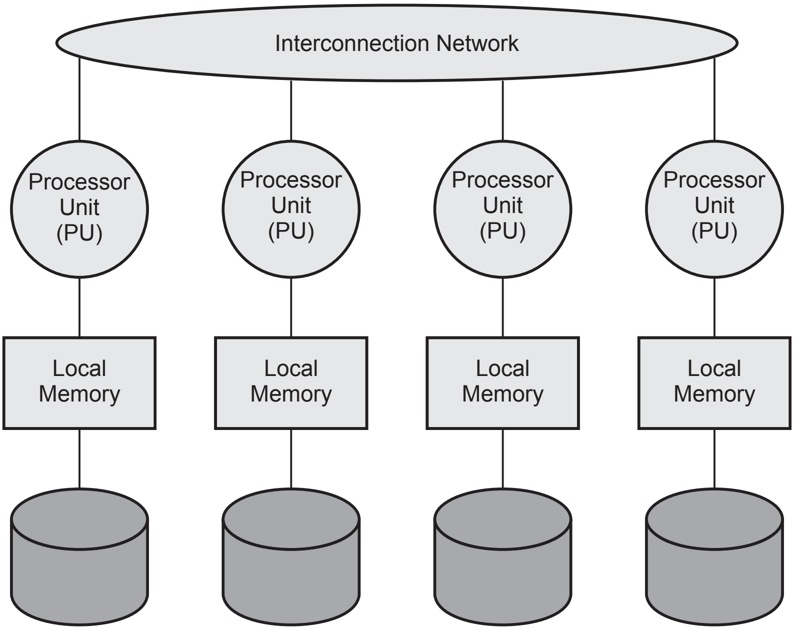


Fig. 1.3.3 Distributed memory architecture

- For implementation of parallel DBMS architecture following are the requirements of shared nothing architecture :
  - Function shipping support :
    - After parallization of SQL query the decomposed statements should be directed for execution to the processor possessing the data for execution of that query.
  - Parallel join strategies :
    - If the rows residing on same partition are joined it is called as colocated join.
    - If the rows reside on different partitions, the techniques like redirected joins need to be adapted in which rows of one table residing on partition are moved to other partition and in turn both table rows are sent to third node for joining.
    - This type of data movement from one node to another need following requirements :
      - Support for data repartitioning.
      - Query compilation
      - Support for database transactions
      - Support for the single image of database environment.

1.3.4 Combined Architecture

- It supports inter server parallelism.
- Each query in parallelized across multiple servers.
- It takes complete advantage of its operating environment.

1.4 Parallel DBMS Vendors

- A Vendor is the company that does the setup, install and maintain the **database.** The **database** is often made by another company who focuses on IT developments, new technology, or industry changes (ala FDA compliance, export law changes, etc.) that necessitate BIG changes to the **database** design.

Oracle

- Each hardware vendor implements parallel processing using operating system dependent layers. These layers serve as communication links between the operating system and the Oracle Parallel Server software
- Parallel database processing is facilitated by Parallel Server option(OPS) for loosely coupled cluster and Parallel Query Option (PQO) to run on SMPs.

Architecture :

- Fundamental component is virtual shared disk capability.
- Process based approach is used.
- PQO used shared disk architecture in which each processor node has access to all the disks.
- Parallel execution of queries is supported.
- Parallel operations like index build, database load, backup, and recovery is supported by PQO.

Data partitioning :

- Oracle 7 supports random stripping of data across multiple disks.
- Dynamic data repartitioning is supported by Oracle.

Parallel operations :

- Oracle supports hash joins.
- Optimizer generated a parallel plan instead of serial plan.
- All the queries are executed serially unless the following conditions are met :
  - The optimizer must encounter at least one full table scan.
  - The DBMS must be instructed to parallelize operations.
- PQO query coordinator breaks the query into subqueries, which are then passed to corresponding pool server processes.
- PQO supports horizontal and vertical parallelism both.

Informix

- Informix software along with sequent computers has built fully parallel DBMS engine.
- It runs on a variety of UNIX platforms.
  - Release 7 available on Windows NT.
  - Release 8(XPS) supports MPP hardware platforms including IBM SP, AT&T 3600, Sun, HP,etc.

Architecture :

- Thread based architecture, Dynamic Scalable Architecture(DSA) by Informix supports shared memory, shared disk, and shared nothing model.
- Release 7 is a shared memory implementation which supports parallel query processing and intelligent data partitioning.
- In Informix 8 partitioned table is distributed across nodes.

Data partitioning

- Online Release 7 supports round robin, schema, hash, key range, and user defined partitioning methods.
- User can define number of partitions.
- Dynamic, on line and parallel repartition is allowed.

Parallel operations

- In Online Release 7 queries, INSERTs and many utilities are executed in parallel.
- In Release 8 parallel UPDATEs and DELETs are also added.

IBM

- DB2 Parallel Edition (DB2 PE) database is based on DB2/6000 server architecture.

Architecture

- It supports shared nothing architecture.
- Excellent scalability is provided and all the database operations and utilities are fully parallelized.
- Each DB2 PE instance consists of
  - Own log
  - Own memory
  - Own storage device

Data partitioning

- Hash partitioning is supported.
- Repartitioning of the data is done across nodes when a node is added or deleted.

Parallel operations

- All database operations are fully parallel.
- Database utility operations are done at partition level.
- Function shipping and join strategies like colocated, redirected, broadcast, and repartioned joins are supported.

Sybase

- SYBASE MPP(jointly developed by Sybase and NCR) and SYBASE IQ are the products by Sybase in which parallel DBMS concepts are exploited.

Architecture

- It supports shared nothing architecture.
- SYBASE MPP is an open server application operating on the top of existing SQL servers.

- Some features like triggers or cross server integrity constraints which are specific to server are difficult to use.
- Two level optimization is performed
  - SYBASE MPP optimizes SQL statements globally.
  - Individual servers optimize SQL queries.
- It consists of three specialized servers :
  - Data Server : Consists of SQL server, Split server and Control Server.
  - DBA Server : responsible for handling optimization, DDL statements, security and global system catalog.
  - Administrative server : Its a GUI for managing SYBASE MPP.

Data partitioning

- Hash, key range and schema partitioning is supported.
- Local indexes and statistics is maintained by each SQL server.

Parallel operations

- SYBASE MPP supports horizontal parallelism.
- SQL statements and queries are executed in parallel across SQL servers.

Microsoft

- With the help of Windows NT 3.5.1, SQL Server 6 was able to provide inter query parallelism and scalability for up to four processors.

Architecture

- It supports shared everything architecture.
- SQL server is tightly integrated with the NT operating system threads.
- Cluster strategy is used to increase the scalability beyond eight processors.
- First implementation of cluster development with two node shared disk hardware model was done on Compaq and Digital running SQL Server 6.5.
- It was developed by Compaq, Digital, HP, NCR, and Tandem along with Microsoft.

1.5 DBMS Schemas for Decision Support

1.5.1 Multidimensional Data Model

- Generally for any business the marketing managers need the data focusing on every aspect and dimension of the data.
- For ex. : Instead of one dimensional question " How much is the revenue generated by product ? " they are interested in more detailed information like : How much as the revenue generated monthwise, in particular region etc.

- These dimensions can be represented in the form of curve.
- In this section different database schemas are described as they are applicable to relational database technology.

1.5.2 Star Schema

- In star schema core data information is classified in two groups :
  1) Facts 2) Dimensions.
- Facts : Analysis of core data elements. ex. individual item sell.
- Dimensions : Attributes about the facts. For ex. product type purchase date etc.
- The use is straightforward as specific facts are elaborated through set of dimensions.
- So, the fact table is larger than dimension table which may affect the performance. Refer Fig. 1.4.1 to understand star schema.

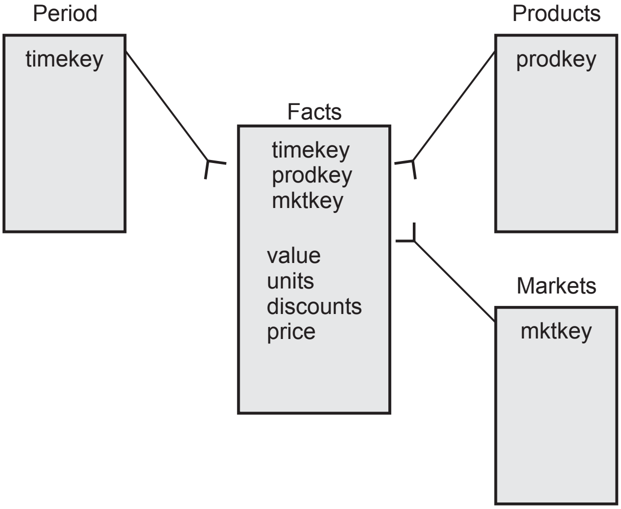


Fig. 1.5.1 Star schema

DBA viewpoint :

- Star schema consists of central fact table joined to smaller dimensional table using foreign key.
- Business facts like price, number of units sold etc. are listed is the fact table.
- Business dimensions are listed in dimensional table in the terms familiar to users.
- Dimensional table contain majority of data elements.
- Typical dimensions are time periods, promotions, discounts, account numbers etc.
- Table are typically joined by GROUP by clause.

- With star schema query processing is simpler.
- Consumer packaged goods, insurance, health care, banking etc are some applications for which star schema can be created.

Performance problems with Star Schema :

- Some of the RDBMS issues associated with star schema are

1) Indexing :

- ○ The tables in star schema contain the attributes in hierarchical form. For ex. : In period dimension the attributes will be day → week → month → quarter → year.
- ○ Due to this following challenges may be faced in indexing.
- ○ Design complexity is increased due to multiple metadata definitions.
- ○ Physical modification is needed to in updation of the fact table to manage additions and deletion.
- ○ Performance and scalability is affected as size of index is increased due to segments of compound dimensional key in fact table.

2) Level Indicator :

- For navigation of the dimensions table includes hierarchy indicator for each record.
- So to retrieve the detailed records through query this indicator must be used to get the connect result.

3) Other problems with the star schema design :

- Other than the above problems some additional problems may be faced due to relational DBMS engine and optimization technology.
- Two of such problems are :
  1. Pairwise join problem
  2. Star schema join problem

1. Pairwise join problem :

- Traditional OLTP RDBMS engines do not support rich set of complex queries.
- Only two table can be joined at a time in OLTP RDBMS.
- So such join techniques does not give proper result in data warehouse environment.

2. Star schema join problem :

- In star schema typically a fact table is related to the other tables.
- Fact table is the largest table.
- Due to this fact through pairwise join large intermediate result set is generated.

1.5.3 STAR Join and STAR Index

- To address the performance and optimization problems discussed above the technique of parallelirable multi table joins can be implemented.
- The technique is called as STAR join, which is implemented by Red Brick's RDBMS.
- STAR join is a high speed, single pass, parallelizable multi table join.
- To accelerate join performance specialized indexes are created called as STAR indexes.
- These indexes contain information to translate column value to list of rows with the value.
- STAR indexes are space efficient.
- They contain highly compressed information related to dimensions of fact table.
- The target rows of fact table are identified quickly which are of intent for a particular set of dimensions.

1.5.4 Bitmap Indexing

- SYBASE IQ is the product which uses bitmapped index structure of the data stored in SYBASE DBMS.

SYBASE IQ :

- It is a stand alone database which provides ideal data mart solution to handle multiuser ad-hoc queries.
- Data is loaded in SYBASE IQ like relational DBMS.
- SYBASE IQ converts this data into series of bitmaps. which are compressed later and stored on disk.
- It satisfies all queries within its own engine.
- It acts as read only database for data marts having size of 100 Gbytes.
- It is more effective on demoralized tables.

Data cardinality :

- Bitmap indexes take into consideration low cardinality data.
- SYBASE IQ use patented technique known as Bit-wise technology for high cardinality data.

Index types :

- Five indexing techniques are used in SYBASE IQ.
- Generally two indexes are applied to every column.

  - ○ Default index is called as fast projection.
  - ○ Second one either low - or high cardinality index.

Prejoin and adhoc join capabilities :

  - ○ Join relationships are defined in advance and indexes are built between tables.
  - ○ This proves advantageous to the uers.

- Some of the shortcomings of indexing are :
  - ○ No updates.
  - ○ Lack of core RDBMS features.
  - ○ Less advantageous for planned querries.
  - ○ High memory usage.
- Due to the reasons listed below SYBASE IQ technology archieves good performance.
- **Bitwise technology :**
  - ○ Due to use bitwise technology quick response is obtained to the querries with various data types.
- **Compression :**
  - ○ In SYBASE, data is compressed into bitmaps using sophisticated algorithms.
  - ○ One fourth amount of disk storage is alloted for data storage.
  - ○ Due to this queries can run faster.
- **Optimized memory based processing :**
  - ○ Based on nature of user queries data columns are chosen.
  - ○ This results in greater processing speed.
- **Column wise processing :**
  - ○ SYBASE IQ scans columns, which results in reduction of amount of data which is to be seached by engine.
- **Low overhead :**
  - ○ Due to optimization of an engine for decision support performance overhead is eliminated.
- **Large block I/O :**
  - ○ Block size can be managed from 512 bytes to 64K bytes.
- **Operating system level parallelism :**
  - ○ Low level operations like sorts, bitman manipulations, load etc. are broken into non blocking operations. So can run in parallel.

1.6 Characteristics of OLAP Systems

- An online analytical processing (OLAP) system is an interactive system that permits a data analyst to view different summaries of multidimensional data.
- Hence it must be able to request new summaries and get responses online, without waiting for a long response time.
- The main characteristics of OLAP are also called as **FASMI** characteristics of OLAP methods.
- **FASMI** is an alternative term for OLAP. The term was coined by Nigel Pendse
- The FASMI stands for **F**ast **A**nalysis **S**hared **M**ultidimensional **I**nformation.
- The characteristics of OLAP systems are as follows:
  - Multidimensional conceptual view : OLAP tools help in carrying slice and dice operations.
  - Multi-User Support : Since the OLAP techniques are shared, the OLAP operation should provide normal database operations, including data retrieval, update, control, integrity, and security.
  - Accessibility : OLAP acts as a middle layer between data warehouses and front-end. Storing OLAP results: OLAP results are kept separate from data sources.
  - Uniform documenting performance : Increasing the number of dimensions or database size should not degrade the reporting performance of the OLAP system.
  - OLAP supports distinction between zero values and no missing values. This is required to get correct computation of aggregation functions.
  - OLAP system should ignore all missing values and compute correct aggregate values.
  - OLAP facilitate interactive query and complex analysis for the users.
  - OLAP allows users to drill down for greater details or roll up for aggregations of metrics along a single business dimension or across multiple dimensions.
  - OLAP provides the ability to perform intricate calculations and comparisons.
  - OLAP presents results in a number of meaningful ways, including charts and graphs.

Typical OLAP operations

- Online analytical processing (OLAP) tools help analysts view data summarized in different ways, so that they can gain insight into the functioning of an organization.
- OLAP tools work on multidimensional data, characterized by dimension attributes and measure attributes.
- The data cube consists of multidimensional data summarized in different ways.
- Pre-computing the data cube helps speed up queries on summaries of data.
- Cross-tab displays permit users to view two dimensions of multidimensional data at a time, along with summaries of the data.

OLAP Operations

- As OLAP servers provides on multidimensional view of data, following are the OLAP operations -

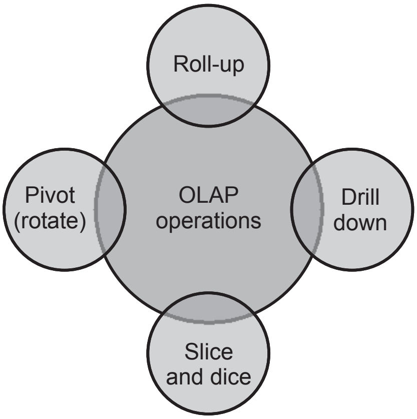


Fig. 1.6.1 Pictorial view of OLAP operations

Roll-up

- Roll-up performs aggregation on a data cube in any of the following ways -
  - By climbing up a concept hierarchy for a dimension
  - By dimension reduction
- The following diagram illustrates how roll-up works.
- Roll-up is performed by climbing up a concept hierarchy for the dimension location.
- Initially the concept hierarchy was "street < city < province < country".

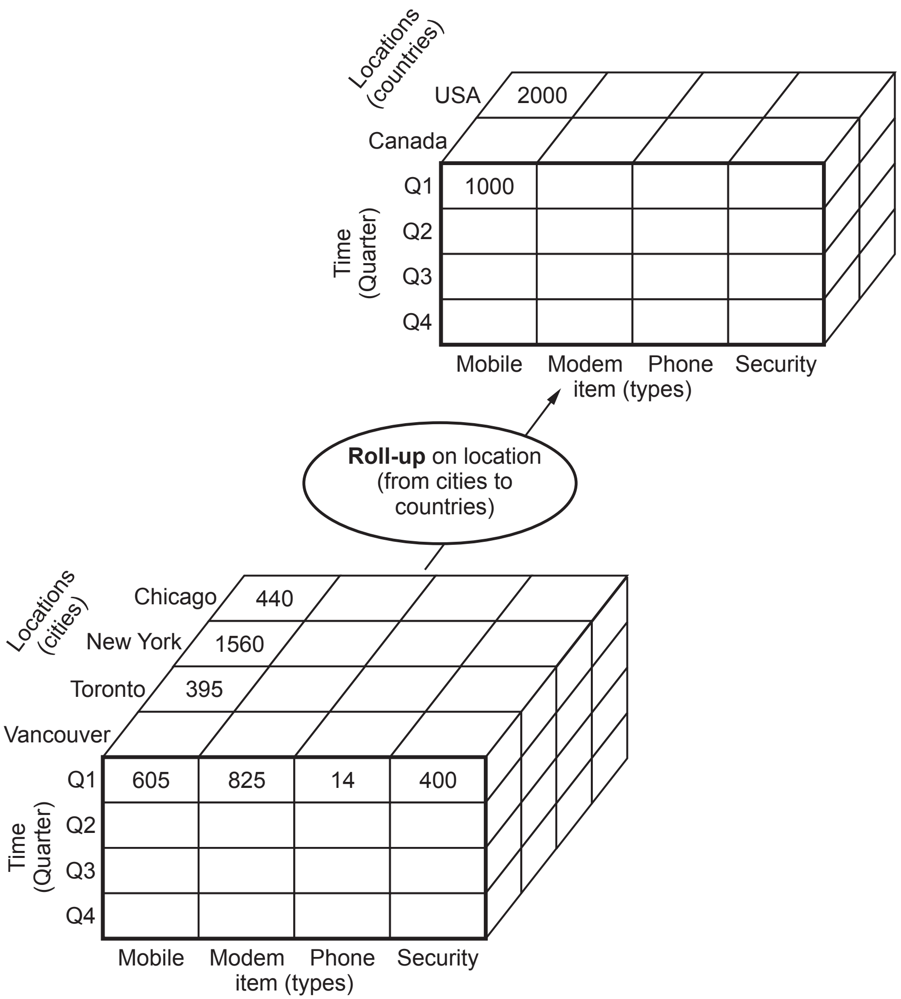


Fig. 1.6.2 Pictorial view of Rollup operations

- On rolling up, the data is aggregated by ascending the location hierarchy from the level of city to the level of country.
- The data is grouped into cities rather than countries.
- When roll-up is performed, one or more dimensions from the data cube are removed.

Drill-down

- Drill-down is the reverse operation of roll-up. It is performed by either of the following ways -
  - By stepping down a concept hierarchy for a dimension
  - By introducing a new dimension.

- The following diagram illustrates how drill-down works -

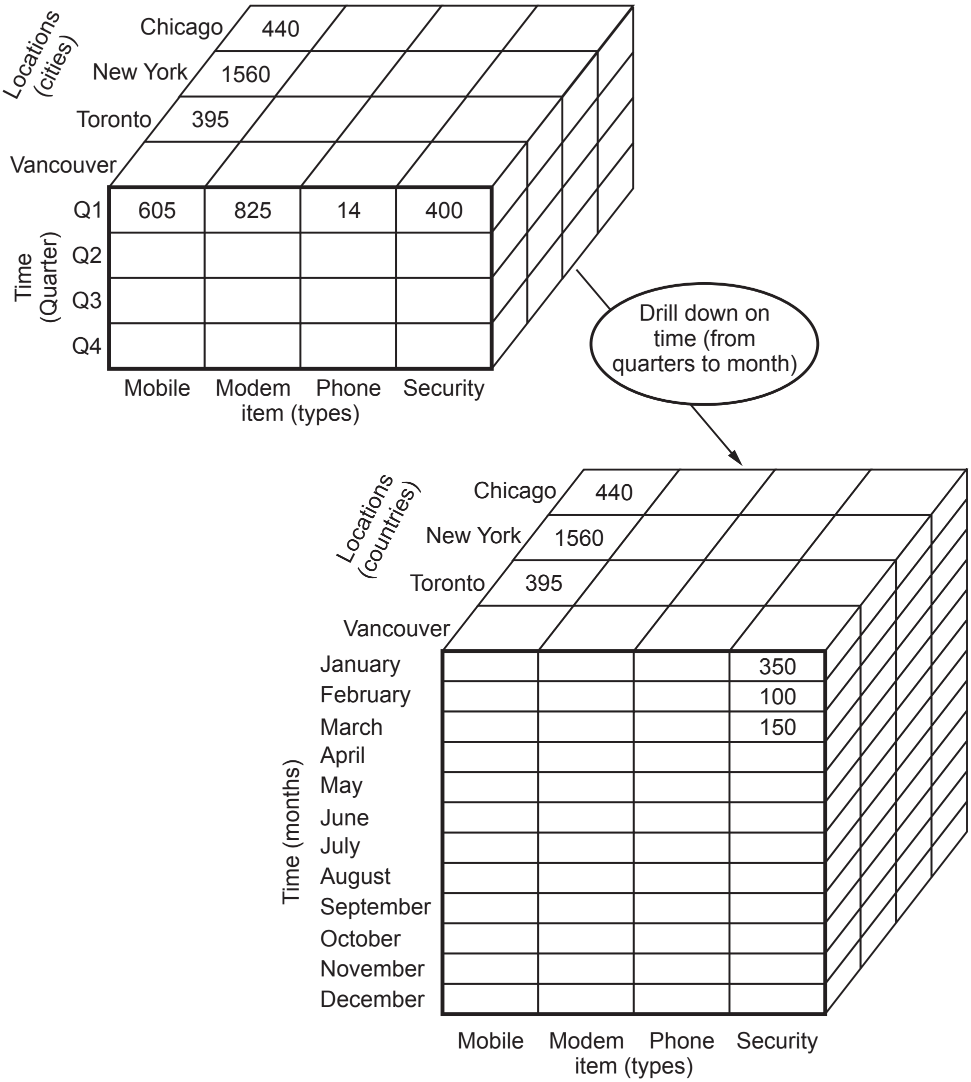


Fig. 1.6.3 Pictorial view of drill-down operations

- Drill-down is performed by stepping down a concept hierarchy for the dimension time.
- Initially the concept hierarchy was "day < month < quarter < year."
- On drilling down, the time dimension is descended from the level of quarter to the level of month.
- When drill-down is performed, one or more dimensions from the data cube are added.
- It navigates the data from less detailed data to highly detailed data.

Slice

- The slice operation selects one particular dimension from a given cube and provides a new sub-cube. Consider the following diagram that shows how slice works.
- Here slice is performed for the dimension "time" using the criterion time = "Q1".
- It will form a new sub-cube by selecting one or more dimensions.

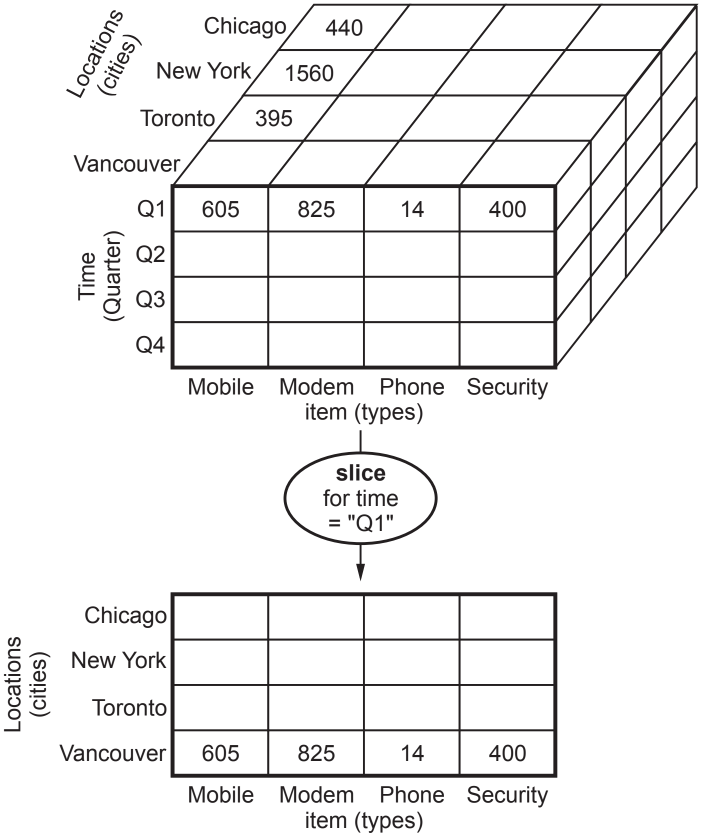


Fig. 1.6.4 Pictorial view of slice operations

Dice

- Dice selects two or more dimensions from a given cube and provides a new sub-cube. Consider the following diagram that shows the dice operation.

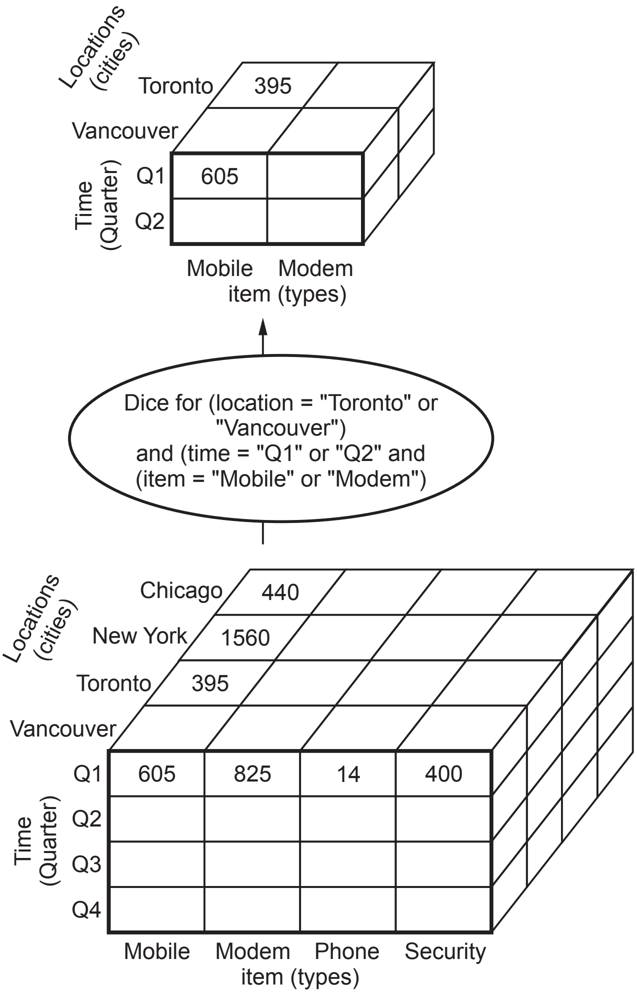
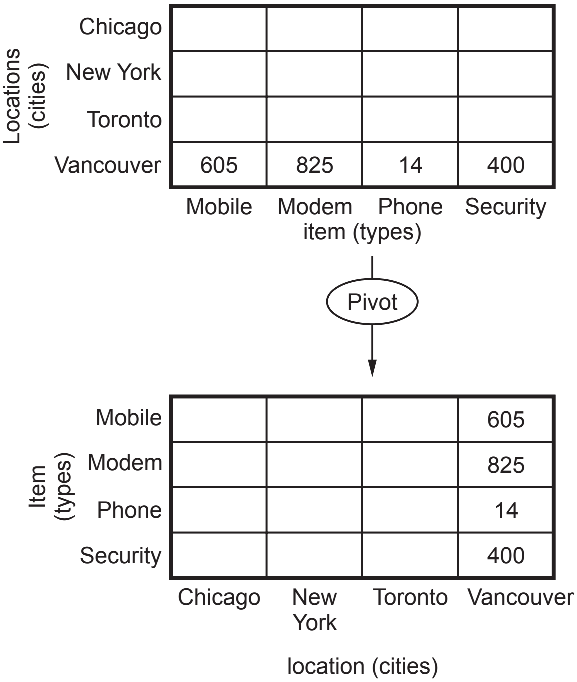


Fig. 1.6.5 Pictorial view of dice operations

- The dice operation on the cube based on the following selection criteria involves three dimensions.
  - (location = "Toronto" or "Vancouver")
  - (time = "Q1" or "Q2")
  - (item =" Mobile" or "Modem")

Pivot

- The pivot operation is also known as rotation. It rotates the data axes in view in order to provide an alternative presentation of data. Consider the following diagram that shows the pivot operation.



Locations (cities)	Mobile	Modem	Phone	Security
Chicago				
New York				
Toronto				
Vancouver	605	825	14	400

Item (types)	Chicago	New York	Toronto	Vancouver
Mobile				605
Modem				825
Phone				14
Security				400

Fig. 1.6.6 Pictorial view of pivot operations

OLAP vs. OLTP

- OLTP and OLAP both are the online processing systems.
- OLTP is a transactional processing while OLAP is an analytical processing system.
- OLTP is a system that manages transaction-oriented applications while OLAP is an online system that reports to multidimensional analytical queries
- The basic difference between OLTP and OLAP is that OLTP is an online database modifying system, whereas, OLAP is an online database query answering system.
- Below table shows database feature-wise differences between OLTP and OLAP

#	Database Features	OLTP	OLAP
1	Basic	It is an online transactional system and manages database modification.	It is an online data retrieving and data analysis system.
2	Focus	Insert, Update, Delete information from the database.	Extract data for analyzing that helps in decision making.
3	Data	OLTP and its transactions are the original source of data.	Different OLTPs database becomes the source of data for OLAP.

4	Transaction	OLTP has short transactions.	OLAP has long transactions.
5	Time	The processing time of a transaction is comparatively less in OLTP.	The processing time of a transaction is comparatively more in OLAP.
6	Queries	Simpler queries.	Complex queries.
7	Normalization	Tables in OLTP database are normalized (3NF).	Tables in OLAP database are not normalized.
8	Integrity	OLTP database must maintain data integrity constraint.	OLAP database does not get frequently modified. Hence, data integrity is not affected.

Two Marks Questions with Answers

Q.1 What is data warehouse database ?

Ans. : • Data warehouse database is usually implemented using RDBMS technology.

- The constraint faced by traditional RDBMS system is, the implementation is optimized for transactional database processing.
- To handle various attributes of data warehouse, like very large database size, adhoc query processing and to provide flexible user view creation like aggregates, multiple joins, various approaches can be adapted.
- Some of the approaches are :
  - ○ Parallel relational design using parallel computing platforms for symmetric multiprocessors (SMPs) massively parallel processors (MPPs).
  - ○ Use of new index structures for by passing relational table scans.
  - ○ Use of multidimensional databases (MDDBs) to overcome the emulation posed by nature of relational data model.

Q.2 What is the rol of sourcing, acquisition, cleanup and transformation tools ?

Ans. : • These tools are responsible for performing all the conversions, summarizations, key changes, structural changes and condensational for transforming disparate data to information which can be used by decision support tool.

- The main functions are :
  - ○ Remove unwanted data from operational database.
  - ○ Calculate summary.
  - ○ Missing data handling, etc.
- The challenges may be faced due to database heterogeneity and data heterogeneity.

Q.3 What are the approaches taken for the development of data warehouse ?

Ans. : • For development of warehouse two approaches can be taken :

1. **Top down approach** : In this enterprise data model is developed first considering various business requirements. Later warehouse is built using data mart.
2. **Bottom-up approach** : In this individual data marts are built first, which are then integrated ratio enterprise data warehouse.

Q.4 List nine decision steps in the design of a data warehouse.

Ans. : • The nine steps are listed in the Table below.

1.	Choosing the subject matter
2.	Deciding what a fact table represents
3.	Identifying and conforming the dimensions
4.	Choosing the facts
5.	Storing precalculations in the fact table
6.	Rounding out the dimension tables
7.	Choosing the duration of the database
8.	The need to track slowly changing dimensions.
9.	Deciding the query priorities and the query modes.

Q.5 List logical steps needed to build a data warehouse.

Ans. : • To build a data warehouse following logical steps need to be taken :

- ○ Business requirement collection and analysis.
- ○ Data model and physical design for warehouse.
- ○ Define data sources.
- ○ Database technology and platform selection for warehouse.
- ○ Data extraction, transformation, cleaning and loading into database.
- ○ Access and reporting tool selection for the database.
- ○ Selection of database correctively software.
- ○ Selection and data analysis and presentation software.
- ○ Data warehouse updation.

Q.6 What selection criteria need to be considered for transformation, consolidation, integration and repairing of data ?

Ans. : • Following selection criteria related to transformation, consolidation, integration, repairing of data should be considered :

- Identification of data in the data source environment is important.
- Flat files, indexed files and legacy DBMS should be supported as most of the data is still stored in these formats.
- Merging data from different data stores is important.
- Datatype and character set translation is needed.
- Summarization, aggregation, etc. capabilities are needed.
- Evaluation of vendor stability is needed.

Q.7 Explain tangible benefits of data warehousing.

Ans. : Tangible benefits :

- One of the major benefit is out of stock conditions can be improved
- Some additional benefits are :
  - It provides big picture of purchasing and inventory patterns which facilitates cost saving.
  - Business intelligence can be enhanced by proper market analysis.
  - Cost effective decisions can be made by spectrum of ad-hoc query processing and operational databases.
  - Target market selection is improved resulting in decrease in cost of product introduction. Also improvement product inventory turnover is observed.

Q.8 Explain Intangible benefits of data warehousing.

Ans. : Intangible benefits :

- Following are the intangible benefits of data warehouse :
  - As all the required data can be kept at a single location productivity is improved.
  - Overlap in decision support applications is reduced by reduction in redundant processing.
  - Customer relations are enhanced as individual requirements and trends can be better understood.
  - Useful insights are provided into work processing, through which the processes can be re-engineered by inclusion of innovative ideas.

Q.9 Explain architecture of ORACLE as a parallel DBMS vendor.

Ans. : Architecture

- Fundamental component is virtual shared disk capability.

- Process based approach is used.
- PQO used shared disk architecture in which each processor node has access to all the disks.
- Parallel execution of queries is supported.
- Parallel operations like index build, database load, backup, and recovery is supported by PQO.

Q.10 Explain data partitioning in informix vendor.

Ans. : Data partitioning

- Online Release 7 supports round robin, schema, hash, key range, and user defined partitioning methods.
- User can define number of partitions.
- Dynamic, on line and parallel repartition is allowed.

Q.11 Explain microsoft as a parellel DBMS vendor.

Ans. : Microsoft

- With the help of Windows NT 3.5.1, SQL Server 6 was able to provide inter query parallelism and scalability for up to four processors.

Architecture

- It supports shared everything architecture.
- SQL server is tightly integrated with the NT operating system threads.
- Cluster strategy is used to increase the scalability beyond eight processors.
- First implementation of cluster development with two node shared disk hardware model was done on Compaq and Digital running SQL Server 6.5.
- It was developed by Compaq, Digital, HP, NCR, and Tandem along with Microsoft.

Q.12 Explain briefly star schema.

Ans. : Star Schema

- In star schema core data information is classified in two groups :
  1) Facts 2) Dimensions.
- Facts : Analysis of core data elements. ex. individual item sell.
- Dimensions : Attributes about the facts. For ex. product type purchase date etc.
- The use is straightforward as specific facts are elaborated through set of dimensions.
- So, the fact table is larger than dimension table which may affect the performance. Refer Fig. 1.5.1 to understand star schema. (Refer Fig. 1.5.1 from page 1 - 22).

Q.13 Explain STAR join and STAR index.

Ans. : STAR join and STAR index

- To address the performance and optimization problems discussed above the technique of parallelirable multi table joins can be implemented.
- The technique is called as STAR join, which is implemented by Red Brick's RDBMS.
- STAR join is a high speed, single pass, parallelizable multi table join.
- To accelerate join performance specialized indexes are created called as STAR indexes.
- These indexes contain information to translate column value to list of rows with the value.
- STAR indexes are space efficient.
- They contain highly compressed information related to dimensions of fact table.
- The target rows of fact table are identified quickly which are of intent for a particular set of dimensions.

Q.14 Explain SYBASE IQ.

Ans. : SYBASE IQ

- It is a stand alone database which provides ideal data mart solution to handle multiuser ad-hoc queries.
- Data is loaded in SYBASE IQ like relational DBMS.
- SYBASE IQ converts this data into series of bitmaps. which are compressed later and stored on disk.
- It satisfies all queries within its own engine.
- It acts as read only database for data marts having size of 100 Gbytes.
- It is more effective on demoralized tables.

Q.15 What are characteristics of OLAP systems ?

Ans. : The characteristics of OLAP systems are as follows :

- Multidimensional conceptual view : OLAP tools help in carrying slice and dice operations.
- Multi-User Support : Since the OLAP techniques are shared, the OLAP operation should provide normal database operations, including data retrieval, update, control, integrity, and security.

- ○ Accessibility : OLAP acts as a middle layer between data warehouses and front-end. Storing OLAP results: OLAP results are kept separate from data sources.
- ○ Uniform documenting performance : Increasing the number of dimensions or database size should not degrade the reporting performance of the OLAP system.

Q.16 Explain Roll up operation in OLAP.

Ans. : • Roll-up performs aggregation on a data cube in any of the following ways -

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

- Roll-up is performed by climbing up a concept hierarchy for the dimension location.
- Initially the concept hierarchy was "street < city < province < country".
- On rolling up, the data is aggregated by ascending the location hierarchy from the level of city to the level of country.
- The data is grouped into cities rather than countries.
- When roll-up is performed, one or more dimensions from the data cube are removed.

Q.17 Explain Drill - down operation in OLAP.

Ans. : • Drill-down is the reverse operation of roll-up. It is performed by either of the following ways -

- ○ By stepping down a concept hierarchy for a dimension
- ○ By introducing a new dimension.

- The following diagram illustrates how drill-down works -
- Drill-down is performed by stepping down a concept hierarchy for the dimension time.
- Initially the concept hierarchy was "day < month < quarter < year."
- On drilling down, the time dimension is descended from the level of quarter to the level of month.
- When drill-down is performed, one or more dimensions from the data cube are added. Fig. 1.6.3 illustrates working of drill - down. (Refer Fig. 1.6.3 from page 1 - 29).

Q.18 Compare OLAP Vs. OLTP.

Ans. :

#	Database Features	OLTP	OLAP
1	Basic	It is an online transactional system and manages database modification.	It is an online data retrieving and data analysis system.
2	Focus	Insert, Update, Delete information from the database.	Extract data for analyzing that helps in decision making.
3	Data	OLTP and its transactions are the original source of data.	Different OLTPs database becomes the source of data for OLAP.
4	Transaction	OLTP has short transactions.	OLAP has long transactions.
5	Time	The processing time of a transaction is comparatively less in OLTP.	The processing time of a transaction is comparatively more in OLAP.
6	Queries	Simpler queries.	Complex queries.
7	Normalization	Tables in OLTP database are normalized (3NF).	Tables in OLAP database are not normalized.
8	Integrity	OLTP database must maintain data integrity constraint.	OLAP database does not get frequently modified. Hence, data integrity is not affected.

❑❑❑

Notes

Unit - II

2 Data Mining - Introduction

Syllabus

Introduction to Data Mining Systems – Knowledge Discovery Process – Data Mining Techniques – Issues – Applications – Data Objects and attribute types, Statistical description of data, Data Preprocessing – Cleaning, Integration, Reduction, Transformation and discretization, Data Visualization, Data similarity and dissimilarity measures.

Contents

2.1 Introduction to Data Mining

- Data mining is a field of research that has emerged in the 1990s, and is very popular today.
- Data mining is a set method that applies to large and complex databases. This is to eliminate the randomness and discover the hidden pattern.
- As these data mining methods are almost always computationally intensive, we use data mining tools, methodologies and theories for revealing patterns in data.
- There are too many driving forces present. And this is the reason why data minig has become such an important ares of study.
- The reason why data mining has become popular is that, storing data electronically has become very cheap and that trasfeering data can now be done very quickly thanks to the fast computer networks that we have today.
- Thus, many organizations now have huge amounts of data stored in databases, that needs to be analyzed.
- Traditionally, data has been analyzed by hand to discover interesting knowledge. However, this is time-consuming, prone to error, doing this may miss some importatnt information, and it is just not realistic to do this on large databases.
- To address this problem, automatic techniques have been designed to analyze data and extract interesting patterns, trends or other useful information. This is the purpose of data mining.
- In general, data mining techniques are designed either to explain or understand the past (e.g. Why a plane has crashed) or predict the future (e.g. Predict if there will be an earthquake tomorrow at a given location).

2.2 Knowledge Discovery Process

- Data mining can also be termed as knowledge discovery from data as KDD.
- The discovery process consists of seven steps as shown in the Fig. 2.2.1. (See Fig. 2.2.1 on next page).
- These steps are :

1) Data cleaning : For removing noise and inconsistent data.

2) Data integration : For combining multiple data sources.

3) Data selection : For retrieves of relevent data from database.

4) Data transformation : Summary or aggregation operations are performed for transforming the data.

5) Data mining : Data patterns are extracted by applying intelligent methods.

6) Pattern evaluation : Interestingness measures are applied for identification of interesting patterns.

7) Knowledge presentation : Mined knowledge is presented to users by visualization and knowledge representation techniques.

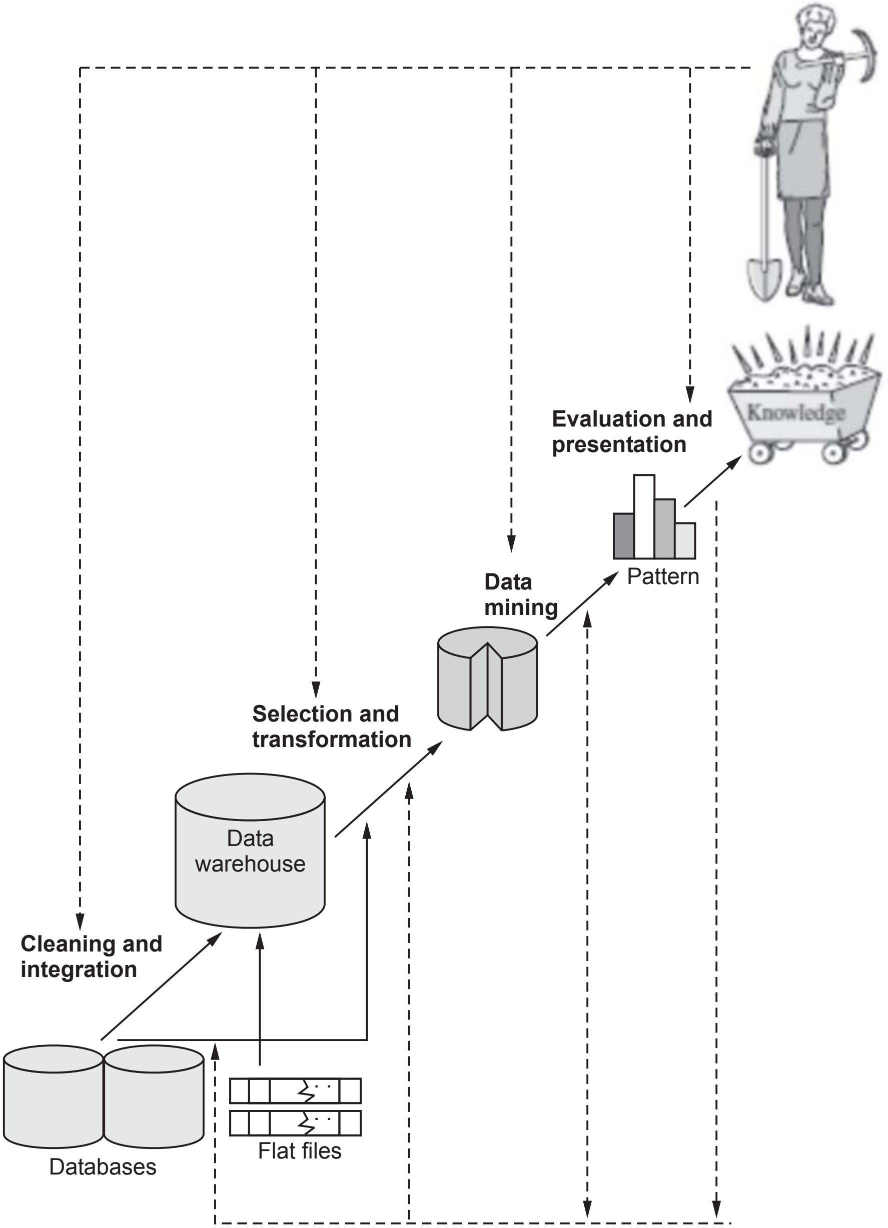


Fig. 2.2.1 Data mining as a step in the process of knowledge discovery

2.3 Data Mining Techniques

- Some of the techniques which are used in development of data mining methods are :

2.3.1 Statistics

- The collection, analysis, interpretation and presentation of data can be studied by applying statistics.
- By application of statistical model the behaviour of objects in a target class is described.
- Statistical methods are used to summarize an describe collection of data.
- These are also used to verify data mining results.
- A statistical hypothesis test works on experiential data to make statistic decision.

2.3.2 Machine Learning

- These techniques facilitate computes to learn the complex patterns and make intelligent decision on data automatically.
- For example : Automatic recognition of postal codes on mail from set of examples.
- Different ways through which it can be done as

1) Supervised learning :

- It is a classification technique.
- Based on the labeled examples in training data set it facilitates the supervised learning of classification model.

2) Unsupervised learning :

- It is a clustering technique.
- Classes are discovered within the data.
- As the data is not labeled the meaning of clusters cannot be found.
- For example : From set of handwritten documents cluster can be formed for digits from 0 to 9.

3) Semi supervised learning :

- It uses both labeled and unlabeled data in learning process.
- There are different approaches. In one, unlabeled examples are considered as boundary elements. In another, labeled and unlabeled examples are considered as positive and negative examples.
- This can be understood from Fig. 2.3.1.

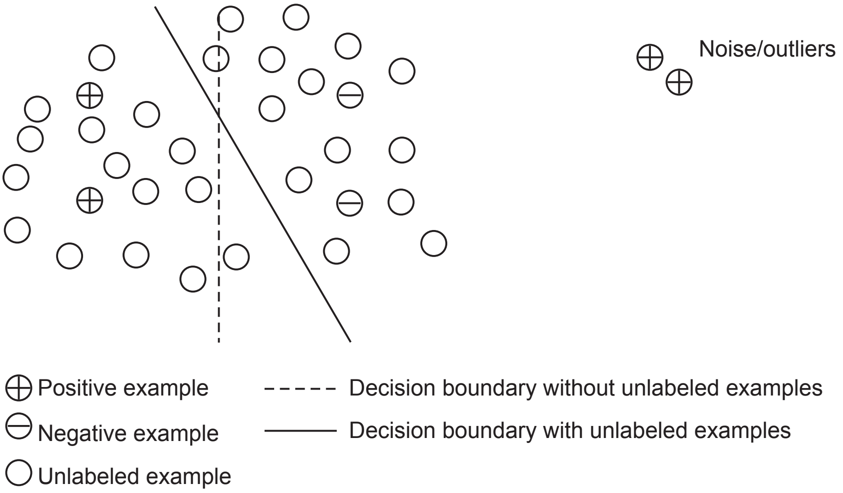

Noise/outliers
⊕ Positive example
⊖ Negative example
○ Unlabeled example
----- Decision boundary without unlabeled examples
—— Decision boundary with unlabeled examples

Fig. 2.3.1 Semi - supervised learning

4) Active learning :

- In this approach user is asked to label an example.
- Based on knowledge from human users the model quality is optimized.

2.3.3 Database Systems and Data Warehouses

- To create, maintain and use the databases for organizations and end user, is the aim.
- Various techniques like query languages, query processing and optimization methods, data storage and indexing and accessing methods are used to facilitate this.
- A data warehouse integrates data from various sources and timeframes.

2.3.4 Information Retrieval

- Documents or information from the documents are searched by this technique.
- The system assumes that -
  1) The data to be searched is unstructured.
  2) The queries are formed by keywords.
- Language model is the probability density function which is used to generate bag of words.
- The topic in text documents can also be modeled as probability distribution over vocabulary known as topic model.

2.4 Major Issues in Data Mining

- Major issues in data mining can be categorized in five groups.
  1. Mining methodology
  2. User interaction
  3. Efficiency and scalability
  4. Diversity of data types
  5. Data mining and society

2.4.1 Mining Methodology

- Mining methodologies are required for investigation of new kinds of knowledge, mining in multidimensional space, etc.
- Some of the issues faced by these methodologies are data uncertainty, noise and incompleteness.
- Various aspects of mining methodologies are :

1. Mining various and new kinds of knowledge :

- ○ The issues are faced due to diversity of applications.

2. Mining knowledge in multidimensional space :

- ○ If data sets are large, the interesting patterns can be found in combinations of dimensions with different abstraction levels.
- ○ To mine this data multi dimensional data mining techniques are used.

3. Data mining - an interdisciplinary effort :

- ○ For interdisciplinary fields like natural language text mining, mixing methods need to be combined with methods of information retrieval and natural language processing.

4. Boosting the power of discovery in a networked environment :

- ○ The objects may reside in interconnected environment like web. Semantic links between multiple data objects is advantages in mining.

5. Handling uncertainty, noise or incompleteness of data :

- ○ Mining process face the issues due to noise, errors, expections, etc.

6. Pattern evaluation and pattern guided mining :

- ○ The techniques are required to know the interesting patterns among all the patterns.

2.4.2 User Interaction

- User is important factor of data mining process.
- The areas of research include -
  1. Interactive mining.
  2. Incorporation of background knowledge.
  3. Adhoc mining and data mining query languages.
  4. Presentation and visualization of data mining results.

1. Interactive mining :

- Flexible user interfaces are necessary.
- The interface should provide facility to the user to first sample set of data, explore its characteristics and estimate mining results.

2. Incorporation of background knowledge :

- Background discovery process is necessary for pattern evaluation.

3. Adhoc data mining and data mining query languages :

- Query languages like SQL allow users to pose adhoc queries.
- This facilities domain knowledge, knowledge to be mind, conditions and constants, etc.

4. Presentation and visualization of data mining results :

- The presentation of data mining results is very important for easy understanding of the results.

2.4.3 Efficiency and Scalability

- The factors need to be considered while comparing mining algorithms are :

  1. Efficiency and scalability of data mining algorithms :

  - The running time of timing algorithm must be small and predictable.
  - The criterias need to be considered for mining algorithms are : efficiency, scalability, performance, optimization and ability to execute in real line.

  2. Parallel, distributed and incremental mining algorithms :

  - Due to humongous size of data sets there is need of such algorithms.
  - Cloud counting and cluster computing can facilitate parallel data mining.

2.4.4 Diversity of Database Types

- Database types include :

  1. Handling complex types of data :

  - The data objects can vary from simple to temporal, biological sequences, sensor data, social network data, etc.

- ○ To handle such a varied data effective and efficient data mining tools are needed.

2. Mining dynamic, networked and global data repositories :

- ○ The knowledge discovery from structured, semi-structured or unstructured interconnected data is very challenging.
- ○ Web mining, multisource data mining techniques are needed to handle this data.

2.4.5 Data Mining and Society

- There are many issues which need to addressed while using data mining in day to day life.
- Some of these are :

1. Social impact of data mining
2. Privacy preserving data mining
3. Invisible data mining

2.5 Applications of Data Mining

- Data mining has many applications in various domains.
- In this section we are going to discuss two successful applications of data mining

1. Business intelligence
2. Search engines

2.5.1 Business Intelligence

- The term Business Intelligence (BI) refers to technologies, applications and practices for the collection, integration, analysis and presentation of business information.
- The purpose of BI is to support better business decision making.
- Data mining is required in BI to perform effective market analysis, compare customer feedback on similar products to discover strengths and weakness of their competitors to retain highly valuable customers and to make smart business decisions.
- Data warehousing and multidimensional data mining is used in online analytical processing tool.
- Classification and prediction techniques are used in predictive analytics.
- Clustering is used in customer relationship management.

2.5.2 Web Search Engines

- A web search engine or internet search engine is a software system that is designed to carry out web search, which means to search the world wide web in a systematic way for particular information specified in a textual web search query.
- User query results are returned at a list or hits.
- The hits consist of web pages, images and other types of files.
- Different data mining techniques are used extensively in web search engines.
- Crawling, indexing and searching are some of them.
- Challenges which can be faced by data mining usage are :
  - Handling of humongous amount of data getting generated daily.
  - Use of computer clouds, consisting of thousands or hundreds of thousands of computers to work on data mining methods and large distributed data sets.
  - To deal with online data. A query classifier need to be built for this to handle the queries on predifined categories.
  - To handle context aware queries. In context aware query search engine tries to find out context of query using users profile to give customized answers in very small amount of time.
  - Most of the queries are asked only once which is challenging for data mining methods.

2.6 Data Objects and Attribute Types

- The data mining process starts with preparation of data.
- It is observed that real world data is humongous and generated from a mixture of different sources and generally contains noise.
- The following information about the data makes it useful for preprocessing :

  a. Types of attributes

  b. Discrete and continuous-valued attributes

  c. Distribution of the values

  d. How data can be visualized better

  e. Outliers present

  f. Similarity of the objects
- A data object is an entity in a data set.
- Data objects are also known as samples, examples, instances, data points, or objects.

- For ex. In a medical database objects are patients, in a university database, the objects may be students, professors and courses, etc.
- Data objects are characterized by attributes.
- Data objects are stored in a database in the form of data tuples.
- Rows of a database contain data objects and columns contain attributes.

Attributes :

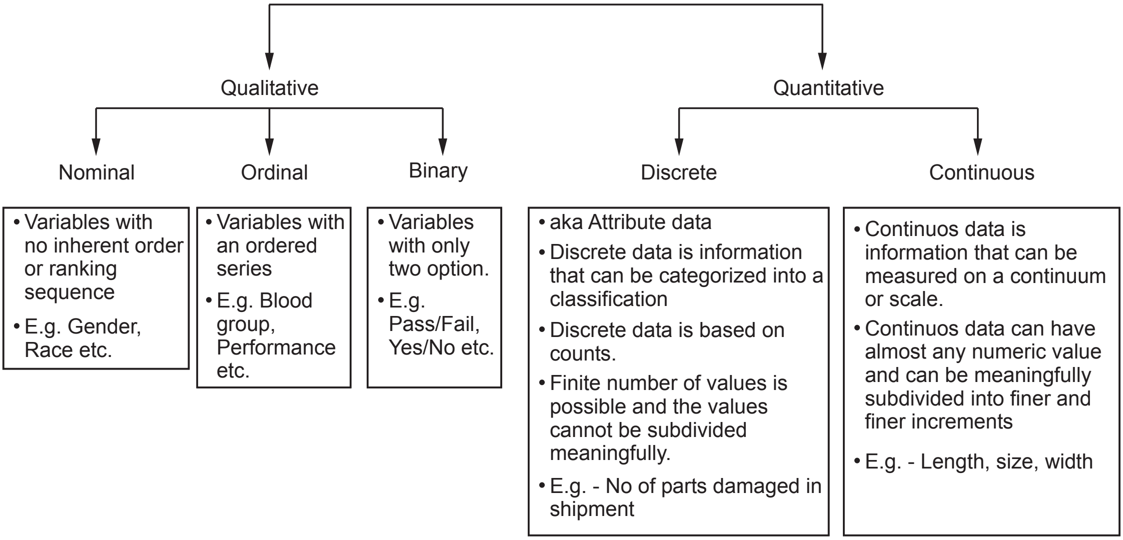
- Attribute is a data field which represents features of a data object.
- It is also known as dimension (used in data warehouse), feature (used in machine learning) or variable(used in statistics).
- For example : for the object customer, attributes are customer ID, name, and address.
- **Observations** are observed values for attributes.
- **Attribute Vector or Feature Vector** consists of set of attributes for describing a given object.
- **Univariate** indicates distribution of data for one attribute and so on.
- Based on the set of possible values attributes can be categorized as :
  1. Nominal
  2. Binary
  3. Ordinal
  4. Numeric

1. Nominal Attributes :

- Nominal attributes relate to names.
- Values of these attributes are :
  - Symbols or
  - Names of things
- Values are also called as enumerations.
- They are also known as categorical attributes as the value of the attribute can be
  - Category
  - Code or
  - State
- The attributes are not arranged in meaningful order.

Types of Attributes

Attribute Type	Description	Examples
Nominal /Binary	The values are just different names that provides only enough information to distinguish one object from another (=, ≠)	Zip codes, employee ID numbers, eye color gender
Ordinal	The values provides enough information to order objects (<, >)	pain level, rating, grades, street numbers
Interval	The differences between values are meaningful i.e., a unit of measurement exists (+, –)	calendar in Celsius or Fahrenheit
Ratio	Both differences and ratios are meaningful. (*, /)	temperature in Kelvin monetary quantities, counts, age, mass lengths.



- Variables with no inherent order or ranking sequence
- E.g. Gender, Race etc.

- Variables with an ordered series
- E.g. Blood group, Performance etc.

- Variables with only two option.
- E.g. Pass/Fail, Yes/No etc.

- aka Attribute data
- Discrete data is information that can be categorized into a classification
- Discrete data is based on counts.
- Finite number of values is possible and the values cannot be subdivided meaningfully.
- E.g. - No of parts damaged in shipment

- Continuos data is information that can be measured on a continuum or scale.
- Continuos data can have almost any numeric value and can be meaningfully subdivided into finer and finer increments
- E.g. - Length, size, width

Fig. 2.6.1 Types of Attributes

- Some more examples of nominal attributes are :

Attribute	Possible Values
hair color	black,brown, blond, red, auburn, gray, and white
marital status	single, married, divorced, and widowed
Occupation	teacher, dentist, programmer, farmer,etc.

- The values in the above table can also be represented by numbers. For ex :

  Black colour : code 0

  Brown colour : code 1, etc.
- Note that these numbers cannot be used quantitatively as they does not contain any meaning with respect to numerical operations. We cannot perform mathematical operations like addition, subtraction, etc. on them.
- In turn the operations like calculation of mean (average) value or median (middle) value for are also not possible.
- The occurrence can be calculated by calculation of mode.
- In turn the operations like calculation of mean (average) value or median (middle) value for are also not possible.

2. Binary Attributes :

- A binary attribute can take only two states, either a 0 or 1.
- State 0 : attribute is absent, State 1 : attribute is present.
- They are also known as Boolean attributes if states are true or false.
- The examples of binary attributes are shown in the below table :

Attribute	Values
Cancer detected	Yes, No
result	Pass, Fail

Categorical Value	Integer Value	x_1	x_2	x_3
awful	0	0	0	0
poor	1	0	0	1
OK	2	0	1	0
good	3	0	1	1
great	4	1	0	0

- If both the states in binary attribute contain equal weightage (for ex : Attribute gender with states male and female) it is said to be symmetric.
- If both the states in binary attribute are not equally important (for ex : Attribute medical test for HIV with states positive and negative) it is said to be asymmetric.

3. Ordinal Attributes :

Ordinal Data

- In an ordinal attribute the values are arranged in a meaningful order or rank.
- However even if the values are in meaningful sequence the magnitude between them is missing.
- For ex :

In the above figure amongst the three possible values hot, hotter and hottest it is difficult to figure out the intensity of each value.

- Some more examples are listed below :

Attribute	Values
Grade	A+, A, A–, B+
Proffesional rank for professors	Assistant, associate and full

- Ordinal attributes focus on subjective assessments of qualities that as these quantities cannot be measured objectively.
- This feature is useful in the application like conduction of a survey, where ratings are important. For ex. In the survey of any product the ordinal categories can be : 0 : very dissatisfied, 1 : somewhat dissatisfied, 2 : neutral, 3 : satisfied, and 4 : very satisfied.
- The central tendency of an ordinal attribute can be calculated by its mode and median (the middle value in an ordered sequence), but the mean cannot be defined.

The three attributes Nominal, Binary and ordinal explained in the above section are qualitative.
They don't provide the actual size or quantity of an object, generally by words assigned to a specific category.

2.6.1 Numeric Attributes

- They are quantitative attribute which can be measured and typically represented in integer or real values.
- They can be
  1. Interval-scaled or 2. ratio-scaled.

1. Interval-Scaled Attributes :

- These attributes are measured on a scale of equal-size units.
- They follow specific order and can have positive, negative or zero value.

- They can be used to compare and quantify the difference between values.

2. Ratio-Scaled Attributes :

- These are numeric attribute with an inherent zero-point that means we can consider the value as a multiple of another value.
- They follow specific order and difference between values, mean, median, and mode can be calculated.

Discrete versus Continuous Attributes :

- A discrete attribute is finite or countably infinite set of values, which may or may not be represented as integers.
- For ex : hair color, smoker, medical test, and drink size, or numeric values 0 to 110 for the attribute age. etc.
- The attributes customer ID can also have countably infinite values.
- Continuous values are typically real numbers.In practice continuous attributes are typically represented as floating-point variables.

2.7 Statistical Description of Data

- For data preprocessing to be successful it is essential to have an overall picture of the data.
- Statistical descriptions is used to
  - identify properties of the data
  - highlight important data values
  - Identify noise or outliers
- For practical utilization various techniques are used to show statistical data in the graphical / visual presentation.

Measuring the Central Tendency of data

- The measures of central tendency provide us with statistical information about a set of data.
- The four primary measurements that we use are the **mean, median, mode** and range.
- Each one of these measurements can provide us with information about our data set.
- This information can then be used to define how the set of are connected.
- Let’s consider the Items as the set of data for understanding central tendency of data.
- The table shows the Quantity and cost of items as

Quantity of items	Cost of items
9	30
5	36
7	47
2	50
5	52
6	52
3	56
7	60
2	63
3	70
1	70
2	110

Table 2.7.1 Sample data - items

Mean

- The first measure is the mean which means average.
- To calculate the mean add together all of the numbers in your data set.
- Then divide that sum by the number of addends.
- Let $x_1 x_2 ... x_N$ be a set of N values or observations such as for some numeric attribute X then The mean of this data is

$$\bar{x} = \frac{\sum_{i=1}^{N} x_i}{N} = \frac{x_1 + x_2 + ... + x_N}{N}$$

- For the given data set of items the mean is calculated as

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

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

Arithmetic Mean or the Weighted Average

- Sometimes each value x_i in a set may be associated with a weight wi for i = 1...N.
- The weights reflect the significance importance or occurrence frequency attached to their respective values.

- The weighted arithmetic mean or the weighted average is calculated as

$$\bar{x} = \frac{\sum_{i=1}^{N} w_i x_i}{\sum_{i=1}^{N} w_i} = \frac{w_1 x_1 + w_2 x_2 + \ldots + w_N x_N}{w_1 + w_2 + \ldots + w_N}$$

- For the given data set of items the quantity of the items is considered as weight w. Hence the arithmetic mean is calculated as

$$\frac{(30\times 9)+(36\times 5)+(47\times 7)+(50\times 2)+(52\times 5)+(52\times 6)+(56\times 3)+(60\times 7)+(63\times 2)+(70\times 3)+(70\times 1)+(110\times 2)}{(9+5+7+2+5+6+3+7+2+3+1+2)}$$

$= \quad 51.52$

- To offset the effect caused by a small number of extreme values one can use **trimmed mean.**
- The trimmed mean is obtained after chopping off values at the high and low extremes.

Median

- Another measure of central tendency is the which is the middle number when listed in order from least to greatest.
- For skewed (asymmetric) data a better measure of the center of data is the median.
- It is the middle value in a set of ordered data values.
- The median is the value that separates the higher half of a data set from the lower half
- To calculate median
  - Suppose that a given data set of N values for an attribute X is sorted in increasing order.
  - If N is odd then the median is the middle value of the ordered set.
  - If N is even then the median is not unique; it is the two middlemost values and any value in between.
  - If X is a numeric attribute in this case by convention the median is taken as the average of the two middlemost values.
- For the sample dataset given we can find the median of the ‘Cost of items’ which has a total 12 values. 12 ia an even number. Hence the median is taken as average of middlemost values i.e. average of 52 and 56. Hence median is 54.
- The median is expensive to compute when we have a large data set then following formula is used to calculate median as

$$\text{Median} \;=\; L_1 + \left(\frac{N/2 - \left(\sum \text{freq}\right)_l}{\text{freq}_{\text{median}}} \right) \text{width}$$

where

- $\boldsymbol{L_1}$ is the lower boundary of the median interval
- **N** is the number of values in the entire data set
- $\left(\sum \text{freq}\right)_l$ is the sum of the frequencies of all of the intervals that are lower than the median interval
- $\mathit{freq}_{\mathit{median}}$ is the frequency of the median interval
- **Width** is the width of the median interval.

Mode

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

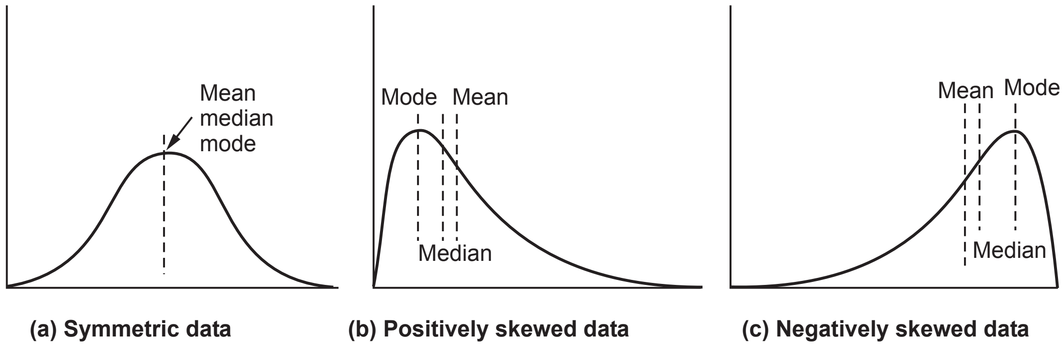
$$\text{Mean} - \text{Mode} \;\approx\; 3 \times (\text{Mean} - \text{Median})$$

- The midrange can also be used to assess the central tendency of a numeric data set. It is the average of the largest and smallest values in the set

$$\frac{\text{Maximum (data)} + \text{Minimum (data)}}{2}$$

- In a unimodal frequency curve with perfect symmetric data distribution, the mean, median, and mode are all at the same center value as shown in Fig. 2.7.1
- Data in most real applications are not symmetric. They may instead be either positively skewed, where the mode occurs at a value that is smaller than the median, or negatively skewed, where the mode occurs at a value greater than the median (Refer Fig. 2.7.1).

Measuring the Dispersion of Data



(a) Symmetric data (b) Positively skewed data (c) Negatively skewed data

Fig 2.7.1 Mean, median, and mode of symmetric versus positively and negatively skewed data

- The spread of dispersion of data can be measured statistically.
- The measures include **range, quantiles, quartiles, percentiles,** and **the interquartile range**.
- The **five-number summary**, which can be displayed as a boxplot, is useful in identifying outliers.
- **Variance** and **standard deviation** also indicate the spread of a data distribution.

Range

- Let $x_1, x_2, ..., x_N$ be a set of observations for some numeric attribute, X.
- The range of the set is the difference between the largest (max()) and smallest (min()) values.

Quartiles

- Quantiles are points taken at regular intervals of a data distribution, dividing it into essentially equal size consecutive sets.
- Suppose that the data for attribute X are sorted in increasing numeric order. Imagine that we can pick certain data points so as to split the data distribution into equal-size consecutive sets. These data points are called quantiles. Refer Fig. 2.7.2
- The k^{th} q-quantile for a given data distribution is the value x such that at most k/q of the data values are less than x and at most (q – k)/q of the data values are more than x, where k is an integer such that $0 < k < q$. There are q – 1 q-quantiles.
- The 2-quantile is the data point dividing the lower and upper halves of the data distribution. It corresponds to the median.

- The 4-quantiles are the three data points that split the data distribution into four equal parts; each part represents one-fourth of the data distribution. They are more commonly referred to as quartiles.
- The 100-quantiles are more commonly referred to as percentiles; they divide the data distribution into 100 equal-sized consecutive sets. The median, quartiles, and percentiles are the most widely used forms of quantiles.

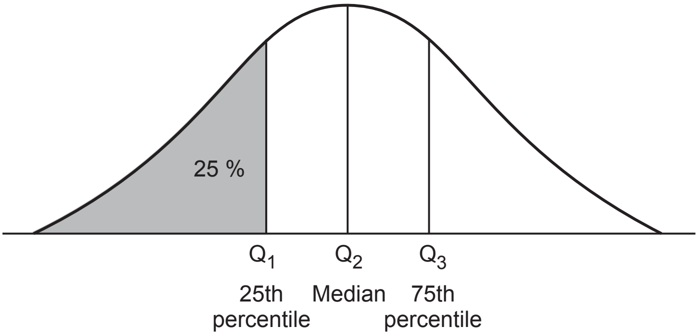


Fig 2.7.2 A plot of the quantile distribution for some attribute X.

Interquartile Range

- The quartiles give an indication of a distribution's center, spread, and shape. The first quartile, denoted by Q1, is the 25th percentile.
- It cuts off the lowest 25 % of the data. The third quartile, denoted by Q3, is the 75th percentile - it cuts off the lowest 75 % (or highest 25 %) of the data.
- The second quartile is the 50th percentile. As the median, it gives the center of the data distribution.
- The distance between the first and third quartiles is a simple measure of spread that gives the range covered by the middle half of the data. This distance is called the **interquartile range (IQR)** and is defined as IQR = Q3 – Q1.

Outliers

- In statistics, an outlier is a data point that differs significantly from other observations or patterns. (Refer Fig. 2.7.3)
- An outlier is an observation that lies an abnormal distance from other values in a random sample from a population.

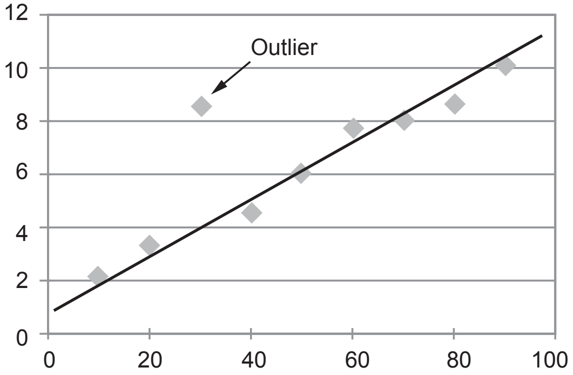


Fig 2.7.3 An outlier in a data set

- In a sense, this definition leaves it up to the analyst (or a consensus process) to decide what will be considered abnormal.
- Before abnormal observations can be singled out, it is necessary to characterize normal observations.
- An outlier may be due to variability in the measurement or it may indicate experimental error; the latter are sometimes excluded from the data set.
- An outlier can cause serious problems in statistical analyses.
- A common rule of thumb for identifying suspected outliers is to single out values falling at least 1.5 × IQR above the third quartile or below the first quartile.

Five-Number Summary

- Because Q1, the median, and Q3 together contain no information about the endpoints (e.g., tails) of the data.
- A fuller summary of the shape of a distribution can be obtained by providing the lowest and highest data values as well. This is known as the **five-number summary**.
- The five-number summary of a distribution consists of
  - The median (Q2),
  - The quartiles Q1 and Q3, and the smallest and largest individual observations,
  - The five numbers are written in the order of ***Minimum, Q1, Median, Q3, Maximum***.

Boxplots

- Boxplots are a popular way of visualizing a distribution.
- A boxplot incorporates the five-number summary as follows :
  - Typically, the ends of the box are at the quartiles so that the box length is the interquartile range.
  - The median is marked by a line within the box.
  - Two lines (called whiskers) outside the box extends to the smallest (Minimum) and largest (Maximum) observations. (Refer Fig. 2.7.4 on next page)
- Boxplots can be computed in On logn time. Approximate boxplots can be computed in linear or sublinear time depending on the quality guarantee required.

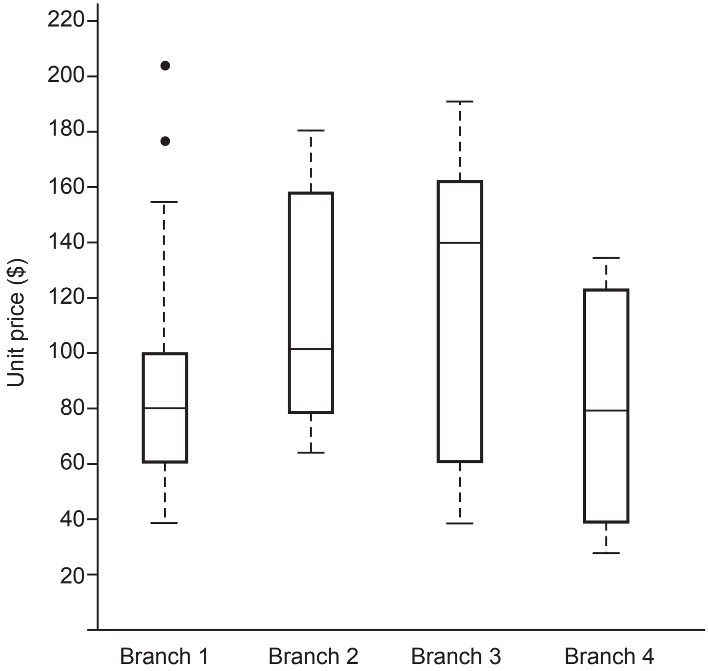


Fig 2.7.4 Boxplot for the unit price data for items sold at four branches of a shop

Variance and Standard Deviation

- Variance and standard deviation are measures of data dispersion. They indicate how spread out a data distribution is. A low standard deviation means that the data observations tend to be very close to the mean, while a high standard deviation indicates that the data are spread out over a large range of values.
  - The variance of N observations, $x_1, x_2, \ldots, x_N$, for a numeric attribute x is

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

where $\bar{x}$ is the mean value of the observations

- The standard deviation, σ of the observations is the square root of the variance, σ^2.
- For the dataset in Table 2.7.1, the variance is calculated as

$$\sigma^2 = \frac{1}{12}\left(30^2 + 36^2 + 47^2 + \ldots + 110^2\right)$$

$$\approx 397.17$$

$$\sigma \approx \sqrt{379.17} \approx 19.47$$

- The basic properties of the standard deviation, σ, as a measure of spread are as follows :

- ○ σ measures spread about the mean and should be considered only when the mean is chosen as the measure of center.
- ○ σ = 0 only when there is no spread, that is, when all observations have the same value. Otherwise, σ > 0.

- The standard deviation is a good indicator of the spread of a data set.
- The computation of the variance and standard deviation is scalable in large databases.

Graphic Displays of Basic Statistical Descriptions of Data

- Graphs are helpful for the visual inspection of data, which is useful for data preprocessing.
- The first three of these show univariate distributions (i.e., data for one attribute), while scatter plots show bivariate distributions (i.e., involving two attributes).

Quantile Plots

- A quantile plot is a simple and effective way to have a first look at a univariate data distribution.
- First, it displays all of the data for the given attribute (allowing the user to assess both the overall behavior and unusual occurrences).
- Second, it plots quantile information
  - ○ Let x_i , for i = 1 to N, be the data sorted in increasing order so that x_1 is the smallest observation.
  - ○ x_N is the largest for some ordinal or numeric attribute X.
  - ○ Each observation, x_i, is paired with a percentage, f_i, which indicates that approximately f_i × 100% of the data are below the value, x_i.
- Note that the 0.25 percentile corresponds to quartile Q1, the 0.50 percentile is the median, and the 0.75 percentile is Q3.

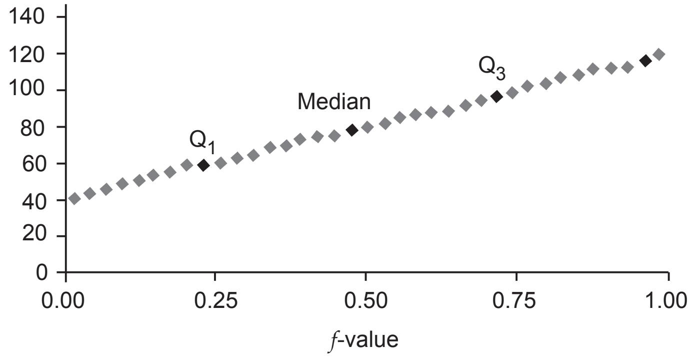


Fig 2.7.5 - A quantile plot for the unit price data

- Let

$$f_i = \frac{i - 0.5}{N}$$

- These numbers increase in equal steps of 1/N, ranging from 1 2N (which is slightly above 0) to 1 - 1 2_N (which is slightly below 1).
- On a quantile plot, x_i is graphed against f_i. This allows us to compare different distributions based on their quantiles.

Quantile - quantile Plots

- A quantile–quantile plot, or q-q plot, graphs the quantiles of one univariate distribution against the corresponding quantiles of another.
- It is a powerful visualization tool in that it allows the user to view whether there is a shift in going from one distribution to another.
- Suppose that we have two sets of observations for the attribute or variable unit price, taken from two different branch locations (Refer Table 2.2.2)

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

Table 2.7.2 - Unit price and count of sale at a shop

- Let $x_1, ..., x_N$ be the data from the first branch, and $y_1,..., y_M$ be the data from the second, where each data set is sorted in increasing order.
- If M = N (i.e., the number of points in each set is the same), then we simply plot y_i against x_i , where y_i and x_i are both (i - 0.5)/N quantiles of their respective data sets.

- If M < N (i.e., the second branch has fewer observations than the first), there can be only M points on the q-q plot. Here, y_i is the (i - 0.5)/M quantile of the y data, which is plotted against the (i - 0.5)/M quantile of the x data. This computation typically involves interpolation.

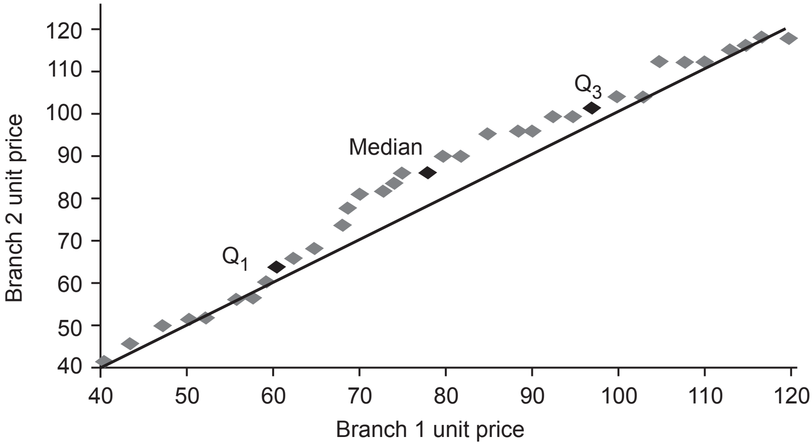


Fig 2.7.6 A q-q plot for unit price data from two branches of a shop

Histograms

- Histograms (or frequency histograms) are at least a century old and are widely used. "Histos" means pole or mast, and "gram" means chart, so a histogram is a chart of poles.
- Plotting histograms is a graphical method for summarizing the distribution of a given attribute, X.
- If X is nominal, such as automobile model or item type, then a pole or vertical bar is drawn for each known value of X.

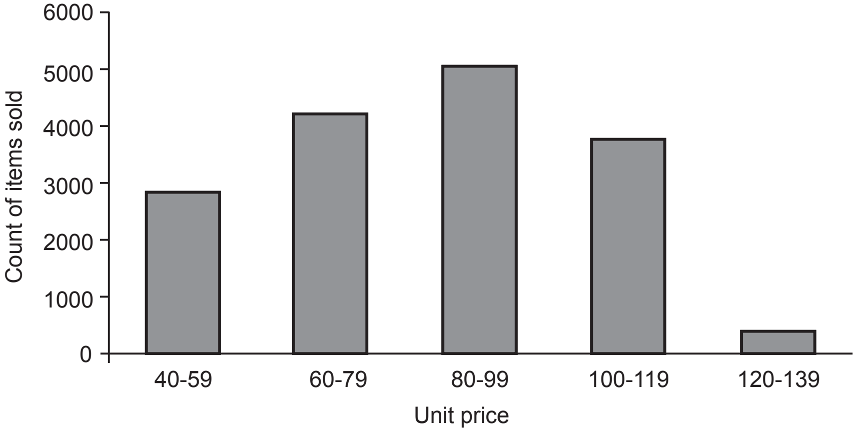


Fig 2.7.7 A histogram for the Table 2.7.2 data set.

- The height of the bar indicates the frequency (i.e., count) of that X value. The resulting graph is more commonly known as a bar chart.
- If X is numeric, the term histogram is preferred.
- The range of values for X is partitioned into disjoint consecutive subranges.
- The subranges, referred to as buckets or bins, are disjoint subsets of the data distribution for X.
- The range of a bucket is known as the width.

Scatter Plots

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

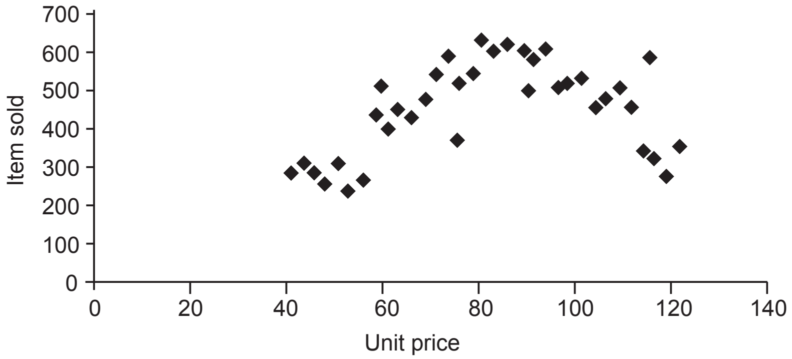


Fig 2.7.8 A scatter plot for the Table 2.7.2 data set

- To construct a scatter plot, each pair of values is treated as a pair of coordinates in an algebraic sense and plotted as points in the plane. (Refer Fig. 2.7.8)
- The scatter plot is a useful method for providing a first look at bivariate data to see clusters of points and outliers, or to explore the possibility of correlation relationships.
- Two attributes, X and Y, are correlated if one attribute implies the other. Correlations can be positive, negative, or null (uncorrelated).
- Fig. 2.7.9 shows examples of positive and negative correlations between two attributes.
  - If the plotted points pattern slopes from lower left to upper right, this means that the values of X increase as the values of Y increase, suggesting a positive correlation Fig 2.7.9 (a).

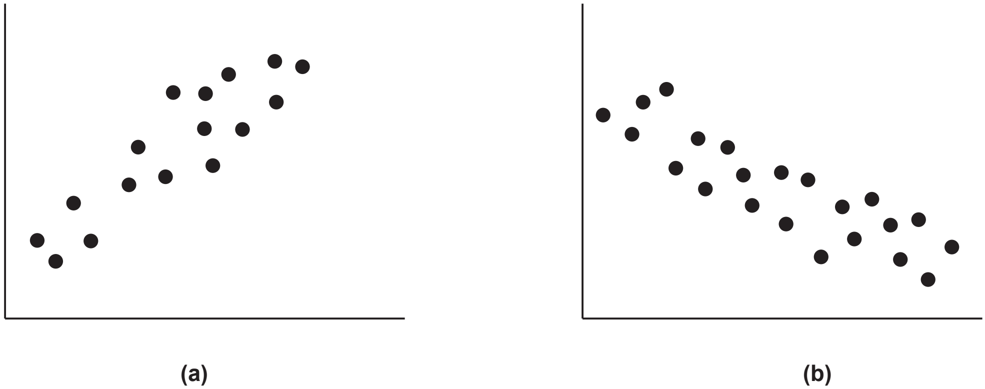


Fig 2.7.9 Scatter plots to find (a) positive or (b) negative correlations between attributes

  - ○ If the pattern of plotted points slopes from upper left to lower right, the values of X increase as the values of Y decrease, suggesting a negative correlation Fig. 2.7.9 (b).

- A line of best fit can be drawn to study the correlation between the variables.
- Refer Fig. 2.7.10 shows three cases for which there is no correlation relationship between the two attributes in each of the given data sets.

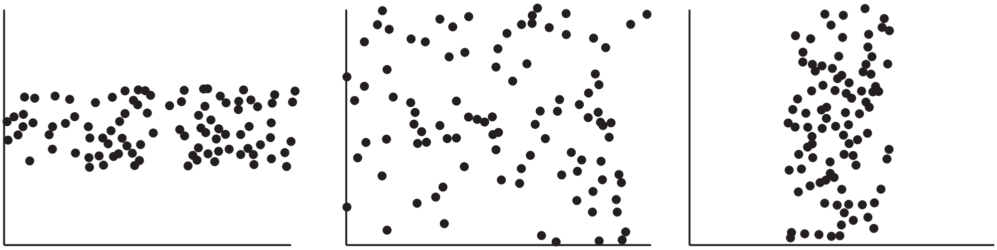


Fig. 2.7.10 Scatter plots to find (a) positive or (b) negative correlations between attributes

Conclusion

- Basic data descriptions (e.g., measures of central tendency and measures of dispersion) and graphic statistical displays (e.g., quantile plots, histograms, and scatter plots) provide valuable insight into the overall behavior of your data. By identification of noise and outliers, these techniques are useful for data cleaning.

2.8 Data Preprocessing

- Due to humongous amount of data generated by real time applications, it is a challenging task to maintain the quality of this data.
- Typically such databases are generally contain noisy, missing and inconsistant data.
- So to improve the quality of this data for getting the correct mining results, processing of the data is required.
- Different preprossing techniques which can be applied are :
  - Data cleaning : To remove noise and correction of inconsistency in data.
  - Data integration : To merge data from different sources to a warehouse
  - Data reduction : To reduce the data size by techniques like aggregation clustering etc.
  - Data transformation : To scale the data by normalization
- The data is said to be a quality data if it is : accurate, complete, consistant, follow the timeline, believable and interpretable.

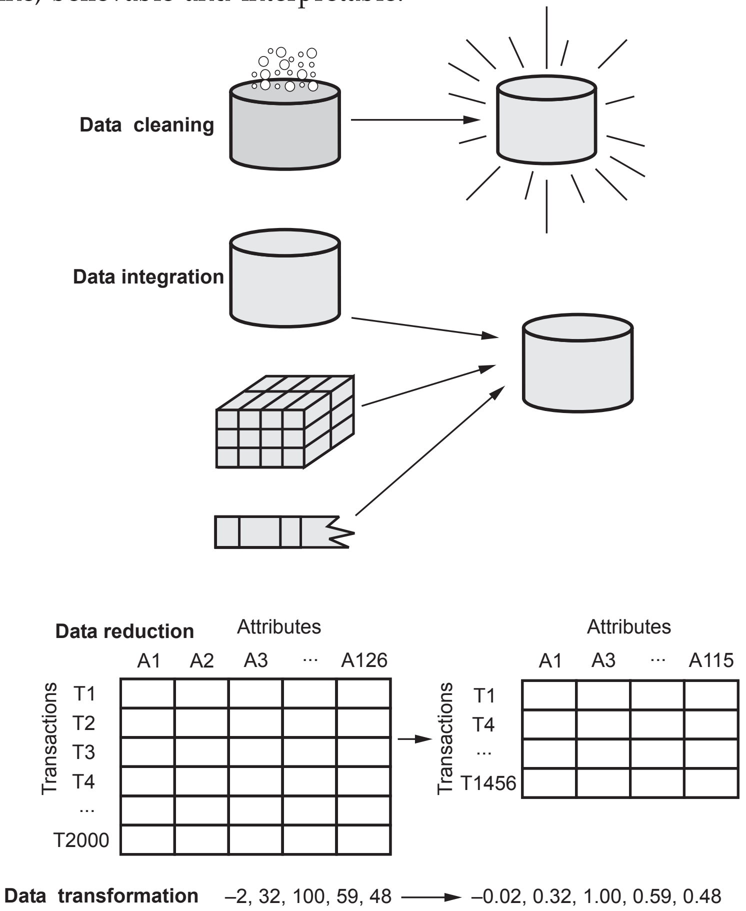


Fig. 2.8.1 Forms of data preprocessing

2.9 Data Cleaning

- As discussed earlier, to handle incomplete, noisy and inconsistent data, first step is to clean the data.
- Data cleaning includes the following tasks :
  - Filling in missing values
  - Identify outliers and smooth out noise
  - Correct inconsistencies in the data

2.9.1 Missing Values

- Consider the example of sales and customer data, if the attributes like customer income are missing from the tuples, following methods can be adapted to fill in such missing values :
  - **Ignore the tuple :**
    - This technique can be used when the class label is missing
    - If the tuple contains several attributes with missing values then only this technique is useful.
    - If the percentage of missing values per attribute varies significantly this method gives poor results.
  - **Fill in the missing value manually :**
    - In general, this technique is time consuming and less feasible for a large data set with many missing values.
  - **Use a global constant to fill in the missing value :**
    - In this approach missing attribute values are replaced by the same constant such as a label like "Unknown" or $-\infty$.
    - The drawback of this method is, if the missing values are replaced by, say, "Unknown," it may be considered as an interesting pattern by the program.
  - **Use a measure of central tendency for the attribute (e.g., the mean or median) to fill in the missing values :**
    - Central tendency takes into consideration middle value of a data distribution.
    - Central tendency of normal (symmetric) data distributions is given by mean and for skewed data distribution it will be given by median.
  - **Use the attribute mean or median for all samples belonging to the same class as the given tuple :**
    - To understand this approach consider the example of customer database. If we want to classify the customers based on credit risk, the missing value can be replaced by mean income value of the same risk category.

- For skewed data distribution median value can be used.

○ **Use the most probable value to fill in the missing value :**

- Using other attribute values missing values can be predicted.
- This can be done by using the techniques like regression, inference-based tools using a Bayesian formalism or decision tree induction.
- For ex : Missing values for income of the customers can be predicted by constructing a decision tree using the other customer attributes in the data set.

2.9.2 Noisy Data

- Noise is a random error or variance in a measured variable.
- Following techniques can be adapted for smoothing of the data :

1) Binning :

- It involves smoothing and sorted data values by considering the neighboring values around it.
- Sorted values are divided in bins or buckets. Consider the example in Fig. 2.9.1.

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

Partition into (equal-frequency) bins :

Bin 1 : 4, 8, 15
Bin 2 : 21, 21, 24
Bin 3 : 25, 28, 34

Smoothing by bin means :

Bin 1 : 9, 9, 9
Bin 2 : 22, 22, 22
Bin 3 : 29, 29, 29

Smoothing by bin boundaries :

Bin 1 : 4, 4, 15
Bin 2 : 21, 21, 24
Bin 3 : 25, 25, 34

Fig. 2.9.1 Binning methods for data smoothing

- As shown in the example after sorting price data, it is partitioned into equal frequency bins of size 3.
- There are three ways of binning :

a) Smoothing by bin means :

○ Each value in a bin is replaced by mean value of a bin.

○ For ex : As shown in 1 mean of values 4, 8 and 15 is 9.

- So bin 1 values are replaced by 9.

b) Smoothing by bin medians :

- Each value in a bin is replaced by bin median value.

c) Smoothing by bin boundaries :

- Minimum and maximum values in a bin are called as bin boundaries.
- Each value in a bin is replaced closet boundary value.

2) Regression :

- It is a technique data values are standardized to a function.
- In linear regression best line is found which will fit two attributes.
- Based on this one attribute can be used to predict other.
- In multiple liner regression, linear regression technique is extended for more than two attributes.

3) Outlier analysis :

- Outlier is an data point which differs significantly from other observations.
- One way of detecting outliners is through clustering.
- The organization of values in groups is called as cluster.

2.9.3 Process of Data Cleaning

- Discrepancy detection in the first step in data cleaning.
- Typically discrepancies are arised from inconsistent data representation and inconsistence use of codes.
- Various tools are used in dicrepency detection are
  - Data scrubbing tools → to detect errors and make corrections.
  - Data auditing tools → to discover rules and relatioships.

2.10 Data Integration

- Merging of data from multiple data stored is called as data integration.
- Data integration in useful in improving accuracy and speed of mining process.
- Through data integration, redundancies and inconsistencies in the data can be avoided.
- The challenge faced by data integration is matching of schemas and objects from different sources which can be resolved by entity idetification problem.
- Some more issues are :
  - Redundancy and correlation analysis

- ○ Tuple duplication
- ○ Data value conflict detection and resolution.

1. Entity Identification Problem :

- In data warehouse, the data from various sources like multiple database, data cubes or flat files combined and stored.
- The challenge faced in this process is schema intergration and object matching.
- For ex. : The same attribute can be named as customer id - in one database and cust-number in another.
- By using metadata errors can be avioded in schema integration.
- Also while intgeration the care must be taken related to structure of data while matching the attributes.
- For example : In one database, word discount may represents each individual line items in order, while in other it represents the order.

Redundancy and Correlation Analysis

- An attribute is said to be redundant is it is derived from other attribute.
- Redundancy can also be caused by inconsistancy in attribute of dimension.
- One way of redundancy detection is correlation analysis to understand how one attribute implies to other.
- χ^2(Chi-square) test is used for this fan normal data.

χ^2 correlation test of Nominal data

- It is used to find relationship between two attributes A and B.
- Consider A with C distinct values $a_1, a_2, ... a_c$.
- B with r distinct values $b_1, b_2, ... b_r$.
- With A as column and B as rows. we can write a contingency table.
- Consider (A_i, B_j) is a joint event.

  where $(A = a_i, \ B = b_j)$.
- Each joint event (A_i, B_j) has one slot in the table.

  Let o_{ij} is observed frequency of (A_i, B_j)

  e_{ij} is expected frequency of (A_i, B_j)

$$\text{where } e_{ij} = \frac{\text{Count } (A = a_i) \times \text{Count } (B = b_j)}{n} \qquad \dots (2.10.1)$$

n = Number of data tables

From this chi values, χ^2 is calculated as

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

- To understand this consider the contingency table below.

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

- The table shown survey of 1500 people for the reading preference as fiction or non fiction.
- The numbers in bracket are the expected frequencies which are calculated from equation 2.10.1 as,

$$e_{11} = \frac{\text{Count (male)} \times \text{Count (fiction)}}{n}$$

$$= \frac{300 \times 450}{1500} = 90 \text{ and so on}$$

- Chi value can be calculated as

$$\chi^2 = \frac{(250-90)^2}{90} + \frac{(50-210)^2}{210} + \frac{(210-360)^2}{360} + \frac{(1000-840)^2}{840}$$

$$= 284.44 + 121.90 + 71.11 + 30.48 = 507.93$$

Correlation coefficient for numeric data :

- Correlation coefficient is also called as Pearson's product moment coefficient.
- It can be computed as

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

$$= \frac{\sum_{i=1}^{n} (a_i - b_i) - n\overline{A}\overline{B})}{n\sigma_A \sigma_B}$$

Where $n \neq$ Number of tuples

a_i, b_i = Values of A and B in tuple I

$\overline{A}, \overline{B}$ = Mean values of A and B

σ_A, σ_B = Standard derivation of A and B

3. Tuple Duplication :

- It may happen that for a unique data entry two or more identical tuples may exists.
- Data redundancy can also be caused by denormalyed tables.
- Duplication issue arises because of inaccurate data entry or due to incorrect updation for ex. same customer name with two different addreses.

4. Data value conflict detection and resolution :

- Due to differences in representations or encoding styles, etc. the attribute values may differ.
- Some examples of these include differences in currencies or in grading schemes of different schools.
- The conflict may also arise due to different abstraction levels of different systems.

For example : Total sales in one database may refer to sale value of one department where as it may indicate total sale value of company in other database.

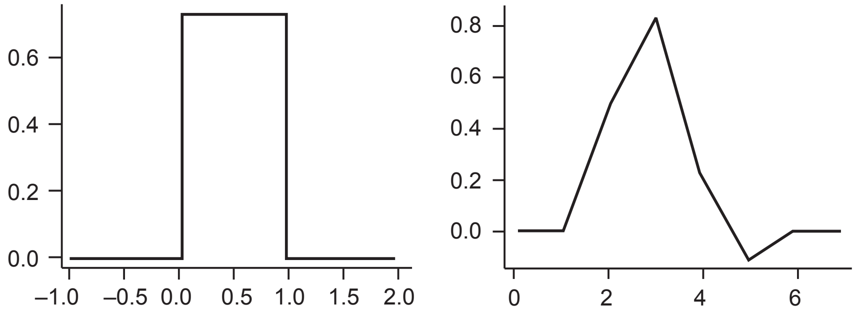
2.11 Data Reduction

- If the data set is very large, it takes a long amount of time to analyse and mine that data.
- If the data size can be reduced the mining process will become more effective without affecting the end result.
- The following data reduction strategies will be discussed in this section :
  - Dimensionality reduction
  - Numerosity reduction
  - Data compression
- Dimensionality reduction : Random variables or attributes under consideration are reduced by this method.
- Dimensionality reduction methods are :
  - Wavelet transforms
  - Principal components analysis
  - Attribute subset selection
- Numerosity reduction techniques are :
  - Parametric methods
    - Regression and log-linear models
  - Nonparametric methods
    - Histograms
    - Clustering

    - Sampling
    - Data
    - Cube aggregation
  - Data compression
    - Lossless
    - Lossy

2.11.1 Wavelet Transforms

- The discrete wavelet transform (DWT) is a linear signal processing technique through which the data vector X is transformed to new vector X' of same length of wavelet coefficients.
- Each tuple is considered as an n-dimensional data vector, where $X = (x_1 , x_2 , . . . , x_n)$ with n database attributes having n attributes.
- The advantage is that the wavelet transformed data can be truncated.
- Only a small fraction of the strongest of the wavelet coefficients is retained, keeping value of all other coefficients 0.
- The resultant data representation is sparse and can be computed with high speed in wavelet space.
- DWT provides better lossy compression as compared to the discrete Fourier transform (DFT).
- It requires less space than DFT.
- As shown in the Fig. 2.11.1 some of the popular transforms are Haar-2, Daubechies-4, and Daubechies-6.
- A hierarchical pyramid algorithm is used to apply discrete wavelet transform as follows :
  - Input data vector must be an integer power of 2. This can be done by padding the data vector with zeros.
  - Two functions are applied on each transform, first for applying data smoothing and second for finding weighted difference.
  - A smoothed or low-frequency version of the input data and its high frequency content is represented after applying the above two functions to data points in X.
  - This process is continued till resulting data sets obtained are of length 2.
  - Thus we obtain designated wavelet coefficients from selected values from the data sets.



(a) Haar - 2 **(b) Daubechies - 4**

Fig. 2.11.1 Examples of wavelet families. The number next to a wavelet name is the number of vanishing moments of the wavelet. This is set of mathematical relationships that the coefficients must satisfy and is related to the number of coefiicients

2.11.2 Principal Components Analysis

- Principal components analysis (PCA) is also called as Karhunen - Loeve or K-L method.
- It reduces the data represented by types or data vectors.
- K n-dimensional orthogonal vectors are searched to represent the data where $K \le n$.
- It creates small set of variables by combining extract of attributes.
- It uses following process :
  1. Normalization of input data to ensure that large well as small domain attributes are considered.
  2. K orthonormal vectors are computed, known as principal components. The direction of each point is perpendicular to others.
  3. Sorting of principal components is done by decreasing strength. As shown in Fig. 2.11.2 two principal components Y_1 and Y_2 are mapped to axes X_1 and X_2.
  4. Due to decreasing order of components data size can be reduced by elimination of weaker components.

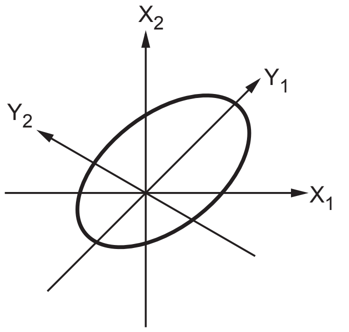


Fig. 2.11.2 Principal component analysis Y_1 and Y_2 are the first two principal components for the given data

2.11.3 Attribute Subset Selection

- To understand need of this, consider the example of purchase of music CD.
- If we want to classify the customers if they will purchase the CD or not, the attributes like telephone no. of a customer are irrelevent.
- Instead we can only consider the attributes like age or music-taste.
- In such a case using attribute subset selection technique the data set size is reduced by removing irrelevent or redundant attributes.
- The aim is to match the probability distribution of minimum set of attributes to a original distribution.
- To achieve this, heuristic methods to explore reduced search space are used.
- These methods adapt greedy approach and looks for the best choice while reaching through attribute space.
- The best attributes are chosen which are independent of one another based on their statistical significance as shown in Fig. 2.11.3.
- As shown in Fig. 2.11.3 there are four techniques.

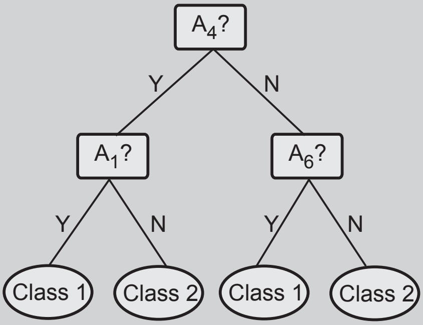
Forward selection	Backward elimination	Decision tree induction
Initial attribute set : $\{A_1, A_2, A_3, A_4, A_5, A_6\}$ Initial reduced set : { } $\Rightarrow$ $\{A_1\}$ $\Rightarrow$ $\{A_1, A_4\}$ $\Rightarrow$ Reduced attribute set : $\{A_1, A_4, A_6\}$	Initial attribute set : $\{A_1, A_2, A_3, A_4, A_5, A_6\}$ $\Rightarrow$ $\{A_1, A_3, A_4, A_5, A_6\}$ $\Rightarrow$ $\{A_1, A_4, A_5, A_6\}$ $\Rightarrow$ Reduced attribute set : $\{A_1, A_4, A_6\}$	Initial attribute set : $\{A_1, A_2, A_3, A_4, A_5, A_6\}$  A4? Y N A1? A6? Y N Y N Class 1 Class 2 Class 1 Class 2 $\Rightarrow$ Reduced attribute set : $\{A_1, A_4, A_6\}$

Fig. 2.11.3 Greedy (heuristic) methods for attribute subset selection

1. Stepwise forward selection :

- At start empty set of attributes is considered as reduced set.
- The best of original attributes are added is this set.
- This process is iterated.

2. Stepwise backward elimination :

- With availability of full set of attributes at each step, worst attributes are removed.

3. Combination of forward selection and backward elimination.

- In this at each step best attribute is selected as per stepward forward selection and worst is removed as per stepwise backward elimination.
- Thus it combines both the methods.

4. Decision tree induction

- Flowchart like structure is created where each internal nonleaf node denotes test on an attribute and each branch denotes outcome of the test.

2.11.4 Regression and Log-Linear models : Parametric Data Reduction.

- This technique is used for approximation of the data.
- In linear regression the equation.

  $y = wx + b$ is used to fit the data on a straight line.
- If we apply the concept of data mining x and y are numeric database attributes and w and b are called as regression coefficients.
- Least squares method is used to minimize the error actual time seprating the data and estimate of the line.
- Multiple linear regression is extension of linear regression which models variable y as linear function of two or more predictor variables.
- In log linear model, for set of tuples in n dimensions each tuple is considered as point in n-dimensional space.
- Probability of each point in multidimensional space is computed for a set of discretized attributes.
- By this higher dimensional data space is constructed from lower dimensional space facilitating dimensionality reductions.

2.11.5 Histograms

- Binning technique is used to approximate data distributions.
- Attribute A is divided into disjoint subacts called as buckets or bins by histogram.

- Singleton bucket is the one which contains single attribute value/frequency pair.
- For example : Consider list of commonly sold items in a store.

  The sorted list is as follows :

1, 1, 5, 5, 5, 5, 5, 8, 8, 10, 10, 10, 10, 12, 14, 14, 14, 15, 15, 15, 15, 15, 15, 15, 15, 18, 18, 18, 18, 18, 18, 18, 18, 18, 18, 18, 20, 20, 20, 20, 20, 20, 20, 21, 21, 21, 21, 25, 25, 25, 25, 25, 28, 28, 30, 30, 30.

The histogram can be plotted as shown in Fig. 2.11.4 and Fig. 2.11.5.

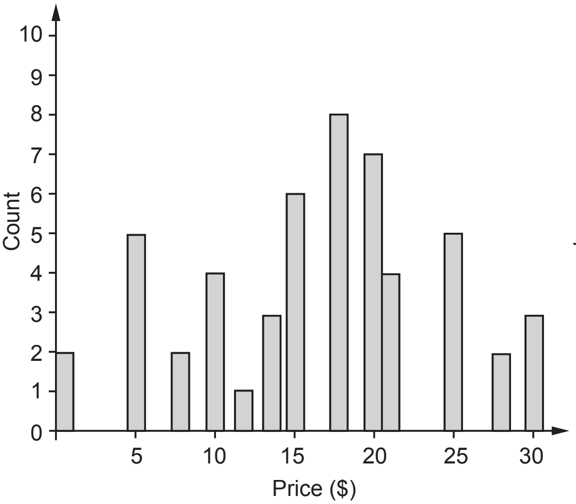


Fig. 2.11.4 Histogram for price using singletone buckets - each bucket represents one price value/frequency pair

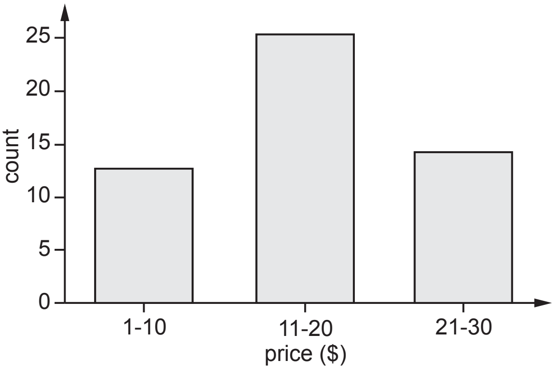


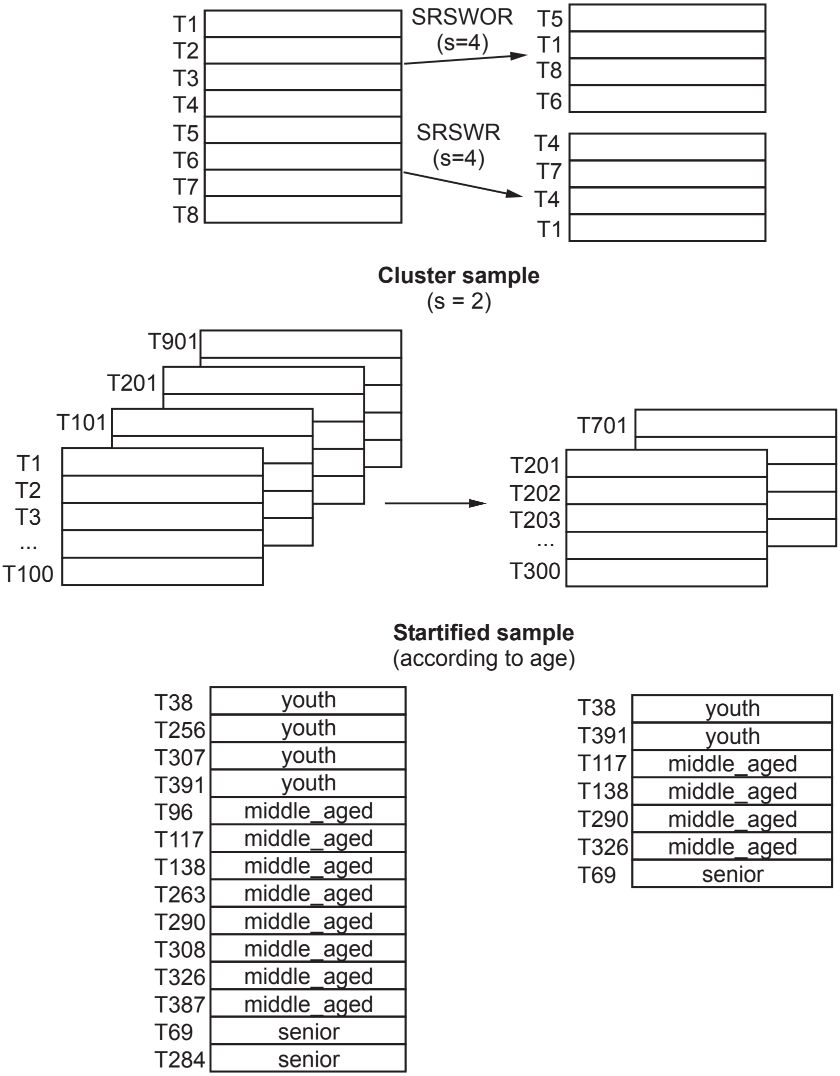
Fig. 2.11.5 an equal - width histogram for price where values are aggregated so that each bucket has a uniform width of $ 10

2.11.6 Clustering

- Data tuples are considered as objects.
- These objects are partitioned into groups known as clusters.
- Objects in one cluster are similar to each other and different from other objects in different clusters.
- The quality of cluster is determined by its diameter.
- Average distance of each cluster object from cluster centerior is called as centroid distance.
- The efficiency of this method is dependent on data's nature.

2.11.7 Sampling

- In this data reduction technique, large data set is represented by much smaller random data sample.
- Consider large data set D of N tuples
- As shown in the Fig. 2.11.6 for sampling following ways can be adapted.



T38	youth
T256	youth
T307	youth
T391	youth
T96	middle_aged
T117	middle_aged
T138	middle_aged
T263	middle_aged
T290	middle_aged
T308	middle_aged
T326	middle_aged
T387	middle_aged
T69	senior
T284	senior

T38	youth
T391	youth
T117	middle_aged
T138	middle_aged
T290	middle_aged
T326	middle_aged
T69	senior

Fig. 2.11.6 Sampling can be used for data reduction

1. Simple random sample without replacement (SRSWOR) of size S

- Sample S is extracted from N tuples of D where S < N and probability of sample generation is $\frac{1}{N}$.

2. Simple random sample with replacement (SRSWR) of size S

- Each drawn tuple is recorded and kept back in D so - that it can be drawn again.

3. Cluster sample

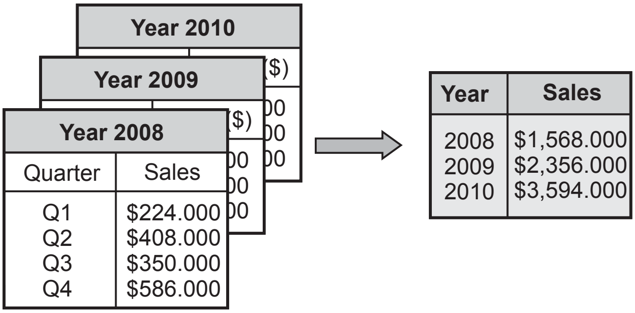
- Tuples in D are grouped in M disjoint clusters.
- SRS of S clusters can be formed where S < N.
- Tuples are generally retrived as pages and each bage is considered a cluster.
- These can be further reduced by applying SRSWOR resulting is cluster sample.

4. Stratified sample

- D is divided in small parts called strata.
- SRS in formed at each stratum.
- For example : In a customer data stratum is created for each customer age group.

2.11.8 Data Cube Aggregation

- If the analysis needs aggregation of data. For ex : Calculation of sales per year based on the sales per quarter, the output data set becomes smaller.
- For this purpose data cubes can be used.
- As shown in Fig. 2.11.7 and Fig. 2.11.8 multidimensional aggregated information is stored in the form of data cubes.


Year 2010
($)
00
00
00
Year 2009
($)
00
00
00

Year 2008	
Quarter	Sales
Q1	$224.000
Q2	$408.000
Q3	$350.000
Q4	$586.000

Year	Sales
2008	$1,568.000
2009	$2,356.000
2010	$3,594.000

Fig. 2.11.7 Sales data for a given branch of ALLelectronics for the years 2008 through 2010. On the left, the sales are shown per quarter. On the right, the data are aggregated to provide the annual sales

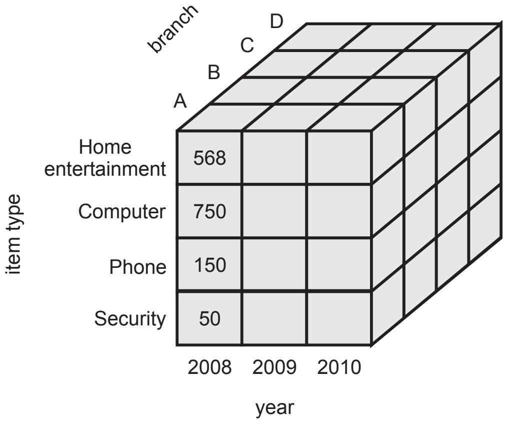


Fig. 2.11.8 A data cube for sales at all electronics

- Each cell contains aggregate data value per data point in multidimensional space.
- Lower abstraction level cube in called as base cuboid and highest level cuboid is known as apex cuboid.

2.12 Data Transformation and Data Discretization

- To increase the efficiency of data mining process transformation of consolidation of the data is performed.
- Following are the strategies for data transformation :

1. Smoothing : Removal of noise from data by the techniques like binning, regression and clustering as discussed is earlier sections.

2. Attribute construction : Construction and addition of new attributes for efficient mining process.

3. Aggregation : Helpful in data analysis as data is summarized. For ex : daily sales data can be aggregated to calculate monthly or yearly sales of a company.

4. Normalization : Scale down the data to a smaller range. For ex : – 1.0 to 1.0

5. Discretization : Generation of concept hierarchy.

- Raw tables are replaced by interval tables.

6. Concept hierarchy generation for nominal data :

- Generalization of the attributes is done.
- For ex : Attribute street in generalized to higher level concepts city or country.

The first three strategies were discussed in the previous section.

In this section normalization, discretization and concept hierarchy generation techniques are discussed.

2.12.1 Data Transformation by Normalization

- Data normalization or standardization is required for avoiding the dependency of data on the choice of measurement units.
- This is required because if we change the measurement unit, for ex, from meters to inches for height attributes from kilogram to pounds for weight attribute, the result will be different.
- Generally to normalize the data, it is transformed to a lower range limits like [– 1, 1] or [0.0, 1.0]
- All attributes are given equal weights.
- Normalization proves useful in algorithms in neural networks, nearest-neighbor classification and clustering.
- Among many methods of data normalization, the following three will be discussed in this section. All the methods consider A as numeric attribute with n.

  1. Min-max normalization observe values V_1, V_2, ...,V_n
  2. Z-score normalization
  3. Normalization by decimal scaling

1. Min-max normalization :

- It works on original data and transform it linearly.
- Consider min A and max A are minimum and maximum values of attributes A.
- Mix max normalization can be applied as :

$$V_i' = \frac{V_i - \min_A}{\max_A - \min_A}(\text{new\_}\max_A - \text{new\_}\min_A) + \text{new\_}\min_A$$

  where V_i' is mapping of value V_i of A in the range [$\text{new\_}\min_A$, $\text{new\_}\max_A$]

- For example : Consider attribute income

  Consider min income = ₹ 12000

  max income = ₹ 98000

If we need to normalize income in the range of [0.0, 1,0], then if value = ₹ 73,600 is transformed to

$$= \frac{73600 - 12000}{98000 - 12000}(1.0 - 0) + 0 = 0.716$$

- The drawback of this technique is, if input falls outside original data range of A, out-of-bounds error occurs.

2. Z-score normalization or zero-mean normalization :

- In this method normalization of the values of A is done using mean and standard deviation of A.

- *Z*-score normalization can be applied as :

$$V_i' = \frac{V_i' - \overline{A}}{\sigma_A}$$

where V_i' is mapping of value of V_i of A

and $$\overline{A} = \text{Mean} = \frac{1}{n}(V_1 + V_2 + \cdots + V_n)$$

$$\sigma_A = \text{Standard deviation} = \text{square root of variance of A}$$

Consider mean of income = ₹ 54,000

Standard deviation of income = ₹ 16,000

If value of income = ₹ 73,000 it can be normalized to

$$\frac{73{,}600 - 54{,}000}{16{,}000} = 1.225$$

If we replace standard deviation of equation by mean absolute deviation

$$S_A = \frac{1}{N}(|V_1 - \overline{A}| + |V_2 - \overline{A}| + \cdots + |V_n - \overline{A}|)$$

The equation becomes,

$$V_i' = \frac{V_i - \overline{A}}{S_A}$$

3. Normalization by decimal scaling :

- In this method the decimal point is moved to maximum absolute value of A.
- It can be computed as

$$V_i' = \frac{V_i}{10^j}$$

where $\max(|V_i'|) < 1$

- Consider values of A in the range – 986 to 917. Absolute value of A = 986
- For decimal scaling divide each value by 1000.
  So – 986 is normalized to – 0.986 and 917 to 0.917

Discretization by binning :

- Binning methods were discussed in section 2.9.2.
- For discretization of attribute values equal width binning is applied and each bin value is replaced by bin mean or median.
- It is unsupervised discretization technique, as it does not use class information.

Discretization by histogram analysis :

Concept hierarchy generation for nominal data

- Nominal attributes are the attributes having finite unordered number of distinct values.
- For ex : geographic_location, item_type, etc.
- The concepts hierarchy defines a sequence of mappings from a set of low-level concepts to higher-level, more general concepts.
- Concept hierarchies transform the data into multiple levels of granularity.
- For nominal data following four methods can be used.

1. Specification of a partial ordering of attributes explicitly at the schema level by users or experts :

- User expert can specify partial or total ordering of attributes at schema level.
- For ex : For the similar attribute group like street, city, province_or_state and country in relational database and data warehouse, the hierarchy can be stated as street<city<province_or_state<country.

2. Specification of a portion of a hierarchy by explicit data grouping :

- If the database size is large it is difficult to specify concept hierarchy by explicit value enumeration.
- This require manual definitions of some portions.
- For ex : For the hierarchy province and country some intermediate levels can be given manually as,

{Alberta, Saskctchewan, Manitoba} Cprairies_canada

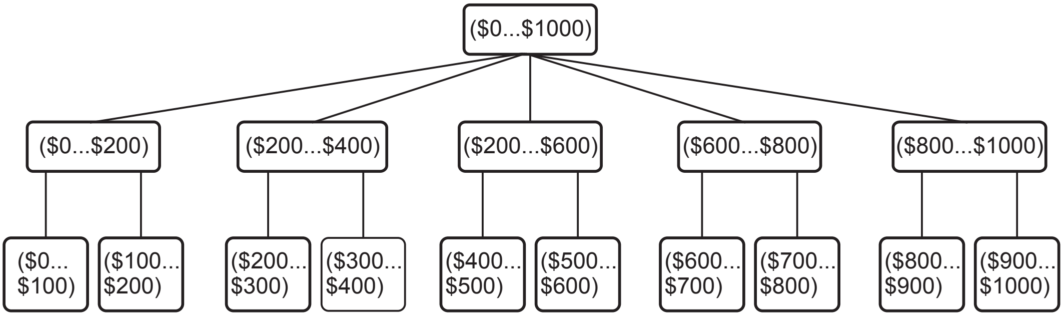


Fig. 2.12.1 A concept hierarchy for the attribute price, where an interval ($X...$Y) denotes the range from $X (exclusive) to $Y (inclusive)

3. Specification of a set of attributes but not of their partial ordering :

- To understand this method consider the example of location oriented attributes : street, country, province or state and city.
- It can be easily figured out from these attributes that the high concept level attribute like city will have less number of distinct values then lower level attributes like street.
- This fact is taken into consideration for this method.
- In concept hierarchy the attribute with more distinct values is kept on higher level and vice versa.
- This can be shown in the diagram below.

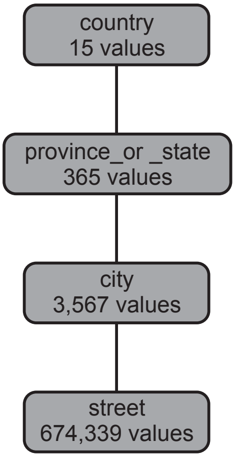


Fig. 2.12.2

4. Specification of only a partial set of attributes :

- It may happen that sometimes the hierarchies are partially defined.
- For ex : If we consider attribute location, the user may specify only street and city instead of all relevant attributes.
- To drag all the attributes for completion of hierarchy, the data semantics need to be embedded in the schema to attach the attributes with light semantic connections.

2.13 Data Visualization

- Clear and effective communication of data is facilitated by graphical representation.
- Data visualization conveys the information effectively to the user.
- Some of the applications in which data visualization proves effective are : reporting, managing, tracking progress of tasks, etc.
- Some of approaches of data visualization are :
  - Pixel oriented techniques
  - Geometric projection techniques
  - Icon based techniques
  - Hierarchical and graph based techniques

2.13.1 Pixel Oriented Visualization Techniques

- In this technique color of pixel reflects dimensions value.
- M windows are created on screen one for each dimension.
- Pixel color reflect corresponding value.
- Data values are arranged in a global order in a window.
- As shown in Fig. 2.13.1, the dimensions income, credit limit, transaction volume and age can be shown based on the shading of colors. For small value light shade can be used.

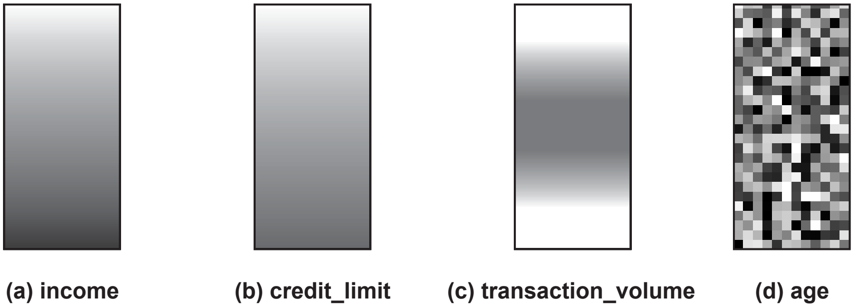


Fig. 2.13.1 Pixel-oriented visualization of four attributes by sorting all customers in income ascending order

- If the window size is wide then first pixel in a row will be far away from last pixel in previous row, though they are arranged in the same order.
- This problem can be eliminated by using space filling curve which covers entire n-dimensional unit hyper cube as shown in Fig. 2.13.2.

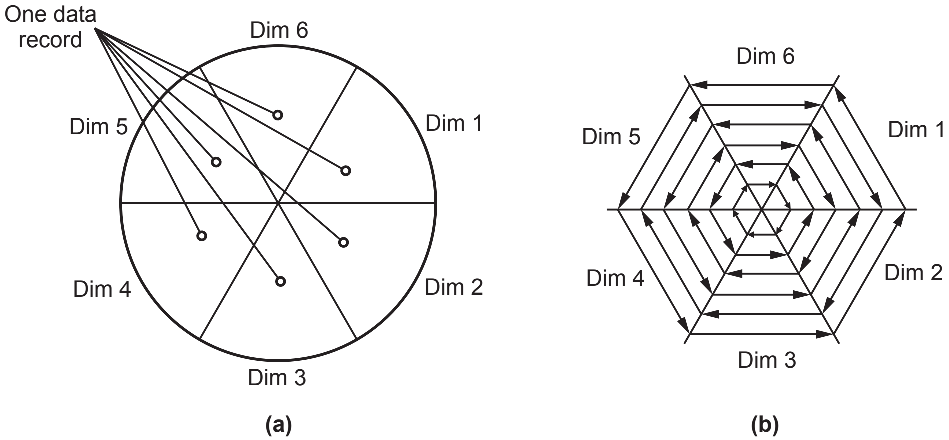


Fig. 2.13.2 The circle segment technique. (a) Representing a data record in circle segment. (b) Laying out pixels in circle segments

2.13.2 Geometric Projection Visualization Techniques

- This technique is suitable for multidimensional space.
- It addresses the challenge of visualization of high dimensions space on 2D display.
- As shown in Fig. 2.13.3 cartesian co-ordinates are used to display scatter plot of 2D data points.

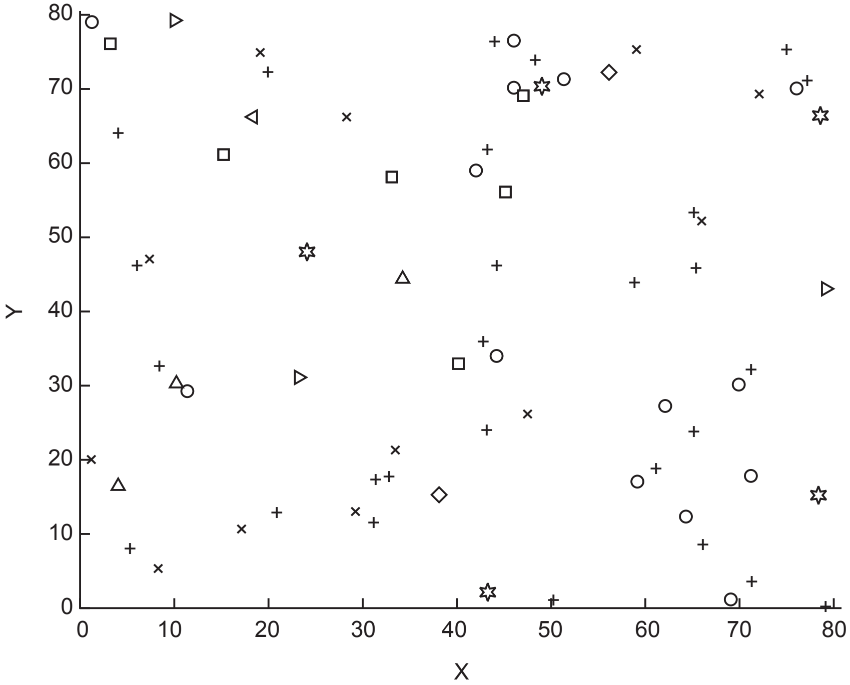


Fig. 2.13.3 Visualization of 2-D data set using a scatter plot.

- We can see in Fig. 2.13.3 X and Y are two spatial attributes. If third dimension need to be added it will be represented by different shape.
- In 3D scatter plot three axes in cartesian co-ordinate system are used, which accommodates display of 4D data points also.
- Scatter plot matrix can be used for data sets having more than 5 dimensions.
- As shown in Fig. 2.13.4 visualization of each dimension with other dimensions is plotted as n × n grid of 2D scatter plots.

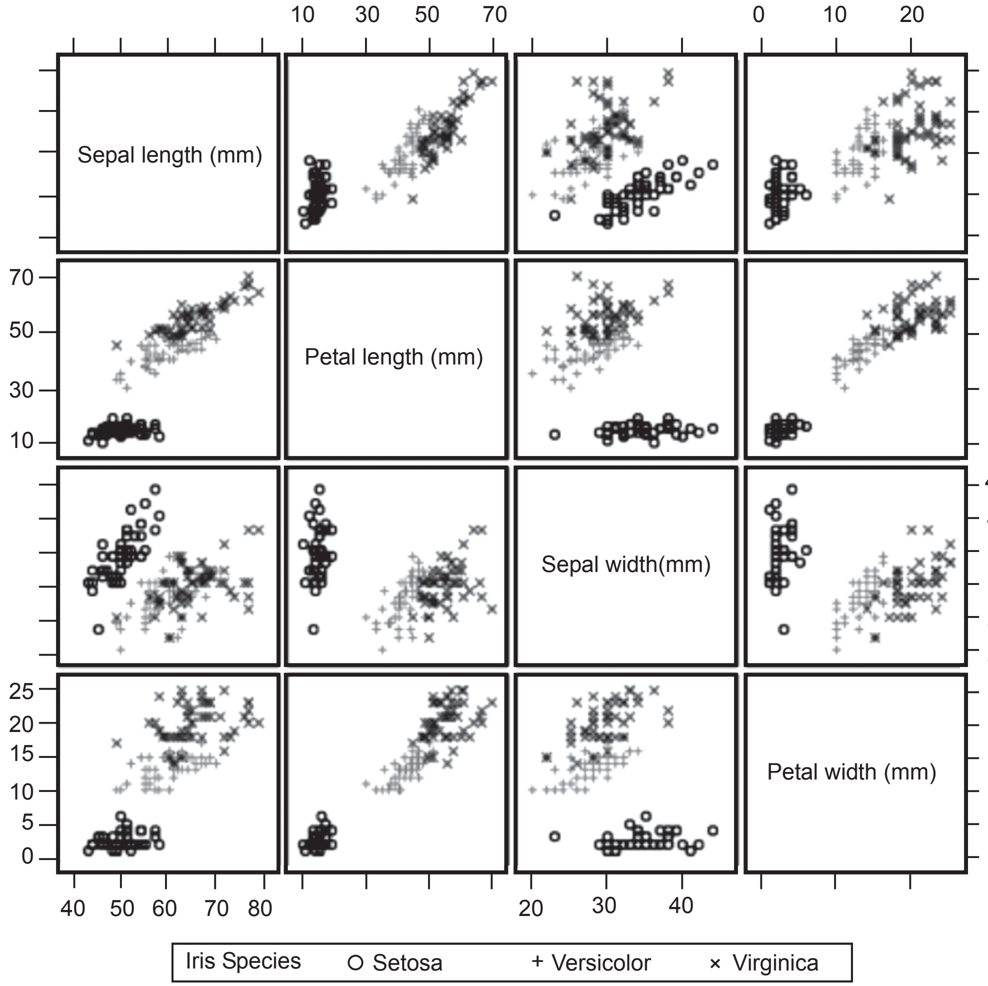


Fig. 2.13.4 Visualization of the iris data set using a scatter-plot matrix

2.13.3 Icon based Visualization Techniques

- In this approach small icons are used to represent multidimensional data values.
- Two popular techniques are :

  1. Chernoff faces 2. Stick figures

1. Chernoff faces :

- Cartoon human figure is used to represent the data upto 18 dimensions.
- The values of dimensions are represented by face components like eyes, ears, mouth, nose based on their shape, size, placement and orientation.
- The mapping of dimensions is done base on facial characteristics like, eye size, eye spacing, nose length, nose width, etc, as shown in Fig. 2.13.5.

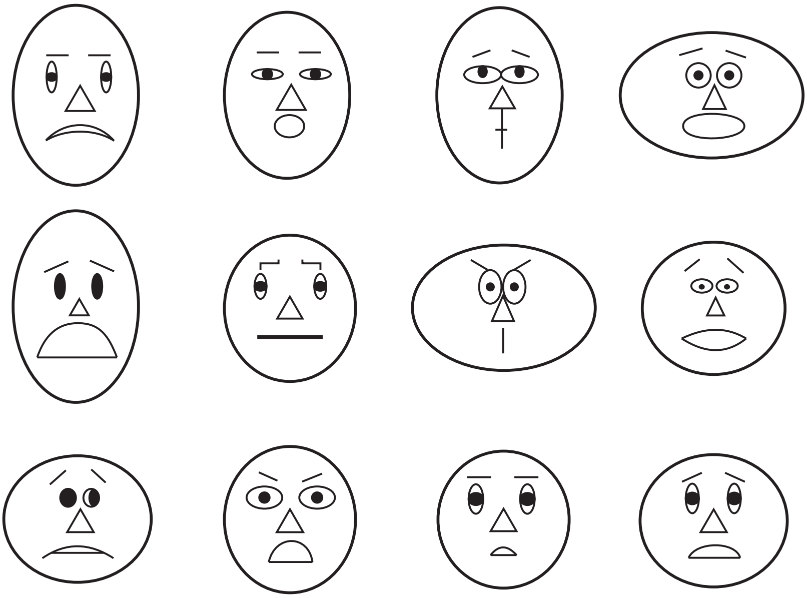


Fig. 2.13.5 Chernoff faces. Each face represents an n-dimentional data point (n ≤ 18)

- It exploits the ability of human brain to notice small differences on face, figuring out regularities and irregularities present.
- Asymmetrical chernoff faces extend this idea upto 36 dimensions.

2. Stick figure :

- Stick figure is of five pieces four limbs and a body.
- Multidimensional data is mapped to this figure.
- Two dimensional → to the display axes remaining → angle of limbs.
- The example of census data in shown Fig. 2.13.6.

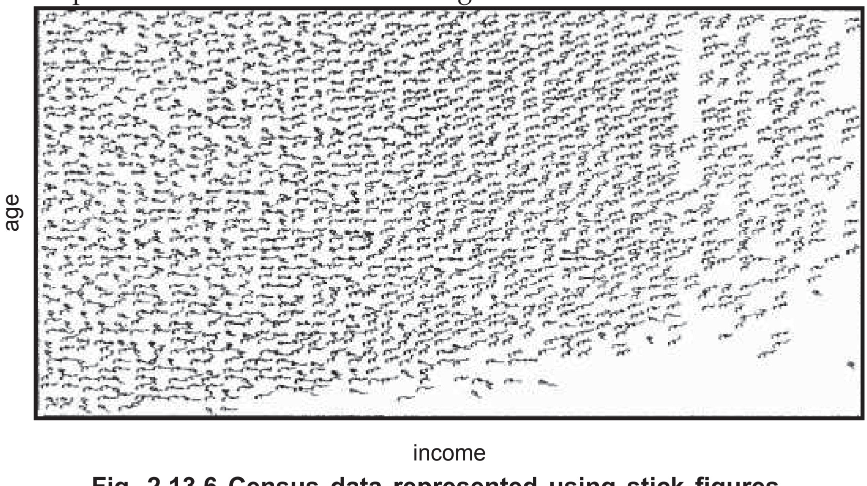


Fig. 2.13.6 Census data represented using stick figures

2.13.4 Heirarchical Visualization Techniques

- The above mentioned methods are suitable for visualising multiple dimensions together.
- This will be challenging for the data of very high dimensionality to visualize all dimensions at a time.
- In hierarchical visualization techniques all the dimensions are partitioned into subjects.
- Two hierarchical visualization techniques are shown in the Fig. 2.13.7 and Fig. 2.13.8. (See Fig. 2.13.8 on next page).
  1. World-within-worlds
  2. Tree-maps

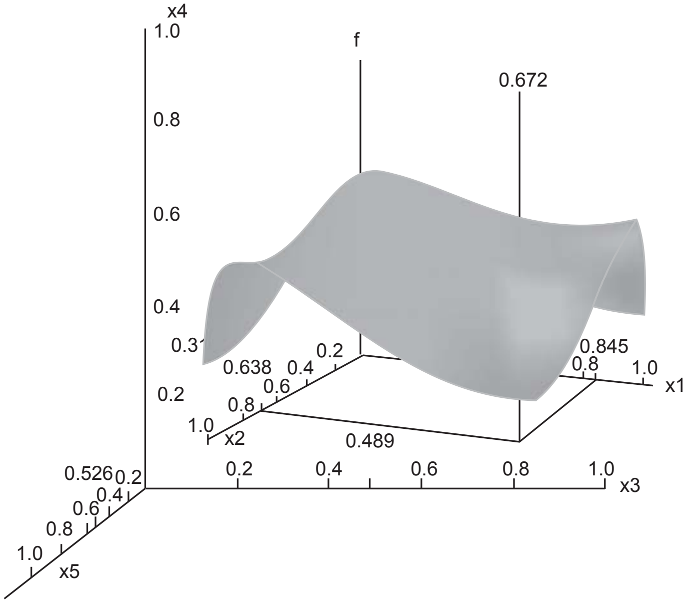


Fig. 2.13.7 Worlds-within-worlds (also known as n-vision)

1. Worlds-within-worlds (n-vision) :

- Consider example of 6D dataset with dimensions F, X, X_2, ..., X_5.
- Let's see how dimension F changes w.r. to other dimension.
- First step is to fix dimension X_3, X_4, X_5 to selected values C_3, C_4, C_5.
- Next visualize F, X, X_2 to 3D plot called as world.
- So Inner world = (C_3 , C_4 , C_5), Outer world = X_3 , X_4 , X_5 .
- Dimension in worlds and number of worlds can be enhanced further according to need.

2. Tree maps :

- Hierarchical data is visualized as set of nested rectangles.
- Consider the example shown in Fig. 2.13.8.

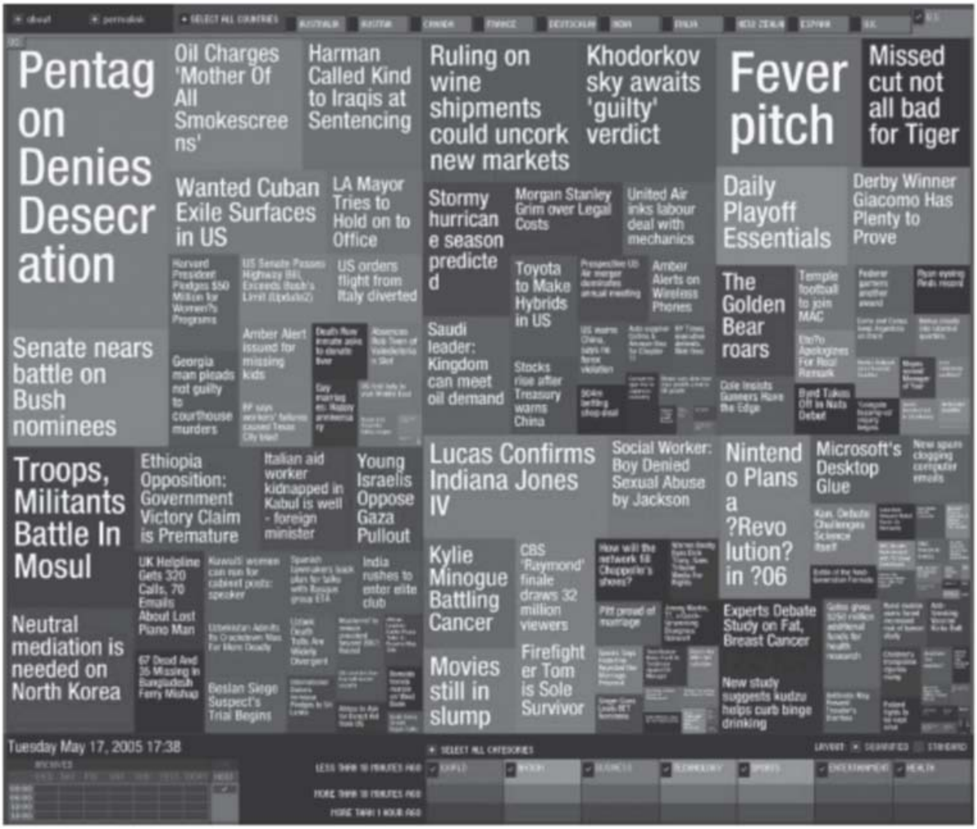

Pentag on Denies Desecr ation
Oil Charges 'Mother Of All Smokescree ns'
Harman Called Kind to Iraqis at Sentencing
Ruling on wine shipments could uncork new markets
Khodorkov sky awaits 'guilty' verdict
Fever pitch
Missed cut not all bad for Tiger
Wanted Cuban Exile Surfaces in US
LA Mayor Tries to Hold on to Office
Stormy hurrican e season predicte d
Morgan Stanley Grim over Legal Costs
United Air inks labour deal with mechanics
Daily Playoff Essentials
Derby Winner Giacomo Has Plenty to Prove
US orders flight from Italy diverted
Toyota to Make Hybrids in US
Amber Alerts on Wireless Phones
The Golden Bear roars
Saudi leader: Kingdom can meet oil demand
Stocks rise after Treasury warns China
Senate nears battle on Bush nominees
Troops, Militants Battle In Mosul
Ethiopia Opposition: Government Victory Claim is Premature
Italian aid worker kidnapped in Kabul is well - foreign minister
Young Israelis Oppose Gaza Pullout
Lucas Confirms Indiana Jones IV
Social Worker: Boy Denied Sexual Abuse by Jackson
Nintend o Plans a ?Revo lution? in ?06
Microsoft's Desktop Glue
Kylie Minogue Battling Cancer
CBS 'Raymond' finale draws 32 million viewers
Experts Debate Study on Fat, Breast Cancer
Neutral mediation is needed on North Korea
UK Helpline Gets 320 Calls, 70 Emails About Lost Piano Man
India rushes to enter elite club
Movies still in slump
Firefight er Tom is Sole Survivor
Beslan Siege Suspect's Trial Begins
New study suggests kudzu helps curb binge drinking
Tuesday May 17, 2005 17:38

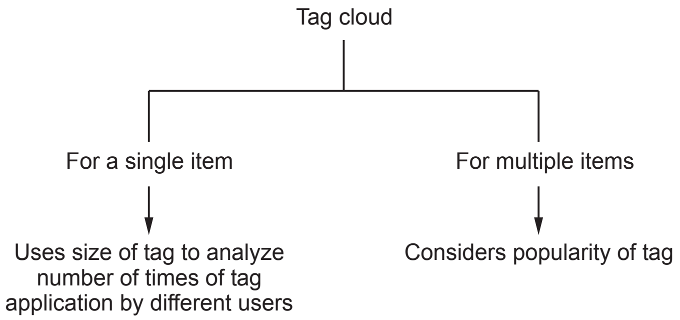
Fig. 2.13.8 Newsmap : Use of tree-map to visualize Google newws headline stories

- In this, each Google news story in represented by a rectangle of unique color based on category.

2.13.5 Visualizing Complex Data and Relations

- This technique on representation of non-numeric data such as text from social media.
- Tag cloud method is used for this.
- The tagged objects like pictures, blog entries and product reviews on web are considered.
- In a cloud these tags are listed alphabetically or accending to user performance.

- Tag cloud is used in two ways.


Tag cloud
For a single item
For multiple items
Uses size of tag to analyze number of times of tag application by different users
Considers popularity of tag

2.14 Measuring Data Similarity and Dissimilarity

- For building the suitable marketing scheme sometimes it is needed to know the likelihood of the objects. For example : Customer object with similar characteristics like similar income area, etc.
- If we consider clustering applications, the object in a cluster are similar to each other and different from objects in other clusters.
- Similarity and dissimilarity measures are known as measures of proximity.
- If object i is completely similar to object j similarity measure = 1 and vice versa.
- Two data structures commonly used for calculation of proximity measures are
  - data matrix
  - dissimialarity matrix

2.14.1 Data Matrix Verses Dissimilarity Matrix

- Consider the objects with multiple attributes.
- Let's consider n objects with p attributes.

n = persons, items, course, etc.

p = age, height, gender, etc. (features)

- Let the objects are

$$x_1 = (x_{11}, x_{12}, \dots, x_{1p}),$$

$$x_2 = (x_{21}, x_{22}, \dots, x_{2p})$$

where x_{ij} = value for object x_i of j^{th} attribute or i.

- Objects are the tuples known as data samples or feature vectors.
- Data matrix stores n data objects in the form of n × p matrix.

$$\begin{bmatrix} x_{11} & \cdots & x_{1f} & \cdots & x_{1p} \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ x_{i1} & \cdots & x_{if} & \cdots & x_{ip} \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ x_{n1} & \cdots & x_{nf} & \cdots & x_{np} \end{bmatrix} \leftarrow \text{row = object}$$

- Data matrix is called as two mode matrix as it contains two entities.
- **Dissimilarity matrix :**
  - ○ The proximities for all pairs of n objects are listed.
  - ○ Represented by n × n table as below :

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

d(i,j) = Dissimilarity or difference between objects i and j.

- ○ d(i, j) is a non-negative number.
- ○ d(i, j) = 0 if i and j are highly similar.
- ○ d(i, j) = d(j, i)

- Dissimilarity matrix is called as one mode matrix as it contain only kind of entity.

2.14.2 Proximity Measures for Nominal Attributes

- Let's consider the states of nominal attributes as M.
- States = Letters, symbols, etc. for 1, 2, 3, …, M.
- The dissimilarity can be computed as

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

where m = No. of matches i.e. number of attributes having i = j

p = Total number of attributes describing objects.

- Similarity can be computed as

$$\text{sim}\,(i,\,j) \;=\; 1 - d\,(i,\,j) = \frac{m}{p}$$

2.14.3 Proximity Measures for Binary Attributes

- As discussed in earlier sections we known that binary attribute has only state 0 or 1.
- Dissimilarity calculations :
  - Let's consider are binary attributes has same weight.
  - We get below 2 × 2 contiguous table

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

- It i and j are symmetric binary attributes (i.e. if i and j are equally valuable) then dissimilarity can be calculated as

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

- If i and j are asymmetric then

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

- Similarity can be calculated as

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

- sim (i, j) is known as Jaccard coefficient.

2.14.4 Dissimilarity of Numeric Data : Minkowski Distance

- The measures which are commonly used for computing dissimilarity of numeric data are
  1. Euclidean distance
  2. Manhttan distance
  3. Minkowski distance
- Typically to cover greater range of the data (for example : height) associated with an attribute, the data is normalized and can be expressed in smaller units.
- Later on the distance calculations are applied on this data.

- **Euclidean distance** or straight line distance is a popular distance measure
- Consider $i = (x_{i1}, x_{i2}, \ldots, x_{ip})$ and $j = (x_{j1}, x_{j2}, \ldots, x_{jp})$ are two objects with p numeric attributes.
- The Euclidean distance between these objects can be measured as :

  $$d(i, j) = \sqrt{(x_{i1} - x_{j1})^2 + (x_{i2} - x_{j2})^2 + \ldots + (x_{ip} - x_{jp})^2}$$
- For example : Consider objects $x_1 = (1, 2)$ and $x_2 = (3, 5)$

  Euclidean distance between these two is $\sqrt{2^2 + 3^2} = 3.61$.
- **Manhattan distance** or city block distance is called so as it is the distance a car would drive in a car (e.g. Manhattan) where the buildings are laid out in square blocks and the straight streets intersect at right angles.
- It is calculated as

  $$d(i, j) = |x_{i1} - x_{j1}| + |x_{i2} - x_{j2}| + \ldots + |x_{ip} - x_{jp}|$$
- For example : For objects $x_1 = (1, 2)$ and $x_2 = (3, 5)$ it is 2 + 3 = 5.
- **Minkowski distance** generalized Euclidean and Manhattan distances.
- It can be calculated as

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

where $h \geq \pm 1$ is real number

- Minkwoski distance is also known as $\mathbf{L_p}$ **norm** where p = h.
- Minkwoski distance can be generalized for $h \to \infty$ to the distance called as supremum distance or L_{max} or $L_{\infty\,norm}$ or chebyshev distance which can be measured as

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

This is also called as uniform norm.

- For example : for $x_1 = (1, 2)$ and $x_2(3, 5)$ the supremum distance is 5 – 2 = 3.

2.15 Dissimilarity for Attributes of Mixed Types

- In the earlier sections we have discussed dissimilarity between attribute types -
  - Nominal
  - Symmetric binary

- ○ Asymmetric binary
- ○ Numeric
- ○ Ordinal

- It is observed that real databases objects contain mixture of different attribute types mentioned above.
- To measure dissimilarity between objects of mixed attribute types each type of attribute is clusted and analysed.
- Another way is to perform single analysis by processing all attributes together.
- Let's consider mixture of p attributes in the data set. In this case dissimilarity d(i, j) between objects i and j is calculated as

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

where indicator $\delta_{ij}^{(f)} = 0$

if x_{if} or x_{jf} is missing

OR

$x_{if} = x_{jf} = 0$ and f is asymmetric binary

else

$$\delta_{ij}^{(f)} = 1$$

- If attribute **f** is

1. Numeric then $d_{if}^{(f)} = \frac{|x_{if} - x_{jf}|}{\max_h x_{hf} - \min_h x_{hf}}$.

2. Nominal or binary then $d_{if}^{(f)} = 0$ if $x_{if} = x_{jf}$ otherwise $d_{if}^{(f)} = 1$.

3. Ordinal then compute ranks

$$r_{if} = z_{if} = \frac{r_{if} - 1}{M_f - 1}$$

and z_{if} = Numeric

Cosine similarity :

- Cosine similarity is used for comparison of the document.
- To understand this measure consider following example of collection of different documents with the occurrance of particular words.

Document	team	coach	score	W_{in}
Document 1	5	0	0	2
Document 2	3	0	0	1
Document 3	0	7	0	3
Document 4	0	1	2	0

Document Vector Table

- In the above table frequency of particular word or purchase in document is recorded.
- Each document is considered as an object which is represented by term-frequency vector.
- These vectors are long and sparse.
- Let's consider x and y are two vectors.
- Then cosine similarity function can be stated as

$$\text{sim}(x, y) = \frac{x \cdot y}{\|x\| \, | \, \|y\|}$$

where $\|x\|$ = Euclidean norm of x where $x = (x_1, x_2, \ldots, x_p)$ and defined as $\sqrt{x_1^2 + x_2^2 + \ldots + x_p^2}$. Similarly $\|y\|$ = Euclidean norm of y.

- If cosine value = 0, then x and y are 90° to each other as they do not match.
- If cosine value is close to 1 and if the angle is small it indicates greater match.
- The cosine similarity has applications in information retrieval, text document clustering, biological taxonomy and gene feature mapping.
- Consider values of x and y as

$$x = (5, 0, 3, 0, 2, 0, 2, 0, 0)$$

$$y = (3, 0, 2, 0, 1, 1, 0, 1, 0, 1)$$

cosine similarity can be calculated as

- $x^t \cdot y = 5 \times 3 + 0 \times 0 + 3 \times 2 + 0 \times 0 + 2 \times 1 + 0 \times 1 + 0 \times 0 + 2 \times 1 + 0 \times 0 + 0 \times 1 = 25$

$$\|x\| = \sqrt{5^2 + 0^2 + 3^2 + 0^2 + 2^2 + 0^2 + 0^2 + 2^2 + 0^2 + 0^2} = 6.48$$

$$\|y\| = \sqrt{3^2 + 0^2 + 2^2 + 0^2 + 1^2 + 1^2 + 0^2 + 1^2 + 0^2 + 1^2} = 4.12$$

$$\text{sim}(x, y) = \frac{x \cdot y}{\|x\| \, | \, \|y\|} = 0.94$$

- For binary valued attributes cosine similarity function can be given as

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

- This function is known as Tanimoto coefficient or Tanimoto distance which has applications in information retrieval and biology taxonomy.

Two Marks Question with Answer

Q.1 What is data mining ?

Ans. : • Data mining is a field of research that has emerged in the 1990s, and is very popular today.

- Data mining is a set method that applies to large and complex databases. This is to eliminate the randomness and discover the hidden pattern.
- The reason why data mining has become popular is that, storing data electronically has become very cheap and that trasfeering data can now be done very quickly thanks to the fast computer networks that we have today.
- Traditionally, data has been analyzed by hand to discover interesting knowledge. However, this is time-consuming, prone to error, doing this may miss some importatnt information, and it is just not realistic to do this on large databases.
- To address this problem, automatic techniques have been designed to analyze data and extract interesting patterns, trends or other useful information. This is the purpose of data mining.
- In general, data mining techniques are designed either to explain or understand the past (e.g. Why a plane has crashed) or predict the future (e.g. Predict if there will be an earthquake tomorrow at a given location).

Q.2 List the steps in knowlwdge discovery process.

Ans. : • These steps are :

1) Data cleaning : For removing noise and inconsistent data.

2) Data integration : For combining multiple data sources.

3) Data selection : For retrieves of relevent data from database.

4) Data transformation : Summary or aggregation operations are performed for transforming the data.

5) Data mining : Data patterns are extracted by applying intelligent methods.

Q.3 What are machine learning techniques ?

Ans. : **1) Supervised learning :**

- It is a classification technique.
- Based on the labeled examples in training data set it facilitates the supervised learning of classification model.

2) Unsupervised learning :

- It is a clustering technique.
- Classes are discovered within the data.

3) Semi supervised learning :

- It uses both labeled and unlabeled data in learning process.
- There are different approaches. In one, unlabeled examples are considered as boundary elements. In another, labeled and unlabeled examples are considered as positive and negative examples.

Q.4 Explain in brief aspects of mining methodology.

Ans. : • Various aspects of mining methodologies are :

1. Mining various and new kinds of knowledge :

- ○ The issues are faced due to diversity of applications.

2. Mining knowledge in multidimensional space :

- ○ To mine large data multi dimensional data mining techniques are used.

3. Data mining - an interdisciplinary effort :

- ○ For interdisciplinary fields like natural language text mining, mixing methods need to be combined with methods of information retrieval and natural language processing.

4. Boosting the power of discovery in a networked environment :

5. Handling uncertainty, noise or incompleteness of data :

6. Pattern evaluation and pattern guided mining :

- ○ The techniques are required to know the interesting patterns among all the patterns.

Q.5 What is business intelligence ?

Ans. : • The term Business Intelligence (BI) refers to technologies, applications and practices for the collection, integration, analysis and presentation of business information.

- The purpose of BI is to support better business decision making.
- Data mining is required in BI to perform effective market analysis, compare customer feedback on similar products to discover strengths and weakness of their competitors to retain highly valuable customers and to make smart business decisions.

- Data warehousing and multidimensional data mining is used in online analytical processing tool.
- Classification and prediction techniques are used in predictive analytics.
- Clustering is used in customer relationship management.

Q.6 What are web search engines ?

Ans. : • A web search engine or internet search engine is a software system that is designed to carry out web search, which means to search the world wide web in a systematic way for particular information specified in a textual web search query.

- User query results are returned at a list or hits.
- The hits consist of web pages, images and other types of files.
- Different data mining techniques are used extensively in web search engines.
- Crawling, indexing and searching are some of them.

Q.7 What are the challenges faced by data mining usage in web search engines ?

Ans. : • Challenges which can be faced by data mining usage are :

- ○ Handling of humongous amount of data getting generated daily.
- ○ Use of computer clouds, consisting of thousands or hundreds of thousands of computers to work on data mining methods and large distributed data sets.
- ○ To deal with online data. A query classifier need to be built for this to handle the queries on predifined categories.
- ○ To handle context aware queries. In context aware query search engine tries to find out context of query using users profile to give customized answers in very small amount of time.
- ○ Most of the queries are asked only once which is challenging for data mining methods.

Q.8 What are nominal attributes ?

Ans. : • Nominal attributes relate to names.

- Values of these attributes are :
  - ○ Symbols or
  - ○ Names of things
- Values are also called as enumerations.
- They are also known as categorical attributes as the value of the attribute can be
  - ○ Category
  - ○ Code or
  - ○ State
- The attributes are not arranged in meaningful order.

Q.9 What are binary attributes ?

Ans. : • A binary attribute can take only two states, either a 0 or 1.

- State 0 : attribute is absent, State 1 : attribute is present.
- They are also known as Boolean attributes if states are true or false.
- If both the states in binary attribute contain equal weightage (for ex : Attribute gender with states male and female) it is said to be symmetric.
- If both the states in binary attribute are not equally important (for ex : Attribute medical test for HIV with states positive and negative) it is said to be asymmetric.

Q.10 What are ordinal attributes ?

Ans. : • In an ordinal attribute the values are arranged in a meaningful order or rank.

- However even if the values are in meaningful sequence the magnitude between them is missing.
- This feature is useful in the application like conduction of a survey, where ratings are important. For ex. In the survey of any product the ordinal categories can be : 0 : very dissatisfied, 1 : somewhat dissatisfied, 2 : neutral, 3 : satisfied, and 4 : very satisfied.

Q.11 What are numeric attributes ?

Ans. : • They are quantitative attribute which can be measured and typically represented in integer or real values.

- They can be
- 1. Interval-scaled or 2. ratio-scaled.

1. Interval-Scaled Attributes :

- These attributes are measured on a scale of equal-size units.
- They follow specific order and can have positive, negative or zero value.
- They can be used to compare and quantify the difference between values.

2. Ratio-Scaled Attributes :

- These are numeric attribute with an inherent zero-point that means we can consider the value as a multiple of another value.
- They follow specific order and difference between values, mean, median, and mode can be calculated.

Discrete versus Continuous Attributes :

- A discrete attribute is finite or countably infinite set of values, which may or may not be represented as integers.
- For ex : hair color, smoker, medical test, and drink size, or numeric values 0 to 110 for the attribute age. etc.

- The attributes customer ID can also have countably infinite values.
- Continuous values are typically real numbers.In practice continuous attributes are typically represented as floating-point variables.

Q.12 What is mean and how it is calculated ?

Ans. : • The first measure is the mean which means average.

- To calculate the mean add together all of the numbers in your data set.
- Then divide that sum by the number of addends.
- Let $x_1 x_2 ... x_N$ be a set of N values or observations such as for some numeric attribute X then The mean of this data is

$$\bar{x} = \frac{\sum_{i=1}^{N} x_i}{N} = \frac{x_1 + x_2 + ... + x_N}{N}$$

- For the given data set of items the mean is calculated as

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

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

Q.13 What is median and how it is calculated ?

Ans. : • It is a measure of central tendency in which the middle number is when listed in order from least to greatest.

- For skewed (asymmetric) data a better measure of the center of data is the median.
- It is the middle value in a set of ordered data values.
- The median is the value that separates the higher half of a data set from the lower half
- To calculate median
  - Suppose that a given data set of N values for an attribute X is sorted in increasing order.
  - If N is odd then the median is the middle value of the ordered set.
  - If N is even then the median is not unique; it is the two middlemost values and any value in between.
  - If X is a numeric attribute in this case by convention the median is taken as the average of the two middlemost values.

Q.14 What are outliners ?

Ans. : • In statistics, an outlier is a data point that differs significantly from other observations or patterns.

- An outlier is an observation that lies an abnormal distance from other values in a random sample from a population.
- An outlier may be due to variability in the measurement or it may indicate experimental error; the latter are sometimes excluded from the data set.
- A common rule of thumb for identifying suspected outliers is to single out values falling at least 1.5 × IQR above the third quartile or below the first quartile.

Q.15 What are boxplots ?

Ans. : • Boxplots are a popular way of visualizing a distribution.

- A boxplot incorporates the five-number summary as follows :
  - Typically, the ends of the box are at the quartiles so that the box length is the interquartile range.
  - The median is marked by a line within the box.
  - Two lines (called whiskers) outside the box extends to the smallest (Minimum) and largest (Maximum) observations. (Refer Fig. 2.7.4 on page 2 - 21).
- Boxplots can be computed in On logn time. Approximate boxplots can be computed in linear or sublinear time depending on the quality guarantee required.

Q.16 What are histograms ?

Ans. : • Histograms (or frequency histograms) are at least a century old and are widely used. "Histos" means pole or mast, and "gram" means chart, so a histogram is a chart of poles.

- Plotting histograms is a graphical method for summarizing the distribution of a given attribute, X.

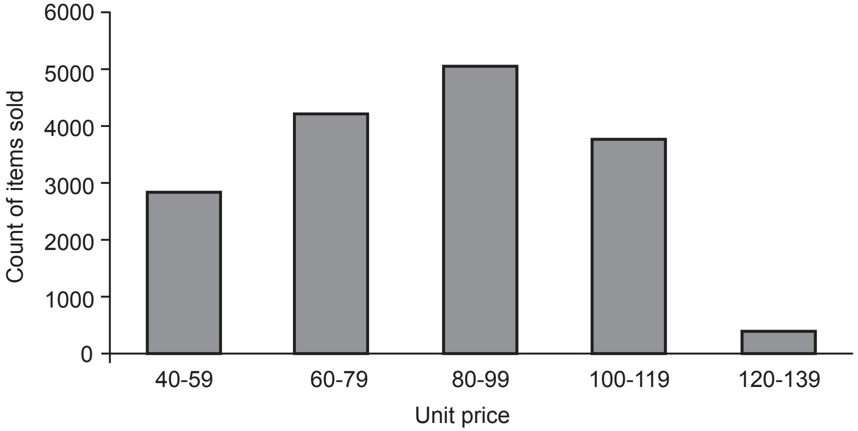


Fig. 1 A histogram for the Table 2.7.2 data set.

- If X is nominal, such as automobile model or item type, then a pole or vertical bar is drawn for each known value of X.
- The height of the bar indicates the frequency (i.e., count) of that X value. The resulting graph is more commonly known as a bar chart.

Q.17 What are scatter plots ?

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

- To construct a scatter plot, each pair of values is treated as a pair of coordinates in an algebraic sense and plotted as points in the plane. (Refer Fig. 2.7.8 on page 2 - 25).
- The scatter plot is a useful method for providing a first look at bivariate data to see clusters of points and outliers, or to explore the possibility of correlation relationships.
- Two attributes, X and Y, are correlated if one attribute implies the other. Correlations can be positive, negative, or null (uncorrelated).

Q.18 List different preprocessing techniques with their roles.

Ans. : • Different preprossing techniques which can be applied are :

- Data cleaning : To remove noise and correction of inconsistency in data.
- Data integration : To merge data from different sources to a warehouse.
- Data reduction : To reduce the data size by techniques like aggregation clustering etc.
- Data transformation : To scale the data by normalization .

Q.19 How to address the problem of missing values in data cleaning process ?

Ans. : • Consider the example of sales and customer data, if the attributes like customer income are missing from the tuples, following methods can be adapted to fill in such missing values :

- Ignore the tuple
- Fill in the missing value manually
- Use a global constant to fill in the missing value
- Use a measure of central tendency for the attribute (e.g., the mean or median) to fill in the missing values
- Use the attribute mean or median for all samples belonging to the same class as the given tuple.
  - For skewed data distribution median value can be used.
- **Use the most probable value to fill in the missing value :**
  - Using other attribute values missing values can be predicted.

- This can be done by using the techniques like regression, inference-based tools using a Bayesian formalism or decision tree induction.
  - For ex : Missing values for income of the customers can be predicted by constructing a decision tree using the other customer attributes in the data set.

Q.20 Explain various ways of binning.

Ans. : • There are three ways of binning :

a) Smoothing by bin means :

- ○ Each value in a bin is replaced by mean value of a bin.
- ○ For ex : As shown in 1 mean of values 4, 8 and 15 is 9.
- ○ So bin 1 values are replaced by 9.

b) Smoothing by bin medians :

- ○ Each value in a bin is replaced by bin median value.

c) Smoothing by bin boundaries :

- Minimum and maximum values in a bin are called as bin boundaries.
- Each value in a bin is replaced closet boundary value.

2) Regression :

- It is a technique data values are standardized to a function.
- In linear regression best line is found which will fit two attributes.
- Based on this one attribute can be used to predict other.
- In multiple liner regression, linear regression technique is extended for more than two attributes.

3) Outlier analysis :

- Outlier is an data point which differs significantly from other observations.
- One way of detecting outliners is through clustering.
- The organization of values in groups is called as cluster.

Q.21 What is X^2 correlation test of nominal data ?

Ans. : X^2 **correlation test of Nominal data**

- It is used to find relationship between two attributes A and B.
- Consider A with C distinct values $a_1, a_2, ... a_c$.
- B with r distinct values $b_1, b_2, ... b_r$.
- With A as column and B as rows. we can write a contingency table.
- Consider (A_i , B_j) is a joint event.

- where $(A = a_i, \ B = b_j)$.
- Each joint event (A_i, B_j) has one slot in the table.

Let o_{ij} is observed frequency of (A_i, B_j)

e_{ij} is expected frequency of (A_i, B_j)

where $e_{ij} = \dfrac{\text{Count }(A = a_i) \times \text{Count }(B = b_j)}{n}$

n = Number of data tables.

Q.22 What are wavelet transforms ?

Ans. : • The discrete wavelet transform (DWT) is a linear signal processing technique through which the data vector X is transformed to new vector X' of same length of wavelet coefficients.

- Each tuple is considered as an n-dimensional data vector, where $X = (x_1, x_2, \ldots, x_n)$ with n database attributes having n attributes.
- The advantage is that the wavelet transformed data can be truncated.
- Only a small fraction of the strongest of the wavelet coefficients is retained, keeping value of all other coefficients 0.
- The resultant data representation is sparse and can be computed with high speed in wavelet space.

Q.23 Explain briefly principal components analysis ?

Ans. : • Principal components analysis (PCA) is also called as Karhunen - Loeve or K-L method.

- It reduces the data represented by types or data vectors.
- K n-dimensional orthogonal vectors are searched to represent the data where $K \leq n$.
- It uses following process :
  1. Normalization of input data.
  2. K orthonormal vectors are computed, known as principal components.
  3. Sorting of principal components is done by decreasing strength.
  4. Data size is reduced by elimination of weaker components

Q.24 List different ways of sampling.

Ans. : 1. Simple random sample without replacement (SRSWOR) of size S

2. Simple random sample with replacement (SRSWR) of size S

3. Cluster sample

4. Stratified sample

Q.25 What are data transformation strategies ?

Ans. : • Following are the strategies for data transformation :

1. Smoothing : Removal of noise from data.

2. Attribute construction : Construction and addition of new attributes for efficient mining process.

3. Aggregation : Helpful in data analysis as data is summarized.

4. Normalization : Scale down the data to a smaller range. For ex : – 1.0 to 1.0

5. Discretization : Generation of concept hierarchy.

6. Concept hierarchy generation for nominal data :

- Generalization of the attributes is done.

Q.26 Explain in brief Icon based visualization techniques.

Ans. : • In this approach small icons are used to represent multidimensional data values.

- Two popular techniques are :
  1. Chernoff faces 2. Stick figures

1. Chernoff faces :

- Cartoon human figure is used to represent the data upto 18 dimensions.
- The values of dimensions are represented by face components like eyes, ears, mouth, nose based on their shape, size, placement and orientation.

2. Stick figure :

- Stick figure is of five pieces four limbs and a body.
- Multidimensional data is mapped to this figure.
- Two dimensional → to the display axes remaining → angle of limbs.

Q.27 What is Eclidean distance ?

Ans. : • Euclidean distance or straight line distance is a popular distance measure

- Consider i = $(x_{i1}, x_{i2}, \dots, x_{ip})$ and j = $(x_{j1}, x_{j2}, \dots, x_{jp})$ are two objects with p numeric attributes.
- The Euclidean distance between these objects can be measured as :
- $d(i, j) = \sqrt{(x_{i1} - x_{j1})^2 + (x_{i2} - x_{j2})^2 + \dots + (x_{ip} - x_{jp})^2}$
- For example : Consider objects x_1 = (1, 2) and x_2 = (3, 5)

  Euclidean distance between these two is $\sqrt{2^2 + 3^2}$ = 3.61.

Q.28 What is Manhattan distance ?

Ans. : • **Manhattan distance** or city block distance is called so as it is the distance a car would drive in a car (e.g. Manhattan) where the buildings are laid out in square blocks and the straight streets intersect at right angles.

- It is calculated as

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

- For example : For objects x_1 = (1, 2) and x_2 = (3, 5) it is 2 + 3 = 5.

Q.29 What is Minkowski distance ?

Ans. : • **Minkowski distance** generalized Euclidean and Manhattan distances.

- It can be calculated as

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

where h ≥ ± 1 is real number

- Minkwoski distance is also known as $\mathbf{L_p}$ **norm** where p = h.
- Minkwoski distance can be generalized for $h \to \infty$ to the distance called as supremum distance or L_{max} or $L_{\infty\,norm}$ or chebyshev distance which can be measured as
- $$d\ (i,\ j) = \lim_{h \to \infty}\left(\sum_{f=1}^{p}\left|x_{if} - x_{if}\right|^h\right)^{1/h} = \max_{f}^{p}\left|x_{if} - x_{if}\right|$$

This is also called as uniform norm.

- For example : for x_1 = (1, 2) and x_2(3, 5) the supremum distance is 5 – 2 = 3.

Q.30 What is cosine similarity ?

Ans. : • Cosine similarity is used for comparison of the document.

- To understand this measure consider following example of collection of different documents with the occurrance of particular words.

Document	team	coach	score	W_{in}
Document 1	5	0	0	2
Document 2	3	0	0	1
Document 3	0	7	0	3
Document 4	0	1	2	0

Document Vector Table

- In the above table frequency of particular word or purchase in document is recorded.
- Each document is considered as an object which is represented by term-frequency vector.

- These vectors are long and sparse.
- Let's consider x and y are two vectors.
- Then cosine similarity function can be stated as

$$\text{sim}(x, y) = \frac{x \cdot y}{\|x\| \, | \, \|y\|}$$

Notes

Unit - III

3 Data Mining - Frequent Pattern Analysis

Syllabus

Mining Frequent Patterns, Associations and Correlations – Mining Methods - Pattern Evaluation Method – Pattern Mining in Multilevel, Multi Dimensional Space – Constraint Based Frequent Pattern Mining, Classification using Frequent Patterns

Contents

3.1 Mining Frequent Patterns, Association and Correlations

- Generally it is observed that customer tend to buy certain products one after other or together due to strong correlation between them.
- For ex. : Milk and bread, camera and memory card.
- Such patterns are known as frequent patterns whose patterns can be itemsets, subsequences or substructures.
- A sequential pattern is the one in which the patterns occur in some order and frequently in the database. For ex. : Pattern of buying PC first then digital camera and then memory card.
- A substructure is the one in which different structural forms like subgraphs, subtrees, etc are combined.
- Frequency of sub-structure in a database is known as structured pattern.
- To exploit these relationships and correlation between items, frequent itemset mining focusses on discovery of association and correlations in these items.
- This is helpful in many strategic decisions in business like catelog design, cross marketing and customer shopping behaviour analysis.
- **Market basket analysis** is one of the techniques to facilitate this.

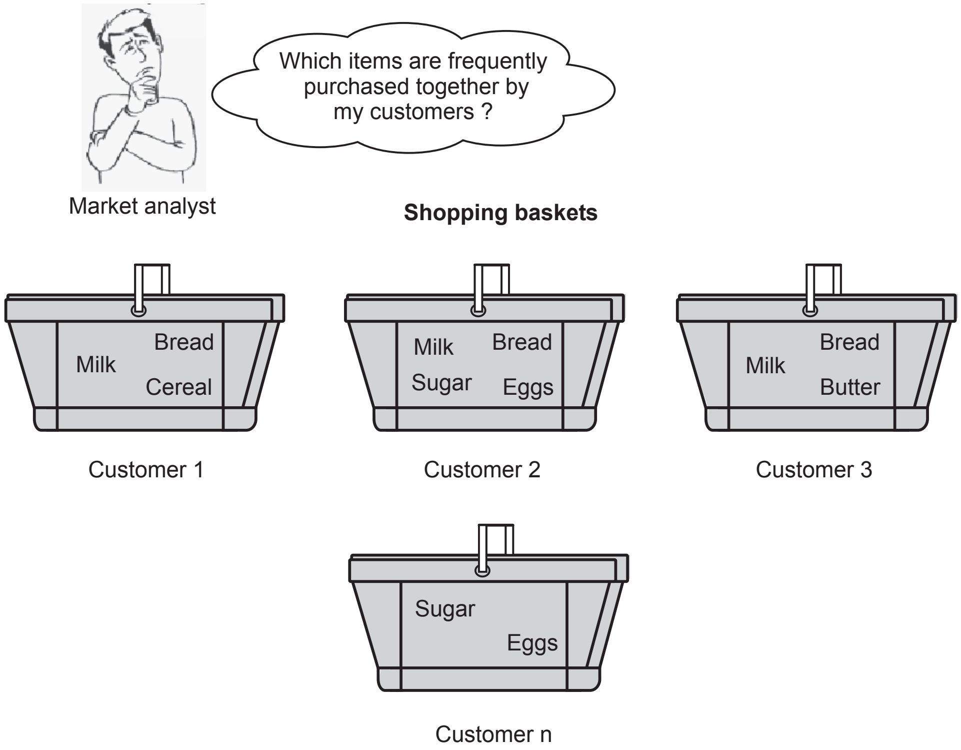


Fig. 3.1.1 Market basket analysis

- In market basket analysis, the customer buying habits are studied.
- As shown in Fig. 3.1.1, the association between different items placed in shopping baskets are studied.
- It helps to understand the items which are frequently and together purchased by the customers.
- For ex. : If a customer is visiting supermarket what is the frequency of buying bread with milk.
- This analysis help shopkeepers to build marketing strategy and to manage the shelf space.
- Various strategies can be planned like design of stone layout or the items are kept together to encourage the combined sale.
- Let's consider universe is a set of items present at store.
- In this each items can be represented by a boolean varaible to know the the presence or absence of the item.
- Each basket can be represented by a boolean vector of values assigned to these variables.
- From these vector buying patterns can be understood.
- These patterns can be represented in the form of association rules.
- For ex. : The association rule

  Computer ⇒ antiviruses - software [Support = 2 %, confindence = 60 %] shows the information about the purchase of computer and antivirus software together where support and rule are measures of rule interestingness.
- In the above example value of support is 2 % indicating the purchase of computer and antivirus together according to transaction analysis.
- Confindence is 60 % indicating the purchase of computer and software by 60 % customers.
- Some threshold value of support and confidence can be set by users or domain experts.
- The rule becomes more useful if both the values satisfy minimum threshold.

3.1.1 Frequent Itemset Mining Methods

Apriori Algorithm :

- Apriori algorithm focusses on mining frequent itemsets for boolean association rules.

- It employs interactive search called level wise search in which k itemsets are used to explore (k + 1) itemsets.
- It takes following steps :

  1. Scan the database.
  2. Calculate and accumulate count of each item
  3. Collect the items having minimum support
  4. This set will be denoted by 11 which is set of 1 itemsets.
  5. Next set of 2 itemsets is found, which is named as L_2.
  6. L_2 is used to find L_3 and so on.
  7. To find L_k by above mentioned steps one full scan of database is needed.
  8. Efficiency of these steps is enhanced by the property known as apriori property.

Apriori property :

- Apriori property can be stated as, "All nonempty subsets of a frequency itemset must also be frequent."
- Apriori property follows the property called antimonotonicity which states that, " If a set cannot pass a test, all of its subsets will fail the same test as well."
- Apriori property is based on below observations :

  1. If set I does not satisfy minimum support thersold min-sup, then I is not frequent i.e. P(I) < min-sup.
  2. If an item A is added to itemset - T resulting itemset (IUA) cannot occur more frequently than I.
  3. This indicates that IUA is also not frequent i.e. P(I U A) < min-sup

- This property can be applied to algorithm as follows :

  L_k is found using L_{k-1} for $k \geq 2$.

- It is a two step process consisting of

  1. The join step
  2. The prune step

1. The join step :

Following operations are perfomed in join steps :

- Join L_{k-1} with itself to generate set of candidate k-itemsets C_k
- Consider l_1 and l_2 as itemsets in L_{k-1}

- $l_i[j]$ refers to jth item l_i
- Items in itemset are considered to be sorted in lexicographic order.
- i.e. for (k – 1) itemset, the items are sorted like $l_i[1] < l_i[2] < \ldots < l_i[k-1]$.
- Later $L_{k-1} \bowtie L_{k-1}$ join is performed where L_{k-1} members can be joined if their first (k – 2) items is common.
- Members l_1 and l_2 of L_{k-1} are joined if

  $(l_1[1] = l_2[1] \wedge (l_1[2] = l_2[1] \wedge \ldots (l_1[k-2] = l_2[k-2]) \wedge (l_1[k-1] < l_2[k-1])$.
- $l_1[k-1] < l_2[k-1]$ ensures that no duplicates are generated.
- The resultant itemset after joining l_1 and l_2 is

  $\{l_1[1], l_1[2], \ldots, l_1[k-2], l_1[k-1], l_2[k-1]\}$.

2. The prune step :

- C_k is considered as superset of L_k, which indicates that even if the members of L_k are not frequent all frequent k itemsets are included is C_k.
- To determine L_k a database scan is performend to determine court of each candidate in C_k.
- C_k involves heavy computation as it is huge.
- Size of C_k can be reduced by apriori property as below :

  1. Candidate can be removed from C_k, if (k – 1).
- Subset of candidate k it itemset is not in L_{k-1} as any non-frequent (k – 1) itemset cannot be a subset of a frequent k-itemset.
- To understand apriori algorithm consider the example below :

Example 3.1.1 *Apriori, Let's look at a concrete example, based on the Allelectronics transaction database, D, of Table 3.1.1. There are nine transactions in this database, that its |D|= 9. We use Fig. 3.1.1 to illustrate the Apriori algorithm for finding frequent itemsets in D.*

Solution :

1. In the first iteration of the algorithm, each item is a member of the set of candidate 1-itemsets, C_1. The algorithm simply scans all of the transactions to count the number of occurrences of each item.

2. Suppose that the minimum support count required is 2, that is, min_sup = 2. The set of frequent 1-itemsets. L_1, can then be determined. All of the candidates in C_1 satisfy minimum support.

3. To discover the set of frequent 2-itemsets, L_2, the algorithm use the join $L_1 \bowtie L_1$ to generate a candidate set of 2-itemsets. C_2^7, C_2 consists of $\binom{|L_1|}{2}$ 2-itemsets. No candidates are removed from C_2 during the prune step because each subset of the candidates is also frequent.

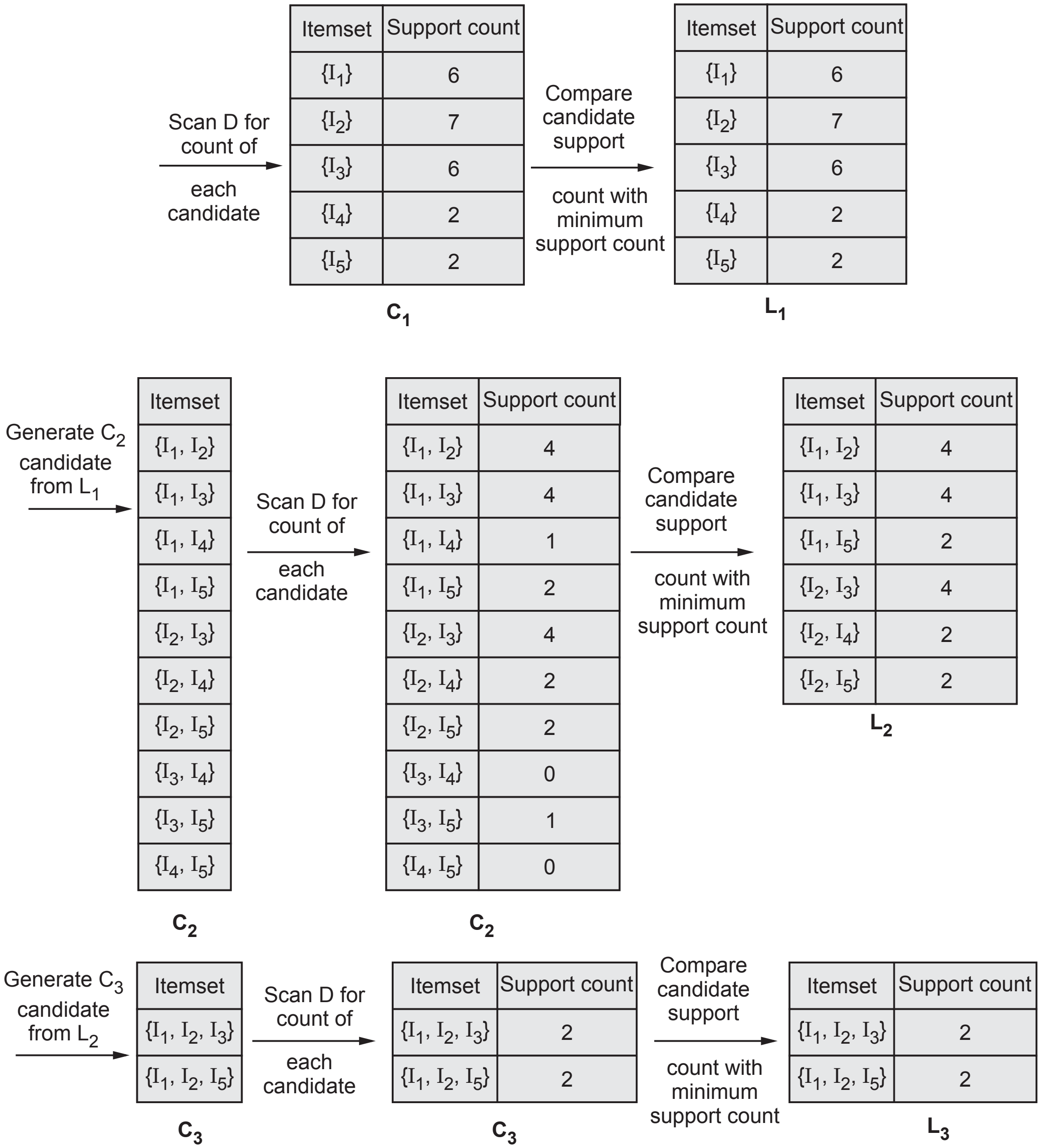
TID	List of item_IDs
T100	I1, I2, I3
T200	I2, I4
T300	I2, I3
T400	I1, I2, I4
T500	I1, I3
T600	I2, 13
T700	11, I3
T800	11, 12, 13, 15
T900	11, 12, 13

Table 3.1.1 Transactional data for an Allelectronics branch

4. Next, the transactions in D are scanned and the support count of each candidate itemset in C_2 is accumulated, as shown in the middle table of the second row in Fig. 3.1.2.

5. The set of frequent 2-itemets, L_2, is then determined, consisting of those candidate 2-itemests in C_2 having minimum support.

6. The generation of the set of the candidate 3-itemsets, C_3, is detailed in Fig. 3.1.2. From the join step, we first get $C_3 = L_2 \bowtie L_2$ = {{11, 12, 13}, (11, 12, 15), (11, 13, 15), {12, 13, 14}, (12, 14, 15). (k – 1) subsets are frequent since the Apriori algorithm uses a level-wise search strategy. The resulting pruned version of C_3 is shown in the first table of the bottom row of Fig. 3.1.2.


Scan D for count of each candidate
Compare candidate support count with minimum support count
Generate C_2 candidate from L_1
Scan D for count of each candidate
Compare candidate support count with minimum support count
Generate C_3 candidate from L_2
Scan D for count of each candidate
Compare candidate support count with minimum support count

C_1

Itemset	Support count
$\{I_1\}$	6
$\{I_2\}$	7
$\{I_3\}$	6
$\{I_4\}$	2
$\{I_5\}$	2

L_1

Itemset	Support count
$\{I_1\}$	6
$\{I_2\}$	7
$\{I_3\}$	6
$\{I_4\}$	2
$\{I_5\}$	2

C_2

Itemset
$\{I_1, I_2\}$
$\{I_1, I_3\}$
$\{I_1, I_4\}$
$\{I_1, I_5\}$
$\{I_2, I_3\}$
$\{I_2, I_4\}$
$\{I_2, I_5\}$
$\{I_3, I_4\}$
$\{I_3, I_5\}$
$\{I_4, I_5\}$

C_2

Itemset	Support count
$\{I_1, I_2\}$	4
$\{I_1, I_3\}$	4
$\{I_1, I_4\}$	1
$\{I_1, I_5\}$	2
$\{I_2, I_3\}$	4
$\{I_2, I_4\}$	2
$\{I_2, I_5\}$	2
$\{I_3, I_4\}$	0
$\{I_3, I_5\}$	1
$\{I_4, I_5\}$	0

L_2

Itemset	Support count
$\{I_1, I_2\}$	4
$\{I_1, I_3\}$	4
$\{I_1, I_5\}$	2
$\{I_2, I_3\}$	4
$\{I_2, I_4\}$	2
$\{I_2, I_5\}$	2

C_3

Itemset
$\{I_1, I_2, I_3\}$
$\{I_1, I_2, I_5\}$

C_3

Itemset	Support count
$\{I_1, I_2, I_3\}$	2
$\{I_1, I_2, I_5\}$	2

L_3

Itemset	Support count
$\{I_1, I_2, I_3\}$	2
$\{I_1, I_2, I_5\}$	2

Fig. 3.1.2 Generation of the candidate itemsets and frequent itemsets, where the minimum support count is 2.

Generation and pruning of candidate 3-itemset, C_3, from L_2 using the Apiori property

(a) Join : $C_3 = L_2 \bowtie L_2$ = {{11, 12}, {11, 13}, {11, 15}, {12, 13}, {12, 14}, {12, 15}}

$\bowtie$ {{11, 12}, {11, 13}, {11, 15}, {12, 13}, {12, 14}, {12, 15}}

= {{11, 12, 13}, {11, 12, 15}, {11, 13, 15}, {12, 13, 14}, {12, 13, 15}, {12, 14, 15}}

(b) Prune using the Apriori property : All nonempty subsets of a frequent itemset must also frequent. Do any of the candidates have a subset that is not frequent ?

- ○ The 2-item subsets of {11, 12, 13} are {11, 12}, {11, 13} and {12, 13}. All 2-item subsets of {11, 12, 13} are members of L_2. Therefore keep {11, 12, 13} in C_3.
- ○ The 2-item subsets of {11, 12, 15} are {11, 12}, {11, 15} and {12, 15}. All 2-item subsets of {11, 12, 15} are members of L_2. Therefore keep {11, 12, 15} in C_3.
- ○ The 2-item subsets of {11, 13, 15} are {11, 13}, {11, 15} and {13, 15}, {13, 15} is not a member of L_2 and so it is not frequent. Therefore, remove {11, 13, 15} from C_3.
- ○ The 2-item subsets of {12, 13, 14} are {12, 13}, {12, 14} and {13, 14}, {13, 14} is not a member of L_2 and so it is not frequent. Therefore, remove {12, 13, 14} from C_3.
- ○ The 2-item subsets of {12, 13, 15} are {12, 13}, {12, 15} and {13, 15}, {13, 15} is not a member of L_2 and so it is not frequent. Therefore, remove {12, 13, 15} from C_3.
- ○ The 2-item subsets of {12, 14, 15} are {12, 14}, {12, 15} and {14, 15), {14, 15} is not a member of L_2 and so it is not frequent. Therefore, remove {12, 14, 15} from C_3.

(c) Therefore, C_3 = {{11, 12, 13}, {11. 12, 15}} after pruning.

Fig. 3.1.2 Generation of pruning of candidate 3-itemsets C_3 from L_2 using Apriori property.

7. The transactions in D are scanned to determine L_3, consisting of those candidate 3-itemsets in C_3 having minimum support (Fig. 3.1.2).
8. The algorithm uses $L_3 \bowtie L_3$ to generate a candidate set of 4-itemsets, C_4. Although the join results in {11, 12, 13, 15}, itemset {11, 12, 13, 15} is pruned because its subset {12, 13, 15} is not frequent. Thus $C_4 = \phi$ and the algorithm terminates, having found all of the frequent itemsets.

Apriori algorithm for discovering frequent itemsets for mining Boolean association rules

- The pseudo code for apriori algorithm in given as below.

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

Input :

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

Output : L. frequent itemsets in D.

Methods :

```
$L_1$ = find_frequent_1-itemsets (D) ;
for (k = 2; $L_{k-1} \neq \phi$ ; k + +) {
    $C_k$ = apriori_gen ($L_{k-1}$) ;
    for each transaction $t \in D$ {// scan D for  counts
        $C_t$ = subset ($C_k$, t); // get the subsets of t that are candidates
        for each candidate $c \in C_t$
            c.count ++;
    }
    $L_k = \{c \in C_k | c.count \geq min\_sup\}$
}
return $L = \cup_k L_k$
```

Procedure apriori_gen (L_{k-1} ; frequent (k – 1) - itemsets)

```
for each itemset $l_1 \in L_{k-1}$
    for each itemset $l_2 \in L_{k-1}$
        if $(l_1[1] = l_2[1]) \wedge (l_1[2] = l_2[2])$
            $\wedge ... \wedge (l_1 [k-2] = l_2[k-2]) \wedge (l_1[k-1] < (l_2[k-1])$ then {
            $c = l_1 \bowtie l_2$ ; // joint step ; generate candidates
            if has_infrequent_subset (c, $L_{k-1}$) then
                delete  c : //prune step; remove unfruitful candidate
            else add c to $C_k$;
        }
return $C_k$;
```

procedure has_infrequent_subset (c : candidate k-itemset;

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

3.1.2 Generating Association Rules from Frequent Itemsets

- After generating frequent itemsets, the strong association rules can be generated from them by following equation of confidence.

$$\text{Confidence } (A \Rightarrow B) = P(B \mid A) = \frac{\text{Support } - \text{ Count}(A \cup B)}{\text{Support } - \text{Count}(A)}$$

where,

Support – count $(A \cup B) \rightarrow$ No. of transactions containing itemset $A \cup B$.

Support – count (A) → No. of transaction containing itemset A.

- Considering this equation, the association rules are stated as :
  - For each frequent itemset l, generate all nonempty subsets of l.
  - For every nonempty subset δ of l, output the rule "$S \Rightarrow (l - S)$" if $\frac{\text{Support} - \text{count}(l)}{\text{Support} - \text{count}(S)} \geq$ min_conf.

    where,

    min_conf is minimum confidence threshold.

3.1.3 Improving the Efficiency of Apriori

Following are the techniques to increase efficiency of apriori algorithm.

1. Hasn based technique :

- It is used to reduce size of candidate k itemsets C_k for $k > 1$.
- Two imtemsets for each transaction are calculated.
- They are hashed or mapped into different buckets of hash table structure.
- Increase bucket count.

2. Transaction reduction :

- Remove the transactions which not contain frequent k itemset as they cannot contain frequent (k + 1) itemsets.

3. Partitioning :

- This technique needs only two database scans for mining frequent itemsets.
- This can be done in two phases :
- **Phase I :**
  - Divide transactions of D in n non-overlappling partitions.
  - If support threshold for tansaction D is min-sup then, minimum support count for a partition is min-sup X the number of transaction in that partition.
  - Calculate local frequent itemsets for each partitioner.
- **Phase II :**
  - The actual support of each candidate is assessed for determination of global frequent itemsets.

3. Sampling :

- In this technique random samples in picked of given data D.
- Later frequent itemsets are searched in S instead of D.

4. Dynamic itemset counting :

- In this database in partitioned into blocks.
- These blocks are marked by start point.
- New candidate itemset can be added at any starting point.
- Count-so-far is calculated and if it passes minimum support, itemset is added into frequent itemsets collection.
- Refer Fig. 3.1.4 for this process.

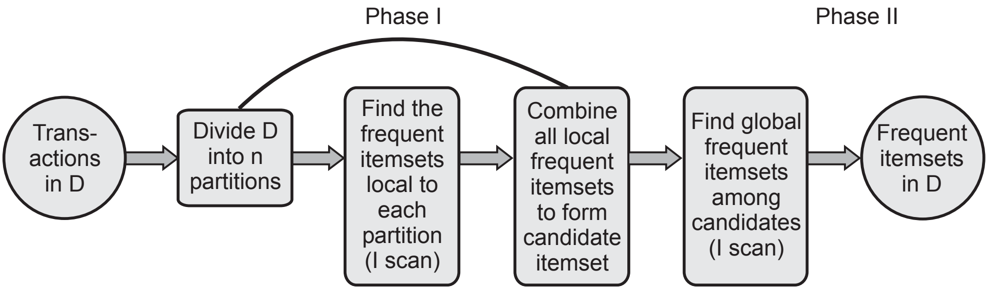

Phase I
Phase II
Trans-actions in D
Divide D into n partitions
Find the frequent itemsets local to each partition (I scan)
Combine all local frequent itemsets to form candidate itemset
Find global frequent itemsets among candidates (I scan)
Frequent itemsets in D

Fig. 3.1.4 Mining by partitioning the data

3.1.4 A Pattern Growth Approach for Mining Frequent Itemsets

- The two points which makes candidate generation process costly in apriori algorithm are :
  - Huge number of candidate sets are generated for example : if there are 10^4 frequent one itemset, more than 10^7 candidate two itemsets.
  - Repeated scanning of database is needed.
- To reduce the cost the technique used is frequent pattern growth or FP growth.
- In this the database is compressed to generate frequent pattern tree or FP tree.
- Next the compressed database is divided into set of conditonal databases.
- First step is same as apriori.
- Frequent items and support counts are calculated and list L is created by descending support count value. So, L = {{I2 : 7}, {I1 : 6}, {I3 :6}, {I4 : 2} {I5 : 2}}

- In the next step FP tree construction takes place by following steps :
  - Creat root
  - Label it with null value
  - Scan database D
  - Create a branch for each transaction
  - Consider the example of Table 3.1.1. First transaction contain three items T100 : I1, I2, I5

By this the construction of first branch is done with three nodes < I2 : 1>, <I1 : 1> and <I5 : 1> as shown in Fig. 3.1.5.

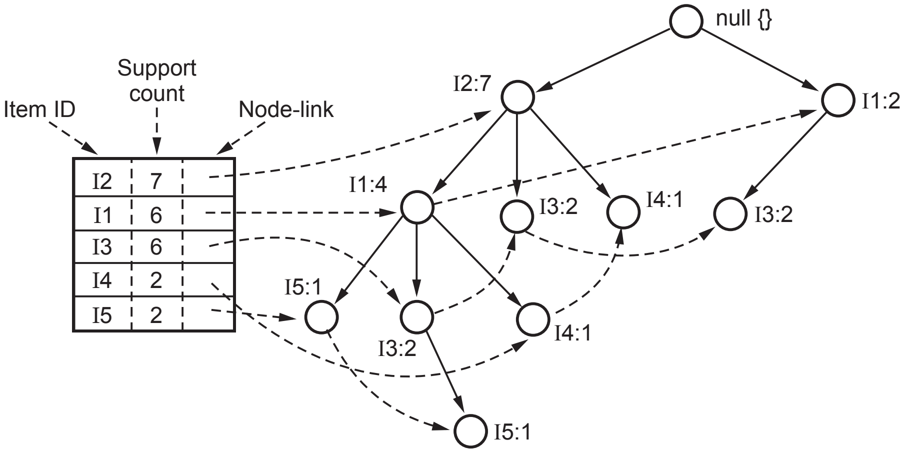


Fig. 3.1.5 An FP-tree registers compressed, frequent pattern information

3.1.5 Mining Frequent Itemsets using Vertical Data Format

- Apriori and Fp-growth methods horizontal data format for mining.
- The data can be stored in item-TID-set format,
  where, item is item name and TID-set is set of transaction identifiers.
- This format is known as vertical data format.

Example 3.1.2 *Mining frequent itemsets using the vertical data format.*

Consider the horizontal data format of the transaction database. D, of Table 3.1.1 in Example 3.1.1. This can be transformed into the vertical data format shown in Table 3.1.1 by scanning the data set once.

Mining can be performed m this data set by intersecting the TID_sets of every pair of frequent single items. The minimum support count is 2. Because every single item is, frequent in Table 3.1.2, there are 10 intersections performed in total, which lead to eight nonempty 2-itemsets, as shown in Table 3.1.3. Notice that because the itemsets {11, 14} and {13, 15} each contain only one transaction, they donot belong to the set of frequent 2-itemsets.

Itemeset	TID_set
11	{T100, T400, T500, T700, T800, T900}
12	{T100, T200, T300, T400, T600, T800, T900}
13	{T300, T500, T600, T700, T800, T900}
14	{T200, T400}
15	{T100, T800}

Table 3.1.2 The vertical data format of the Transaction Data Set D of Table 3.1.1.

Itemeset	TID_set
{11, 12}	{T100, T400, T800, T900}
{11, 13}	{T500, T700, T800, T900}
{11, 14}	{T400}
{11, 15}	{T100, T800}
{12, 13}	{T300, T600, T800, T900}
{12, 14}	{T200, T400}
{12, 15}	{T100, T800}
{13, 15}	{T800}

Table 3.1.3 2-Itemsets in Vertical Data Format

Itemeset	TID_set
{11, 12, 13}	{T800, T900}
{11, 12, 15}	{T100, T800}

Table 3.1.4 3-Itemses in Vertical Data Format

Based on the Apriori property, a given 3-itemset is a candidate 3-itemset is a candidate 3-itemset only if every one of its 2-itemest subsets is frequent. The candidate generation process here will generate only two 3-itemsets; {11, 12, 13} and {11, 12, 15}. By intersecting the TID_sets of any two corresponding 2-temsets of these candidate 3-itemesets, it derives Table 3.1.4, where there are only two frequent 3-itemsets : {11, 12, 13 : 2} and {11, 12, 15 : 2}

- As shown in the example first horizontally formatted data is transformed to vertically formatted data.
- Support cannot of itemset = length of TID set and itemset.
- Start with k = 1 and build candidate (k+1) itemsets.
- Increment k each time.
- Use the technique diffset to track the differences by TID_sets of (k+1) itemset and corresponding k_itemset to reduce the cost.

3.1.6 Mining closed and Max Patterns

- Instead of deriving set of all frequent itemsets it is sometimes useful to derive the closed frequent itemsets.
- The approach is :
  - To search for closed frequent itemsets directly by pruning the search space.
  - Pruning strategies include :

    1. Item merging : It follows following rule :

    If every transaction containing a frequent itemset X also contain itemset Y but not any proper subset of Y, then X U V forms a frequent closed itemset and there is no need to search for any itemset containing X but no Y.

    2. Sub itemset pruning : It follows following rule :

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

    3. **Item skipping :** It follows the rule :

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

3.2 Pattern Evaluation Methods

3.2.1 Strong Rules are not Necessarily Interesting

- It is not always necessary that strong association rules give correct result.
- To understand this, consider the below example.

Example 3.2.1 *A misleading "strong" association rule.*

Solution : Suppose we are intersted in analyzing transactions at AllElectronics with respect tot he purchase of computer games and videos. Let game refer to the transactions containing computer games and video refer to those containing videos. Of the 10,000 transactions analyzed, the data show that 6000 of the customer transactions included computer games, while 7500 included video and 4000 included both computer games and videos. Suppose that a data mining program for discovering association rules is run on the data, using a minimum support of say, 30 % and a minmum confidence of 60 %.

The following association rule is discovered.

buys (X, "computer games") $\Rightarrow$ buys (X, "videos")

[support = 40%, confidence = 66%] ...(3.2.1)

Rule (3.2.1) is a strong association rule and would therefore be reported, since its support value of $\frac{4000}{10{,}000} = 40\ \%$ and confidence value of $\frac{4000}{6000} = 66\ \%$ satisfy the minimum support and minimum confidence thresholds, respectively. However, Rule (3.2.1) is misleading because the probability of purchasing videos is 75 %, which is even larger than 66 %. In fact, computer games and videos are negatively associated because the purchase of one of these iterms actually decreases the likelihood of purchasing the other. Without fully understanding this phenomenon, we could easily make unwise business decisions based on Rule (3.2.1).

3.2.2 From Association Analysis to Correlation Analysis

- Support and confidence parameters are not able to handle come associations rules.
- By addition of correlation measure the rule will be measured by additional parameter of correlation between itemsets.
- So, the rule can be given as
  $A \Rightarrow B$ [Support, confidence, correlation]
- Let A and B are two itemsets
  A is independent of B if $P(A \cup B) = P(A)\,P(B)$
  otherwise A and B are dependent and correlated.
- This concept leads to the correlation measure lift.

  where,

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

- If $\text{lift}(A, B) < 1$ and vice versa → occurrence of A is negatively correlated w.r.t. B.

- If lift $(A,B) = 1 \rightarrow$ A and B are independent.
- Lift can also be measured as,
  $P(B|A) / P(B)$ OR conf $(A \Rightarrow B) / \sup(B)$

3.2.3 A Comparison of Pattern Evaluation Measures

- The efficiency of the measures like lift and x^2 can be measured against four more measures named as :
  1. All-confidence
  2. Max-confidence
  3. Kulczyski
  4. Cosine

1. all-confidence measure :

- For itemsets A and B all-confidence measure is defined as

$$\text{all-conf}\ (A,B) = \frac{\sup(A \cup B)}{\max\left\{\sup(A), \sup(B)\right\}}$$

$$= \min\left\{P(A|B), P(B|A)\right\}$$

where,
$\text{Max.}\left\{\sup(A), \sup(B)\right\}$ = Max. Support of A and B

2. max-confidence measure :

- It is defined as max-conf $(A,B) = \text{Max}\left\{(A|B), P(B|A)\right\}$

3. Kulczynski measure :

- It can be defined as

$$\text{kulc}\ (A, B) = \frac{1}{2}(P(A|B) + P(B|A))$$

- It is average of two confidence measures.

4. Cosine measure :

- It is defined as

$$\text{Cosine}\ (A, B) = \frac{P(A \cup B)}{\sqrt{P(A) \times P(B)}}$$

$$= \frac{\sup(A \cup B)}{\sqrt{\sup(A) \times \sup(B)}}$$

$$= \sqrt{P(A|B) \times P(B|A)}$$

- It is known as **harmonized lift** measure.
- All the four measures explained above takes into account the fact that the value of each measure is influenced by the conditional probabilities P(A|B) and P(B|A) and not by total number of transactions.

3.3 Pattern Mining in Multilevel Mutidimensional Space

- This section first focuses on multilevel associations involving different abstraction levels.

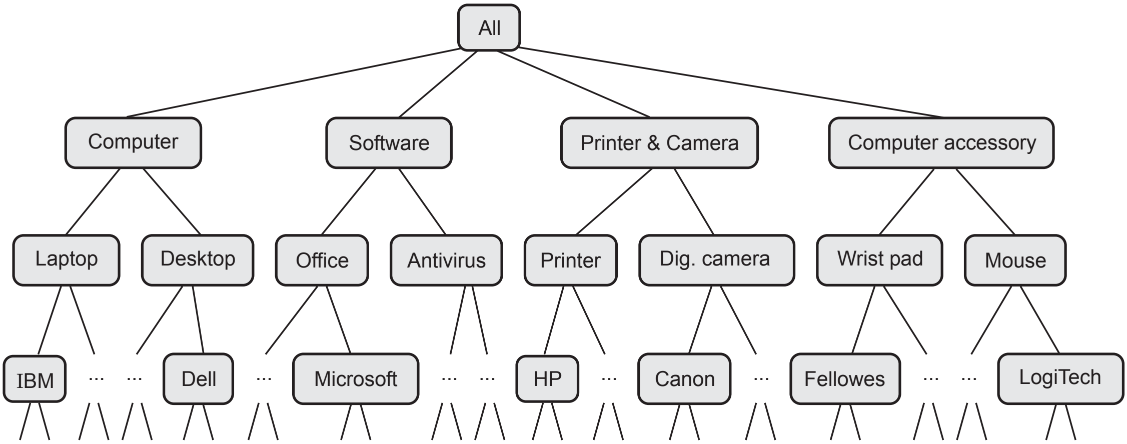


Fig. 3.1.6 Concept hierarchy for AllElectronic computer items.

3.3.1 Mining Mutilevel Association

- Consider the example shown in Fig. 3.1.6.
- It shows the concept hierarchy of the transactions from Table 3.1.5.

TID	Items Purchased
T100	Apple 17" MacBook Pro Notebook. HP Photosmart Pro b9180
T200	Microsoft Office Professional 2010. Microsoft Wireless Optical Mouse 5000
T300	Logitech VX Nano cordiess Laser Mouse, Fellowes GEL Wrist Rest
T400	Dell studio XPS 16 Notebook. Canon Power Shot SD1400

T500	Lenow Think Pad X200. Table PC, Siystemec. Norton Antiivirus 2010.

Table 3.1.5 Task - Relevant Data, D

- Concept hierarchy contains sequence of mapping from low to high level concepts.
- As shown in Fig. 3.1.6 concept hierarchy contain four levels 0 to 4.
  where,
  level 0 → root node
  level 4 → specific abstraction level containing raw data values.
- It is observed that the data present in Table 3.1.5 is raw data which is at the lowest level in concept hierarchy.
- With this raw data it is difficult to find the purchase patterns.
- The rules which are generated from mining data called as multiple level or multilevel association rules.
- They are generally calculated using support confidence framework.
- Three variations in this approach are described below :

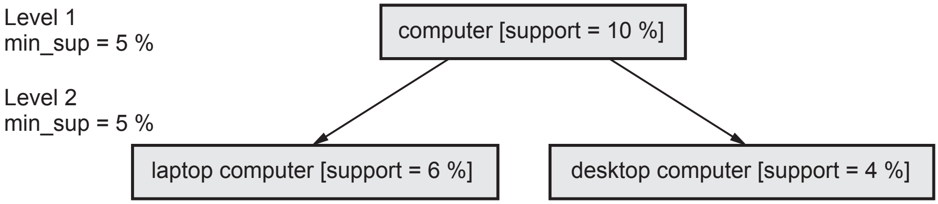

Level 1
min_sup = 5 %
Level 2
min_sup = 5 %
computer [support = 10 %]
laptop computer [support = 6 %]
desktop computer [support = 4 %]

Fig. 3.1.7 Multilevel mining with uniform support

1. Uniform support for all levels

- As shown in Fig. 3.1.7 minimum support level of 5 % is provided.
- This simplifies the search procedure as the search is avoided to find the itemsets whose ancestors has minimum support.
- The drawbacks of this technique are some meaningful associations may be missed at low abstraction level.

2. Using reduced support at lower levels :

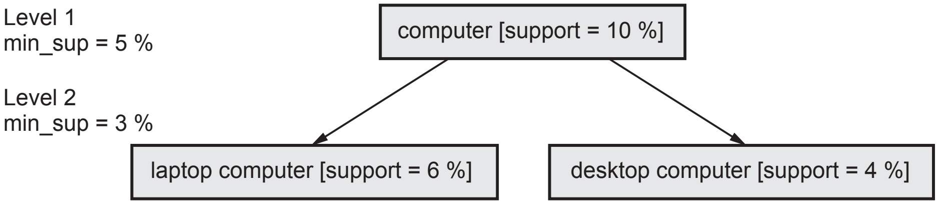

Level 1
min_sup = 5 %
Level 2
min_sup = 3 %
computer [support = 10 %]
laptop computer [support = 6 %]
desktop computer [support = 4 %]

Fig. 3.1.8 Multilevel mining with reduced support

- As shown in Fig. 3.1.8 for deep abstraction levels minimum support threshold is provided.

3. Using item or group based minimum support :

- Based on the importance of groups according to the opinion of users and experts, group based threshold is assigned.
  For example : Camera with price over 10,000 is given a low threshold value.
- The drawback of multilevel association os for multiple abstraction levels redundant rules are generated.

3.3.2 Mining Multidimensional Associations

- The boolean association rule like below buys (X, "digital camera") ⇒ buys (X, "HP printer") is a single dimensional association rule as only one predicate is involved.
- The rule which has two or more dimensions is known as multidimensional association rule.
  For ex. : age (X, "20…29") ∧ occupation (X, "student") ⇒ buys (X, "laptop").
- In the above example all the predicates appear once such rules are known as interdimensional association rules.
- If the predicates appear more than once in the rule it is in called as hybrid-dimensional association rule.
  For ex. : age (X, "20…29") ∧ buys (X, "laptop") ⇒ buys (X, "HP printers").
- If the attributes are of quantitative type there are two approaches -
  - Mining rules using static discretization of quantitative attributes.
  - Dynamic quantitative association rules.

3.3.3 Mining Quantitative Association Rules

- In this section three methods of a mining quantitative association rules are discussed.

  1) A data cube method

  2) A clustering based method

  3) A statistical analysis method

Data cube method

- In this numeric values are replaced by interval labels.

- After applications of the suitable mining algorithm transformed multidimensional data is used to construct a data cube.
- n-predicate sets are stored in the cells of n-dimensional cuboid.
- Due to enhancement in data warehouse and OLAP technology it may happen that the dimensions which are of interest of user may already exists.

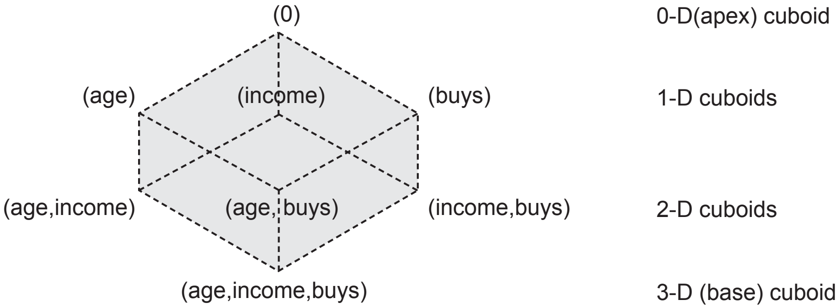


Fig. 3.1.9 Lattice of cuboids, making up a 3-D data cube. Each cuboid represents a different group-by. The base cuboid contains the three predicates age, income and buys.

- In this case these values can be computed by low level aggregates and values are relieved.
- The structure of cuboid is shown in figure 3.1.9

Clustering Based method

- It considers the fact that frequent patterns are found in dense clusters.
- Top down approach can be applied as follows :
  - First standard clustering algorithm is applied to find clusters and satisfying minimum support threshold.
  - 2D spaces are examined by combining two clusters.
  - If this combination passes minimum support threshold, continue searching for the clusters.
- Bottom up approach can be applied as follows :
  - High dimension space is used to form clusters.
  - These clusters are then projected and merged.
  - This is not so feasible approach as finding high dimensional clustering is a difficult problem.

Using statistical theory to disclose exceptional behaviour

- It focuses on the exceptional behaviour, defined based on statistical theory.

- To understand the exceptional behaviour consider following rule.

  sex = female ⇒ meanwage 7.90 ₹/hr (overall_mean_wage = ₹ 9.02/hr)

- The lower wage than the mean wage calculated makes this rule exceptional.
- The statistical test for ex. Z test in above case is applied to confirm validity of the rule.
- Based on this new definition is formed.

  Population_subset ⇒ mean_of_values_for_the_subset

Mining Rase Patterns and Negative Patterns

- Sometimes it is useful to find the rare patterns of patterns with negative correlation.

3.4 Constraint-Based Frequent Pattern Mining

- In the process of data mining many rules may be formed which are sometimes not related to the user.
- If user specify intuitions or expectations in the form of constraints to generate useful rules it will be interesting.
- This technique in known as constraint based mining.
- Different types of constraints :

  1. **Knowledge type constraints :** Considers knowledge to be mined like association, correlation, classification or clustering.
  2. **Data constraints :** Specify set of task relevant data.
  3. **Dimension/level constraints :** Specify desired dimensions
  4. **Interestingness constraints :** Specify threshold on statistical measures like support, confidence and correlation.
  5. **Rule constraints :** Specify conduction on rules to be mined.
  6. **Syntactic rule constraints :** Can be specified in template structure by the technique metarule-guided mining.

3.4.1 Metarule Guided Mining of Association Rules

- Metarules are used to specify syntactic form of interested rules.
- They are based on analysis, experience, expectations or intuitions regarding data.

Example 3.4.1 *Metarule-guided mining.*

Solutions : Suppose that as a market analyst for AllElectronics you have access to the data describing customers (e.g. customer age, address and credit rating) as well as the list of customer transactions. You are interested in finding associations between customer traits and the items that customers buy. However rather than finding all of the association rules reflecting these relationships, you are interested only in determining which pairs of customer traits promote the sale of office software. A metarule can be used to specify this information describing the form of rules you are interested in finding. An example of such a metarule is,

$$P_1(X, Y) \wedge P_2(X, W) \Rightarrow \text{buys } (X, \text{"office software"}) \qquad \ldots(3.4.1)$$

where P_1 and P_2 are **predicate variables** that are instantiated to attributes from the given database during the mining process, X is a variable representing a customer, and Y and W take on values of the attributes assigned to P_1 and P_2 respectively. Typically a user will specify a list of attributes to be considered for instantiation with P_1 and P_2. Otherwise a default set may be used.

Constraint based pattern generation :

Pruning pattern space and pruning data space :

- To indicate expected set as subset relationship of the varieties, constraints on aggregate functions etc. the rule constraints are used.
- Consider following example.

Example 3.4.2 *Constraints for mining association rules.*

Solution : Suppose that AllElectronics has a sales multidimensional database with the following interrelated relations :

item(item_ID,item_name, description, category, price)

sales (transaction_ID, day, month, year, store_ID, city)

trans_itemi(item_ID, transaction_ID)

Here, the item table contains attributes item_ID, item_name, description, category and price the sales table contains attributes transaction_ID day, month, year, store_ID and city and the tow tables are linked via the foreign key attributes, item_ID and transaction_ID, in the table trans_item.

Suppose our association mining query is "find the patterns or rules about the sales of which cheap items (where the sum of the prices is less than \$10) may promote (i.e. appear in the same transaction) the sales of which expensive items (where the minimum price is \$50), shown in the sales in Chicago in 2010"

This query contains the following four constraints :

1) sum(I.price) < \$10, where I represents the item_ID of a cheap item;

2) min(Jprice) ≥ \$50), where J represents the item_ID of an expensive item;

3) T.city = Chicago and

4) T.year = 2010, where T represents a transaction_ID. for conciseness, we do not show the mining query explicitly here, however, the constraints' context is clear from the mining query semantics.

- From the earlier discussion it is clear that pruning a technique used to reduce the size of decision tree by remaining its sections.

- With this context the constraint which facilitates patterns space pruning is called as pattern pruning constraint and the one used for data pruning is called data pruning control.

3.4.2 Pruning Pattern Space with Pattern Pruning Constraints

Constraints interact with pattern mining process by five ways

1) Anti monotonic :

- It follows the rule that if an itemset does not satisfy the rule constraint that even if we add the items in itemset it will make is more expensive, none of its suspects can satisfy the constraint.
- This is known as anti monotonic property

2) Monotonic :

- It follows the rule that if itemset satisfy the constraints that the addition of more items to itemset will increase the cost and satisfy the constraint, its supersets will also follow this rule.
- This is known as monotonic property.

3) Succinct :

- It follows the rule that the sets that are guaranteed to satisfy the constraints can be enumerated.

4) Convertible and inconvertible :

- The constraints which belongs to none of the above category they are convertible rules.

3.5 Classification using Frequent Patterns

- The interesting relationships between attribute_value pairs occurring frequently in a given data set is given by frequent patterns.
- The frequent patterns like age = youth and credit = ok can be considered as a single item.
- Search of such frequent patterns is called as frequent pattern mining or frequent itemset mining.
- These frequent patterns can be used for classification by- two ways.
  - associative classification
  - discriminative frequent pattern based classification.

3.5.1 Associative Classification

- The different methods for associative classification are :
  1. CBA (Classification Based on Associations)
  2. CMAR (Classification Based on Multiple Association Rule)
  3. CPAR (Classification Based on Predictive Association Rule)
- Before discussing the above methods lets first understand the basics of associative classification.
- Association rule mining is done in two steps
  - ○ Step 1 : Mining frequent itemsets
  - ○ Step 2 : Rule generation

For example :

For dataset D the association rule can be given as,

$$\begin{aligned} \text{age} &= \text{youth} \quad \text{credit} = \text{OK} \quad \text{buys} - \text{computer} \\ &= \text{yes} \;\; [\text{support} = 20\ \%, \text{confidence} = 93\ \%] \end{aligned}$$

where is a logical AND

Let

D data set of tuple

D has $A_1, A_2, \ldots , A_n$ attributes

A_{class} class attribute

p attribute value pain $(A_i, \;)$

where A_i takes value

$$x = (x_1, x_2, \cdots, x_n) \text{ satisfy item } P = (A_i, \;)$$

$$\text{if } x_i = \quad \text{where } x_i \text{ is } i^{\text{th}} \text{ attribute of X.}$$

For association rule the form

$p_1 \quad p_2 \quad \cdots p_1 \quad A_{\text{class}} = c$ is used

where C is a class table

With this context for the given rule R,
confidence and support of R = percentage of tuples in D

satisfying the rule antecedent, with class lable C.

Associative classification follows following steps :

1. Find attribute_value pairs is the data after running it for frequent itemsets.

2. Generate association rules per class, satisfying confidence and support criteria.

3. Organize the rules and build a rule based classifier.

As mentioned above lets have look at different algorithms for associative classification.

1. CBA (Classification Based on Association) :

- It's a iterative approach of frequent itemset mining.
- Frequent itemsets are found after multiple passes.
- No. of passes made = length of longest rule obtained
- Uses heuristic method for construction of classifier.
- In case rules have same antecedent, highest confidence rule is selected.
- Decision list in formed for the set of rules forming a classifier.
- It shows more accuracy then C 4.5

2. CMAR (Classification Based on Multiple Association Rule)

- It uses variant of FP-growth algorithm for finding complete set of rules, satisfying minimum confidence and minimum support threshold.
- FP-tree is used to record all frequent itemset information is D in two scans.
- To find strongest group of rules it uses X^2 measure.
- Run time scalability and use of memory of CMAR is more efficient.

3. CPAR (Classification Based on Predictive Association Rules) :

- It uses classification called as FOIL, to build the rules to identify positive tuples from negative.
- In case of multi class problem, FOIL is applied to each class.
- Every time with generation of rule the positive tuples in data set are covered or removed.
- It uses best k rules of each group for prediction of class label of X.

3.5.2 Discriminative Frequent Pattern-Based Classification

- It considers combination of frequent patterns and single features for building a classification model.
- Frequent pattern based classification is based on learning of classification model in the feature space of single attribute as well as frequent patterns.
- The framework for discriminative frequent pattern based classification is :

1. Feature generation :

- ○ Partition data D according to class labels.
- ○ Find frequent patterns, which satisfy minimum support criteria by frequent itemset mining.
- ○ F is set of feature candidates from collection of frequent patterns.

2. Feature selection :

- ○ Generate F_S which is set of selected frequent patterns from applying feature selection to F.
- ○ Fisher score or other evaluation measures are used for this.
- ○ Transformation of D of D happens

  where, D = Single feature + selected frequent patterns F_s.

3. Learning of classification model :

- ○ Use any learning algorithm for building a classifier.

Two Marks Questions with Answers

Q.1 What is market basket analysis ?

Ans. :

- In market basket analysis, the customer buying habits are studied.
- As shown in Fig. 3.1.1, the association between different items placed in shopping baskets are studied. (For Fig. 3.1.1 refer section 3.1).
- It helps to understand the items which are frequently and together purchased by the customers.
- For ex. : If a customer is visiting supermarket what is the frequency of buying bread with milk.
- This analysis help shopkeepers to build marketing strategy and to manage the shelf space.
- Various strategies can be planned like design of stone layout or the items are kept together to encourage the combined sale.

Q.2 Explain how closed and max patterns are mined ?

Ans. :

- The approach is :
  - ○ To search for closed frequent itemsets directly by pruning the search space.
  - ○ Pruning strategies include :

    1. **Item merging :** It follows following rule :

If every transaction containing a frequent itemset X also contain itemset Y but not any proper subset of Y, then X U V forms a frequent closed itemset and there is no need to search for any itemset containing X but no Y.

2. **Sub itemset pruning :** It follows following rule :

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

3. **Item skipping : It follows the rule :**

In depth first mining of closed itemsets at each level, there will be a prefix itemset X associated with a header table and a projected database.

Q.3 How to mine multidimensional associations ?

Ans. : • The boolean association rule like below buys (X, "digital camera") buys (X, "HP printer") is a single dimensional association rule as only one predicate is involved..

- The rule which has two or more dimensions is known as multidimensional association rule.
  For ex. : age (X, "20…29") occupation (X, "student") buys (X, "laptop").
- In the above example all the predicates appear once such rules are known as interdimensional association rules.
- If the predicates appear more than once in the rule it is in called as hybrid-dimensional association rule.
  For ex. : age (X, "20…29") buys (X, "laptop") buys (X, "HP printers").

Q.4 Explain data cube method.

Ans. :
- In this numeric values are replaced by interval labels
- After applications of the suitable mining algorithm transformed multidimensional data is used to construct a data cube.
- n-predicate sets are stored in the cells of n-dimensional cuboid.
- Due to enhancement in data warehouse and OLAP technology it may happen that the dimensions which are of interest of user may already exists.
- In this case these values can be computed by low level aggregates and values are relieved.
- The structure of cuboid is shown in figure 3.1.9. (For Fig. 3.1.9 refer section 3.3.3).

Q.5 What is clustering based method ?

Ans. : Clustering Based method

- It considers the fact that frequent patterns are found in dense clusters.

- Top down approach can be applied as follows :
  - First standard clustering algorithm is applied to find clusters and satisfying minimum support threshold.
  - 2D spaces are examined by combining two clusters.
  - If this combination passes minimum support threshold, continue searching for the clusters.
- Bottom up approach can be applied as follows :
  - High dimension space is used to form clusters.
  - These clusters are then projected and merged.
  - This is not so feasible approach as finding high dimensional clustering is a difficult problem.

Q.6 What is constraint based frequent pattern mining ?

Ans. : Constraint-Based Frequent Pattern Mining

- In the process of data mining many rules may be formed which are sometimes not related to the user.
- If user specify intuitions or expectations in the form of constraints to generate useful rules it will be interesting.
- This technique in known as constraint based mining.
- Different types of constraints :

1. Knowledge type constraints : Considers knowledge to be mined like association, correlation, classification or clustering.

2. Data constraints : Specify set of task relevant data.

3. Dimension/level constraints : Specify desired dimensions

4. Interestingness constraints : Specify threshold on statistical measures like support, confidence and correlation.

5. Rule constraints : Specify conduction on rules to be mined.

6. Syntactic rule constraints : Can be specified in template structure by the technique metarule-guided mining.

Q.7 How to prun pattern space with pattern prunning constraints ?

Ans. : Pruning Pattern Space with Pattern Pruning Constraints

Constraints interact with pattern mining process by five ways

1) Anti monotonic :

- It follows the rule that if an itemset does not satisfy the rule constraint that even if we add the items in itemset it will make is more expensive, none of its suspects can satisfy the constraint.
- This is known as anti monotonic property

2) Monotonic :

- It follows the rule that if itemset satisfy the constraints that the addition of more items to itemset will increase the cost and satisfy the constraint, its supersets will also follow this rule.
- This is known as monotonic property.

3) Succinct :

- It follows the rule that the sets that are guaranteed to satisfy the constraints can be enumerated.

4) Convertible and inconvertible :

- The constraints which belongs to none of the above category they are convertible rules.

Q.8 What is CBA ?

Ans. : CBA (Classification Based on Association) :

- It's a iterative approach of frequent itemset mining.
- Frequent itemsets are found after multiple passes.
- No. of passes made = length of longest rule obtained
- Uses heuristic method for construction of classifier.
- In case rules have same antecedent, highest confidence rule is selected.
- Decision list in formed for the set of rules forming a classifier.
- It shows more accuracy then C 4.5

Q.9 What is CMAR ?

Ans. : CMAR (Classification Based on Multiple Association Rule) :

- It uses variant of FP-growth algorithm for finding complete set of rules, satisfying minimum confidence and minimum support threshold.
- FP-tree is used to record all frequent itemset information is D in two scans.
- To find strongest group of rules it uses X^2 measure
- Run time scalability and use of memory of CMAR is more efficient.

Q.10 What is CPAR ?

Ans. : CPAR (Classification Based on Predictive Association Rules) :

- It uses classification called as FOIL, to build the rules to identify positive tuples from negative.
- In case of multi class problem, FOIL is applied to each class.
- Every time with generation of rule the positive tuples in data set are covered or removed.
- It uses best k rules of each group for prediction of class label of X.

Q.11 What is discriminative frequent pattern based classification ?

Ans. : Discriminative Frequent Pattern-Based Classification

- It considers combination of frequent patterns and single features for building a classification model.
- Frequent pattern based classification is based on learning of classification model in the feature space of single attribute as well as frequent patterns.
- The framework for discriminative frequent pattern based classification is :

1. Feature generation :

○ Partition data D according to class labels.

○ Find frequent patterns, which satisfy minimum support criteria by frequent itemset mining.

○ F is set of feature candidates from collection of frequent patterns.

2. Feature selection :

○ Generate F_S which is set of selected frequent patterns from applying feature selection to F.

○ Fisher score or other evaluation measures are used for this.

○ Transformation of D of D happens

where, D = Single feature + selected frequent patterns F_s.

3. Learning of classification model :

○ Use any learning algorithm for building a classifier.

❑❑❑

Unit - IV

4 Classification and Clustering

Syllabus

Decision Tree Induction - Bayesian Classification – Rule Based Classification – Classification by Back Propagation – Support Vector Machines — Lazy Learners – Model Evaluation and Selection-Techniques to improve Classification Accuracy.

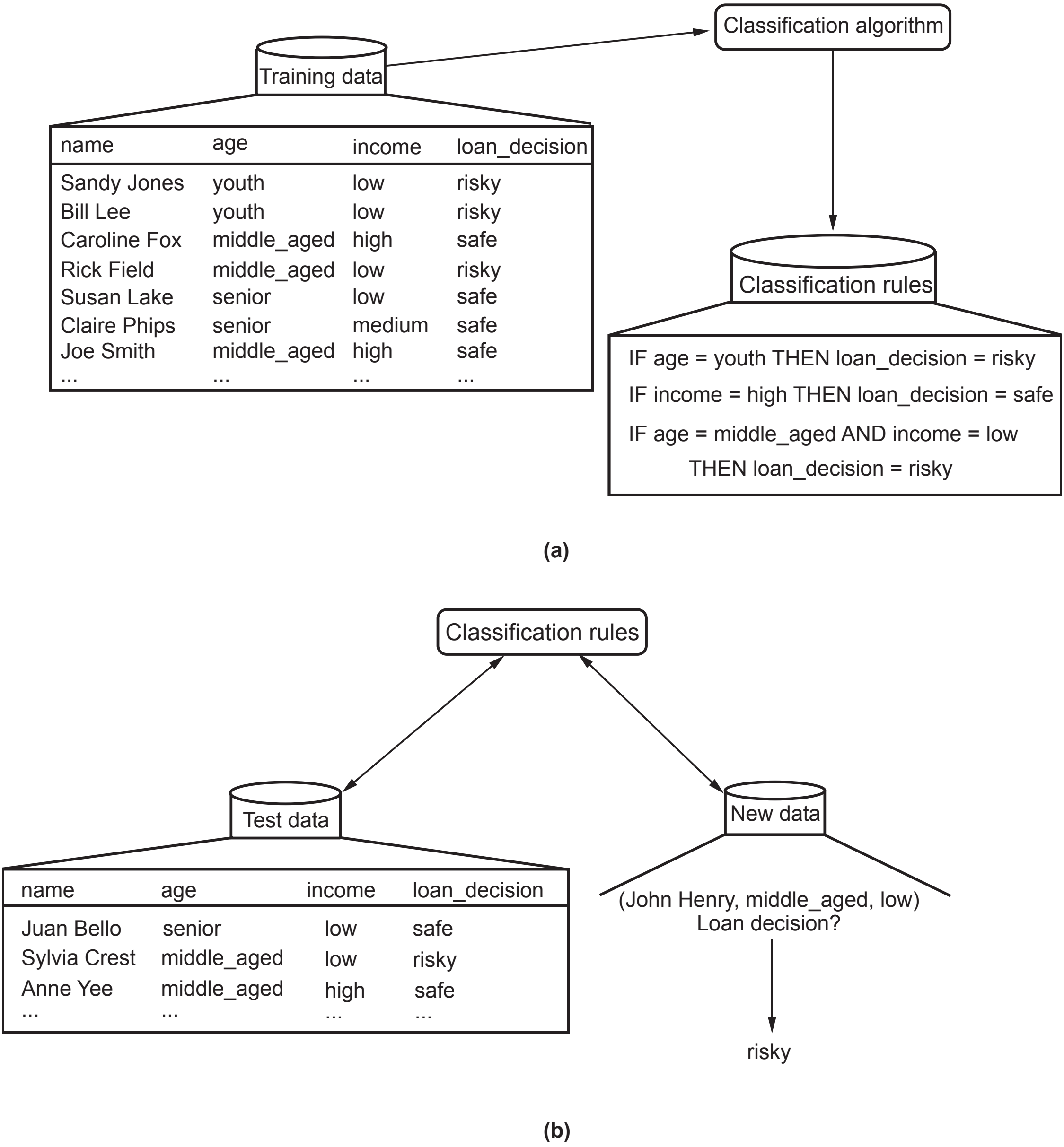
Contents

4.1 Classification

4.1.1 Introduction to Classification

- Definition
  - ○ Data classification is the process of organizing data into categories for its most effective and efficient use. A well-planned data classification system makes essential data easy to find and retrieve. Data classification is the process of sorting and categorizing data into various types, forms or any other distinct class.
  - ○ Data classification enables the separation and classification of data according to data set requirements for various business or personal objectives.
  - ○ Data Classification is a form of data analysis that extracts models describing important data classes. Such models, called classifiers, predict categorical (discrete, unordered) class labels.
- Data classification as a part of the Information Lifecycle Management (ILM) process can be defined as a tool for categorization of data to enable/help organizations to effectively answer the following questions :
  - ○ What data types are available ?
  - ○ Where are certain data located ?
  - ○ What access levels are implemented ?
  - ○ What protection level is implemented and does it adhere to compliance regulations ?
- For example,
  - ○ A classification model can be built to categorize bank loan applications as either safe or risky. Such analysis helps us for a better understanding of the data at large
  - ○ A medical researcher wants to analyse breast cancer data to predict which one of three
  - ○ Specific treatments a patient should receive.
  - ○ A marketing manager at Elexmart - an electronics outlet needs data analysis to help guess whether a customer with a given profile will buy a new laptop.
- Data Classification is mainly a data management process.
- Classification has many applications in various domains. These include fraud detection, target marketing, performance prediction, manufacturing, and medical diagnosis.
- There are many classification methods used in machine learning, pattern recognition, and statistics. These algorithms are designed assuming a small data size. Most of these algorithms are memory resident.

- With high data volumes in the recent days, researchers have put efforts in developing scalable classification and prediction techniques capable of handling large amounts of disk-resident data.
- How does classification work ?
  - ○ Data classification is a two-step process.
    - ■ **A learning step :** In this step a classification model is constructed.
    - ■ **A classification step :** In this step the model is used to predict class labels for given data.
  - ○ Learning step
    - ■ The learning step (or training phase), where a classification algorithm builds the classifier by analysing or "learning from" a training set made up of database tuples and their associated class labels.
    - ■ A classifier is built describing a predetermined set of data classes or concepts.
    - ■ This is the learning step (or training phase), where a classification algorithm builds the classifier by analyzing or "learning from" a training set made up of database tuples and their associated class labels.
    - ■ The learning process :
    - ■ A tuple, X, is represented by an n-dimensional attribute vector, X_D .x_1, x_2, ……, x_n, depicting n measurements made on the tuple from n database attributes, respectively, A_1, A_2, ….., A_n.
    - ■ Each tuple, X, is assumed to belong to a predefined class as determined by another database attribute called the class label attribute. The class label attribute is discrete-valued and unordered.
    - ■ It is categorical (or nominal) in that each value serves as a category or class.
    - ■ The individual tuples making up the training set are referred to as training tuples and are randomly sampled from the database under analysis.
    - ■ In the context of classification, data tuples can be referred to as samples, examples, instances, data points, or objects.
    - ■ If the class label of each training tuple is provided, this step is also known as **Supervised Learning**. If the class label of each training tuple is not known, and the number or set of classes to be learned may not be known in advance, it is called **Unsupervised Learning**.
    - ■ Refer Fig. 4.1.1 (a) - Training data are analyzed by a classification algorithm. Here, the class label attribute is loan decision, and the learned model or classifier is represented in the form of classification rules.



name	age	income	loan_decision
Sandy Jones	youth	low	risky
Bill Lee	youth	low	risky
Caroline Fox	middle_aged	high	safe
Rick Field	middle_aged	low	risky
Susan Lake	senior	low	safe
Claire Phips	senior	medium	safe
Joe Smith	middle_aged	high	safe
...	...	...	...

IF age = youth THEN loan_decision = risky

IF income = high THEN loan_decision = safe

IF age = middle_aged AND income = low
THEN loan_decision = risky

(a)

name	age	income	loan_decision
Juan Bello	senior	low	safe
Sylvia Crest	middle_aged	low	risky
Anne Yee	middle_aged	high	safe
...	...	...	...

(b)

Fig. 4.1.1 Learning and Classification

- **Classification step**
  - This first step of the classification process can also be viewed as the learning of a mapping or function, y = f(X) that can predict the associated class label y of a given tuple X.
  - Typically, this mapping is represented in the form of classification rules, decision trees, or mathematical formulae.

- In this step, the predictive accuracy of the classifier is estimated. For this purpose, a test set is used, made up of test tuples and their associated class labels. This test data is independent of the training tuples and not used for constructing the classifier tuple.
- The accuracy of a classifier on a given test set is the percentage of test set tuples that are correctly classified by the classifier.
- Refer Fig 4.1.1 (b) - Test data are used to estimate the accuracy of the classification rules. If the accuracy is considered acceptable, the rules can be applied to the classification of new data tuples.

4.1.2 Decision Tree Induction

- Decision tree induction is the learning of decision trees from class-labeled training tuples. A decision tree is a flowchart-like tree structure, where
  - Each internal node (non-leaf node) denotes a test on an attribute
  - Each branch represents an outcome of the test
  - Each leaf node (or terminal node) holds a class label
  - The topmost node in a tree is the root node
- Example - Refer Fig. 4.1.2 for a decision tree for the concept buys computer, indicating whether an Elexmart - an electronics outlet customer is likely to purchase a computer. Each internal (non-leaf) node represents a test on an attribute. Each leaf node represents a class (either buys computer = yes or buys computer = no).

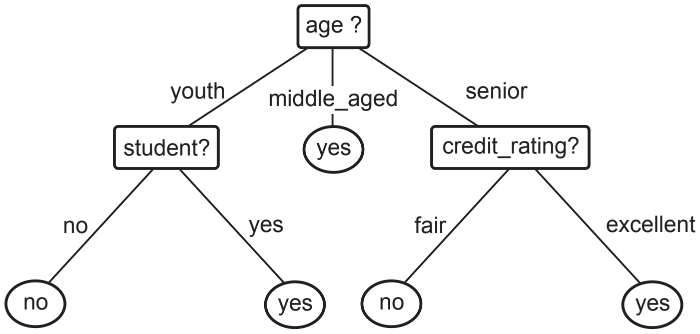


Fig. 4.1.2 A decision tree for Elexmart

- A decision tree at Fig. 4.1.2 represents the concept 'a customer buys computer'.

   - It predicts whether a customer at Elexmart is likely to purchase a computer. Internal nodes are denoted by rectangles, and leaf nodes are denoted by ovals.
   - Some decision tree algorithms produce only binary trees (where each internal node branches to exactly two other nodes), whereas others can produce nonbinary trees.
- How are decision trees used for classification ?
   - Given a tuple, X, for which the associated class label is unknown, the attribute values of the tuple are tested against the decision tree.
   - A path is traced from the root to a leaf node, which holds the class prediction for that tuple.
   - Decision trees can easily be converted to classification rules.
- Why are decision tree classifiers so popular ?
   - The construction of decision tree classifiers does not require any domain knowledge or parameter setting, and therefore is appropriate for exploratory knowledge discovery.
   - Decision trees can handle multidimensional data.
   - Their representation of acquired knowledge in tree form is intuitive and generally easy to assimilate by humans.
   - The learning and classification steps of decision tree induction are simple and fast.
   - In general, decision tree classifiers have good accuracy.
   - However, successful use may depend on the data at hand.
   - Decision tree induction algorithms have been used for classification in many application areas such as medicine, manufacturing and production, financial analysis, astronomy, and molecular biology.
   - Decision trees are the basis of several commercial rule induction systems.
- **Algorithm : Generate decision tree.**

  Generate a decision tree from the training tuples of data partition, D.

Input :

Data partition, D, which is a set of training tuples and their associated class labels;

attribute list, the set of candidate attributes;

Attribute selection method, a procedure to determine the splitting criterion that "best"

partitions the data tuples into individual classes. This criterion consists of a

splitting attribute and, possibly, either a *split-point* or *splitting subset*.

Output : A decision tree.

Method :

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

- Applying algorithm for **generating decision tree** for Elexmart case.
  - Let A be the splitting attribute. A has v distinct values, fa1, a2, ….. , avg, based on the training data.
  - There are three possible scenarios, as illustrated in Fig. (4.1.2), These three possibilities for partitioning tuples based on the splitting criterion, each with examples.
  - Let A be the splitting attribute.

    (a) If A is discrete-valued, then one branch is grown for each known value of A.

    (b) If A is continuous-valued, then two branches are grown, corresponding to A <= split point and A > split point.

(c) If A is discrete-valued and a binary tree must be produced, then the test is of the form A € S_A, where S_A is the splitting subset for A.

- The algorithm uses the same process recursively to form a decision tree for the tuples at each resulting partition, Dj , of D (step 14).
  - The recursive partitioning stops only when any one of the following terminating conditions is true :
  - All the tuples in partition D (represented at node N) belong to the same class (steps 2 and 3).
  - There are no remaining attributes on which the tuples may be further partitioned (step 4). In this case, majority voting is employed (step 5).
  - This involves converting node N into a leaf and labeling it with the most common class in D.
  - Alternatively, the class distribution of the node tuples may be stored.
  - There are no tuples for a given branch, that is, a partition Dj is empty (step 12).
  - In this case, a leaf is created with the majority class in D (step 13).
  - The resulting decision tree is returned (step 15).
- Tree Pruning
  - Tree pruning is performed in order to remove anomalies in the training data due to noise or outliers. The pruned trees are smaller and less complex.
  - Tree Pruning Approaches -There are two approaches to prune a tree ?
- Pre-pruning - The tree is pruned by halting its construction early.
- Post-pruning - This approach removes a sub-tree from a fully-grown tree.
- Cost Complexity
  - The cost complexity is measured by the following two parameters ?
    - Number of leaves in the tree, and
    - Error rate of the tree.
- Strengths of Decision Tree approach
  - Decision trees are able to generate understandable rules.
  - Decision trees perform classification without requiring much computation.
  - Decision trees are able to handle both continuous and categorical variables.
  - Decision trees provide a clear indication of which fields are most important for prediction or classification.
- Weaknesses of Decision Tree approach
  - Decision trees are less appropriate for estimation tasks where the goal is to predict the value of a continuous attribute.
  - Decision trees are prone to errors in classification problems with many class and relatively small number of training examples.

- Decision tree can be computationally expensive to train. The process of growing a decision tree is computationally expensive. At each node, each candidate splitting field must be sorted before its best split can be found. In some algorithms, combinations of fields are used and a search must be made for optimal combining weights. Pruning algorithms can also be expensive since many candidate sub-trees must be formed and compared.

4.1.3 Bayesian Classification

- Bayesian classification is based on Bayes' theorem.
- Bayesian classifiers are the statistical classifiers.
- Bayesian classifiers can predict class membership probabilities such as the probability that a given tuple belongs to a particular class.
- Bayesian classifiers results with high accuracy and speed when applied to large databases.
- Na¨Ive Bayesian classifiers assume that the effect of an attribute value on a given class is independent of the values of the other attributes.
- This assumption is called class conditional independence.
- It is made to simplify the computations involved and, in this sense, is considered "na¨Ive."

- **Baye's Theorem**
  - Bayes' Theorem is named after Thomas Bayes.
  - There are two types of probabilities -
    - Posterior Probability [P(H/X)]
    - Prior Probability [P(H)]

    where X is data tuple and H is some hypothesis.
- According to Bayes' Theorem,

$$\boxed{P(H/X) = P(X/H)P(H) / P(X)}$$

- **Na¨Ive Bayesian classifiers -**
  - Naive Bayes is a simple, yet effective and commonly-used, learning classifier.
  - It is a probabilistic classifier that makes classifications using the Maximum A Posteriori decision rule in a Bayesian setting.
  - Na¨ive Bayesian classifiers assume that the effect of an attribute value on a given class is independent of the values of the other attributes. This assumption is called class conditional independence.
  - It is made to simplify the computations involved and, in this sense, is considered "na¨Ive."

- It can also be represented using a very simple Bayesian network. Naive Bayes classifiers have been especially popular for text classification, and are a traditional solution for problems such as spam detection.
- Naive Bayes is a simple technique for constructing classifiers: models that assign class labels to problem instances, represented as vectors of feature values, where the class labels are drawn from some finite set.
- There is not a single algorithm for training such classifiers, but a family of algorithms based on a common principle : all naive Bayes classifiers assume that the value of a particular feature is independent of the value of any other feature, given the class variable.
- For example, an object may be considered to be a ball if it is red, round, and about 7 cm in diameter.
- A naive Bayes classifier considers each of these features to contribute independently to the probability that this object is a ball , regardless of any possible correlations between the color, roundness, and diameter features.
- Naive Bayes is a conditional probability model: given a problem instance to be classified, represented by a vector

$$x = (x_1, ..., x_n)$$

representing some n features (independent variables), it assigns to this instance probabilities

$$p(C_k | x_1, ..., x_n)$$

for each of K possible outcomes or classes C_k.

- The problem with the above formulation is that if the number of features n is large or if a feature can take on a large number of values, then basing such a model on probability tables is infeasible.
- The model is reformulated to make it more tractable.
- Using Bayes' theorem, the conditional probability can be decomposed as

$$p(C_k | x) = \frac{p(C_k)\, p(x | C_k)}{p(x)}$$

- In plain English, using Bayesian probability terminology, the above equation can be written as

$$\text{Posterior} = \frac{\text{Prior} \times \text{Likelihood}}{\text{Evidence}}$$

- Effectiveness of Bayesian classifiers -
  - Various empirical studies of this classifier in comparison to decision tree and neural network classifiers have found it to be comparable in some domains.

  - In theory, Bayesian classifiers have the minimum error rate in comparison to all other classifiers.
  - In practice this is not always the case. The inaccuracies are due to assumptions made for its use, such as class-conditional independence, and the lack of available probability data.
  - Bayesian classifiers are also useful in theoretical justification for other classifiers that do not explicitly use Bayes' theorem.
  - For example, under certain assumptions, many neural network and curve-fitting algorithm output the maximum posteriori hypothesis, like the na¨Ive Bayesian classifier.

4.1.4 Rule Based Classification

- Rule-based classifier makes use of a set of IF-THEN rules for classification. We can express a rule in the following from -

  IF condition THEN conclusion

- Let us consider a rule R,

  R: IF age > 30 AND credit = good

  THEN buy_computer = yes

- Rule R can also be written as as follows -

  R1: (age = youth) ^ (student = yes))(buys computer = yes)

- Points to remember while applying rule -
  - The IF part of the rule is called rule antecedent or precondition.
  - The THEN part of the rule is called rule consequent.
  - The antecedent part the condition consist of one or more attribute tests and these tests are logically ANDed.
  - The consequent part consists of class prediction.
- Rule Extraction from a Decision Tree
  - Rule Extraction is to build a rule-based classifier by extracting IF-THEN rules from a decision tree.
  - To extract a rule from a decision tree -
    - One rule is created for each path from the root to the leaf node.
    - To form a rule antecedent, each splitting criterion is logically ANDed.
    - The leaf node holds the class prediction, forming the rule consequent.
  - Rule Induction Using Sequential Covering Algorithm
    - Sequential Covering Algorithm can be used to extract IF-THEN rules form the training data.
    - In this algorithm, each rule for a given class covers many of the tuples of that class.

    - No decision tree is generated.
  - As per the general strategy the rules are learned one at a time.
  - For each time rules are learned, a tuple covered by the rule is removed and the process continues for the rest of the tuples.
  - This is because the path to each leaf in a decision tree corresponds to a rule.
  - The Decision tree induction can be considered as learning a set of rules simultaneously.
- Algorithm for Sequential Covering
  - The Following is the sequential learning Algorithm where rules are learned for one class at a time.
  - While learning a rule from a class Ci, a rule should cover all the tuples from class C only and no tuple form any other class.

```
Algorithm : Sequential Covering
Input :
D, a data set class-labeled tuples,
Att_vals, the set of all attributes and their possible values.
Output :  A Set of IF-THEN rules.
Method :
Rule_set={ }; // initial set of rules learned is empty
for each class c do
   repeat
      Rule = Learn_One_Rule(D, Att_valls, c);
      remove tuples covered by Rule form D;
   until termination condition;
   Rule_set=Rule_set+Rule; // add a new rule to rule-set
end for
return Rule_Set;
```

- Rule Pruning - The rule is pruned is due to the following reason -
  - The Assessment of quality is made on the original set of training data. The rule may perform well on training data but less well on subsequent data. That's why the rule pruning is required.
  - The rule is pruned by removing conjunct. The rule R is pruned, if pruned version of R has greater quality than what was assessed on an independent set of tuples.

- ○ FOIL(First, Outer, Inner, Last) is one of the simple and effective method for rule pruning. For a given rule R,
- ○ FOIL_Prune = pos - neg / pos + neg
- ○ where pos and neg is the number of positive tuples covered by R, respectively.
- ○ This value will increase with the accuracy of R on the pruning set. Hence, if the FOIL_Prune value is higher for the pruned version of R, then we prune R.

4.1.5 Classification by Backpropogation

- The BackPropagation (BP) algorithm learns the classification model by training a multilayer feed-forward neural network.
- The generic architecture of the neural network for BP is shown in the following diagrams, with one input layer, some hidden layers, and one output layer.
- Each layer contains some units or perceptron. Each unit might be linked to others by weighted connections.
- The values of the weights are initialized before the training.
- The number of units in each layer, number of hidden layers, and the connections will be empirically defined at the very start.
- The back propagation algorithm performs learning on amultilayer fee-forward neural network (Refer Fig. 4.1.3).

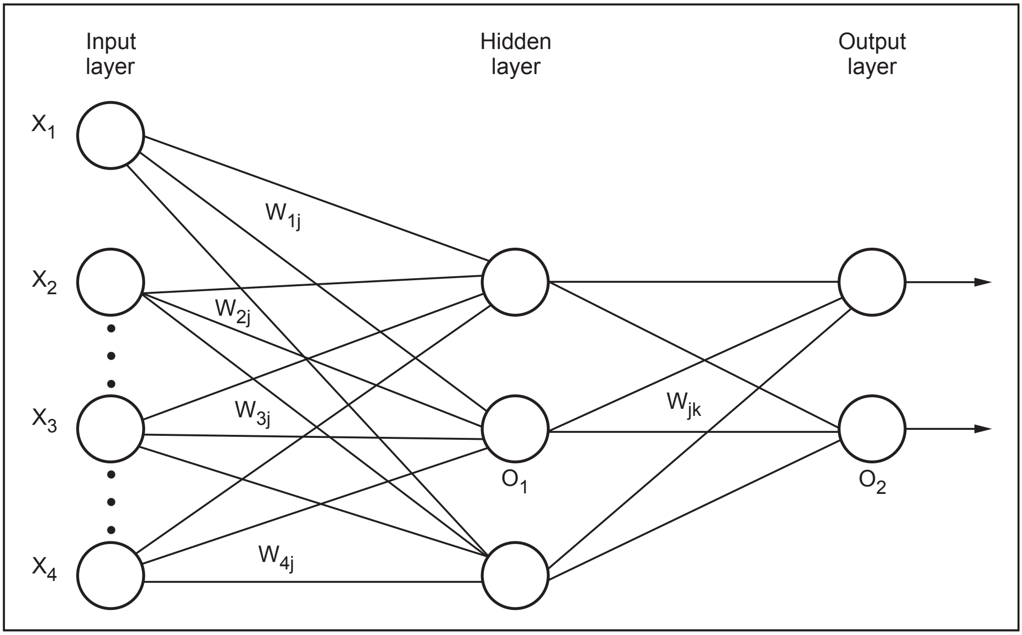


Fig. 4.1.3 Backpropagation using Multilayer feedforward Network

- The inputs correspond to the attributes measured for each raining sample. The inputs are fed simultaneously into layer of units making up the input layer.
- The weighted outputs of these units are, in turn, fed simultaneously to a second layer of neuron like units, known as a hidden layer.
- The hidden layer s weighted outputs can be input to another hidden layer, and so on. The number of hidden layers is arbitrary, although in practice, usually only one is used.
- The weighted outputs of the last hidden layer are input to units making up the output layer, which emits the network's prediction for given samples.
- The units in the hidden layers and output layer are sometimes referred to as neurodes, due to their symbolic biological basis, or as output units.
- Multilayer feed-forward networks of linear threshold functions, given enough hidden units, can closely approximate any function.
- Backpropagation
  - Back propagation learns by iteratively processing a set of training samples, comparing the network's prediction for each sample with the actual known class label.
  - For each training sample, the weights are modified so as to minimize the mean squared error between the network's prediction and the actual class.
  - These modifications are made in the "backwards" direction, that is , form the output layer through each hidden layer down to the first hidden layer (hence the name backpropagation).
  - Although it is not guaranteed in general the weights will eventually converge, and the learning process stops.

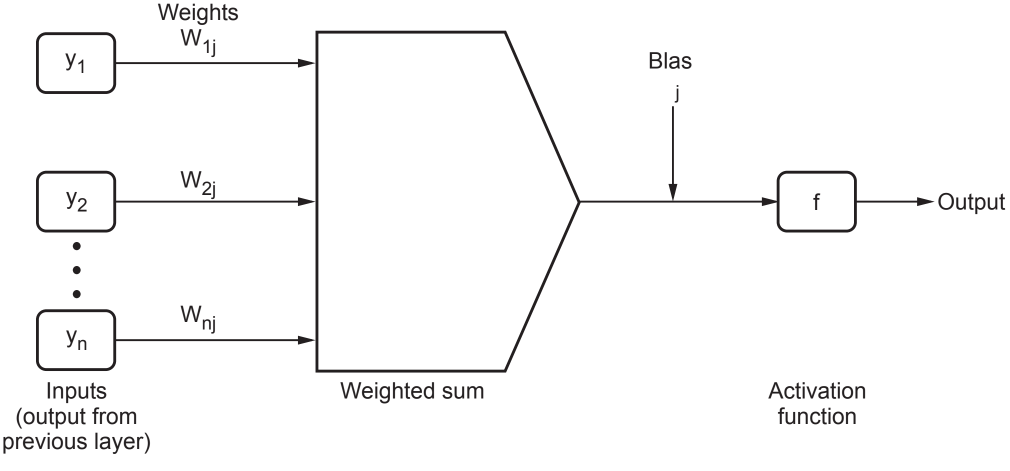


Fig. 4.1.4 Back propagation

- The Backpropagation algorithm

  Initialize the weights.

  The weights in the network are initialized to small random number(e.g., ranging from – 1.0 to 1.0, or – 0.5 to 0.5).

  Initialize the biases to small random numbers.

  For Each training sample: X, is processed by the following steps.

    Perform Feed-forward computation to reduce error

    Back propagation to the output layer

    Back propagation to the hidden layer

    Weights are adjusted / updated

  The algorithm is stopped when the value of the error function has become sufficiently small.

4.1.6 Support Vector Machines

- The Support-Vector Machines (SVMs) algorithm, created by Hava Siegelmann and Vladimir Vapnik, applies the statistics of support vectors, developed in the support vector machines algorithm, to categorize unlabeled data,
- Support Vector Machines (SVMs), is one of the most widely used clustering algorithms in industrial applications.
- Support Vector Machines (SVMs) is a discriminative classifier formally defined by a separating hyperplane.
- In other words, given labeled training data (supervised learning), the algorithm outputs an optimal hyperplane which categorizes new examples.
- In two-dimensional space this hyperplane is a line dividing a plane in two parts where in each class lay in either side.
- Support Vector Machines (SVMs), a method for the classification of both linear and nonlinear data.
- Support Vector Machines (SVMs) is mostly used in classification problems.
- In this algorithm, we plot each data item as a point in n-dimensional space (where n is number of features you have) with the value of each feature being the value of a particular coordinate.
- Then, we perform classification by finding the hyper-plane that differentiate the two classes very well (Refer Fig. 4.1.5).

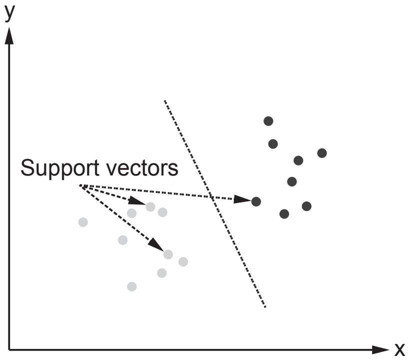


Fig. 4.1.5 Support Vector Machines

- SVM searches for the hyperplane with the largest margin, that is, the Maximum Marginal Hyperplane (MMH). The associated margin gives the largest separation between classes.
- Support Vectors are simply the co-orinates of individual observation. Support Vector segregates the two classes (hyper-plane/ line).
- An SVM is an algorithm that works as follows.
    - SVM uses a nonlinear mapping to transform the original training data into a higher dimension.
    - Within this new dimension, it searches for the linear optimal separating hyperplane (i.e., a "decision boundary" separating the tuples of one class from another).
    - With an appropriate nonlinear mapping to a sufficiently high dimension, data from two classes can always be separated by a hyperplane.
    - The SVM finds this hyperplane using support vectors ("essential" training tuples) and margins (defined by the support vectors).
- In addition to performing linear classification, SVMs can efficiently perform a non-linear classification using what is called the kernel trick, implicitly mapping their inputs into high-dimensional feature spaces.
- Parameter selection - The effectiveness of SVM depends on the selection of kernel, the kernel's parameters, and soft margin parameter C
- SVMs can be used to solve various real-world problems :
    - SVMs are helpful in text and hypertext categorization, as their application can significantly reduce the need for labeled training instances in both the standard inductive and transductive settings.
    - Classification of images can also be performed using SVMs. SVM is also used in image segmentation
    - Hand-written characters can be recognized using SVM.

  - The SVM algorithm has been widely applied in the biological and other sciences.
- Drawbacks of Support Vector Machines (SVMs)
  - It requires full labeling of input data
  - It has uncalibrated class membership probabilities-SVM avoids estimating probabilities on finite data
  - The SVM is only directly applicable for two-class tasks. Therefore, algorithms that reduce the multi-class task to several binary problems have to be applied
  - In SVMs, the parameters of a solved model are difficult to interpret.
- The SVMs are extended to achieve higher classification results. These are
  - Support-Vector Clustering (SVC)
    SVC is a similar method that also builds on kernel functions but is appropriate for unsupervised learning. It is considered a fundamental method in data science.
  - Multiclass SVM
    Multiclass SVM aims to assign labels to instances by using support-vector machines, where the labels are drawn from a finite set of several elements.

4.1.7 Lazy Learners (Learning from Your Neighbors)

Understanding Eager Learners

- The classification methods discussed in the earlier sections are examples of eager learners.
- Eager learners, when given a set of training tuples, construct a generalization (i.e., classification) model before receiving new (e.g., test) tuples to classify
- Examples of Eager Learners are
  - Decision tree induction,
  - Bayesian classification,
  - Rule-based classification
  - Classification by backpropagation,
  - Support vector machines
  - Classification based on association rule mining

Understanding Lazy Learners

- A lazy learner delays abstracting from the data until it is asked to make a prediction
- Lazy Learners
- Simply Stores the training data without doing any further processing on it, till it gets the next test set.

- It's slow as it calculates based on the current data set instead of coming up with an algorithm based on historic data
- It has large localized data so generalization takes time at every iteration
- Lazy learners do less work when a training tuple is presented and more work when making a classification or numeric prediction.
- Because lazy learners store the training tuples or "instances," they are also referred to as instance-based learners, even though all learning is essentially based on instances.
- While performing classification or numeric prediction, lazy learners can be computationally expensive.
- They require efficient storage techniques and are well suited to implementation on parallel hardware.
- They offer little explanation or insight into the data's structure. Lazy learners, however, naturally support incremental learning.
- They are able to model complex decision spaces having hyperpolygonal shapes that may not be as easily describable by other learning algorithms
- Examples of lazy learners :
  - K-nearest-neighbor classifiers
  - Case-based reasoning classifiers

Key differences in Eager Learners and Lazy Learners are

Aspect	Eager Learner	Lazy Learner
Technique	Eager learning (eg. Decision trees, SVM, NN) : Given a set of training set, constructs a classification model before receiving new (e.g., test) data to classify	Lazy learning (e.g., instance-based learning) : Simply stores training data (or only minor processing) and waits until it is given a test tuple
Accuracy	Eager : must commit to a single hypothesis that covers the entire instance space	Lazy method effectively uses a richer hypothesis space since it uses many local linear functions to form its implicit global approximation to the target function
Approach	1 k-nearest neighbour approach-Instances represented as points in a Euclidean space. 2 Locally weighted regression - Constructs local approximation	Store training examples and delay the processing ("lazy evaluation") until a new instance must be classified

- **k-Nearest-Neighbor Classifiers (k-NN)**
  - The k-nearest-neighbor method was first described in the early 1950s.
  - This method is labor intensive when given large training sets.
  - It has since been widely used in the area of pattern recognition.
  - k-NN is a type of instance-based learning, or lazy learning, where the function is only approximated locally and all computation is deferred until classification.
  - Nearest-neighbor classifiers are based on learning by analogy, that is, by comparing a given test tuple with training tuples that are similar to it. The training tuples are described by n attributes. Each tuple represents a point in an n-dimensional space.
  - Thus, all the training tuples are stored in an n-dimensional pattern space.
  - When given an unknown tuple, a k-nearest-neighbor classifier searches the pattern space for the k training tuples that are closest to the unknown tuple. These k training tuples are the k "nearest neighbors" of the unknown tuple.
  - "Closeness" is defined in terms of a distance metric, such as Euclidean distance.
  - The Euclidean distance between two points or tuples $X_1 = (x_{11}, x_{12}, :::, x_{1n})$ and $X_2 = (x_{21}, x_{22}, :::, x_{2n})$ is

$$\text{dist}(X_1, X_2) = \sqrt{\sum_{i=1}^{n}(x_{1i} - x_{2i})^2}$$

  - A peculiarity of the k-NN algorithm is that it is sensitive to the local structure of the data.
  - In k-NN classification, the output is a class membership. An object is classified by a plurality vote of its neighbours, with the object being assigned to the class most common among its k nearest neighbours (k is a positive integer, typically small). If k = 1, then the object is simply assigned to the class of that single nearest neighbour
  - The naive version of the algorithm is easy to implement by computing the distances from the test example to all stored examples, but it is computationally intensive for large training sets.
  - Using an approximate nearest neighbour search algorithm makes k-NN computationally tractable even for large data sets. Many nearest neighbour search algorithms have been proposed over the years; these generally seek to reduce the number of distance evaluations actually performed.
  - The techniques used to speed up classification time are
    - Partial distance calculations and editing the stored tuples.

- In the partial distance method, we compute the distance based on a subset of the n attributes. If this distance exceeds a threshold, then further computation for the given stored tuple is halted, and the process moves on to the next stored tuple.
- The editing method removes training tuples that prove useless.
- This method is also referred to as pruning or condensing because it reduces the total number of tuples stored.

- **Case-based reasoning classifiers**
  - Case-Based Reasoning (CBR) classifiers use a database of problem solutions to solve a new problems.
  - CBR stores the tuples or "cases" for problem solving as complex symbolic descriptions.
  - When given a new case to classify, a case-based reasoner will first check if an identical training case exists.
  - If one is found, then the accompanying solution to that case is returned.
  - If no identical case is found, then the case-based reasoner will search for training cases having components that are similar to those of the new case.
  - Conceptually, these training cases may be considered as neighbors of the new case. If cases are represented as graphs, this involves searching for subgraphs that are similar to subgraphs within the new case.
  - The case-based reasoner tries to combine the solutions of the neighboring training cases to propose a solution for the new case.
  - If incompatibilities arise with the individual solutions, then backtracking to search for other solutions may be necessary.
  - The case-based reasoner may employ background knowledge and problem-solving strategies to propose a feasible combined solution.
  - Business applications of CBR include problem resolution for customer service help desks, where cases describe product-related diagnostic problems.
  - CBR has also been applied to areas such as engineering and law, where cases are either technical designs or legal rulings, respectively.
  - Medical education is another area for CBR, where patient case histories and treatments are used to help diagnose and treat new patients.
  - **Limitations of Case Based Reasoning**
  - Challenges in case-based reasoning include finding a good similarity metric (e.g., for matching subgraphs) and suitable methods for combining solutions.

- ○ Other challenges include the selection of salient features for indexing training cases and the development of efficient indexing techniques.
- ○ A trade-off between accuracy and efficiency evolves as the number of stored cases becomes very large.

4.1.8 Model Evaluation and Selection

- Evaluating a model is a core part of building an effective machine learning model
- There are several evaluation metrics, like confusion matrix, cross-validation, AUC-ROC curve, etc.
- Different evaluation metrics are used for different kinds of problems.

1) Confusion Matrix

- A confusion matrix shows the number of correct and incorrect predictions made by the classification model compared to the actual outcomes (target value) in the data.
- The matrix is N x N, where N is the number of target values (classes).
- Performance of such models is commonly evaluated using the data in the matrix.
- The following table displays a 2 x 2 confusion matrix for two classes (Positive and Negative).

Confusion Matrix		**Target**			
		Positive	Negative		
Model	Positive	a	b	Positive Predictive Value	a/(a+b)
	Negative	c	d	Negative Predictive Value	d/(c+d)
		Sensitivity	Specificity	**Accuracy** = (a+d)/(a+b+c+d)	
		a/(a+c)	d/(b+d)		

Fig. 4.1.6 Confusion Matrix

Where

- **Accuracy :** The proportion of the total number of predictions that were correct.
- Positive predictive **Value or Precision :** The proportion of positive cases that were correctly identified.
- **Negative predictive value :** The proportion of negative cases that were correctly identified.
- **Sensitivity or Recall :** The proportion of actual positive cases which are correctly identified.
- **Specificity :** The proportion of actual negative cases which are correctly identified.

Here is an example

Confusion Matrix		**Target**			
		Positive	Negative		
Model	Positive	70	20	Positive Predictive Value	0.78
	Negative	30	80	Negative Predictive Value	0.73
		Sensitivity	Specificity	**Accuracy** = 0.75	
		0.70	0.80		

Fig. 4.1.7 Confusion Matrix Example

2) Gain or Lift Charts

- Gain or lift is a measure of the effectiveness of a classification model calculated as the ratio between the results obtained with and without the model.
- Gain and lift charts are visual aids for evaluating performance of classification models.
- However, in contrast to the confusion matrix that evaluates models on the whole population gain or lift chart evaluates model performance in a portion of the population.
- Gain and Lift chart are mainly concerned to check the rank ordering of the probabilities. Here are the steps to build a Lift/Gain chart :

1) Calculate probability for each observation

2) Rank these probabilities in decreasing order.

3) Build deciles with each group having almost 10 % of the observations.

4) Calculate the response rate at each deciles for Good (Responders), Bad (Non-responders) and total.

You will get following table from which you need to plot Gain/Lift charts :

Score Table

Target	Score
0	235
1	724
1	556
0	345
0	480

⇒

Sorted by Score

Target	Score
1	880
1	724
1	676
1	556
0	480

⇒

Gain Table

Count %	Target %
10	36
20	54
30	66
40	76
50	85

⇒

Lift Table

Count %	Lift
10	3.6
20	2.7
30	2.2
40	1.9
50	1.7

1	676
0	195
1	880
0	368
…	…

0	368
0	345
0	235
0	195
…	…

60	90
70	94
80	98
90	100
100	100

60	1.5
70	1.3
80	1.2
90	1.1
100	1

Fig 4.1.8 Sample data for Gain/Lift Chart

The gain chart formed is

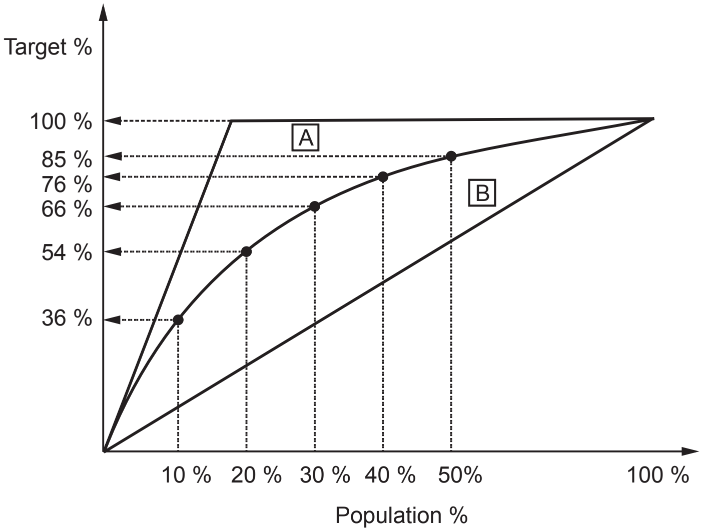


Fig. 4.1.9 Gain Chart

The lift charts

- The lift chart shows how much more likely we are to receive positive responses than if we contact a random sample of customers.
- For example, by contacting only 10 % of customers based on the predictive model we will reach 3 times as many respondents, as if we use no model.

The lift chart formed with above table data is

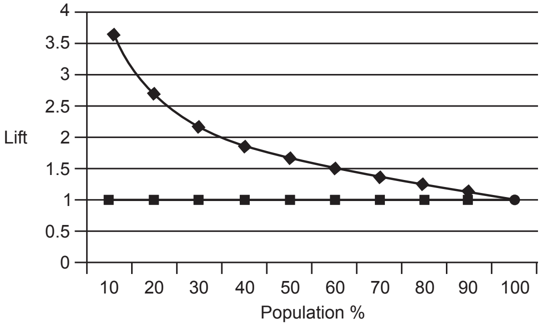


Fig. 4.1.10 Lift Chart

3) K-S Chart

- K-S or Kolmogorov-Smirnov chart measures performance of classification models.
- More accurately, K-S is a measure of the degree of separation between the positive and negative distributions.
- The K-S is 100 if the scores partition the population into two separate groups in which one group contains all the positives and the other all the negatives.
- On the other hand, If the model cannot differentiate between positives and negatives, then it is as if the model selects cases randomly from the population.
- The K-S would be 0. In most classification models the K-S will fall between 0 and 100, and that the higher the value the better the model is at separating the positive from negative cases.

Example :

The following example shows the results from a classification model. The model assigns a score between 0-1000 to each positive (Target) and negative (Non-Target) outcome.

Source Range		Count		Cummulative Count			
Lower	upper	Targrt	Non-Target	Target	Non-Target	K-S	
0	100	3	62	0.5 %	0.8 %	0.3 %	
100	200	0	23	0.5 %	1.1 %	0.6 %	
200	300	1	66	0.7 %	2.0 %	1.3 %	
300	400	7	434	2.0 %	7.7 %	5.7 %	
400	500	181	5627	34.3 %	81.7 %	47.4 %	K-S
500	600	112	886	54.3 %	98.3 %	39.0 %	
600	700	83	332	69.1 %	97.7 %	28.6 %	
700	800	45	63	77.1 %	98.5 %	21.4 %	
800	900	29	37	82.3 %	99.0 %	16.7 %	
900	1000	99	77	100.0 %	100.0 %	0.0 %	

K(0.95) = 6.0 %
K(0.99) = 7.1 %

Fig. 4.1.11 Cumulative count in K-S Chart

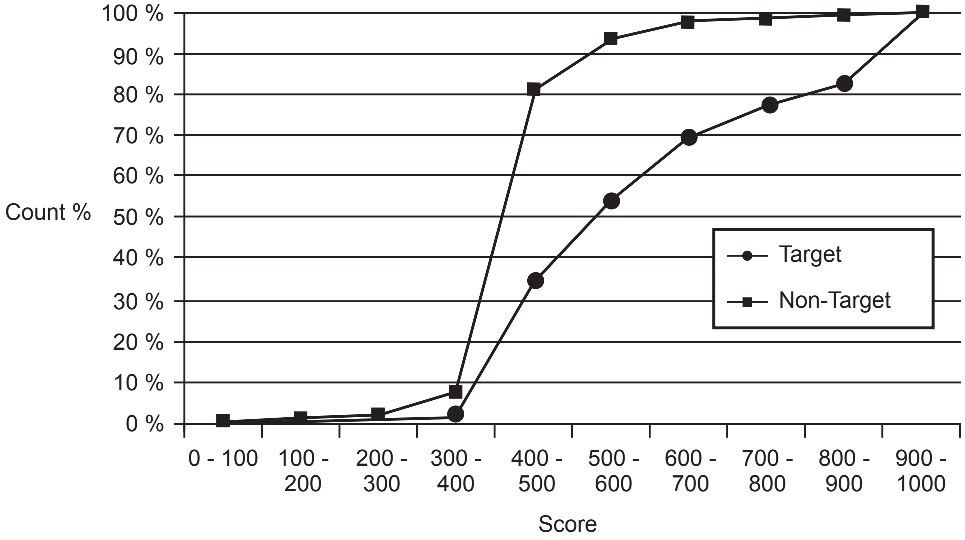


Fig 4.1.12 K-S Chart

4) ROC Chart

- The ROC chart is similar to the gain or lift charts in that they provide a means of comparison between classification models.
- The ROC chart shows false positive rate (1-specificity) on X-axis, the probability of target=1 when its true value is 0, against true positive rate (sensitivity) on Y-axis, the probability of target=1 when its true value is 1.
- Ideally, the curve will climb quickly toward the top-left meaning the model correctly predicted the cases.
- The diagonal red line is for a random model (ROC101).

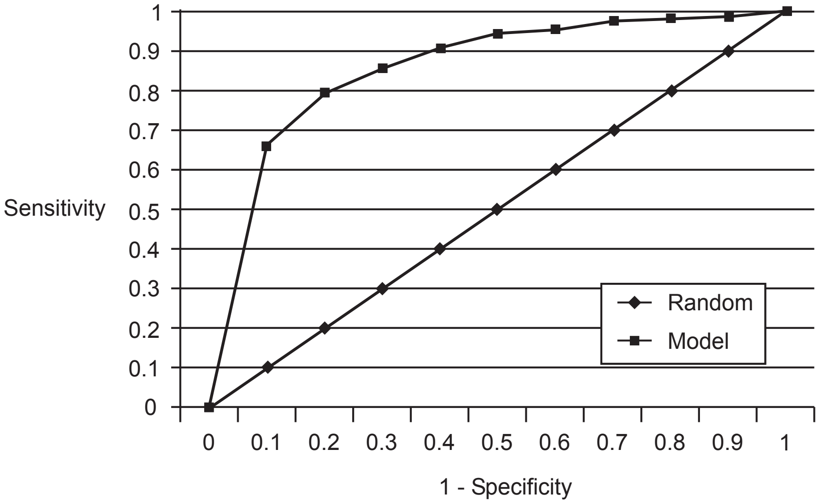


Fig 4.1.13 ROC chart

5) Area Under the Curve (AUC)

- Area under ROC curve is often used as a measure of quality of the classification models.
- A random classifier has an area under the curve of 0.5, while AUC for a perfect classifier is equal to 1.
- In practice, most of the classification models have an AUC between 0.5 and 1.

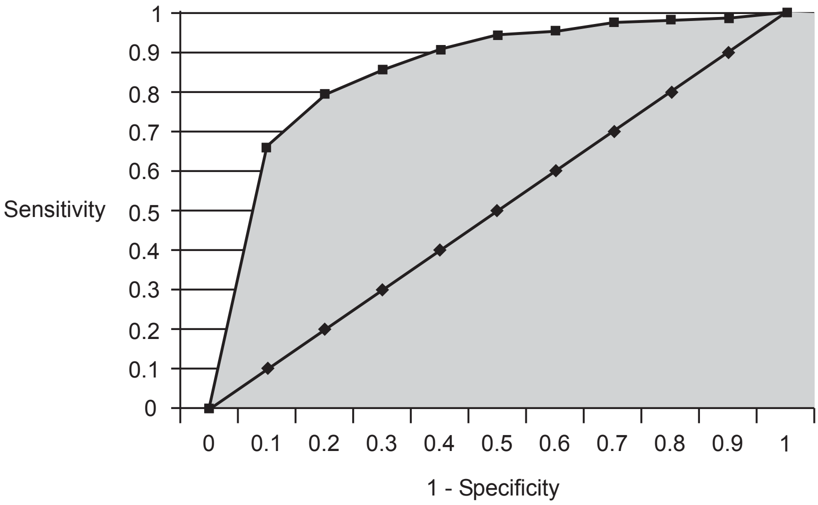


Fig 4.1.14 AUC Coverage

- An area under the ROC curve of 0.8, for example, means that a randomly selected case from the group with the target equals 1 has a score larger than that for a randomly chosen case from the group with the target equals 0 in 80 % of the time.
- When a classifier cannot distinguish between the two groups, the area will be equal to 0.5 (the ROC curve will coincide with the diagonal).
- When there is a perfect separation of the two groups, i.e., no overlapping of the distributions, the area under the ROC curve reaches to 1 (the ROC curve will reach the upper left corner of the plot).

4.1.9 Technique to Improve Classification Accuracy

- An ensemble method for classification is a composite model, made up of a combination of classifiers.
- The individual classifiers vote, and a class label prediction is returned by the ensemble based on the collection of votes.
- Ensembles tend to be more accurate than their component classifiers.

- The model development cycle goes through various stages, starting from data collection to model building.
- Here are the basic techniques to improve classification accuracy.

1) Add more data

- Having more data is always a good idea.
- It allows the "data to tell for itself," instead of relying on assumptions and weak correlations.
- Presence of more data results in better and accurate models.
- Increasing the size of data reduces pain of working on limited data sets and making assumptions on the input data

2) Treat missing and outlier values

- The unwanted presence of missing and outlier values in the training data often reduces the accuracy of a model or leads to a biased model.
- It leads to inaccurate predictions. This is because we don't analyse the behavior and relationship with other variables correctly.
- So, it is important to treat missing and outlier values well.
- Refer the Fig. 4.1.15. It shows that, in presence of missing values, the chances of playing cricket by females is similar as males. But, if you look at the second table (after treatment of missing values based on salutation of name, "Miss"), we can see that females have higher chances of playing cricket compared to males.

With missing Values

Name	Weight	Gender	Play Cricket / Not
Mr. Amit	55	M	Y
Mr. Anil	61	M	Y
Miss Swati	58	F	N
Miss Richa	55		Y
Mr. Steve	55	M	N
Miss Reena	64	F	Y
Miss Rashmi	57		Y
Mr. Kunal	57	M	N

After imputation of missing values

Name	Weight	Gender	Play Cricket / Not
Mr. Amit	58	M	Y
Mr. Anil	61	M	Y
Miss Swati	58	F	N
Miss Richa	55	F	Y
Mr. Steve	55	M	N
Miss Reena	64	F	Y
Miss Rashmi	57	F	Y
Mr. Kunal	57	M	N

Gender	#Students	#Play Cricket	%Play Cricket
F	2	1	50 %
M	4	2	50 %
Missing	2	2	100 %

Gender	#Students	#Play Cricket	%Play Cricket
F	4	3	75 %
M	4	2	50 %

Fig. 4.1.15 Example for treating missing and Outlier values

- There are various methods to deal with missing and outlier values :

  i) Missing :

  ○ In case of continuous variables, you can impute the missing values with mean, median, mode.

  ○ For categorical variables, you can treat variables as a separate class. You can also build a model to predict the missing values.

  ○ KNN imputation offers a great option to deal with missing values. To know more about these methods refer article "Methods to deal and treat missing values".

  ii) Outlier :

  ○ One can delete the observations, perform transformation, binning, Imputation (Same as missing values) or can also treat outlier values separately.

3) Feature engineering

- This step helps to extract more information from existing data. New information is extracted in terms of new features.
- These features may have a higher ability to explain the variance in the training data. Thus, giving improved model accuracy.
- Feature engineering is highly influenced by hypotheses generation. Good hypothesis result in good features. It is recommended to invest quality time in hypothesis generation.
- Feature engineering process can be divided into two steps :

  i) Feature transformation : There are various scenarios where feature transformation is required :

  ○ Changing the scale of a variable from original scale to scale between zero and one. This is known as data normalization. For example: If a data set has 1st variable in meter, 2nd in centi-meter and 3rd in kilo-meter, in such case, before applying any algorithm, we must normalize these variable in same scale.

  ○ Some algorithms works well with normally distributed data. Therefore, we must remove skewness of variable(s). There are methods like log, square root or inverse of the values to remove skewness.

- Some times, creating bins of numeric data works well, since it handles the outlier values also. Numeric data can be made discrete by grouping values into bins. This is known as data discretization.

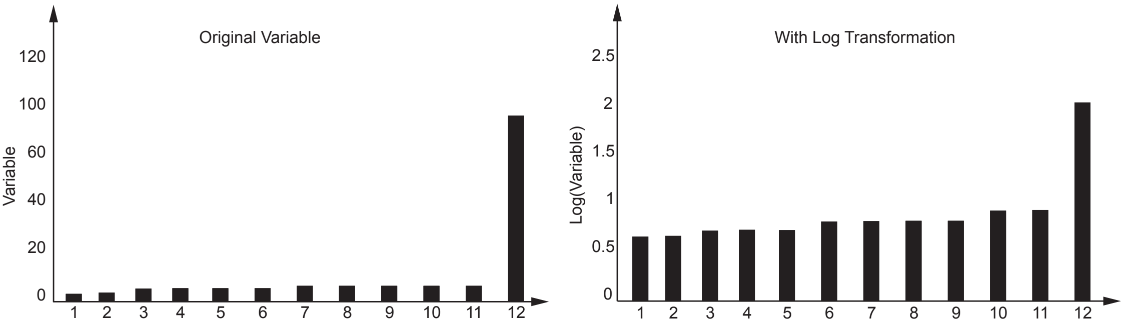


Fig 4.1.16 Example of feature engineering

ii) Feature Creation : Deriving new variable(s) from existing variables is known as feature creation.

- It helps to unleash the hidden relationship of a data set.
- Let's say, we want to predict the number of transactions in a store based on transaction dates. Here transaction dates may not have direct correlation with number of transaction, but if we look at the day of a week, it may have a higher correlation.
- In this case, the information about day of a week is hidden. We need to extract it to make the model better.

4) Feature selection

- Feature Selection is a process of finding out the best subset of attributes which better explains the relationship of independent variables with target variable.
- One can select the useful features based on various metrics like :
  - Domain Knowledge : Based on domain experience, we select feature(s) which may have higher impact on target variable.
  - Visualization : As the name suggests, it helps to visualize the relationship between variables, which makes your variable selection process easier.
  - box-plot
  - Statistical Parameters : We also consider the p-values, information values and other statistical metrics to select right features.
    - PCA : It helps to represent training data into lower dimensional spaces, but still characterize the inherent relationships in the data. It is a type of dimensionality reduction technique. There are various methods to reduce the

dimensions (features) of training data like factor analysis, low variance, higher correlation, backward/ forward feature selection and others.

5) Multiple algorithms

- Mapping to accurate algorithm is the ideal approach to achieve higher accuracy. But, it is difficult to achieve
- This intuition comes with experience and incessant practice. Some algorithms are better suited to a particular type of data sets than others. Hence, we should apply all relevant models and check the performance.

6) Algorithm tuning

- The classification algorithms are driven by parameters. These parameters majorly influence the outcome of learning process.
- The objective of parameter tuning is to find the optimum value for each parameter to improve the accuracy of the model.
- To tune these parameters, one must have a good understanding of these meaning and their individual impact on model. One can repeat this process with a number of well performing models.

7) Ensemble methods

- This is the most common approach found majorly in winning solutions of Data science competitions.
- This technique simply combines the result of multiple weak models and produce better results. This can be achieved through many ways :
  - Bagging (Bootstrap Aggregating)
  - Boosting
- It is always a better idea to apply ensemble methods to improve the accuracy of your model. Because
  - They are generally more complex than traditional methods
  - The traditional methods give you a good base level from which one can improve and draw from to create your ensembles.

8) Cross validation :

- To find the right answer of this question, we must use **cross validation** technique.
- Cross validation is one of the most important concepts in data modeling. It says, try to leave a sample on which you do not train the model and test the model on this sample before finalizing the model.
- This method helps us to achieve more generalized relationships.

4.2 Clustering Techniques

4.2.1 Basic Concepts

- Clustering is the process of grouping a set of data objects into multiple groups or clusters so that objects within a cluster have high similarity, but are very dissimilar to objects in other clusters.
- Dissimilarities and similarities are assessed based on the attribute values describing the objects and often involve distance measures.
- Generally, a group of abstract objects into classes of similar objects is made.
- A cluster of data objects as one group.
- While doing cluster analysis, data is partitioned into groups. This partitioning is based on data similarity and then assign the labels to the groups.
- The main advantage of over-classification is that it is adaptable to changes. And helps single out useful features that distinguish different groups.
- Clustering as a data mining tool has its roots in many application areas such as biology, security, business intelligence, and Web search.

4.2.2 Cluster Analysis

- **Cluster analysis** or **Simply clustering** is the process of partitioning a set of data objects (or observations) into subsets.
  - ○ Each subset is a cluster, such that objects in a cluster are similar to one another, yet dissimilar to objects in other clusters.
  - ○ The set of clusters resulting from a cluster analysis can be referred to as a clustering.
- Different clustering methods may generate different clustering(s) on the same data set.
- The partitioning is not performed by humans, but by the clustering algorithm. Hence, clustering is useful in that it can lead to the discovery of previously unknown groups within the data.

Applications of clustering

- Data Clustering analysis is used in many applications. Such as market research, pattern recognition, data analysis, and image processing.
- Data Clustering can also help marketers discover distinct groups in their customer base. And they can characterize their customer groups based on the purchasing patterns.

- In the field of biology, it can be used to derive plant and animal taxonomies. categorize genes with similar functionalities and gain insight into structures inherent to populations.
- Clustering in Data Mining helps in identification of areas e.g. geographic locations. It also helps in the identification of groups of houses in a city. That is according to house type, value, and geographic location.
- Clustering in Data Mining also helps in classifying documents on the web for information discovery
- Also, we use Data clustering in outlier detection applications. Such as detection of credit card fraud.
- As a data mining function, cluster analysis serves as a tool. That is to gain insight into the distribution of data. Also, need to observe characteristics of each cluster.
- Clustering is also called data segmentation in some applications because clustering partitions large data sets into groups according to their similarity.
- Clustering is also used for outlier detection, where outliers (values that are "far away" from any cluster) may be more interesting than common cases.
- Applications of outlier detection include the detection of credit card fraud and the monitoring of criminal activities in electronic commerce.
- As a branch of statistics, cluster analysis has been extensively studied, with the main focus on distance-based cluster analysis.
- Cluster analysis tools are based on k-means, k-medoids, and several other statistical methods. These are used in building statistical analysis software packages or systems, such as S-Plus, SPSS, and SAS.
- Clustering is known as unsupervised learning because the class label information is not present. For this reason, clustering is a form of learning by observation, rather than learning by examples.
- In data mining, the focus is on data in large datasets. Hence focus is on finding methods for efficient and effective cluster analysis in large databases.
- Active themes of research focus on the scalability of clustering methods in large databases, the effectiveness of
  - Methods for clustering complex shapes (e.g., nonconvex)
  - Types of data (e.g., text, graphs, and images),
  - High-dimensional clustering techniques (e.g., clustering objects with thousands of features)
  - Methods for clustering mixed numerical and nominal data.

Requirements of Clustering in Data Mining

- Clustering algorithms have several requirements. These factors include scalability and the ability to deal with different types of attributes, noisy data, incremental updates, clusters of arbitrary shape, and constraints.
- Interpretability and usability are also important. In addition, clustering methods can differ with respect to the partitioning level, whether or not clusters are mutually exclusive, the similarity measures used, and whether or not subspace clustering is performed.
- The following points specify the requirement of clustering in Data Mining :
  - **Scalability -** Highly scalable clustering algorithms are required to deal with large databases.
  - **Ability to deal with different kinds of attributes -** Algorithms should be capable to be applied to any kind of data. Such as interval-based data, categorical, and binary data.
  - **Discovery of clusters with attribute shape -** The clustering algorithm should be capable of detecting clusters of arbitrary shape. They should not be bounded by only distance measures. That tends to find a spherical cluster of small sizes.
  - **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.
  - **High dimensionality -** The clustering algorithm should not only be able to handle low-dimensional data. Although, need to handle 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.
  - **Incremental clustering and insensitivity to input order -** In many applications, incremental updates (representing newer data) may arrive at any time. Clustering algorithms may also be sensitive to the input data order. Hence in Datamining applications, Incremental clustering algorithms and input order insensitive algorithms are needed.
  - **Capability of clustering high-dimensionality data :** A data set can contain numerous dimensions or attributes
  - **Constraint-based clustering :** Real-world applications may need to perform clustering under various kinds of constraints.
  - **Interpretability and usability :** Users want clustering results to be interpretable, comprehensible, and usable. That is, clustering may need to be tied in with specific semantic interpretations and applications.

- **Clustering Methods**
  - Clustering methods can be classified in the following types

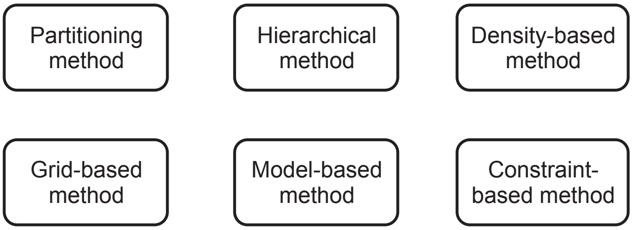


Fig. 4.2.1 Types of clustering methods

Orthogonal Aspect of Clustering methods

- **The partitioning criteria :** In some methods, all the objects are partitioned so that no hierarchy exists among the clusters. That is, all the clusters are at the same level conceptually. Such a method is useful, for example, for partitioning customers into groups so that each group has its own manager.
- **Separation of clusters :** Some methods partition data objects into mutually exclusive clusters. When clustering customers into groups so that each group is taken care of by one manager, each customer may belong to only one group
- **Similarity measure :** Some methods determine the similarity between two objects by the distance between them. Such a distance can be defined on Euclidean space, a road network, a vector space, or any other space.
- **Clustering space :** Many clustering methods search for clusters within the entire given data space. These methods are useful for low-dimensionality data sets.

4.2.3 Partitioning Methods

- The simplest and most fundamental version of cluster analysis is partitioning.
- For given a data set, D, of n objects, and k, the number of clusters to form, a partitioning algorithm organizes the objects into k partitions (k<n) where each partition represents a cluster.
- The clusters are formed to optimize an objective partitioning criterion, such as a dissimilarity function based on distance, so that the objects within a cluster are "similar" to one another and "dissimilar" to objects in other clusters in terms of the data set attributes
- Partitioning methods conduct one-level partitioning on data sets.
- The basic partitioning methods typically adopt exclusive cluster separation. That is, each object must belong to exactly one group.

- Most partitioning methods are distance-based. Given k, the number of partitions to construct, a partitioning method creates an initial partitioning.
- It then uses an iterative relocation technique that attempts to improve the partitioning by moving objects from one group to another.
- The general criterion of a good partitioning is that objects in the same cluster are "close" or related to each other, whereas objects in different clusters are "far apart" or very different.
- Traditional partitioning methods can be extended for subspace clustering, rather than searching the full data space. This is useful when there are many attributes and the data are sparse.
- Greedy approaches like the k-means and the k-medoids algorithms, progressively improve the clustering quality and approach a local optimum.
- **K-Means : A Centroid-Based Technique**
- Suppose a data set, D, contains n objects in Euclidean space. Partitioning methods distribute the objects in D into k clusters, $C_1, ..., C_k$ that is, $C_i \subset D$ and $C_i \cap C_j = \phi$ of $((1 \le i, \ j \le k)$
- An objective function is used to assess the partitioning quality so that objects within a cluster are similar to one another but dissimilar to objects in other clusters.
- A centroid-based partitioning technique uses the centroid of a cluster, C_i to represent that cluster. Conceptually, the centroid of a cluster is its center point.
- The main objective of the K-Means algorithm is to minimize the sum of distances between the points and their respective cluster centroid.
- The quality of cluster C_i can be measured by the within cluster variation, which is the sum of squared error between all objects in C_i and the centroid c_i, defined as

$$E = \sum_{i=1}^{k} \sum_{p \in C_i}^{k} \text{dist}\,(p, c_i)^2,$$

- Where E is the sum of the squared error for all objects in the data set; p is the point in space representing a given object; and ci is the centroid of cluster C_i

How K-means algorithm works

- The k-means algorithm defines the centroid of a cluster as the mean value of the points within the cluster. It proceeds as follows.
- First, it randomly selects k of the objects in D, each of which initially represents a cluster mean or center.

- For each of the remaining objects, an object is assigned to the cluster to which it is the most similar, based on the Euclidean distance between the object and the cluster mean.
- The k-means algorithm then iteratively improves the within-cluster variation.
- For each cluster, it computes the new mean using the objects assigned to the cluster in the previous iteration.
- All the objects are then reassigned using the updated means as the new cluster centers.
- The iterations continue until the assignment is stable, that is, the clusters formed in the current round are the same as those formed in the previous round.

K-means partitioning algorithm

Algorithm : k-means. The k-means algorithm for partitioning, where each cluster's center is represented by the mean value of the objects in the cluster.

Input :

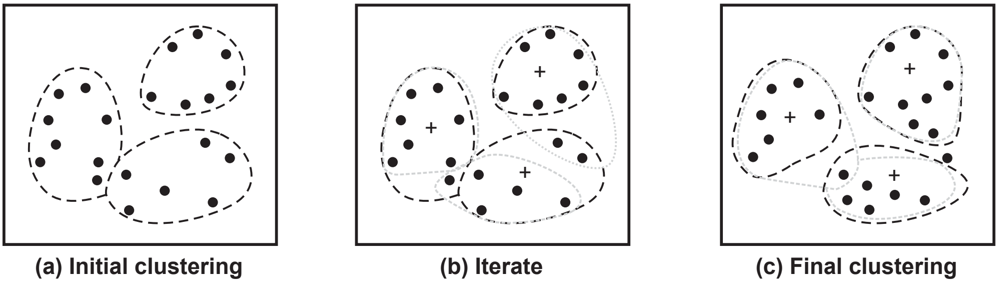
- K : the number of clusters,
- D : a data set containing n objects.

Output : A set ofk clusters.

Method :

```
(1) arbitary choose k objects from D as the initial cluster centers;
(2) repeat
(3) (re)assign each object to the cluster to which the object is most similar,
based on the mean value of the objects in the cluster;
(4) update the cluster means, that is, calculate the mean value of the objets for
each cluster;
(5) until no change;
```

Example of K-means partitioning algorithm



(a) Initial clustering

(b) Iterate

(c) Final clustering

Fig. 4.2.2 K-means Partitioning Example

- Clustering of a set of objects using the k-means method; for (b) update cluster centers and reassign objects accordingly (the mean of each cluster is marked by a C).

K-Medoids : A Representative Object-Based Technique

- The k-means algorithm is sensitive to outliers because such objects are far away from the majority of the data.
- Thus, when assigned to a cluster, they can dramatically distort the mean value of the cluster.
- This inadvertently affects the assignment of other objects to clusters. This is primarily due to due to the use of the squared-error function in k-means algorithm
- The modified k-means algorithm picks actual objects to represent the clusters. This is K-medoids method
- The partitioning method is then performed based on the principle of minimizing the sum of the dissimilarities between each object **p** and its corresponding representative object.
- An absolute-error criterion is used, defined as

$$E = \sum_{i=1}^{k} \sum_{p \in C_i} \text{dist}(p, o_i)$$

Where, **E** is the sum of the absolute error for all objects p in the data set,

oi is the representative object of Ci.

- This is the basis for the k-medoids method, which groups n objects into k clusters by minimizing the absolute error.
- When k = 1, we can find the exact median in $O(n^2)$ time.
- However, when k is a general positive number, the k-medoid problem is NP-hard.

The Partitioning Around Medoids (PAM) algorithm

- It is a popular realization of k-medoids clustering. It tackles the problem in an iterative, greedy way.
- The initial representative objects (called seeds) are chosen arbitrarily.
- All the possible replacements are tried out. The iterative process of replacing representative objects by other objects continues until the quality of the resulting clustering cannot be improved by any replacement.
- This quality is measured by a cost function of the average dissimilarity between an object and the representative object of its cluster.
- PAM, a k-medoids partitioning algorithm is given below.

Algorithm : k-medoids. PAM, a k-medoids algorithm for partitioning based on medoid or central objects.

Input :

- k : the number of clusters,
- D : a data set contatining n objects.

Output : A set of k clusters.

Method :

```
(1) arbitarily choose k objects in D as the initial representative objects or seeds;
(2) repeat
(3)     assign each remaining object to the cluster with the nearest representative object;
(4)     randomly select a nonrepresentstive object, $o_{random}$;
(5)     Compute the total cost, S, of swapping representative object, $o_j$, with $o_{random}$;
(6)     if S<0 then swap $o_j$ with $o_{random}$ to form the new set of k representative objects;
(7) until no change;
```

- A typical k-medoids partitioning algorithm like PAM works effectively for small data sets, but not scalable for large data sets.
- For larger data sets, a sampling-based method called CLARA (Clustering LARge Applications) can be used.

4.2.4 Hierarchical Methods

- While partitioning methods meet the basic clustering requirement of organizing a set of objects into a number of exclusive groups, in some situations we may want to partition our data into groups at different levels such as in a hierarchy.
- A hierarchical clustering method works by grouping data objects into a hierarchy or "tree" of clusters.
- Representing data objects in the form of a hierarchy is useful for data summarization and visualization.
- A hierarchical method creates a hierarchical decomposition of the given set of data objects.
- A hierarchical method can be classified as being either agglomerative or divisive, based on how the hierarchical decomposition is formed.
- Hierarchical clustering methods can be distance-based or density- and continuity based.

- Various extensions of hierarchical methods consider clustering in subspaces as well.
- Hierarchical methods suffer from the fact that once a step (merge or split) is done, it can never be undone.
- This rigidity is useful in that it leads to smaller computation costs by not having to worry about a combinatorial number of different choices.
- **The agglomerative approach -**
  - It is also called the bottom-up approach, starts with each object forming a separate group.
  - It successively merges the objects or groups close to one another, until all the groups are merged into one (the topmost level of the hierarchy), or a termination condition holds.
  - The agglomerative algorithm It typically starts by letting each object form its own cluster and iteratively merges clusters into larger and larger clusters, until all the objects are in a single cluster or certain termination conditions are satisfied.
  - The single cluster becomes the hierarchy's root.
  - For the merging step, it finds the two clusters that are closest to each other (according to some similarity measure), and combines the two to form one cluster.
  - Because two clusters are merged per iteration, where each cluster contains at least one object, an agglomerative method requires at most n iterations.
- **The divisive approach -**
  - It is also called the top-down approach, starts with all the objects in the same cluster.
  - In each successive iteration, a cluster is split into smaller clusters, until eventually each object is in one cluster, or a termination condition holds.
  - The divisive approach employs a top-down strategy. It starts by placing all objects in one cluster, which is the hierarchy's root.
  - It then divides the root cluster into several smaller sub-clusters, and recursively partitions those clusters into smaller ones.
  - The partitioning process continues until each cluster at the lowest level is coherent enough-either containing only one object, or the objects within a cluster are sufficiently similar to each other.
  - In either agglomerative or divisive hierarchical clustering, a user can specify the desired number of clusters as a termination condition.
  - Whether using an agglomerative method or a divisive method, a core need is to measure the distance between two clusters, where each cluster is generally a set of objects.

- Four widely used measures (known as linkage measures) for distance between clusters are given in Fig. 4.2.3 where

  |p-p'| is the distance between two objects or points, p and p';

  m_i is the mean for cluster, C_i;

  n_i is the number of objects in C_i.

Minimum distance : $\text{dist}_{\min}(C_i, C_j) = \min_{p\in C_i, p'\in C_j} \{|p-p'|\}$

Maximum distance : $\text{dist}_{\max}(C_i, C_j) = \max_{p\in C_i, p'\in C_j} \{|p-p'|\}$

Mean distance : $\text{dist}_{\text{mean}}(C_i, C_j) = |m_i - m_j|$

Average distance : $\text{dist}_{\text{avg}}(C_i, C_j) = \frac{1}{n_i, n_j} \sum_{p\in C_i,\ p'\in C_j} |p-p'|$

Fig 4.2.3 Linkage measures

- An agglomerative hierarchical clustering algorithm that uses the minimum distance measure is also called a minimal spanning tree algorithm.
- A spanning tree of a graph is a tree that connects all vertices, and a minimal spanning tree is the one with the least sum of edge weights.
- When an algorithm uses the minimum distance, $d_{\min}(C_i, C_j)$ to measure the distance between clusters, it is sometimes called a **nearest-neighbor clustering algorithm**.
- if the clustering process is terminated when the distance between nearest clusters exceeds a user-defined threshold, it is called a **single-linkage algorithm**.
- When an algorithm uses the maximum distance, $d_{\max}(C_i, C_j)$, to measure the distance between clusters, it is sometimes called a **farthest-neighbor clustering algorithm.**

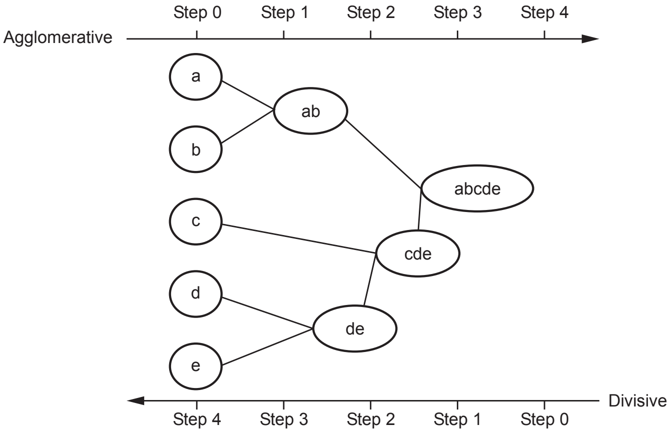


Fig 4.2.4 Agglomerative and divisive hierarchical clustering on data objects a,b, c,d, e

BIRCH : Multiphase Hierarchical Clustering (Using Clustering Feature Trees)

- Balanced Iterative Reducing and Clustering using Hierarchies (BIRCH) is designed for clustering a large amount of numeric data by integrating hierarchical clustering (at the initial micro-clustering stage) and other clustering methods such as iterative partitioning (at the later macro-clustering stage).
- It overcomes the two difficulties in agglomerative clustering methods :
  - Scalability
  - The inability to undo what was done in the previous step.
- BIRCH uses the notions of clustering feature to summarize a cluster, and clustering feature tree (CF-tree) to represent a cluster hierarchy.
- An advantage of BIRCH is its ability to incrementally and dynamically cluster incoming, multi-dimensional metric data points in an attempt to produce the best quality clustering for a given set of resources (memory and time constraints).
- In most cases, BIRCH only requires a single scan of the database.
- Its inventors claim BIRCH to be the "first clustering algorithm proposed in the database area to handle 'noise' (data points that are not part of the underlying pattern) effectively".
- The primary phases of BRICH are
  - BIRCH scans the database to build an initial in-memory CF-tree, which can be viewed as a multilevel compression of the data that tries to preserve the data's inherent clustering structure.
  - BIRCH applies a (selected) clustering algorithm to cluster the leaf nodes of the CF-tree, which removes sparse clusters as outliers and groups dense clusters into larger ones.

Chameleon : Multiphase Hierarchical Clustering Using Dynamic Modelling

- Chameleon is a hierarchical clustering algorithm that uses dynamic modeling to determine the similarity between pairs of clusters.
- In Chameleon, cluster similarity is assessed based on
  - How well connected objects are within a cluster
  - The proximity of clusters.
- Two clusters are merged if their interconnectivity is high and they are close together.
- Thus, Chameleon does not depend on a static, user-supplied model and can automatically adapt to the internal characteristics of the clusters being merged.
- The merge process facilitates the discovery of natural and homogeneous clusters and applies to all data types as long as a similarity function can be specified.

- Figure 4.2.5 shows how Chameleon works.

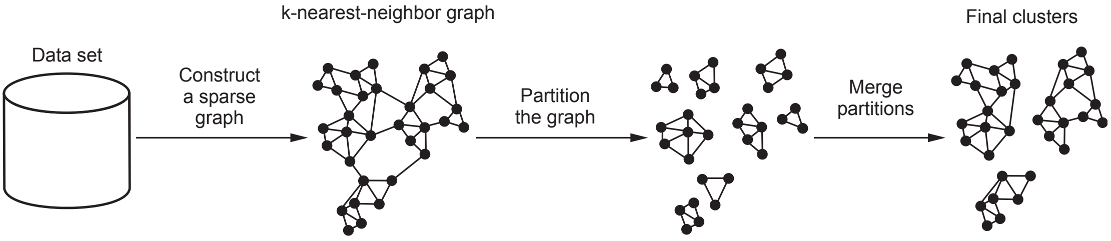


Fig 4.2.5 Chameleon: hierarchical clustering based on k-nearest neighbors and dynamic modeling.

- Chameleon uses a k-nearest-neighbor graph approach to construct a sparse graph, where each vertex of the graph represents a data object, and there exists an edge between two vertices (objects) if one object is among the k-most similar objects to the other.
- The edges are weighted to reflect the similarity between objects.
- Chameleon uses a graph partitioning algorithm to partition the k-nearest-neighbor graph into a large number of relatively small sub-clusters such that it minimizes the edge cut.

Probabilistic Hierarchical Clustering

- Probabilistic hierarchical clustering aims to overcome disadvantages of earlier models by using probabilistic models to measure distances between clusters.
- A probabilistic hierarchical clustering method can adopt the agglomerative clustering framework, but use probabilistic models to measure the distance between clusters.
- A probabilistic hierarchical clustering scheme can start with one cluster per object, and merge two clusters, C_i and C_j , if the distance between them is negative.
- Probabilistic hierarchical clustering methods are easy to understand, and generally have the same efficiency as algorithmic agglomerative hierarchical clustering methods;in fact, they share the same framework.
- Probabilistic models are more interpretable, but sometimes less flexible than distance metrics. Probabilistic models can handle partially observed data.
- A drawback of using probabilistic hierarchical clustering is that it outputs only one hierarchy with respect to a chosen probabilistic model. It cannot handle the uncertainty of cluster hierarchies
- A probabilistic hierarchical clustering algorithm is as follows

Algorithm : A probabilistic hierarchical clustering algorithm.

Input :

- $D = \{o_1, \ldots\ldots, o_n\}$: a data set containing n objects.

Output : A hierarchy of clusters.

Method :

```
1) Create a clster for each object $C_i = \{o_i\}, 1 \le i \le n$;
2) for i = 1 to n
3)     find pair of clusters $C_i$ and $C_j$ such that $C_i, C_j = \arg\max_{i \neq j} \log \frac{P(C_i \cup C_j)}{P(C_i)P(C_j)}$;
4)     If log $\log \frac{P(C_i \cup C_j)}{P(C_i)P(C_j)} > 0$ hen merge $C_i$ and $C_j$;
5)     else step;
```

4.2.5 Density based Methods

- Partitioning and hierarchical methods are designed to find spherical-shaped clusters.
- They have difficulty finding clusters of arbitrary shape such as the "S" shape and oval Clusters
- Most partitioning methods cluster objects based on the distance between objects. Such methods can find only spherical-shaped clusters and encounter difficulty in discovering clusters of arbitrary shapes.
- Other clustering methods have been developed based on the notion of density.
- Their general idea is to continue growing a given cluster as long as the density (number of objects or data points) in the "neighborhood" exceeds some threshold.
- For example, for each data point within a given cluster, the neighborhood of a given radius has to contain at least a minimum number of points.
- Such a method can be used to filter out noise or outliers and discover clusters of arbitrary shape.
- Density-based methods can divide a set of objects into multiple exclusive clusters, or a hierarchy of clusters.
- Typically, density-based methods consider exclusive clusters only, and do not consider fuzzy clusters. Moreover, density-based methods can be extended from full space to subspace clustering.

DBSCAN (Density-Based Spatial Clustering of Applications with Noise)

- It finds core objects, that is, objects that have dense neighborhoods. It connects core objects and their neighborhoods to form dense regions as clusters.
- A user-specified parameter $\varepsilon > 0$ is used to specify the radius of a neighborhood we consider for every object.
- The ε - neighborhood of an object o is the space within a radius ε centered at o.
- Due to the fixed neighborhood size parameterized by ε, the density of a neighborhood can be measured simply by the number of objects in the neighborhood. To determine whether a neighborhood is dense or not, DBSCAN uses another user-specified parameter, MinPts, which specifies the density threshold of dense regions.
- An object is a core object if the ε - neighborhood of the object contains at least MinPts objects. Core objects are the pillars of dense regions.
- The DBSCAN algorithm is as follows

Algorithm : DBSCAN : a density-based clustering algorithm.

Input :

- D : a data set containing n objects
- $\in$: the radius parameter, and
- Minpts : the neighborhood density threshold.

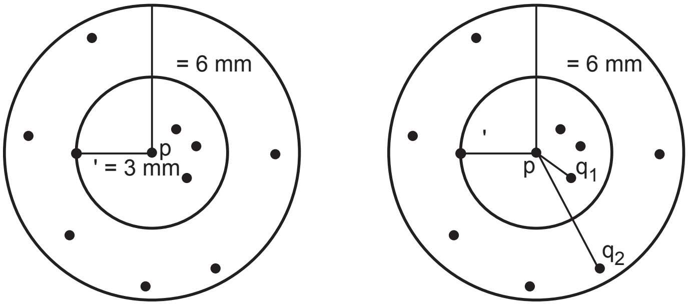
Output : A set of density-based clusters.

Method :

```
mark all objcets as unvisited ;
do
  randomly select an unvisited object p ;
  mark P as visited;
  if the ∈ - neighborhood of p has at least Minpts objects.
        create a new cluster C and add p to C;
        let N be the set of objects in the ε-neighborhood of p;
        for cach point p′ in N.
            if p′ is unvisited
                mark p′ as visited ;
                if the ε-neighborhood of p′ has at least Minpts points. Add
                those points of N;
            if p′ is not yet a member of any cluster, add p′ to C;
        end for
        output C;
else mark p as noise;
until no object is unvisited;
```

OPTICS : Ordering Points to Identify the Clustering Structure

- Although DBSCAN can cluster objects given input parameters such as (the maximum radius of a neighborhood) and MinPts (the minimum number of points required in the neighborhood of a core object), it hampers users with the responsibility of selecting parameter values that will lead to the discovery of acceptable clusters
- Most algorithms are sensitive to these parameter values: Slightly different settings may lead to very different clusterings of the data.
- Moreover, real-world, high-dimensional data sets often have very skewed distributions such that their intrinsic clustering structure may not be well characterized by a single set of global density parameters
- To overcome the difficulty in using one set of global parameters in clustering analysis, a cluster analysis method called OPTICS was proposed.
- OPTICS does not explicitly produce a data set clustering. Instead, it outputs a cluster ordering.
- This is a linear list of all objects under analysis and represents the density-based clustering structure of the data.
- Objects in a denser cluster are listed closer to each other in the cluster ordering.
- This ordering is equivalent to density-based clustering obtained from a wide range of parameter settings.
- Thus, OPTICS does not require the user to provide a specific density threshold.
- OPTICS needs two important pieces of information per object:
- The core-distance of an object p is the smallest value ε' such that the ε'-neighborhood of p has at least MinPts objects.
- The reachability-distance to object p from q is the minimum radius value that makes p density-reachable from q.
- An object p may be directly reachable from multiple core objects. Therefore, p may have multiple reachability-distances with respect to different core objects.
- The smallest reachability-distance of p is of particular interest because it gives the shortest path for which p is connected to a dense cluster.
- OPTICS computes an ordering of all objects in a given database and, for each object in the database, stores the core-distance and a suitable reachability-distance.
- OPTICS maintains a list called OrderSeeds to generate the output ordering.



Core-distance of p

Reachability - distance (p, q_1) = ' = 3 mm
Reachability - distance (p, q_2) = dist (p, q_2)

Fig. 4.2.6 OPTICS terminology

DENCLUE : Clustering Based on Density Distribution Functions

- Density estimation is a core issue in density-based clustering methods.
- DENCLUE (DENsity-based CLUstEring) is a clustering method based on a set of density distribution functions
- It is one of the most effective unsupervised classification methods, that allows to classify voluminous data.
- In probability and statistics, density estimation is the estimation of an unobservable underlying probability density function based on a set of observed data
- This method is based on the concept of density and the Hill Climbing algorithm.
- The Hill Climbing helps in the crucial phase of the reconstruction of the classes.
- DENCLUE uses a Gaussian kernel to estimate density based on the given set of objects to be clustered.
- A cluster is defined by a local maximum of the estimated density function. Data points going to the same local maximum are put into the same cluster.
- A point x* is called a density attractor if it is a local maximum of the estimated density function.
- DENCLUE has several advantages.
    - It can be regarded as a generalization of several well-known clustering methods such as single-linkage approaches and DBSCAN.
    - DENCLUE is invariant against noise.
    - The kernel density estimation can effectively reduce the influence of noise by uniformly distributing noise into the input data.
- However, DENCLUE doesn't work on data with uniform distribution.
- In high dimensional space, the data always look like uniformly distributed because of the curse of dimensionality.

- Therefore, DENCLUDE doesn't work well on high-dimensional data in general.

4.2.6 Grid Based Methods

- Grid-based methods quantize the object space into a finite number of cells that form a grid structure.
- All the clustering operations are performed on the grid structure (i.e., on the quantized space).
- The main advantage of this approach is its fast processing time, which is typically independent of the number of data objects and dependent only on the number of cells in each dimension in the quantized space.
- Using grids is often an efficient approach to many spatial data mining problems, including clustering. Therefore, grid-based methods can be integrated with other clustering methods such as density-based methods and hierarchical methods.
- A grid-based clustering method takes a space-driven approach by partitioning the embedding space into cells independent of the distribution of the input objects.
- The grid-based clustering approach uses a multiresolution grid data structure.
- It quantizes the object space into a finite number of cells that form a grid structure on which all of the operations for clustering are performed.
- The main advantage of the approach is its fast processing time, which is typically independent of the number of data objects, yet dependent on only the number of cells in each dimension in the quantized space.

STING : STatistical INformation Grid

- STING is a grid-based multiresolution clustering technique in which the embedding spatial area of the input objects is divided into rectangular cells.
- The space can be divided in a hierarchical and recursive way.
- Several levels of such rectangular cells correspond to different levels of resolution and form a hierarchical structure
- Each cell at a high level is partitioned to form a number of cells at the next lower level.
- Statistical information regarding the attributes in each grid cell, such as the mean, maximum, and minimum values, is precomputed and stored as statistical parameters.
- These statistical parameters are useful for query processing and for other data analysis tasks.
- The statistical parameters of higher-level cells can easily be computed from the parameters of the lower-level cells.

- These parameters include the following : the attribute-independent parameter, count; and the attribute-dependent parameters, mean, stdev (standard deviation), min (minimum), max (maximum), and the type of distribution that the attribute value in the cell follows such as normal, uniform, exponential, or none
- The value of distribution may either be assigned by the user if the distribution type is known beforehand or obtained by hypothesis tests.

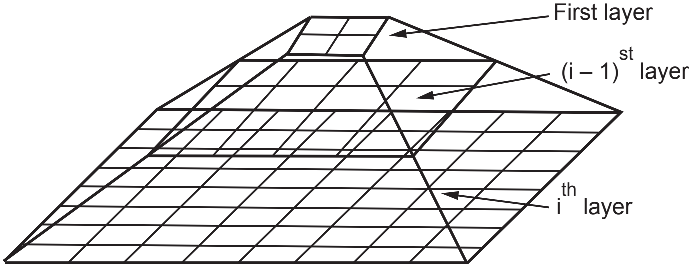


Fig 4.2.7 Hierarchical structure for STING clustering

- STING offers several advantages :
  - ○ The grid-based computation is query-independent because the statistical information stored in each cell represents the summary information of the
  - ○ Data in the grid cell, independent of the query.
  - ○ The grid structure facilitates parallel processing and incremental updating.
  - ○ Because STING uses a multiresolution approach to cluster analysis, the quality of STING clustering depends on the granularity of the lowest level of the grid structure.

CLIQUE : An Apriori-like Subspace Clustering Method

- The data object often has tens of attributes, many of which may be irrelevant.
- The values of attributes may vary considerably. These factors can make it difficult to locate clusters that span the entire data space.
- It may be more meaningful to instead search for clusters within different subspaces of the data.
- CLIQUE (CLustering In QUEst) is a simple grid-based method for finding densitybased clusters in subspaces.
- CLIQUE partitions each dimension into nonoverlappingintervals, thereby partitioning the entire embedding space of the data objects into cells.
- It uses a density threshold to identify dense cells and sparse ones.
- A cell is dense if the number of objects mapped to it exceeds the density threshold.

- The main strategy behind CLIQUE for identifying a candidate search space uses the monotonocity of dense cells with respect to dimensionality.
- This is based on the Apriori property used in frequent pattern and association rule mining.
- CLIQUE performs clustering in two steps.

Step1

- ○ In the first step, CLIQUE partitions the d-dimensional data space into nonoverlapping rectangular units, identifying the dense units among these.
- ○ CLIQUE finds dense cells in all of the subspaces.
- ○ CLIQUE partitions every dimension into intervals, and identifies intervals containing at least l points, where l is the density threshold. CLIQUE then iteratively joins two k-dimensional dense cells, c1 and c2, in subspaces

Step2

- ○ In the second step, CLIQUE uses the dense cells in each subspace to assemble clusters, which can be of arbitrary shape.

- CLIQUE scales linearly with the size of the input and has good scalability as the number of dimensions in the data is increased.

Summary of partitioning methods

Method	General characteristics
Partitioning methods	- Find mutually exclusive clusters of spherical shape - Distance-based. - May use mean or medoid (etc.) to represent cluster center - Effective for small to medium sie data sets.
Hierachical methods	- Clustering is a hierarchical decompositin (i.e., multiple levels) - Cannot correct erroneous merges or splits - May incorporate other techniques like microculustering or consider object "linkages"
Density-based methods	- Can find arrbitrarily shaped clusters. - Clusters are dense regions of objects in space that are separated by low-density regions. - Cluster density : Each point must have a minimum number of points within its "neighborhood" - May filter out outliers
Grid-based methods	- Use a multiresolution grid data structure - Fast processing time (typically independent of the number of data objects, yet dependent on grid size)

4.2.7 Evaluation of Clustering

- The major tasks of clustering evaluation include the following:
  - Assessing clustering tendency.
    - In this 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 the 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 clusterings and thus can compare two sets of clustering results on the same data set.

4.2.8 Clustering High Dimensional Data

- Clustering high-dimensional data is the search for clusters and the space in which exist. Thus, there are two major kinds of methods:
  - Subspace clustering approaches search for clusters existing in subspaces of the given high-dimensional data space, where a subspace is defined using a subset of attributes in the full space.
  - Dimensionality reduction approaches try to construct a much lower-dimensional space and search for clusters in such a space. Often, a method may construct new dimensions by combining some dimensions from the original data.
- In general, clustering high-dimensional data raises several new challenges in addition to those of conventional clustering :

- A major issue is how to create appropriate models for clusters in high-dimensional data. Unlike conventional clusters in low-dimensional spaces, clusters hidden in high-dimensional data are often significantly smaller.
- There are typically an exponential number of possible subspaces or dimensionality reduction options, and thus the optimal solutions are often computationally prohibitive.

Subspace clustering

- Subspace clustering is an evolving methodology which, instead of finding clusters in the entire feature space, aims at finding clusters in various overlapping or nonoverlapping subspaces of the high dimensional dataset.
- Subspace clustering is a technique which finds clusters within different subspaces (a selection of one or more dimensions).
- The underlying assumption is that we can find valid clusters which are defined by only a subset of dimensions (it is not needed to have the agreement of all N features).
- The resulting clusters may be overlapping both in the space of features and observations.
- Based on the search strategy, we can differentiate 2 types of subspace clustering -
- bottom up approaches start by finding clusters in low dimensional (1 D) spaces and iteratively merging them to process higher dimensional spaces (up to N D).
- Top down approaches find clusters in the full set of dimensions and evaluate the subspace of each cluster.

4.2.9 Clustering with Constraints

- Users often have background knowledge that they want to integrate into cluster analysis. There may also be application-specific requirements. Such information can be modeled as clustering constraints.
- There are three types of constraints
  - Constraints on instances -

    A constraint on instances specifies how a pair or a set of instances should be grouped in the cluster analysis. Two common types of constraints from this category include :
    - Must-link constraints. If a must-link constraint is specified on two objects x and y, then x and y should be grouped into one cluster in the output of the cluster analysis. These must-link constraints are transitive.
    - Cannot-link constraints. Cannot-link constraints are the opposite of must-link constraints. If a cannot-link constraint is specified on two objects, x and y,then

in the output of the cluster analysis, x and y should belong to different clusters. Cannot-link constraints can be entailed.

- Constraints on clusters -

  A constraint on clusters specifies a requirement on the clusters, possibly using attributes of the clusters.

  For example, a constraint may specify the minimum number of objects in a cluster, the maximum diameter of a cluster, or the shape of a cluster (e.g., a convex).

  The number of clusters specified for partitioning clustering methods can be regarded as a constraint on clusters.

- Constraints on similarity measurement -

  Often, a similarity measure, such as Euclidean distance, is used to measure the similarity between objects in a cluster analysis.

  In some applications, exceptions apply.

  A constraint on similarity measurement specifies a requirement that the similarity calculation must respect.

4.2.10 Outliner Analysis

- An outlier is a data object that deviates significantly from the rest of the objects, as if it were generated by a different mechanism.
- The data objects that are not outliers as "normal" or expected data. Similarly, we can refer to outliers as "abnormal" data.
- An outlier is an element of a data set that distinctly stands out from the rest of the data. In other words, outliers are those data points that lie outside the overall pattern of distribution as shown in Fig. 4.2.8 below.
- Outlier analysis is the identification of rare items, events or observations which raise suspicions by differing significantly from the majority of the data.

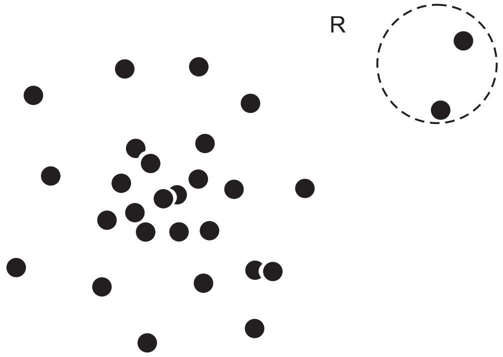


Fig 4.2.8 Understanding an Outlier (Datapoints in Region R)

- In general, outliers can be classified into three categories -
  - **Global outliers -**

    In a given data set, a data object is a global outlier if it deviates significantly from the rest of the data set.

    Global outliers are sometimes called point anomalies, and are the simplest type of outliers. Most outlier detection methods are aimed at finding global outliers.
  - **Contextual (or conditional) outliers -**

    In a given data set, a data object is a contextual outlier if it deviates significantly with respect to a specific context of the object.

    Contextual outliers are also known as conditional outliers because they are conditional on the selected context.

    Therefore, in contextual outlier detection, the context has to be specified as part of the problem definition.

    Generally, in contextual outlier detection, the attributes of the data objects in question are divided into two groups :
    - Contextual attributes : The contextual attributes of a data object define the object's context.
    - Behavioral attributes : These define the object's characteristics, and are used to evaluate whether the object is an outlier in the context to which it belongs.
  - **Collective outliers**

    Given a data set, a subset of data objects forms a collective outlier if the objects as a whole deviate significantly from the entire data set. Importantly, the individual data objects may not be outliers.

Challenges of Outlier Detection

Outlier detection is useful in many applications yet faces many challenges such as-

- Modeling normal objects and outliers effectively -
  - Outlier detection quality highly depends on the modeling of normal (nonoutlier) objects and outliers. Often, building a comprehensive model for data normality is very challenging, if not impossible.
- Application-specific outlier detection -
  - Technically, choosing the similarity/distance measure and the relationship model to describe data objects is critical in outlier detection.
- Handling noise in outlier detection -
  - Outliers are different from noise. It is also well known that the quality of real data sets tends to be poor. Noise often unavoidably exists in data collected in many applications. Noise may be present as deviations in attribute values or even as missing values. Low data quality and the presence of noise bring a huge challenge to outlier detection.
- Understandability -
  - In some application scenarios, a user may want to not only detect outliers, but also understand why the detected objects are outliers.

4.2.11 Outliner Detection Methods

There are many outlier detection methods in the literature and in practice

- **Supervised Methods**
  - ○ Supervised methods model data normality and abnormality. Domain experts examine and label a sample of the underlying data.
  - ○ Outlier detection can then be modeled as a classification problem . The task is to learn a classifier that can recognize outliers.
  - ○ Although many classification methods can be applied, challenges to supervised outlier detection include the following :
    - ▪ The two classes (i.e., normal objects versus outliers) are imbalanced. That is, the population of outliers is typically much smaller than that of normal objects.
    - ▪ In many outlier detection applications, catching as many outliers as possible (i.e., the sensitivity or recall of outlier detection) is far more important than not mislabeling normal objects as outliers.
  - ○ In summary, supervised methods of outlier detection must be careful in how they train and how they interpret classification rates due to the fact that outliers are rare in comparison to the other data samples.
- **Un-Supervised Methods**
  - ○ In some application scenarios, objects labeled as "normal" or "outlier" are not available. Hence, an unsupervised learning method has to be used.
  - ○ Unsupervised outlier detection methods make an implicit assumption: The normal objects are somewhat "clustered." In other words, an unsupervised outlier detection method expects that normal objects follow a pattern far more frequently than outliers.
  - ○ Normal objects do not have to fall into one group sharing high similarity. Instead, they can form multiple groups, where each group has distinct features.
  - ○ However, an outlier is expected to occur far away in feature space from any of those groups of normal objects.
  - ○ This assumption may not be true all the time. The collective outliers, however, share high similarity in a small area. Unsupervised methods cannot detect such outliers effectively.
  - ○ Many clustering methods can be adapted to act as unsupervised outlier detection methods. The central idea is to find clusters first, and then the data objects not belonging to any cluster are detected as outliers.
  - ○ However, such methods suffer from two issues.
    - ▪ Data object not belonging to any cluster may be noise instead of an outlier.
    - ▪ It is often costly to find clusters first and then find outliers
  - ○ The latest unsupervised outlier detection methods develop various smart ideas to tackle outliers directly without explicitly and completely finding clusters.

- **Semi-Supervised Methods**
  - Semi-supervised outlier detection methods can be regarded as applications of semi-supervised learning methods.
  - The model of normal objects then can be used to detect outliers-those objects not fitting the model of normal objects are classified as outliers.
  - If only some labeled outliers are available, semi-supervised outlier detection is trickier.
  - A small number of labeled outliers are unlikely to represent all the possible outliers.
  - Therefore, building a model for outliers based on only a few labeled outliers is unlikely to be effective. To improve the quality of outlier detection, we can get help from models for normal objects learned from unsupervised methods.
- **Statistical Methods**
  - Statistical methods (also known as model-based methods) make assumptions of data normality.
  - They assume that normal data objects are generated by a statistical (stochastic) model, and that data not following the model are outliers.
  - The effectiveness of statistical methods highly depends on whether the assumptions made for the statistical model hold true for the given data.
  - There are many kinds of statistical models. For example, the statistic models used in the methods may be parametric or nonparametric.
- **Proximity-Based Methods**
  - Proximity-based methods assume that an object is an outlier if the nearest neighbors of the object are far away in feature space, that is, the proximity of the object to its neighbors significantly deviates from the proximity of most of the other objects to their neighbors in the same data set.
  - The effectiveness of proximity-based methods relies heavily on the proximity (or distance) measure used. In some applications, such measures cannot be easily obtained.
  - Moreover, proximity-based methods often have difficulty in detecting a group of outliers if the outliers are close to one another.
  - There are two major types of proximity-based outlier detection, namely distance based and density-based outlier detection.
- **Clustering-Based Methods**
  - Clustering-based methods assume that the normal data objects belong to large and dense clusters, whereas outliers belong to small or sparse clusters, or do not belong to any clusters.

- The notion of outliers is highly related to that of clusters. Clustering-based approaches detect outliers by examining the relationship between objects and clusters.
- Intuitively , an outlier is an object that belongs to a small and remote cluster, or does not belong to any cluster.
- This leads to three general approaches to clustering-based outlier detection. Consider an object.
- Does the object belong to any cluster ? If not, then it is identified as an outlier.
- Is there a large distance between the object and the cluster to which it is closest ? If yes, it is an outlier.
- Is the object part of a small or sparse cluster ? If yes, then all the objects in that cluster are outliers.

Review Questions

1. *Explain the classification by backpropagation.*
2. *Discuss in detail : Support Vector Machines.*
3. *Differentiate between Eager Learners and Lazy Learners.*
4. *Write a note on k-nearest-neighbor method.*
5. *Explain the Case Based Reasoning (CBR).*
6. *List out three applications of K-NN and CBR.*
7. *Explain the following :*
   (a) K-S Chart (b) ROC chart (c) Area Under the Curve (AUC)
8. *Discuss in detail, the basic techniques to improve classification accuracy ?*
9. *Explain clustering and clustering analysis.*
10. *Describe the applications of clustering techniques.*
11. *Explain in detail, requirements of clustering in Data Mining.*
12. *Explain k-Means as a centroid-based technique.*
13. *Discuss how k-Means partitioning algorithm works.*
14. *Explain the k-Medoids technique.*
15. *Write a note on the Partitioning Around Medoids (PAM) algorithm.*
16. *Discuss the following hierarchical clustering approaches : agglomerative, divisive, BIRCH, Chameleon.*
17. *Elaborate the Probabilistic Hierarchical Clustering ?*
18. *What is DBSCAN and OPTICS ?*
19. *Discuss grid-based clustering methods.*
20. *Discuss the tasks involved in clustering evaluation*
21. *Explain the outlier and outlier analysis*

22. Describe the challenges of outlier detection.

23. Discuss any two Outlier detection methods.

Two Marks Questions with Answers

Q.1 What is data classification ?

OR Define Data classification. (Write any 2 definitions)

Ans. : • Data classification is the process of organizing data into categories for its most effective and efficient use. A well-planned data classification system makes essential data easy to find and retrieve.

- Data classification is the process of sorting and categorizing data into various types, forms or any other distinct class.
- Data classification enables the separation and classification of data according to data set requirements for various business or personal objectives.
- Data Classification is a form of data analysis that extracts models describing important data classes. Such models, called classifiers, predict categorical (discrete, unordered) class labels.

Q.2 What are the applications domains of data classification ?

Ans. : • Classification has many applications in various domains. These include fraud detection, target marketing, performance prediction, manufacturing, and medical diagnosis.

- There are many classification methods used in machine learning, pattern recognition, and statistics. These algorithms are designed assuming a small data size. Most of these algorithms are memory resident.
- With high data volumes in the recent days, researchers have put efforts in developing scalable classification and prediction techniques capable of handling large amounts of disk-resident data.

Q.3 How does classification work ?

OR What are the two steps in data classification ?

Ans. : • Data classification is a two-step process.

- A learning step: In this step a classification model is constructed.
- A classification step: In this step the model is used to predict class labels for given data.

Q.4 Describe the learning step in data classification.

Ans. : • The learning step (or training phase), where a classification algorithm builds the classifier by analysing or "learning from" a training set made up of database tuples and their associated class labels.

- A classifier is built describing a predetermined set of data classes or concepts.
- This is the learning step (or training phase), where a classification algorithm builds the classifier by analyzing or "learning from" a training set made up of database tuples and their associated class labels.

Q.5 Differentiate in supervised and unsupervised learning.

OR Define supervised and unsupervised learning.

Ans. : • If the class label of each training tuple is provided, this step is also known as supervised learning.

- If the class label of each training tuple is not known, and the number or set of classes to be learned may not be known in advance, it is called unsupervised learning.

Q.6 What is decision tree induction ?

OR What is decision tree ?

OR Define decision tree.

Ans. : • Decision tree induction is the learning of decision trees from class-labeled training tuples.

- A decision tree is a flowchart-like tree structure, where
  - ○ Each internal node (non-leaf node) denotes a test on an attribute
  - ○ Each branch represents an outcome of the test
  - ○ Each leaf node (or terminal node) holds a class label
  - ○ The topmost node in a tree is the root node

Q.7 How are decision trees used for classification ?

Ans. : • Given a tuple, X, for which the associated class label is unknown, the attribute values of the tuple are tested against the decision tree.

- A path is traced from the root to a leaf node, which holds the class prediction for that tuple.
- Decision trees can easily be converted to classification rules.

Q.8 Give any two reasons to explain why are decision tree classifiers so popular ?

Ans. : • The construction of decision tree classifiers does not require any domain knowledge or parameter setting, and therefore is appropriate for exploratory knowledge discovery.

- Decision trees can handle multidimensional data.
- Their representation of acquired knowledge in tree form is intuitive and generally easy to assimilate by humans.
- The learning and classification steps of decision tree induction are simple and fast.

- In general, decision tree classifiers have good accuracy.
- However, successful use may depend on the data at hand.
- Decision tree induction algorithms have been used for classification in many application areas such as medicine, manufacturing and production, financial analysis, astronomy, and molecular biology.
- Decision trees are the basis of several commercial rule induction systems.

Q.9 What is tree pruning ?

OR What are the two approaches of tree pruning ?

Ans. : • Tree pruning is performed in order to remove anomalies in the training data due to noise or outliers. The pruned trees are smaller and less complex.

- Tree Pruning Approaches - There are two approaches to prune a tree.
  - Pre-pruning - The tree is pruned by halting its construction early.
  - Post-pruning - This approach removes a sub-tree from a fully-grown tree.

Q.10 How the complexity cost of tree pruning is measured ?

Ans. : • The cost complexity is measured by the following two parameters -

  - Number of leaves in the tree, and
  - Error rate of the tree.

Q.11 Write any two strengths of decision tree approach.

Ans. : • Decision trees are able to generate understandable rules.

- Decision trees perform classification without requiring much computation.
- Decision trees are able to handle both continuous and categorical variables.
- Decision trees provide a clear indication of which fields are most important for prediction or classification.

Q.12 Write any two weaknesses of decision tree approach.

Ans. : • Decision trees are less appropriate for estimation tasks where the goal is to predict the value of a continuous attribute.

- Decision trees are prone to errors in classification problems with many class and relatively small number of training examples.
- Decision tree can be computationally expensive to train. The process of growing a decision tree is computationally expensive. At each node, each candidate splitting field must be sorted before its best split can be found. In some algorithms, combinations of fields are used and a search must be made for optimal combining weights. Pruning algorithms can also be expensive since many candidate sub-trees must be formed and compared.

Q.13 Explain brief - Bayesian classification.

Ans. : • Bayesian classification is a statistical classifier and is based on Bayes' Theorem.

- Bayesian classifiers can predict class membership probabilities such as the probability that a given tuple belongs to a particular class.
- Bayesian classifiers results with high accuracy and speed when applied to large databases.

Q.14 Define class conditional independence.

Ans. : • Naive Bayesian classifiers assume that the effect of an attribute value on a given class is independent of the values of the other attributes. This assumption is called class conditional independence.

Q.15 Explain in brief - Baye's Theorem.

Ans. : • Bayes' Theorem is named after Thomas Bayes and is defined as,

$P(H/X) = P(X/H)P(H) \ / \ P(X)$

where,

- X is data tuple and H is some hypothesis.
- Posterior Probability [P(H/X)]
- Prior Probability [P(H)]

Q.16 What is Naive Bayesian classifiers ?

Ans. : • Naive Bayes is a simple, yet effective and commonly-used, learning classifier.

- It is a probabilistic classifier that makes classifications using the Maximum A Posteriori decision rule in a Bayesian setting.
- Naive Bayesian classifiers assume that the effect of an attribute value on a given class is independent of the values of the other attributes.

Q.17 Discuss the effectiveness of Bayesian classifiers.

Ans. : • In theory, Bayesian classifiers have the minimum error rate in comparison to all other classifiers.

- In practice this is not always the case. The inaccuracies are due to assumptions made for its use, such as class-conditionalindependence, and the lack of available probability data.
- Bayesian classifiers are also useful in theoretical justification for other classifiers that do not explicitly use Bayes' theorem.

Q.18 What is rule-based classification ?

OR How rule based classification works ?

OR Give an example of rule-based classification.

Ans. : Rule-based classifier makes use of a set of IF-THEN rules for classification. We can express a rule in the following from -

IF condition THEN conclusion

Let us consider a rule R,

R: IF age > 30 AND credit = good
THEN buy_computer = yes

- Rule R can also be written as as follows -
  R1 : (age = youth) ^ (student = yes))(buys computer = yes)

Q.19 Explain rule extraction from a decision tree.

Ans. : • Rule Extraction is to build a rule-based classifier by extracting IF-THEN rules from a decision tree.

- To extract a rule from a decision tree.
  - One rule is created for each path from the root to the leaf node.
  - To form a rule antecedent, each splitting criterion is logically ANDed.
  - The leaf node holds the class prediction, forming the rule consequent.

Q.20 What is rule induction using sequential covering algorithm ?

Ans. : • Sequential covering algorithm can be used to extract IF-THEN rules form the training data.

- No decision tree is generated.
- As per the general strategy the rules are learned one at a time.
- For each time rules are learned, a tuple covered by the rule is removed and the process continues for the rest of the tuples.

Q.21 What is classification by backpropagation ?

Ans. : • The backpropagation (BP) algorithm learns the classification model by training a multilayer feed-forward neural network.

- The generic architecture of the neural network for BP is shown in the following diagrams, with one input layer, some hidden layers, and one output layer.

Q.22 Discuss the backpropagation concept in brief.

Ans. : • Back propagation learns by iteratively processing a set of training samples, comparing the network's prediction for each sample with the actual known class label.

- For each training sample, the weights aremodified so as to minimize the mean squared error between the network's prediction and the actual class.
- These modifications are made in the "backwards" direction, that is , form the output layer through each hidden layer down to the first hidden layer (hence the name backpropagation).

Q.23 What is support vector machines ?

Ans. : • Support Vector Machines (SVMs), is one of the most widely used clustering algorithms in industrial applications.

- Support Vector Machines (SVMs) is a discriminative classifier formally defined by a separating hyperplane.
- Support vector machines (SVMs), a method for the classificationof both linear and nonlinear data.

Q.24 What are the popular applications of SVM ?

Ans. : • SVMs can be used to solve various real-world problems :

- ○ SVMs are helpful in text and hypertext categorization, as their application can significantly reduce the need for labeled training instances in both the standard inductive and transductive settings.
- ○ Classification of images can also be performed using SVMs. SVM is also used in image segmentation
- ○ Hand-written characters can be recognized using SVM.
- ○ The SVM algorithm has been widely applied in the biological and other sciences.

Q.25 List down the drawbacks of Support Vector Machines (SVMs).

Ans. : • It requires full labeling of input data

- It has uncalibrated class membership probabilities - SVM avoids estimating probabilities on finite data.
- The SVM is only directly applicable for two-class tasks. Therefore, algorithms that reduce the multi-class task to several binary problems have to be applied.
- In SVMs, the parameters of a solved model are difficult to interpret.

Q.26 What are the extensions of SVM ?

Ans. : The SVMs are extended to achieve higher classification results. These are

- ○ Support-vector clustering (SVC) - SVC is a similar method that also builds on kernel functions but is appropriate for unsupervised learning. It is considered a fundamental method in data science.
- ○ Multiclass SVM
- ○ Multiclass SVM aims to assign labels to instances by using support-vector machines, where the labels are drawn from a finite set of several elements.

Q.27 Write any four examples of eager learner methods.

Ans. : • Examples of eager learners are

- ○ decision tree induction,

- ○ Bayesian classification,
- ○ Rule-based classification
- ○ Classification by backpropagation,
- ○ Support vector machines
- ○ Classification based on association rule mining

Q.28 What is lazy learners method ?

Ans. : • A lazy learner delays abstracting from the data until it is asked to make a prediction

- It Simply Stores the training data without doing any further processing on it, till it gets the next test set
- It's slow as it calculates based on the current data set instead of coming up with an algorithm based on historic data
- It has large localized data so generalization takes time at every iteration

Q.29 List the examples of lazy learners methods.

Ans. : • There are two examples of lazy learner methods -

- ○ K-nearest-neighbor classifiers
- ○ Case-based reasoning classifiers

Q.30 What is k-Nearest-Neighbor classifiers (k-NN) ?

Ans. : • k-NN is a type of instance-based learning, or lazy learning, where the function is only approximated locally and all computation is deferred until classification.

- Nearest-neighbor classifiers are based on learning by analogy, that is, by comparing a given test tuple with training tuples that are similar to it. The training tuples are described by n attributes. Each tuple represents a point in an n-dimensional space.

Q.31 What techniques are used to speed up classification time in K-NN ?

Ans. : • There are two methods used to speed up classification time in K-NN,

- In the partial distance method, we computethe distance based on a subset of the n attributes. If this distance exceeds a threshold,then further computation for the given stored tuple is halted, and the process moves onto the next stored tuple.
- The editing method removes training tuples that prove useless. This method is also referred to as pruning or condensing because it reduces the totalnumber of tuples stored.

Q.32 Case-based reasoning classifiers.

Ans. : • Case-based reasoning (CBR) classifiers use a database of problem solutions to solve new problems.

- CBR stores the tuples or "cases" for problem solving as complexsymbolic descriptions.
- When given a new case to classify, a case-based reasoner will first check if an identical training case exists.
- If one is found, then the accompanying solution to that case is returned.
- If no identical case is found, then the case-based reasoner will search for training cases having components that are similar to those of the new case.

Q.33 What are the business applications of Case Based Reasoning (CBR) ?

Ans. : • Business applications of CBR include problem resolution forcustomer service help desks, where cases describe product-related diagnostic problems.

- CBR has also been applied to areas such as engineering and law, where cases are either technical designs or legal rulings, respectively.
- Medical education is another area for CBR, where patient case histories and treatments are used to help diagnose and treat new patients.

Q.34 List out the limitations of case-based reasoning.

Ans. : • Challenges in case-based reasoning include finding a good similarity metric (e.g., for matching subgraphs) and suitable methods for combining solutions.

- Other challenges include the selection of salient features for indexing training cases and the developmentof efficient indexing techniques.
- A trade-off between accuracy and efficiency evolves asthe number of stored cases becomes very large.

Q.35 What is confusion matrix ?

Ans. : • A confusion matrix shows the number of correct and incorrect predictions made by the classification model compared to the actual outcomes (target value) in the data.

- The matrix is N*N, where N is the number of target values (classes).
- Performance of such models is commonly evaluated using the data in the matrix.

Q.36 Define (Accuracy, Value or precision, Sensitivity or Recall, Specificity) one line in the context of confusion matrix.

Ans. : • **Accuracy :** The proportion of the total number of predictions that were correct.

- **Positive Predictive Value or Precision :** The proportion of positive cases that were correctly identified.

- **Sensitivity or Recall :** The proportion of actual positive cases which are correctly identified.
- **Specificity :** The proportion of actual negative cases which are correctly identified.

Q.37 What are gain or lift charts ?

Ans. : • Gain or lift is a measure of the effectiveness of a classification model calculated as the ratio between the results obtained with and without the model.

- Gain and lift charts are visual aids for evaluating performance of classification models.
- Gain and Lift chart are mainly concerned to check the rank ordering of the probabilities.

Q.38 Write the steps for building a lift / gain chart.

Ans. : • The steps for building a lift/gain chart are :

1) Calculate probability for each observation
2) Rank these probabilities in decreasing order.
3) Build deciles with each group having almost 10% of the observations.
4) Calculate the response rate at each deciles for good (Responders), bad (Non-responders) and total.

Q.39 What is a K-S Chart ?

Ans. : • K-S or Kolmogorov-Smirnov chart measures performance of classification models.

- K-S is a measure of the degree of separation between the positive and negative distributions.

Q.40 What is a ROC chart ?

Ans. : • The ROC chart is similar to the gain or lift charts in that they provide a means of comparison between classification models.

The ROC chart shows false positive rate (1 - specificity) on X-axis, the probability of target = 1 when its true value is 0, against true positive rate (sensitivity) on Y-axis, the probability of target = 1 when its true value is 1.

Q.41 What is Area Under the Curve (AUC) ?

Ans. : • Area under ROC curve is often used as a measure of quality of the classification models.

- A random classifier has an area under the curve of 0.5, while AUC for a perfect classifier is equal to 1.
- In practice, most of the classification models have an AUC between 0.5 and 1.

Q.42 List down any four basic techniques to improve classification accuracy ?

Ans. : • Select any four from below :

- ○ Add more data
- ○ Treat missing and Outlier values
- ○ Feature Engineering
- ○ Feature Selection
- ○ Multiple algorithms
- ○ Ensemble methods
- ○ Cross Validation

Q.43 What is clustering ?

OR Define clustering ?

Ans. : • Clustering is the process of grouping a set of data objects into multiple groups or clusters so that objects within a cluster have high similarity, but are very dissimilar to objects in other clusters.

- In clustering process, dissimilarities and similarities are assessed based on the attribute values describing the objects and often involve distance measures.

Q.44 What is cluster analysis ?

Ans. : • Cluster analysis or simply clustering is the process of partitioning a set of data objects(or observations) into subsets.

- Each subset is a cluster, such that objects in a clusterare similar to one another, yet dissimilar to objects in other clusters.
- The set of clustersresulting from a cluster analysis can be referred to as a clustering.

Q.45 List down the applications of clustering.

Ans. : • Data clustering analysis is used in many applications, such as -

- ○ Market research
- ○ Pattern recognition
- ○ Data analysis
- ○ Image processing.

Q.46 Mention at least four clustering methods.

Ans. : • The various clustering methods are

- ○ Partitioning
- ○ Hierarchical
- ○ Density Based

- ○ Grid based
- ○ Model based
- ○ Constraint based

Q. 47 List out the orthogonal aspect of clustering methods.

Ans. : • There are four orthogonal aspect of clustering methods, these are :

- ○ The partitioning criteria
- ○ Separation of clusters
- ○ Similarity measure
- ○ Clustering space
- ○ Partitioning Methods

Q.48 What is a k-Medoids ?

Ans. : • K-mediod is a modified version of K-means algorithm that picks actual objects to represent the clusters.

- It is a Representative Object-Based Technique and it is NP-hard.

Q.49 What are the hierarchical clustering methods ?

Ans. : • A hierarchical clustering method works by grouping data objects into a hierarchy or "tree" of clusters.

- Representing data objects in the form of a hierarchy is useful for data summarization and visualization.
- A hierarchical method creates a hierarchical decomposition of the given set of data objects.
- A hierarchical method can be classified as being either agglomerative or divisive, based on how the hierarchical decomposition is formed.

Q.50 What are the different hierarchical clustering approaches ?

Ans. : • The agglomerative approach

- The divisive approach
- BIRCH

Q.51 What is BIRCH ?

Ans. : • BIRCH is a Multiphase Hierarchical Clustering (Using Clustering Feature Trees)

- Balanced Iterative Reducing and Clustering using Hierarchies (BIRCH) is designed for clustering a large amount of numeric data by integrating hierarchical clustering (at theinitial micro-clustering stage) and other clustering methods such as iterative partitioning(at the later macro-clustering stage).

Q.52 What are the primary phases of BIRCH ?

Ans. : • The primary phasesof BRICH are

- ○ BIRCH scans the database to build an initial in-memory CF-tree, whichcan be viewed as a multilevel compression of the data that tries to preserve the data'sinherent clustering structure.
- ○ BIRCH applies a (selected) clustering algorithm to cluster the leaf nodes ofthe CF-tree, which removes sparse clusters as outliers and groups dense clusters into larger ones.

Q.53 What is Chameleon ?

Ans. : • Chameleon is a hierarchical clustering algorithm that uses dynamic modeling to determine the similarity between pairs of clusters.

- In Chameleon, cluster similarity is assessed based on
  - ○ How well connected objects are within a cluster
  - ○ The proximity ofclusters.

Q.54 What is Probabilistic Hierarchical Clustering ?

Ans. : • Probabilistic hierarchical clustering aims to overcome disadvantages of earlier models by using probabilistic models to measure distances between clusters.

- A probabilistic hierarchical clustering method can adopt the agglomerative clustering framework, but use probabilistic models to measure the distance between clusters.

Q.55 What are density-based methods ?

Ans.: • Density based methods divide a set of objects into multiple exclusive clusters,or a hierarchy of clusters.

- Typically, density-based methods consider exclusiveclusters only, and do not consider fuzzy clusters. Moreover, density-based methodscan be extended from full space to subspace clustering.

Q.56 What is DBSCAN, OPTICS and DENCLUE ?

Ans. : • DBSCAN, OPTICS and DENCLUE are the clustering methods that stand for,

- ○ DBSCAN : (Density-Based Spatial Clustering of Applications with Noise)
- ○ OPTICS : Ordering Points to Identify the Clustering Structure
- ○ DENCLUE : Clustering Based on DensityDistribution Functions

Q.57 What are the advantages of DENCLUE over DBSCAN and OPTICS ?

Ans.: • DENCLUE has several advantages.

- ○ It can be regarded as a generalization of severalwell-known clustering methods such as single-linkage approaches and DBSCAN.
- ○ DENCLUE is invariant against noise.

- ○ The kernel density estimation can effectivelyreduce the influence of noise by uniformly distributing noise into the input data.

Q.58 Explain in brief grid-based clustering methods.

Ans. : • Grid-based methods quantize the object space into a finitenumber of cells that form a grid structure.

- All the clustering operations are performedon the grid structure (i.e., on the quantized space).
- The main advantage of this approach is its fast processing time.
- A grid-based clustering method takes a space-driven approach by partitioning the embedding space into cells independent of the distribution of the input objects.

Q.59 What are the advantages of STING ?

Ans. : • STING offers several advantages :

- ○ The grid-based computation is query-independent because the statistical information stored in each cell represents the summary information of the
- ○ Data in the grid cell, independent of the query.
- ○ The grid structure facilitates parallel processing and incremental updating.
- ○ Because STING uses a multiresolution approach to cluster analysis, the quality of STING clustering depends on the granularity of the lowest level of the grid structure.

Q.60 Write the clustering steps of CLIQUE.

Ans. : • CLIQUE performs clustering in two steps.

- ○ In the first step, CLIQUE partitions the d-dimensional data space into nonoverlapping rectangular units, identifying the dense units among these.
- ○ In the second step, CLIQUE uses the dense cells in each subspace to assemble clusters, which can be of arbitrary shape.

Q.61 List down the tasks of clustering evaluation.

Ans. : • The major tasks of clustering evaluation include the following :

- ○ Assessing clustering tendency.
- ○ Determining the number of clusters in a data set.
- ○ Measuring clustering quality.

Q.62 What is subspace clustering ?

Ans. : • Subspace clustering is an evolving methodology which aims at finding clusters in various overlapping or nonoverlapping subspaces of the high dimensional dataset.

- Subspace clustering is a technique which finds clusters within different subspaces (a selection of one or more dimensions).

Q.63 What are the constraints on clustering ?

Ans. : • There are three types of constraints.

- ○ Constraints on instances
- ○ Constraints on clusters
- ○ Constraints on similarity measurement

Q.64 What is outlier ? What is outlier analysis ?

Ans. : • An outlier is a data object that deviates significantly from the rest of the objects.

OR

- An outlier is an element of a data set that distinctly stands out from the rest of the data.
- Outlier analysis is the identification of rare items, events or observations which raise suspicions by differing significantly from the majority of the data.

Q.65 How Outliers are classified ?

Ans. : • In general, outliers can be classified into three categories -

- ○ Global outliers
- ○ Contextual (or conditional) outliers
- ○ Collective outliers

Q.66 What are the challenges of outlier detection ?

Ans. : • Outlier detection faces many challenges

- ○ Modeling normal objects and outliers effectively
- ○ Application-specific outlier detection
- ○ Handling noise in outlier detection
- ○ Understandability

Q.67 List down any four Outlier detection methods.

Ans. : • There are many outlier detection methods in the literature and in practice

- ○ Supervised Methods
- ○ Un-Supervised Methods
- ○ Semi-Supervised Methods
- ○ Statistical Methods
- ○ Proximity-Based Methods
- ○ Clustering-Based Methods

❑❑❑

Unit - V

5 WEKA Tool

Syllabus

Datasets - Introduction, Iris plants database, Breast cancer database, Auto imports database - Introduction to WEKA, The Explorer - Getting started, Exploring the explorer, Learning algorithms, Clustering algorithms, Association - rule learners.

Contents

5.1 Introduction to WEKA

What is WEKA ?

- WEKA stands for Waikato Environment for Knowledge Analysis (WEKA)
- WEKA was developed at the University of Waikato in New Zealand, and the name stands for Waikato Environment for Knowledge Analysis.
- The system is written in Java and distributed under the terms of the GNU General Public License.
- It runs on almost any platform and has been tested under Linux, Windows, and Macintosh operating systems-and even on a personal digital assistant.
- It provides a uniform interface to many different learning algorithms, along with methods for pre- and postprocessing
- WEKA is a collection of machine learning algorithms for data mining tasks.
- WEKA is a collection of machine learning algorithms for solving real-world data mining problems. The algorithms can either be applied directly to a dataset or called from your own Java code.
- It contains tools for data preparation, classification, regression, clustering, association rules mining, and visualization.
- All of WEKA's techniques are predicated on the assumption that the data is available as one flat file or relation, where each data point is described by a fixed number of attributes.
- WEKA provides access to SQL databases using Java Database Connectivity and can process the result returned by a database query. WEKA provides access to deep learning with deep learning.
- The key features of WEKA is the extended support for data mining tasks. It includes
  - 49 data preprocessing tools
  - 76 classsification / regression algorithms
  - 8 clustering algorithms
  - 3 algorithms for finding association rules
  - 15 attribute /subset evaluators + 10 search algorithms for feature selection.
- It is free software licensed under the GNU General Public License.

History

- In 1993, the University of Waikato in New Zealand began development of the original version of WEKA, which became a mix of Tcl/Tk, C, and Makefiles.

- In 1997, the decision was made to redevelop WEKA from scratch in Java, including implementations of modeling algorithms.
- In 2005, WEKA received the SIGKDD Data Mining and Knowledge Discovery Service Award.
- In 2006, Pentaho Corporation acquired an exclusive licence to use WEKA for business intelligence. It forms the data mining and predictive analytics component of the Pentaho business intelligence suite.

5.1.1 Getting Started

- This section provides information required to start with WEKA tool. The section is subdivided into
  - System requirements for WEKA
  - Downloading WEKA
  - Installing WEKA on UBUNTU
  - Documentation and learning resources for WEKA

System Requirements

- WEKA supports both 32-bit and 64-bit Operating Systems (including Windows, Linux and Mac OS)
- The current version of WEKA needs JVM (Java Virtual Machine). The following matrix shows which minimum version of Java is necessary to run a specific WEKA version. The latest official releases of WEKA require Java 8 or later.

		Java				
WEKA		1.4	1.5	1.6	1.7	1.8
	<3.4.0	X	X	X	X	X
	3.4.x	X	X	X	X	X
	3.5.x	3.5.0-3.5.2	>3.5.2	X	X	X
	3.6.x		X	X	X	X
	3.7.x		3.7.0	>3.7.0	>3.7.13	X
	3.8.x				3.8.0-3.8.1	>3.8.1
	3.9.x				3.9.0-3.9.1	>3.9.1

Downloading WEKA

- WEKA is an open source tool, hence freely available to download and installation.

- Visit **https://www.cs.waikato.ac.nz/ml/WEKA/downloading.html** to get OS specific WEKA version.
- The download package also includes Oracle's Java VM (Virtual Machine). JVM is required with the current version of WEKA.
- Before downloading, check the Operating Systems (OS) on your machine. There are separate downloader packages for 32-bit and 64-bit OS for Windows, Linux and Mac OS.
- There are two versions of WEKA to download with self-extracting executable(s) :
- WEKA stable version (3.8 is the latest)
  - This branch of WEKA receives bug fixes only, although new features may become available in packages.
- WEKA development version (3.9 is the latest)
  - This is the trunk of WEKA and continues from the stable-3.8 code line. It receives both bug fixes and new features.

Installing WEKA

- WEKA on Ubuntu/Linux
  - WEKA can be installed on Ubuntu / Linux Family OS by two ways
  - Using apt-get utility using terminal window
    - Installing WEKA on Ubuntu / Linux family is easy.
    - **Step 1** - Open the terminal window and switch to superuser privileges using

```
sudo su.
```

Provide, superuser credentials to get privileges.

- **Step 2** - Run update command to update package repositories and get latest package information.

```
sudo apt-get update -y
```

- **Step 3** - Run the install command with -y flag to quickly install the packages and dependencies.

```
sudo apt-get install -y WEKA
```

Note : y flag means to assume yes and silently install, without asking you questions in most cases.

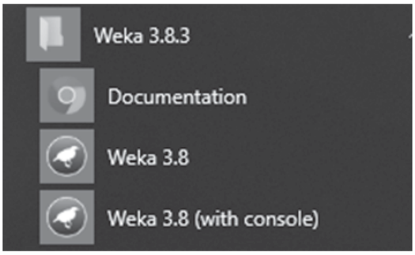
- ○ Using manual installation. Refer the **Downloading section** to get the details.
  - After the WEKA3.8.x.zip is downloaded, Unzip this file to required folder
  - Name the unzip folder as 'WEKA'
  - Change current directory to 'WEKA'

```
cd WEKA/WEKA-3-8 -1/
```

  - Run the WEKA.jar file to get the WEKA screen

```
java -jar WEKA.jar
```

  - You will see the WEKA tool wecome screen as follows
- On successful installation, you will be able to see following components installed on your computer



Weka 3.8.3

Documentation

Weka 3.8

Weka 3.8 (with console)

Fig 5.1.1 WEKA components installed

Documentation and learning resources for WEKA

- For an overview of the techniques implemented in WEKA, refer the data mining book available at **https://www.cs.waikato.ac.nz/~ml/WEKA/book.html.**
- General documentation -The online appendix on The WEKA Workbench, distributed as a free PDF.
- The WEKA manual (WEKA 3.6.15, WEKA 3.8.3, WEKA 3.9.3), as included in WEKA installer. If you have installed WEKA successfully you already have WEKA Documentation on your computer. Refer Fig. 5.1.1.
- Videos and slides for our three online courses on data mining with WEKA : Data Mining with WEKA, More Data Mining with WEKA, and Advanced Data Mining with WEKA.
  https://www.cs.waikato.ac.nz/ml/WEKA/mooc/dataminingwithWEKA/

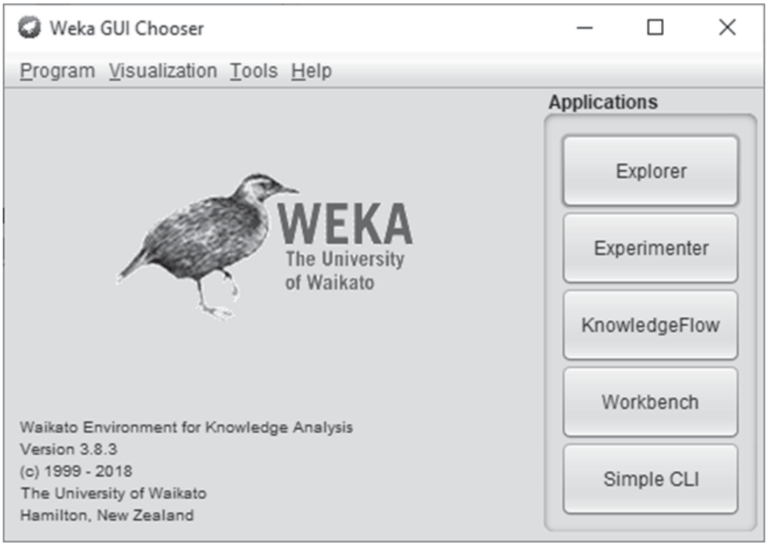
5.1.2 The WEKA Explorer

Starting the WEKA tool

- WEKA tool can be loaded using TWO methods
  - WEKA 3.8 GUI
  - WEKA 3.8 (with Console / Terminal)
- Locate the WEKA logo on your computer and select the choice as mentioned above to start WEKA tool
- We will continue the further part of the unit with reference to WEKA 3.8 GUI

WEKA application interfaces :

- There are totally five application interfaces available for WEKA. When we open WEKA, it will start the WEKA GUI Chooser screen from where we can open the WEKA application interface.
- The interface is as



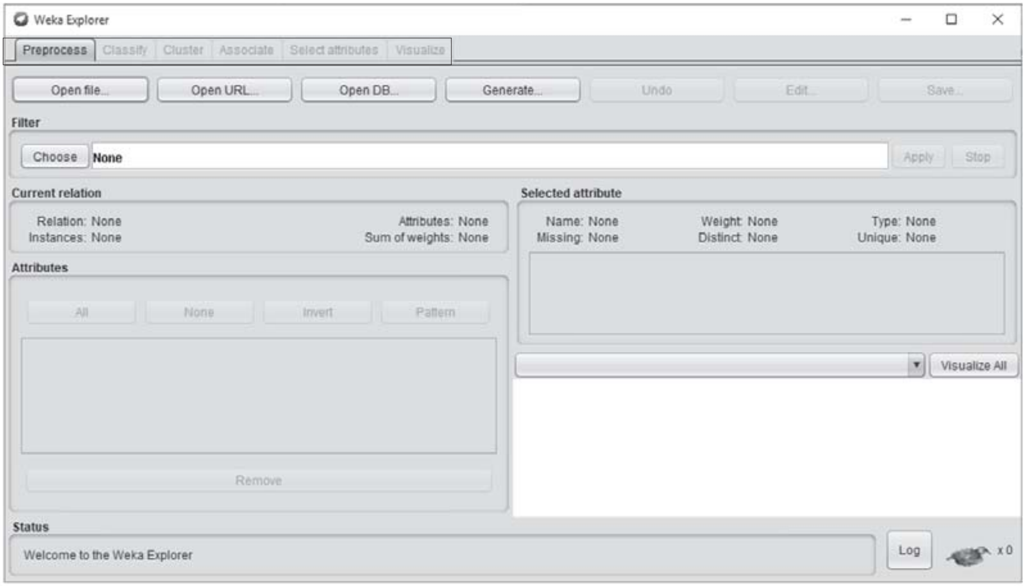
Waikato Environment for Knowledge Analysis
Version 3.8.3
(c) 1999 - 2018
The University of Waikato
Hamilton, New Zealand

Fig 5.1.2 WEKA GUI chooser

- The WEKA GUI interface has the basic components as
  1) Explorer
  2) Experimenter
  3) KnowledgeFlow
  4) Workbench
  5) Simple CLI

The Explorer

- The easiest way to use WEKA is through a graphical user interface called the Explorer. This gives access to all of its facilities using menu selection and form filling.



Filter: None

Relation: None
Attributes: None
Instances: None
Sum of weights: None

Name: None
Weight: None
Type: None
Missing: None
Distinct: None
Unique: None

Welcome to the Weka Explorer

Fig. 5.1.3 WEKA explorer

- The easiest way to use WEKA is through a graphical user interface called the explorer.
- This gives access to all of its facilities using menu selection and form filling.
- For example, you can quickly read in a dataset from an ARFF file (or spreadsheet) and build a decision tree from it.
- The explorer guides you by presenting choices as menus, by forcing you to work in an appropriate order by graying out options until they are applicable, and by presenting options as forms to be filled out.

Helpful tool tips - Pop up as the mouse passes over items on the screen to explain what they do.

The Knowledge Flow

- The knowledge interface allows you to design configurations for streamed data processing.

- A fundamental disadvantage of the explorer is that it holds everything in main memory-when you open a dataset, it immediately loads it all in.
- This means that it can only be applied to small to medium-sized problems.
- However,WEKA contains some incremental algorithms that can be used to process very large datasets.
- The knowledge flow interface lets you drag boxes representing learning algorithms and data sources around the screen and join them together into the configuration you want.
- It enables you to specify a data stream by connecting components representing data sources, preprocessing tools, learning algorithms, evaluation methods, and visualization modules.
- If the filters and learning algorithms are capable of incremental learning, data will be loaded and processed incrementally.

Experimenter

- Experimenter is designed to help you answer a basic practical question when applying classification and regression techniques:which methods and parameter values work best for the given problem ?
- There is usually no way to answer this question a priori, and one reason we developed
- The workbench was to provide an environment that enables WEKA users to compare a variety of learning techniques. This can be done interactively using the explorer.
- However, the experimenter allows you to automate the process by making it easy to run classifiers and filters with different parameter settings on a corpus of datasets, collect performance statistics and perform significanc tests.
- Advanced users can employ the experimenter to distribute the computing load across multiple machines using Java Remote Method Invocation (RMI).
- In this way you can set up large-scale statistical experiments and leave them to run.

5.1.3 Exploring the Explorer

Getting started

- Suppose you have some data and you want to build a decision tree from it.
- First, you need to prepare the data then fire up the Explorer and load in the data.
- Next you select a decision tree construction method, build a tree, and interpret the output.

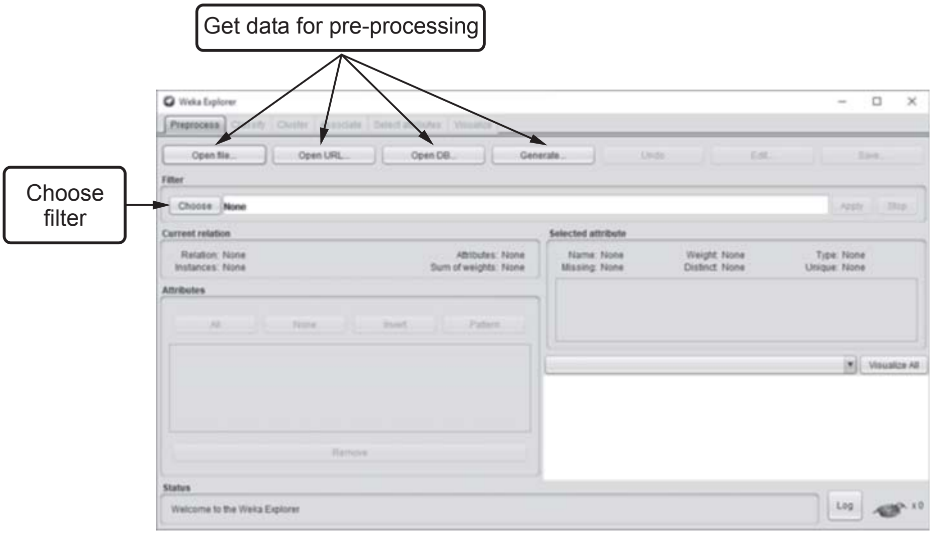
- It's easy to do it again with a different tree construction algorithm or a different evaluation method.
- In the Explorer you can flip back and forth between the results you have obtained, evaluate the models that have been built on different datasets, and visualize graphically both the models and the datasets themselves-including any classification errors the models make.

Getting data for pre-processing

The explorer facilitates pre-processing of data in multiple ways. Refer the fig 5.1.4 to understand the controls in WEKA explorer

WEKA data formats

- WEKA uses the Attribute Relation File Format for data analysis, by default. But listed below are some formats that WEKA supports, from where data can be imported :
  - CSV
  - ARFF
  - Database using ODBC-JDBC



Filter: None

Relation: None
Instances: None
Attributes: None
Sum of weights: None

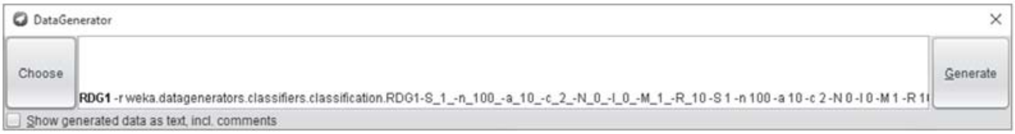
Name: None
Missing: None
Weight: None
Distinct: None
Type: None
Unique: None

Welcome to the Weka Explorer

Fig 5.1.4 Data Input methods for WEKA

- WEKA explorer facilitates inputting dataset for analysis using following ways
- **Generate data on the fly**

If user wishes to generate random data at runtime, then click Generate option. It opens the pop-up window to generate data randomly as in Fig 5.1.5



```
RDG1 -r weka.datagenerators.classifiers.classification.RDG1-S_1_-n_100_-a_10_-c_2_-N_0_-I_0_-M_1_-R_10 -S 1 -n 100 -a 10 -c 2 -N 0 -I 0 -M 1 -R 1(
```

□ Show generated data as text, incl. comments

Fig 5.1.5 Data generator window

- Click on Choose button to select the pre-exisiting classification source from which dataset can be generated . Refer Fig 5.1.6 for this

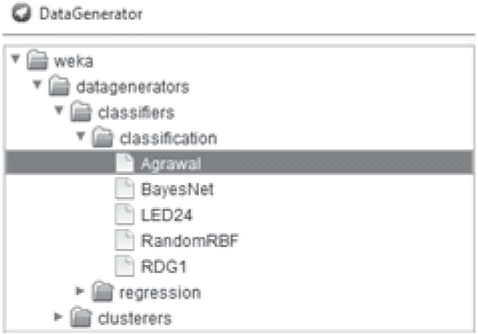
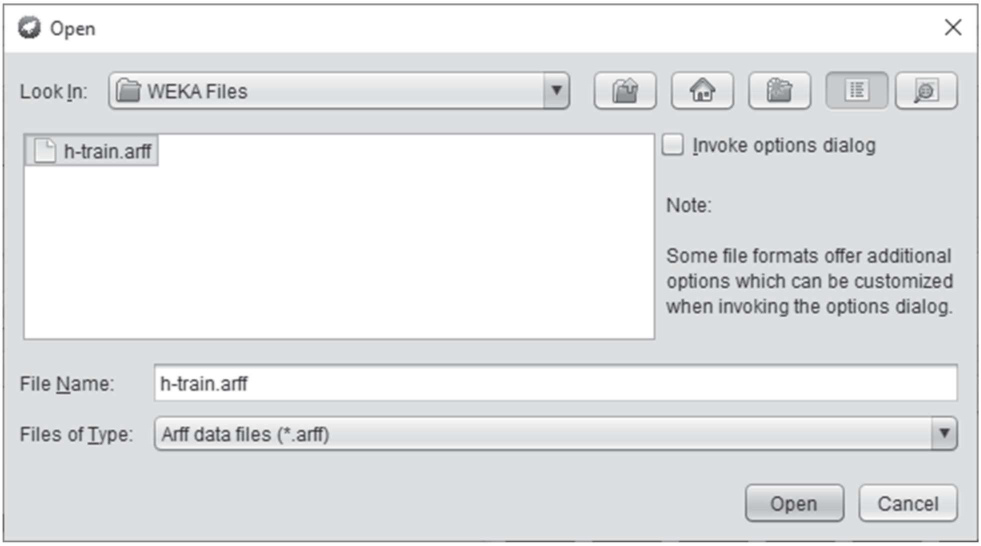


Fig 5.1.6 Choosing classifier for generating data

- **Loading data from files**
  - WEKA provides facility to use dataset in .csv and .arff format. Some of the commonly used ready datasets can be downloaded from WEKA website **https://www.cs.waikato.ac.nz/ml/WEKA/datasets.html**. Here we are using Hydrophobicity dataset for the demo
  - Click on the Open File control to load the dataset as in Fig 5.1.7



Open

Look In: WEKA Files

h-train.arff

□ Invoke options dialog

Note:

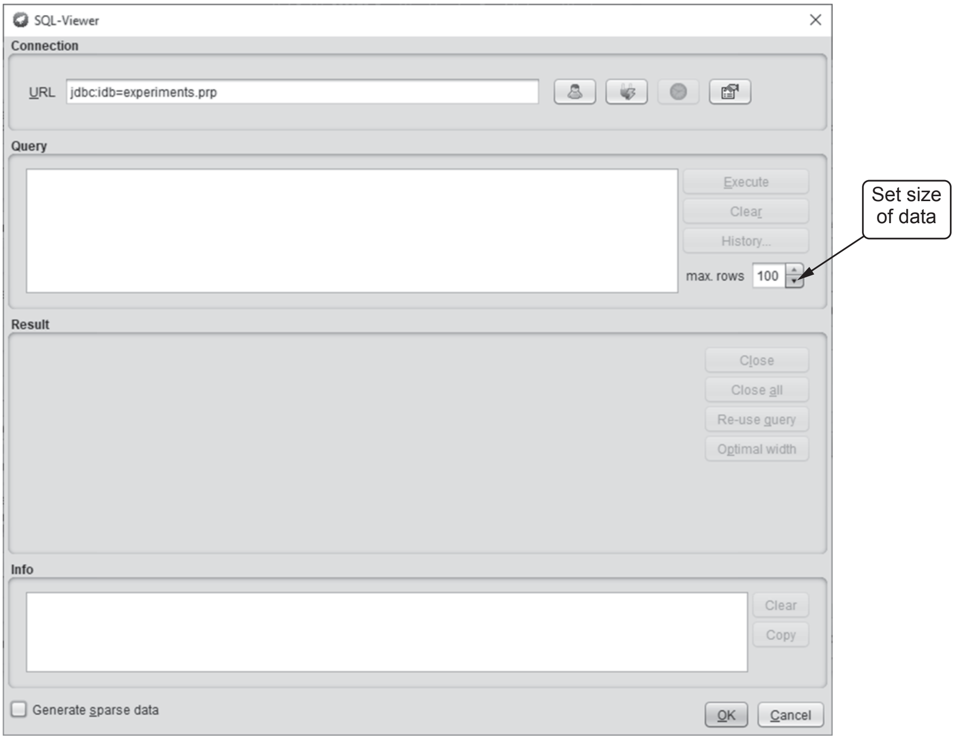
Some file formats offer additional options which can be customized when invoking the options dialog.

File Name: h-train.arff

Files of Type: Arff data files (*.arff)

Fig 5.1.7 Loading data from .arff file

- **from existing DB**
  - WEKA can also connect to DB using username and password.
  - Click on the Open DB control to load the dataset as in Fig. 5.1.8



SQL-Viewer

Connection

URL jdbc:idb=experiments.prp

Query

max. rows 100

Result

Info

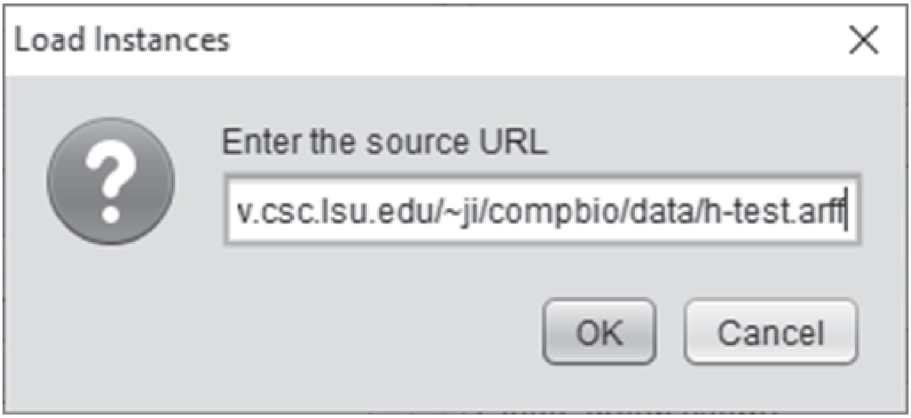
□ Generate sparse data

Set size of data

Fig 5.1.8 Connect to DB to load data

Reading data from URL

WEKA lets, user to load data from remote URLs by using Open URL control. Refer Fig 5.1.9 for OpenURL prompt.



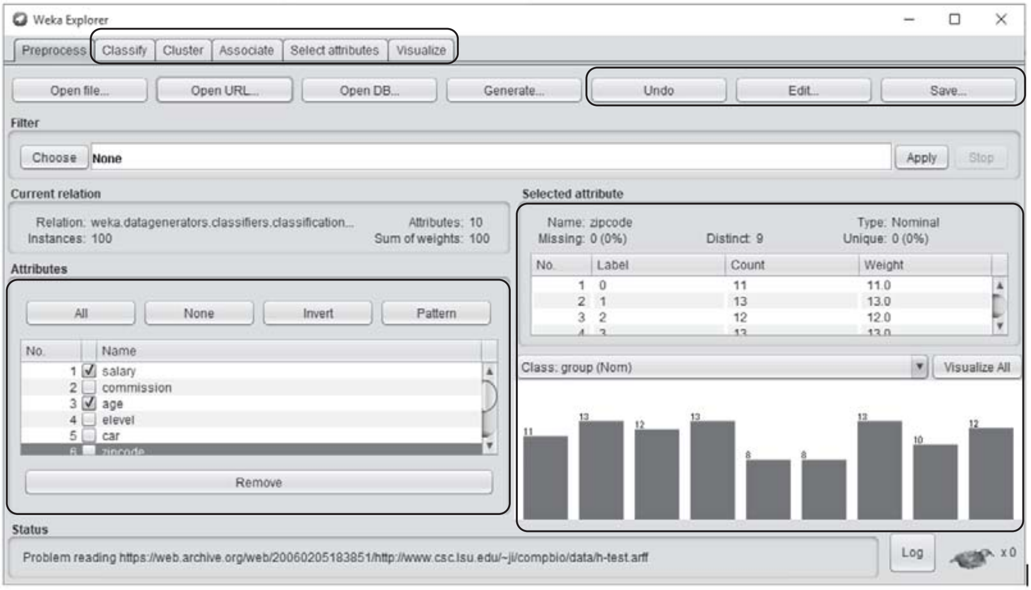
Load Instances

Enter the source URL

v.csc.lsu.edu/~ji/compbio/data/h-test.arff

Fig 5.1.9 for OpenURL - Load Instances

After the data is loaded using one of the four method, remaining controls of WEKA tool are enabled as shown in the Fig. 5.1.10. The highlighted controls are shown in the Fig. 5.1.10.



Filter: None

Current relation

Relation: weka.datagenerators.classifiers.classification... Attributes: 10
Instances: 100 Sum of weights: 100

Attributes

No.		Name
1	☑	salary
2	□	commission
3	☑	age
4	□	elevel
5	□	car
6	□	zipcode

Selected attribute

Name: zipcode Type: Nominal
Missing: 0 (0%) Distinct: 9 Unique: 0 (0%)

No.	Label	Count	Weight
1	0	11	11.0
2	1	13	13.0
3	2	12	12.0
4	3	13	13.0

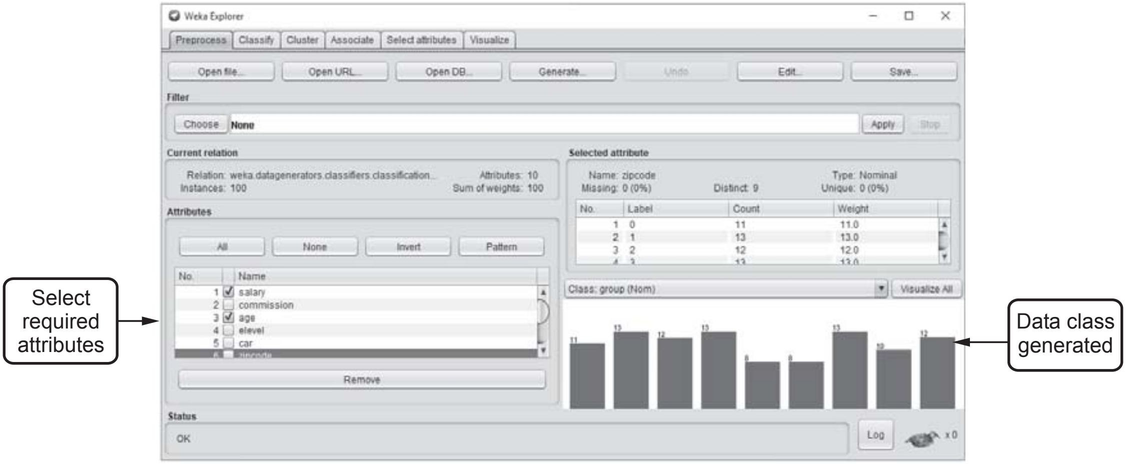
Class: group (Nom)

Status

Problem reading https://web.archive.org/web/20060205183851/http://www.csc.lsu.edu/~ji/compbio/data/h-test.arff

Fig 5.1.10 After datais loaded

A user can select which attributes to show in the visualization by selecting attribute names from the Attribute section. Refer Fig 5.1.11 for this.



Filter: None

Relation: weka.datagenerators.classifiers.classification... Attributes: 10
Instances: 100 Sum of weights: 100

Name: zipcode Type: Nominal
Missing: 0 (0%) Distinct: 9 Unique: 0 (0%)

No.	Label	Count	Weight
1	0	11	11.0
2	1	13	13.0
3	2	12	12.0
4	3	13	13.0

No.		Name
1	☑	salary
2	□	commission
3	☑	age
4	□	elevel
5	□	car
6	□	zipcode

Class: group (Nom)

Status: OK

Select required attributes

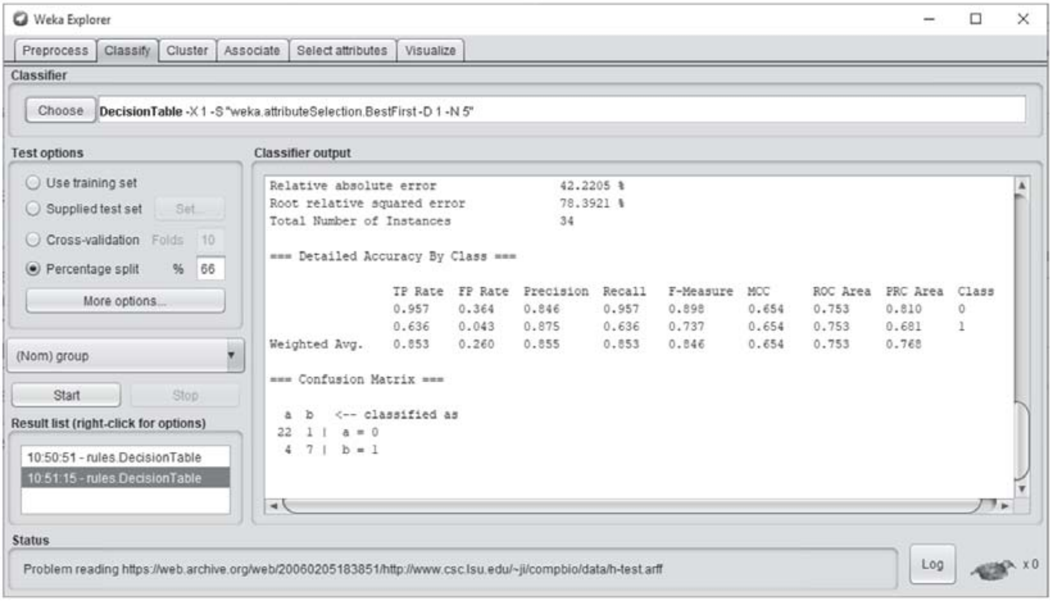
Data class generated

Fig 5.1.11 First dataset generated with class generation

As a result of attribute selection, one can see the data classes graph generated by WEKA tool as shown in Fig 5.1.7

Selecting Classifiers

- The user can select the appropriate classifier for data analysis by selecting classifier tab as shown in Fig 5.1.12



Classifier: DecisionTable -X 1 -S "weka.attributeSelection.BestFirst -D 1 -N 5"

Test options:
○ Use training set
○ Supplied test set
○ Cross-validation Folds 10
● Percentage split % 66

(Nom) group

Result list (right-click for options)
10:50:51 - rules.DecisionTable
10:51:15 - rules.DecisionTable

```
Relative absolute error                 42.2205 %
Root relative squared error             78.3921 %
Total Number of Instances               34

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0.957    0.364    0.846      0.957    0.898      0.654    0.753     0.810     0
                 0.636    0.043    0.875      0.636    0.737      0.654    0.753     0.681     1
Weighted Avg.    0.853    0.260    0.855      0.853    0.846      0.654    0.753     0.768

=== Confusion Matrix ===

  a  b   <-- classified as
 22  1 |  a = 0
  4  7 |  b = 1
```

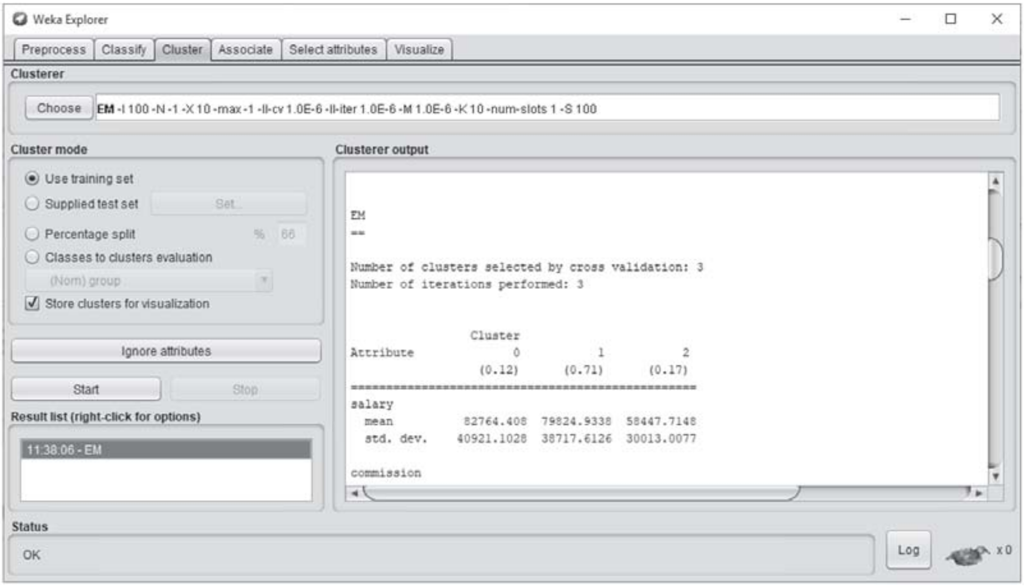
Status: Problem reading https://web.archive.org/web/20060205183851/http://www.csc.lsu.edu/~jl/compbio/data/h-test.arff

Fig 5.1.12 Classifier tab

- In this tab, user can specify whether the loaded data is a training data or a test data. If the supplied data has both training and test component, user can specify the split percent of this data.
- Click on the start button to generate the classification of the data. One can see the output of classification on the right side pane, in Fig. 5.1.12.

Cluster formation of data

- Go to the cluster tab and click on the start button. This initiates the cluster formation process. At the end of the process, WEKA shows clusters generated on the right pane as in Fig. 5.1.13.
- One can select or deselect option for visualization of clusters in this tab.



EM -I 100 -N -1 -X 10 -max -1 -ll-cv 1.0E-6 -ll-iter 1.0E-6 -M 1.0E-6 -K 10 -num-slots 1 -S 100

● Use training set

○ Supplied test set

○ Percentage split % 66

○ Classes to clusters evaluation

(Nom) group

☑ Store clusters for visualization

11:38:06 - EM

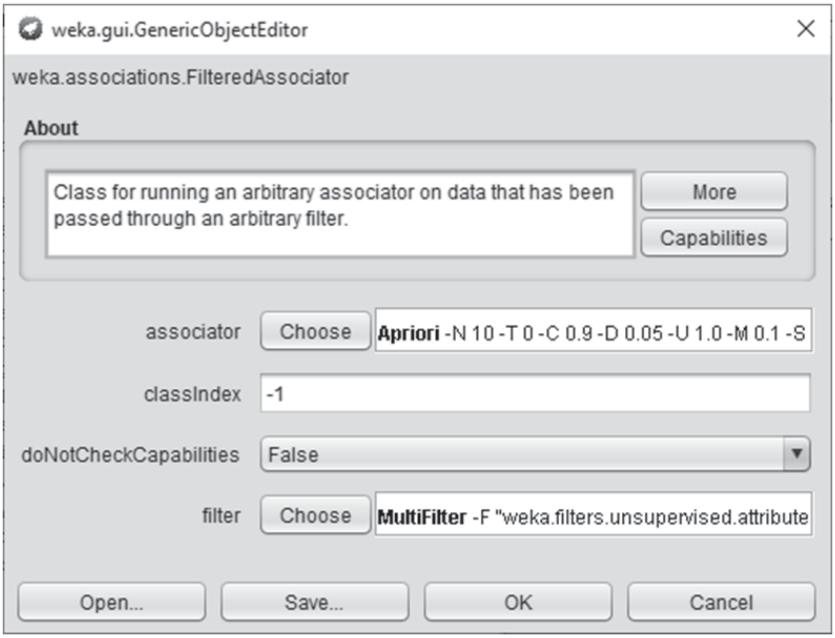
```
EM
==

Number of clusters selected by cross validation: 3
Number of iterations performed: 3


                   Cluster
Attribute                0           1           2
                    (0.12)      (0.71)      (0.17)
==================================================
salary
  mean           82764.408  79824.9338  58447.7148
  std. dev.     40921.1028  38717.6126  30013.0077

commission
```

OK

Fig 5.1.13 Cluster formation

Data Association

- One can associate the input data by selecting associate tab in WEKA Explorer. This tab allows user to select Associator rule to be applied.
- The detailed associator can be selected by clicking on Choose control as shown in Fig. 5.1.14.



weka.gui.GenericObjectEditor

weka.associations.FilteredAssociator

About

Class for running an arbitrary associator on data that has been passed through an arbitrary filter.

associator: **Apriori** -N 10 -T 0 -C 0.9 -D 0.05 -U 1.0 -M 0.1 -S

classIndex: -1

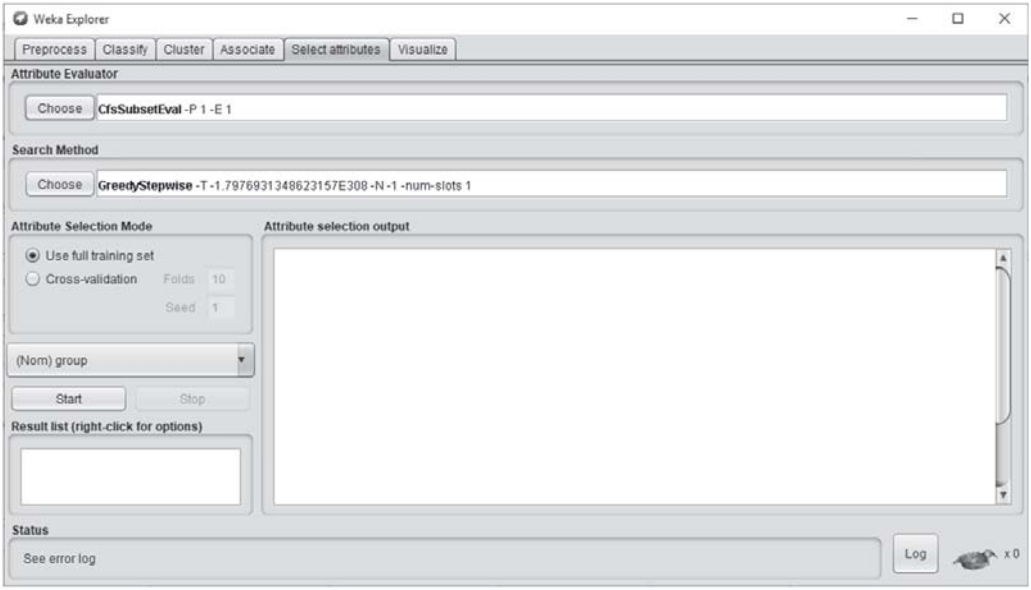
doNotCheckCapabilities: False

filter: **MultiFilter** -F "weka.filters.unsupervised.attribute

Fig 5.1.14 Detailing of associator

Selecting the attributes for visualization

- WEKA provides facility of attribute selection for visualization. Refer Fig 5.1.15 for this



Weka Explorer

Attribute Evaluator: **CfsSubsetEval** -P 1 -E 1

Search Method: **GreedyStepwise** -T -1.7976931348623157E308 -N -1 -num-slots 1

Attribute Selection Mode

● Use full training set

○ Cross-validation Folds 10 Seed 1

(Nom) group

Result list (right-click for options)

Attribute selection output

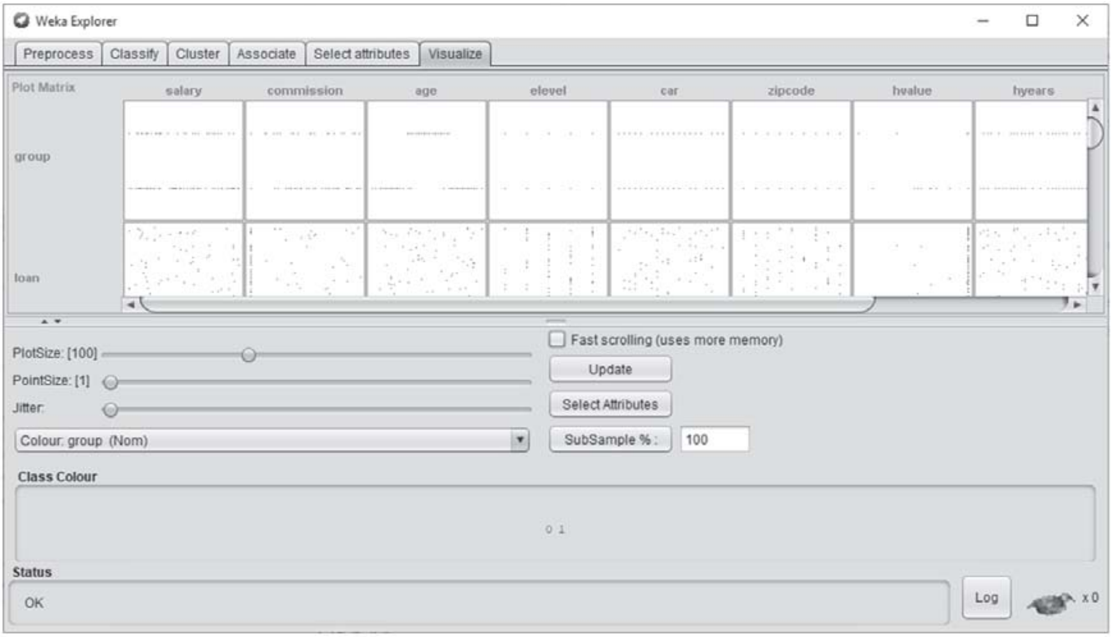
Status

See error log

Fig 5.1.15 Setting attribute selector

Data Visualization (Final Output)

- The data visualization is generated by WEKA based on the selected parameters in the earlier tab. One can see this graphical visualization using Visualize tab as shown in Fig. 5.1.16.



PlotSize: [100]

PointSize: [1]

Jitter:

Colour: group (Nom)

□ Fast scrolling (uses more memory)

SubSample % : 100

Status

OK

Fig 5.1.16 Data visualization

- The user can click on every plot (graph) generated to enlarge it.
- The look and colouring scheme can be changed for better look by selecting appropriate options.

Summarizing the WEKA Explorer

- We have briefly investigated two of the six tabs at the top of the Explorer window. In summary, here's what all of the tabs do :
  - Pre-process : Choose the dataset and modify it in various ways.
  - Classify : Train learning schemes that perform classification or regression and evaluate them.
  - Cluster : Learn clusters for the dataset.
  - Associate : Learn association rules for the data and evaluate them.
  - Select attributes : Select the most relevant aspects in the dataset.
  - Visualize : View different two-dimensional plots of the data and interact with them.
- Each tab gives access to a whole range of facilities

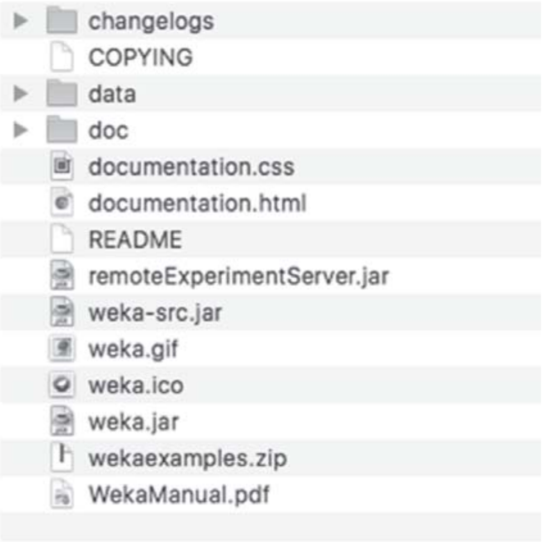
5.2 Datasets

5.2.1 Introduction

- It is a good idea to have small well understood datasets when getting started in machine learning and learning a new tool.
- The WEKA machine learning workbench provides a directory of small well understood datasets in the installed directory.
- In this section the user will discover some of these small well understood datasets distributed with WEKA, their details and where to learn more about them.
- We will focus on a handful of datasets of differing types. After reading this post you will know:
- Where the sample datasets are located or where to download them afresh if you need them.
- Specific standard datasets you can use to explore different aspects of classification and regression predictive models.
- How to get more information about specific datasets and state of the art results.

5.2.2 WEKA Datasets

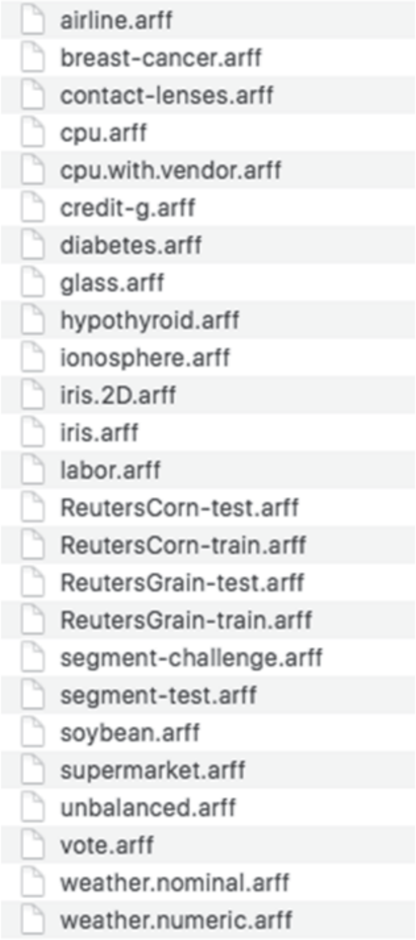
- An installation of the open source WEKA machine learning workbench includes a data/ directory full of standard machine learning problems. Refer Fig 5.2.1 for the WEKA Workbench parent directory structure



changelogs
COPYING
data
doc
documentation.css
documentation.html
README
remoteExperimentServer.jar
weka-src.jar
weka.gif
weka.ico
weka.jar
wekaexamples.zip
WekaManual.pdf

Fig 5.2.1 WEKA workbench parent directory structure

- This is very useful when you are getting started in machine learning or learning how to get started with the WEKA platform. It provides standard machine learning datasets for common classification and regression problems, for example, refer Fig. 5.2.2.
- All datasets are in the WEKA native ARFF file format and can be loaded directly into WEKA, meaning you can start developing practice models immediately.
- There are some special distributions of WEKA that may not include the data/ directory. If you have chosen to install one of these distributions, you can download the .zip distribution of WEKA, unzip it and copy the data/ directory to somewhere that you can access it easily from WEKA.
- There are many datasets to play with in the data/ directory, in the following sections I will point out a few that you can focus on for practicing and investigating predictive modelling problems.



airline.arff
breast-cancer.arff
contact-lenses.arff
cpu.arff
cpu.with.vendor.arff
credit-g.arff
diabetes.arff
glass.arff
hypothyroid.arff
ionosphere.arff
iris.2D.arff
iris.arff
labor.arff
ReutersCorn-test.arff
ReutersCorn-train.arff
ReutersGrain-test.arff
ReutersGrain-train.arff
segment-challenge.arff
segment-test.arff
soybean.arff
supermarket.arff
unbalanced.arff
vote.arff
weather.nominal.arff
weather.numeric.arff

Fig. 5.2.2 WEKA datasets that come with installation

Attribute Relation File Format (ARFF) :

- ARFF stands for Attribute-Relation File Format.
- It is an ASCII text file that describes a list of instances sharing a set of attributes.
- ARFF files were developed by the Machine Learning Project at the Department of Computer Science of The University of Waikato for use with the Weka machine learning software.
- ARFF files have two distinct sections.

  1) The header section defines the relation (data set) name, attribute name and the type.

  2) The data section lists the data instances.
- An ARFF file requires the declaration of the relation, attribute and data.
  - @relation : This is the first line in any ARFF file, written in the header section, followed by the relation/data set name. The relation name must be a string and if it contains spaces,then it should be enclosed between quotes.
  - @attribute : These are declared with their names and the type or range in the header section. Weka supports the following data types for attributes :
    - Numeric
    - <nominal-specification>
    - String
    - date
  - @data - Defined in the Data section followed by the list of all data segments.

Binary Classification Datasets

- Binary classification is where the output variable to be predicted is nominal comprised of two classes.
- It is the most well studied type of predictive modeling problem and the type of problem that is good to start with.
- There are three standard binary classification problems in the data/ directory that you can focus on :
  - **Pima Indians Onset of Diabetes : (diabetes.arff)**
    - Each instance represents medical details for one patient and the task is to predict whether the patient will have an onset of diabetes within the next five years.
    - There are 8 numerical input variables all of which have varying scales.
    - You can learn more about this dataset on the UCI Machine Learning Repository.
    - Top results are in the order of 77% accuracy.

- **Breast Cancer : (breast-cancer.arff)**
  - Each instance represents medical details of patients and samples of their tumor tissue and the task is to predict whether or not the patient has breast cancer.
  - There are 9 input variables all of which a nominal.
  - You can learn more about the datasets in the UCI Machine Learning Repository.
  - Top results are in the order of 75% accuracy.
- **Ionosphere (ionosphere.arff)**
  - Each instance describes the properties of radar returns from the atmosphere and the task is to predict whether or not there is structure in the ionosphere.
  - There are 34 numerical input variables of generally the same scale.
  - You can learn more about this dataset on the UCI Machine Learning Repository.
  - Top results are in the order of 98% accuracy.

Multi-Class Classification Datasets

- There are many classification type problems, where the output variable has more than two classes. These are called multi-class classification problems.
- This is a good type of problem to look at after you have some confidence with binary classification.
- Three standard multi-class classification problems in the data/ directory that you can focus on are :
  - **Iris Flowers Classification : (iris.arff)**
- Each instance describes measurements of iris flowers and the task is to predict to which species of 3 iris flower the observation belongs.
- There are 4 numerical input variables with the same units and generally the same scale.
- You can learn more about the datasets in the UCI Machine Learning Repository.
- Top results are in the order of 96% accuracy.

- **Large Soybean Database : (soybean.arff)**
  - Each instance describes properties of a crop of soybeans and the task is to predict which of the 19 diseases the crop suffers.
  - There are 35 nominal input variables. You can learn more about this dataset on the UCI Machine Learning Repository.
- **Glass Identification: (glass.arff)**
  - Each instance describes the chemical composition of samples of glass and the task is to predict the type or use of the class from one of 7 classes.
  - There are 10 numeric attributes that describe the chemical properties of the glass ad its refractive index.
  - You can learn more about this dataset on the UCI Machine Learning Repository.

5.2.3 Iris Plants Dataset

- Iris plant dataset is one of the best-known databases to be found in the pattern recognition literature.
- The iris flower data set is a multivariate data set introduced by the British statistician and biologist Ronald Fisher in his 1936 paper The use of multiple measurements in taxonomic problems.
- It is sometimes called Anderson's Iris data set because Edgar Anderson collected the data to quantify the morphologic variation of Iris flowers of three related species.
- The data set consists of 50 samples from each of three species of Iris (Iris Setosa, Iris virginica, and Iris versicolor).
- Four features were measured from each sample: the length and the width of the sepals and petals, in centimeters.
- This dataset became a typical test case for many statistical classification techniques in machine learning such as support vector machines
- This dataset is free and is publicly available at the UCI Machine Learning Repository at **https://archive.ics.uci.edu/ml/datasets/iris**
- Source
  - Creator : R.A. Fisher
  - Donor : Michael Marshall (MARSHALL%PLU '@' io.arc.nasa.gov)

- Details -

Data Set Characteristics :	Multivariate	**Number of Instances :**	150	**Area :**	Life
Attribute Characteristics :	Real	**Number of Attributes :**	4	**Date Donated**	1988-07-01
Associated Tasks :	Classification	**Missing Values ?**	No	**Number of Web Hits :**	2914804

- The data set contains 3 classes of 50 instances each, where each class refers to a type of iris plant.
- One class is linearly separable from the other 2; the latter are NOT linearly separable from each other.
- Predicted attribute : class of iris plant. This is an exceedingly simple domain.
- This data differs from the data presented in Fishers article (identified by Steve Chadwick, spchadwick '@' espeedaz.net).
  - The 35^{th} sample should be: 4.9,3.1,1.5,0.2,"Iris-setosa" where the error is in the fourth feature.
  - The 38th sample : 4.9,3.6,1.4,0.1,"Iris-setosa" where the errors are in the second and third features
- Number of Instances : 150 (50 in each of three classes)
- Number of attributes : 4 numeric, predictive attributes and the class
- Attribute information :
  - sepal length in cm
  - sepal width in cm
  - petal length in cm
  - petal width in cm
  - class :
    - Iris Setosa
    - Iris Versicolour
    - Iris Virginica
- Missing Attribute Values : None
- Summary Statistics:
  - Min Max Mean SD Class Correlation
  - sepal length : 4.3 7.9 5.84 0.83 0.7826

- sepal width : 2.0 4.4 3.05 0.43 -0.4194
- petal length : 1.0 6.9 3.76 1.76 0.9490 (high!)
- petal width : 0.1 2.5 1.20 0.76 0.9565 (high!)
- Class Distribution : 33.3% for each of 3 classes.

5.2.4 Breast Cancer Dataset

- This breast cancer domain was obtained from the University Medical Centre, Institute of Oncology, Ljubljana, Yugoslavia.
- This is one of three domains provided by the Oncology Institute that has repeatedly appeared in the machine learning literature.
- This data set includes 201 instances of one class and 85 instances of another class. The instances are described by 9 attributes, some of which are linear and some are nominal.
- This is a free dataset that can be downloaded from **https://archive.ics.uci.edu/ml/machine-learning-databases/breast-cancer/**
- Source:
  - Creators : Matjaz Zwitter and Milan Soklic (physicians), Institute of Oncology University Medical Center Ljubljana, Yugoslavia
  - Donors : Ming Tan and Jeff Schlimmer (Jeffrey.Schlimmer '@' a.gp.cs.cmu.edu)
- Details

Data Set Characteristics :	Multivariate	**Number of Instances :**	286	**Area :**	Life
Attribute Characteristics :	Categorical	**Number of Attributes :**	9	**Date Donated**	1988-07-11
Associated Tasks :	Classification	**Missing Values ?**	Yes	**Number of Web Hits :**	417884

- Number of instances - 286
- Number of attributes 9
- Attribute information
  - Class : no-recurrence-events, recurrence-events
  - Age : 10-19, 20-29, 30-39, 40-49, 50-59, 60-69, 70-79, 80-89, 90-99.
  - Menopause : lt40, ge40, premeno.
  - Tumor-size : 0-4, 5-9, 10-14, 15-19, 20-24, 25-29, 30-34, 35-39, 40-44, 45-49, 50-54, 55-59.

- ○ inv-nodes : 0-2, 3-5, 6-8, 9-11, 12-14, 15-17, 18-20, 21-23, 24-26, 27-29, 30-32, 33-35, 36-39.
- ○ Node-caps : yes, no.
- ○ Deg-malig : 1, 2, 3.
- ○ Breast : left, right.
- ○ Breast-quad: left-up, left-low, right-up, right-low, central.
- ○ Irradiat : yes, no.

5.3 Auto Imports

This section demonstrates how to convert a .CSV (comma-separated values) file to the. ARFF (attribute-relation file format) using AUTO IMPORTformat.

1. Open Weka. If you're working in Weka, you have a built-in tool that will convert your .CSV files to the .ARFF format. (Refer Fig 5.3.1)

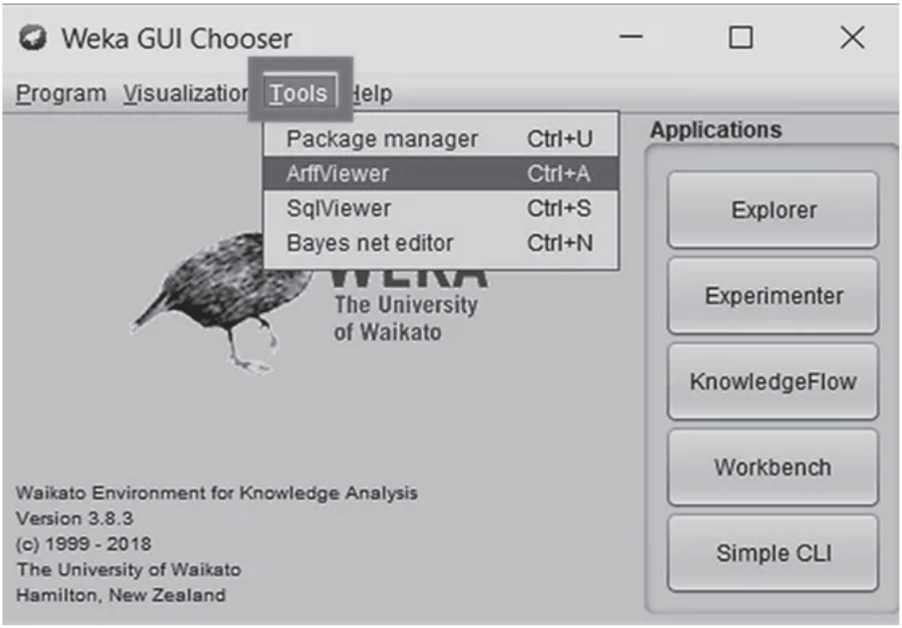


Fig. 5.3.1 - WEKA Tools

2. Click the Tools menu. It's in the menu bar at the top of the Weka window.

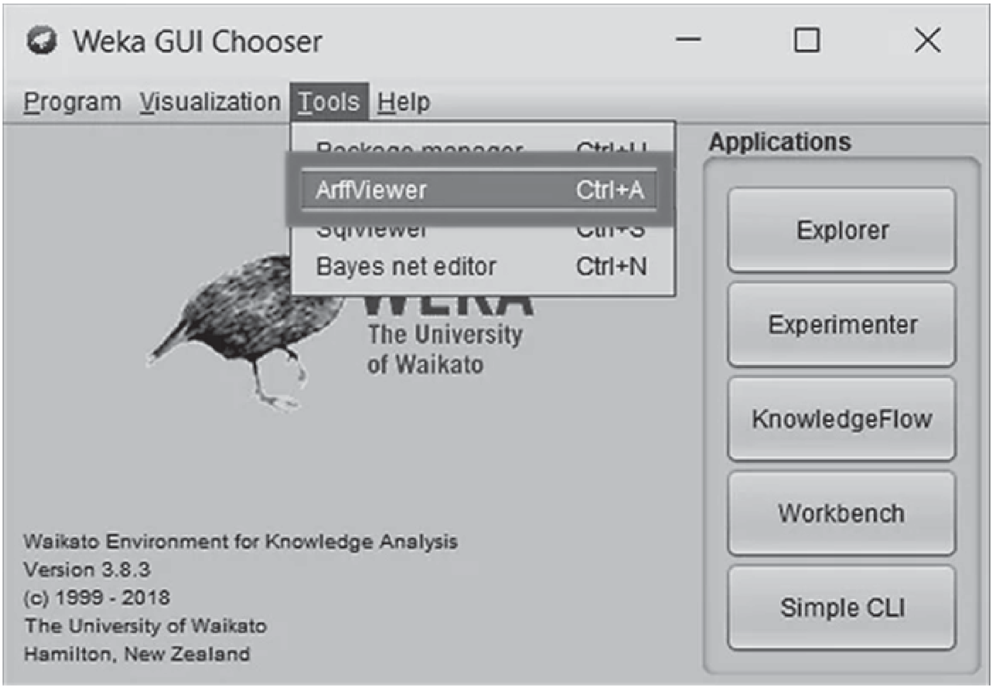


Fig. 5.3.2 - WEKA Tools ArffViewer

3. Click ArffViewer. This opens a blank window called "ARFF-Viewer"

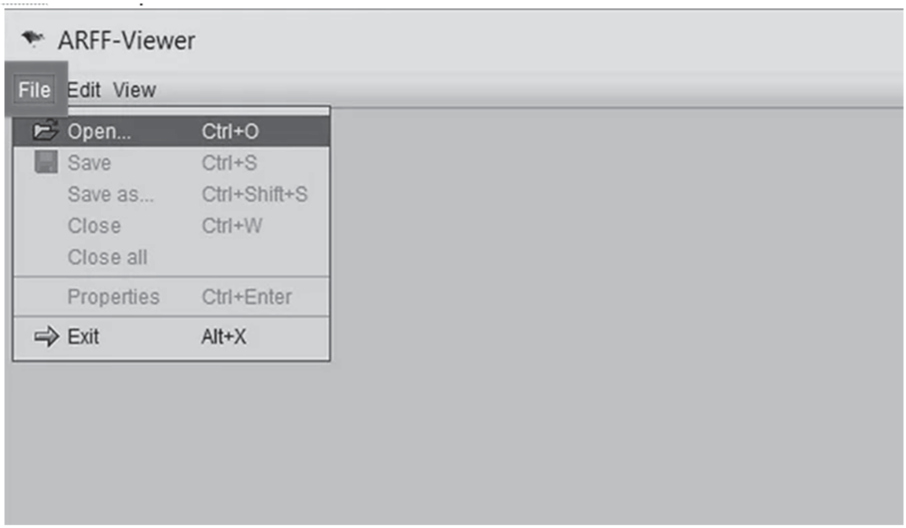
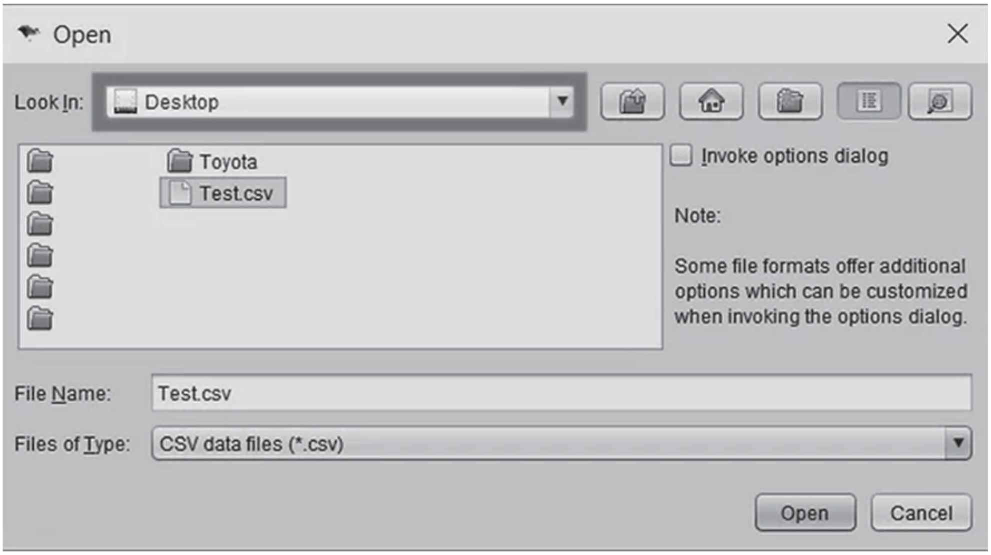


Fig. 5.3.3 - WEKA Tools ArffViewer

4. Click the File menu. It's at the top of the ARFF-Viewer window. Then Click Open. A file browser window will appear. Navigate to the folder that contains the .CSV file.



Open

Look In: Desktop

Toyota

Test.csv

□ Invoke options dialog

Note:

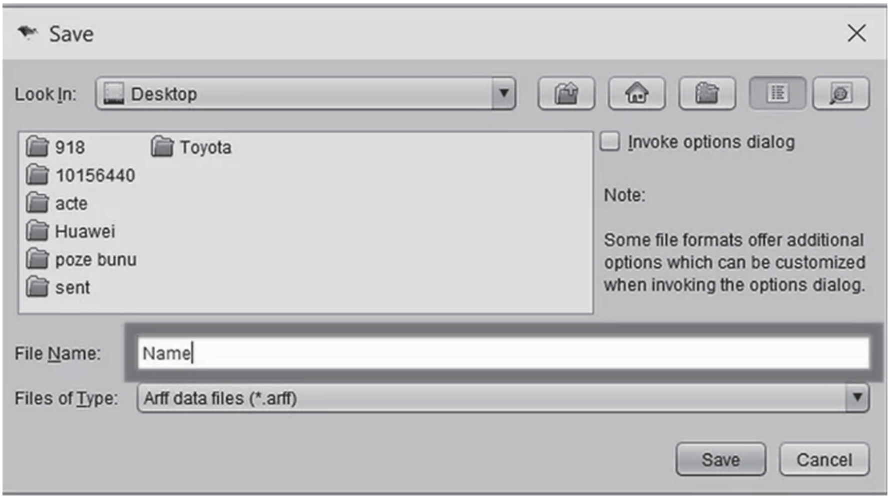
Some file formats offer additional options which can be customized when invoking the options dialog.

File Name: Test.csv

Files of Type: CSV data files (*.csv)

Fig. 5.3.4 - WEKA Tools ArffViewer Open

5. Select the .CSV and click Open. This opens the file in the viewer. Click the File menu. And Click Save As.



Save

Look In: Desktop

918
10156440
acte
Huawei
poze bunu
sent
Toyota

□ Invoke options dialog

Note:

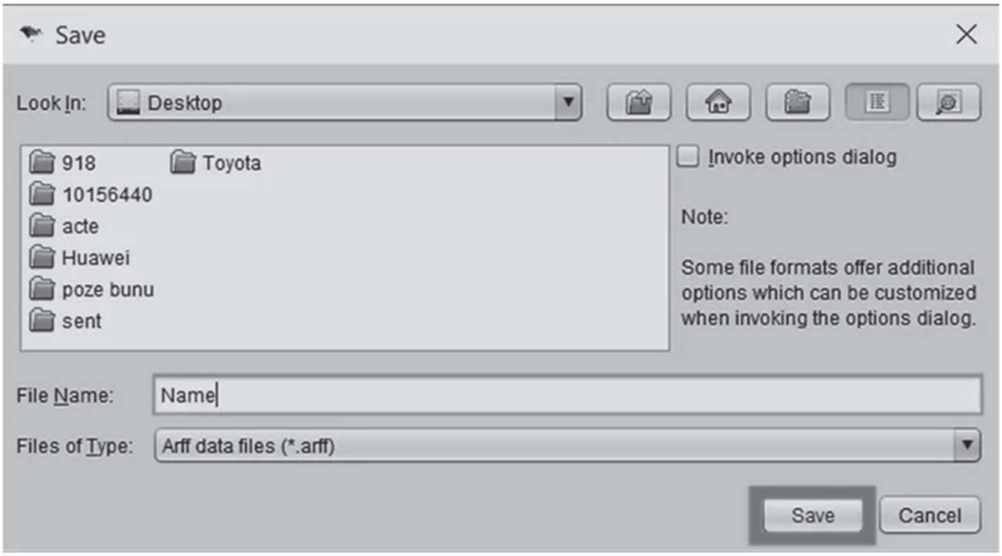
Some file formats offer additional options which can be customized when invoking the options dialog.

File Name: Name

Files of Type: Arff data files (*.arff)

Fig. 5.3.5 - WEKA Tools ArffViewer SaveAs

6. Name the file. The file name must end with ?.ARFF ? (e.g., mydata.ARFF). Click Save. The .CSV file is now converted to the .ARFF format.



Save

Look In: Desktop

918

Toyota

10156440

acte

Huawei

poze bunu

sent

□ Invoke options dialog

Note:

Some file formats offer additional options which can be customized when invoking the options dialog.

File Name: Name

Files of Type: Arff data files (*.arff)

Fig. 5.3.6 - WEKA Tools ArffViewer SaveAs

- Thus the .csv dataset can be auto imported to .arff format using WEKA tool.

Use Excel for Other File Formats

- If you have data in another format, load it in Microsoft Excel first.
- It is common to get data in another format such as CSV using a different delimiter or fixed width fields.
- Excel has powerful tools for loading tabular data in a variety of formats. Use these tools and first load your data into Excel.
- Once you have loaded your data into Excel, you can export it into CSV format. You can then work with it in Weka, either directly or by first converting it to ARFF format.

5.4 Algorithms

5.4.1 Learning Algorithms using WEKA

- A big benefit of using the Weka platform is the large number of supported machine learning algorithms.
- The more algorithms that you can try on your problem the more you will learn about your problem and likely closer you will get to discovering the one or few algorithms that perform best.
- In this section you will discover the machine learning algorithms supported by Weka.
- The different types of machine learning algorithms supported and key algorithms to try in Weka.

- Weka has a lot of machine learning algorithms. This is a great benefits of using Weka as a platform for machine learning.
- A down side is that it can be a little overwhelming to know which algorithms to use, and when.
- Also, the algorithms have names that may not be familiar to you, even if you know them in other contexts.
- In this section we will start off by looking at some well known algorithms supported by Weka.
- What we will learn in this post applies to the machine learning algorithms used across the Weka platform, but the Explorer is the best place to learn more about the algorithms as they are all available in one easy place.
- Open the Weka GUI Chooser.
- Click the "Explorer" button to open the Weka explorer.
- Open a dataset, such as the Pima Indians dataset from the data/diabetes.arff file in your Weka installation.
- Click "Classify" to open the Classify tab.
- The classify tab of the Explorer is where you can learn about the various different algorithms and explore predictive modeling.
- You can choose a machine learning algorithm by clicking the "Choose" button.

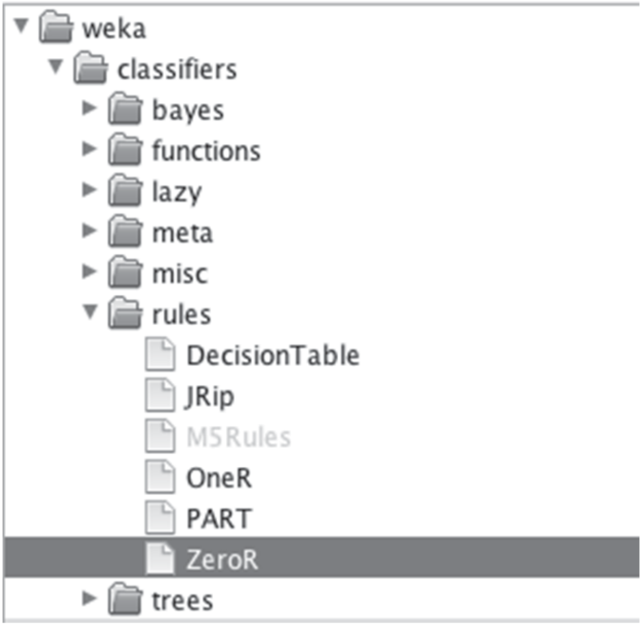


Fig 5.4.1 - WEKA explorer

- Clicking on the "Choose" button presents you with a list of machine learning algorithms to choose from. They are divided into a number of main groups :
    - **bayes** : Algorithms that use Bayes Theorem in some core way, like Naive Bayes.
    - **function** : Algorithms that estimate a function, like Linear Regression.
    - **lazy** : Algorithms that use lazy learning, like k-Nearest Neighbors.
    - **meta** : Algorithms that use or combine multiple algorithms, like Ensembles.
    - **misc** : Implementations that do not neatly fit into the other groups, like running a saved model.
    - **rules** : Algorithms that use rules, like One Rule.
    - **trees** : Algorithms that use decision trees, like Random Forest.
    - The tab is called "Classify" and the algorithms are listed under an overarching group called "Classifiers". Nevertheless, Weka supports both classification (predict a category) and regression (predict a numeric value) predictive modeling problems.

5.4.2 Selecting Algorithms - Clustering, Association - Rule Learners

- Generally, when working on a machine learning problem you cannot know which algorithm will be the best for your problem before hand.
- If you had enough information to know which algorithm would achieve the best performance, you probably would not be doing applied machine learning. You would be doing something else like statistics.
- The solution therefore is to try a suite of algorithms on your problem and see what works best.
- Some of the machine learning algorithms in Weka have non-standard names.
- You may already know the names of some machine learning algorithms, but feel confused by the names of the algorithms in Weka.

Linear Machine Learning Algorithms

- Linear algorithms assume that the predicted attribute is a linear combination of the input attributes.
- Linear regression : function.LinearRegression
- Logistic regression : function.Logistic

Nonlinear Machine Learning Algorithms

- Nonlinear algorithms do not make strong assumptions about the relationship between the input attributes and the output attribute being predicted.
- Naive Bayes : bayes.NaiveBayes

- Decision Tree (specifically the C4.5 variety) : trees.J48
- k-Nearest Neighbors (also called KNN : lazy.IBk
- Support Vector Machines (also called SVM) : functions.SMO
- Neural Network : functions.MultilayerPerceptron

Ensemble Machine Learning Algorithms

Ensemble methods combine the predictions from multiple models in order to make more robust predictions.

- Random Forest : trees.RandomForest
- Bootstrap Aggregation (also called Bagging) : meta.Bagging
- Stacked Generalization (also called Stacking or Blending) : meta.Stacking

Following are the FIVE algorithms that can be used for the classification and association using WEKA

- Logistic regression
- Naive bayes
- Decision tree
- k-Nearest neighbors
- Support vector machines

Common steps for executing classification and association

1 Start the Weka Explorer

2 Open the Weka GUI Chooser.

3 Click the "Explorer" button to open the Weka Explorer.

4 Load the Ionosphere dataset from the data/ionosphere.arff file.

5 Click "Classify" to open the Classify tab.

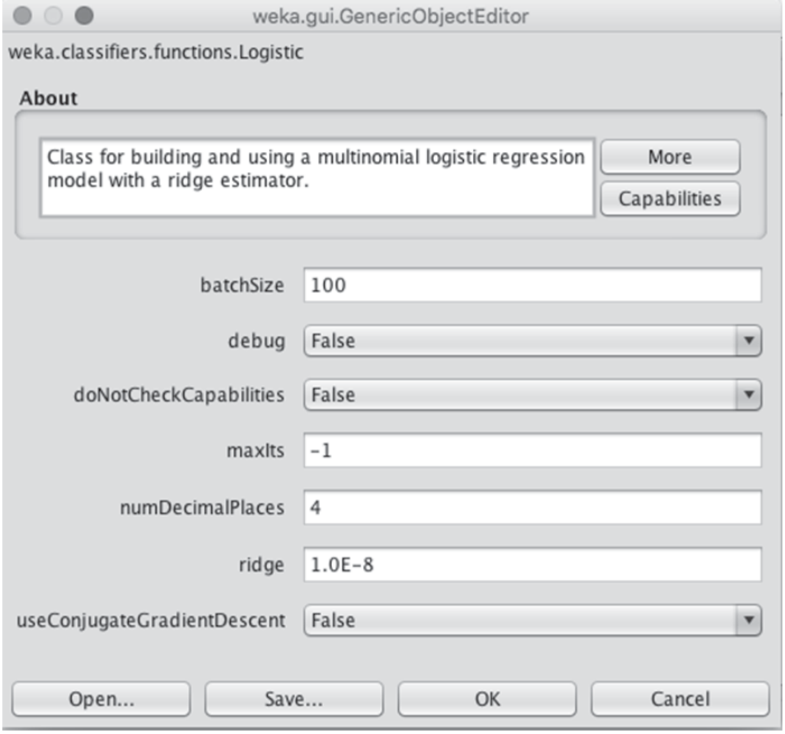
Logistic Regression

- Logistic regression is a binary classification algorithm.
- It assumes the input variables are numeric and have a Gaussian (bell curve) distribution.
- This last point does not have to be true, as logistic regression can still achieve good results if your data is not Gaussian.
- In the case of the Ionosphere dataset, some input attributes have a Gaussian-like distribution, but many do not.
- The algorithm learns a coefficient for each input value, which are linearly combined into a regression function and transformed using a logistic (s-shaped)

function. Logistic regression is a fast and simple technique, but can be very effective on some problems.

- The logistic regression only supports binary classification problems, although the Weka implementation has been adapted to support multi-class classification problems.
- Choose the logistic regression algorithm: (Refer Fig. 5.4.2)

1. Click the "Choose" button and select "Logistic" under the "functions" group.
2. Click on the name of the algorithm to review the algorithm configuration.



weka.gui.GenericObjectEditor

weka.classifiers.functions.Logistic

About

Class for building and using a multinomial logistic regression model with a ridge estimator.

batchSize 100

debug False

doNotCheckCapabilities False

maxIts -1

numDecimalPlaces 4

ridge 1.0E-8

useConjugateGradientDescent False

Fig. 5.4.2 Logistic regression

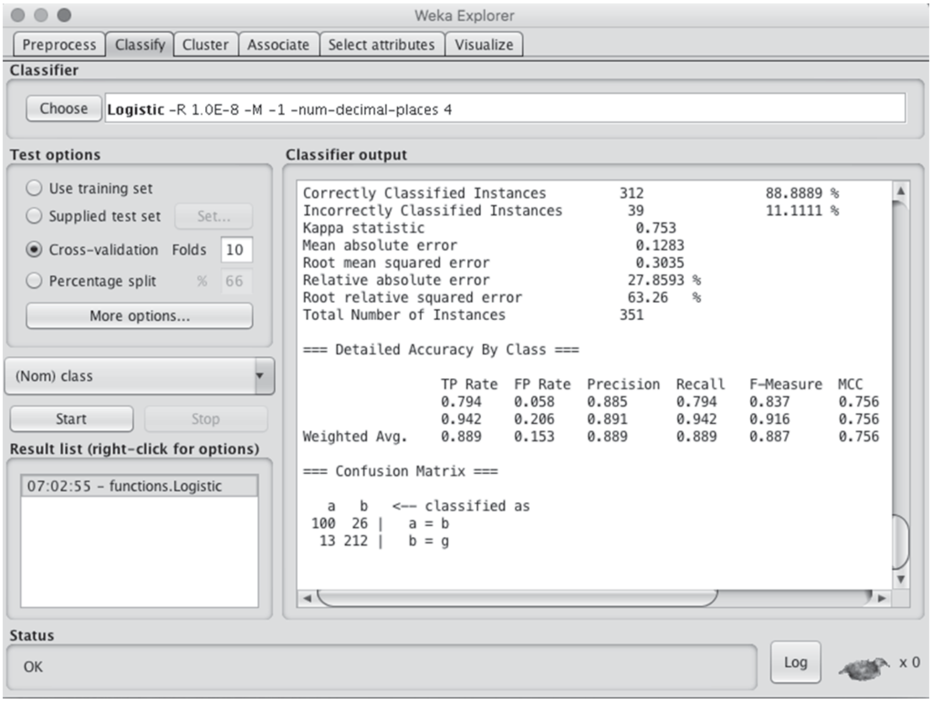
- The algorithm can run for a fixed number of iterations (maxIts), but by default will run until it is estimated that the algorithm has converged.
- The implementation uses a ridge estimator which is a type of regularization. This method seeks to simplify the model during training by minimizing the coefficients learned by the model. The ridge parameter defines how much pressure to put on the algorithm to reduce the size of the coefficients. Setting this to 0 will turn off this regularization.

1. Click "OK" to close the algorithm configuration.
2. Click the "Start" button to run the algorithm on the Ionosphere dataset.

- You can see that with the default configuration that logistic regression achieves an accuracy of 88%. (Refer Fig 5.4.3)

Naive Bayes

- Naive Bayes is a classification algorithm. Traditionally it assumes that the input values are nominal, although it numerical inputs are supported by assuming a distribution.
- Naive Bayes uses a simple implementation of Bayes Theorem (hence naive) where the prior probability for each class is calculated from the training data and assumed to be independent of each other (technically called conditionally independent).
- This is an unrealistic assumption because we expect the variables to interact and be dependent, although this assumption makes the probabilities fast and easy to calculate. Even under this unrealistic assumption, Naive Bayes has been shown to be a very effective classification algorithm.
- Naive Bayes calculates the posterior probability for each class and makes a prediction for the class with the highest probability. As such, it supports both binary classification and multi-class classification problems.



Classifier

Logistic -R 1.0E-8 -M -1 -num-decimal-places 4

Test options

○ Use training set
○ Supplied test set
● Cross-validation Folds 10
○ Percentage split % 66

(Nom) class

Result list (right-click for options)

07:02:55 - functions.Logistic

Classifier output

```
Correctly Classified Instances         312               88.8889 %
Incorrectly Classified Instances        39               11.1111 %
Kappa statistic                          0.753
Mean absolute error                      0.1283
Root mean squared error                  0.3035
Relative absolute error                 27.8593 %
Root relative squared error             63.26   %
Total Number of Instances              351

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC
                 0.794    0.058    0.885      0.794    0.837      0.756
                 0.942    0.206    0.891      0.942    0.916      0.756
Weighted Avg.    0.889    0.153    0.889      0.889    0.887      0.756

=== Confusion Matrix ===

   a   b   <-- classified as
 100  26 |   a = b
  13 212 |   b = g
```

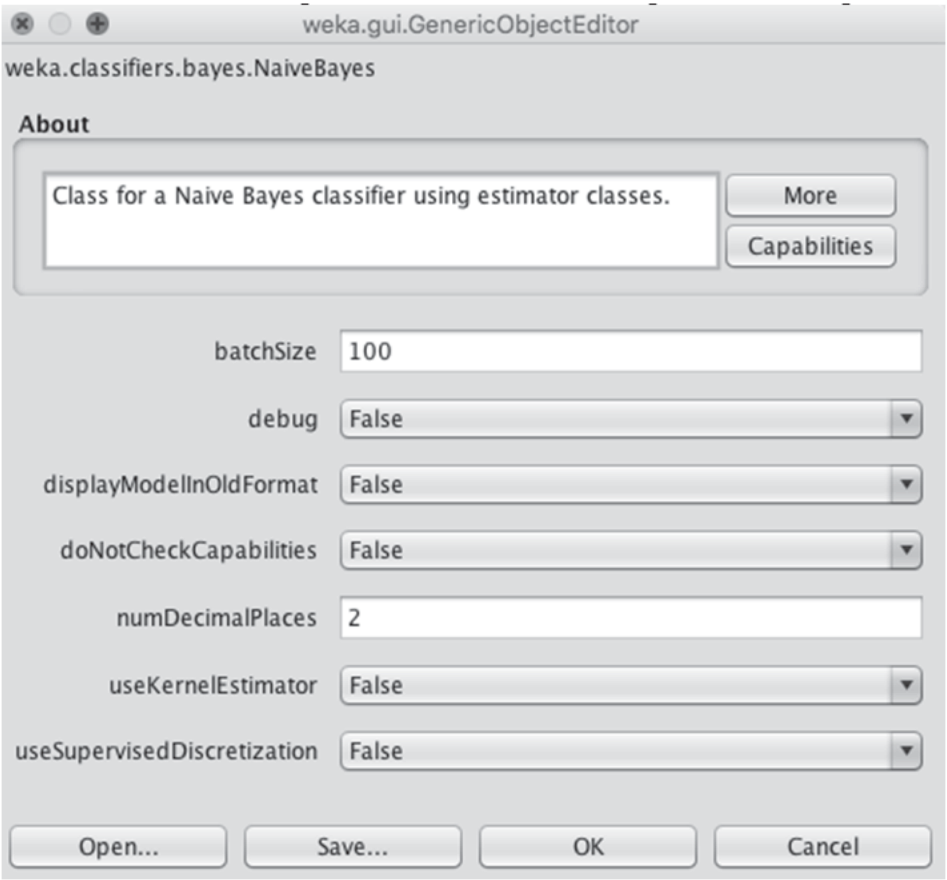
Status

OK

Fig. 5.4.3 - Output of logistic regression

- Choose the Naive Bayes algorithm:

  1. Click the "Choose" button and select "NaiveBayes" under the "bayes" group.
  2. Click on the name of the algorithm to review the algorithm configuration.



weka.gui.GenericObjectEditor

weka.classifiers.bayes.NaiveBayes

About

Class for a Naive Bayes classifier using estimator classes.

batchSize 100

debug False

displayModelInOldFormat False

doNotCheckCapabilities False

numDecimalPlaces 2

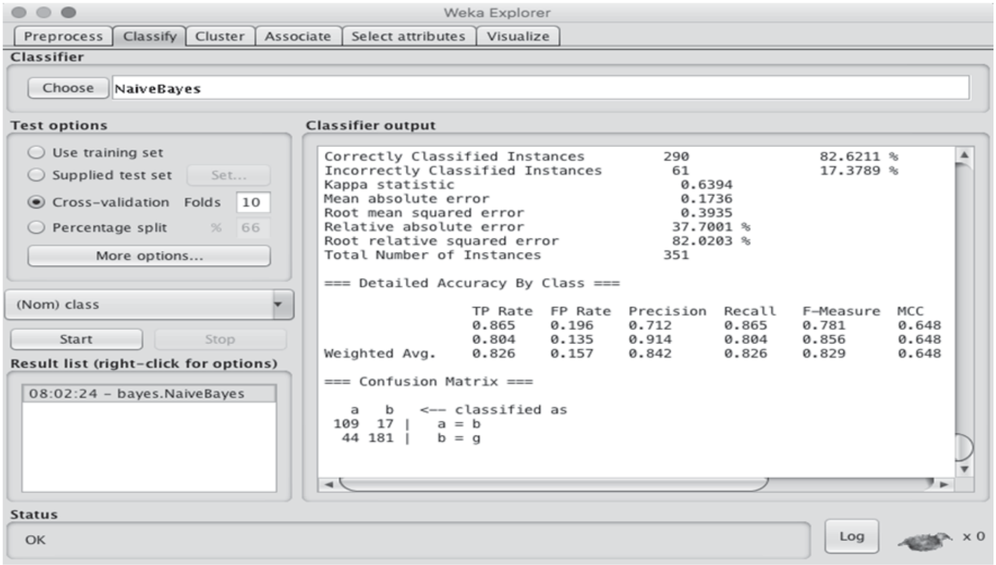
useKernelEstimator False

useSupervisedDiscretization False

Fig 5.4.4 - Naïve Bayes configuration

- By default a Gaussian distribution is assumed for each numerical attributes.
- You can change the algorithm to use a kernel estimator with the useKernelEstimator argument that may better match the actual distribution of the attributes in your dataset. Alternately, you can automatically convert numerical attributes to nominal attributes with the use SupervisedDiscretization parameter.

  1. Click "OK" to close the algorithm configuration.
  2. Click the "Start" button to run the algorithm on the Ionosphere dataset.
- You can see that with the default configuration that Naive Bayes achieves an accuracy of 82%.(Refer Fig. 5.4.5)



Classifier

NaiveBayes

Test options

○ Use training set

○ Supplied test set

● Cross-validation Folds 10

○ Percentage split % 66

(Nom) class

Result list (right-click for options)

08:02:24 - bayes.NaiveBayes

Classifier output

```
Correctly Classified Instances         290               82.6211 %
Incorrectly Classified Instances        61               17.3789 %
Kappa statistic                          0.6394
Mean absolute error                      0.1736
Root mean squared error                  0.3935
Relative absolute error                 37.7001 %
Root relative squared error             82.0203 %
Total Number of Instances              351

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC
                 0.865    0.196    0.712      0.865    0.781      0.648
                 0.804    0.135    0.914      0.804    0.856      0.648
Weighted Avg.    0.826    0.157    0.842      0.826    0.829      0.648

=== Confusion Matrix ===

   a   b   <-- classified as
 109  17 |   a = b
  44 181 |   b = g
```

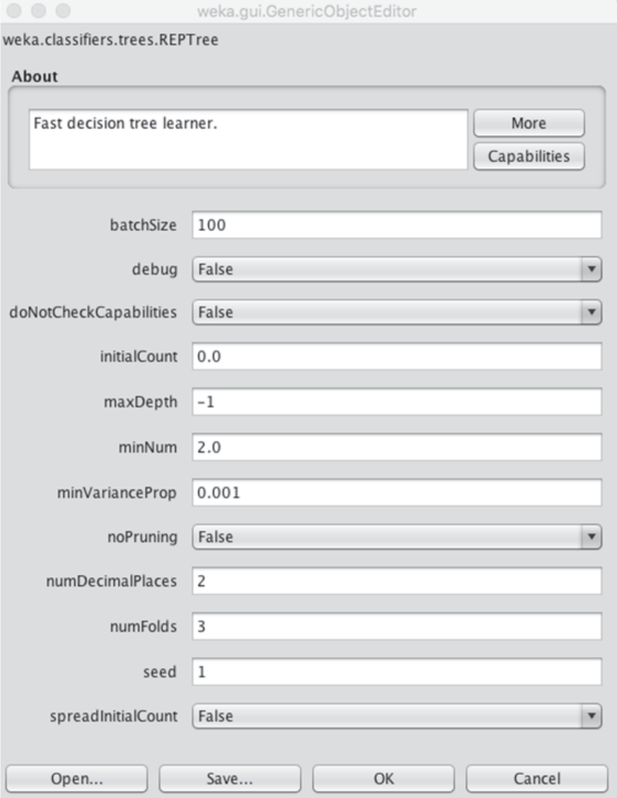
Status

OK

Fig 5.4.5 - Naïve Bayes output

Decision Tree

- Decision trees can support classification and regression problems.



weka.classifiers.trees.REPTree

About

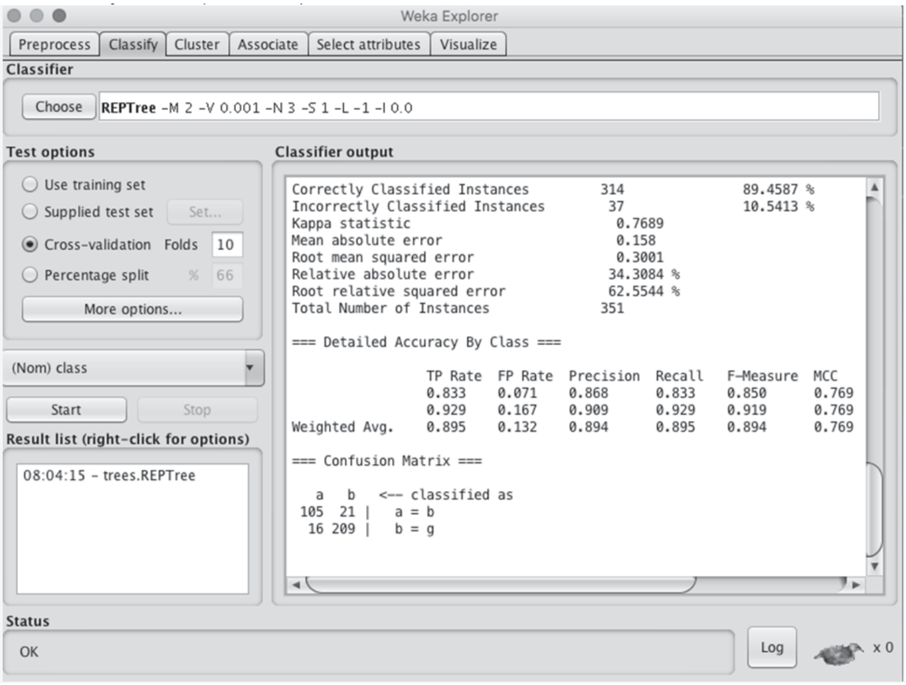
Fast decision tree learner.

batchSize	100
debug	False
doNotCheckCapabilities	False
initialCount	0.0
maxDepth	-1
minNum	2.0
minVarianceProp	0.001
noPruning	False
numDecimalPlaces	2
numFolds	3
seed	1
spreadInitialCount	False

Fig 5.4.6 - Decision tree configuration

- Decision trees are more recently referred to as Classification And Regression Trees (CART). They work by creating a tree to evaluate an instance of data, start at the root of the tree and moving town to the leaves (roots) until a prediction can be made. The process of creating a decision tree works by greedily selecting the best split point in order to make predictions and repeating the process until the tree is a fixed depth.
- After the tree is constructed, it is pruned in order to improve the model's ability to generalize to new data.
- Choose the decision tree algorithm :

  1. Click the "Choose" button and select "REPTree" under the "trees" group.
  2. Click on the name of the algorithm to review the algorithm configuration. (Refer Fig. 5.4.6)
- The depth of the tree is defined automatically, but a depth can be specified in the maxDepth attribute.
- You can also choose to turn of pruning by setting the noPruning parameter to True, although this may result in worse performance.
- The minNum parameter defines the minimum number of instances supported by the tree in a leaf node when constructing the tree from the training data.
- Click "OK" to close the algorithm configuration.
- Click the "Start" button to run the algorithm on the Ionosphere dataset.
- You can see that with the default configuration that the decision tree algorithm achieves an accuracy of 89%. (Refer Fig. 5.4.7 shown on next page)



REPTree -M 2 -V 0.001 -N 3 -S 1 -L -1 -I 0.0

Folds 10

% 66

(Nom) class

08:04:15 - trees.REPTree

OK

```
Correctly Classified Instances         314               89.4587 %
Incorrectly Classified Instances        37               10.5413 %
Kappa statistic                          0.7689
Mean absolute error                      0.158
Root mean squared error                  0.3001
Relative absolute error                 34.3084 %
Root relative squared error             62.5544 %
Total Number of Instances              351

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC
                 0.833    0.071    0.868      0.833    0.850      0.769
                 0.929    0.167    0.909      0.929    0.919      0.769
Weighted Avg.    0.895    0.132    0.894      0.895    0.894      0.769

=== Confusion Matrix ===

   a   b   <-- classified as
 105  21 |   a = b
  16 209 |   b = g
```

Fig. 5.4.7 Decision Tree Output

- Another more advanced decision tree algorithm that you can use is the C4.5 algorithm, called J48 in Weka.

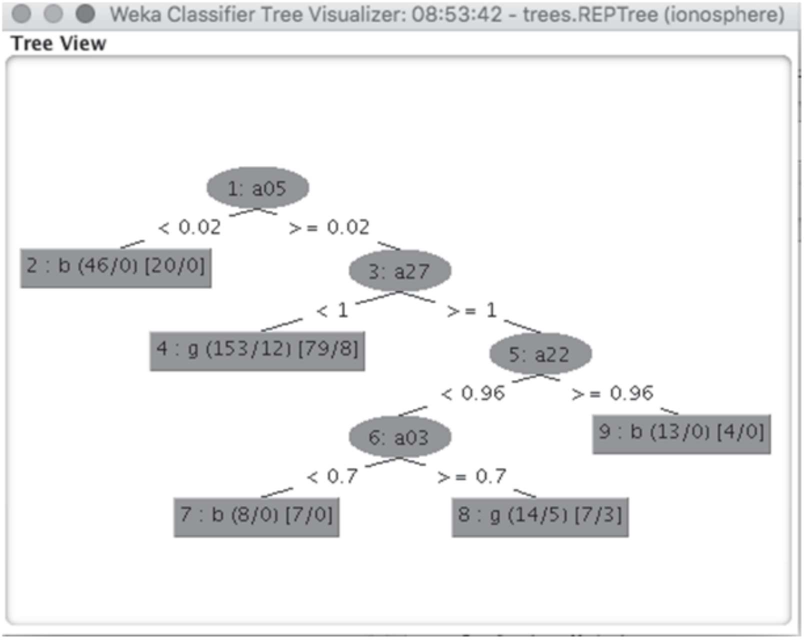


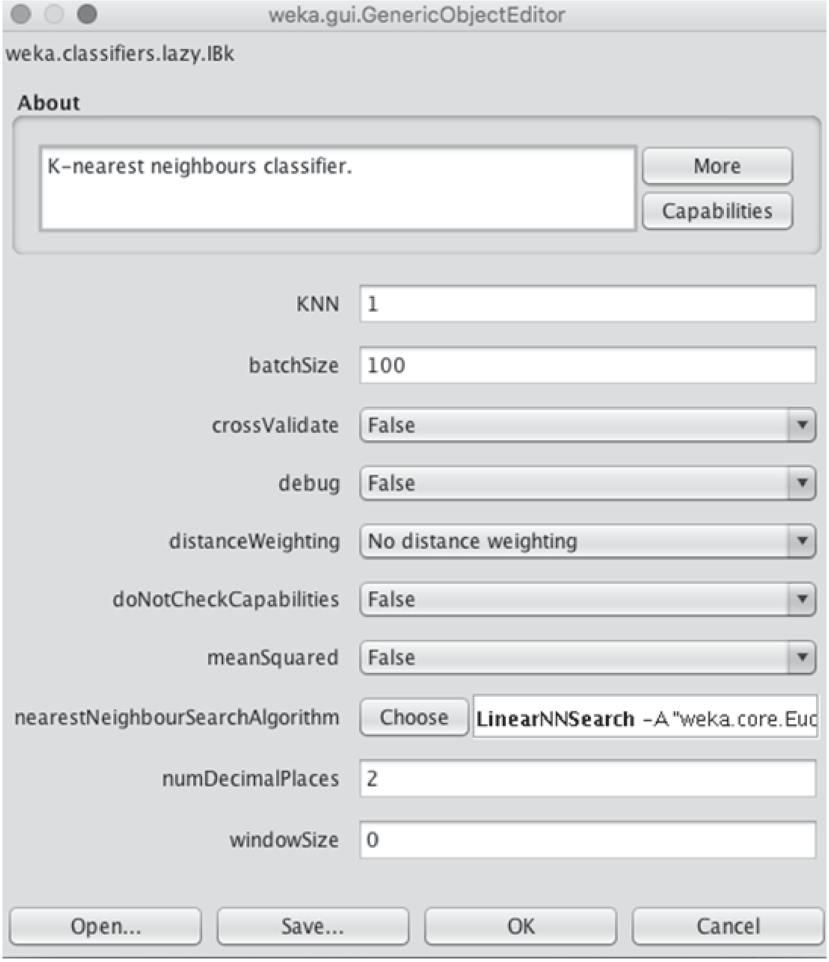
Fig. 5.4.8 Decision Tree Output

- You can review a visualization of a decision tree prepared on the entire training data set by right clicking on the "Result list" and clicking "Visualize Tree".

k-Nearest Neighbors

- The k-nearest neighbors algorithm supports both classification and regression. It is also called kNN for short.
- It works by storing the entire training dataset and querying it to locate the k most similar training patterns when making a prediction. As such, there is no model other than the raw training dataset and the only computation performed is the querying of the training dataset when a prediction is requested.
- It is a simple algorithm, but one that does not assume very much about the problem other than that the distance between data instances is meaningful in making predictions. As such, it often achieves very good performance.
- When making predictions on classification problems, KNN will take the mode (most common class) of the k most similar instances in the training dataset.
- Choose the k-Nearest Neighbors algorithm :

  1. Click the "Choose" button and select "IBk" under the "lazy" group.
  2. Click on the name of the algorithm to review the algorithm configuration.



weka.classifiers.lazy.IBk

K-nearest neighbours classifier.

KNN 1

batchSize 100

crossValidate False

debug False

distanceWeighting No distance weighting

doNotCheckCapabilities False

meanSquared False

nearestNeighbourSearchAlgorithm LinearNNSearch -A "weka.core.Euc

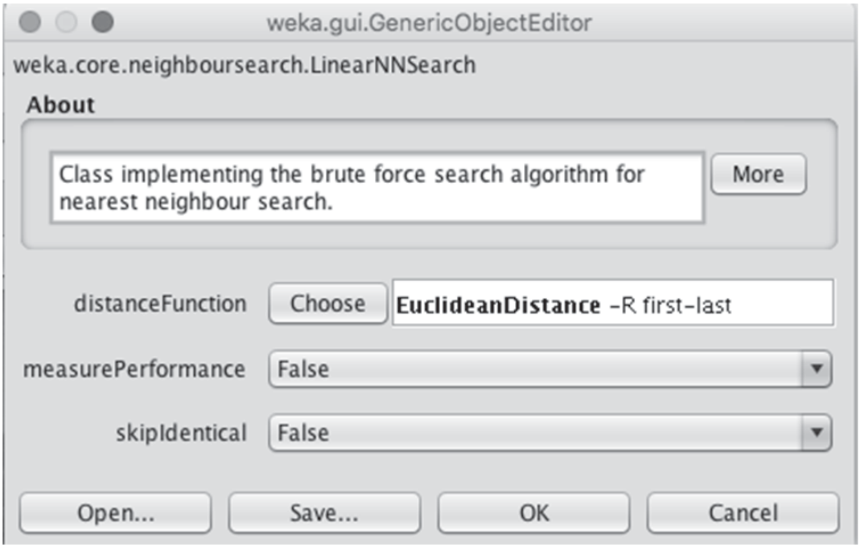
numDecimalPlaces 2

windowSize 0

Fig. 5.4.9 - k-Nearest neighbors configuration

- The size of the neighborhood is controlled by the k parameter.
- For example, if k is set to 1, then predictions are made using the single most similar training instance to a given new pattern for which a prediction is requested. Common values for k are 3, 7, 11 and 21, larger for larger dataset sizes. Weka can automatically discover a good value for k using cross validation inside the algorithm by setting the crossValidate parameter to True.
- Another important parameter is the distance measure used. This is configured in the nearestNeighbourSearchAlgorithm which controls the way in which the training data is stored and searched.
- The default is a LinearNNSearch. Clicking the name of this search algorithm will provide another configuration window where you can choose a distanceFunction parameter. By default, Euclidean distance is used to calculate the distance between instances, which is good for numerical data with the same scale. Manhattan

distance is good to use if your attributes differ in measures or type. It is a good idea to try a suite of different k values and distance measures on your problem and see what works best.



weka.gui.GenericObjectEditor

weka.core.neighboursearch.LinearNNSearch

About

Class implementing the brute force search algorithm for nearest neighbour search.

distanceFunction: **EuclideanDistance** -R first-last

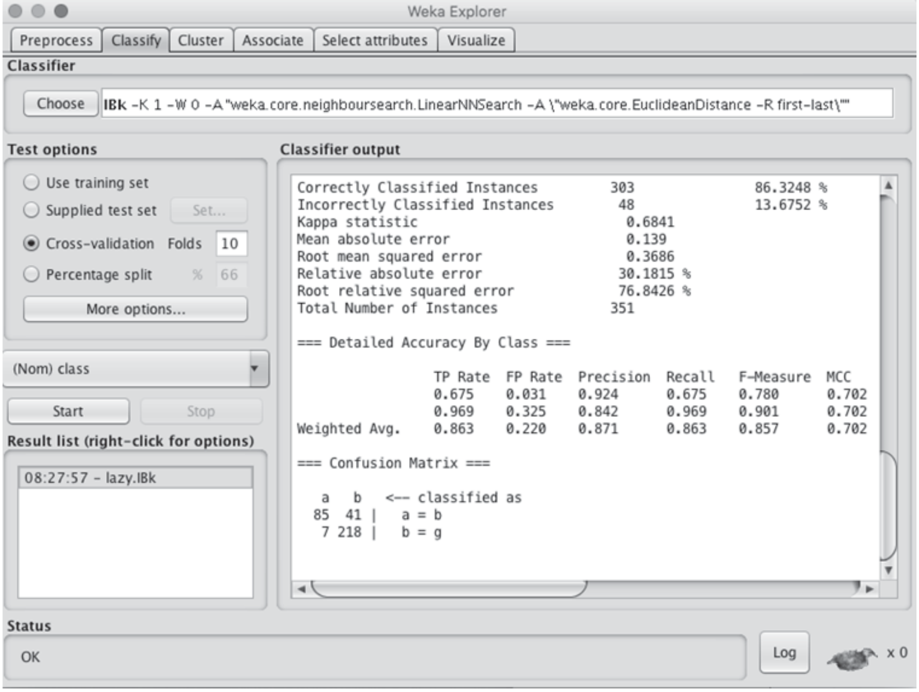
measurePerformance: False

skipIdentical: False

Fig. 5.4.10 - k-Nearest Neighbors configuration

- Click "OK" to close the algorithm configuration.

  Click the "Start" button to run the algorithm on the Ionosphere dataset.



Classifier: **IBk** -K 1 -W 0 -A "weka.core.neighboursearch.LinearNNSearch -A \"weka.core.EuclideanDistance -R first-last\""

Test options

○ Use training set

○ Supplied test set Set...

● Cross-validation Folds 10

○ Percentage split % 66

(Nom) class

Result list (right-click for options)

08:27:57 - lazy.IBk

Classifier output

```
Correctly Classified Instances         303               86.3248 %
Incorrectly Classified Instances        48               13.6752 %
Kappa statistic                          0.6841
Mean absolute error                      0.139
Root mean squared error                  0.3686
Relative absolute error                 30.1815 %
Root relative squared error             76.8426 %
Total Number of Instances              351

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC
                 0.675    0.031    0.924      0.675    0.780      0.702
                 0.969    0.325    0.842      0.969    0.901      0.702
Weighted Avg.    0.863    0.220    0.871      0.863    0.857      0.702

=== Confusion Matrix ===

   a   b   <-- classified as
  85  41 |   a = b
   7 218 |   b = g
```

Status

OK

Fig 5.4.11 - k-Nearest Neighbors result

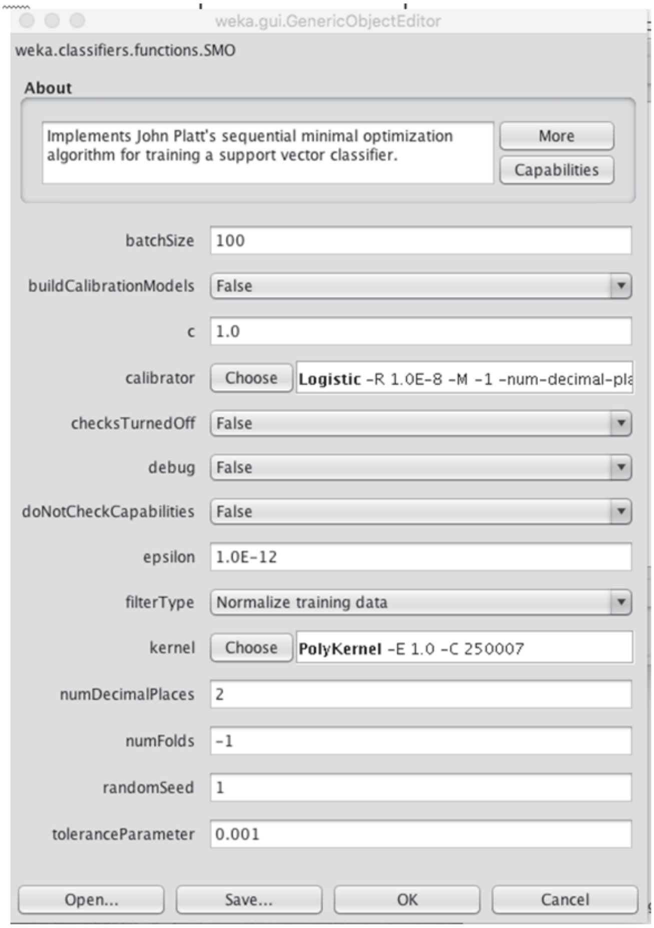
- You can see that with the default configuration that the kNN algorithm achieves an accuracy of 86%.

Support Vector Machines (SVM)

- Support vector machines were developed for binary classification problems, although extensions to the technique have been made to support multi-class classification and regression problems. The algorithm is often referred to as SVM for short.
- SVM was developed for numerical input variables, although will automatically convert nominal values to numerical values. Input data is also normalized before being used.
- SVM work by finding a line that best separates the data into the two groups. This is done using an optimization process that only considers those data instances in the training dataset that are closest to the line that best separates the classes. The instances are called support vectors, hence the name of the technique.
- In almost all problems of interest, a line cannot be drawn to neatly separate the classes, therefore a margin is added around the line to relax the constraint, allowing some instances to be misclassified but allowing a better result overall.
- Finally, few datasets can be separated with just a straight line. Sometimes a line with curves or even polygonal regions need to be marked out. This is achieved with SVM by projecting the data into a higher dimensional space in order to draw the lines and make predictions. Different kernels can be used to control the projection and the amount of flexibility in separating the classes.
- Choose the SVM algorithm :

  1. Click the "Choose" button and select "SMO" under the "function" group.
  2. Click on the name of the algorithm to review the algorithm configuration.

- SMO refers to the specific efficient optimization algorithm used inside the SVM implementation,which stands for Sequential Minimal Optimization.
- The C parameter, called the complexity parameter in Weka controls how flexible the process for drawing the line to separate the classes can be. A value of 0 allows no violations of the margin, whereas the default is 1.
- A key parameter in SVM is the type of Kernel to use. The simplest kernel is a Linear kernel that separates data with a straight line or hyperplane. The default in Weka is a Polynomial Kernel that will separate the classes using a curved or wiggly line, the higher the polynomial, the more wiggly (the exponent value).
- A popular and powerful kernel is the RBF Kernel or Radial Basis Function Kernel that is capable of learning closed polygons and complex shapes to separate the classes.



weka.gui.GenericObjectEditor

weka.classifiers.functions.SMO

About

Implements John Platt's sequential minimal optimization algorithm for training a support vector classifier.

batchSize: 100

buildCalibrationModels: False

c: 1.0

calibrator: **Logistic** -R 1.0E-8 -M -1 -num-decimal-pl

checksTurnedOff: False

debug: False

doNotCheckCapabilities: False

epsilon: 1.0E-12

filterType: Normalize training data

kernel: **PolyKernel** -E 1.0 -C 250007

numDecimalPlaces: 2

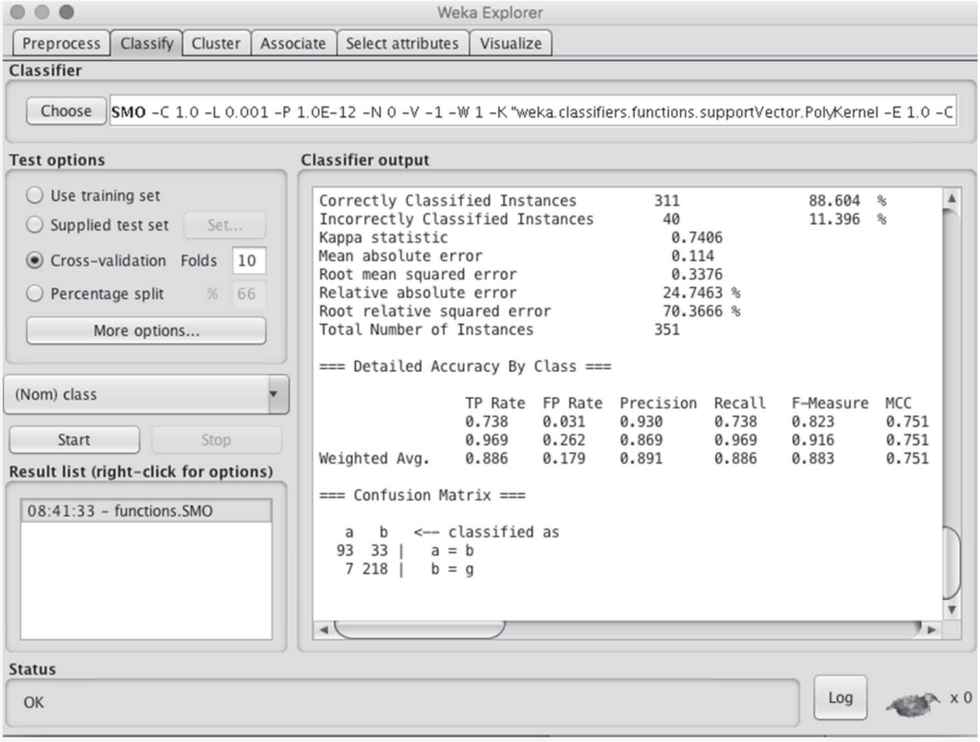
numFolds: -1

randomSeed: 1

toleranceParameter: 0.001

Fig. 5.4.12 SVM algorithm configuration

- It is a good idea to try a suite of different kernels and C (complexity) values on your problem and see what works best.
- Click "OK" to close the algorithm configuration.
- Click the "Start" button to run the algorithm on the Ionosphere dataset.
- You can see that with the default configuration that the SVM algorithm achieves an accuracy of 88%.



Classifier: SMO -C 1.0 -L 0.001 -P 1.0E-12 -N 0 -V -1 -W 1 -K "weka.classifiers.functions.supportVector.PolyKernel -E 1.0 -C

Test options:

○ Use training set

○ Supplied test set

● Cross-validation Folds 10

○ Percentage split % 66

(Nom) class

Result list: 08:41:33 - functions.SMO

Status: OK

```
Correctly Classified Instances         311               88.604  %
Incorrectly Classified Instances        40               11.396  %
Kappa statistic                          0.7406
Mean absolute error                      0.114
Root mean squared error                  0.3376
Relative absolute error                 24.7463 %
Root relative squared error             70.3666 %
Total Number of Instances              351

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC
                 0.738    0.031    0.930      0.738    0.823      0.751
                 0.969    0.262    0.869      0.969    0.916      0.751
Weighted Avg.    0.886    0.179    0.891      0.886    0.883      0.751

=== Confusion Matrix ===

   a   b   <-- classified as
  93  33 |   a = b
   7 218 |   b = g
```

Fig. 5.4.13 SVM algorithm output

Summary

- Weka is a collection of machine learning algorithms for data mining tasks. The algorithms can either be applied directly to a dataset or called from your own Java code. Weka contains tools for data pre-processing, classification, regression, clustering, association rules, and visualization.

Review Questions

1. *Write a note of WEKA tool.*
2. *Describe in detail, the basic components of WEKA tool.*
3. *Write a short note on any THREE - a. Explorer b. Experimenter c. KnowledgeFlow d. Workbenche e. Simple CLI.*
4. *Write a note on Cluster formation of data in WEKA.*
5. *How the Classifiers are selected in WEKA ?*
6. *Summarize the six tabs in WEKA Explorer.*
7. *Elaborate on Attribute Relation File Format (ARFF).*
8. *Write a note on Binary Classification Datasets.*

9. *Describe in short*

a. Pima Indians Onset of Diabetes : (diabetes.arff)

b. Breast Cancer: (breast-cancer.arff)

c. Ionosphere (ionosphere.arff)

10. *Write a note on Multi-Class Classification Datasets*

11. *Write a short note on*

a. Iris Flowers Classification: (iris.arff)

b. Large Soybean Database: (soybean.arff)

c. Glass Identification : (glass.arff)

12. *Write a short note on following using WEKA*

a. Logistic Regression

b. Naive Bayes

c. Decision Tree

d. k-Nearest Neighbors

e. Support Vector Machines

Two Marks Questions with Answers

Q.1 What is WEKA ?

Ans. :
- WEKA stands for Waikato Environment for Knowledge Analysis (WEKA).
- WEKA was developed at the University of Waikato in New Zealand, and the name stands for Waikato Environment for Knowledge Analysis.
- The system is written in Java and distributed under the terms of the GNU General Public License.
- WEKA is a collection of machine learning algorithms for data mining tasks.

Q.2 What are the system requirements of WEKA.

Ans. : • WEKA supports both 32-bit and 64-bit Operating Systems (including Windows, Linux and Mac OS).

- The current version of WEKA needs JVM (Java Virtual Machine). The following matrix shows which minimum version of Java is necessary to run a specific WEKA version. The latest official releases of WEKA require Java 8 or later.

Q.3 What are the different versions of WEKA available for usage ?

Ans. : • There are two versions of WEKA to download with self-extracting executable(s) :

- WEKA stable version (3.8 is the latest)
  - ○ This branch of WEKA receives bug fixes only, although new features may become available in packages.

- WEKA development version (3.9 is the latest).
  - This is the trunk of WEKA and continues from the stable-3.8 code line. It receives both bug fixes and new features.

Q.4 Write the names of basic components of WEKA GUI interface.

Ans. : • The WEKA GUI interface has the basic components as

- Explorer
- Experimenter
- KnowledgeFlow
- Workbench
- Simple CLI

Q.5 What are the WEKA data formats ?

Ans. : • WEKA uses the attribute relation file format for data analysis, by default. But listed below are some formats that WEKA supports, from where data can be imported :

- CSV
- ARFF
- Database using ODBC-JDBC

Q.6 What are the different ways of getting data in WEKA tool ?

Ans. : • WEKA tool can load data using different menthods, these are :

- Loading data by creating new dataset
- Loading data by importing from files
- Loading data by connecting to a database
- Loading data by opening a URL

Q.7 List any the four tabs in WEKA Explorer.

Ans. :

- Pre-process
- Classify
- Cluster
- Associate
- Select attributes
- Visualize

Q.8 What is WEKA datasets ?

Ans. : • All datasets are in the WEKA native ARFF file format and can be loaded directly into WEKA, meaning you can start developing practice models immediately.

- There are some datasets like breast cancer datasets, diabetes datasets available for research purpose.

Q.9 What are binary classification datasets ?

Ans. : • Binary classification is where the output variable to be predicted is nominal comprised of two classes.

- There are three standard binary classification problems in the data/ directory that you can focus on :
  - Pima Indians Onset of Diabetes : (diabetes.arff)
  - Breast Cancer : (breast-cancer.arff)
  - Ionosphere (ionosphere.arff)

Q.10 What are multi-class classification datasets ?

Ans. : • There are many classification type problems, where the output variable has more than two classes. These are called multi-class classification problems.

- Three standard multi-class classification problems in the data/ directory that you can focus on are :
  - Iris Flowers Classification : (iris.arff)
  - Large Soybean Database : (soybean.arff)
  - Glass Identification : (glass.arff)

Q.11 What is auto import in WEKA

Ans. : • Auto import is a data loading facility in WEKA tool for the data source that is not in standard ARFF format.

- The WEKA tool converts a .CSV (comma-separated values) file to the .ARFF (attribute-relation file format) using AUTO IMPORTformat.

Q.12 Write the names of any FOUR types of algorithms supported by WEKA.

Ans. : • **Bayes** : Algorithms that use Bayes Theorem in some core way, like Naive Bayes.

- **Function** : Algorithms that estimate a function, like Linear Regression.
- **Lazy** : Algorithms that use lazy learning, like k-Nearest Neighbors.
- **Meta** : Algorithms that use or combine multiple algorithms, like Ensembles.
- **Misc** : Implementations that do not neatly fit into the other groups, like running a saved model.
- **Rules** : Algorithms that use rules, like One Rule.
- **trees** : Algorithms that use decision trees, like Random Forest.

Q.13 Write the names of any four algorithms that can be used for the classification& association using WEKA.

Ans. :
- Logistic Regression.
- Naive Bayes
- Decision Tree
- k-Nearest Neighbors
- Support Vector Machines

Q.14 What are linear machine learning algorithms supported by WEKA ?

Ans. : • Linear algorithms assume that the predicted attribute is a linear combination of the input attributes

- Linear Regression : function.LinearRegression
- Logistic Regression : function.Logistic

Q.15 What are nonlinear machine learning algorithms ?

Ans. : • Nonlinear algorithms do not make strong assumptions about the relationship between the input attributes and the output attribute being predicted.

- Examples are Naive Bayes : bayes.NaiveBayes, Decision Tree (specifically the C4.5 variety) : trees.J48, k-Nearest Neighbors (also called KNN: lazy.IBk , Support Vector Machines (also called SVM): functions.SMO, Neural Network: functions.MultilayerPerceptron

Q.16 What are ensemble machine learning algorithms ?

Ans. : • Ensemble methods combine the predictions from multiple models in order to make more robust predictions.

- ○ Random Forest: trees RandomForest
- ○ Bootstrap Aggregation (also called Bagging): meta.Bagging
- ○ Stacked Generalization (also called Stacking or Blending): meta.Stacking

Q.17 What is Naive Bayes ?

Ans. : • Naive Bayes is a classification algorithm. Traditionally it assumes that the input values are nominal, although it numerical inputs are supported by assuming a distribution.

- Naive Bayes uses a simple implementation of Bayes Theorem (hence naive) where the prior probability for each class is calculated from the training data and assumed to be independent of each other (technically called conditionally independent).

Q.18 Describe Decision tree.

Ans. : • Decision trees can support classification and regression problems.

- Decision trees are more recently referred to as Classification And Regression Trees (CART). They work by creating a tree to evaluate an instance of data, start at the root of the tree and moving town to the leaves (roots) until a prediction can be made

Q.19 What is k-nearest neighbors algorithm ?

Ans. : • The k-nearest neighbors algorithm supports both classification and regression. It is also called kNN for short.

- It works by storing the entire training dataset and querying it to locate the k most similar training patterns when making a prediction. As such, there is no model other than the raw training dataset and the only computation performed is the querying of the training dataset when a prediction is requested.

❑❑❑

Notes

SOLVED MODEL QUESTION PAPER

[As per New Syllabus]

Data Warehousing and Data Mining

Semester - VI (CSE)

Time : Three Hours] **[Maximum Marks : 100**

Answer ALL Questions

PART A - (10 $\times$ 2 = 20 Marks)

Q.1 *What is the role of sourcing, acquisition, clean up, and transformation tools ?* **[Refer section 1.1.3]**

Q.2 *List nine decision steps in the design of a data warehouse ?* **[Refer section 1.2.2]**

Q.3 *What is CBA ?* **[Refer section 3.5.1]**

Q.4 *What is market basket analysis ?* **[Refer section 3.1]**

Q.5 *What is Business Intelligence ?* **[Refer section 2.5.1]**

Q.6 *What is mean and how it is calculated ?* **[Refer section 2.7]**

Q.7 *What are the two approaches of tree pruning ?* **[Refer section 4.1.2]**

Q.8 *Explain rule extraction from a decision tree.* **[Refer section 4.1.4]**

Q.9 *What are the different ways of getting data in WEKA tool ?* **[Refer section 5.1.3]**

Q.10 *Write the names of any FOUR algorithms supported by WEKA.* **[Refer section 5.3.1]**

PART B - (5 $\times$ 13 = 65 Marks)

Q.11 a) *i) Explain the concept of metadata.* **[Refer section 1.1.4]** **[7]**

ii) Explain star schema. **[Refer section 1.5.2]** **[6]**

OR

b) *Explain database architectures for parallel processing.* **[Refer section 1.3]** **[13]**

Q.12 a) *Explain various approaches of data visualization.* **[Refer section 2.13]** **[13]**

OR

b) *i) Explains various techniques that can be adapted for smoothing of the data.* **[Refer section 2.9.2]** **[7]**

ii) Explain nominal and binary attributes. **[Refer section 2.6]** **[6]**

Q.13 a) *Explain Apriori algorithm with example.* **[Refer section 3.1.1]** **[13]**

OR

b) *Explain different methods for associative classification.* **[Refer section 3.5.1]** **[13]**

Q.14 a) *i) Explain k-means as a centroid-based technique.* **[Refer section 4.2.3]** **[7]**

ii) Write and K-means partitioning algorithm **[Refer section 4.2.3]** **[6]**

OR

b) *i) Explain the Outlier and Outlier analysis.* **[Refer section 4.12.10]** **[7]**

ii) Describe the challenges of outlier detection. **[Refer section 4.2.10]** **[6]**

Q.15 a) *i) Write a note of WEKA tool.* **[Refer section 5.2.3]** **[7]**

ii) Write a note on binary classification datasets. **[Refer section 5.2.2]** **[6]**

OR

b) *i) Elaborate on Attribute Relation File Format (ARFF).* **[Refer section 5.2.2]** **[7]**

ii) Write a note on Iris plant dataset. **[Refer section 5.2.3]** **[6]**

PART - C (1 × 15 = 15 Marks)

Q.16 a) *i) Explain all OLAP operations.* **[Refer section 1.5]**

OR

b) *Explain selection of algorithms using WEKA tool for - Clustering, Association and Rule learners* **[Refer section 5.3.2]**

❑❑❑