Course Pack

Data Warehousing & Mining (CBCS 2019)

Course: BCA

Course Code: 602

Semester: VI

Year: 2021-22

Course Leader &

Course Instructor: **Ms. Neha Sabharwal**

Forwarded by: **Prog. Co-ordinator (I/C)**	**Forwarded by:** **HOD**	**Approved by: Director**

BHARATI VIDYAPEETH

(DEEMED TO BE UNIVERSITY)

Institute of management& research, New Delhi

An ISO 9001:2015 Certified Institute

"A+" grade accreditation by NAAC

Note : " Strictly for internal academic use only "

Table of Content

Institute of Management and Research, New Delhi

Course BCA -VI SEM ***Academic Year 2021-22***

Course Title: Data Warehousing & Data Mining

Course Code: 601 **Credits: 4** **Credit hours: 40**

Course Overview:

One of the core objectives of the computer science is data handling. As we are deluged by data which comes from different areas like medical, demographic, financial, marketing & scientific, there is a need to organize and store this data efficiently with some better architecture. Also through this paper students are expected to enhance their skills to write efficient algorithms and analyzing, classifying and summarizing data as well as they will be acquainted with the application of statistics, visualization, artificial intelligence and machine learning is the field.

Course Objective:

One of the core objective of computer science is data handling. As we are overloaded by data which comes from different areas like medical, demographic, financial, marketing & scientific, there is a need to organize and store this data efficiently with some better architecture. Also through this paper students are expected to enhance their skills to write efficient algorithms for analyzing, classifying and summarizing data as well as they will be acquainted with the application of statistics, visualization, artificial intelligence and machine learning in this field.

Learning Outcome:

After undergoing this course, the student will be able to:

CO1.Understand the concept of data warehousing and mining, it need and usage.

CO2.Understand various operations OLAP & OLTP operations.

CO3. Differentiate various types of OLAP servers.

CO4.Understand various data preprocessing techniques.

CO5.Students will be able to understand need of data mining, knowledge discovery database process and mining architecture.

CO6: able to Demonstrate concepts of association rule mining, classification, prediction, cluster analysis, data mining and confidentiality.

List of topics/modules

Topic/Module	Contents/ concepts
UNIT-1 Introduction to Data warehousing:	
Data Warehousing, Difference between operational database system and data warehouse, Data Warehouse Users, Benefits of Data Warehousing, Metadata, Classification of Metadata, and Importance of Metadata. Data Marts, Reasons for creating Data Marts, Building Data Marts: Top down Approach & Bottom up Approach, Data Warehouse Architecture, Two Tier Architecture, Three Tier Architecture. Data Warehouse Schema, Star, Snowflake & Fact Constellation Schema. OLAP, Need for OLAP, OLAP Operations, OLAP Models.	
UNIT-2: Data Preprocessing	
Need, Objectives and Techniques, Descriptive data summarization, Data Cleaning, Data Integration, Data Transformation, Data Reduction.	
Unit-3: Introduction to Datamining	
Introduction, Need for Data Mining, KDD Process, Data Mining Architecture, Data Mining Functionalities, Data Mining Task Primitives, Integration of a Data Mining System with a Database or Data Warehouse System	
UNIT-4: Mining Frequent Items and Associations:	
Frequent Item Set, Closed Item Set, Association Rule Mining, Market Basket Analysis, Classification of Association Rules, Apriori Algorithm	
Module 5: Classification and prediction	
Classification and Prediction: Classification & Prediction, Issues regarding classification & Prediction, Comparing Classification Methods, Classification by Decision Tree Induction	
UNIT-6:Clustering:	
Introduction, Cluster Analysis, Need, Categorization of Major clustering methods. Types of Data in Cluster Analysis, Partitioning Methods: K-Means Method, K-Mediods Method, Applications of data mining in various sectors	

SESSION PLAN

Session No	Details	Learning Resources	Learning Outcomes
UNIT-1: Introduction to Data warehousing:			
1	Definition & Characteristics of Data warehouse	Pg. 11,12	able to define data warehouse and will be able to list out its characteristics LO1
2	benefits & need of separate data warehouse	13	Understand need and benefits of separate data warehouse.LO1
3	Applications of data warehouse	13	Be able to get the areas where Data Warehouse is used LO1
4	OLAP & OLTP	14	be able to differentiate between OLAP &OLTP LO2
5	Schema for Multidimensional Data model	14	be understand schema for multidimensional data model and OLAP operations LO3
6	Schema definition	17	Understanding the schema description
7	OLAP Operations	19	Be able to understand the different operations of OLAP
8	Data Marts and Types of Data Marts	24	be able to understand concepts Data Mart ,its types and its use LO1
9	Three tier data warehouse architecture	27	be able understand tree tier architecture of data warehouse LO1
10	Types of OLAP Servers:ROLAP,MOLAP,HOLAP data warehouse Backend tools and Techniques	28	be able to differentiate different OLAP servers , will be able list various backend tools and techniques of a data warehouse LO2
Module II: Data Pre-processing			
11	Introduction to data pre-processing	32	Be able to describe the data preprocessing

12	Need objective	33	be understand why data preprocessing is required and various techniques of data preprocessing LO4
13	Techniques	35	
14	Data cleaning, integration	36,39	Will have fare understanding of data cleaning and integration process LO4
15	Data transformation, Reduction	42,44	How data transformation and data reduction is done during data preprocessing LO4
16	Discretization **(CES-1)- Class test**	41	Beable to understand what is done during discretization and its importance. LO4
Module III: Introduction to Datamining			
17	Definition	Pg. 45	Define data mining and why it is important. LO5
18	, Need for Data Mining	Pg. 46	
19	**KDD Process**	Pg.46,48	able to recall KDD Process and will be able understand conversion of data into valuable information LO5
20	Data Mining Architecture	Pg.48,52	Be understand and list out various component of data mining architecture and its use. LO5
21	Evolution of Database Technology	Pg. 52	Student will be able to recall history of data science. LO5
22	Types of data that can be mined	Pg.53	Be able to list out type of data where data mining can be applied to get valuable insight. LO5
23	Data mining functionalities	Pg.59	be able to list of data mining functionsLO5

24	Classification of Data mining systems	Pg.62	Be able to classify data mining system. LO5
25	Major Issues of Data mining**(CES 2)- Class Test/Moodle**	Pg.62	Be able understand major data mining issue/ challenges. LO5
Module IV: Association Rule data mining			
26	Market basket analysis, basic concepts	Pg.67	Be able comprehend the concept of market basket analysis. LO6
27	Road Map, Classification of Association rules	Pg.69	Be able to recall association rules. LO6
Module V:Classification and prediction			
28	What is classification, what is prediction? **(CES-3) –*QUIZ***	Pg.71,72	Be able to differentiate classification and prediction. LO6
29	How does classification works	Pg.72	
30	Issues Regarding Classification and prediction	Pg.74	Be list out various challenges in process of classification and prediction.LO6
31	Comparing classification methods	Pg.74	Be able to compare various classification methods. L06
32	Classification by Decision Tree Induction	Pg.75	be able to use Decision tree in classification process LO6
33	Attributes selection methods	Pg.76	be to identify attributes of database by using attribute selection methods .LO6
Module VI: Cluster Analysis			
34	Introduction, Need	Pg.79,80	be able define cluster and will be able to understand it analysis and need. LO6
35	Application of cluster analysis	Pg.79	
36	Major Clustering Method	Pg.81	be able to internalize various clustering methods. LO6
37	Partitioning Method K-Mean	Pg.81	be able to define working of K-mean
38	k medoids	Pg. 83	Be able to define the working of K-Medio's L06

39	Advantages and disadvantages of k medoids	Pg.84	Be able to define the benefits/drawbacks of k-medoid L06
40	Presentation or doubt session or question paper discussion		Doubt session
41	Presentation or doubt session or question paper discussion		Doubt session

Evaluation Criteria:

SN	**Type**	Number CES under this category	Marks allotted
1	Internal Exam	2	10*2=20
2	CES1 class Test	1	5
3	CES2 class test	1	5 best 2 out of 3 if all CES given
4	CES3 class test	1	5
5	Attendance	10 marks if attendance is >=75%	

Recommended/ Reference Text Books and Resources:

Text Books	**Datamining Concepts & Techniques By Jiawei Han &MichelineKamber**
Reference books	1:- Data Warehousing, Data Mining & OLAP By Berson& Smith 2:-Mastering Data Mining :Berry &Linoff
Internet Resource:	1. https://www.youtube.com/watch?v=jzDZZ-msoQc&list=PLx8GvJKPfHM-ooGurE99P9gmfW14qKWXw 2. https://www.youtube.com/watch?v=m-aKj5ovDfg&list=PL97D13C16B8A3C304 3. https://www.youtube.com/watch?v=6rs55Wg46dI

9. Contact Details:

Course leader:	**NEHA SABHARWAL**

Office Location:	**BVIMR , A-4 PaschimVihar New Delhi**
Website:	**www.bvimr.com**
Subject Faculty	**Ms. NehaSabharwal**
Email:	**Neha.sabharwal.ext@bvp.edu.in**

PROFILE
Ms. Neha Sabharwal

- 13+ years of Experience as Assistant Professor and working as a Content Writer.
- Visitng Faculty at different Indian Universities.
- MCA from Guru Gobind Singh Indraprastha University.
- The Fundamental of Digital Marketing by Google Digital Unlocked.
- Content Writing Certification from Content Vidhya
- Author to a Book on “Management Information System” with ISBN number: 81-8218-062-7, pages 131 and published by GALGOTIA PUBLISHING COMPANY

UNIT 1:Data warehousing
Definition & Characteristics of Data warehouse
benefits & need of separate data warehouse
OLAP & OLTP
Multidimensional Data Model: Tables, Spreadsheet, and data cubes
Schema for Multidimensional Data model, OLAP Operations
Data Marts and Type of Data Marts
Design of Data warehouse, Process of Data warehouse Design
Three tier data warehouse architecture
Types of OLAP Servers: ROLAP, MOLAP, HOLAP data warehouse Backend tools and Techniques

Overview of Data warehouse

The term "Data Warehouse" was first coined by Bill Inmon in 1990. According to Inmon, a data warehouse is a subject-oriented, integrated, time-variant, and non-volatile collection of data. This data helps analysts to take informed decisions in an organization.

An operational database undergoes frequent changes on a daily basis on account of the transactions that take place. Suppose a business executive wants to analyze previous feedback on any data such as a product, a supplier, or any consumer data, then the

executive will have no data available to analyze because the previous data has been updated due to transactions.

A data warehouses provides us generalized and consolidated data in multidimensional view. Along with generalized and consolidated view of data, a data warehouses also provides us Online Analytical Processing (OLAP) tools. These tools help us in interactive and effective analysis of data in a multidimensional space. This analysis results in data generalization and data mining.

Data mining functions such as association, clustering, classification, prediction can be integrated with OLAP operations to enhance the interactive mining of knowledge at multiple level of abstraction. That's why data warehouse has now become an important platform for data analysis and online analytical processing.

Understanding a Data Warehouse

- A data warehouse is a database, which is kept separate from the organization's operational database.
- There is no frequent updating done in a data warehouse.
- It possesses consolidated historical data, which helps the organization to analyze its business.
- A data warehouse helps executives to organize, understand, and use their data to take strategic decisions.
- Data warehouse systems help in the integration of diversity of application systems.
- A data warehouse system helps in consolidated historical data analysis.

Data Warehouse Features

The key features of a data warehouse are discussed below:

 Subject Oriented - A data warehouse is subject oriented because it provides information around a subject rather than the organization's ongoing operations. These subjects can be product, customers, suppliers, sales, revenue, etc. A data warehouse does not focus on the ongoing operations, rather it focuses on modelling and analysis of data for decision making.

 Integrated – A data warehouse is constructed by integrating data from heterogeneous sources such as relational databases, flat files, etc. This integration enhances the effective analysis of data.

 Time Variant - The data collected in a data warehouse is identified with a particular time period. The data in a data warehouse provides information from the historical point of view.

 Non-volatile - Non-volatile means the previous data is not erased when new data is added to it. A data warehouse is kept separate from the operational database and therefore frequent changes in operational database is not reflected in the data warehouse.

Note: **A** data warehouse does not require transaction processing, recovery, and concurrency controls, because it is physically stored and separate from the operational database.

Data Warehouse Applications

As discussed before, a data warehouse helps business executives to organize, analyze, and use their data for decision making. A data warehouse serves as a sole part of a plan-execute-assess "closed-loop" feedback system for the enterprise management. Data warehouses are widely used in the following fields:

- Financial services
- Banking services
- Consumer goods
- Retail sectors
- Controlled manufacturing

Types of Data Warehouse

Information processing, analytical processing, and data mining are the three types of data warehouse applications that are discussed below:

- **Information Processing** – A data warehouse allows to process the data stored in it. The data can be processed by means of querying, basic statistical analysis, reporting using crosstabs, tables, charts, or graphs.

- **Analytical Processing** – A data warehouse supports analytical processing of the information stored in it. The data can be analyzed by means of basic OLAP operations, including slice-and-dice, drill down, drill up, and pivoting.
- **Data Mining** - Data mining supports knowledge discovery by finding hidden patterns and associations, constructing analytical models, performing classification and prediction. These mining results can be presented using visualization tools.

Online Transactional Processing (OLTP)
VS
Online Analytical Processing (OLAP)

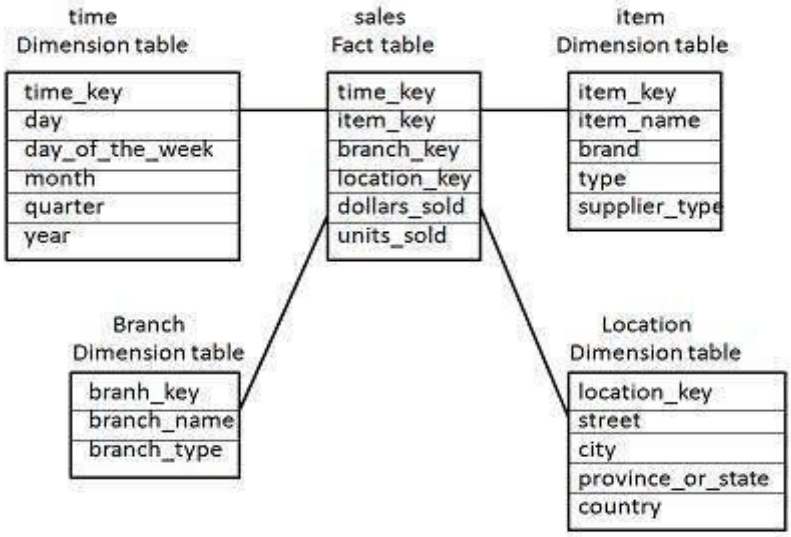
	OLTP	**OLAP**
Users	clerk, IT professional	knowledge worker
Function	day to day operations	decision support
DB design	application-oriented	subject-oriented
Data	current, up-to-date detailed, flat relational isolated	historical, summarized, multidimensional integrated, consolidated
Usage	repetitive	ad-hoc
Access	read/write Index/hash on prim. key	lots of scans
Unit Of Work	short, simple transaction	complex query
# Records Accessed	tens	millions
#Users	thousands	hundreds
DB size	100MB-GB	100GB-TB
Metric	transaction throughput	query throughput, response

Schema for Multidimensional Data Model **(Schema For Data Warehouse)**

Schema is a logical description of the entire database. It includes the name and description of records of all record types including all associated data-items and aggregates. Much like a database, a data warehouse also requires to maintain a schema. A database uses relational model, while a data warehouse uses **Star, Snowflake**, **and Fact Constellation schema**. In this chapter, we will discuss the schemas used in a data warehouse.

Star Schema

- Each dimension in a star schema is represented with only one-dimension table.
- This dimension table contains the set of attributes.
- The following diagram shows the sales data of a company with respect to the four dimensions, namely time, item, branch, and location.

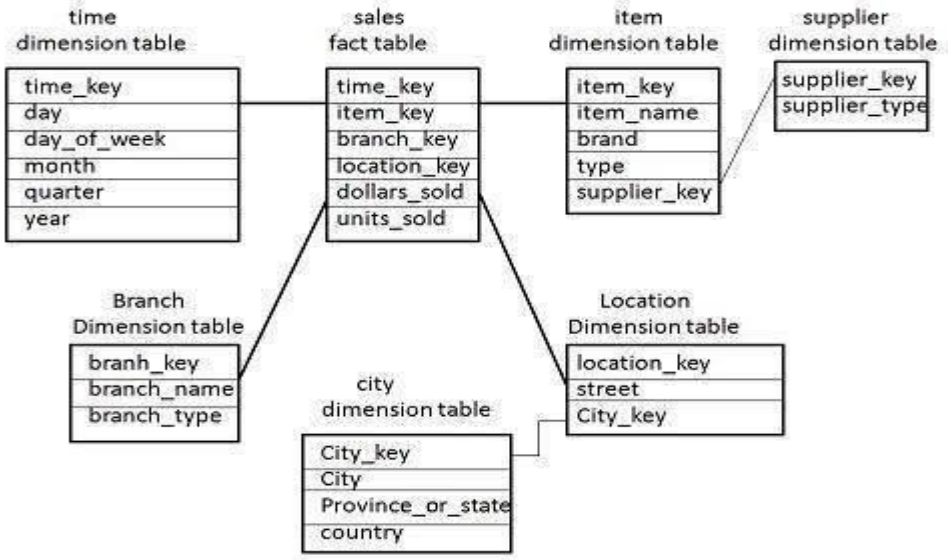


- There is a fact table at the center. It contains the keys to each of four dimensions.
- The fact table also contains the attributes, namely dollars sold and units sold.

Note: Each dimension has only one dimension table and each table holds a set of attributes. For example, the location dimension table contains the attribute set {location_key, street, city, province_or_state,country}. This constraint may cause data redundancy. For example, "Vancouver" and "Victoria" both the cities are in the Canadian province of British Columbia. The entries for such cities may cause data redundancy along the attributesprovince_or_state and country.

Snowflake Schema

- Some dimension tables in the Snowflake schema are normalized.
- The normalization splits up the data into additional tables.
- Unlike Star schema, the dimensions table in a snowflake schema are normalized. For example, the item dimension table in star schema is normalized and split into two dimension tables, namely item and supplier table.

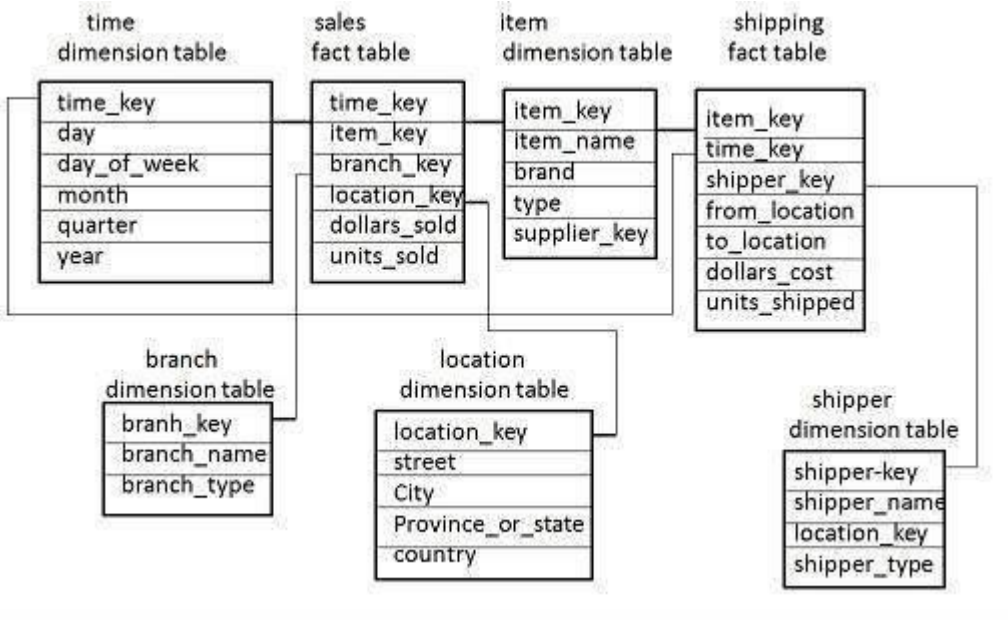


- Now the item dimension table contains the attributes item_key, item_name, type, brand, and supplier-key.
- The supplier key is linked to the supplier dimension table. The supplier dimension table contains the attributes supplier_key and supplier_type.

Note: Due to normalization in the Snowflake schema, the redundancy is reduced and therefore, it becomes easy to maintain and the save storage space.

Fact Constellation Schema

- A fact constellation has multiple fact tables. It is also known as galaxy schema.
- The following diagram shows two fact tables, namely sales and shipping.



- The sales fact table is same as that in the star schema.
- The shipping fact table has the five dimensions, namely item_key, time_key, shipper_key, from_location, to_location.
- The shipping fact table also contains two measures, namely dollars sold and units sold.
- It is also possible to share dimension tables between fact tables. For example, time, item, and location dimension tables are shared between the sales and shipping fact table.

Schema Definition

Multidimensional schema is defined using Data Mining Query Language (DMQL). The two primitives, cube definition and dimension definition, can be used for defining the data warehouses and data marts.

Syntax for Cube Definition

define cube <cube_name> [< dimension-list > }: <measure_list>

Syntax for Dimension Definition

define dimension <dimension_name> as (<attribute_or_dimension_list>)

Star Schema Definition

The star schema that we have discussed can be defined using Data Mining Query Language (DMQL) as follows:

define cube sales star [time, item, branch, location]:
dollars sold = sum(sales in dollars), units sold = count(*)
define dimension time as (time key, day, day of week, month, quarter, year)
define dimension item as (item key, item name, brand, type, supplier type)
define dimension branch as (branch key, branch name, branch type)
define dimension location as (location key, street, city, province or state, country)

Snowflake Schema Definition

Snowflake schema can be defined using DMQL as follows:

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

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

define dimension time as (time key, day, day of week, month, quarter, year)
define dimension item as (item key, item name, brand, type, supplier (supplier key, supplier type))
define dimension branch as (branch key, branch name, branch type)
define dimension location as (location key, street, city (city key, city, province or state, country))

Fact Constellation Schema Definition

Fact constellation schema can be defined using DMQL as follows:

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

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

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

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

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

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

define cube shipping [time, item, shipper, from location, to location]:

dollars cost = sum(cost in dollars), units shipped = count(*)

define dimension time as time in cube sales

define dimension item as item in cube sales

define dimension shipper as (shipper key, shipper name, location as location in cube sales, shipper type)

define dimension from location as location in cube sales

define dimension to location as location in cube sales

OLAP Operations

Since OLAP servers are based on multidimensional view of data, we will discuss OLAP operations in multidimensional data.

Here is the list of OLAP operations:

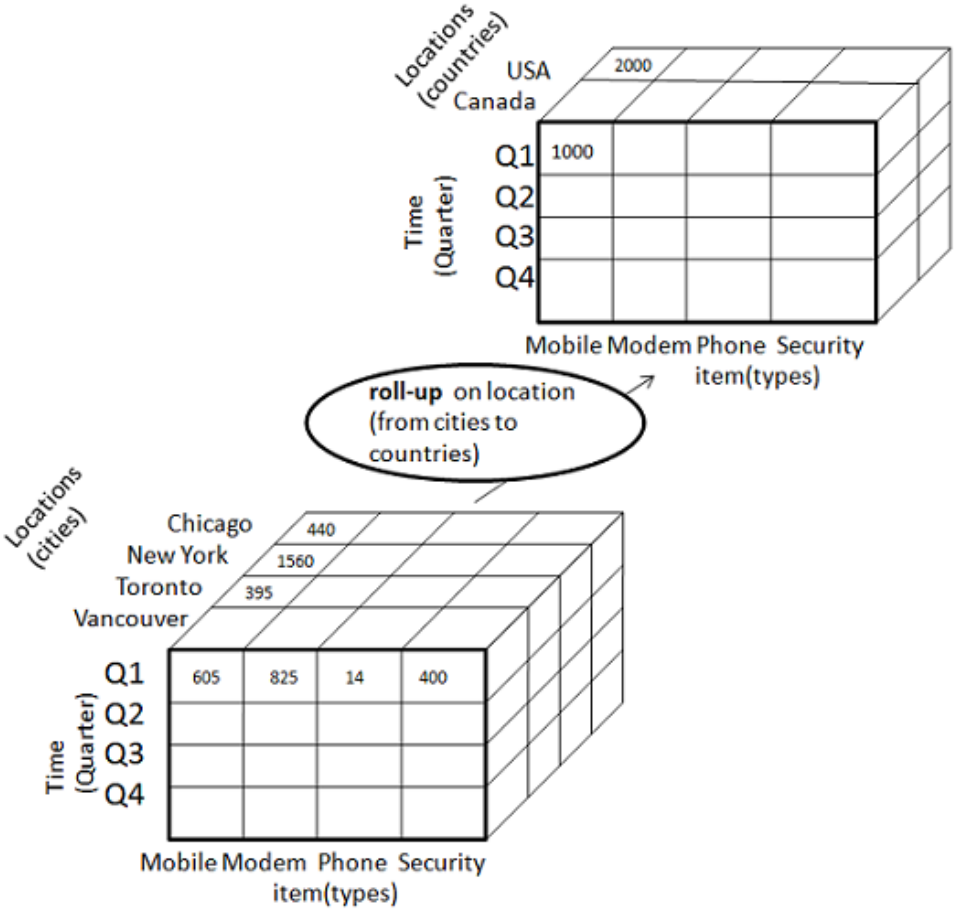
- Roll-up
- Drill-down
- Slice and dice
- Pivot (rotate)

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.



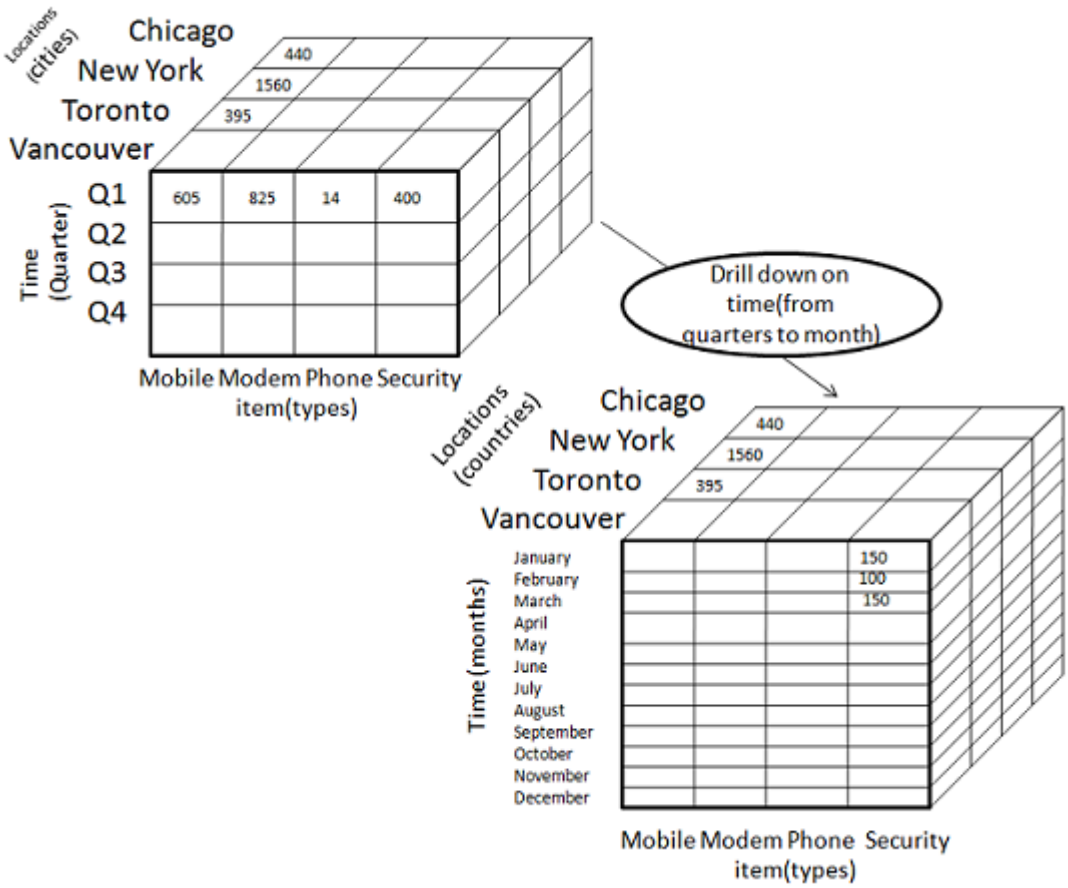
- Roll-up is performed by climbing up a concept hierarchy for the dimension location.
- Initially the concept hierarchy was "street < city < province < country".
- On rolling up, the data is aggregated by ascending the location hierarchy from the level of city to the level of country.
- The data is grouped 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.

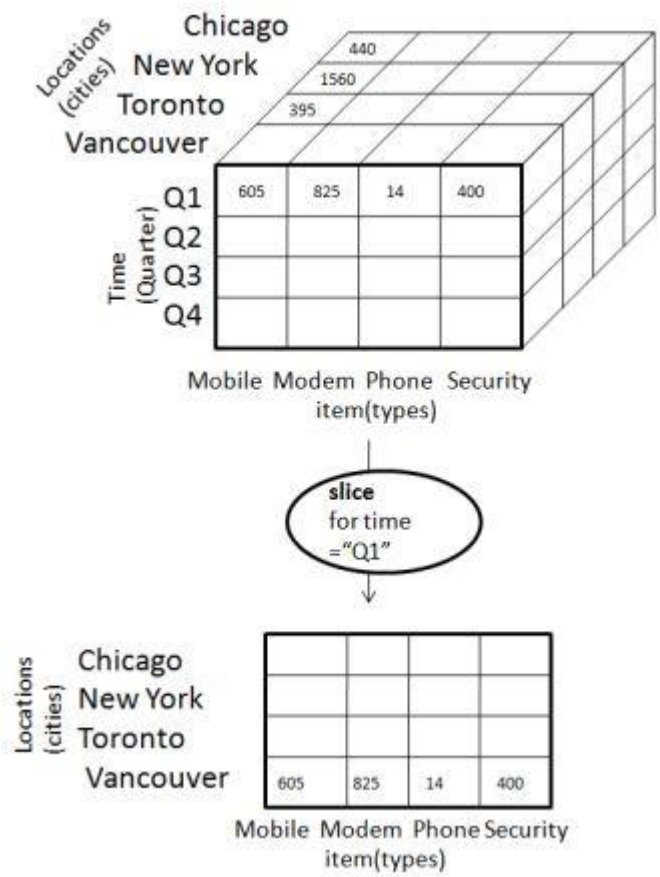
The following diagram illustrates how drill-down works:



- 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

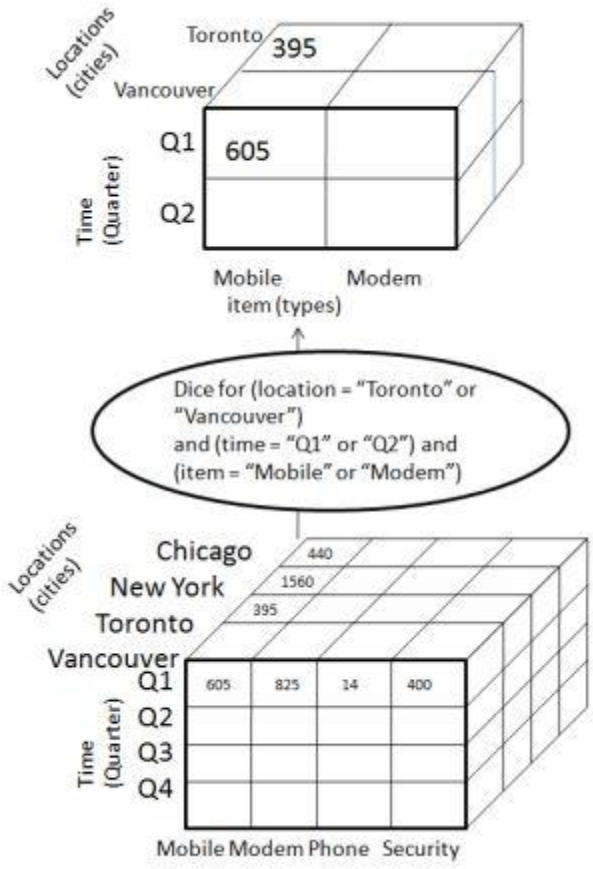
The slice operation selects one particular dimension from a given cube and provides a new sub-cube. Consider the following diagram that shows how slice works.



- 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.

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.

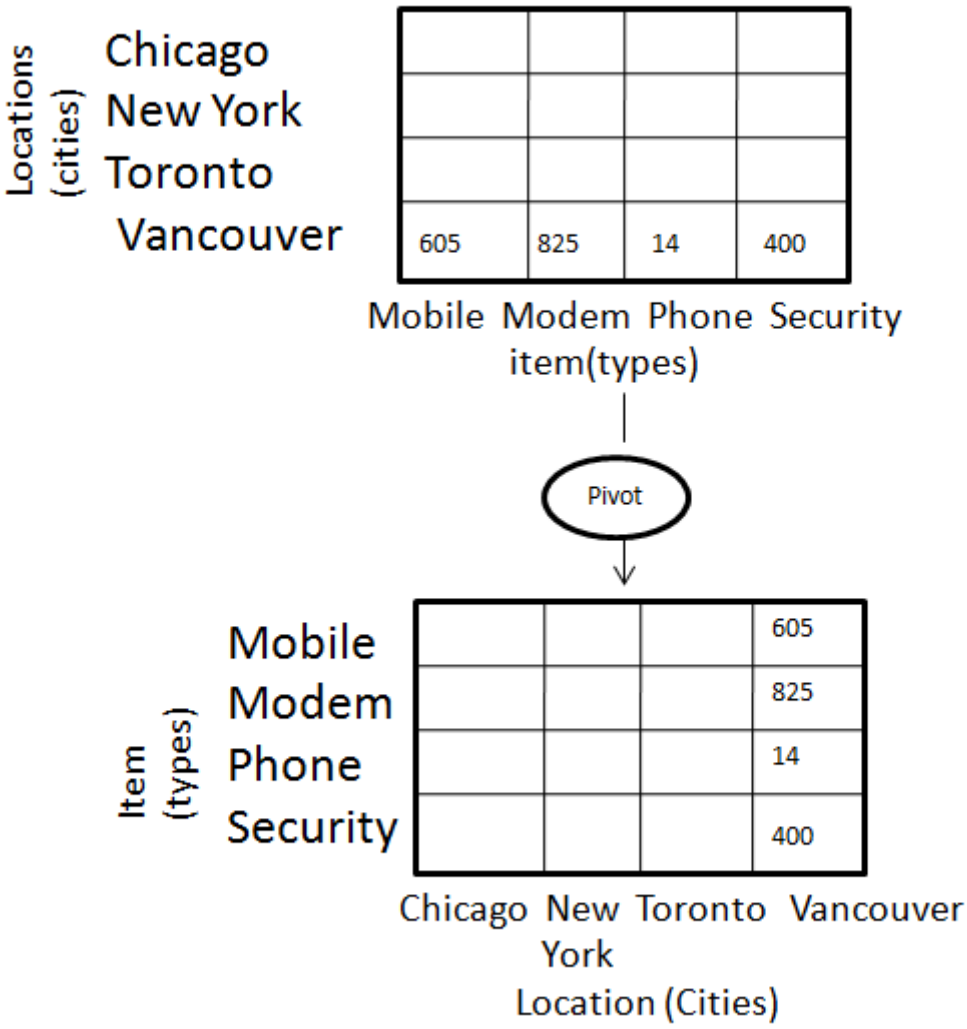


The dice operation on the cube based on the following selection criteria involves three dimensions.

- (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.



In this the item and location axes in 2-D slice are rotated.

Data Marts and Type of Data Marts

What Is a Data Mart?

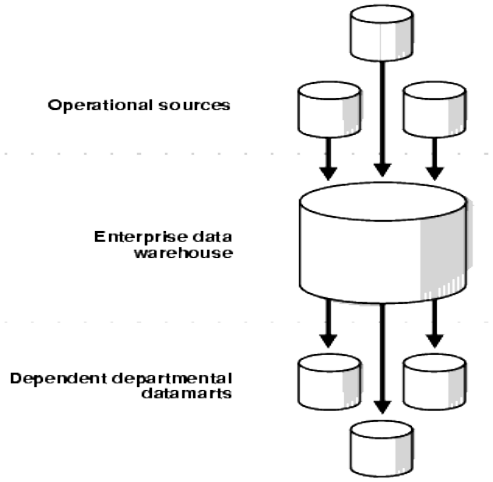
A *datamart* is a simple form of a data warehouse that is focused on a single subject (or functional area), such as Sales or Finance or Marketing. Data marts are often built and controlled by a single department within an organization. Given their single-subject focus, data marts usually draw data from only a few sources. The sources could be internal operational systems, a central data warehouse, or external data.

Three basic types of data marts are ***dependent***, ***independent***, and ***hybrid***. The categorization is based primarily on the data source that feeds the data mart.*Dependentdatamarts* draw data from a central data warehouse that has already been created. *Independentdatamarts*, in contrast, are standalone systems built by drawing data directly from operational or external sources of data or both. *Hybrid data marts* can draw data from operational systems or data warehouses.

Dependent Data Marts

A dependent data mart allows you to unite your organization's data in one data warehouse. This gives you the usual advantages of centralization. Figure 20-1 illustrates a dependent data mart.

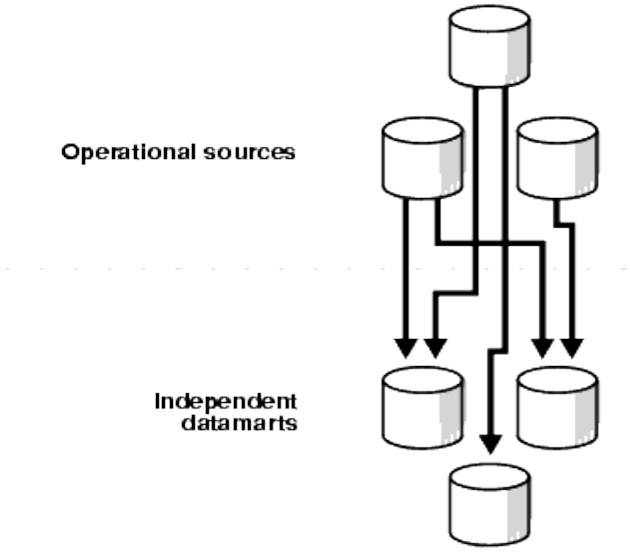
Figure 20-1 Dependent Data Mart



Independent Data Marts

An independent data mart is created without the use of a central data warehouse. This could be desirable for smaller groups within an organization. It is not, however, the focus of this Guide. See the *Data Mart Suites* documentation for further details regarding this architecture. Figure 20-2 illustrates an independent data mart.

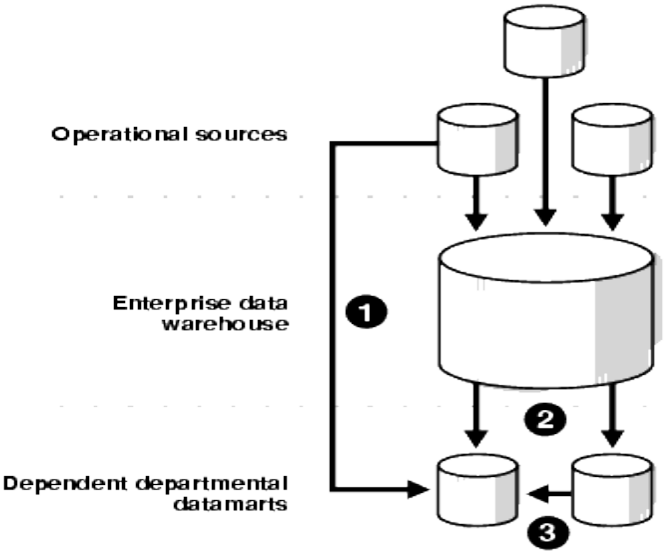
Figure 20-2 Independent Data Marts



Hybrid Data Marts

A hybrid data mart allows you to combine input from sources other than a data warehouse. This could be useful for many situations, especially when you need ad hoc integration, such as after a new group or product is added to the organization. Figure 20-3 illustrates a hybrid data mart.

Figure 20-3 Hybrid Data Mart

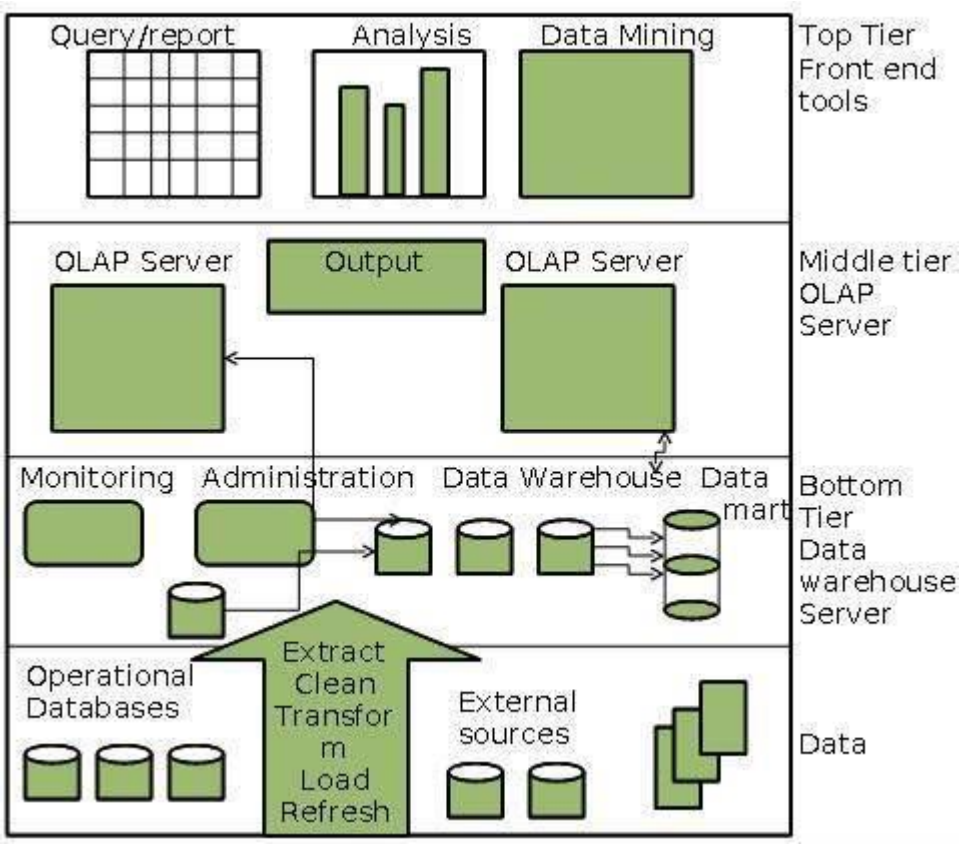


Three-Tier Data Warehouse Architecture

Generally a data warehouses adopts a three-tier architecture. Following are the three tiers of the data warehouse architecture.

- **Bottom Tier** - The bottom tier of the architecture is the data warehouse database server. It is the relational database system. We use the back end tools and utilities to feed data into the bottom tier. These back end tools and utilities perform the Extract, Clean, Load, and refresh functions.
- **Middle Tier** - In the middle tier, we have the OLAP Server that can be implemented in either of the following ways.
    - By Relational OLAP (ROLAP), which is an extended relational database management system. The ROLAP maps the operations on multidimensional data to standard relational operations.
    - By Multidimensional OLAP (MOLAP) model, which directly implements the multidimensional data and operations.
- **Top-Tier** - This tier is the front-end client layer. This layer holds the query tools and reporting tools, analysis tools and data mining tools.

The following diagram depicts the three-tier architecture of data warehouse:



Online Analytical Processing Server (OLAP) is based on the multidimensional data model. It allows managers, and analysts to get an insight of the information through fast, consistent, and interactive access to information. This chapter cover the types of OLAP, operations on OLAP, difference between OLAP, and statistical databases and OLTP.

Types of OLAP Servers

We have four types of OLAP servers:

- Relational OLAP (ROLAP)
- Multidimensional OLAP (MOLAP)
- Hybrid OLAP (HOLAP)
- Specialized SQL Servers

Relational OLAP

ROLAP servers are placed between relational back-end server and client front-end tools. To store and manage warehouse data, ROLAP uses relational or extended-relational DBMS.

ROLAP includes the following:

- Implementation of aggregation navigation logic.
- Optimization for each DBMS back end.
- Additional tools and services.

Multidimensional OLAP

MOLAP uses array-based multidimensional storage engines for multidimensional views of data. With multidimensional data stores, the storage utilization may be low if the data set is sparse. Therefore, many MOLAP server use two levels of data storage representation to handle dense and sparse data sets.

Hybrid OLAP (HOLAP)

Hybrid OLAP is a combination of both ROLAP and MOLAP. It offers higher scalability of ROLAP and faster computation of MOLAP. HOLAP servers allows to store the large data volumes of detailed information. The aggregations are stored separately in MOLAP store.

Specialized SQL Servers

Specialized SQL servers provide advanced query language and query processing support for SQL queries over star and snowflake schemas in a read-only environment.

Data warehouse: Back-end Tools

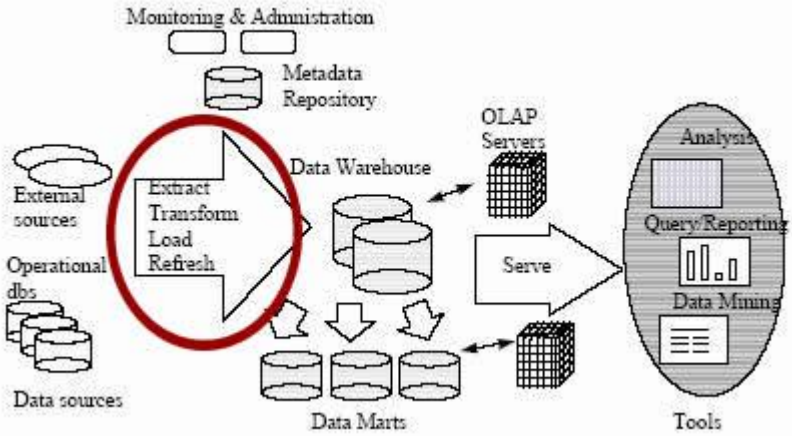


Figure 1. Data Warehouse Architecture

As highlighted in the circle in the diagram, data warehousing systems use various data extraction and cleaning tools, and load and refresh utilities for populating data warehouses. Below we describe the back-end tools and utilities.

Data Extraction

Data extraction is the act or process of retrieving **data** out of (usually unstructured or poorly structured) **data** sources for further **data** processing or **data** storage (**data** migration).

Data Cleaning

The data warehouse involves large volumes of data from multiple sources, which can lead to a high probability of errors and anomalies in the data. Inconsistent field lengths, inconsistent descriptions, inconsistent value assignments, missing entries and violation of integrity constraints are some of the examples. The three classes of data cleaning tools are popularly used to help detect data anomalies and correct them:

- Data migration tools - allow simple transformation rules to be specified.
- Data scrubbing tools - use domain specific knowledge to do the scrubbing of data from multiple sources.
- Data auditing tools - discover rules and relationships by scanning data.

Load

After extracting, cleaning, and transforming, data will be loaded into the data warehouse. A load utility has to allow the system administrator to monitor status, to cancel, suspend and resume a load, and to restart after failure with no loss of data integrity. Sequential loads can take a very long time to complete especially when it deals with terabytes of data. Therefore, pipelined and partitioned parallelism are typically used. Also incremental loading over full load is more popularly used with most commercial utilities since it reduces the volume of data that has to be incorporated into the data warehouse.

Refresh

There are two issues to consider when refreshing a data warehouse: when and how to refresh. Typically the warehouse is refreshed periodically. The refresh policy is set by the data warehouse administrator, depending on the business needs and traffic. Refreshing techniques depend on the characteristics of the source database and the capabilities of the database servers. Most contemporary database systems provide replication servers that support incremental techniques for moving updates from a primary database to one or more replicas. There are two basic replication techniques:

- Data shipping - treats a table in the data warehouse as a remote snapshot of a table in the source database. A trigger is used to update a snapshot log table whenever the source table changes.
- Transaction shipping - uses the regular transaction log instead of triggers and a special snapshot log table. At the source database site, the transaction log is used to detect updates on replicated tables, and those log records are transferred to a replication server.

Unit-II
Data Preprocessing

1. Introduction
2. Need of Data preprocessing
3. Objective of data preprocessing
4. Techniques of data preprocessing
    a. Data cleaning,
    b. integration,
    c. Data transformation,
    d. Reduction
    e. Discretization

Data Preprocessing

Today's real-world databases are highly prone to noisy, missing, and unreliable data due to their typically huge size (often several gigabytes or more) and their likely origin from multiple, heterogeneous sources. Low-quality data will lead to low-quality mining results. "*How can the data be preprocessed in order to help improve the quality of the data and, consequently, of the mining results? How can the data be preprocessed so as to improve the efficiency and ease of the mining process?*"

There are several data preprocessing techniques. *Data cleaning* can be applied to remove noise and correct inconsistencies in data. *Data integration* merges data from multiple sources into a coherent data store such as a data warehouse. *Data reduction* can reduce data size by, for instance, aggregating, eliminating redundant features, or clustering. *Data transformations* (e.g., normalization) may be applied, where data are scaled to fall within a smaller range like 0.0 to 1.0. This can improve the accuracy and efficiency of mining algorithms involving distance measurements. These techniques are not mutually exclusive; they may work together. For example, data cleaning can involve transformations to correct wrong data, such as by transforming all entries for a *date* field to a common format.

Data Quality: Why Preprocess the Data? (Need to Data preprocessing)

Data have quality if they satisfy the requirements of the intended use. There are many factors comprising **data quality**, including *accuracy, completeness, consistency, timeliness, believability*, and *interpretability*.

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

This scenario illustrates three of the elements defining data quality: **accuracy**, **completeness**, and **consistency**. Inaccurate, incomplete, and inconsistent data are everyday properties of large real-world databases and data warehouses. There are many possible reasons for inaccurate data (i.e., having incorrect attribute values). The data collection instruments used may be faulty. There may have been human or computer errors occurring at data entry. Users may purposely submit incorrect data values for mandatory fields when they do not wish to submit personal information (e.g., by choosing the default value “January 1” displayed for birthday). This is known as *disguised missing data*. Errors in data transmission can also occur. There may be technology limitations such as limited buffer size for coordinating synchronized data transfer and consumption. Incorrect data

may also result from inconsistencies in naming conventions or data codes, or inconsistent formats for input fields (e.g., *date*). Duplicate tuples also require data cleaning.

Incomplete data can occur for a number of reasons. Attributes of interest may not always be available, such as customer information for sales transaction data. Other data may not be included simply because they were not considered important at the time of entry. Relevant data may not be recorded due to a misunderstanding or because of equipment malfunctions. Data that were inconsistent with other recorded data may have been deleted. Furthermore, the recording of the data history or modifications may have been overlooked. Missing data, particularly for tuples with missing values for some attributes, may need to be inferred.

Recall that data quality depends on the intended use of the data. Two different users may have very different assessments of the quality of a given database. For example, a marketing analyst may need to access the database mentioned before for a list of customer addresses. Some of the addresses are outdated or incorrect, yet overall, 80% of the addresses are accurate. The marketing analyst considers this to be a large customer database for target marketing purposes and is pleased with the database's accuracy, although, as sales manager, you found the data inaccurate.

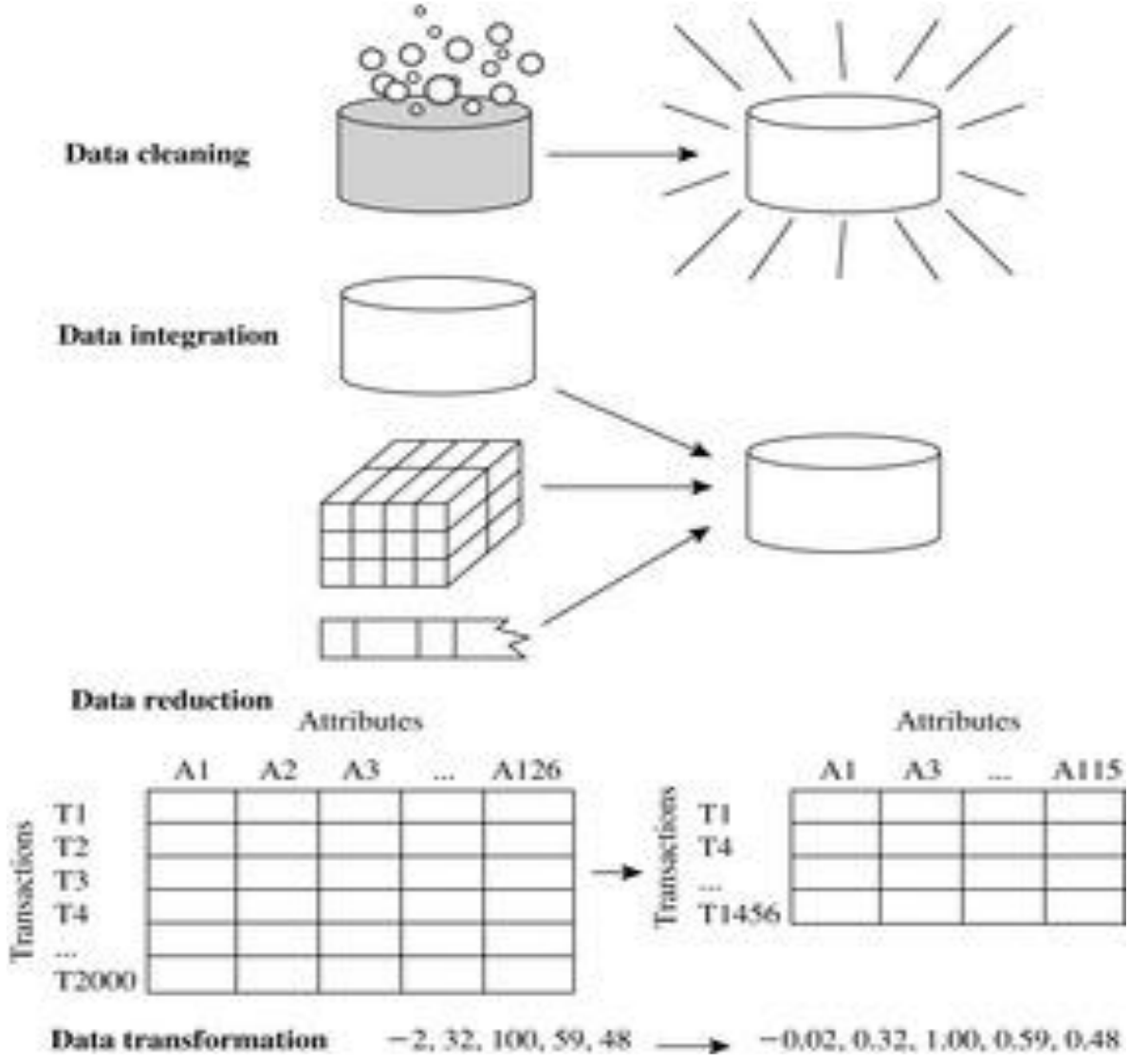
Timeliness also affects data quality. Suppose that you are managing the distribution of monthly sales bonuses to the top sales representatives at *AllElectronics*. Several sales representatives, however, fail to submit their sales records on time at the end of the month. There are also a number of corrections and adjustments that flow in after the month's end. For a period of time following each month, the data stored in the database are incomplete. However, once all of the data are received, it is correct. The fact that the month-end data are not updated in a timely fashion has a negative impact on the data quality.

Two other factors affecting data quality are believability and interpretability. **Believability** reflects how much the data are trusted by users, while **interpretability** reflects how easy the data are understood. Suppose that a database, at one point, had several errors, all of which have since been corrected. The past errors, however, had caused many problems for sales department users, and so they no longer trust the data. The data also use many accounting codes, which the sales department does not know how to interpret. Even

though the database is now accurate, complete, consistent, and timely, sales department users may regard it as of low quality due to poor believability and interpretability.

Methods/Techniques of Data Preprocessing

- Data cleaning,
- Integration,
- Data transformation,
- Reduction
- Discretization



Data cleaning

Data cleaning routines work to “clean” the data by filling in missing values, smoothing noisy data, identifying or removing outliers, and resolving inconsistencies. If users believe the data are dirty, they are unlikely to trust the results of any data mining that has been applied. Furthermore, dirty data can cause confusion for the mining procedure, resulting in unreliable output. Although most mining routines have some procedures for dealing with incomplete or noisy data, they are not always robust. Instead, they may concentrate on avoiding overfitting the data to the function being modeled. Therefore, a useful preprocessing step is to run your data through some data cleaning routines.

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

Handling Missing Values

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

1. **Ignore the tuple**: This is usually done when the class label is missing (assuming the mining task involves classification). This method is not very effective, unless the tuple contains several attributes with missing values. It is especially poor when the percentage of missing values per attribute varies considerably. By ignoring the tuple, we do not make use of the remaining attributes' values in the tuple. Such data could have been useful to the task at hand.

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

3. **Use a global constant to fill in the missing value**: Replace all missing attribute values by the same constant such as a label like *“Unknown”* or $-\infty$. If missing values are replaced by, say, “*Unknown*,” then the mining program may mistakenly think that they form an

interesting concept, since they allhave a value in common—that of "*Unknown*." Hence, although this method is simple, it is not foolproof.

4. **Use a measure of central tendency for the attribute (e.g., the mean or median) to fill in the missing value**: measures of central tendency, which indicate the "middle" value of a data distribution. For normal (symmetric) data distributions, the mean can be used, while skewed data distribution should employ the median. For example, suppose that the data distribution regardingthe income of *AllElectronics*customers is symmetric and that the mean income is $56,000. Use this value to replace the missing value for *income*.

5. **Use the attribute mean or median for all samples belonging to the same class as the given tuple**: For example, if classifying customers according to *credit_risk*, we may replace the missing value with the mean *income* value for customers in the same credit risk category as that of the given tuple. If the data distribution for a given class is skewed, the median value is a better choice.

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

Methods 3 through 6 bias the data—the filled-in value may not be correct. Method 6, however, is a popular strategy. In comparison to the other methods, it uses the most information from the present data to predict missing values. By considering the other attributes' values in its estimation of the missing value for *income*, there is a greater chance that the relationships between *income* and the other attributes are preserved.

Handling noisy Data

What is noise?" **Noise** is a random error or variance in a measured variable. Let's look at the following data smoothing techniques.

Binning: Binning methods smooth a sorted data value by consulting its "neighborhood," that is, the values around it. The sorted values are distributed into a number of "buckets," or *bins*. Because binning methods consult the neighborhood of values, they perform *local* smoothing. Figure illustrates some binning techniques. In this example, the data for *price* are first sorted and then partitioned into *equal-frequency* bins of size 3 (i.e., each bin

contains three values). In **smoothing by bin means**, each value in a bin is replaced by the mean value of the bin. For example, the mean of the values 4, 8, and 15 in Bin 1 is 9. Therefore, each original value in this bin is replaced by the value 9.

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

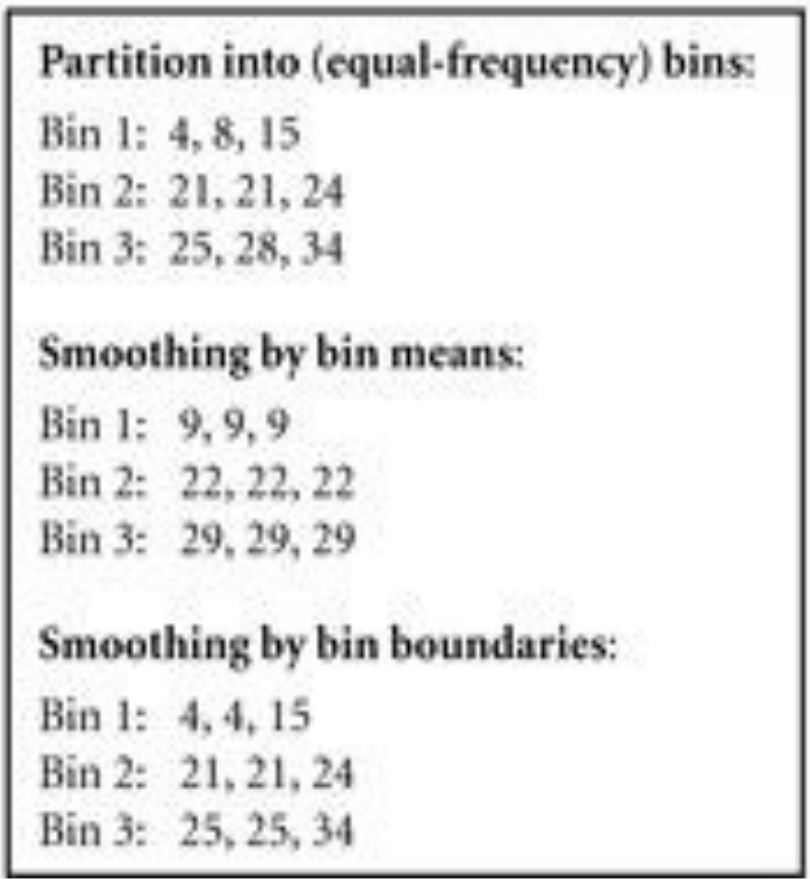


Figure: inning methods for data smoothing.

Similarly, **smoothing by bin medians** can be employed, in which each bin value is replaced by the bin median. In **smoothing by bin boundaries**, the minimum and maximum values in a given bin are identified as the *bin boundaries*. Each bin value is then replaced by the closest boundary value. In general, the larger the width, the greater the effect of the smoothing. Alternatively, bins may be *equal width*, where the interval range of values in each bin is constant.

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

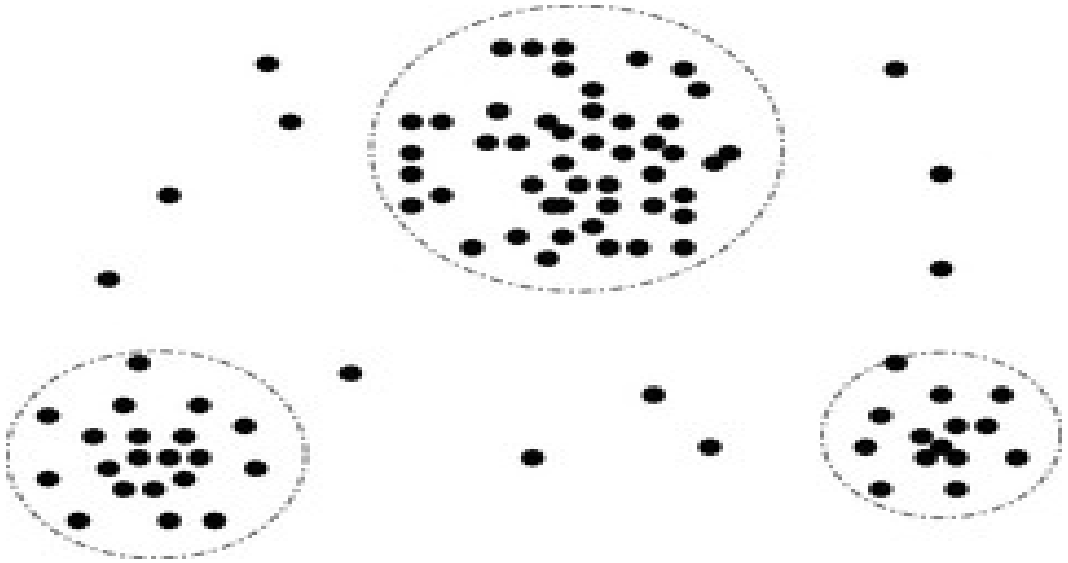


Figure:*A 2-D customer data plot with respect to customer locations in a city, showing three data clusters. Outliers may be detected as values that fall outside of the cluster sets.*

Data Integration

Data mining often requires data integration—the merging of data from multiple data stores. Careful integration can help reduce and avoid redundancies and inconsistencies in the resulting data set. This can help improve the accuracy and speed of the subsequent data mining process. The semantic heterogeneity and structure of data pose great challenges in data integration. How can we match schema and objects from different sources? This is the essence of the *entity identification problem*

Entity Identification Problem:

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

There are a number of issues to consider during data integration. *Schema integration* and *object matching* can be tricky. How can equivalent real-world entities from multiple data sources be matched up? This is referred to as the **entity identification problem**. For example, how can the data analyst or the computer be sure that *customer_id*in one database and *cust_number*in another refer to the same attribute? Examples of metadata for each attribute include the name, meaning, data type, and range of values permitted for the attribute, and null rules for handling blank, zero, or null values. Such metadata can be

used to help avoid errors in schema integration. The metadata may also be used to help transform the data (e.g., where data codes for *pay_type*in one database may be "*H*" and "*S*" but *1* and *2* in another). Hence, this step also relates to data cleaning, as described earlier.

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

Redundancy and Correlation Analysis

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

Some redundancies can be detected by **correlation analysis**. Given two attributes, such analysis can measure how strongly one attribute implies the other, based on the available data. For nominal data, we use the χ^2 (*chi-square*) test. For numeric attributes, we can use the *correlation coefficient* and *covariance*, both of which access how one attribute's values vary from those of another.

Tuple Duplication

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

Data Value Conflict Detection and Resolution

Data integration also involves the *detection and resolution of data value conflicts*. For example, for the same real-world entity, attribute values from different sources may differ. This may be due to differences in representation, scaling, or encoding. For instance, a *weight* attribute may be stored in metric units in one system and British imperial units in another. For a hotel chain, the *price* of rooms in different cities may involve not only different currencies but also different services (e.g., free breakfast) and taxes. When exchanging Information between schools, for example, each school may have its own curriculum and grading scheme. One university may adopt a quarter system, offer three courses on database systems, and assign grades from A+ to F, whereas another may adopt a semester system, offer two courses on databases, and assign grades from 1 to 10. It is difficult to work out precise course-to-grade transformation rules between the two universities, making information exchange difficult.

Attributes may also differ on the abstraction level, where an attribute in one system is recorded at, say, a lower abstraction level than the "same" attribute in another. For example, the *total_sales*in one database may refer to one branch of *All_Electronics*, while an attribute of the same name in another database may refer to the total sales for *All_Electronics*stores in a given region. The topic of discrepancy detection is further described.

Data Transformation and Data Discretization:

This section presents methods of data transformation. In this preprocessing step, the data are transformed or consolidated so that the resulting mining process may be more efficient, and the patterns found may be easier to understand. Data discretization, a form of data transformation, is also discussed.

In *data transformation*, the data are transformed or consolidated into forms appropriate for mining. Strategies for data transformation include the following:

1. **Smoothing**, which works to remove noise from the data. Techniques include binning, regression, and clustering.

2. **Attribute construction** (or *feature construction*), where new attributes are constructed and added from the given set of attributes to help the mining process.
3. **Aggregation**, where summary or aggregation operations are applied to the data. For example, the dailysales data may be aggregated so as to compute monthly and annual total amounts. This step is typically usedin constructing a data cube for data analysis at multiple abstraction levels.
4. **Normalization**, where the attribute data are scaled so as to fall within a smaller range, such as −1.0 to 1.0,or 0.0 to 1.0.
5. **Discretization**, where the raw values of a numeric attribute (e.g., *age*) are replaced by interval labels (e.g.,0–10, 11–20, etc.) or conceptual labels (e.g., *youth, adult, senior*). The labels, in turn, can be recursivelyorganized into higher-level concepts, resulting in a *concept hierarchy* for the numeric attribute.
6. **Concept hierarchy generation for nominal data**, where attributes such as *street* can be generalized tohigher-level concepts, like *city* or *country*. Many hierarchies for nominal attributes are implicit within thedatabase schema and can be automatically defined at the schema definition level.

Data Reduction

Imagine that you have selected data from the *AllElectronics*data warehouse for analysis. The data set will likely be huge! Complex data analysis and mining on huge amounts of data can take a long time, making such analysis impractical or infeasible.

Data reduction techniques can be applied to obtain a reduced representation of the data set that is much smaller in volume, yet closely maintains the integrity of the original data. That is, mining on the reduced data set should be more efficient yet produce the same (or almost the same) analytical results. In this section, we first present an overview of data reduction strategies, followed by a closer look at individual techniques.
Data reduction strategies include *dimensionality reduction*, *numerosity reduction*, and *data compression*.
Dimensionality reduction is the process of reducing the number of random variables or attributes under consideration. Dimensionality reduction methods include *wavelet transforms* and *principal components analysis,* which transform or project the original data

onto a smaller space. *Attribute subset selection* is a method of dimensionality reduction in which irrelevant, weakly relevant, or redundant attributes or dimensions are detected and removed.

Numerosity reduction techniques replace the original data volume by alternative, smaller forms of datarepresentation. These techniques may be parametric or nonparametric. For *parametric methods*, a model is usedto estimate the data, so that typically only the data parameters need to be stored, instead of the actual data.(Outliers may also be stored.) Regression and log-linear models are examples. *Nonparametric methods* for storing reduced representations of the data include *histograms*, *clustering*, *sampling*, and *data cube aggregation.*

In **data compression**, transformations are applied so as to obtain a reduced or "compressed" representation ofthe original data. If the original data can be *reconstructed* from the compressed data without any informationloss, the data reduction is called **lossless**. If, instead, we can reconstruct only an approximation of the originaldata, then the data reduction is called **lossy**. There are several lossless algorithms for string compression;however, they typically allow only limited data manipulation. Dimensionality reduction and numerosityreduction techniques can also be considered forms of data compression.There are many other ways of organizing methods of data reduction. The computational time spent on datareduction should not outweigh or "erase" the time saved by mining on a reduced data set size.

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

Module III: Introduction to Datamining

	Definition, Need for Data Mining
	KDD Process
	Data Mining Architecture
	Evolution of Database Technology
	Types of data that can be mined
	Data mining functionalities
	Classification of Data mining systems
	Major Issues of Data mining

Data Mining?

It is no surprise that data mining, as a truly interdisciplinary subject, can be defined in many ways. Even the term *data mining* does not really present all the major components in the picture. To refer to theMining of gold from rocks or sand, we say *gold mining* instead of rock or sand mining. Analogously, data mining should have been more appropriately named "knowledge mining from data," which is unfortunately somewhat long. However, the shorter term, *knowledge mining* may not reflect the emphasis on

FIG 3.1

Data mining—searching for knowledge (interesting patterns) in data.

Mining from large amounts of data. Nevertheless, mining is a glowing term characterizing the process that finds a small set of precious pieces from a great deal of raw material. Thus, such a misleading term carrying both "data "and "mining" became a popular choice. In addition, many other terms have a similar meaning to data mining— for example,

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

Definition of data Mining:-

Data Mining is defined as extracting information from huge sets of data. In other words, we can say that data mining is the procedure of mining knowledge from data.

Data Mining Applications/need

Data mining is highly useful in the following domains:

1. Market Analysis and Management
2. Corporate Analysis & Risk Management
3. Fraud Detection

Apart from these, data mining can also be used in the areas of production control, customer retention, science exploration, sports, astrology, and Internet Web Surf-Aid

1. Market Analysis and Management

Listed below are the various fields of market where data mining is used:

a. **Customer Profiling** - Data mining helps determine what kind of people buy what kind of products.
b. **Identifying Customer Requirements** - Data mining helps in identifying the best products for different customers. It uses prediction to find the factors that may attract new customers.
c. **Cross Market Analysis -** Data mining performs Association/correlations between product sales.
d. **Target Marketing -** Data mining helps to find clusters of model customers who share the same characteristics such as interests, spending habits, income, etc.
e. **Determining Customer purchasing pattern -** Data mining helps in determining customer purchasing pattern.
f. **Providing Summary Information -** Data mining provides us various multidimensional summary reports.

2. Corporate Analysis and Risk Management

Data mining is used in the following fields of the Corporate Sector:

- **Finance Planning and Asset Evaluation** - It involves cash flow analysis and prediction, contingent claim analysis to evaluate assets.
- **Resource Planning** - It involves summarizing and comparing the resources and spending.
- **Competition** - It involves monitoring competitors and market directions.

3. Fraud Detection

Data mining is also used in the fields of credit card services and telecommunication to detect frauds. In fraud telephone calls, it helps to find the destination of the call, duration of the call, time of the day or week, etc. It also analyzes the patterns that deviate from expected norms

What are Data Mining and Knowledge Discovery?

With the enormous amount of data stored in files, databases, and other repositories, it is increasingly important, if not necessary, to develop powerful means for analysis and perhaps interpretation of such data and for the extraction of interesting knowledge that could help in decision-making.

Data Mining, also popularly known as *Knowledge Discovery in Databases* (KDD), refers to the nontrivial extraction of implicit, previously unknown and potentially useful information from data in databases. While data mining and knowledge discovery in databases (or KDD) are frequently treated as synonyms, data mining is actually part of the knowledge discovery process. The following Figure 1.1 shows data mining as a step in an iterative knowledge discovery process.

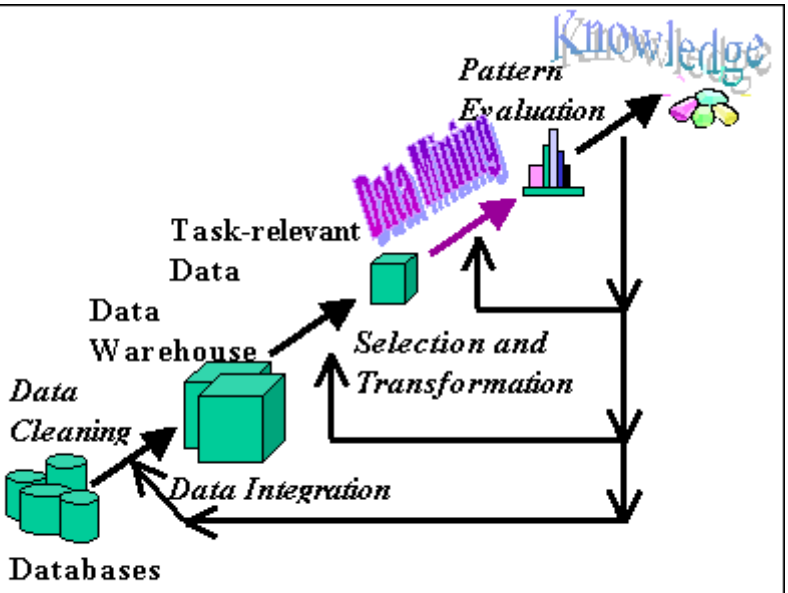


Figure 1.1: Data Mining is the core of Knowledge Discovery process

The Knowledge Discovery in Databases process comprises of a few steps leading from raw data collections to some form of new knowledge. The iterative process consists of the following steps:

- **Data cleaning**: also known as data cleansing, it is a phase in which noise data and irrelevant data are removed from the collection.
- **Data integration**: at this stage, multiple data sources, often heterogeneous, may be combined in a common source.
- **Data selection**: at this step, the data relevant to the analysis is decided on and retrieved from the data collection.
- **Data transformation**: also known as data consolidation, it is a phase in which the selected data is transformed into forms appropriate for the mining procedure.
- **Data mining**: it is the crucial step in which clever techniques are applied to extract patterns potentially useful.
- **Pattern evaluation**: in this step, strictly interesting patterns representing knowledge are identified based on given measures.
- **Knowledge representation**: is the final phase in which the discovered knowledge is visually represented to the user. This essential step uses visualization techniques to help users understand and interpret the data mining results.

It is common to combine some of these steps together. For instance, *data cleaning* and *data integration* can be performed together as a pre-processing phase to generate a data warehouse. *Data selection* and *data transformation* can also be combined where the consolidation of the data is the result of the selection, or, as for the case of data warehouses, the selection is done on transformed data.

The KDD is an iterative process. Once the discovered knowledge is presented to the user, the evaluation measures can be enhanced, the mining can be further refined, new data can be selected or further transformed, or new data sources can be integrated, in order to get different, more appropriate results.

Data mining derives its name from the similarities between searching for valuable information in a large database and mining rocks for a vein of valuable ore. Both imply either sifting through a large amount of material or ingeniously probing the material to exactly pinpoint where the values reside. It is, however, a misnomer, since mining for gold in rocks is usually called "gold mining" and not "rock mining", thus by analogy, data mining should have been called "knowledge mining" instead. Nevertheless, data mining became the accepted customary term, and very rapidly a trend that even overshadowed more general terms such as knowledge discovery in databases (KDD) that describe a more complete process. Other similar terms referring to data mining are: data dredging, knowledge extraction and pattern discovery.

Components of a Data Mining System*: (Data Mining architecture)*

Basic components of a data mining system are the user interface, data mining services, data access services and the data itself. The user interface allows the user to select and prepare data sets and apply data mining techniques to them. Formatting and presenting the results of a data mining session is also an important task of the user interface. Datamining services consist of all components of the system that process a special data mining algorithm, for example association rule discovery.Components access data through data access services. Access services can be optimized for special database management systems or can offer a standard interface like ODBC. The data itself constitute the fourth component of a datamining system. Since data mining is

particularly interesting in large scale enterprise environments we assume the data to reside in a data warehouse.

Four basic components are present in all data mining systems. In the following we describe three different architectures where these components are appropriately distributed over the various tiers. For each approach we check with which of the prerequisites of section 3.1 it complies. Figure 2 outlines the architectures discussed below.

One-tier Architecture

Classical architecture of data mining systems is a one-tier architecture. Such a system is completely client based. Basically all data mining systems of the first generation are based on this architecture. The user has to select a small subset of data warehouse data and load it on the client in order to make it accessible to the data mining to tool. This tool may offer several data mining techniques. The most obvious drawback of the one-tier approach is the size of the data set that can be mined and the speed of the mining process. This is often overcome by selecting a random sample from the data. A truly random (unbiased) sample is needed to ensure the accuracy of mined patterns, and even then patterns relating to small segments of the data can be lost. The data resides in raw files of the client's file system. Another dis-advantage is the absence of a multi-user functionality. Each user has to define his own subset of the data warehouse and load it separately onto the client machine. Since each user runs his own client-based data mining software, there is no way for datamining specific access control and control of system resources. Optimization of the data mining process restricted to choosing more efficient implementations of the data mining techniques.

Two-Tier architecture:

In a two-tier architecture the data mining tool completely resides on the client but there is no need to copy data to it in advance. The data mining application may choose to load parts of the data during different stages of the mining computation. There are several alternatives for running data mining algorithms in this architecture.

Download Approach

The connection to the data warehouse can be used to load data to the client and make it accessible for data mining. This can be done dynamically, therefore avoiding the problems with storing huge data sets on the client. Even if data is loaded in advance, this approach is superior compared to the one-tier architecture. The automatic loading of data by

The client enables it to store pre-processed data depending on the user's needs. Pre-processed data may be of reduced size and stored in a way that supports the data mining algorithm. Hence, better performance of the discovery process and less space consumption is achieved.

Query Approach

For some data mining techniques it is possible to formulate parts of the algorithm in a query language like SQL. The client sends SQL statements to the data warehouse and uses the results for the data mining process. One advantage compared to the download approach is that only data which is really needed is sent to the client because filtering and aggregation is already carried out by the database system. Since parts of the application logic are formulated in SQL, query processing capabilities of the data warehouse system can be exploited.

Database Approach

In this approach the complete data mining algorithm is processed by the database system. This can be realized by stored procedures and user defined functions. Only the data mining results have to be sent to the client which is responsible for displaying them. The data mining process is able to exploit the efficient processing capabilities offered by the data warehouse.

The two-tier architecture has evident advantages over purely client-based data mining. It enables direct access to the data warehouse. No data extraction is necessary as a prerequisite for data mining. The two-tier architecture does not limit the size of a database that can be mined. New information can be discovered in the masses of data stored in the data warehouse. Additionally, the data mining process can take advantage of the query processing capabilities provided by special data warehousing hardware and software. Besides the advantages of this approach some problems still remain. One problem is the limited access to the data warehouse system. Data warehouse systems are often in a “dark" site with restricted access. It is not allowed to install and configure applications on this system. Only the download approach and the query approach are applicable.

Additionally, there is limited control over system resources. When all users directly access the data warehouse it is not possible to control the bandwidth and the CPU cycles each user needs for its data mining application. Many users concurrently access the data warehouse for data mining purposes. In the two-tier environment there is no way to control this access by data mining specific user priorities and user groups. The last drawback we want to mention here is the limited scope for optimizations. There are only two strategies to make the data mining process more efficient: the exploitation of the query processing

capabilities of the data warehouse and the enhancement of the data mining algorithm. There is limited scope for parallel algorithms and reuse of results by different clients.

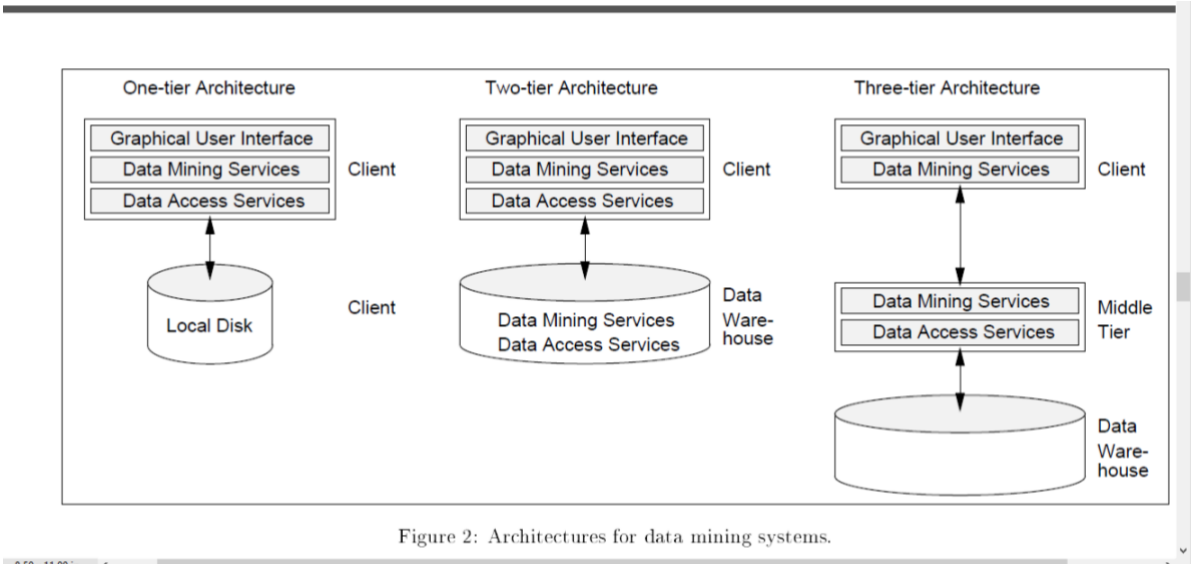


Figure 2: Architectures for data mining systems.

Three-tier Architecture:

A three-tier architecture addresses the problems remaining with a two-tier architecture. This is achieved by an additional layer holding the data access services and parts of the data mining services. Data mining services may also be present on the client. Which part of the data mining services should be client based depends on data mining techniques and algorithms?

The data mining process works as follows in this architecture. First, the user define the parameters for data mining by the graphical user interface. The data mining services on the client perform some pre-processing prior to calling the data mining services on the middle tier. The first task on the middle-tier is authentication and authorization of the users. Then the data mining services queue and execute the tasks of several clients and send back the results. These are used in the post-processing of the client, which computes the final outcome and presents it to the user. A client may start several data mining tasks in one session. Each of them includes a number of calls to the middle tier. Data mining Services use the data access services on the middle tier in order to read from different types of data sources.

This three-tier approach has several advantages com- pared to the two-tier architecture. First, the data mining services can fully control bandwidth and CPU cycles for each user cause there is a centralized service that man- ages users' tasks and resources. This enables the system to guarantee a maximum usage of system resources for data mining purposes. Second, the system can service user's ac- cording to their priority and to their membership in user groups. This includes restricted access to data mining tables as well as user specific response behavior. Third, a wide range of optimization strategies can be realized. The tasks of the data mining services can be distributed over the client and the middle tier. The middle tier can exploit parallelism by parallel processing on the middle tier hardware and parallel connections to the database layer. Additionally, the data mining services can reuse the outcome of data mining sessions and pre-compute common intermediate results.

Evolution of Database Technology

1960s:

(Electronic) Data collection, database creation, IMS (hierarchical database system by IBM) and network DBMS

1970s:

Relational data model, relational DBMS implementation

1980s:

RDBMS, advanced data models (extended-relational, OO, deductive, etc.)

Application-oriented DBMS (spatial, scientific, engineering, etc.

1990s:

Data mining, data warehousing, multimedia databases, and Web databases

2000s:

Stream data management and mining

Data mining and its applications

Web technology

XML

Data integration

Social Networks

Cloud Computing

What kind of Data can be mined? (Typesof datacan be mined?)

In principle, data mining is not specific to one type of media or data. Data mining should be applicable to any kind of information repository. However, algorithms and approaches may differ when applied to different types of data. Indeed, the challenges presented by different types of data vary significantly. Data mining is being put into use and studied for databases, including relational databases, object-relational databases and object-oriented databases, data warehouses, transactional databases, unstructured and semi-structured repositories such as the World Wide Web, advanced databases such as spatial databases, multimedia databases, time-series databases and textual databases, and even flat files. Here are some examples in more detail:

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

Relational Databases: Briefly, a relational database consists of a set of tables containing either values of entity attributes, or values of attributes from entity relationships. Tables have columns and rows, where columns represent attributes and rows represent tuples. A tuple in a relational table corresponds to either an object or a relationship between objects and is identified by a set of attribute values representing a unique key. In Figure 1.2 we present some relations *Customer*, *Items*, and *Borrow* representing business activity in a fictitious video store OurVideoStore. These relations are just a subset of what could be a database for the video store and is given as an example.

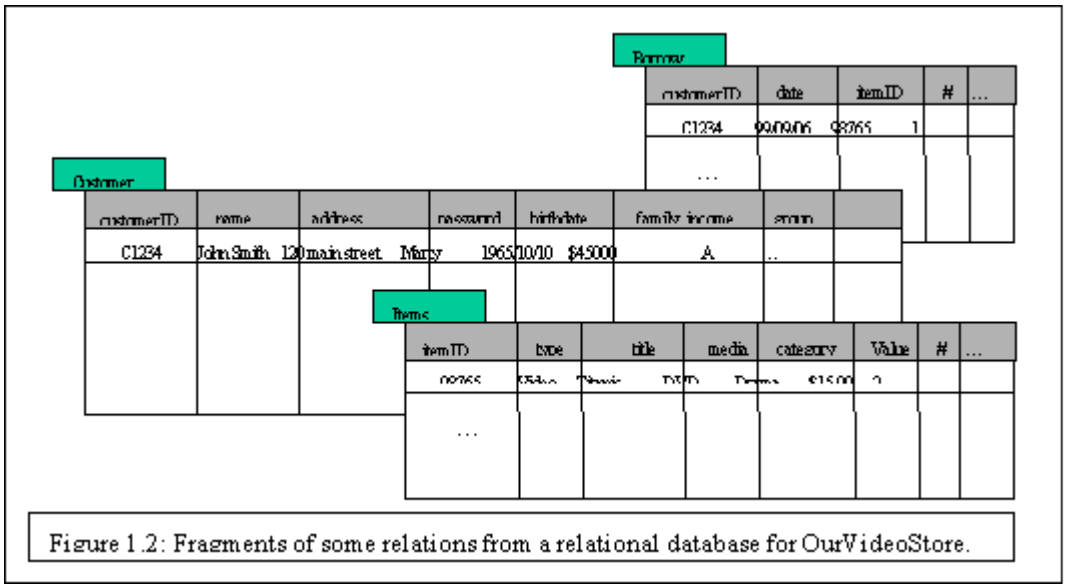


Figure 1.2: Fragments of some relations from a relational database for OurVideoStore.

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

select count(*) FROM Items WHERE type=video GROUP BY category.

Data mining algorithms using relational databases can be more versatile than data mining algorithms specifically written for flat files, since they can take advantage of the structure inherent to relational databases. While data mining can benefit from SQL for data selection, transformation and consolidation, it goes beyond what SQL could provide, such as predicting, comparing, detecting deviations, etc.

Data Warehouses: A data warehouse as a storehouse, is a repository of data collected from multiple data sources (often heterogeneous) and is intended to be used as a whole under the same unified schema. A data warehouse gives the option to analyze data from different sources under the same roof. Let us suppose that OurVideoStore becomes a franchise in North America. Many video stores belonging to OurVideoStorecompany may have different databases and different structures. If the executive of the company wants to access the data from all stores for strategic decision-making, future direction, marketing, etc., it would be more appropriate to store all the data in one site with a homogeneous

structure that allows interactive analysis. In other words, data from the different stores would be loaded, cleaned, transformed and integrated together. To facilitate decision-making and multi-dimensional views, data warehouses are usually modeled by a multi-dimensional data structure. Figure 1.3 shows an example of a three dimensional subset of a data cube structure used for OurVideoStore data warehouse.

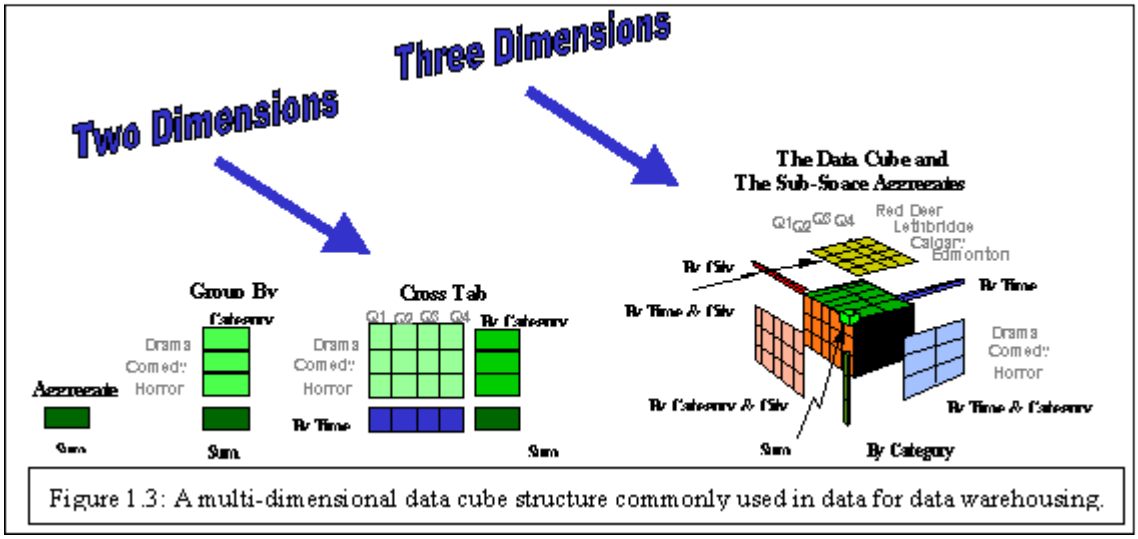


Figure 1.3: A multi-dimensional data cube structure commonly used in data for data warehousing.

The figure shows summarized rentals grouped by film categories, then a cross table of summarized rentals by film categories and time (in quarters).The data cube gives the summarized rentals along three dimensions: category, time, and city. A cube contains cells that store values of some aggregate measures (in this case rental counts), and special cells that store summations along dimensions. Each dimension of the data cube contains a hierarchy of values for one attribute.

Because of their structure, the pre-computed summarized data they contain and the hierarchical attribute values of their dimensions, data cubes are well suited for fast interactive querying and analysis of data at different conceptual levels, known as On-Line Analytical Processing (OLAP). OLAP operations allow the navigation of data at different levels of abstraction, such as drill-down, roll-up, slice, dice, etc. Figure 1.4 illustrates the drill-down (on the time dimension) and roll-up (on the location dimension) operations.

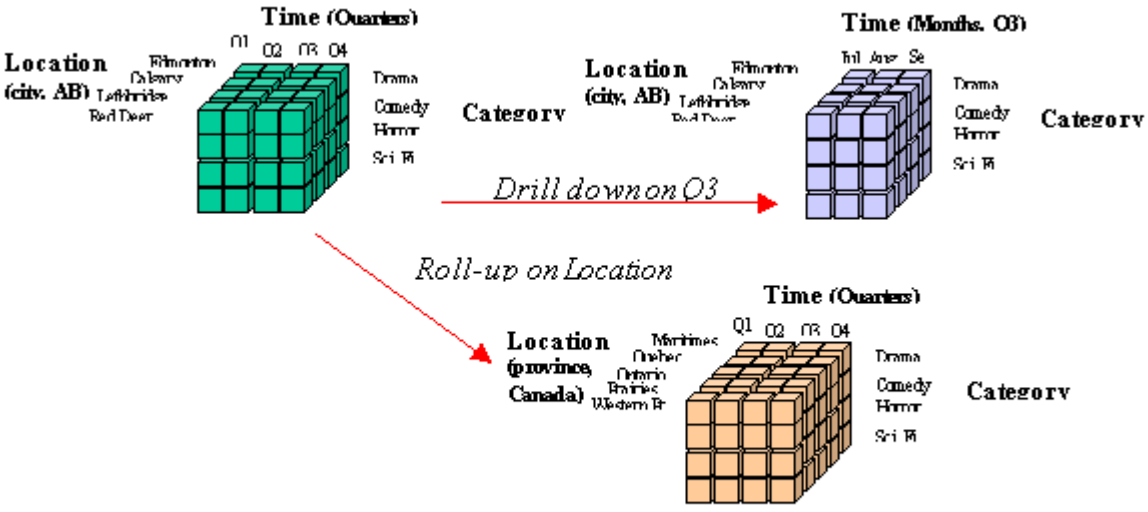


Figure 1.4: Summarized data from *OurVideoStore* before and after *drill-down* and *roll-up* operations.

Transaction Databases: A transaction database is a set of records representing transactions, each with a time stamp, an identifier and a set of items. Associated with the transaction files could also be descriptive data for the items. For example, in the case of the video store, the rentals table such as shown in Figure 1.5, represents the transaction database. Each record is a rental contract with a customer identifier, a date, and the list of items rented (i.e. video tapes, games, VCR, etc.). Since relational databases do not allow nested tables (i.e. a set as attribute value), transactions are usually stored in flat files or stored in two normalized transaction tables, one for the transactions and one for the transaction items. One typical data mining analysis on such data is the so-called market basket analysis or association rules in which associations between items occurring together or in sequence are studied.

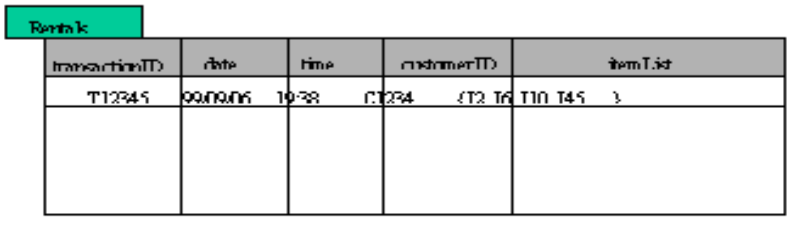


Figure 1.5: Fragment of a transaction database for the rentals at OurVideoStore.

Multimedia Databases: Multimedia databases include video, images, audio and text media. They can be stored on extended object-relational or object-oriented databases, or simply on a file system. Multimedia is characterized by its high dimensionality, which makes data mining even more challenging. Data mining from multimedia repositories may require computer vision, computer graphics, image interpretation, and natural language processing methodologies.

Spatial Databases: Spatial databases are databases that, in addition to usual data, store geographical information like maps, and global or regional positioning. Such spatial databases present new challenges to data mining algorithms.

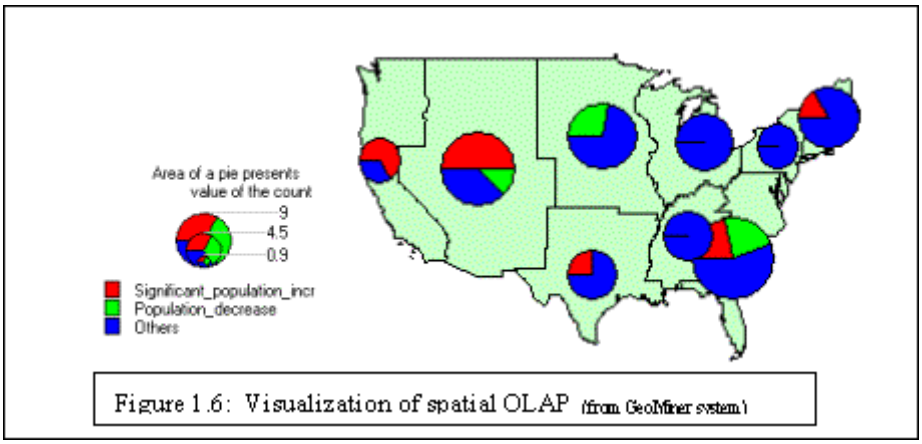


Figure 1.6: Visualization of spatial OLAP (from GeoMiner system)

Time-Series Databases: Time-series databases contain time related data such stock market data or logged activities. These databases usually have a continuous flow of new data coming in, which sometimes causes the need for a challenging real time analysis. Data mining in such databases commonly includes the study of trends and correlations between evolutions of different variables, as well as the prediction of trends and movements of the variables in time. Figure 1.7 shows some examples of time-series data.

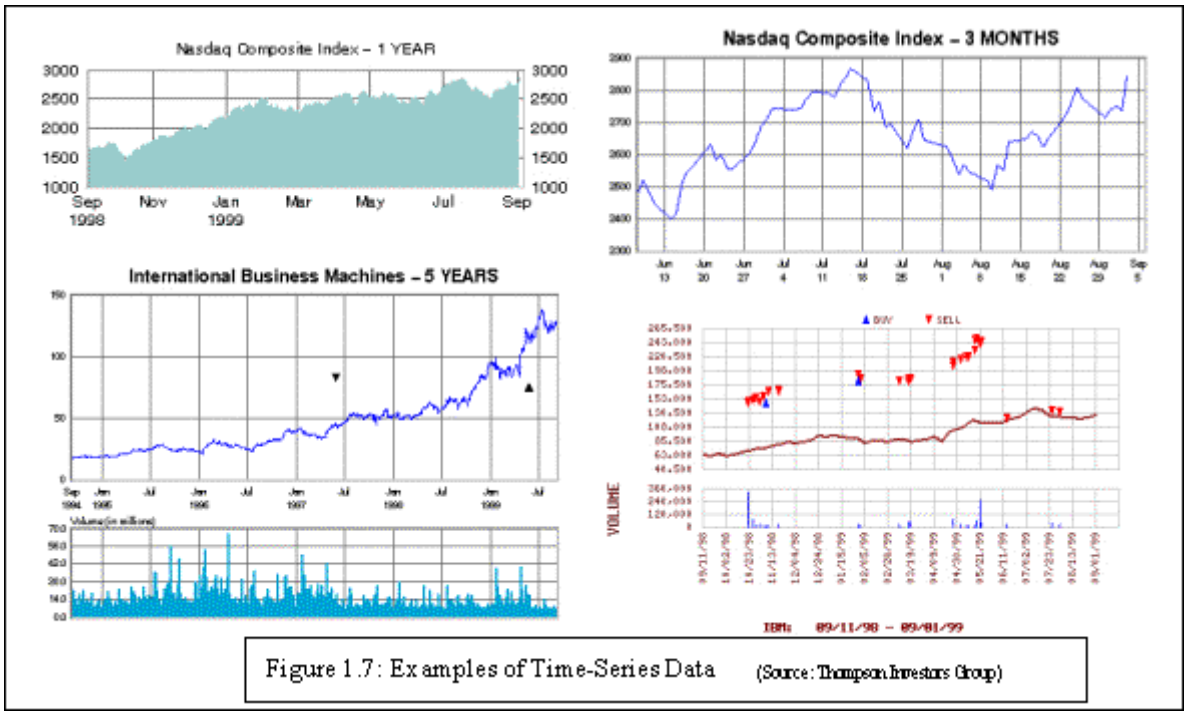


Figure 1.7: Examples of Time-Series Data (Source: Thompson Investors Group)

- **World Wide Web**: The World Wide Web is the most heterogeneous and dynamic repository available. A very large number of authors and publishers are continuously contributing to its growth and metamorphosis, and a massive number of users are accessing its resources daily. Data in the World Wide Web is organized in inter-connected documents. These documents can be text, audio, video, raw data, and even applications. Conceptually, the World Wide Web is comprised of three major components: The content of the Web, which encompasses documents available; the structure of the Web, which covers the hyperlinks and the relationships between documents; and the usage of the web, describing how and when the resources are accessed. A fourth dimension can be added relating the dynamic nature or evolution of the documents. Data mining in the World Wide Web, or web mining, tries to address all these issues and is often divided into web content mining, web structure mining and web usage mining.

Data Mining Functionalities

The data mining functionalities and the variety of knowledge they discover are briefly presented in the following list:

1. **Characterization**:
2. **Discrimination**:
3. **Association analysis**:
4. **Classification**:
5. **Prediction**:
6. **Clustering**:
7. **Outlier analysis**
8. **Evolution and deviation analysis**

- **Characterization**: Data characterization is a summarization of general features of objects in a target class, and produces what is called *characteristic rules*. The data relevant to a user-specified class are normally retrieved by a database query and run through a summarization module to extract the essence of the data at different levels of abstractions. For example, one may want to characterize the OurVideoStore customers who regularly rent more than 30 movies a year. With concept hierarchies on the attributes describing the target class, the *attribute-oriented induction* method can be used, for example, to carry out data summarization. Note that with a data cube containing summarization of data, simple OLAP operations fit the purpose of data characterization.
- **Discrimination**: Data discrimination produces what are called *discriminant rules* and is basically the comparison of the general features of objects between two classes referred to as the *target class* and the *contrasting class*. For example, one may want to compare the general characteristics of the customers who rented more than 30 movies in the last year with those whose rental account is lower than 5. The techniques used for data discrimination are very similar to the techniques used for data characterization with the exception that data discrimination results include comparative measures.
- **Association analysis**: Association analysis is the discovery of what are commonly called *association rules*. It studies the frequency of items occurring together in transactional databases, and based on a threshold called *support*, identifies the frequent item sets. Another threshold, *confidence*, which is the conditional probability than an item appears in a transaction when another item appears, is used

to pinpoint association rules. Association analysis is commonly used for market basket analysis. For example, it could be useful for the OurVideoStore manager to know what movies are often rented together or if there is a relationship between renting a certain type of movies and buying popcorn or pop. The discovered association rules are of the form: P -> Q [s,c], where P and Q are conjunctions of attribute value-pairs, and s (for support) is the probability that P and Q appear together in a transaction and c (for confidence) is the conditional probability that Q appears in a transaction when P is present. For example, the hypotheticassociation rule:

- *RentType(X, "game") AND Age(X, "13-19") -> Buys(X, "pop") [s=2% ,c=55%]* would indicate that 2% of the transactions considered are of customers aged between 13 and 19 who are renting a game and buying a pop, and that there is a certainty of 55% that teenage customers who rent a game also buy pop.
- **Classification**: Classification analysis is the organization of data in given classes. Also known as *supervised classification*, the classification uses given class labels to order the objects in the data collection. Classification approaches normally use a *training set* where all objects are already associated with known class labels. The classification algorithm learns from the training set and builds a model. The model is used to classify new objects. For example, after starting a credit policy, the OurVideoStore managers could analyze the customers'behaviours vis-à-vis their credit, and label accordingly the customers who received credits with three possible labels "safe", "risky" and "very risky". The classification analysis would generate a model that could be used to either accept or reject credit requests in the future.
- **Prediction**: Prediction has attracted considerable attention given the potential implications of successful forecasting in a business context. There are two major types of predictions: one can either try to predict some unavailable data values or pending trends, or predict a class label for some data. The latter is tied to classification. Once a classification model is built based on a training set, the class label of an object can be foreseen based on the attribute values of the object and the attribute values of the classes. Prediction is however more often referred to the forecast of missing numerical values, or increase/ decrease trends in time related

data. The major idea is to use a large number of past values to consider probable future values.

- **Clustering**: Similar to classification, clustering is the organization of data in classes. However, unlike classification, in clustering, class labels are unknown and it is up to the clustering algorithm to discover acceptable classes. Clustering is also called *unsupervised classification*, because the classification is not dictated by given class labels. There are many clustering approaches all based on the principle of maximizing the similarity between objects in a same class (*intra-class similarity*) and minimizing the similarity between objects of different classes (*inter-class similarity*).
- **Outlier analysis**: Outliers are data elements that cannot be grouped in a given class or cluster. Also known as *exceptions* or *surprises*, they are often very important to identify. While outliers can be considered noise and discarded in some applications, they can reveal important knowledge in other domains, and thus can be very significant and their analysis valuable.
- **Evolution and deviation analysis**: Evolution and deviation analysis pertain to the study of time related data that changes in time. Evolution analysis models evolutionary trends in data, which consent to characterizing, comparing, classifying or clustering of time related data. Deviation analysis, on the other hand, considers differences between measured values and expected values, and attempts to find the cause of the deviations from the anticipated values.

It is common that users do not have a clear idea of the kind of patterns they can discover or need to discover from the data at hand. It is therefore important to have a versatile and inclusive data mining system that allows the discovery of different kinds of knowledge and at different levels of abstraction. This also makes interactivity an important attribute of a data mining system.

How do we categorize data mining systems?((Classification of Data mining systems)

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

- **Classification according to the type of data source mined**: this classification categorizes data mining systems according to the type of data handled such as spatial data, multimedia data, time-series data, text data, World Wide Web, etc.
- **Classification according to the data model drawn on**: this classification categorizes data mining systems based on the data model involved such as relational database, object-oriented database, data warehouse, transactional, etc.
- **Classification according to the king of knowledge discovered**: this classification categorizes data mining systems based on the kind of knowledge discovered or data mining functionalities, such as characterization, discrimination, association, classification, clustering, etc. Some systems tend to be comprehensive systems offering several data mining functionalities together.
- **Classification according to mining techniques used**: Data mining systems employ and provide different techniques. This classification categorizes data mining systems according to the data analysis approach used such as machine learning, neural networks, genetic algorithms, statistics, visualization, database-oriented or data warehouse-oriented, etc. The classification can also take into account the degree of user interaction involved in the data mining process such as query-driven systems, interactive exploratory systems, or autonomous systems. A comprehensive system would provide a wide variety of data mining techniques to fit different situations and options, and offer different degrees of user interaction.

What are the issues in Data Mining?

Data mining algorithms embody techniques that have sometimes existed for many years, but have only lately been applied as reliable and scalable tools that time and again outperform older classical statistical methods. While data mining is still in its infancy, it is becoming a trend and ubiquitous. Before data mining develops into a conventional, mature and trusted discipline, many still pending issues have to be addressed. Some of these

issues are addressed below. Note that these issues are not exclusive and are not ordered in any way.

1. **Security and social issues**
2. **User interface issues**
3. **Mining methodology issues**:
4. **Performance issues**
5. **Data source issues**:

1. **Security and social issues**: Security is an important issue with any data collection that is shared and/or is intended to be used for strategic decision-making. In addition, when data is collected for customer profiling, user behavior understanding, correlating personal data with other information, etc., large amounts of sensitive and private information about individuals or companies is gathered and stored. This becomes controversial given the confidential nature of some of this data and the potential illegal access to the information. Moreover, data mining could disclose new implicit knowledge about individuals or groups that could be against privacy policies, especially if there is potential dissemination of discovered information. Another issue that arises from this concern is the appropriate use of data mining. Due to the value of data, databases of all sorts of content are regularly sold, and because of the competitive advantage that can be attained from implicit knowledge discovered, some important information could be withheld, while other information could be widely distributed and used without control.

2. **User interface issues**: The knowledge discovered by data mining tools is useful as long as it is interesting, and above all understandable by the user. Good data visualization eases the interpretation of data mining results, as well as helps users better understand their needs. Many data exploratory analysis tasks are significantly facilitated by the ability to see data in an appropriate visual presentation. There are many visualization ideas and proposals for effective data graphical presentation. However, there is still much research to accomplish in order to obtain good visualization tools for large datasets that could be used to display and manipulate

mined knowledge.�The major issues related to user interfaces and visualization are "screen real-estate", information rendering, and interaction. Interactivity with the data and data mining results is crucial since it provides means for the user to focus and refine the mining tasks, as well as to picture the discovered knowledge from different angles and at different conceptual levels.

3. **Mining methodology issues**: These issues pertain to the data mining approaches applied and their limitations. Topics such as versatility of the mining approaches, the diversity of data available, the dimensionality of the domain, the broad analysis needs (when known), the assessment of the knowledge discovered, the exploitation of background knowledge and metadata, the control and handling of noise in data, etc. are all examples that can dictate mining methodology choices. For instance, it is often desirable to have different data mining methods available since different approaches may perform differently depending upon the data at hand. Moreover, different approaches may suit and solve user's needs differently.

4. Most algorithms assume the data to be noise-free. This is of course a strong assumption. Most datasets contain exceptions, invalid or incomplete information, etc., which may complicate, if not obscure, the analysis process and in many cases compromise the accuracy of the results. As a consequence, data preprocessing (data cleaning and transformation) becomes vital. It is often seen as lost time, but data cleaning, as time-consuming and frustrating as it may be, is one of the most important phases in the knowledge discovery process. Data mining techniques should be able to handle noise in data or incomplete information.

5. More than the size of data, the size of the search space is even more decisive for data mining techniques. The size of the search space is often depending upon the number of dimensions in the domain space. The search space usually grows

exponentially when the number of dimensions increases. This is known as the *curse of dimensionality*. This "curse" affects so badly the performance of some data mining approaches that it is becoming one of the most urgent issues to solve

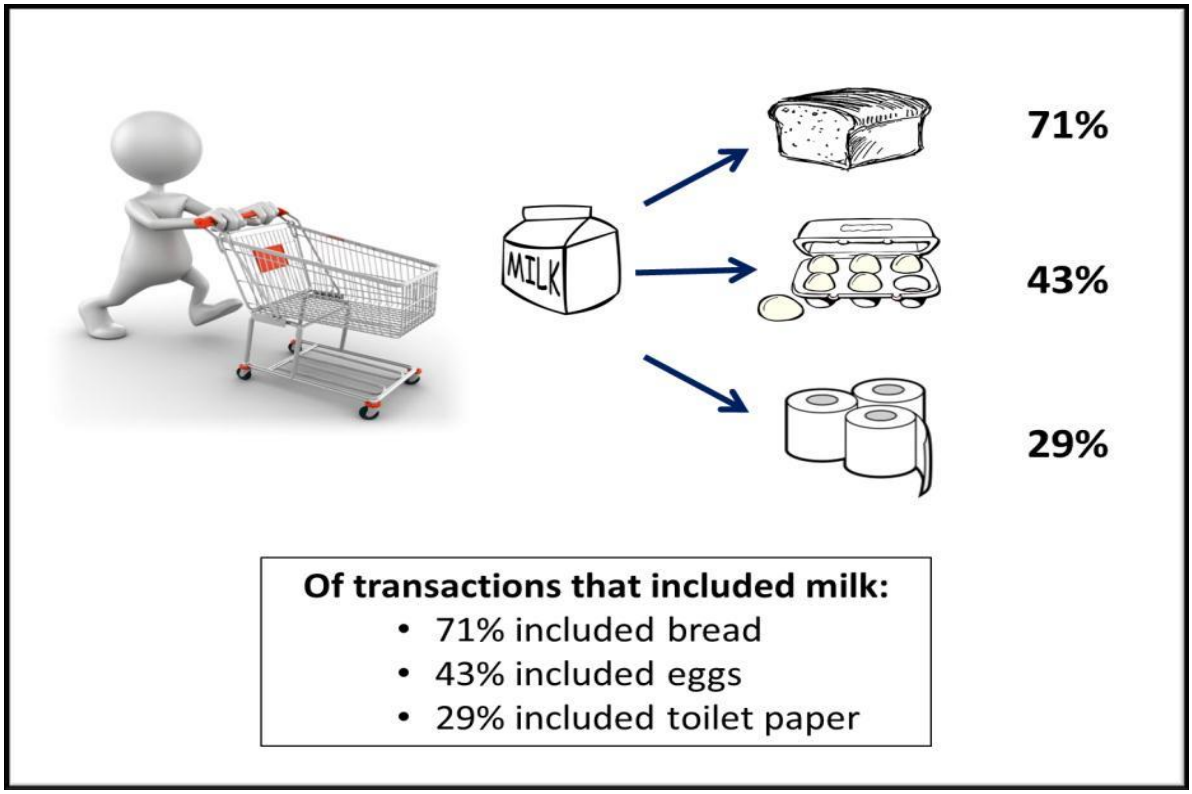
6. **Performance issues**: Many artificial intelligence and statistical methods exist for data analysis and interpretation. However, these methods were often not designed for the very large data sets data mining is dealing with today. Terabyte sizes are common. This raises the issues of scalability and efficiency of the data mining methods when processing considerably large data. Algorithms with exponential and even medium-order polynomial complexity cannot be of practical use for data mining. Linear algorithms are usually the norm. In same theme, sampling can be used for mining instead of the whole dataset. However, concerns such as completeness and choice of samples may arise. Other topics in the issue of performance are *incremental updating*, and parallel programming.There is no doubt that parallelism can help solve the size problem if the dataset can be subdivided and the results can be merged later. Incremental updating is important for merging results from parallel mining, or updating data mining results when new data becomes available without having to re-analyze the complete dataset.

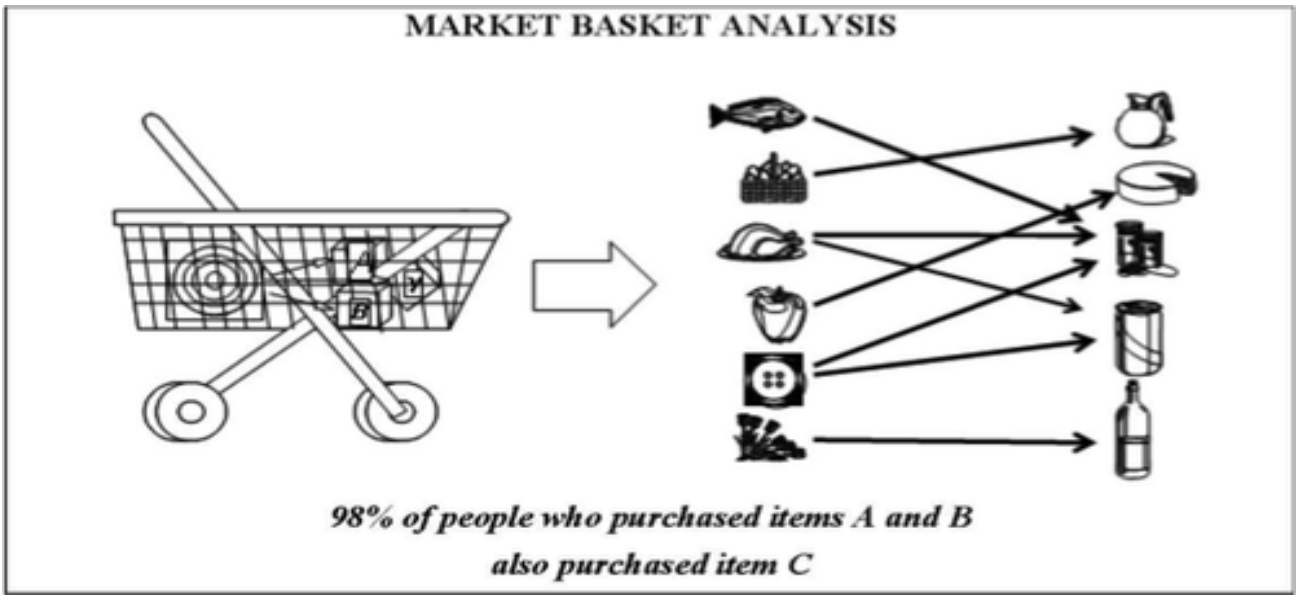
7. **Data source issues**: There are many issues related to the data sources, some are practical such as the diversity of data types, while others are philosophical like the data glut problem. We certainly have an excess of data since we already have more data than we can handle and we are still collecting data at an even higher rate. If the spread of database management systems has helped increase the gathering of information, the advent of data mining is certainly encouraging more data harvesting. The current practice is to collect as much data as possible now and process it, or try to process it, later. The concern is whether we are collecting the right data at the appropriate amount, whether we know what we want to do with it, and whether we distinguish between what data is important and what data is

insignificant. Regarding the practical issues related to data sources, there is the subject of heterogeneous databases and the focus on diverse complex data types. We are storing different types of data in a variety of repositories. It is difficult to expect a data mining system to effectively and efficiently achieve good mining results on all kinds of data and sources. Different kinds of data and sources may require distinct algorithms and methodologies. Currently, there is a focus on relational databases and data warehouses, but other approaches need to be pioneered for other specific complex data types. A versatile data mining tool, for all sorts of data, may not be realistic. Moreover, the proliferation of heterogeneous data sources, at structural and semantic levels, poses important challenges not only to the database community but also to the data mining community.

Module IV :Association Rule data mining
Market basket analysis, basic concepts
Road Map, Classification of Association rules

Market basket analysis is the study of items that are purchased (or otherwise grouped) together in a single transaction or multiple, sequential transactions. Understanding the relationships and the strength of those relationships is valuable information that can be used to make recommendations, cross-sell, up-sell, offer coupons, etc. The analysis reveals patterns such as that of the well-known study which found an association between purchases of diapers and beer.





Market Basket Example

For more detail visit: https://www.youtube.com/watch?v=umMA4wfJ4ac

How is it used?

In retailing, *most purchases are bought on impulse*. Market basket analysis gives clues as to what a customer might have bought *if the idea had occurred to them.* (For some real insights into consumer behavior, see Why We Buy: The Science of Shopping by Paco Underhill.)

As a first step, therefore, market basket analysis can be used in deciding the location and promotion of goods inside a store. If, as has been observed, purchasers of Barbie dolls have are more likely to buy candy, then high-margin candy can be placed near to the Barbie doll display. Customers who would have bought candy with their Barbie dolls *had they thought of it* will now be suitably tempted.

But this is only the first level of analysis. **Differential market basket analysis** can find interesting results and can also eliminate the problem of a potentially high volume of trivial results.

In differential analysis, we compare results between different stores, between customers in different demographic groups, between different days of the week, different seasons of the year, etc.

If we observe that a rule holds in one store, but not in any other (or does not hold in one store, but holds in all others), then we know that there is something interesting about that store. Perhaps its clientele are different, or perhaps it has organized its displays in a novel and more lucrative way. Investigating such differences may yield useful insights which will improve company sales.

Other Application Areas

Although Market Basket Analysis conjures up pictures of shopping carts and supermarket shoppers, it is important to realize that there are many other areas in which it can be applied. These include:

- Analysis of credit card purchases.
- Analysis of telephone calling patterns.
- Identification of fraudulent medical insurance claims. (Consider cases where common rules are broken).
- Analysis of telecom service purchases.

Module V: Classification and prediction	
	What is classification, what is prediction?
	Issues Regarding Classification and prediction
	Comparing classification methods
	Classification by Decision Tree Induction
	Attributes selection methods
	Attributes selection methods

Data Mining - Classification & Prediction

There are two forms of data analysis that can be used for extracting models describing important classes or to predict future data trends. These two forms are as follows −

- Classification
- Prediction

Classification models predict categorical class labels; and prediction models predict continuous valued functions. For example, we can build a classification model to categorize bank loan applications as either safe or risky, or a prediction model to predict the expenditures in dollars of potential customers on computer equipment given their income and occupation.

What is classification?

Following are the examples of cases where the data analysis task is Classification −

- A bank loan officer wants to analyze the data in order to know which customer (loan applicant) are risky or which are safe.
- A marketing manager at a company needs to analyze a customer with a given profile, who will buy a new computer.

In both of the above examples, a model or classifier is constructed to predict the categorical labels. These labels are risky or safe for loan application data and yes or no for marketing data.

What is prediction?

Following are the examples of cases where the data analysis task is Prediction −

Suppose the marketing manager needs to predict how much a given customer will spend during a sale at his company. In this example we are bothered to predict a numeric value. Therefore the data analysis task is an example of numeric prediction. In this case, a model or a predictor will be constructed that predicts a continuous-valued-function or ordered value.

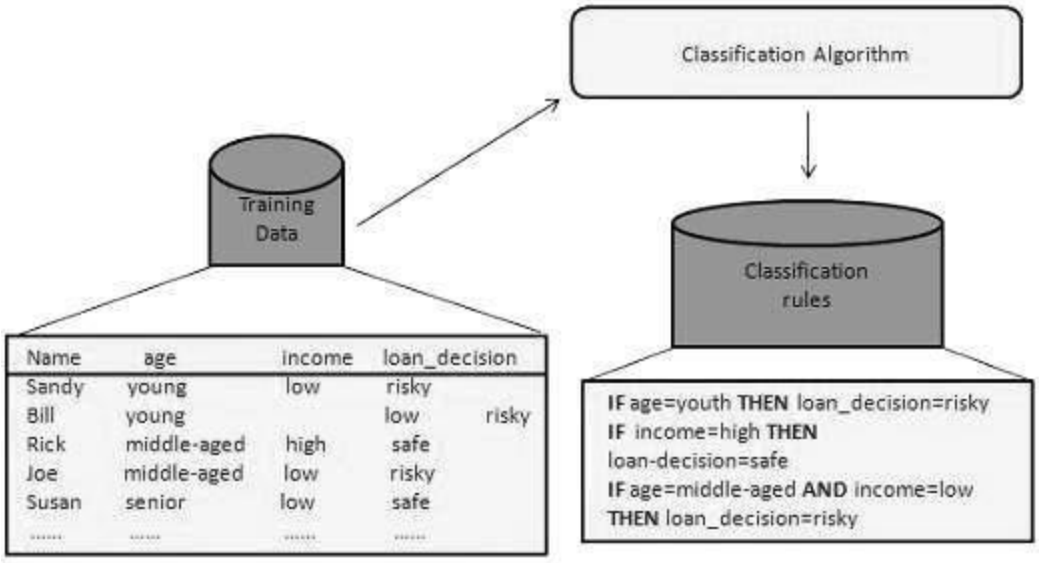
How Does Classification Works?

With the help of the bank loan application that we have discussed above, let us understand the working of classification. The Data Classification process includes two steps −

- Building the Classifier or Model
- Using Classifier for Classification

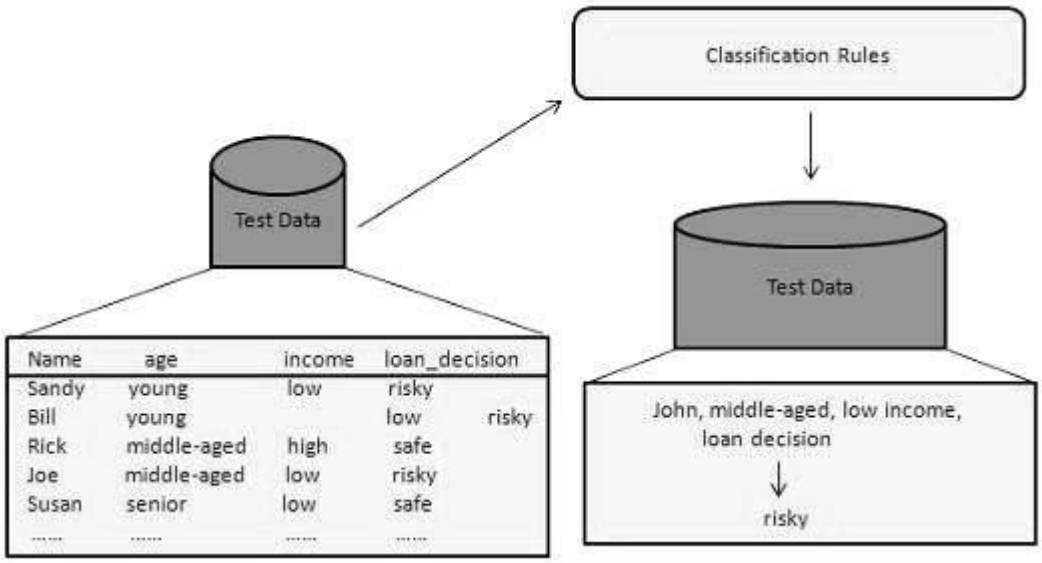
Building the Classifier or Model

- This step is the learning step or the learning phase.
- In this step the classification algorithms build the classifier.
- The classifier is built from the training set made up of database tuples and their associated class labels.
- Each tuple that constitutes the training set is referred to as a category or class. These tuples can also be referred to as sample, object or data points.



Using Classifier for Classification

In this step, the classifier is used for classification. Here the test data is used to estimate the accuracy of classification rules. The classification rules can be applied to the new data tuples if the accuracy is considered acceptable.



Classification and Prediction Issues

The major issue is preparing the data for Classification and Prediction. Preparing the data involves the following activities −

- **Data Cleaning** − Data cleaning involves removing the noise and treatment of missing values. The noise is removed by applying smoothing techniques and the problem of missing values is solved by replacing a missing value with most commonly occurring value for that attribute.
- **Relevance Analysis** − Database may also have the irrelevant attributes. Correlation analysis is used to know whether any two given attributes are related.
- **Data Transformation and reduction** − the data can be transformed by any of the following methods.
    - **Normalization** − the data is transformed using normalization. Normalization involves scaling all values for given attribute in order to make them fall within a small specified range. Normalization is used when in the learning step, the neural networks or the methods involving measurements are used.
    - **Generalization** − the data can also be transformed by generalizing it to the higher concept. For this purpose we can use the concept hierarchies.

Comparison of Classification and Prediction Methods

Here is the criteria for comparing the methods of Classification and Prediction −

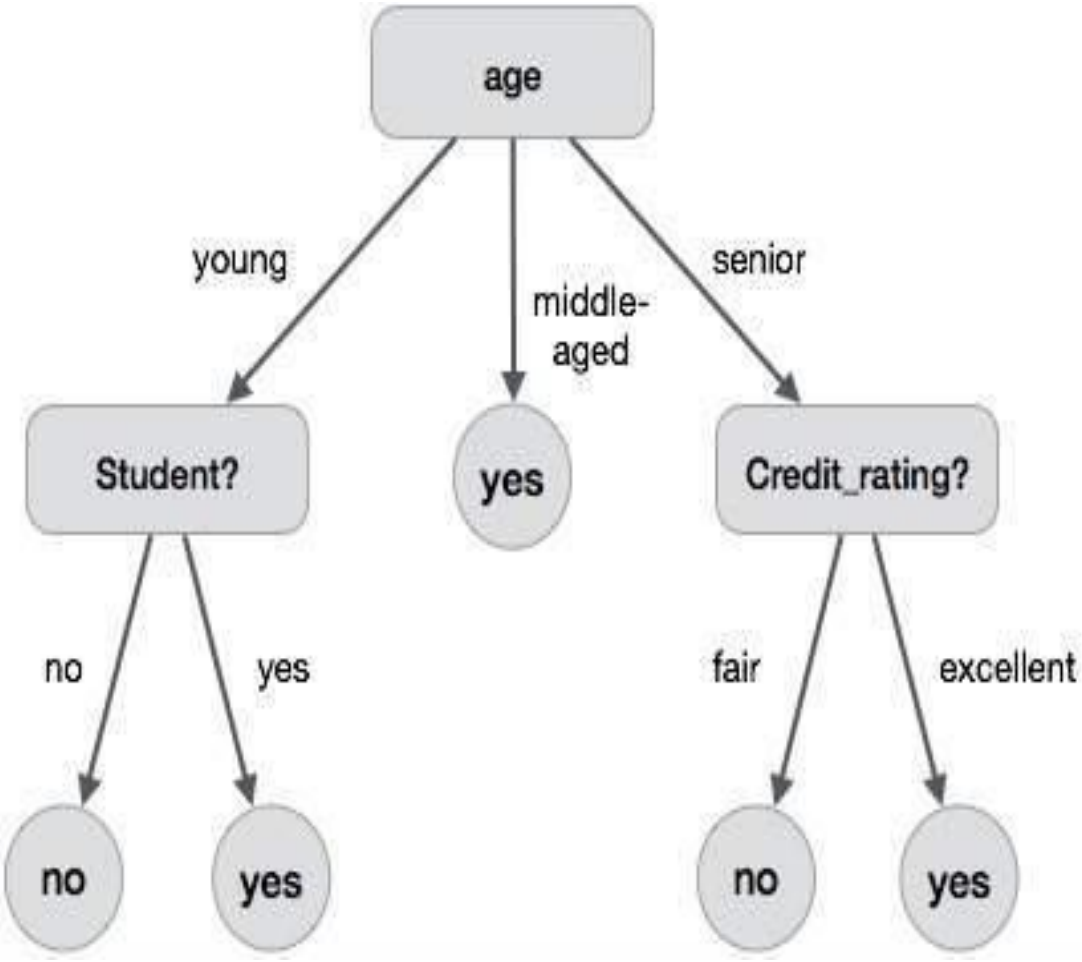
- **Accuracy** − Accuracy of classifier refers to the ability of classifier. It predict the class label correctly and the accuracy of the predictor refers to how well a given predictor can guess the value of predicted attribute for a new data.
- **Speed** − This refers to the computational cost in generating and using the classifier or predictor.
- **Robustness** − It refers to the ability of classifier or predictor to make correct predictions from given noisy data.
- **Scalability** − Scalability refers to the ability to construct the classifier or predictor efficiently; given large amount of data.

- **Interpretability** − It refers to what extent the classifier or predictor understands.

Classification by Decision Tree Induction

A decision tree is a structure that includes a root node, branches, and leaf nodes. Each internal node denotes a test on an attribute, each branch denotes the outcome of a test, and each leaf node holds a class label. The topmost node in the tree is the root node.

The following decision tree is for the concept buy computer that indicates whether a customer at a company is likely to buy a computer or not. Each internal node represents a test on an attribute. Each leaf node represents a class.



The benefits of having a decision tree are as follows −

- It does not require any domain knowledge.

- It is easy to comprehend.
- The learning and classification steps of a decision tree are simple and fast.

Attribute selection

In machine learning and statistics, **feature selection**, also known as **variable selection**, **attribute selection** or **variable subset selection**, is the process of selecting a subset of relevant features (variables, predictors) for use in model construction. Feature selection techniques are used for three reasons:

- simplification of models to make them easier to interpret by researchers/users,
- shorter training times,
- enhanced generalization by reducing over fitting

The central premise when using a feature selection technique is that the data contains many features that are either *redundant* or *irrelevant*, and can thus be removed without incurring much loss of information.*Redundant* or *irrelevant* features are two distinct notions, since one relevant feature may be redundant in the presence of another relevant feature with which it is strongly correlated.

Feature selection techniques should be distinguished from feature extraction. Feature extraction creates new features from functions of the original features, whereas feature selection returns a subset of the features. Feature selection techniques are often used in domains where there are many features and comparatively few samples (or data points). Archetypal cases for the application of feature selection include the analysis of written texts and DNA microarray data, where there are many thousands of features, and a few tens to hundreds of samples.

Main principles

The feature selection methods are typically presented in three classes based on how they combine the selection algorithm and the model building

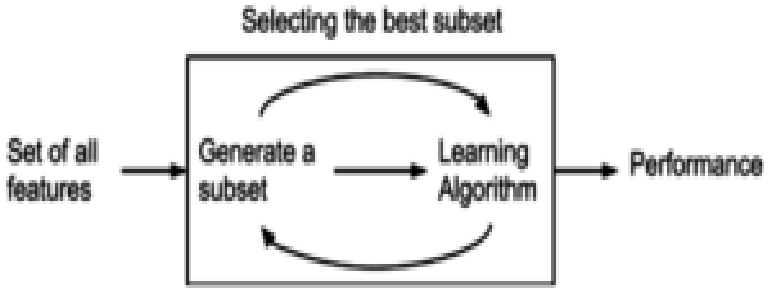
Filter Method

Filter Method for feature selection

Filter type methods select variables regardless of the model. They are based only on general features like the correlation with the variable to predict. Filter methods suppress the least interesting variables. The others variables will be part of a model classification, a regression used to classify or a data prediction. These methods are particularly effective in computation time and robust to overfitting.

However, filter methods tend to select redundant variables because they do not consider the relationships between variables. Therefore, they are mainly used as a pre-process method.

Wrapper Method

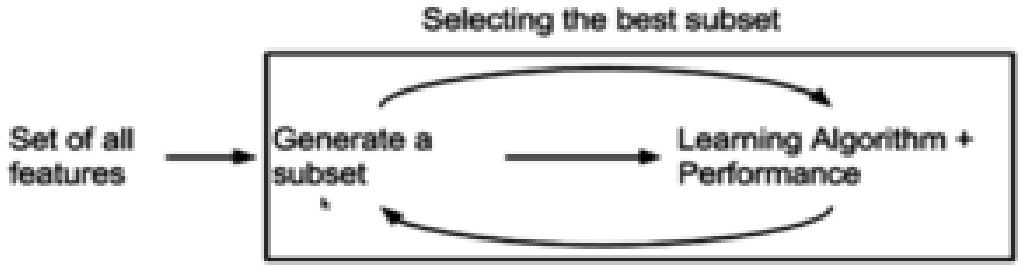


Wrapper Method for Feature selection

Wrapper methods evaluate subsets of variables which allows, unlike filter approaches, to detect the possible interactions between variables.[The two main disadvantages of these methods are :

- The increasing overfitting risk when the number of observations is insufficient.
- The significant computation time when the number of variables is large.

Embedded Method



Embedded method for Feature selection

Recently, embedded methods have been proposed to reduce the classification of learning. They try to combine the advantages of both previous methods. The learning algorithm takes advantage of its own variable selection algorithm. So, it needs to know preliminary what a good selection is, which limits their exploitation.

Module VI: Cluster Analysis
Introduction, Need,
Major Clustering Method
Partitioning Method K-Mean Method,
K-Medios Method

What is Cluster?

Cluster is a group of objects that belongs to the same class. In other words, similar objects are grouped in one cluster and dissimilar objects are grouped in another cluster.

What is Clustering?

Clustering is the process of making a group of abstract objects into classes of similar objects.

Points to Remember

- A cluster of data objects can be treated as one group.
- While doing cluster analysis, we first partition the set of data into groups based on data similarity and then assign the labels to the groups.
- The main advantage of clustering over classification is that, it is adaptable to changes and helps single out useful features that distinguish different groups.

Applications of Cluster Analysis

- Clustering analysis is broadly used in many applications such as market research, pattern recognition, data analysis, and image processing.
- 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 also helps in identification of areas of similar land use in an earth observation database. It also helps in the identification of groups of houses in a city according to house type, value, and geographic location.
- Clustering also helps in classifying documents on the web for information discovery.
- Clustering is also used in outlier detection applications such as detection of credit card fraud.
- As a data mining function, cluster analysis serves as a tool to gain insight into the distribution of data to observe characteristics of each cluster.

Requirements of Clustering in Data Mining

The following points throw light on why clustering is required in data mining −

- **Scalability** − We need highly scalable clustering algorithms to deal with large databases.
- **Ability to deal with different kinds of attributes** − Algorithms should be capable to be applied on any kind of data such as interval-based (numerical) 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 to only distance measures that tend to find spherical cluster of small sizes.
- **High dimensionality** − The clustering algorithm should not only be able to handle low-dimensional data but also the high dimensional space.
- **Ability to deal with noisy data** − Databases contain noisy, missing or erroneous data. Some algorithms are sensitive to such data and may lead to poor quality clusters.
- **Interpretability** − The clustering results should be interpretable, comprehensible, and usable.

Clustering Methods

Clustering methods can be classified into the following categories −

- Partitioning Method
- Hierarchical Method
- Density-based Method
- Grid-Based Method
- Model-Based Method
- Constraint-based Method

Partitioning Method

Suppose we are given a database of 'n' objects and the partitioning method constructs 'k' partition of data. Each partition will represent a cluster and k ≤ n. It means that it will classify the data into k groups, which satisfy the following requirements −

- Each group contains at least one object.
- Each object must belong to exactly one group.

Points to remember −

- For a given number of partitions (say k), the partitioning method will create an initial partitioning.
- Then it uses the iterative relocation technique to improve the partitioning by moving objects from one group to other.
- K-Means Clustering is an unsupervised learning algorithm that is used to solve the clustering problems in machine learning or data science. In this topic, we will learn what is K-means clustering algorithm, how the algorithm works, along with the Python implementation of k-means clustering.

What is K-Means Algorithm?

- K-Means Clustering is an Unsupervised Learning algorithm, which groups the unlabeled dataset into different clusters. Here K defines the number of pre-defined clusters that need to be created in the process, as if K=2, there will be two clusters, and for K=3, there will be three clusters, and so on.

- It is an iterative algorithm that divides the unlabeled dataset into k different clusters in such a way that each dataset belongs only one group that has similar properties.

- It allows us to cluster the data into different groups and a convenient way to discover the categories of groups in the unlabeled dataset on its own without the need for any training.
- It is a centroid-based algorithm, where each cluster is associated with a centroid. The main aim of this algorithm is to minimize the sum of distances between the data point and their corresponding clusters.

Hierarchical Methods

This method creates a hierarchical decomposition of the given set of data objects. We can classify hierarchical methods on the basis of how the hierarchical decomposition is formed. There are two approaches here −

- Agglomerative Approach
- Divisive Approach

Agglomerative Approach

This approach is also known as the bottom-up approach. In this, we start with each object forming a separate group. It keeps on merging the objects or groups that are close to one another. It keep on doing so until all of the groups are merged into one or until the termination condition holds.

Divisive Approach

This approach is also known as the top-down approach. In this, we start with all of the objects in the same cluster. In the continuous iteration, a cluster is split up into smaller clusters. It is down until each object in one cluster or the termination condition holds. This method is rigid, i.e., once a merging or splitting is done, it can never be undone.

Approaches to Improve Quality of Hierarchical Clustering

Here are the two approaches that are used to improve the quality of hierarchical clustering −

- Perform careful analysis of object linkages at each hierarchical partitioning.

- Integrate hierarchical agglomeration by first using a hierarchical agglomerative algorithm to group objects into micro-clusters, and then performing macro-clustering on the micro-clusters.

Density-based Method

This method is based on the notion of density. The basic idea is to continue growing the given cluster as long as the density in the neighborhood exceeds some threshold, i.e., for each data point within a given cluster, the radius of a given cluster has to contain at least a minimum number of points.

Grid-based Method

In this, the objects together form a grid. The object space is quantized into finite number of cells that form a grid structure.

Advantage

- The major advantage of this method is fast processing time.
- It is dependent only on the number of cells in each dimension in the quantized space.

Model-based methods

In this method, a model is hypothesized for each cluster to find the best fit of data for a given model. This method locates the clusters by clustering the density function. It reflects spatial distribution of the data points.

This method also provides a way to automatically determine the number of clusters based on standard statistics, taking outlier or noise into account. It therefore yields robust clustering methods.

Constraint-based Method

In this method, the clustering is performed by the incorporation of user or application-oriented constraints. A constraint refers to the user expectation or the properties of desired clustering results. Constraints provide us with an interactive way of communication with the clustering process. Constraints can be specified by the user or the application requirement.

K-Mediods

K-Medoids (also called as Partitioning AroundMedoid) algorithm was proposed in 1987 by Kaufman and Rousseeuw. A medoid can be defined as the point in the cluster, whose dissimilarities with all the other points in the cluster is minimum.

The dissimilarity of the medoid(Ci) and object(Pi) is calculated by using $E = |Pi - Ci|$

The **time complexity** is $O(k*(n-k)^2$

Advantages:

1. It is simple to understand and easy to implement.
2. K-Medoid Algorithm is fast and converges in a fixed number of steps.
3. PAM is less sensitive to outliers than other partitioning algorithms.

Disadvantages:

1. The main disadvantage of K-Medoid algorithms is that it is not suitable for clustering non-spherical (arbitrary shaped) groups of objects. This is because it relies on minimizing the distances between the non-medoid objects and the medoid (the cluster centre) – briefly, it uses compactness as clustering criteria instead of connectivity.
2. It may obtain different results for different runs on the same dataset because the first k medoids are chosen randomly.

B.C.A. (2010 COURSE SEM- VI : SUMMER - 2018

SUBJECT: DATA WAREHOUSING AND DATA MINING

Day : **Saturday**
Date : **28/04/2018**

S-2018-1744

Time: **10.00 AM TO 01.00 PM**
Max. Marks: 70.

N.B.:

1) Q. No. 1 is **COMPULSORY.**
2) Attempt any **FOUR** questions from Q. No. 2 to Q. No.7.
3) Figures to the **RIGHT** indicate full marks.

Q.1 Explain an application of data mining in financial data analysis. **(14)**

Q.2 What is a Data Warehouse? Explain the need for a separate Data warehouse apart from operational systems. **(14)**

Q.3 Explain why data cleaning and data transformation functions are considered to be a vital task in data integration process. **(14)**

Q.4 "Clustering is called unsupervised classification." State true or false and justify. **(14)**

Q.5 What is classification? Explain classification by Decision Tree induction with an example. **(14)**

Q.6 What are the various OLAP operations performed to mine data from data warehouse? Briefly explain each of them. **(14)**

Q.7 Write short notes on any **TWO** of the following: **(14)**

a) Data marts and types of data marts

b) K-means clustering algorithm

c) Market Basket Analysis.

* * *

Notes