DATA WAREHOUSING AND DATA MINING

Subject Code: 10IS74/10CS755 | **I.A. Marks : 25**
Hours/Week: 04 | **Exam Hours: 03**
Total Hours: 52 | **Exam Marks: 100**

PART – A

UNIT – 1 **6 Hours**
Data Warehousing: Introduction, Operational Data Stores (ODS), Extraction Transformation Loading (ETL), Data Warehouses. Design Issues, Guidelines for Data Warehouse Implementation, Data Warehouse Metadata

UNIT – 2 **6 Hours**
Online Analytical Processing (OLAP): Introduction, Characteristics of OLAP systems, Multidimensional view and Data cube, Data Cube Implementations, Data Cube operations, Implementation of OLAP and overview on OLAP Softwares.

UNIT – 3 **6 Hours**
Data Mining: Introduction, Challenges, Data Mining Tasks, Types of Data, Data Preprocessing, Measures of Similarity and Dissimilarity, Data Mining Applications

UNIT – 4 **8 Hours**
Association Analysis: Basic Concepts and Algorithms: Frequent Itemset Generation, Rule Generation, Compact Representation of Frequent Itemsets, Alternative methods for generating Frequent Itemsets, FP Growth Algorithm, Evaluation of Association Patterns

PART - B

UNIT – 5 **6 Hours**
Classification -1 : Basics, General approach to solve classification problem, Decision Trees, Rule Based Classifiers, Nearest Neighbor Classifiers.

UNIT – 6 **6 Hours**
Classification - 2: Bayesian Classifiers, Estimating Predictive accuracy of classification methods, Improving accuracy of clarification methods, Evaluation criteria for classification methods, Multiclass Problem.

UNIT – 7 **8 Hours**
Clustering Techniques: Overview, Features of cluster analysis, Types of Data and Computing Distance, Types of Cluster Analysis Methods, Partitional Methods, Hierarchical Methods, Density Based Methods, Quality and Validity of Cluster Analysis

UNIT – 8 **6 Hours**
Web Mining: Introduction, Web content mining, Text Mining, Unstructured Text, Text clustering, Mining Spatial and Temporal Databases.

Text Books:

1. Pang-Ning Tan, Michael Steinbach, Vipin Kumar: Introduction to Data Mining, Addison-Wesley, 2005.
2. G. K. Gupta: Introduction to Data Mining with Case Studies, 3rd Edition, PHI, New Delhi, 2009.

TABLE OF CONTENTS

UNIT 1: DATA WAREHOUSING

INTRODUCTION

• A large company might have the following systems:
→Human resources(HR) →Financials →Billing
→Sales leads →Web sales →Customer support
Such systems are called *OLTP systems* (OnLine Transaction Processing).
• The systems are mostly relational database systems designed for transaction processing.
• The performance of OLTP systems is usually very important,
since such systems are used to support users(i.e. staff) who provide service to customers.
• The systems must be able to deal with
→ insert & update operations and
→ quickly answering queries
• The systems are not designed to handle management-queries efficiently (since management-queries are often complex, requiring many joins & aggregations).
• Focus of operational-managers is on improving business-processes across various business-units (for example: customer support, inventory, and marketing). To achieve this, they require:
→ a single sign-on path to the enterprise-information
→ a single version of the enterprise-information
→ a high level of data accuracy
→ a user-friendly interface to the information
→ easy sharing of data across business-units of enterprise
• Following are some solutions to meet the needs of management-staff in an enterprise:
1) Managers pose queries of interest to a mediator-system that
→ decomposes each query into appropriate subqueries for the systems
→ obtains results from those systems and
→ then combines & presents the result to the user
This is called *lazy (or on-demand) query processing*.
Advantage: The user is provided with up-to-date information.
Disadvantage: The management-queries may generate a heavy load on some OLTP systems which may degrade the performance of the systems.
2) Collect the most common queries that the managers ask and
then run them regularly and
finally have the results available when a manager poses one of those queries.
This approach is called the *eager query processing*.
Advantage: Provides quick response.
Disadvantage: The information may not be up-to-date.
3) This approach
→ involves creation of a separate database that only stores information that is of interest to the management-staff and
→ involves the following 2 steps:
i) The information needs of management-staff are analyzed and the systems that store some or all of the information are identified.
ii) The new database is then used to answer management-queries and the OLTP systems are not accessed for such queries.

ODS

• ODS stands for Operational Data Store.

• This is defined as a subject-oriented, integrated, volatile, current-valued data store, containing only corporate-detailed data.

→ ODS is subject-oriented i.e. it is organized around the major data-subjects of an enterprise

→ ODS is integrated i.e. it is a collection of data from a variety of systems to provide an enterprise-wide view of the data

→ ODS is volatile i.e. data in the ODS changes frequently as new information refreshes ODS

→ ODS is current-valued i.e. it is up-to-date and reflects the current status of the information

→ ODS is detailed i.e. it is detailed enough to serve needs of management-staff in enterprise

• Benefits of ODS to an enterprise:

1) An ODS is the unified-operational view of enterprise. It provides operational-managers improved access to important operational-data. This view may assist in better understanding of business & customer.

2) The ODS can be much more effective in generating current reports without having to access the OLTP.

3) The ODS can shorten the time required to implement & populate a data warehouse system.

• An ODS may be of following types:

1) An ODS that is essentially a reporting-tool for administrative purposes may be fairly simple. Such an ODS is usually updated daily. (It provides reports about business transactions for that day, such as sales totals or orders filled).

2) An ODS may be designed to track more complex information such as product & location codes. Such an ODS is usually updated hourly.

3) An ODS may be designed to support CRM (Customer Relationship Management).

ODS DESIGN & IMPLEMENTATION

• The extraction of information from source databases needs to be efficient (Figure 7.1).

• The quality of data needs to be maintained.

• Suitable checks are required to ensure quality of data after each refresh.

• An ODS would be required to satisfy normal integrity constraints, for example,

→ existential- & referential-integrity

→ appropriate action to deal with nulls

• An ODS is a read only database. Users should not be allowed to update information in ODS.

• Populating an ODS involves an acquisition process of extracting, transforming & loading data from source systems. This process is called *ETL(*Extraction, Transformation and Loading).

• Checking for anomalies & testing for performance are necessary before an ODS system can go online.

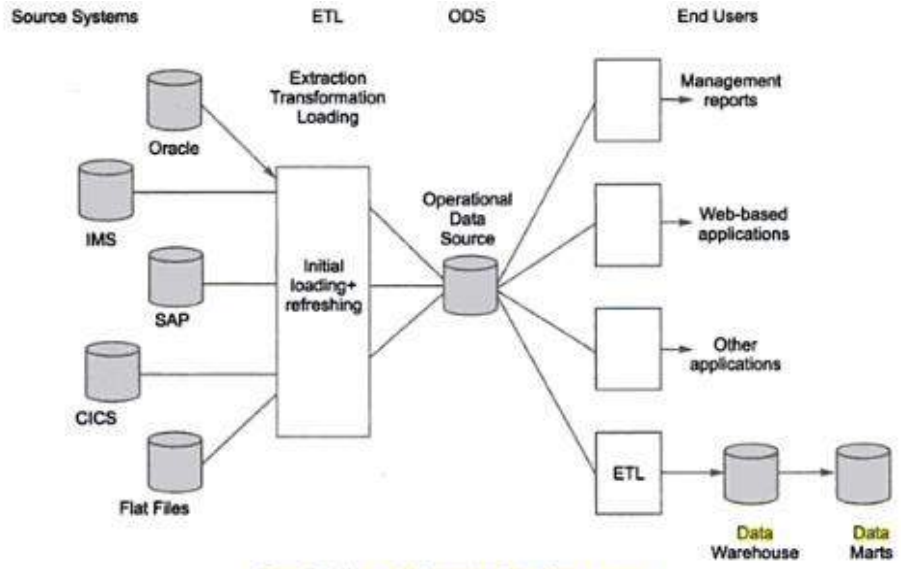


Figure 7.1 A possible Operational Data Store structure.

WHY A SEPARATE DATABASE?

• Q: Why an ODS should be separate from the operational-databases?

Ans: Because from time to time, complex queries are likely to degrade performance of OLTP systems. The OLTP systems have to provide a quick response to operational-users as businesses cannot afford to have response time suffer when a manager is running a complex query.

ZLE

• ZLE stands for Zero Latency Enterprise.

• This is used for near real-time integration of operational-data so that there is no significant delay in getting information from one system to another system in an enterprise.

• The heart of a ZLE system is an ODS.

• A ZLE data store(ZDS) is something like an ODS that is integrated & up-to-date.

• Aim of a ZLE data store is to:

→ allow management a single view of enterprise-information by bringing together relevant data in real-time &

→ provide management a "360-degree" view of the customer

• Characteristics of ZLE:

→ It has a unified view of the enterprise operational-data.

→ It has a high level of availability and it involves online refreshing of information.

• Since a ZLE needs to support a large number of concurrent users (for example call centre users), a fast turnaround time for transactions and 24/7 availability is required.

• Q: How does a ZLE data store fit into enterprise-information architecture?

Ans: Either a ZLE data store can be used almost like an ODS that is refreshed in real-time or a ZLE can take over some of the roles of the data-warehouse.

ETL

• The ETL process involves extracting, transforming and loading data from multiple source-systems.

• The process is much more complex and tedious. The process may require significant resources to implement.

• Different data-sources tend to have

→ different conventions for coding information &

→ different standards for the quality of information

• Building an ODS requires data filtering, data cleaning and integration.

• Data-errors at least partly arise because of unmotivated data-entry staff.

• Successful implementation of an ETL system involves resolving following issues:

1) What are the source systems? These systems may include relational database systems, legacy systems.
2) To what extent are the source systems and the target system interoperable? The more different the sources and target, the more complex the ETL process.
3) What ETL technology will be used?
4) How big is the ODS likely to be at the beginning and in the long term? Database systems tend to grow with time. Consideration may have to be given to whether some of the data
   → from the ODS will be archived regularly as the data becomes old and
   → is no longer needed in the ODS
5) How frequently will the ODS be refreshed or reloaded?
6) How will the quality and integrity of the data be monitored? Data cleaning will often required to deal with issues like missing values, data formats, code values, primary keys and referential integrity.
7) How will a log be maintained? A dispute may arise about the origin of some data. It is therefore necessary to be able to not only log which information came from where but also when the information was last updated.
8) How will recovery take place?
9) Would the extraction process only copy data from the source systems and not delete the original data?
10) How will the transformation be carried out? Transformation could be done in either source OLTP system, ODS or staging area.
11) How will data be copied from non-relational legacy systems that are still operational?

ETL FUNCTIONS

• The ETL process consists of
→ data extraction from source systems
→ data transformation which includes data cleaning and
→ data loading in the ODS or the data warehouse

• *Data cleaning* deals with detecting & removing errors/inconsistencies from the data, in particular the data that is sourced from a variety of computer systems.

• Building an integrated database from a number of source-systems may involve solving some of the following problems:

Instance Identity Problem

• The same customer may be represented slightly differently in different source-systems.
• There is a possibility of mismatching between the different systems that needs to be identified & corrected.
• Checking for homonyms & synonyms is necessary.
• Achieving very high consistency in names & addresses requires a huge amount of resources.

Data Errors

• Following are different types of data errors
→ data may have some missing attribute values
→ there may be duplicate records
→ there may be wrong aggregations
→ there may be inconsistent use of nulls, spaces and empty values
→ some attribute values may be inconsistent(i.e. outside their domain)
→ there may be non-unique identifiers

Record Linkage Problem

• This deals with problem of linking information from different databases that relates to the same customer.
• Record linkage can involve a large number of record comparisons to ensure linkages that have a high level of accuracy.

Semantic Integration Problem

• This deals with integration of information found in heterogeneous OLTP & legacy sources.
→ Some of the sources may be relational.
→ Some may be even in text documents
→ Some data may be character strings while others may be integers.

Data Integrity Problem

• This deals with issues like
→ referential integrity → null values → domain of values

• Data cleaning should be based on the following 5 steps:

1) Parsing

• This involves
→ identifying various components of the source data-files and
→ then establishing relationships between those and the fields in the target files.

2) Correcting

• Correcting the identified components is usually based on a variety of sophisticated techniques including mathematical algorithms.

3) Standardizing

• Business rules of enterprise may be used to transform the data to standard form.

4) Matching

• Much of the data extracted from a number of source-systems is likely to be related. Such data needs to be matched.

5) Consolidating

• All corrected, standardized and matched data can now be consolidated to build a single version of the enterprise-data.

DATA WAREHOUSE

• This is an integrated subject-oriented & time-variant repository of information in support of management's decision making process.

• In other words, it is a process of integrating enterprise-wide data, originating from a variety of sources, into a single repository.

• ODS is a current-valued data store
while a DW is a time-variant repository of data (Table: 7.6).

• Benefits of implementing a data warehouse:

1) To provide a single version of truth about enterprise information
2) To speed up adhoc reports and queries that involves aggregations across many attributes
3) To provide a system in which managers who do not have a strong technical background are able to run complex queries
4) To provide a database that stores relatively clean data
5) To provide a DB that stores historical data that may have been deleted from OLTP systems

• Q: Why a data warehouse must be separate from OLTP systems?

Ans: Complex queries involving number of aggregations are likely to degrade performance of systems.

• Smaller data warehouse is called *data marts* (Figure: 7.3).

• A data-mart stores information for a limited number of subject-areas.

• The data-mart approach is attractive since beginning with a single data-mart is relatively inexpensive and easier to implement.

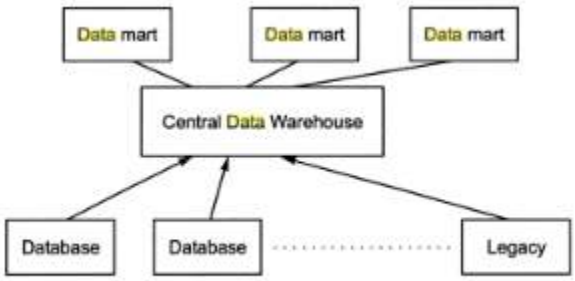


Figure 7.3 Simple structure of a data warehouse system.

Table 7.5 Comparing OLTP and data warehouse systems (Based on Oracle, 2002)

Property	*OLTP*	*Data warehouse*
Nature of the database	3NF	Multidimensional
Indexes	Few	Many
Joins	Many	Some
Duplicated data	Normalized data	Denormalized data
Derived data and aggregates	Rare	Common
Queries	Mostly predefined	Mostly ad hoc
Nature of queries	Mostly simple	Mostly complex
Updates	All the time	Not allowed, only refreshed
Historical data	Often not available	Essential

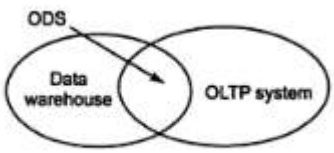


Figure 7.2 Relationship between OLTP, ODS and DW systems.

Table 7.6 Comparison of the ODS and data warehouse (Based on IBM, 2001)

ODS	*DW*
Data of high quality at detailed level and assured availability	Data may not be perfect, but sufficient for strategic analysis; data does not have to be highly available
Contains current and near-current data	Contains historical data
Real-time and near real-time data loads	Normally batch data loads
Mostly updated at data field level (even if it may be appended)	Data is appended, not updated
Typically detailed data only	Contains summarized and detailed data
Modelled to support rapid data updates (3NF)	Variety of modelling techniques used, typically multidimensional for data marts to optimize query performance
Transactions similar to those in OLTP systems	Complex queries processing larger volumes of data
Used for detailed decision making and operational reporting	Used for long-term decision making and management reporting
Used at the operational level	Used at the managerial level

ODS & DW ARCHITECTURE

- The architecture involves
  - → extracting information from source systems by using an ETL process and
  - → then storing the information in a staging area (Figure: 7.4).
- Staging area is a temporary database, separate from source systems and ODS where in the data is
  - → copied from source systems
  - →then transformed and
  - → finally copied to the ODS.
- The daily changes also come to the staging area.
- Another ETL process is used to transform information from the staging area to populate the ODS.
- The ODS could then be used for producing a variety of reports for management.
- The ODS is then used for supplying information via another ETL process to the data warehouse which in turn feeds a number of data marts that generate the reports required by management.

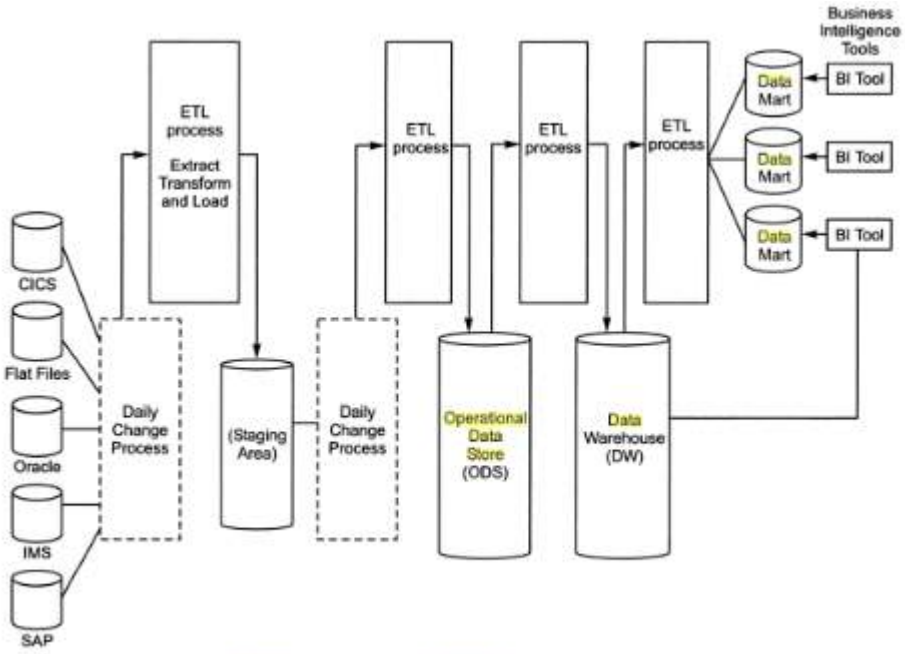


Figure 7.4 Another structure for ODS and DW.

DATA WAREHOUSE DESIGN

• A dimension (essentially an attribute) is an ordinate within a multidimensional structure consisting of a list of ordered values (called members).
• A member is a distinct value (or name or identifier) for the dimension.
• Values that depend on the dimensions are called measures.
• The fact table is a collection of related data items, consisting of values of dimensions of interest and the value of measure (Figure: 7.8).
• A data warehouse model consists of
→ a central fact table &
→ a set of surrounding dimensions tables on which the facts depend. Such a model is called a star scheme (Figure: 7.6).
• The characteristic of a star scheme is that all the dimensions directly link to the fact table.
• Star schemas may be refined into snowflake schemas if we wish to provide support for dimension hierarchies by allowing dimension tables to have subtables to represent the hierarchies (Fig: 7.7).
.

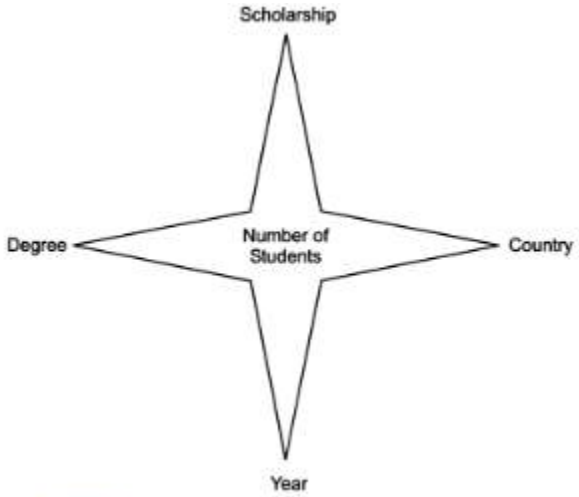


Figure 7.5 A simple example of a star schema.

Table 7.7 An example of the fact table

Year	*Degree name*	*Country name*	*Scholarship name*	*Number*
200301	BSc	Australia	Govt	35
199902	MBBS	Canada	None	50
200002	LLB	USA	ABC	22
199901	BCom	UK	Commonwealth	7
200102	LLB	Australia	Equity	2

Table 7.8 An example of the degree dimension table

Name	*Faculty*	*Scholarship eligibility*	*Number of semesters*
BSc	Science	Yes	6
MBBS	Medicine	No	10
LLB	Law	Yes	8
BCom	Business	No	6
BA	Arts	No	6

Table 7.9 An example of the country dimension table

Name	*Continent*	*Education level*	*Major religion*
Nepal	Asia	Low	Hinduism
Indonesia	Asia	Low	Islam
Norway	Europe	High	Christianity
Singapore	Asia	High	NULL
Colombia	South America	Low	Christianity

Table 7.10 An example of the scholarship dimension table

Name	*Amount (%)*	*Scholarship eligibility*	*Number*
Colombo	100	All	6
Equity	100	Low income	10
Asia	50	Top 5%	8
Merit	75	Top 5%	5
Bursary	25	Low income	12

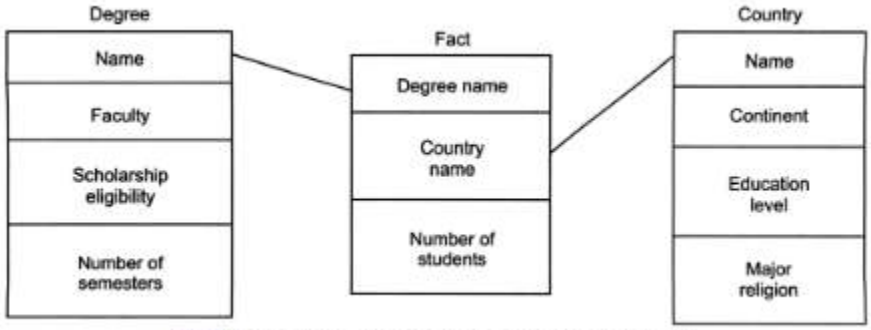


Figure 7.6 Star schema for a two-dimensional example.

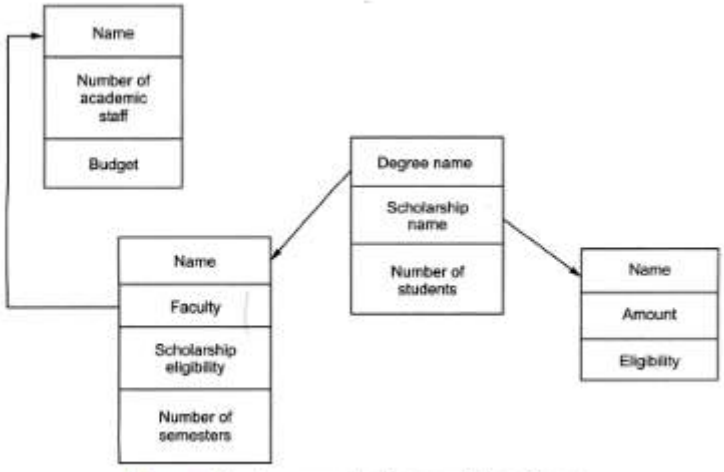


Figure 7.8 An example of a snowflake schema.

DW IMPLEMENTATION STEPS

1) Requirements Analysis & Capacity Planning

- The first step involves
  - → defining enterprise needs
  - → defining architecture
  - → carrying out capacity planning and
  - → selecting the hardware and software tools
- This step also involves consulting
  - → with senior management &
  - → with the various stakeholders

2) Hardware Integration

- Both hardware and software need to be put together by integrating
  - → servers
  - → storage devices &
  - → client software tools

3) Modeling

- This involves designing the warehouse schema and views. This may involve using a modeling tool if the data warehouse is complex.

4) Physical Modeling

- This involves designing
  - → data warehouse organization
  - → data placement
  - → data partitioning
  - → deciding on access methods and indexing

5) Sources

- This involves identifying and connecting the sources using gateways.

6) ETL

- This involves
  - →identifying a suitable ETL tool vendor and
  - →purchasing and implementing the tool
- This may include customizing the tool to suit the needs of the enterprise.

7) Populate DW

- This involves testing the required ETL tools, perhaps using a staging area. Then, ETL tools may be used in populating the warehouse.

8) User Applications

- This involves designing and implementing applications required by the end users.

9) Roll-out the DW and Applications

DW METADATA

- Metadata is data about data or documentation about the data that is needed by the users.
- This is not the actual data warehouse, but answers the "who, what, where, when, why and how" questions about the data warehouse.
- This describes the characteristics of a resource.
- Metadata may be classified into two groups: back room metadata and front room metadata.

1) Much important information is included in the *back room metadata* that is process related, for example, the ETL processes.
Furthermore, this could include
- → data extraction, cleaning and loading specifications
- → dates on which the ETL process was carried out and
- → a list of data that were loaded

2) The *front room metadata* is more descriptive and could include information needed by the users, for example,
- → user security privileges
- → various statistics about usage

DW IMPLEMENTATION GUIDELINES

Build Incrementally

- Firstly, a data mart may be built with one particular project in mind.
- Then, a number of other sections of enterprise may be implemented in a similar manner.
- An enterprise data warehouse can then be implemented in an iterative manner, allowing all data marts to extract information from the data warehouse.

Need a Champion

- The project must have a champion who is willing to carry out considerable research into expected costs and benefits of the project.
- The projects require inputs from many units in an enterprise and therefore need to be driven by someone who is capable of interacting with people in the enterprise.

Senior Management Support

- Given the resource intensive nature of such projects and the time they can take to implement, the project calls for a sustained commitment from senior management.

Ensure Quality

- Only data that has been cleaned and only data that are of a quality should be loaded in the data warehouse.

Corporate Strategy

- The project must fit with corporate strategy and business objectives.

Business Plan

- Financial costs, expected benefits & a project plan must be clearly outlined and understood by all stakeholders.

Training

- The users must be trained
  - → to use the warehouse and
  - → to understand its capabilities

Adaptability

- Project should build in adaptability so that changes may be made to DW if & when required.

Joint mManagement

- The project must be managed by both IT and business professionals in the enterprise.

EXERCISES

1) Explain ODS along with its structure. (6)
2) Explain ETL. List out the issues that need to be resolved for successful implementation of an ETL system. (6)
3) What is data cleaning? Explain the 5 steps in data cleaning. (4)
4) Explain the problems that need to be solved for building an integrated database from a number of source systems. (6)
5) Write a short note on
   i) Data warehouse (4)
   ii) ZLE (4)
6) Compare the following:
   i) Lazy query vs. eager query processing (4)
   ii) OLTP vs. DW (6)
   iii) ODS vs. DW (6)
7) Explain ODS & DW architecture along with its structure. (4)
8) What are star & snowflake schemas? (2)
9) Explain the steps for DW implementation. (6)
10) Explain the guidelines for DW implementation. (6)
11) Write a short note on DW metadata. (4)

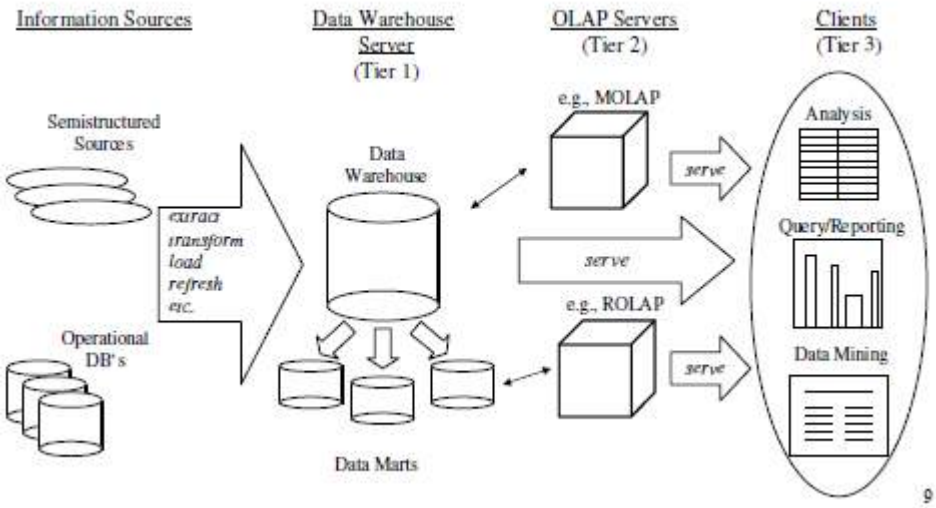
UNIT 2: ONLINE ANALYTICAL PROCESSING (OLAP)

INTRODUCTION

• A dimension is an attribute within a multidimensional model consisting of a list of values (called members).
• A fact is defined by a combination of dimension values for which a non-null value exists.
• The non-null values of facts are the numerical values stored in each data cube cell. They are called measures.
• A data cube computes aggregates over all subsets of dimensions specified in the cube.

OLAP

• OLAP stands for Online Transaction Processing Systems.
• This is primarily a software-technology concerned with fast analysis of enterprise-information.
• In other words, OLAP is the dynamic enterprise analysis required to create, manipulate, animate & synthesize information from exegetical, contemplative and formulaic data analysis models.
• Business-Intelligence(BI) is used to mean both data-warehousing and OLAP.
• In other words, BI is defined as a user-centered process of
→ exploring data, data-relationships and trends
→ thereby helping to improve overall decision-making.



MOTIVATIONS FOR USING OLAP

1) Understanding and Improving Sales

• For an enterprise that has many products and many channels for selling the products, OLAP can assist in finding the most popular products & the most popular channels.
• In some cases, it may be possible to find the most profitable customers.
• Analysis of business-data can assist in improving the enterprise-business.

2) Understanding and Reducing Costs of doing Business

• OLAP can assist in
→ analyzing the costs associated with sales &
→ controlling the costs as much as possible without affecting sales
• In some cases, it may also be possible to identify expenditures that produce a high ROI (return on investment).

CHARACTERISTICS OF OLAP SYSTEMS

1) Users

• OLTP systems are designed for office-workers (100-1000 users).

Whereas, OLAP systems are designed for decision-makers(few business-users).

2) Functions

• OLTP systems are mission-critical. They support an enterprise's day-to-day operations. They are mostly performance-driven.

Whereas, OLAP systems are management-critical. They support an enterprise's decision-functions using analytical-investigations.

3) Nature

• OLTP systems are designed to process one record at a time, for ex, a record related to the customer.

Whereas, OLAP systems involve queries that deal with many records at a time and provide aggregate data to a manager.

4) Design

• OLTP systems are designed to be application-oriented.

Whereas, OLAP systems are designed to be subject-oriented.

• OLTP systems view the enterprise-data as a collection of tables (based on ER model).

Whereas, OLAP systems view enterprise-information as multidimensional model.

5) Data

• OLTP systems normally deal only with the current-status of information. The old information may have been archived and may not be accessible online.

Whereas, OLAP systems require historical-data over several years.

6) Kind of use

• OLTP systems are used for read & write operations.

Whereas, OLAP systems normally do not update the data.

Table 8.2 Comparison of OLTP and OLAP systems

Property	*OLTP*	*OLAP*
Nature of users	Operations workers	Decision makers
Functions	Mission-critical	Management-critical
Nature of queries	Mostly simple	Mostly complex
Nature of usage	Mostly repetitive	Mostly ad hoc
Nature of design	Application oriented	Subject oriented
Number of users	Thousands	Dozens
Nature of data	Current, detailed, relational	Historical, summarized, multidimensional
Updates	All the time	Usually not allowed

FASMI CHARACTERISTICS OF OLAP SYSTEMS

Fast

- Most queries should be answered very quickly, perhaps within seconds.
- The performance of the system has to be like that of a search-engine.
- The data-structures must be efficient.
- The hardware must be powerful enough for
  - → amount of data &
  - → number of users
- One approach can be
  - → pre-compute the most commonly queried aggregates and
  - → compute the remaining aggregates on-the-fly

Analytic

- The system must provide rich analytic-functionality.
- Most queries should be answered without any programming.
- The system should be able to cope with any relevant queries for application & user.

Shared

- The system is
  - → likely to be accessed only by few business-analysts and
  - → may be used by thousands of users
- Being a shared system, the OLAP software should provide adequate security for confidentiality & integrity.
- Concurrency-control is obviously required if users are writing or updating data in the database.

Multidimensional

- This is the basic requirement.
- OLAP software must provide a multidimensional conceptual view of the data.
- A dimension often has hierarchies that show parent/child relationships between the members of dimensions. The multidimensional structure should allow such hierarchies.

Information

- The system should be able to handle a large amount of input-data.
- The capacity of system to handle information and its integration with the data warehouse may be critical.

CODD'S OLAP CHARACTERISTICS

Multidimensional Conceptual View

- This is the central characteristics.
- By requiring a multidimensional view, it is possible to carry out operations like slice and dice.

Accessibility (OLAP as a Mediator)

- The OLAP software should be sitting between data-sources (e.g. data warehouse) and an OLAP front-end.

Batch Extraction vs. Interpretive

- The system should provide multidimensional data staging plus partial pre-calculation of aggregates in large multidimensional databases.

Multi-user Support

- Being a shared system, the OLAP software should provide many normal database operations including retrieval, update, concurrency-control, integrity and security.

Storing results of OLAP

- OLAP results data should be kept separate from source-data.
- Read-write OLAP applications should not be implemented directly on live transaction-data if OLTP source systems are supplying information to the OLAP system directly.

Extraction of Missing Values

- The system should distinguish missing-values from zero-values. If a distinction is not made, the aggregates are likely to be computed incorrectly.

Uniform Reporting Performance

- Increasing the number of dimensions (or database-size) should not significantly degrade the reporting performance of the system.

Treatment of Missing Values

- The system should ignore all missing-values regardless of their source.

Generic Dimensionality

- The system should treat each dimension as equivalent in both its structure and operational capabilities.

Unlimited Dimensions and Aggregation Levels

- The system should allow unlimited dimensions and aggregation levels.
- In practice, number of dimensions is rarely more than 10 and number of hierarchies rarely more than 6.

MULTIDIMENSIONAL VIEW AND DATA CUBE

• The multidimensional view of data is in some ways a natural view of any enterprise for managers • The triangle shows that as we go higher in the triangle hierarchy the managers need for detailed information declines (Figure 8.1).

Figure 8.1 A typical university management hierarchy.

• The multidimensional view of data by using an example of a simple OLTP database consists of the three tables:

student(Student_id, Student_name, Country, DOB, Address)
enrolment(Student_id, Degree_id, SSemester)
degree(Degree_id, Degree_name, Degree_length, Fee, Department)

Table 8.3 The relation student

Student_id	*Student_name*	*Country*	*DOB*	*Address*
8656789	Peta Williams	Australia	1/1/1980	Davis Hall
8700020	John Smith	Canada	2/2/1981	9 Davis Hall
8900020	Arun Krishna	USA	3/3/1983	90 Second Hall
8801234	Peter Chew	UK	4/4/1983	88 Long Hall
8654321	Reena Rani	Australia	5/5/1984	88 Long Hall
8712374	Kathy Garcia	Malaysia	6/6/1980	88 Long Hall
8612345	Chris Watanabe	Singapore	7/7/1981	11 Main Street
8744223	Lars Anderssen	Sweden	8/8/1982	Null
8977665	Sachin Singh	UAE	9/9/1983	Null
9234567	Rahul Kumar	India	10/10/1984	Null
9176543	Saurav Gupta	UK	11/11/1985	1, Captain Drive

Table 8.4 The relation enrolment

Student_id	*Degree_id*	*SSemester*
8900020	1256	2002–01
8700074	3271	2002–01
8700074	3321	2002–02
8900020	4444	2000–01
8801234	1256	2001–01
8801234	3321	1999–02
8801234	3333	1999–02
8977665	3333	2000–02

Table 8.5 The relation degree

Degree_id	*Degree_name*	*Degree_length*	*Fee*	*Department*
1256	BIT	6	18	Computer Science
2345	BSc	6	20	Computer Science
4325	BSc	6	20	Chemistry
3271	BSc	6	20	Physics
3321	BCom	6	16	Business
4444	MBBS	12	30	Medicine
3333	LLB	8	22	Law

• It is clear that the information given in Tables 8.3, 8.4 and 8.5, although suitable for a student enrolment OLTP system, is not suitable for efficient management decision making.
• The managers do not need information about the individual students, the degree they are enrolled in, and the semester they joined the university.
• What the managers needs the trends in student numbers in different degree programs and from different countries.
• We first consider only two dimensions. Let us say we are primarily interested in finding out how many students from each country came to do a particular degree (Table: 8.6). Therefore we may visualize the data as two-dimensional, i.e., *Country x Degree*

Table 8.6 A two-dimensional table of aggregates for semester 2000–01

Country \ *Degree*	BSc	LLB	MBBS	BCom	BIT	ALL
Australia	5	20	15	50	11	101
India	10	0	15	25	17	67
Malaysia	5	1	10	12	23	51
Singapore	2	2	10	10	31	55
Sweden	5	0	5	25	7	42
UK	5	15	20	20	13	73
USA	0	2	20	15	19	56
ALL	32	40	95	157	121	445

• Using this two-dimensional view we are able to find the number of students joining any degree from any country (only for semester 2000-01). Other queries that we are quickly able to answer are:
→How many students started BIT in 2000-01?
→How many students joined from Singapore in 2000-01?

Table 8.7 Two-dimensional table of aggregates for semester 2001–01

Country \ *Degree*	BSc	LLB	MBBS	BCom	BIT	ALL
Australia	7	10	16	53	10	96
India	9	0	17	22	13	61
Malaysia	5	1	19	19	20	64
Singapore	2	2	10	12	23	49
Sweden	8	0	5	16	7	36
UK	4	13	20	26	11	74
USA	4	2	10	10	12	38
ALL	39	28	97	158	96	418

Table 8.8 Two-dimensional table of aggregates for both semesters

Country \ Degree	BSc	LLB	MBBS	BCom	BIT	ALL
Australia	12	30	31	103	21	197
India	19	0	32	47	30	128
Malaysia	10	2	29	31	43	115
Singapore	4	4	20	22	54	104
Sweden	13	0	10	41	14	78
UK	9	28	40	46	24	147
USA	4	4	30	25	31	94
ALL	71	68	192	315	217	863

• Tables 8.6, 8.7 and 8.8 together now form a three-dimensional cube. The table 8.7 provides totals for the two semesters and we are able to "drill-down" to find numbers in individual semesters.
• Putting the three tables together gives a cube of 8 x 6 x 3 (= 144) cells including the totals along every dimension.
• A cube could be represented by:

Country x Degree x Semester

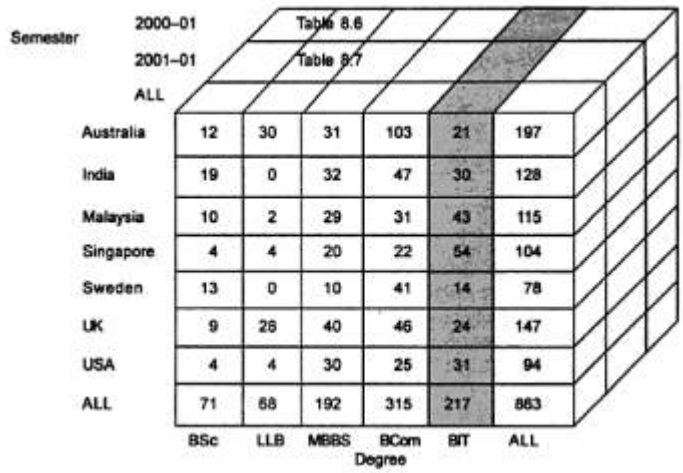


Figure 8.2: cube formed by tables 8.6, 8.7 & 8.8

• In the three-dimensional cube, the following eight types of aggregations or queries are possible:

1. null (e.g. how many students are there? Only 1 possible query)
2. degrees (e.g. how many students are doing BSc? 5 possible queries if we assume 5 different degrees)
3. semester (e.g. how many students entered in semester 2000-01? 2 possible queries if we only have data about 2 semesters)
4. country (e.g. how many students are from the USA? 7 possible queries if there are 7 countries)
5. degrees, semester (e.g. how many students entered in 2000-01 to enroll in BCom? With 5 degrees and 2 different semesters 10 queries)
6. semester, country (e.g. how many students from the UK entered in 2000-01? With 7 countries and 2 different semesters 14 queries)
7. degrees, country (e.g. how many students from Singapore are enrolled in BCom? 7*5=35 queries)
8. all (e.g. how many students from Malaysia entered in 2000-01 to enroll in BCom? 7*5*2=70 queries)

• All the cell in the cube represents measures or aggregations.
• 2^n types of aggregations are possible for n dimensions.

DATA CUBE IMPLEMENTATION

Pre-compute and store all

- Millions of aggregates need to be computed and stored.
- This is the best solution as far as query response-time is concerned.
- This solution is impractical for a large data-cube,
  since resources required to compute & store the aggregates will be prohibitively large
- Indexing large amounts of data is also expensive.

Pre-compute(and store) none

- The aggregates are computed on-the-fly using the raw data whenever a query is posed.
- This approach does not require additional space for storing the cube.
- The query response-time is likely to be very poor for large data-cubes.

Pre-compute and store some

- We pre-compute and store the most frequently queried aggregates and compute others as the need arises.
- We may also be able to derive some of the remaining aggregates using the aggregates that have already been computed (Figure 8.3). .
- The more aggregates we are able to pre-compute, the better the query performance
- Data-cube products use different techniques for pre-computing aggregates and storing them. They are generally based on one of two implementation models.
  i) ROLAP (relational OLAP)
  ii) MOLAP (multidimensional OLAP)

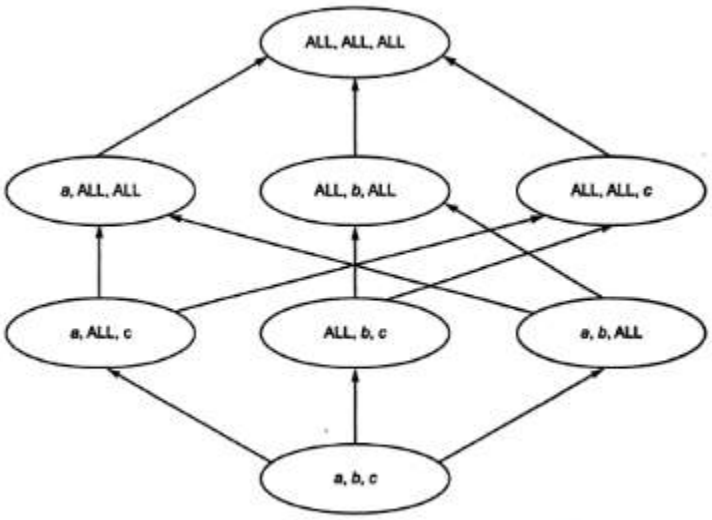


Figure 8.3 Relationships between aggregations of a three-dimensional cube.

ROLAP

• This uses relational or extended-relational DBMS to store and manage warehouse data.
• This may be considered a bottom-up approach to OLAP.
• This is typically based on using a data warehouse that has been designed using a star scheme.
• The data warehouse provides the multidimensional capabilities by representing data in fact-table and dimension-tables.
• The fact-table contains one column for each dimension and one column for each measure and every row of the table provides one fact.
• A fact then is represented as with the last column as 30. An OLAP tool is then provided to manipulate the data in these data warehouse tables.
• This tool essentially groups the fact-table to find aggregates and uses some of the aggregates already computed to find new aggregates.
• Advantages:
→ This is more easily used with existing relational DBMS and
→ The data can be stored efficiently using tables.
→ Greater scalability
• Disadvantage:
→ Poor query performance
• Some products in this category are Oracle OLAP mode and OLAP Discoverer.

MOLAP

• This is based on using a multidimensional DBMS rather than a data-warehouse to store and access data.
• This may be considered as a top-down approach to OLAP.
• This does not have a standard approach to storing and maintaining their data.
• This often uses special-purpose file systems or indexes that store pre-computation of all aggregations in the cube.
• Advantages:
→ Implementation is usually exceptionally efficient
→ Easier to use and therefore may be more suitable for inexperienced users
→ Fast indexing to pre-computed summarized-data
• Disadvantages:
→ More expensive than ROLAP
→ Data is not always current
→ Difficult to scale a MOLAP system for very large OLAP problems
→ Storage utilization may be low if the data-set is sparse
• Some MOLAP products are Hyperion Essbase and Applix iTM1.

Table 8.10 Comparison of MOLAP and ROLAP

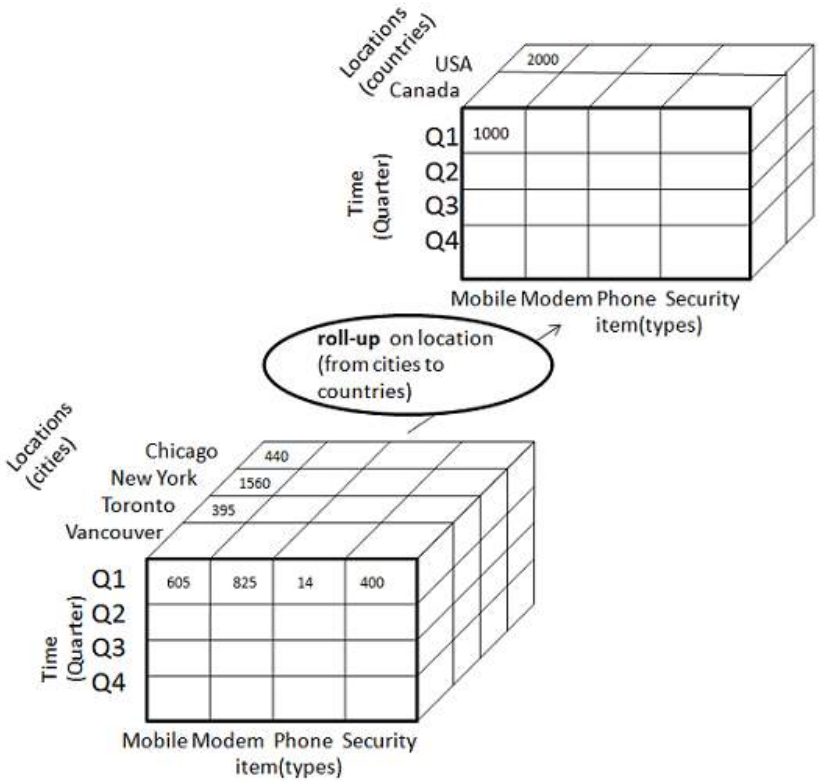
Property	*MOLAP*	*ROLAP*
Data structure	Multidimensional database using sparse arrays	Relational tables (each cell is a row)
Disk space	Separate database for data cube; large for large data cubes	May not require any space other than that available in the data warehouse
Retrieval	Fast (pre-computed)	Slow (computes on-the-fly)
Scalability	Limited (cubes can be very large)	Excellent
Best suited for	Inexperienced users, limited set of queries	Experienced users, queries change frequently
DBMS facilities	Usually weak	Usually very strong

DATA CUBE OPERATIONS

• A number of operations may be applied to data cubes. The common ones are:
→ roll-up
→ drill-down
→ pivot or rotate
→ slice & dice

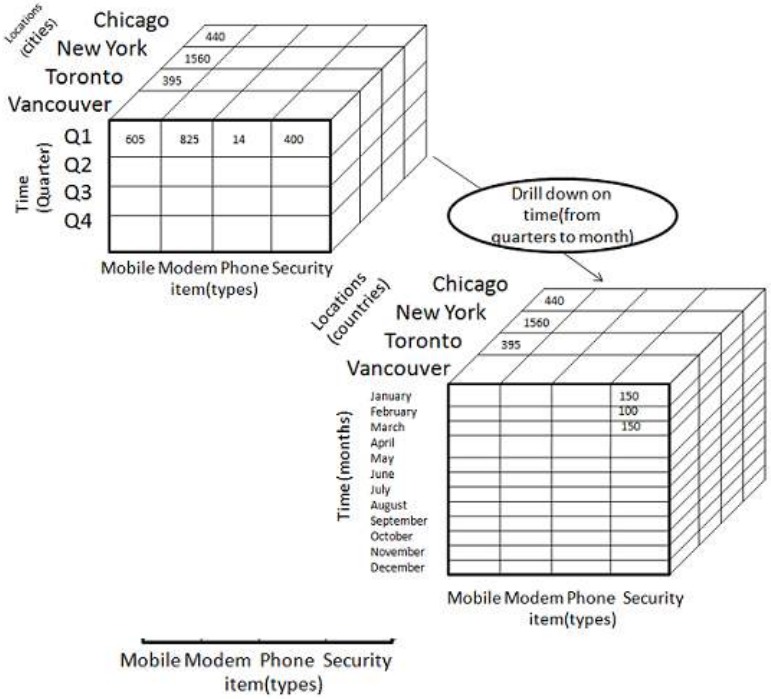
ROLL-UP

• This is like zooming out on the data cube.
• This is required when the user needs further abstraction or less detail.
• 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 level of country.
• The data is grouped into cities rather than countries.



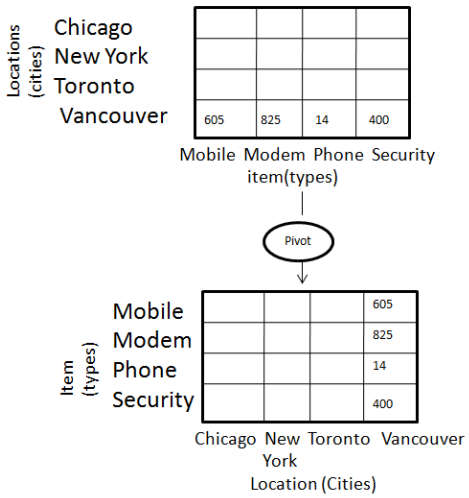
DRILL DOWN

• This is like zooming in on the data and is therefore the reverse of roll-up.
• This is an appropriate operation when the user needs further details or when the user wants to partition more finely or wants to focus on some particular values of certain dimensions.
• This adds more details to the data.
• Initially the concept hierarchy was "day < month < quarter < year."
• On drill-up the time dimension is descended from the level quarter to the level of month.
• When drill-down operation is performed then one or more dimensions from the data cube are added.



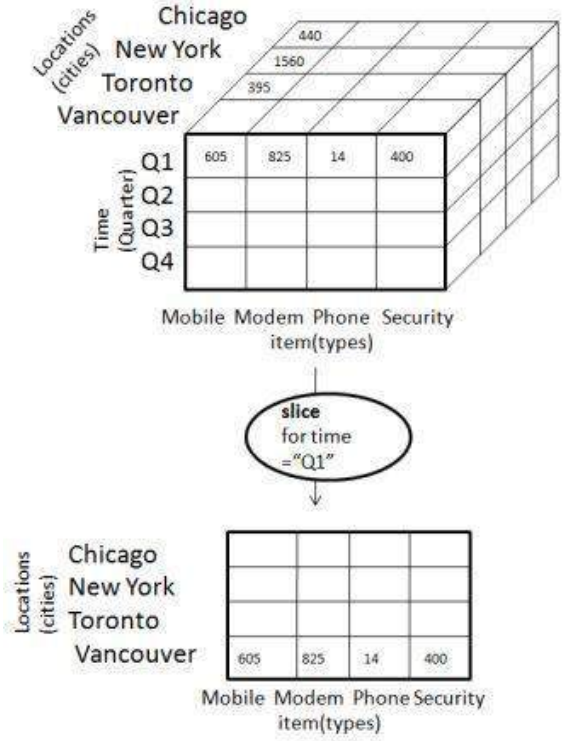
PIVOT OR ROTATE

- This is used when the user wishes to re-orient the view of the data cube.
- This may involve
  - → swapping the rows and columns, or
  - → moving one of the row dimensions into the column dimension
- In this, the item and location axes in 2-D slice are rotated.

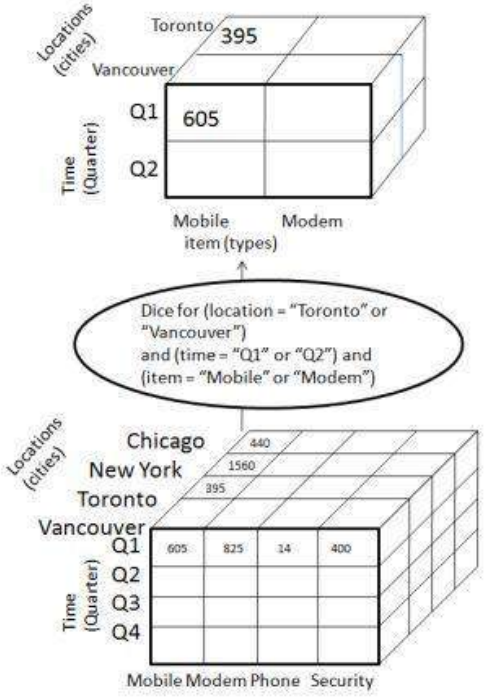


SLICE & DICE

• These are operations for browsing the data in the cube. The terms refer to the ability to look at information from different viewpoints.
• A slice is a subset of cube corresponding to a single value for 1 or more members of dimensions.
• The slice operation is performed for the dimension time using the criterion time ="Q1".



• The dice operation is similar to slice but dicing does not involve reducing number of dimensions.
• A dice is obtained by performing a selection on two or more dimensions.
• The dice operation on cube based on the following selection criteria that involve three dimensions.
(location = "Toronto" or "Vancouver")
(time = "Q1" or "Q2")
(item =" Mobile" or "Modem").



GUIDELINES FOR OLAP IMPLEMENTATION

Vision

- The OLAP team must, in consultation with the users, develop a clear vision for the OLAP system.
- This vision (including the business-objectives) should be clearly defined, understood and shared by the stakeholders.

Senior Management Support

- The project should be fully supported by the senior-managers.
- Since a data warehouse may have been developed already, this should not be difficult.

Selecting an OLAP Tool

- The team should familiarize themselves with the ROLAP and MOLAP tools available in the market.
- Since tools are quite different, careful planning may be required in selecting a tool that is appropriate for the enterprise.

Corporate Strategy

- The OLAP-strategy should fit with the enterprise-strategy and business-objectives. A good fit will result in the OLAP tools being used more widely.

Focus on the Users

- The project should be focused on the users.
- Users should, in consultation with the technical professionals, decide what tasks will be done first and what will be done later.
- A good GUI should be provided to non-technical users.
- The project can only be successful with the full support of the users.

Joint Management

- The project must be managed by both the IT and business professionals.
- Many other people should be involved in supplying ideas.

Review and Adapt

- Regular reviews of the project may be required to ensure that the project is meeting the current needs of the enterprise.

OLAP SOFTWARE

- SQL Server 2000 Analysis Service from Microsoft. SQL Server 2000 analysis services is the OLAP services component in SQL Server 7.0.
- BI2M(Business Intelligence to Marketing and Management) from B&M Service has 3 modules, one of which is for OLAP. The OLAP module allows database exploring including slice and dice, roll-up, drill-down and displays results as 2D Charts, 3D charts and tables.
- Business Objects OLAP Intelligence from BusinessObjects allows access to OLAP servers from Microsoft, Hyperion, IBM and SAP. BusinessObjects also has widely used Crystal Analysis and Reports.
- Usual operations like slice and dice, and drill directly on multidimensional sources are possible.
- ContourCube from Contour Components is an OLAP product that enables users to slice and dice, roll up, drill down and pivot efficiently.
- DB2 Cube Views from IBM includes features and functions for managing and deploying multidimensional data.
- Express and the Oracle OLAP option. Express is a multidimensional database and application development environment for building OLAP applications.

EXERCISES

1) What is OLAP? Explain the motivations for using OLAP. (4)
2) Compare the following:
   i) OLTP vs. OLAP (6) ii) ROLAP vs. MOLAP (4)
3) Explain the FASMI characteristics of OLAP systems. (6)
4) Explain the Codd's OLAP characteristics. (6)
5) Explain the 3 methods for data cube implementation. (6)
6) Explain various operations on data cube. (6)
7) Explain the guidelines for OLAP implementation. (6)

UNIT 3: DATA MINING

WHAT IS DATA MINING?

• Data Mining is the process of automatically discovering useful information in large data-repositories.

• DM techniques

→ can be used to search large DB to find useful patterns that might otherwise remain unknown

→ provide capabilities to predict the outcome of future observations

Why do we need Data Mining?

• Conventional database systems provide users with query & reporting tools.

• To some extent the query & reporting tools can assist in answering questions like, where did the largest number of students come from last year?

• But these tools cannot provide any intelligence about why it happened.

Taking an Example of University Database System

• The OLTP system will quickly be able to answer the query like "how many students are enrolled in university"

• The OLAP system using data warehouse will be able to show the trends in students' enrollments (ex: how many students are preferring BCA),

• Data mining will be able to answer where the university should market.

DATA MINING AND KNOWLEDGE DISCOVERY

• Data Mining is an integral part of KDD (Knowledge Discovery in Databases).

KDD is the overall process of converting raw data into useful information (Figure: 1.1).

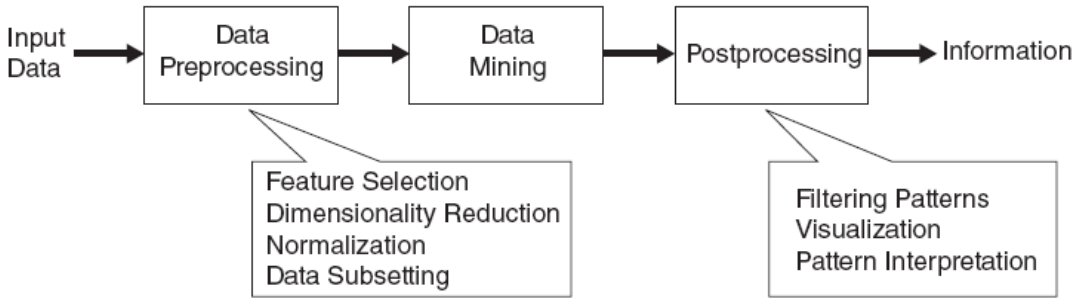


Figure 1.1. The process of knowledge discovery in databases (KDD).

• The input-data is stored in various formats such as flat files, spread sheet or relational tables.

• Purpose of preprocessing: to transform the raw input-data into an appropriate format for subsequent analysis.

• The steps involved in data-preprocessing include

→ combine data from multiple sources

→ clean data to remove noise & duplicate observations, and

→ select records & features that are relevant to the DM task at hand

• Data-preprocessing is perhaps the most time-consuming step in the overall knowledge discovery process.

• "Closing the loop" refers to the process of integrating DM results into decision support systems.

• Such integration requires a postprocessing step. This step ensures that only valid and useful results are incorporated into the decision support system.

• An example of postprocessing is visualization.

Visualization can be used to explore data and DM results from a variety of viewpoints.

• Statistical measures can also be applied during postprocessing to eliminate bogus DM results.

MOTIVATING CHALLENGES

Scalability

• Nowadays, data-sets with sizes of terabytes or even petabytes are becoming common.
• DM algorithms must be scalable in order to handle these massive data sets.
• Scalability may also require the implementation of novel data structures to access individual records in an efficient manner.
• Scalability can also be improved by developing parallel & distributed algorithms.

High Dimensionality

• Traditional data-analysis technique can only deal with low dimensional data.
• Nowadays, data-sets with hundreds or thousands of attributes are becoming common.
• Data-sets with temporal or spatial components also tend to have high dimensionality.
• The computational complexity increases rapidly as the dimensionality increases.

Heterogeneous and Complex Data

• Traditional analysis methods can deal with homogeneous type of attributes.
• Recent years have also seen the emergence of more complex data-objects.
• DM techniques for complex objects should take into consideration relationships in the data, such as
→ temporal & spatial autocorrelation
→ parent-child relationships between the elements in semi-structured text & XML documents

Data Ownership & Distribution

• Sometimes, the data is geographically distributed among resources belonging to multiple entities.
• Key challenges include:
1) How to reduce amount of communication needed to perform the distributed computation
2) How to effectively consolidate the DM results obtained from multiple sources &
3) How to address data-security issues

Non Traditional Analysis

• The traditional statistical approach is based on a hypothesized and test paradigm.
In other words, a hypothesis is proposed, an experiment is designed to gather the data, and then the data is analyzed with respect to hypothesis.
• Current data analysis tasks often require the generation and evaluation of thousands of hypotheses, and consequently, the development of some DM techniques has been motivated by the desire to automate the process of hypothesis generation and evaluation.

THE ORIGIN OF DATA MINING

• Data mining draws upon ideas from
→ Sampling, estimation, and hypothesis test from statistics
→ Search algorithms, modeling techniques machine learning, learning theories from AI pattern recognition, statistics database systems
• Traditional techniques may be unsuitable due to
→Enormity of data →High dimensionality of data →Heterogeneous nature of data
• Data mining also had been quickly to adopt ideas from other areas including
→Optimization →Evolutionary computing →Signal processing →Information theory
• Database systems are needed to provide supports for efficient storage, indexing, query processing.
• The parallel computing and distribute technology are two major data addressing issues in data mining to increase the performance.

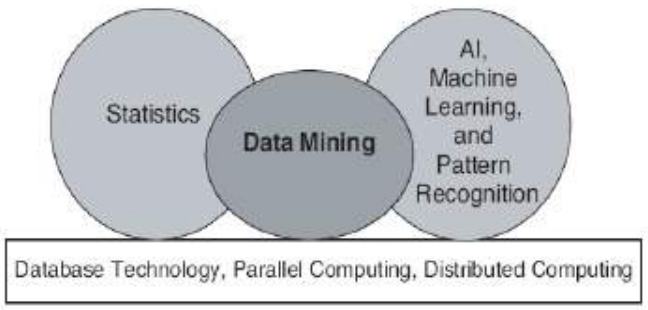


Figure 1.2. Data mining as a confluence of many disciplines.

DATA MINING TASKS

• DM tasks are generally divided into 2 major categories.

Predictive Tasks

• The objective is to predict the value of a particular attribute based on the values of other attributes.

• The attribute to be predicted is commonly known as the target or dependent variable, while the attributes used for making the predication are known as the explanatory or independent variables.

Descriptive Tasks

• The objective is to derive patterns (correlations, trends, clusters, trajectories and anomalies) that summarize the relationships in data.

• Descriptive DM tasks are often exploratory in nature and frequently require postprocessing techniques to validate and explain the results.

Four of the Core Data Mining Tasks

1) Predictive Modeling

• This refers to the task of *building a model for the target variable* as a function of the explanatory variable.

• The goal is to learn a model that minimizes the error between the predicted and true values of the target variable.

• There are 2 types of predictive modeling tasks:

i) Classification: used for discrete target variables

Ex: Web user will make purchase at an online bookstore is a classification task, because the target variable is binary valued.

ii) Regression: used for continuous target variables.

Ex: forecasting the future price of a stock is regression task because price is a continuous values attribute

2) Cluster Analysis

• This seeks to find groups of closely related observations so that observations that belong to the same cluster are more similar to each other than observations that belong to other clusters.

• Clustering has been used

→ to group sets of related customers

→ to find areas of the ocean that have a significant impact on the Earth's climate

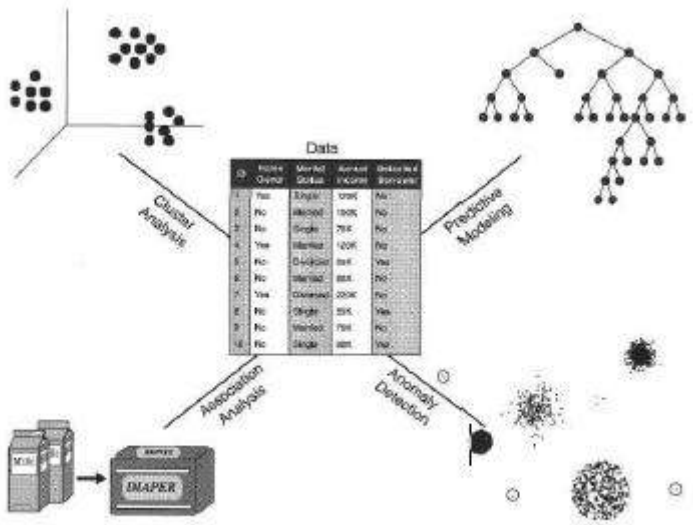


Figure 1.3. Four of the core data mining tasks.

3) Association Analysis

• This is used to discover patterns that describe strongly associated features in the data.
• The goal is to extract the most interesting patterns in an efficient manner.
• Useful applications include
→ finding groups of genes that have related functionality or
→ identifying web pages that are accessed together
• Ex: market based analysis

We may discover the rule that {diapers} -> {Milk}, which suggests that customers who buy diapers also tend to buy milk.

4) Anomaly Detection

• This is the task of identifying observations whose characteristics are significantly different from the rest of the data. Such observations are known as anomalies or outliers.
• The goal is
→ to discover the real anomalies and
→ to avoid falsely labeling normal objects as anomalous.
• Applications include the detection of fraud, network intrusions, and unusual patterns of disease.

Example 1.4 (Credit Card Fraud Detection).

• A credit card company records the transactions made by every credit card holder, along with personal information such as credit limit, age, annual income, and address.
• Since the number of fraudulent cases is relatively small compared to the number of legitimate transactions, anomaly detection techniques can be applied to build a profile of legitimate transactions for the users.
• When a new transaction arrives, it is compared against the profile of the user.
• If the characteristics of the transaction are very different from the previously created profile, then the transaction is flagged as potentially fraudulent

EXERCISES

1. What is data mining? Explain Data Mining and Knowledge Discovery? (10)
2. What are different challenges that motivated the development of DM? (10)
3. Explain Origins of data mining (5)
4. Discuss the tasks of data mining with suitable examples. (10)
5. Explain Anomaly Detection .Give an Example? (5)
6. Explain Descriptive tasks in detail? (10)
7. Explain Predictive tasks in detail by example? (10)

UNIT 3: DATA MINING (CONT.)

WHAT IS A DATA OBJECT?

• A data-set refers to a collection of data-objects and their attributes.

• Other names for a data-object are record, transaction, vector, event, entity, sample or observation.

• Data-objects are described by a number of attributes such as

→ mass of a physical object or

→ time at which an event occurred.

• Other names for an attribute are dimension, variable, field, feature or characteristics.

WHAT IS AN ATTRIBUTE?

• An attribute is a characteristic of an object that may vary, either

→ from one object to another or

→ from one time to another.

• For example, eye color varies from person to person.

Eye color is a symbolic attribute with a small no. of possible values {brown, black, blue, green}.

Example 2.2 (Student Information).

• Often, a data-set is a file, in which the objects are records(or rows) in the file and each field (or column) corresponds to an attribute.

• For example, Table 2.1 shows a data-set that consists of student information.

• Each row corresponds to a student and each column is an attribute that describes some aspect of a student, such as grade point average(GPA) or identification number(ID).

Table 2.1. A sample data set containing student information.

Student ID	Year	Grade Point Average (GPA)	...
	⋮		
1034262	Senior	3.24	...
1052663	Sophomore	3.51	...
1082246	Freshman	3.62	...
	⋮		

PROPERTIES OF ATTRIBUTE VALUES

• The type of an attribute depends on which of the following properties it possesses:

1) Distinctness: = ≠

2) Order: < >

3) Addition: + -

4) Multiplication: * /

• Nominal attribute: Uses only distinctness.

Examples: ID numbers, eye color, pin codes

• Ordinal attribute: Uses distinctness & order.

Examples: Grades in {SC, FC, FCD}

Shirt sizes in {S, M, L, XL}

• Interval attribute: Uses distinctness, order & addition

Examples: calendar dates, temperatures in Celsius or Fahrenheit.

• Ratio attribute: Uses all 4 properties

Examples: temperature in Kelvin, length, time, counts

DIFFERENT TYPES OF ATTRIBUTES

Table 2.2. Different attribute types.

Attribute Type		Description	Examples	Operations
Categorical (Qualitative)	Nominal	The values of a nominal attribute are just different names; i.e., nominal values provide only enough information to distinguish one object from another. $(=, \neq)$	zip codes, employee ID numbers, eye color, gender	mode, entropy, contingency correlation, χ^2 test
	Ordinal	The values of an ordinal attribute provide enough information to order objects. $(<, >)$	hardness of minerals, $\{good, better, best\}$, grades, street numbers	median, percentiles, rank correlation, run tests, sign tests
Numeric (Quantitative)	Interval	For interval attributes, the differences between values are meaningful, i.e., a unit of measurement exists. $(+, -)$	calendar dates, temperature in Celsius or Fahrenheit	mean, standard deviation, Pearson's correlation, t and F tests
	Ratio	For ratio variables, both differences and ratios are meaningful. $(*, /)$	temperature in Kelvin, monetary quantities, counts, age, mass, length, electrical current	geometric mean, harmonic mean, percent variation

Table 2.3. Transformations that define attribute levels.

Attribute Type		Transformation	Comment
Categorical (Qualitative)	Nominal	Any one-to-one mapping, e.g., a permutation of values	If all employee ID numbers are reassigned, it will not make any difference.
	Ordinal	An order-preserving change of values, i.e., $new\_value = f(old\_value)$, where f is a monotonic function.	An attribute encompassing the notion of good, better, best can be represented equally well by the values $\{1, 2, 3\}$ or by $\{0.5, 1, 10\}$.
Numeric (Quantitative)	Interval	$new\_value = a * old\_value + b$, a and b constants.	The Fahrenheit and Celsius temperature scales differ in the location of their zero value and the size of a degree (unit).
	Ratio	$new\_value = a * old\_value$	Length can be measured in meters or feet.

DESCRIBING ATTRIBUTES BY THE NUMBER OF VALUES

1) Discrete

- Has only a finite or countably infinite set of values.
- Examples: pin codes, ID Numbers, or the set of words in a collection of documents.
- Often represented as integer variables.
- Binary attributes are a special case of discrete attributes and assume only 2 values. E.g. true/false, yes/no, male/female or 0/1

2) Continuous

- Has real numbers as attribute values.
- Examples: temperature, height, or weight.
- Often represented as floating-point variables.

ASYMMETRIC ATTRIBUTES

- Binary attributes where only non-zero values are important are called asymmetric binary attributes.
- Consider a data-set where each object is a student and each attribute records whether or not a student took a particular course at a university.
- For a specific student, an attribute has a value of 1 if the student took the course associated with that attribute and a value of 0 otherwise.
- Because students take only a small fraction of all available courses, most of the values in such a data-set would be 0.
- Therefore, it is more meaningful and more efficient to focus on the non-zero values.
- This type of attribute is particularly important for association analysis.

TYPES OF DATA SETS

1) Record data
   → Transaction (or Market based data)
   → Data matrix
   → Document data or Sparse data matrix
2) Graph data
   → Data with relationship among objects (World Wide Web)
   → Data with objects that are Graphs (Molecular Structures)
3) Ordered data
   → Sequential data (Temporal data)
   → Sequence data
   → Time series data
   → Spatial data

GENERAL CHARACTERISTICS OF DATA SETS

• Following 3 characteristics apply to many data-sets:

1) Dimensionality

• Dimensionality of a data-set is no. of attributes that the objects in the data-set possess.
• Data with a small number of dimensions tends to be qualitatively different than moderate or high-dimensional data.
• The difficulties associated with analyzing high-dimensional data are sometimes referred to as the curse of dimensionality.
• Because of this, an important motivation in preprocessing data is dimensionality reduction.

2) Sparsity

• For some data-sets with asymmetric feature, most attribute of an object have values of 0.
• In practical terms, sparsity is an advantage because usually only the non-zero values need to be stored & manipulated.
• This results in significant savings with respect to computation-time and storage.
• Some DM algorithms work well only for sparse data.

3) Resolution

• This is frequently possible to obtain data at different levels of resolution, and often the properties of the data are different at different resolutions.
• Ex: the surface of the earth seems very uneven at a resolution of few meters, but is relatively smooth at a resolution of tens of kilometers.
• The patterns in the data also depend on the level of resolution.
• If the resolution is too fine, a pattern may not be visible or may be buried in noise.
  If the resolution is too coarse, the pattern may disappear.

RECORD DATA

- Data-set is a collection of records.
  Each record consists of a fixed set of attributes.
- Every record has the same set of attributes.
- There is no explicit relationship among records or attributes.
- The data is usually stored either
  → in flat files or → in relational databases

Tid	Refund	Marital Status	Taxable Income	Defaulted Borrower
1	Yes	Single	125K	No
2	No	Married	100K	No
3	No	Single	70K	No
4	Yes	Married	120K	No
5	No	Divorced	95K	Yes
6	No	Married	60K	No
7	Yes	Divorced	220K	No
8	No	Single	85K	Yes
9	No	Married	75K	No
10	No	Single	90K	Yes

(a) Record data.

TID	ITEMS
1	Bread, Soda, Milk
2	Beer, Bread
3	Beer, Soda, Diaper, Milk
4	Beer, Bread, Diaper, Milk
5	Soda, Diaper, Milk

(b) Transaction data.

Projection of x Load	Projection of y Load	Distance	Load	Thickness
10.23	5.27	15.22	27	1.2
12.65	6.25	16.22	22	1.1
13.54	7.23	17.34	23	1.2
14.27	8.43	18.45	25	0.9

(c) Data matrix.

	team	coach	play	ball	score	game	win	lost	timeout	season
Document 1	3	0	5	0	2	6	0	2	0	2
Document 2	0	7	0	2	1	0	0	3	0	0
Document 3	0	1	0	0	1	2	2	0	3	0

(d) Document-term matrix.

Figure 2.2. Different variations of record data.

TYPES OF RECORD DATA

1) Transaction (Market Basket Data)

- Each transaction consists of a set of items.
- Consider a grocery store.
  The set of products purchased by a customer represents a transaction while the individual products represent items.
- This type of data is called market basket data because the items in each transaction are the products in a person's "market basket."
- Data can also be viewed as a set of records whose fields are asymmetric attributes.

2) Data Matrix

- An m*n matrix, where there are m rows, one for each object, & n columns, one for each attribute.
  This matrix is called a data-matrix.
- Since data-matrix consists of numeric attributes, standard matrix operation can be applied to manipulate the data.

3) Sparse Data Matrix

- This is a special case of a data-matrix.
- The attributes are of the same type and are asymmetric i.e. only non-zero values are important.

Document Data

- A document can be represented as a 'vector',
  where each term is a attribute of the vector and value of each attribute is the no. of times corresponding term occurs in the document.

GRAPH BASED DATA

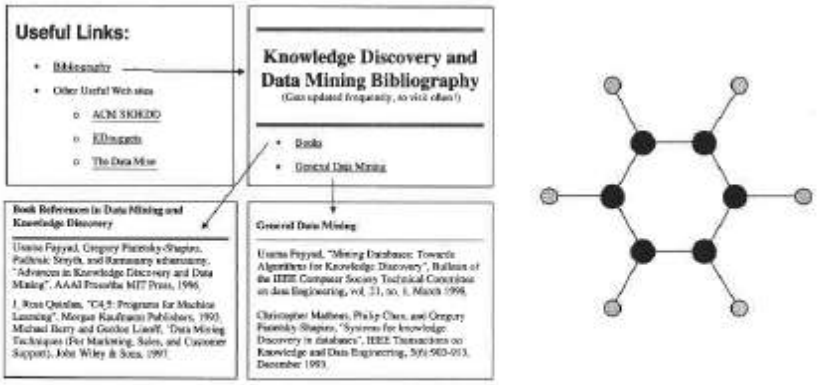
- Sometimes, a graph can be a convenient and powerful representation for data.
- We consider 2 specific cases:

1) Data with Relationships among Objects

- The relationships among objects frequently convey important information.
- In particular, the data-objects are mapped to nodes of the graph, while relationships among objects are captured by link properties such as direction & weight.
- For ex, in web, the links to & from each page provide a great deal of information about the relevance of a web-page to a query, and thus, must also be taken into consideration.

2) Data with Objects that are Graphs

- If the objects contain sub-objects that have relationships, then such objects are frequently represented as graphs.
- For ex, the structure of chemical compounds can be represented by a graph, where nodes are atoms and links between nodes are chemical bonds.



(a) Linked Web pages. (b) Benzene molecule.

Figure 2.3. Different variations of graph data.

ORDERED DATA

Sequential Data (Temporal Data)

- This can be thought of as an extension of record-data, where each record has a time associated with it.
- A time can also be associated with each attribute.
- For example, each record could be the purchase history of a customer, with a listing of items purchased at different times.
- Using this information, it is possible to find patterns such as "people who buy DVD players tend to buy DVDs in the period immediately following the purchase."

Sequence Data

- This consists of a data-set that is a sequence of individual entities, such as a sequence of words or letters.
- This is quite similar to sequential data, except that there are no time stamps; instead, there are positions in an ordered sequence.
- For example, the genetic information of plants and animals can be represented in the form of sequences of nucleotides that are known as genes.

Time Series Data

- This is a special type of sequential data in which a series of measurements are taken over time.
- For example, a financial data-set might contain objects that are time series of the daily prices of various stocks.
- An important aspect of temporal-data is temporal-autocorrelation i.e. if two measurements are close in time, then the values of those measurements are often very similar.

Spatial Data

- Some objects have spatial attributes, such as positions or areas.
- An example is weather-data (temperature, pressure) that is collected for a variety of geographical location.
- An important aspect of spatial-data is spatial-autocorrelation i.e. objects that are physically close tend to be similar in other ways as well.

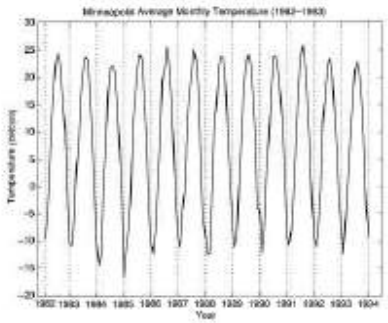
Time	Customer	Items Purchased
t1	C1	A, B
t2	C3	A, C
t2	C1	C, D
t3	C2	A, D
t4	C2	E
t5	C1	A, E

Customer	Time and Items Purchased
C1	(t1: A,B) (t2:C,D) (t5:A,E)
C2	(t3: A, D) (t4: E)
C3	(t2: A, C)

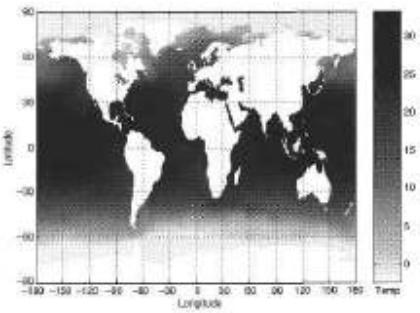
(a) Sequential transaction data.

```
GGTTCCGCCTTCAGCCCCGCGCC
CGCAGGGCCCGCCCCGCGCCGTC
GAGAAGGGCCCGCCTGGCGGGCG
GGGGGAGGCGGGGCCGCCCGAGC
CCAACCGAGTCCGACCAGGTGCC
CCCTCTGCTCGGCCTAGACCTGA
GCTCATTAGGCGGCAGCGGACAG
GCCAAGTAGAACACGCGAAGCGC
TGGGCTGCCTGCTGCGACCAGGG
```

(b) Genomic sequence data.



(c) Temperature time series.



(d) Spatial temperature data.

Figure 2.4. Different variations of ordered data.

DATA PREPROCESSING

• Data preprocessing is a broad area and consists of a number of different strategies and techniques that are interrelated in complex way.

• Different data processing techniques are:

1. Aggregation
2. Sampling
3. Dimensionality reduction
4. Feature subset selection
5. Feature creation
6. Discretization and binarization
7. Variable transformation

AGGREGATION

• This refers to combining 2 or more attributes (or objects) into a single attribute (or object).
For example, merging daily sales figures to obtain monthly sales figures

• Motivations for aggregation:

1) Data reduction: The smaller data-sets require
→ less memory → less processing time.
Because of aggregation, more expensive algorithm can be used.

2) Aggregation can act as a *change of scale* by providing a high-level view of the data instead of a low-level view. E.g. Cities aggregated into districts, states, countries, etc

3) The behavior of groups of objects is often *more stable* than that of individual objects.

• Disadvantage: The potential loss of interesting details.

SAMPLING

• This is a method used for selecting a subset of the data-objects to be analyzed.

• This is often used for both
→ preliminary investigation of the data → final data analysis

• Q: Why sampling?
Ans: Obtaining & processing the entire set of "data of interest" is too expensive or time consuming.

• Sampling can reduce data-size to the point where better & more expensive algorithm can be used.

• Key principle for effective sampling: Using a sample will work almost as well as using entire data-set, if the *sample is representative*.

Sampling Methods

1) Simple Random Sampling

• There is an equal probability of selecting any particular object.

• There are 2 variations on random sampling:

i) Sampling without Replacement

• As each object is selected, it is removed from the population.

ii) Sampling with Replacement

• Objects are not removed from the population as they are selected for the sample.

• The same object can be picked up more than once.

• When the population consists of different types(or number) of objects, simple random sampling can fail to adequately represent those types of objects that are less frequent.

2) Stratified Sampling

• This starts with pre-specified groups of objects.

• In the simplest version, equal numbers of objects are drawn from each group even though the groups are of different sizes.

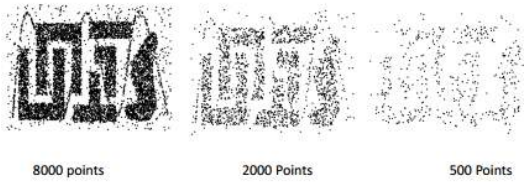
• In another variation, the number of objects drawn from each group is proportional to the size of that group.

3) Progressive Sampling

• If proper sample-size is difficult to determine then progressive sampling can be used.

• This method starts with a small sample, and
then increases the sample-size until a sample of sufficient size has been obtained.

• This method requires a way to evaluate the sample to judge if it is large enough.



DIMENSIONALITY REDUCTION

• Key benefit: many DM algorithms work better if the dimensionality is lower.

Purpose

- May help to eliminate irrelevant features or reduce noise.
- Can lead to a more understandable model (which can be easily visualized).
- Reduce amount of time and memory required by DM algorithms.
- Avoid curse of dimensionality.

The Curse of Dimensionality

- Data-analysis becomes significantly harder as the dimensionality of the data increases.
- For classification, this can mean that there are not enough data-objects to allow the creation of a model that reliably assigns a class to all possible objects.
- For clustering, the definitions of density and the distance between points (which are critical for clustering) become less meaningful.
- As a result, we get
  → reduced classification accuracy &
  → poor quality clusters.

FEATURE SUBSET SELECTION

• Another way to reduce the dimensionality is to use only a subset of the features.

• This might seem that such approach would lose information, this is not the case if redundant and irrelevant features are present.

1) *Redundant features* duplicate much or all of the information contained in one or more other attributes.
For example: purchase price of a product and the amount of sales tax paid.
2) *Irrelevant features* contain almost no useful information for the DM task at hand.
For example: students' ID numbers are irrelevant to the task of predicting students' grade point averages.

Techniques for Feature Selection

1) Embedded approaches: Feature selection occurs naturally as part of DM algorithm. Specifically, during the operation of the DM algorithm, the algorithm itself decides which attributes to use and which to ignore.
2) Filter approaches: Features are selected before the DM algorithm is run.
3) Wrapper approaches: Use DM algorithm as a black box to find best subset of attributes.

An Architecture for Feature Subset Selection

• The feature selection process is viewed as consisting of 4 parts:

1) A measure of evaluating a subset,
2) A search strategy that controls the generation of a new subset of features,
3) A stopping criterion and
4) A validation procedure.

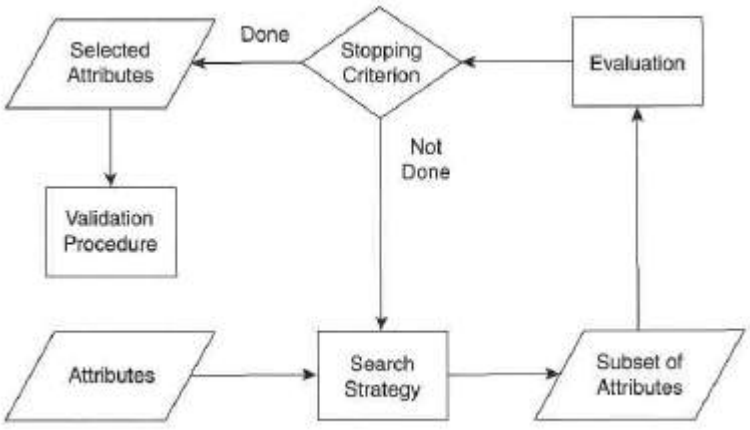


Figure 2.11. Flowchart of a feature subset selection process.

DISCRETIZATION AND BINARIZATION

• Some DM algorithms (especially classification algorithms) require that the data be in the form of categorical attributes.
• Algorithms that find association patterns require that the data be in the form of binary attributes.
• Transforming continuous attributes into a categorical attribute is called *discretization.*
And transforming continuous & discrete attributes into binary attributes is called as *binarization*.

Binarization

• A simple technique to binarize a categorical attribute is the following: If there are m categorical values, then uniquely assign each original value to an integer in interval [0,m-1].
• Next, convert each of these m integers to a binary number.
• Since n=[$\log_2$(m)] binary digits are required to represent these integers, represent these binary numbers using 'n' binary attributes.

Table 2.5. Conversion of a categorical attribute to three binary attributes.

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

Table 2.6. Conversion of a categorical attribute to five asymmetric binary attributes.

Categorical Value	Integer Value	x_1	x_2	x_3	x_4	x_5
awful	0	1	0	0	0	0
poor	1	0	1	0	0	0
OK	2	0	0	1	0	0
good	3	0	0	0	1	0
great	4	0	0	0	0	1

Discretizction of continuous attributes

• Discretization is typically applied to attributes that are used in classification or association analysis.
• In general, the best discretization depends on
→ the algorithm being used, as well as
→ the other attributes being considered
• Transformation of a continuous attribute to a categorical attribute involves two subtasks:
→ deciding how many categories to have and
→ determining how to map the values of the continuous attribute to these categories

VARIABLE TRANSFORMATION

• This refers to a transformation that is applied to all the values of a variable.
• Ex: converting a floating point value to an absolute value.
• Two types are:

1) Simple Functions

• A simple mathematical function is applied to each value individually.
• If x is a variable, then examples of transformations include e^x, 1/x, log(x), sin(x).

2) Normalization (or Standardization)

• The goal is to make an entire set of values have a particular property.
• A traditional example is that of "standardizing a variable" in statistics.
• If $\bar{x}$ is the mean of the attribute values and s_x is their standard deviation, then the transformation x'=(x-$\bar{x}$)/s_x creates a new variable that has a mean of 0 and a standard deviation of 1.

MEASURE OF SIMILARITY AND DISSIMILARITY

- Similarity & dissimilarity are important because
  they are used by a no. of DM techniques such as clustering, classification & anomaly detection.
- *Proximity* is used to refer to either similarity or dissimilarity.
- The *similarity* between 2 objects is a numerical measure of degree to which the 2 objects are alike.
- Consequently, similarities are higher for pairs of objects that are more alike.
- Similarities are usually non-negative and are often between 0(no similarity) and 1(complete similarity).
- The *dissimilarity* between 2 objects is a numerical measure of the degree to which the 2 objects are different.
- Dissimilarities are lower for more similar pairs of objects.
- The term distance is used as a synonym for dissimilarity.
- Dissimilarities sometimes fall in the interval [0,1] but is also common for them to range from 0 to infinity.

Table 2.7. Similarity and dissimilarity for simple attributes

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

DISSIMILARITIES BETWEEN DATA OBJECTS

Distances

• The Euclidean distance, d, between 2 points, x and y, in one-,two-,three- or higher-dimensional space, is given by the following familiar formula:

$$d(\mathbf{x},\mathbf{y}) = \sqrt{\sum_{k=1}^{n}(x_k - y_k)^2}, \qquad (2.1)$$

where n=number of dimensions

x_k and y_k are respectively the k^{th} attributes of x and y.

• The Euclidean distance measure given in equation 2.1 is generalized by the Minkowski distance metric given by

$$d(\mathbf{x},\mathbf{y}) = \left(\sum_{k=1}^{n}|x_k - y_k|^r\right)^{1/r}, \qquad (2.2)$$

where r=parameter.

• The following are the three most common examples of minkowski distance:

r=1. City block(Manhattan L_1 norm) distance.
A common example is the Hamming distance, which is the number of bits that are different between two objects that have only binary attributes ie between two binary vectors.

r=2. Euclidean distance (L_2 norm)

r=∞. Supremum(L_∞ or L_{max} norm) distance. This is the maximum difference between any attribute of the objects. Distance is defined by

$$d(\mathbf{x},\mathbf{y}) = \lim_{r\to\infty}\left(\sum_{k=1}^{n}|x_k - y_k|^r\right)^{1/r}. \qquad (2.3)$$

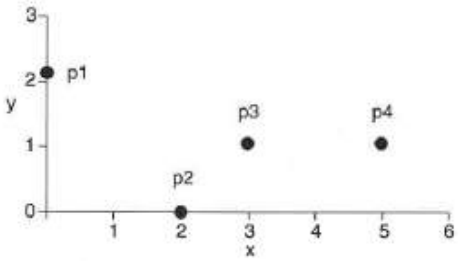


Figure 2.15. Four two-dimensional points.

Table 2.8. x and y coordinates of four points.

point	x coordinate	y coordinate
p1	0	2
p2	2	0
p3	3	1
p4	5	1

Table 2.9. Euclidean distance matrix for Table 2.8.

	p1	p2	p3	p4
p1	0.0	2.8	3.2	5.1
p2	2.8	0.0	1.4	3.2
p3	3.2	1.4	0.0	2.0
p4	5.1	3.2	2.0	0.0

Table 2.10. L_1 distance matrix for Table 2.8.

L_1	p1	p2	p3	p4
p1	0.0	4.0	4.0	6.0
p2	4.0	0.0	2.0	4.0
p3	4.0	2.0	0.0	2.0
p4	6.0	4.0	2.0	0.0

Table 2.11. L_∞ distance matrix for Table 2.8.

L_∞	p1	p2	p3	p4
p1	0.0	2.0	3.0	5.0
p2	2.0	0.0	1.0	3.0
p3	3.0	1.0	0.0	2.0
p4	5.0	3.0	2.0	0.0

• If d(x,y) is the distance between two points, x and y, then the following properties hold

1) Positivity
   d(x,x)>=0 for all x and y d(x,y)=0 only if x=y
2) Symmetry
   d(x,y)=d(y,x) for all x and y.
3) Triangle inequality
   d(x,z)<=d(x,y)+d(y,z) for all points x,y and z.

• Measures that satisfy all three properties are known as *metrics.*

SIMILARITIES BETWEEN DATA OBJECTS

• For similarities, the triangle inequality typically does not hold, but symmetry positivity typically do.
• If s(x,y) is the similarity between points x and y, then the typical properties of similarities are the following

1) s(x,y)=1 only if x=y
2) s(x,y)=s(y,x) for all x and y. (Symmetry)

• For ex, cosine and Jaccard similarity.

EXAMPLES OF PROXIMITY MEASURES

• Similarity measures between objects that contain only binary attributes are called similarity coefficients.
• Typically, they have values between 0 and 1.
• A value of 1 indicates that the two objects are completely similar,
while a value of 0 indicates that the objects are not at all similar.
• Let x and y be 2 objects that consist of n binary attributes.
• Comparison of 2 objects, ie, 2 binary vectors, leads to the following four quantities (frequencies):

f_{00}=the number of attributes where x is 0 and y is 0.
f_{01}=the number of attributes where x is 0 and y is 1.
f_{10}=the number of attributes where x is 1 and y is 0.
f_{11}=the number of attributes where x is 1 and y is 1.

SIMPLE MATCHING COEFFICIENT

• One commonly used similarity coefficient is the SMC, which is defined as

$$SMC = \frac{\text{number of matching attribute values}}{\text{number of attributes}} = \frac{f_{11} + f_{00}}{f_{01} + f_{10} + f_{11} + f_{00}}. \quad (2.5)$$

• This measure counts both presences and absences equally.

JACCARD COEFFICIENT

• Jaccard coefficient is frequently used to handle objects consisting of asymmetric binary attributes.
• The jaccard coefficient is given by the following equation:

$$J = \frac{\text{number of matching presences}}{\text{number of attributes not involved in 00 matches}} = \frac{f_{11}}{f_{01} + f_{10} + f_{11}}. \quad (2.6)$$

Example 2.17 (The SMC and Jaccard Similarity Coefficients). To illustrate the difference between these two similarity measures, we calculate SMC and J for the following two binary vectors.

$\mathbf{x} = (1, 0, 0, 0, 0, 0, 0, 0, 0, 0)$
$\mathbf{y} = (0, 0, 0, 0, 0, 0, 1, 0, 0, 1)$

$f_{01} = 2$ the number of attributes where $\mathbf{x}$ was 0 and $\mathbf{y}$ was 1
$f_{10} = 1$ the number of attributes where $\mathbf{x}$ was 1 and $\mathbf{y}$ was 0
$f_{00} = 7$ the number of attributes where $\mathbf{x}$ was 0 and $\mathbf{y}$ was 0
$f_{11} = 0$ the number of attributes where $\mathbf{x}$ was 1 and $\mathbf{y}$ was 1

$SMC = \frac{f_{11}+f_{00}}{f_{01}+f_{10}+f_{11}+f_{00}} = \frac{0+7}{2+1+0+7} = 0.7$

$J = \frac{f_{11}}{f_{01}+f_{10}+f_{11}} = \frac{0}{2+1+0} = 0$ ■

COSINE SIMILARITY

• Documents are often represented as vectors, where each attribute represents the frequency with which a particular term (or word) occurs in the document.
• This is more complicated, since certain common words are ignored and various processing techniques are used to account for
→ different forms of the same word
→ differing document lengths and
→ different word frequencies.
• The cosine similarity is one of the most common measure of document similarity.
• If x and y are two document vectors, then

$$\cos(\mathbf{x}, \mathbf{y}) = \frac{\mathbf{x} \cdot \mathbf{y}}{\|\mathbf{x}\| \, \|\mathbf{y}\|}, \tag{2.7}$$

where $\cdot$ indicates the vector dot product, $\mathbf{x} \cdot \mathbf{y} = \sum_{k=1}^{n} x_k y_k$, and $\|\mathbf{x}\|$ is the length of vector $\mathbf{x}$, $\|\mathbf{x}\| = \sqrt{\sum_{k=1}^{n} x_k^2} = \sqrt{\mathbf{x} \cdot \mathbf{x}}$.

Example 2.18 (Cosine Similarity of Two Document Vectors). This example calculates the cosine similarity for the following two data objects, which might represent document vectors:

$\mathbf{x} = (3, 2, 0, 5, 0, 0, 0, 2, 0, 0)$
$\mathbf{y} = (1, 0, 0, 0, 0, 0, 0, 1, 0, 2)$

$\mathbf{x} \cdot \mathbf{y} = 3*1 + 2*0 + 0*0 + 5*0 + 0*0 + 0*0 + 0*0 + 2*1 + 0*0 + 0*2 = 5$
$\|\mathbf{x}\| = \sqrt{3*3 + 2*2 + 0*0 + 5*5 + 0*0 + 0*0 + 0*0 + 2*2 + 0*0 + 0*0} = 6.48$
$\|\mathbf{y}\| = \sqrt{1*1 + 0*0 + 0*0 + 0*0 + 0*0 + 0*0 + 0*0 + 1*1 + 0*0 + 2*2} = 2.24$
$\cos(\mathbf{x}, \mathbf{y}) = 0.31$

• As indicates by figure 2.16, cosine similarity really is a measure of the angle between x and y.
• Thus, if the cosine similarity is 1,the angle between x and y is 0',and x and y are the same except for magnitude(length).
• If cosine similarity is 0, then the angle between x and y is 90' and they do not share any terms.

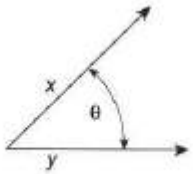


Figure 2.16. Geometric illustration of the cosine measure.

EXTENDED JACCARD COEFFICIENT(TANIMOTO COEFFICIENT)

• This can be used for document data.
• this coefficient is defined by following equation

$$EJ(\mathbf{x}, \mathbf{y}) = \frac{\mathbf{x} \cdot \mathbf{y}}{\|\mathbf{x}\|^2 + \|\mathbf{y}\|^2 - \mathbf{x} \cdot \mathbf{y}}. \tag{2.9}$$

ISSUES IN PROXIMITY CALCULATION

1) How to handle the case in which attributes have different scales and/or are correlated.
2) How to calculate proximity between objects that are composed of different types of attributes e.g. quantitative and qualitative.
3) How to handle proximity calculation when attributes have different weights.

COMBINING SIMILARITIES FOR HETEROGENEOUS ATTRIBUTES

• A general approach is needed when the attributes are of different types.
• One straightforward approach is to compute the similarity between each attribute separately and then combine these similarities using a method that results in a similarity between 0 and 1.
• Typically, the overall similarity is defined as the average of all the individual attribute similarities.

DATA MINING APPLICATIONS

Prediction & Description

- Data mining may be used to answer questions like
  - → "would this customer buy a product" or
  - → "is this customer likely to leave?"
- DM techniques may also be used for sales forecasting and analysis.

Relationship Marketing

- Customers have a lifetime value, not just the value of a single sale.
- Data mining can help
  - → in analyzing customer profiles and improving direct marketing plans
  - → in identifying critical issues that determine client loyalty and
  - → in improving customer retention

Customer Profiling

- This is the process of using the relevant and available information
  - → to describe the characteristics of a group of customers
  - → to identify their discriminators from ordinary consumers and
  - → to identify drivers for their purchasing decisions
- This can help an enterprise identify its most valuable customers so that the enterprise may differentiate their needs and values.

Outliers Identification & Detecting Fraud

- For this, examples include:
  - → identifying unusual expense claims by staff
  - → identifying anomalies in expenditure between similar units of an enterprise
  - → identifying fraud involving credit cards

Customer Segmentation

- This is a way to assess & view individuals in market based on their status & needs.
- Data mining may be used
  - → to understand & predict customer behavior and profitability
  - → to develop new products & services and
  - → to effectively market new offerings

Web site Design & Promotion

- Web mining may be used to discover how users navigate a web site and the results can help in improving the site design.
- Web mining may also be used in cross-selling by suggesting to a web customer, items that he may be interested in.

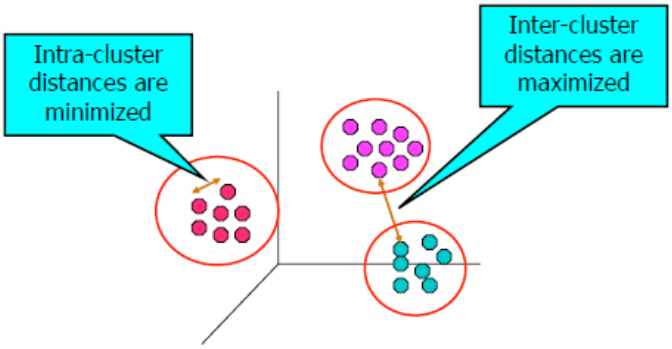
EXERCISES

1. Explain Data set. Give an Example? (5)
2. Explain 4 types of attributes by giving appropriate example? (10)
3. With example, explain
   i) Continious ii) Discrete iii) Asymmetric Attributes. (10)
4. Explain general characteristics of data sets. (6)
5. Explain record data & its types. (10)
6. Explain graph based data. (6)
7. Explain ordered data & its types. (10)
8. Explain shortly any five data pre-processing approaches. (10)
9. What is sampling? Explain simple random sampling vs. stratified sampling vs. progressive sampling. (10)
10. Write a short note on the following: (15)
    i)Dimensionality reduction ii)Variable transformation iii)Feature selection
11. Distinguish between
    i) SMC & Jaccard coefficient ii) Discretization & binarization (6)
12. Explain various distances used for measuring dissimilarity between objects. (6)
13. Consider the following 2 binary vectors
    X=(1,0,0,0,0,0,0,0,0,0) Y=(0,0,0,0,0,0,1,0,0,1)
    Find i)hamming distance ii)SMC iii)Jaccard coefficient (4)
14. List out issues in proximity calculation. (4)
15. List any 5 applications of data mining. (8)

UNIT 7: CLUSTERING TECHNIQUES

WHAT IS CLUSTER ANALYSIS?

- A cluster is a collection of data-objects
  - → similar to one another within the same group &
  - → dissimilar to the objects in other groups
- Cluster analysis is process of grouping a set of data-objects into clusters.



APPLICATIONS OF CLUSTER ANALYSIS

- In university, one may wish to find
  - → clusters of students or
  - → clusters of courses
- In medicine, one may be interested in finding
  - → clusters of patients or
  - → clusters of diseases
- In business, one may want to identify
  - → clusters of customers
  - → clusters of products
- Practical applications include
  - → character/pattern recognition
  - → web document classification
  - → image processing

CLASSIFICATION VS. CLUSTER ANALYSIS

• Classification is used mostly as a supervised learning method
while clustering is used as unsupervised learning.

Classification

• The classes are predefined (Table 4.1).
• The user already knows what classes there are.
• Some training data that is already labeled by their class-membership is available to build a model.
• The classification-problem then is to build a model that would be able to classify newly encountered data.

Table 4.1 An example of classification training data

Owns Home?	*Married*	*Gender*	*Employed*	*Credit Rating*	*Risk Class*
Yes	Yes	Male	Yes	*A*	*B*
No	No	Female	Yes	*A*	*A*
Yes	Yes	Female	Yes	*B*	*C*
Yes	No	Male	No	*B*	*B*
No	Yes	Female	Yes	*B*	*C*
No	No	Female	Yes	*B*	*A*
No	No	Male	No	*B*	*B*
Yes	No	Female	Yes	*A*	*A*
No	Yes	Female	Yes	*A*	*C*
Yes	Yes	Female	Yes	*A*	*C*

Cluster Analysis

• One does not know what classes or clusters exist (Table 4.2).
• The problem to be solved is to group the given data into meaningful clusters.
• The aim of cluster analysis is to find meaningful groups with
→ small within-group variations &
→ large between-group variation
• Most of the algorithms developed are based on some concept of similarity or distance.
• Drawbacks:
→ This process may be prohibitively expensive for large sets.
→ Cost of computing distances between groups of objects grows as no. of attributes grows.
→ Computing distances between categorical attributes is more difficult (compared to computing distances between objects with numeric attributes)

Table 4.2 An example without training data

Owns Home?	*Married*	*Gender*	*Employed*	*Credit Rating*
Yes	Yes	Male	Yes	*A*
No	No	Female	Yes	*A*
Yes	Yes	Female	Yes	*B*
Yes	No	Male	No	*B*
No	Yes	Female	Yes	*B*
No	No	Female	Yes	*B*
No	No	Male	No	*B*
Yes	No	Female	Yes	*A*
No	Yes	Female	Yes	*A*
Yes	Yes	Female	Yes	*A*

DESIRED FEATURES OF CLUSTER ANALYSIS METHOD

Scalability

- Data-mining problems can be large and therefore a cluster-analysis method should be able to deal with large problems gracefully.
- Ideally, performance should be linear with data-size.
- The method should also scale well to datasets in which number of attributes is large.

Only one Scan of the Dataset

- For large problems, data must be stored on disk, so cost of I/O disk can become significant in solving the problem.
- Therefore, the method should not require more than one scan of disk-resident data.

Ability to Stop & Resume

- For large dataset, cluster-analysis may require huge processor-time to complete the task.
- In such cases, task should be able to be stopped & then resumed when convenient.

Minimal Input Parameters

- The method should not expect too much guidance from the data-mining analyst.
- Therefore, the analyst should not be expected
  → to have domain knowledge of data and
  → to posses' insight into clusters that might exist in the data

Robustness

- Most data obtained from a variety of sources has errors.
- Therefore, the method should be able to deal with noise, outlier & missing values gracefully.

Ability to Discover Different Cluster-Shapes

- Clusters appear in different shapes and not all clusters are spherical.
- Therefore, method should be able to discover cluster-shapes other than spherical.

Different Data Types

- Many problems have a mixture of data types, for e.g. numerical, categorical & even textual.
- Therefore, the method should be able to deal with
  → numerical data → boolean data → categorical data

Result Independent of Data Input Order

- Therefore, the method should not be sensitive to data input-order.
- Irrespective of input-order, result of cluster-analysis of the same data should be same.

TYPES OF DATA

Numerical Data

- Examples include weight, marks, height, price, salary, and count.
- There are a number of methods for computing similarity between these data.
  E.g. Euclidean distance, manhattan distance.

Binary Data

- Examples include gender, marital status.
- A simple method involves
  → counting how many attribute values of 2 objects are different amongst n attributes &
  → using this as an indication of distance

Qualitative Nominal Data

- This is similar to binary data which may take more than 2 values but has no natural order. Examples include religion, foods or colors.

Qualitative Ranked Data

- This is similar to qualitative nominal data except that data has an order associated with it.
- Examples include: 1) grades A, B, C, and D 2) sizes S, M, L and XL
- One method of computing distance involves transferring the values to numeric values according to their rank. For example, grades A, B, C, D could be transformed to 4.0, 3.0, 2.0 and 1.0.

COMPUTING DISTANCE

- Most cluster-analysis methods are based on measuring similarity between objects.
- Distances are normally used to measure the similarity or dissimilarity between 2 objects.
- Let the distance between 2 points x and y be D(x,y).
- Distance has following simple properties:
    1) Distance is always positive. i.e. D(x,y) >= 0
    2) Distance from point x to itself is always 0. i.e. D(x,y) = 0
    3) Distance from point x to point y is always less than the sum of the distance from x to some other point z and distance from z to y. i.e. D(x,y) <= D(x,z) + D(z,y)
    4) Distance from x to y is always the same as from y to x. i.e. D(x,y) = D(y,x)
- Following are some of the distance measures:
    1) Euclidean distance (L_2 norm of difference vector)
    2) Manhattan distance (L_1 norm of the difference vector)
    3) Chebychev distance (L_∞ norm of the difference vector)
    4) Categorical data distance

Euclidean Distance

- This is most commonly used to compute distances and has an intuitive appeal.
- The largest valued attribute may dominate the distance.
- Requirement: The attributes should be properly scaled.
- This metric is more appropriate when the data is not standardized.

$$D(x, y) = (\Sigma(x_i - y_i)^2)^{1/2}$$

Manhattan Distance

- In most cases, the result obtained by this measure is similar to those obtained by using the Euclidean distance.
- The largest valued attribute may dominate the distance.

$$D(x, y) = \sum_i |x_i - y_i|$$

Chebychev Distance

- This metric is based on the maximum attribute difference.

$$D(x, y) = \text{Max}|x_i - y_i|$$

Categorical Data Distance

- This metric may be used if many attributes have categorical values with only a small number of values (e.g. metric binary values).
- Let N=total number of categorical attributes.

$$D(x, y) = (\text{number of } x_i - y_i)/N$$

TYPES OF CLUSTER ANALYSIS METHODS

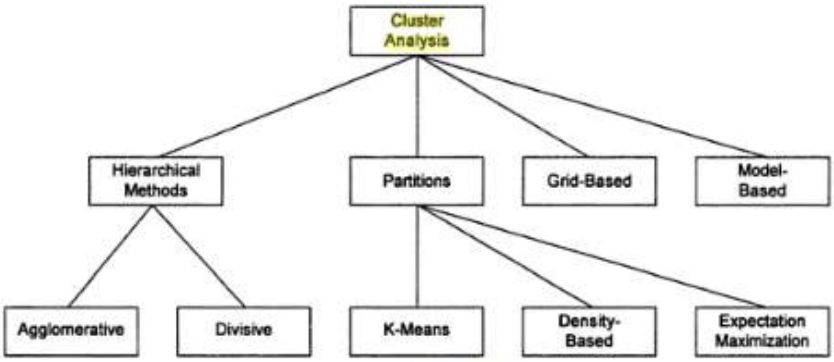


Figure 4.1 Taxonomy of cluster analysis methods.

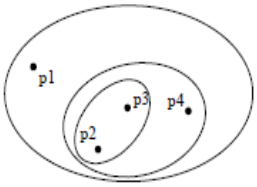
Partitional Method

- The objects are divided into non-overlapping clusters (or partitions) such that each object is in exactly one cluster (Figure 4.1a).
- The method obtains a single-level partition of objects.
- The analyst has
  → to specify number of clusters(k) prior → to specify starting seeds of clusters
- The analyst may have to use iterative approach in which he has to run the method many times
  → specifying different numbers of clusters & different starting seeds
  → then selecting the best solution
- The method converges to a local minimum rather than the global minimum.

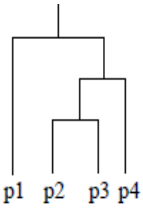
Original Points

A Partitional Clustering

Figure 4.1a



Traditional Hierarchical Clustering



Traditional Dendrogram

Figure 4.1b

Hierarchical Methods

- A set of nested clusters is organized as a hierarchical tree (Figure 4.1b).
- The method either
  → starts with one cluster & then splits into smaller clusters (called divisive or top down) or
  → starts with each object in an individual cluster & then tries to merge similar clusters into larger clusters(called agglomerative or bottom up)
- Tentative clusters may be merged or split based on some criteria.

Density based Methods

- A cluster is a dense region of points, which is separated by low-density regions, from other regions of high density.
- Typically, for each data-point in a cluster, at least a minimum number of points must exist within a given radius.
- The method can deal with arbitrary shape clusters (especially when noise & outliers are present).

Grid-based methods

- The object-space rather than the data is divided into a grid.
- This is based on characteristics of the data.
- The method can deal with non-numeric data more easily.
- The method is not affected by data-ordering.

Model-based Methods

- A model is assumed, perhaps based on a probability distribution.
- Essentially, the algorithm tries to build clusters with
  → a high level of similarity within them → a low level of similarity between them.
- Similarity measurement is based on the mean values and the algorithm tries to minimize the squared error function.

PARTITIONAL METHODS

- The objects are divided into non-overlapping clusters (or partitions) such that each object is in exactly one cluster (Figure 4.1a).
- The method obtains a single-level partition of objects.
- The method requires the analyst
  - → to specify number of clusters(k) prior
  - → to specify starting seeds of the clusters
- The analyst may have to use iterative approach in which he has to run the method many times
  - → specifying different numbers of clusters & different starting seeds
  - → then selecting the best solution
- The method converges to a local minimum rather than the global minimum.
- These are popular since they tend to be computationally efficient and are more easily adapted for very large datasets.
- The aim of partitional method is
  - → to reduce the variance within each cluster &
  - → to have large variance between the clusters

THE K-MEANS METHOD

- This method can only be used if the data-object is located in the main memory.
- The method is called K-means since each of the K clusters is represented by the mean of the objects(called the centriod) within it.
- The method is also called the centroid-method since
  - → at each step, the centroid-point of each cluster is assumed to be known and
  - → each of the remaining points are allocated to the cluster whose centroid is closest to it
- The algorithm is as follows
  1) Select the number of clusters=k (Figure 4.1c).
  2) Pick k seeds as centroids of k clusters. The seeds may be picked randomly unless the user has some insight into the data.
  3) Compute Euclidean distance of each object in the dataset from each of the centroids.
  4) Allocate each object to the cluster it is nearest to (based on the distances computed in the previous step).
  5) Compute the centroids of clusters by computing the means of the attribute values of the objects in each cluster.
  6) Check if the stopping criterion has been met (e.g. the cluster-membership is unchanged). If yes, go to step 7. If not, go to step 3.
  7) One may decide
     - → to stop at this stage or
     - → to split a cluster or combine two clusters until a stopping criterion is met

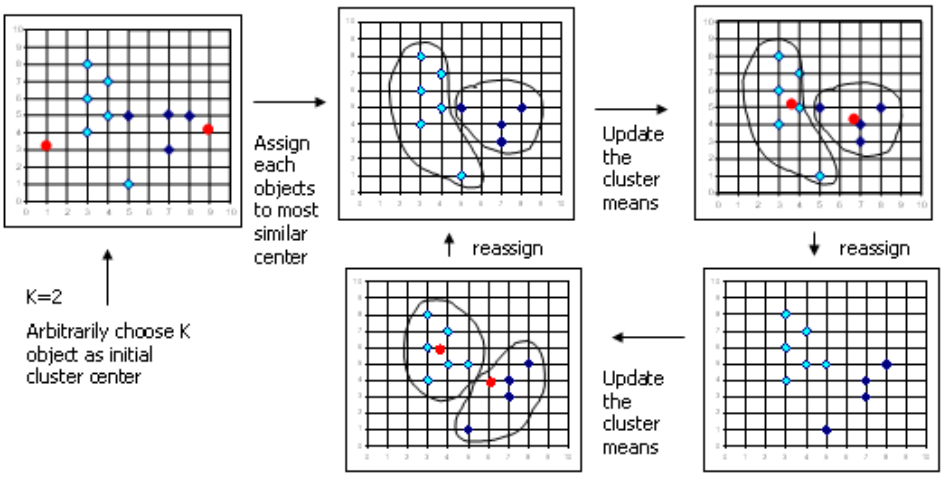


Figure 4.1c

Example 4.1

• Consider the data about students given in Table 4.3.

Table 4.3 Data for Example 4.1

Student	*Age*	*Mark1*	*Mark2*	*Mark3*
S_1	18	73	75	57
S_2	18	79	85	75
S_3	23	70	70	52
S_4	20	55	55	55
S_5	22	85	86	87
S_6	19	91	90	89
S_7	20	70	65	60
S_8	21	53	56	59
S_9	19	82	82	60
S_{10}	47	75	76	77

Step 1 and 2:

• Let the three seeds be the first three students as shown in Table 4.4.

Table 4.4 The three seeds for Example 4.1

Student	*Age*	*Mark1*	*Mark2*	*Mark3*
S_1	18	73	75	57
S_2	18	79	85	75
S_3	23	70	70	52

Step 3 and 4:

• Now compute the distances using the 4 attributes and using the sum of absolute differences for simplicity. The distance values for all the objects are given in Table 4.5.

Table 4.5 First iteration—allocating each object to the nearest cluster

					Distances from clusters			
C_1 C_2 C_3	*18.0* *18.0* *23.0*	*73.0* *79.0* *70.0*	*75.0* *85.0* *70.0*	*57.0* *75.0* *52.0*	*From* C_1	*From* C_2	*From* C_3	*Allocation to the nearest cluster*
S_1	18.0	73.0	75.0	57.0	0.0	34.0	18.0	C_1
S_2	18.0	79.0	85.0	75.0	34.0	0.0	52.0	C_2
S_3	23.0	70.0	70.0	52.0	18.0	52.0	0.0	C_3
S_4	20.0	55.0	55.0	55.0	42.0	76.0	36.0	C_3
S_5	22.0	85.0	86.0	87.0	57.0	23.0	67.0	C_2
S_6	19.0	91.0	90.0	89.0	66.0	32.0	82.0	C_2
S_7	20.0	70.0	65.0	60.0	18.0	46.0	16.0	C_3
S_8	21.0	53.0	56.0	59.0	44.0	74.0	40.0	C_3
S_9	19.0	82.0	82.0	60.0	20.0	22.0	36.0	C_1
S_{10}	47.0	75.0	76.0	77.0	52.0	44.0	60.0	C_2

Step 5:

• Table 4.6 compares the cluster means of clusters found in Table 4.5 with the original seeds.

Table 4.6 Comparing new centroids and the seeds

	Age	*Mark1*	*Mark2*	*Mark3*
C_1	18.5	77.5	78.5	58.5
C_2	26.5	82.5	84.3	82.0
C_3	21	61.5	61.5	56.5
Seed1	18	73	75	57
Seed2	18	79	85	75
Seed3	23	70	70	52

Step 3 and 4:

• Use the new cluster means to re-compute the distance of each object to each of the means, again allocating each object to the nearest cluster. Table 4.7 shows the second iteration.

Table 4.7 Second iteration—allocating each object to the nearest cluster

					Distances from clusters			
C_1	18.5	77.5	78.5	58.5	From C_1	From C_2	From C_3	Allocation to the nearest cluster
C_2	26.5	82.5	84.3	82.0				
C_3	21.0	62.0	61.5	56.5				
S_1	18.0	73.0	75.0	57.0	10.0	52.3	28.0	C_1
S_2	18.0	79.0	85.0	75.0	25.0	19.8	62.0	C_2
S_3	23.0	70.0	70.0	52.0	27.0	60.3	23.0	C_3
S_4	20.0	55.0	55.0	55.0	51.0	90.3	16.0	C_3
S_5	22.0	85.0	86.0	87.0	47.0	13.8	79.0	C_2
S_6	19.0	91.0	90.0	89.0	56.0	28.8	92.0	C_2
S_7	20.0	70.0	65.0	60.0	24.0	60.3	16.0	C_3
S_8	21.0	53.0	56.0	59.0	50.0	86.3	17.0	C_3
S_9	19.0	82.0	82.0	60.0	10.0	32.3	46.0	C_1
S_{10}	47.0	75.0	76.0	77.0	52.0	41.3	74.0	C_2

• Number of students in cluster 1 is again 2 and the other two clusters still have 4 students each.

• A more careful look shows that the clusters have not changed at all. Therefore, the method has converged rather quickly for this very simple dataset.

• The cluster membership is as follows

Cluster C_1= {S_1, S_9} Cluster C_2= {S_2, S_5, S_6, S_{10}} Cluster C_3= {S_3, S_4, S_7, S_8}

SCALING AND WEIGHTING

• For clustering to be effective, all attributes should be converted to a similar scale.

• There are a number of ways to transform the attributes.

One possibility is to transform the attributes

→ to a normalized score or → to a range(0,1)

Such transformations are called *scaling.*

• Some other approaches are:

1) Divide each attribute by the mean value of that attribute.

This approach reduces the mean of each attribute to 1.

This does not control the variation; some values may still be large, others small.

2) Divide each attribute by the difference between largest-value and smallest-value.

This approach

→ decreases the mean of attributes that have a large range of values &

→ increases the mean of attributes that have small range of values.

3) Convert the attribute values to "standardized scores" by subtracting the mean of the attribute from each attribute value and dividing it by the standard deviation.

Now, the mean & standard-deviation of each attribute will be 0 and 1 respectively.

SUMMARY OF THE K MEANS METHOD

• K means is an iterative improvement greedy method.

• A number of iterations are normally needed for convergence and

therefore the dataset is processed a number of times.

• Following are number of issues related to the method (Disadvantages)

1) The results of the method depend strongly on the initial guesses of the seeds. Need to specify k, the number of clusters, in advance.

2) The method can be sensitive to outliers. If an outlier is picked as a starting seed, it may end-up in a cluster at its own.

3) The method does not consider the size of the clusters.

4) The method does not deal with overlapping clusters.

5) Often, the local optimum is not as good as the global optimum.

6) The method implicitly assumes spherical probability distribution.

7) The method needs to compute Euclidean distances and means of the attribute values of objects within a cluster. (i.e. Cannot be used with categorical data).

EXPECTATION MAXIMIZATION METHOD

• Assumption is that the objects in the dataset have attributes whose values are distributed according to some linear combination of simple probability distributions.

• K-means method involves assigning objects to clusters to minimize within-group variation, whereas the EM method assigns objects to different clusters with certain probabilities in an attempt to maximize expectation(or likelihood) of assignment.

• The EM method consists of a two-step iterative algorithm:

1) The estimation-step (or E-step) involves estimating the probability distributions of the clusters given the data.

2) The maximization-step(M-step) involves finding the model parameters that maximize the likelihood of the solution.

HIERARCHICAL METHODS

• A set of nested clusters is organized as a hierarchical tree (Figure 4.1d).

• This approach allows clusters to be found at different levels of granularity.

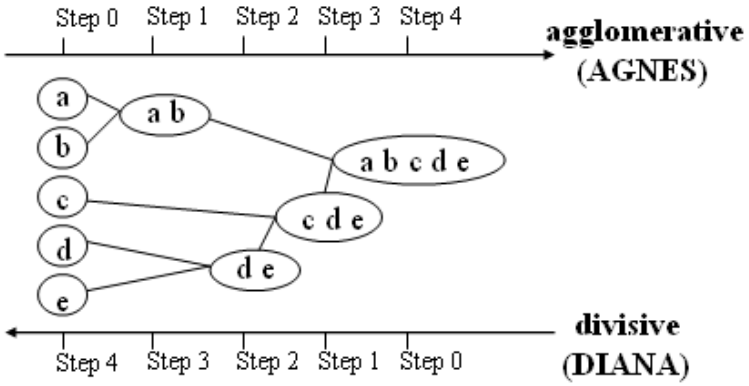


Figure 4.1d

• Two types of hierarchical approaches:

1. Agglomerative approach: Each object at the start is a cluster by itself and the nearby clusters are repeatedly merged resulting in larger clusters until some stopping criterion is met or all the objects are merged into a single large cluster (Figure 4.1e).

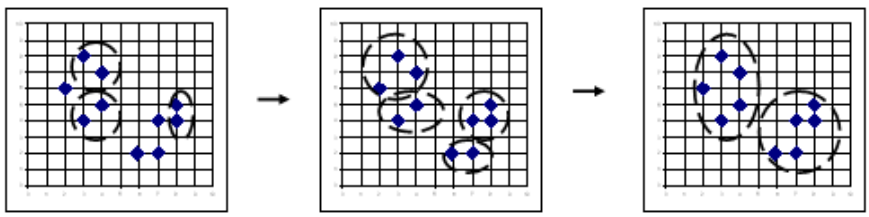


Figure 4.1e

2. Divisive approach: All the objects are put in a single cluster to start. The method then repeatedly resulting in smaller clusters until a stopping criterion is reached or each cluster has only one object in it (Figure 4.1f).

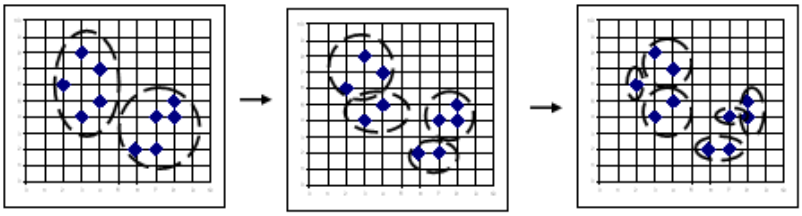


Figure 4.1f

DISTANCE BETWEEN CLUSTERS

• The hierarchical methods require distances between clusters to be computed. These distance metrics are often called *linkage metrics*.

• Following methods are used for computing distances between clusters:

1) Single-link(nearest neighbor) 2) Complete-link(farthest neighbor)
3) Centriod 4) Average 5) Ward's minimum variance

SINGLE-LINK

• The distance between 2 clusters is defined as the minimum of the distances between all pairs of points(x,y),

where x is from the first cluster &
y is from the second cluster. $D(x, y) = \min(x_i, y_j)$

• If there are m elements in one cluster and n in another cluster,
all mn pairwise distances must be computed and
the smallest is chosen (Figure 4.2).

• Disadvantages:

→ Each cluster may have an outlier and the 2 outliers may be nearby and so the distance between the 2 clusters would be computed to be small.

→ Single link can form a chain of objects as clusters are combined since there is no constraint on distance between objects that are far away from each other.

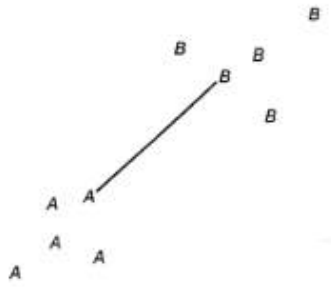


Figure 4.2 Single-link distance between two clusters.

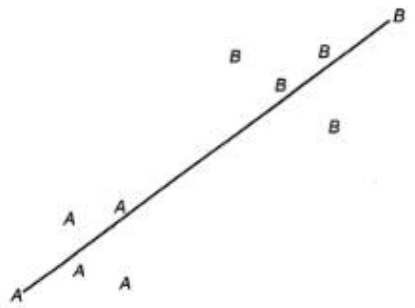


Figure 4.3 Complete-link distance between two clusters.

COMPLETE-LINK

• The distance between 2 clusters is defined as the maximum of the distances between all pairs of points(x,y). i.e. $D(x, y)=\max(x_i, y_j)$

• This is strongly biased towards compact clusters (Figure 4.3).

• Disadvantages:

→ Each cluster may have an outlier and the 2 outliers may be very far away and so the distance between the 2 clusters would be computed to be large.

→ If a cluster is naturally of a shape, say, like a banana then perhaps this method is not appropriate.

CENTRIOD

• The distance between 2 clusters is defined as the distance between the centroids of the clusters.

• Usually, the squared Euclidean distance between the centroids is used. i.e. $D(x, y)=D(C_i, C_j)$

• Advantages:

→ This method is easy and generally works well.

→ This method is more tolerant of somewhat longer clusters than complete link algorithm.

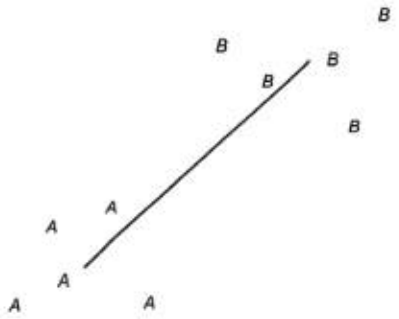


Figure 4.4 The centroid distance between two clusters.

AVERAGE

- The distance between 2 clusters is defined as the average of all pairwise distances between
  → an object from one cluster and
  → another from the other cluster. i.e. $D(x, y)=avg(x_i, y_j)$
- Therefore, if there are m elements in one cluster and n in the other,
  there are mn distances to be computed, added and divided by mn.
- Advantages:
  → This method tends to join clusters with small variances.
  → This method is easy and generally works well.
  → This method is more tolerant of somewhat longer clusters than complete link algorithm.

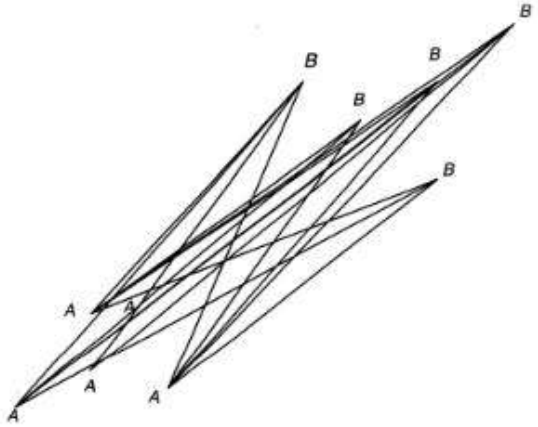


Figure 4.5 The average-link distance between two clusters.

WARD'S MINIMUM VARIANCE METHOD

- Ward's distance is the difference between
  → total within the cluster sum of squares for the 2 clusters separately and
  → total within the cluster sum of squares resulting from merging the 2 clusters
- An expression for ward's distance is given by

$$D_W(A, B) = N_A N_B D_c(A, B)/(N_A + N_B)$$

where
$D_W(A,B)$ = ward's minimum variance distance between clusters A and B with N_A & N_B objects in them respectively.
$D_c(A,B)$ = centriod distance between the 2 clusters computed as squared Euclidean distance between the centroids.

- Advantages:
  → This method generally works well and results in creating small tight clusters
  → This method produces clusters with roughly the same number of objects
  → This method tends to join clusters with a small number of objects
  → The distance measure can be sensitive to outliers

AGGLOMERATIVE METHOD

- This method is basically a bottom-up approach.
- The algorithm is as follows:

1) Allocate each point to a cluster of its own. Thus, we start with n clusters for n objects.
2) Create a distance-matrix by computing distances between all pairs of clusters (either using the single link metric or the complete link metric). Sort these distances in ascending order.
3) Find the 2 clusters that have the smallest distance between them.
4) Remove the pair of objects and merge them.
5) If there is only one cluster left then stop.
6) Compute all distances from the new cluster and update the distance-matrix after the merger and go to step 3.

Example 4.3

- We now use agglomerative technique for clustering the data given in Table 4.10.

Table 4.10 Data for Example 4.3

Student	*Age*	*Mark1*	*Mark2*	*Mark3*
S_1	18	73	75	57
S_2	18	79	85	75
S_3	23	70	70	52
S_4	20	55	55	55
S_5	22	85	86	87
S_6	19	91	90	89
S_7	20	70	65	60
S_8	21	53	56	59
S_9	19	82	82	60
S_{10}	47	75	76	77

Steps 1 and 2:

- Allocate each point to a cluster and compute the distance-matrix using the centroid method.
- The distance-matrix is symmetric, so we only show half of it in Table 4.11.
- Table 4.11 gives the distance of each object with every other object.

Table 4.11 Distance matrix for data in Example 4.3

	S_1	S_2	S_3	S_4	S_5	S_6	S_7	S_8	S_9
S_1	0								
S_2	34	0							
S_3	18	52	0						
S_4	42	76	36	0					
S_5	57	23	67	95	0				
S_6	66	32	82	106	15	0			
S_7	18	46	16	30	65	76	0		
S_8	44	74	40	8	91	104	28	0	
S_9	20	22	36	60	37	46	30	115	0
S_{10}	52	44	60	90	55	70	60	98	58

Steps 3 and 4:

- The smallest distance is 8 between objects S_4 and S_8. They are combined and removed and we put the combined cluster (C_1) where the object S_4 was.
- Table 4.12 is now the new distance-matrix. All distances except those with cluster C_1 remain unchanged.

Table 4.12 Distance matrix after merging S_4 and S_8 into cluster C_1

	S_1	S_2	S_3	C_1	S_5	S_6	S_7	S_9
S_1	0							
S_2	34	0						
S_3	18	52	0					
C_1	41	75	38	0				
S_5	57	23	67	93	0			
S_6	66	32	82	105	15	0		
S_7	18	46	16	29	65	76	0	
S_9	20	22	36	59	37	46	30	0
S_{10}	52	44	60	88	55	70	60	58

Steps 5 and 6:

• The smallest distance now is 15 between objects S_5 and S_6. They are combined in C_2 cluster and S_5 and S_6 are removed.

Steps 3, 4, 5 and 6: Table 4.13 is the updated distance-matrix.

Table 4.13 Distance matrix after merging S_5 and S_6 into cluster C_2

	S_1	S_2	S_3	C_1	C_2	S_7	S_9
S_1	0						
S_2	34	0					
S_3	18	52	0				
C_1	41	75	38	0			
C_2	61.5	27.5	74.5	97.5	0		
S_7	18	46	16	29	69.5	0	
S_9	20	22	36	59	41.5	30	0
S_{10}	52	44	60	88	62.5	60	58

• The result of using the agglomerative method could be something like that shown in Figure 4.6.

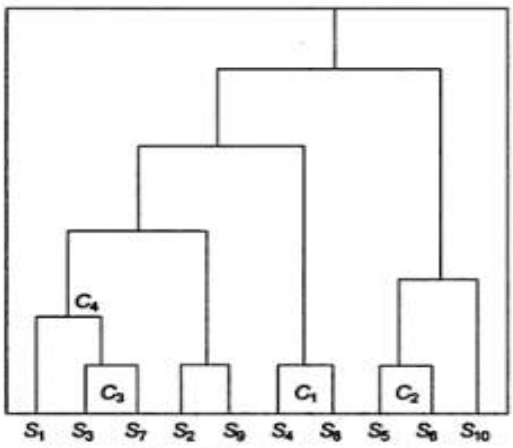


Figure 4.6 A possible result of using the agglomerative method.

DIVISIVE HIERARCHICAL METHODS

- These methods
  - → start with the whole dataset as one cluster
  - → then proceed to recursively divide the cluster into two sub-clusters and
  - → continue until each cluster has only one object.
- Two types of divisive methods are:
  1) Monothetic: This splits a cluster using only one attribute at a time.
     An attribute that has the most variation could be selected.
  2) Polythetic: This splits a cluster using all of the attributes together.
     Two clusters far apart could be build based on distance between objects.
- The algorithm is as follows:
  1) Decide on a method of measuring the distance between 2 objects. Also, decide a threshold distance.
  2) Create a distance-matrix by computing distances between all pairs of objects within the cluster. Sort these distances in ascending order.
  3) Find the 2 objects that have the largest distance between them. They are the most dissimilar objects.
  4) If the distance between the 2 objects is smaller than the pre-specified threshold and there is no other cluster that needs to be divided then stop, otherwise continue.
  5) Use the pair of objects as seeds of a K-means method to create 2 new clusters.
  6) If there is only one object in each cluster then stop otherwise continue with step 2.
- We need to resolve the following 2 issues:
  1) Which cluster to split next?
     i) Split the cluster in some sequential order.
     ii) Split the cluster that has the largest number of objects.
     iii) Split the cluster that has the largest variation within it.
  2) How to split a cluster?
     A distance-matrix is created and the 2 most dissimilar objects are selected as seeds of 2 new clusters. The K-means method is then used to split the cluster.

Example 4.4

- Consider the distance-matrix in Table 4.14.

Table 4.14 Distance matrix for objects in Example 4.4

	S_1	S_2	S_3	S_4	S_5	S_6	S_7	S_8	S_9
S_1	0								
S_2	34	0							
S_3	18	52	0						
S_4	42	76	36	0					
S_5	57	23	67	95	0				
S_6	66	32	82	106	15	0			
S_7	18	46	16	30	65	76	0		
S_8	44	74	40	8	91	104	28	0	
S_9	20	22	36	60	37	46	30	115	0
S_{10}	52	44	60	90	55	70	60	98	58

- The largest distance is 115 between objects S_8 and S_9. They become the seeds of 2 new clusters. K means is used to split the group into 2 clusters.

Table 4.15 Distances from the seeds of the two clusters

	S_1	S_2	S_3	S_4	S_5	S_6	S_7	S_8	S_9	S_{10}
S_8	44	74	40	8	91	104	28	0	115	98
S_9	20	22	36	60	37	46	30	115	0	58

- Cluster C_1 includes S_4, S_7, S_8 and S_{10}.
  Cluster C_2 includes S_1, S_2, S_3, S_5, S_6, S_9.
- Since none of the stopping criteria have been met, we decide to split the larger cluster next and then repeat the process.
- We find the largest distance in C_2 first as shown in Table 4.16. The largest distance in C_2 is 82 between S_3 & S_6. C_2 can therefore be split with S_3 & S_6 as seeds.

Table 4.16 Distance matrix for objects in cluster C_2

	S_1	S_2	S_3	S_5	S_6
S_1	0				
S_2	34	0			
S_3	18	52	0		
S_5	57	23	67	0	
S_6	66	32	82	15	0
S_9	20	22	36	37	46

Table 4.17 Distance matrix for objects in cluster C_1

	S_4	S_7	S_8
S_4	0		
S_7	30	0	
S_8	8	28	0
S_{10}	90	60	98

• The distance-matrix of C_1 is given in Table 4.17. The largest distance is 98 between S_8 and S_{10}. C_1 can therefore be split with S_8 and S_{10} as seeds.
• The method continues like this until the stopping criteria is met.

SUMMARY OF HIERARCHICAL METHODS

Advantages

1) This method is conceptually simpler and can be implemented easily.
2) In some applications, only proximity-data is available and then this method may be better.
3) This method can provide clusters at different levels of granularity.
4) This method can provide more insight into data by showing a hierarchy of clusters (than a flat cluster structure created by a partitioning method like the K-means method).
5) Do not have to assume any particular number of clusters.

Disadvantage

1) Do not scale well: Time complexity of at least $O(n^2)$, where n is number of total objects.
2) The distance-matrix requires $O(n^2)$ space and becomes very large for a large number of objects.
3) Different distance metrics and scaling of data can significantly change the results.
4) Once a decision is made to combine two clusters, it cannot be undone.

DENSITY-BASED METHODS

• A cluster is a dense region of points, which is separated by low-density regions, from other regions of high density.
• Typically, for each data-point in a cluster, at least a minimum number of points must exist within a given radius.
• Data that is not within such high-density clusters is regarded as outliers or noise.
• **DBSCAN** is one example of a density-based method for clustering.
• DBSCAN stands for Density Based Spatial Clustering of Applications with Noise.
• This requires 2 input parameters:
→ size of the neighborhood(R) &
→ minimum points in the neighborhood(N).
• The point-parameter N
→ determines the density of acceptable clusters &
→ determines which objects will be labeled outliers or noise.
• The size-parameter R determines the size of the clusters found.
• If R is big enough, there would be one big cluster and no outliers.
If R is small, there will be small dense clusters and there might be many outliers.
• We define a number of concepts that are required in the DBSCAN method:
1) Neighbourhood: The neighborhood of an object y is defined as all the objects that are within the radius R from y.
2) Core-object: An object y is called a core-object if there are N objects within its neighborhood.
3) Proximity: Two objects are defined to be in proximity to each other if they belong to the same cluster. Object x_1 is in proximity to object x_2 if two conditions are satisfied:
i) The objects are close enough to each other, i.e. within a distance of R.
ii) x_2 is a core object.
4) Connectivity: Two objects x_1 and x_n are connected if there is a chain of objects $x_1, x_2 \ldots x_n$ from x_1 to x_n such that each x_{i+1} is in proximity to object x_i.
• The algorithm is as follows
1) Select values of R and N (Figure 4.7).
2) Arbitrarily select an object p.
3) Retrieve all objects that are connected to p, given R and N.
4) If p is a core object, a cluster is formed.
5) If p is a border object, no objects are in its proximity. Choose another object. Go to step 3.
6) Continue the process until all of the objects have been processed.

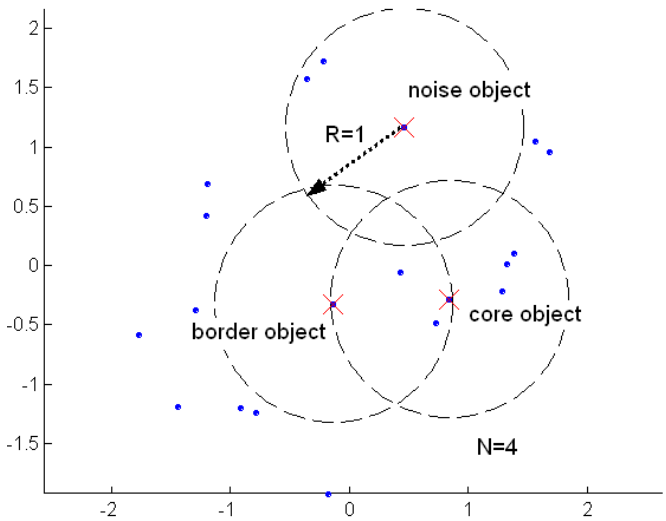


Figure 4.7: DBSCAN

DEALING WITH LARGE DATABASES

• A method requiring multiple scans of data that is disk-resident could be quite inefficient for large problems.

K MEANS METHOD FOR LARGE DATABASES

• The method
  → first picks the number of clusters & their seed centriods and
  → then attempts to classify each object to belong to one of the following 3 groups:
    1) Those that are certain to belong to a cluster. These objects together are called *discard-set*.
       Some information about these objects is computed and saved. This includes
       → number of objects n
       → a vector sum of all attribute values of the n objects and
       → a vector sum of squares of all attribute values of n objects
    2) Those that are sufficiently close to each other to be replaced by their summary. These objects together are called the *compression-set*.
       The objects are however sufficiently far away from each cluster's centroid that they cannot be put in the discard set.
    3) The remaining objects that are too difficult to assign to either of the 2 groups above. These objects are called the *retained-set*.
       These are stored as individual objects.
       They cannot be replaced by a summary.

HIERARCHICAL METHOD FOR LARGE DATABASES - CONCEPT OF FRACTIONATION

• Main idea is as follows:
  1) First split the data into manageable subsets called *fractions* &
  2) Then apply a hierarchical method to each fraction. The concept is called *fractionation*.
• Let M = largest number of objects that the hierarchical method may be applied to.
• The size M may be determined based on the size of the main memory.
• The algorithm is as follows:
  1) Split the large dataset into fractions of size M.
  2) The hierarchical method is applied to each fraction.
     Let C = number of clusters obtained from all the fractions.
  3) For each cluster, compute the mean of the attribute values of the objects.
     Let this mean vector be m_i, i=1, 2, . . C.
     These cluster means are called *meta-observation*.
     The meta-observation now becomes the data values that represent the fractions.
  4) If the C meta-observations are too large, go to step 1,
     otherwise apply same hierarchical method to the meta-observations obtained in step 3.
  5) Allocate each object of the original dataset to the cluster with the nearest mean obtained in step 4.

CLUSTER ANALYSIS SOFTWARE

• ClustanGraphics7 from Clustan offers a variety of clustering methods including K means, density based and hierarchical cluster analysis. The software provides facilities to display results of clustering including dendograms and scatterplots.
• CViz Cluster visualization from IBM is a visualization tool designed for analyzing high dimensional data in large, complex data set.
• Cluster 3.0, open source software. This uses the k means method, which includes multiple trials to find the best clustering solution.
• CLUTO provides a set of clustering methods including partitional, agglomerative and graph partitioning based on a variety of similarity/distance metrics.

QUALITY AND VALIDITY OF CLUSTER ANALYSIS METHODS

• Let number of clusters = k
Let clusters = C_i, i=1... . . . . k
Let total number of objects = N
Let number of objects in cluster = M_i so that

$$M_1 + M_2 + \cdots + M_k = N$$

• The within-cluster variation between the objects in a cluster is defined as the average squared distance of each object from the centroid of the cluster.

• If m_i is the centroid of the cluster C_i
then the *mean* of the cluster is given by

$$m_i = \{\Sigma x_j\}/M_i$$

and the internal cluster *variation* is given by

$$I_i = \{\Sigma(x_j - m_i)^2\}/M_i$$

• The average within-cluster variation is given by

$$I = (1/k)(I_1 + I_2 + \cdots + I_k) = (\Sigma I_i)/k$$

• The between cluster distances E is the average sum of squares of pairwise distances between the centroids of the k clusters. We may write E as

$$E = (1/k^2)\sum_i \sum_j (\mu_i - \mu_j)^2$$

• To achieve the best result of cluster analysis,
one possible approach might be choosing result that has largest E/I from the results available.

• If E is large, it shows good separation between the clusters.
If I is small, it means that we have tight clusters.

• The quality of a clustering method involves a number of criteria
1) Efficiency of the method.
2) Ability of the method to deal with noisy and missing data.
3) Ability of the method to deal with large problems.
4) Ability of the method to deal with a variety of attributes types and magnitudes.

EXERCISES

1) What is cluster analysis? What are its applications? (2)
2) Compare classification vs. cluster analysis. (6)
3) List out and explain desired features of cluster analysis method. (6)
4) List out and explain different types of data. (4)
5) List out and explain different distance measures. (4)
6) List out and explain different types of cluster analysis methods. (6)
7) Write algorithm for k-means method. (6)
8) Apply k-means method for clustering the data given in Table 4.3. (6)
9) List out disadvantages of k-means method. (6)
10) Explain scaling and weighting. (4)
11) Explain expectation maximization method. (4)
12) Compare agglomerative approach vs. divisive approach. (4)
13) Explain different methods used for computing distances b/t clusters. (6)
14) Write algorithm for agglomerative approach. (6)
15) Apply agglomerative technique for clustering data given in Table4.10. (6)
16) Write algorithm for divisive approach. (6)
17) List out advantages and disadvantages of hierarchical methods. (6)
18) Explain DBSCAN with its algorithm. (6)
19) Explain K means method for large databases. (4)
20) Explain hierarchical method for large databases. (6)
21) Explain quality and validity of cluster analysis methods (6)

UNIT 8: WEB MINING

WEB MINING

• Web-mining is the application of data-mining techniques to extract knowledge from web-data. (i.e. web-content, web-structure, and web-usage data).

• We interact with the web for the following purposes:

1) Finding Relevant Information

• We use the search-engine to find specific information on the web.

• *Query triggered process:* We specify a simple keyword-query and the response from a search-engine is a list of pages, ranked by their similarity to the query.

• Search tools have the following problems:

i) Low precision: This is due to the irrelevance of many of the search-results. We may get many pages of information which are not really relevant to our query.

ii) Low recall: This is due to inability to index all the information available on the web. Because some of the relevant pages are not properly indexed.

2) Discovering New Knowledge from the Web

• *Data triggered process:* This assumes that

→ we already have a collection of web-data and

→ we want to extract potentially useful knowledge out of it

3) Personalized Web Page Synthesis

• We may wish to synthesize a web-page for different individuals from the available set of web-pages.

• While interacting with the web, individuals have their own preferences for the style of the content and presentation.

4) Learning about Individual Users

• This is about knowing

→what the customers do and

→what the customers want

• Within this problem, there are sub-problems such as

→ problems related to effective web-site design and management

→ problems related to marketing etc

• Techniques from web-mining can be used to solve these problems.

Other related techniques from different research areas, such as DB(database), IR(information retrieval) & NLP(natural language processing), can also be used.

• Web-mining has 3 main operations:

→ clustering (e.g. finding natural groupings of users, pages)

→ associations (e.g. which URLs tend to be requested together)

→ sequential analysis (e.g. the order in which URLs tend to be accessed)

• Web mining techniques can be classified into 3 areas of interest (Figure 10.1):

→ web-content mining (e.g. text, image, records, etc.)

→ web-structure mining (e.g. hyperlinks, tags, etc)

→ web-usage mining (e.g. http logs, app server logs, etc.)

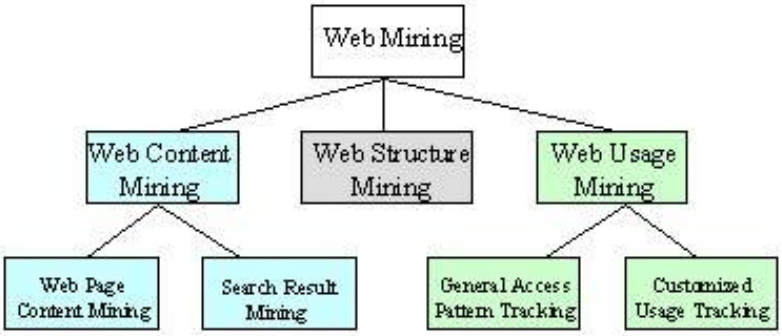


Figure 10.1: Web mining

WEB CONTENT MINING

- This is the process of extracting useful information from the contents of web-documents.
- We see more & more government information are gradually being placed on the web in recent years.
- We have
  - → digital libraries which users can access from the web
  - → web-applications which users can access through web-interfaces
- Some of the web-data are hidden-data, and some are generated dynamically as a result of queries and reside in the DBMSs.
- The web-content consists of different types of data such as text, image, audio, video as well as hyperlinks.
- Most of the research on web-mining is focused on the text or hypertext contents.
- The textual-parts of web-data consist of
  - → unstructured-data such as free texts
  - → semi structured-data such as HTML documents &
  - → structured-data such as data in the tables
- Much of the web-data is unstructured, free text-data. As a result, text-mining techniques can be directly employed for web-mining.
- Issues addressed in text mining are, topic discovery, extracting association patterns, clustering of web documents and classification of Web Pages.
- Research activities on this topic have drawn heavily on techniques developed in other disciplines such as IR(Information Retrieval) and NLP(Natural Language Processing).

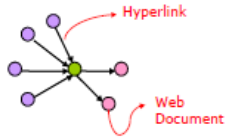
WEB USAGE MINING

- This deals with studying the data generated by the web-surfer's sessions (or behaviors).
- Web-content/structure mining utilise the primary-data(or real) on the web. On the other hand, web-usage mining extracts the secondary-data derived from the interactions of the users with the web.
- The secondary-data includes the data from
  - → web-server access logs
  - → browser logs
  - → user transactions/queries
  - → user profiles
  - → registration/bookmark data
  - → cookies
- There are 2 main approaches in web-usage mining:
  - **1) General Access Pattern Tracking**
    - This can be used to learn user-navigation patterns.
    - This can be used to analyze the web-logs to understand access-patterns and trends.
    - This can shed better light on the structure & grouping of resource providers.
  - **2) Customized Usage Tracking**
    - This can be used to learn a user-profile in adaptive interfaces (personalized).
    - This can be used to analyze individual trends.
    - Main purpose: is to customize web-sites to users.
    - Based on user access-patterns, following things can be dynamically customized for each user over time:
      - → information displayed
      - → depth of site-structure
      - → format of resources
- The mining techniques can be classified into 2 commonly used approaches:
  1) The first approach maps the usage-data of the web-server into relational-tables before a traditional data-mining technique is applied.
  2) The second approach uses the log-data directly by utilizing special pre-processing techniques.

WEB STRUCTURE MINING

• The structure of a typical web-graph consists of web-pages as nodes, and hyperlinks as edges connecting related pages.

• Web-Structure mining is the process of discovering structure information from the web.



Web Graph Structure

• This type of mining can be performed either at the (intra-page) document level or at the (inter-page) hyperlink level.

• This can be used to classify web-pages.

• This can be used to generate information such as the similarity & relationship between different web-sites.

PageRank

• PageRank is a metric for ranking hypertext documents based on their quality.

• The key idea is that a page has a high rank if it is pointed to by many highly ranked pages

• The PageRank of a page A is given by

$$PR(p) = d/n + (1-d) \sum_{(q,p)\in G} \left(\frac{PR(q)}{Outdegree(q)}\right)$$

Here, n = number of nodes in the graph and

OutDegree(q) = number of hyperlinks on page q

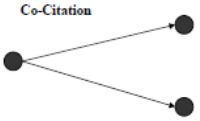
d = damping factor which can be set between 0 and 1 and is usually set to 0.85

Clustering & Determining Similar Pages

• For determining the collection of similar pages, we need to define the similarity measure between the pages. There are 2 basic similarity functions:

1) Co-citation

For a pair of nodes p and q, the co-citation is the number of nodes that point to both p and q.



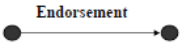
2) Bibliographic coupling

For a pair of nodes p and q, the bibliographic coupling is equal to the number of nodes that have links from both p and q.

Social Network Analysis

• This can be used to measure the relative standing or importance of individuals in a network.

• The basis idea is that if a web-page points a link to another web-page, then the former is, in some sense, endorsing the importance of the latter.



• Links in the network may have different weights, corresponding to the strength of endorsement.

TEXT MINING

• This is concerned with various tasks, such as
  → extraction of information implicitly contained in the collection of documents or
  → similarity-based structuring
• Text-collection lacks the imposed structure of a traditional database.
• The text expresses a vast range of information, but encodes the information in a form that is difficult to decipher automatically.
• Traditional data-mining techniques have been designed to operate on structured-databases.
• In structured-databases, it is easy to define the set of items and hence, it becomes easy to employ the traditional mining techniques.
  In textual-database, identifying individual items (or terms) is not so easy.
• Text-mining techniques have to be developed to process the unstructured textual-data to aid in knowledge discovery.
• The inherent nature of textual-data, namely unstructured characteristics, motivates the development of separate text-mining techniques.
• Following approaches can be used for text-mining:
  1) One way is to impose a structure on the textual-database and use any of the known data-mining techniques meant for structured-databases.
  2) The other approach would be to develop a very specific technique for mining that exploits the inherent characteristics of textual-databases.

INFORMATION RETRIEVAL (IR)

• This is concerned with finding and ranking documents that match the users' information needs.
• The way of dealing with textual-information is a keyword-based document representation.
• A body of text is analyzed by its constituent-words, and various techniques are used to build the core words for a document.
• The goals are
  → to find documents that are similar based on some specification of the user
  → to find right index terms in a collection, so that querying will return appropriate document
• Recent trends in IR research include
  → document classification
  → data visualization
  → filtering

INFORMATION EXTRACTION (IE)

• This is concerned with transforming a collection of documents, usually with the help of an IR system, into information that is more readily digested and analyzed.
• IE extracts relevant facts from the documents
  whereas IR selects relevant documents.
• In general, IE works at a finer granularity level than IR does on the documents.
• Most IE systems use data-mining(or machine-learning) techniques to learn the extraction patterns (or rules) for documents semi-automatically or automatically.
• The results of the IE process could be in the form of
  → structured database or
  → summary/compression of the original document

UNSTRUCTURED TEXT

• Unstructured-documents are free texts, such as news stories.

• To convert an unstructured-document to a structured form, following features can be extracted:

Word Occurrences

• The vector-representation (or bag of words) takes single words found in the training-set as features, ignoring the sequence in which the words occur.

• The feature is said to be boolean, if we consider whether a word either occurs or does not occur in a document.

• The feature is said to be frequency-based, if the frequency of the word in a document is taken into consideration.

Stopword Removal

• Stopwords are frequently occurring and insignificant words in a language that help construct sentences but do not represent any content of the documents.

• Common stopwords in English include:

a, about, an, are, as, at, be, by, for, from, how, in, is, of, on, or, that, the, these, this, to, was, what, when, where, who, will, with

• Such words can be removed before documents are indexed and stored.

Stemming

• *Stemming* refers to the process of reducing words to their morphological roots or stems

• A stem is the portion of a word that is left after removing its prefixes and suffixes.

• For example, "informing", "information", "informer" & "informed" are reduced to "inform".

POS

• POS stands for Part of Speech.

• There can be 25 possible values for POS tags.

• Most common tags are noun, verb, adjective and adverb.

• Thus, we can assign a number 1,2,3,4 or 5, depending on whether the word is a noun, verb, adjective, adverb or any other, respectively.

Positional Collocations

• The values of this feature are the words that occur one or two position to the right or left of the given word.

n-gram

• An n-gram is a contiguous sequence of n items from a given sequence of text.

• This can be used for predicting the next item in a sequence.

Sample sequence	1-gram sequence	2-gram sequence	3-gram sequence
... to be or not to be ...	..., to, be, or, not, to, be, ...	..., to be, be or, or not, not to, to be, ...	..., to be or, be or not, or not to, not to be, ...

LSI

• LSI stands for Latent Semantic Indexing.

• *LSI* is an indexing and retrieval method to identify the relationships between the terms and concepts contained in an unstructured collection of text.

TEXT CLUSTERING

• Once the features of an unstructured-text are identified or the structured-data of the text is available, text-clustering can be done by using any clustering technique.
• One popular text-clustering algorithm is ward's minimum variance method.
• Ward's method is an agglomerative hierarchical clustering technique and it tends to generate very compact clusters.
• We can take either the Euclidean metric or hamming distance as the measure of the dissimilarities between feature vectors.
• The clustering method begins with 'n' clusters, one for each text.
• At any stage, 2 clusters are merged to generate a new cluster based on the following criterion:

$$D_{KL} = \frac{\left\|\overline{x_K} - \overline{x_L}\right\|^2}{\frac{1}{N_K} + \frac{1}{N_L}}$$

where x_k is mean value of the dissimilarity for the cluster C_k and
n_k is the number of elements in this cluster.

Scatter/Gather

• This is a method of grouping the documents based on the overall similarities in their content.
• Scatter/gather is so named because it allows the user to scatter documents into groups(or clusters), then gather a subset of these groups and re-scatter them to form new groups.
• Each cluster is represented by a list of topical terms, that is, a list of words that attempt to give the user an idea of what the documents in the cluster are about.

EXERCISES

• Write a short note on following (5 marks each)
1) Web mining
2) Web content mining
3) Web structure mining
4) Web usage mining
5) Text mining
6) IR and IE
7) Unstructured text
8) Text clustering

UNIT 8(CONT.): TEMPORAL & SPATIAL DATA MINING

TEMPORAL DATA MINING

• This can be defined as non-trivial extraction of potentially-useful & previously-unrecorded information with an implicit/explicit temporal-content, from large quantities of data.
• This has the capability to infer causal and temporal-proximity relationships.
• Consider, for example, an association-rule - "Any person who buys a car also buys a steering-lock". But if we take the temporal-aspect into consideration, this rule would be - "Any person who buys a car also buys a steering-lock *after* that".

TYPES OF TEMPORAL DATA

• There can be 4 different levels of temporality:

Static

- Static-data are free of any temporal-reference.
- Inferences derived from static-data are also free of any temporality.

Sequences (Ordered Sequences of Events)

- Though there may not be any explicit reference to time, there exists a temporal-relationship between data-items.
- For example, market-basket transaction. The entry-sequence of transactions automatically incorporates a sort of temporality.
- While most collections are often limited to the sequence-relationships *before* and *after*, this category also includes the richer relationships , such as *during, meet, overlap* etc.
- Sequence-mining is one of the major activities in temporal-mining.

Timestamped

- The temporal-information is explicit.
- The relationship can be quantitative, in the sense that
  → we can say the exact temporal-distance between the data-elements &
  → we can say that one transaction occurred before another
- For example, census data, land-use data etc.
- Inferences derived from this data can be temporal or non-temporal.
- Time-series data are a special case of this category. The events are uniformly spaced on the time-scale.

Fully Temporal

- The validity of the data-element is time-dependent.
- Inferences derived from this data are necessarily temporal.

TEMPORAL DATA MINING TASKS

• There can be 4 different tasks:

Temporal Association

- Here, we attempt to discover temporal-associations between non-temporal itemsets.
- For example, "70% of the readers who buy a DBMS book also buy a Data Mining book after a semester".

Temporal Classification

- We can extend the concept of decision-tree construction on temporal-attributes.
- For example, a rule could be: "The first case of malaria is normally reported after the first pre-monsoon rain and during the months of May-August".

Trend Analysis

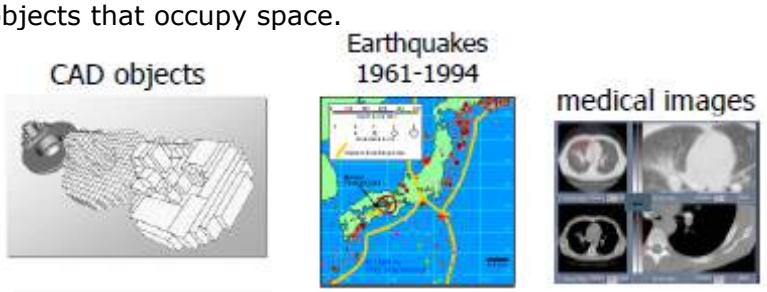
- The analysis of one or more time series of continuous data may show similar trends i.e. similar shapes across the time axis.
- For example, "The deployment of the Android OS is increasingly becoming popular in the smartphone industry".

TEMPORAL ASSOCIATION RULES

• Association rules identify whether a particular subset of items is supported by an adequate number of transactions.
• The presence of a temporal association rule may suggest a number of interpretations such as
→ The earlier event plays some role in causing the later event
→ There is a third set of reasons that cause both events
→ The confluence of events is coincidental
• Temporal association rules are sometimes viewed in the literature as causal rules.
• Causal rules describe relationships, where changes in one event cause subsequent changes in other parts of the domain.
• They are common targets of scientific investigation within the medical domain, where the search for factors that may cause or aggravate a particular disease is important.
• The static properties, such as gender, and the temporal properties, such as medical treatments, are taken into account during mining.

SPATIAL DATABASE

• A spatial database stores a large amount of space-related data(or objects), such as
→ map navigation data
→ preprocessed remote sensing data or
→ medical imaging data
• Spatial objects are objects that occupy space.



• Spatial objects usually consist of both spatial and non-spatial data.
→ Spatial data: data related to spatial description of the objects such as coordinates, areas, latitudes, perimeters, spatial relations (distance, topology, direction).
Example: earthquake points, town coordinates on map, etc.
→ Non-spatial data: other data associated to spatial objects.
Example: earthquake degrees, population of a town, etc
• Spatial objects can be classified into different classes (hospital class, airport class, etc.), several classes can be hierarchically organized in layers (countries, provinces, districts, etc.)
• Non-spatial data can be stored in relational databases
Spatial data are typically described by
→ Geometric properties: coordinates(schools), lines(roads, rivers), area(countries, cites)
→ Relational properties: adjacency(A is neighbor of B), inclusion(A is inside in B)
• Applications and Problems:
→ GIS(Geographic information systems) store information related to geographic locations on Earth (Weather, community infrastructure needs, disaster management)
→ Homeland security issues such as prediction of unexpected events & planning of evacuation.
→ Remote sensing and image classification
→ Biomedical applications include medical imaging and illness diagnosis

SPATIAL DATA MINING

• This refers to the extraction of knowledge, spatial relationships, or other interesting patterns not explicitly stored in spatial-databases.
• Consider a map of the city of Mysore containing various natural and man-made geographic features, and clusters of points (where each point marks the location of a particular house).
• The houses might be important because of their size, or their current market value.
• Clustering algorithms can be used to assign each point to exactly one cluster, with the number of clusters being defined by the user.
• We can mine varieties of information by identifying likely relationships.
• For ex, "the land-value of cluster of residential area around 'Mysore Palace' is high".
• Such information could be of value to realtors, investors, or prospective home buyers.
• This problem is not so simple because there may be a large number of features to consider.
• We need to be able to detect relationships among large numbers of geo-referenced objects without incurring significant overheads.

SPATIAL MINING TASKS

• This includes
→ finding characteristics rules
→ discriminant rules
→ association rules
• A spatial-characteristic rule is a general description of spatial-data. For example, a rule describing the general price ranges of houses in various geographic regions in a city.
• A spatial-discriminant rule is a general description of the features discriminating a class of spatial-data from other classes, For example, the comparison of price range of houses in different geographical regions.

Spatial Association Rles

• These describe the association between spatially related objects.
• We can associate spatial attributes with non spatial attributes. For example, "the *monthly rental* of houses *around the market area* is mostly Rs 500 per sq mt."
• We can associate spatial attributes with spatial attributes. For example, "uncontrolled tapping of borewell water in the *market area* may most likely cause earthquake in the *neighbouring areas*".

Attribute-oriented Induction

• The concept hierarchies of spatial and non-spatial attributes can be used to determine relationships between different attributes.
• One may be interested in a particular category of land-use patterns.
• A built-up area may be a recreational facility or a residential complex.
• Similarly, a recreational facility may be a cinema or a restaurant.
• One has to be very specific regarding the level of details to which one wants to discover spatial knowledge.

Aggregate Proximity Relationships

• This problem is concerned with relationships between spatial-clusters based on spatial and non-spatial attributes.
• Given n input clusters, we want to associate the clusters with classes of features (e.g. educational institutions which, in turn, may be comprised of secondary schools and junior colleges or higher institutions).
• The problem is to find classes of features that are in close proximity to most(or all) of the spatial clusters.

SPATIAL CLUSTERING

• The key idea of a density based cluster is that for each point of a cluster, its espilon neighbourhood has to contain at least a minimum number of points.

• We can generalize this concept in 2 different ways:

1) First, any other symmetric & reflexive neighbourhood relationship can be used instead of an espilon neighbourhood. It may be more appropriate to use topological relations such as intersects, meets or above/below to group spatially extended objects.

2) Second, instead of simply counting the objects in a neighbourhood of an object, other measures to define the "cardinality" of that neighbourhood can be used as well.

Spatial Characterization

• A spatial-characterization is a description of the spatial and non-spatial properties, which are typical for the target-objects but not for the whole database.

• For instance, different object types in a geographic database are mountains, lakes, highways, railroads etc.

• Spatial characterization considers both

→ properties of the target objects & → properties of their neighbours

• A spatial characterization rule of the form - "Apartments in Sainikpur have a high occupancy rate of retired army officers"- is an example.

SPATIAL TREND

• This can be defined as a regular change of one or more non-spatial attributes, when moving away from a given spatially referenced-object.

• For instance, "when we move away northwards from the 'Highway Circle', the rentals of residential houses decreases approximately at the rate of 5% per kilometer".

• One can also show the spatial trends pictorially by overlaying the direction of trend on a map.

SPATIO TEMPORAL DATA MINING

• A spatiotemporal database is a database that manages both space and time information.

• Common examples include:

→ Tracking of moving objects, which typically can occupy only a single position at a given time.

→ A database of wireless communication networks, which may exist only for a short timespan within a geographic region.

• Spatio-temporal data mining is an emerging research area dedicated to the development and application of novel computational techniques for the analysis of large spatiotemporal databases

• This encompasses techniques for discovering useful spatial and temporal relationships or patterns that are not explicitly stored in spatiotemporal datasets.

• Both the temporal and spatial dimensions add substantial complexity to data mining tasks.

• Classical data mining techniques often perform poorly when applied to spatiotemporal data sets for many reasons.

• First, spatial data is embedded in a continuous space, whereas classical datasets are in discrete notions like transactions.

• Second, a common assumption about independence of data samples in classical statistical analysis is generally false because spatial data tends to be highly auto-correlated. (For example, people with similar characteristics, occupation and background tend to cluster together in the same neighborhoods).

EXERCISES

• Write a short note on following (5 marks each)

1) Temporal data mining
2) Temporal data mining tasks
3) Temporal association rules
4) Spatial mining
5) Spatial mining tasks
6) Spatial clustering
7) Spatial trend
8) Spatio Temporal data mining