Data Warehousing & Data Mining

BSC.CSIT, 8th sem

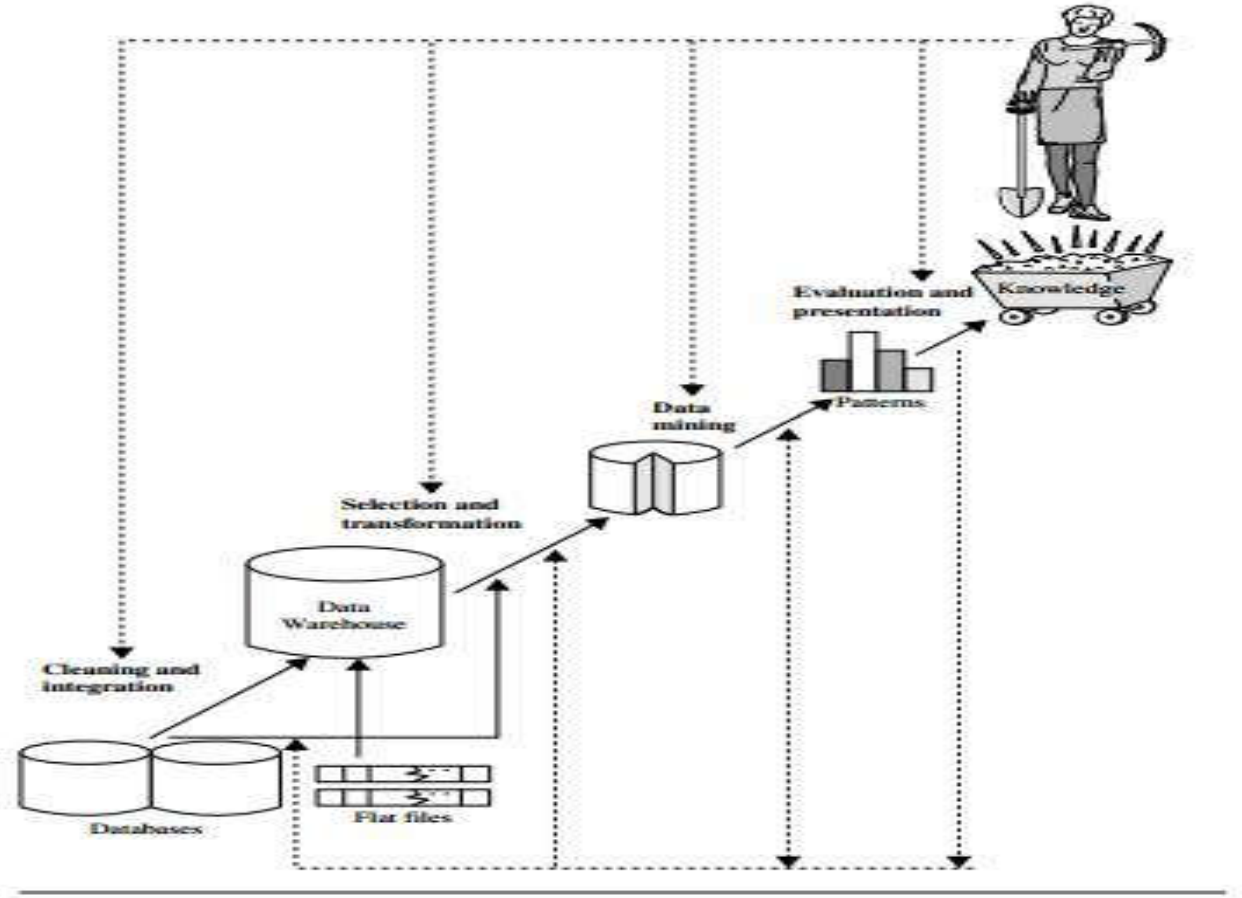
HCOE

Introduction:
Data Mining

- Data mining is the process of discovering interesting patterns and knowledge from the huge amount of data.
- Also known as knowledge extraction or knowledge mining from data, data/pattern analysis.
- Data Mining is one of the essential step in the process of KDD(Knowledge Discovery from Data).

Stages of KDD

-> Data mining as an essential step in the process of KDD



Stages of KDD Contd…

- Data Cleaning- Used to remove noise or inconsistent data.
- Data Integration- where data from multiple heterogeneous sources are combined.
- Data Selection- where data relevant to the analysis task are retrieved.
- Data Transformation- where data are transformed and changed into form appropriate for mining by performing summary or aggregation operations.

Stages of KDD Contd…

- Data Mining- where intelligent methods are applied to extract data patterns.
- Pattern evaluation- used to identify truly interesting patterns representing knowledge based on the interestingness measure.
- Knowledge presentation- where visualization and knowledge presentation techniques are used to present mined knowledge to user.

What kinds of data can be mined by data mining?

- A data mining can mined the

1. **Database data**

→ Database management system consists of collections of inter-related data, known as database and a set of program to access those data.

→ A relational database consists of collection of tables, each of which is assigned a unique name.

→ The table consists of rows and columns. The row contains a large set of tuples (records) and the column contains a set of attributes(fields).

Knowledge to be Mined

2. Data warehouse data

→ A data warehouse is a repository of information collected from multiple, heterogeneous sources and placed in a single site.

→ A data warehouse is a subject oriented,integrated,time variant and non-volatile collection of data that helps in the management decision making process.

→ Data warehouses are constructed via a process

of data cleaning, data integration,data transformation,data loading and periodic data refreshing.

→ A data warehouse is usually modeled by a multi-dimensional data structure called **data cube** which allows data to be modeled and viewed in a multiple dimension.

→ Each data cube consists of **dimensions** which corresponds to an attribute or a set of attributes in the schema and a **Cell** stores the value of some aggregated measure.

Knowledge to be Mined

3. Transactional data

→A transaction data base consists of a transactions like customer purchase, flight bookings etc..

→A transaction typically includes a unique transaction identifier (TID) and a set of items associated with that transaction.

4. Other kinds of data

a) Time series data
b) Spatial data
c) Multi media data
d) Web data etc…

Functionalities of Data Mining

- Multidimensional concept description: Characterization and discrimination
  - Generalize, summarize, and contrast data characteristics, e.g., dry vs. wet regions
- Frequent patterns, association, correlation vs. causality
  - Diaper → Beer [0.5%, 75%] (Correlation or causality?)
- Classification and prediction
  - Construct models (functions) that describe and distinguish classes or concepts for future prediction
    - E.g., classify countries based on (climate), or classify cars based on (gas mileage)
  - Predict some unknown or missing numerical values

Functionalities of Data Mining

- Cluster analysis
  - Class label is unknown: Group data to form new classes, e.g., cluster houses to find distribution patterns
  - Maximizing intra-class similarity & minimizing interclass similarity
- Outlier analysis
  - Outlier: Data object that does not comply with the general behavior of the data
  - Noise or exception? Useful in fraud detection, rare events analysis
- Trend and evolution analysis
  - Trend and deviation: e.g., regression analysis
  - Sequential pattern mining: e.g., digital camera → large SD memory
  - Periodicity analysis
  - Similarity-based analysis
- Other pattern-directed or statistical analyses

Data Warehouse

→A data warehouse is a repository of information collected from multiple, heterogeneous sources and placed in a single site.

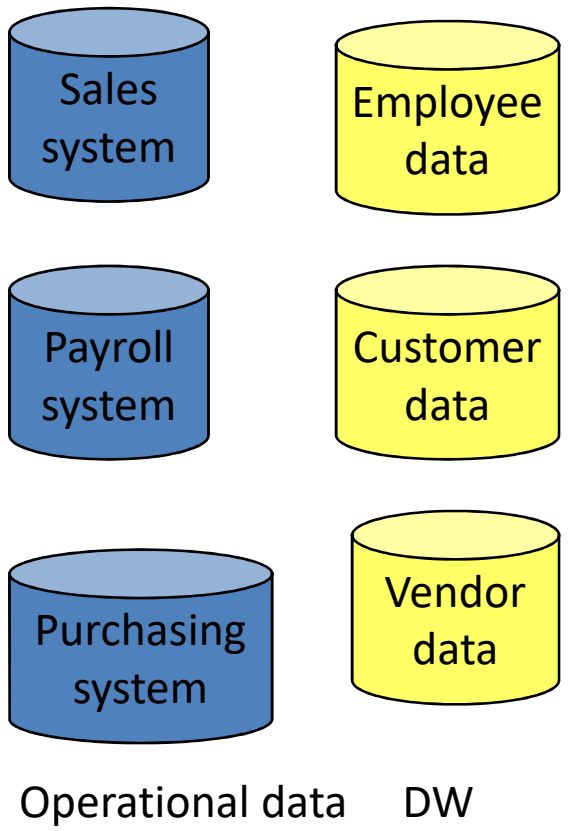
→ A data warehouse is a subject oriented, integrated, time variant and non-volatile collection of data that helps in the management decision making process.

→ Features or Characterstics of data warehouse

a) Subject oriented
b) Integrated
c) Time Variant
d) Non-Volatile

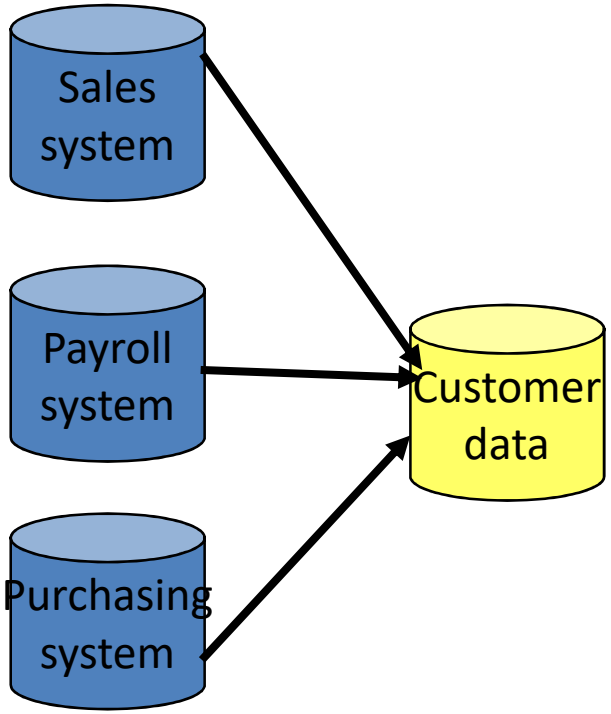
Subject Oriented

- Organized around major subjects, such as customer, product, sales.
- Focusing on the modeling and analysis of data for decision makers, not on daily operations or transaction processing.
- Provide a simple and concise view around particular subject issues by excluding data that are not useful in the decision support process.



Integrated

- Constructed by integrating multiple, heterogeneous data sources
  - relational databases, flat files,
  - on-line transaction records
- Data cleaning and data integration techniques are applied.
  - Ensure consistency in naming conventions,
  - encoding structures, attribute measures, etc.
  - among different data sources
    E.g., Hotel price: currency, tax, breakfast covered, etc.
  - When data is moved to the warehouse, it is converted.

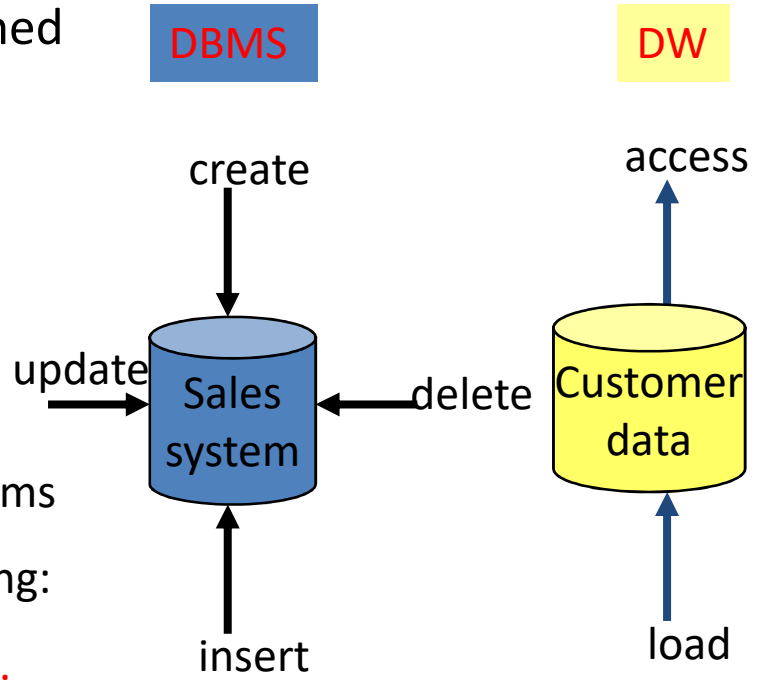


Time Variant

- The time horizon for the data warehouse is significantly longer than that of operational systems.
  - Operational database: current value data.
  - Data warehouse data: provide information from a historical perspective (e.g., past 5-10 years)
- Every key structure in the data warehouse
  - Contains an element of time, explicitly or implicitly
  - But the key of operational data may or may not contain “time element”.

Non volatile

- A physically separate store of data transformed from the operational environment.
- Operational update of data does not occur in the data warehouse environment.
  - Does not require transaction processing, recovery, and concurrency control mechanisms
  - Requires only two operations in data accessing:
    - *initial loading of data* and *access of data*.



Applications of Data warehouse

- Three kinds of data warehouse applications
  - Information processing
    - supports querying, basic statistical analysis, and reporting using crosstabs, tables, charts and graphs
  - Analytical processing
    - multidimensional analysis of data warehouse data
    - supports basic OLAP operations, slice-dice, drilling, pivoting
  - Data mining
    - knowledge discovery from hidden patterns
    - supports associations, constructing analytical models, performing classification and prediction, and presenting the mining results using visualization tools.

Applications of Data Mining

- Data analysis and decision support
  - Market analysis and management
    - Target marketing, customer relationship management (CRM), market basket analysis, cross selling, market segmentation
  - Risk analysis and management
    - Forecasting, customer retention, improved underwriting, quality control, competitive analysis
  - Fraud detection and detection of unusual patterns (outliers)
- Other Applications
  - Text mining (news group, email, documents) and Web mining
  - Stream data mining
  - Bioinformatics and bio-data analysis

Issues in data mining

→In data mining, the algorithm used is complex and data is not available from single sources so these factors also create some issues.

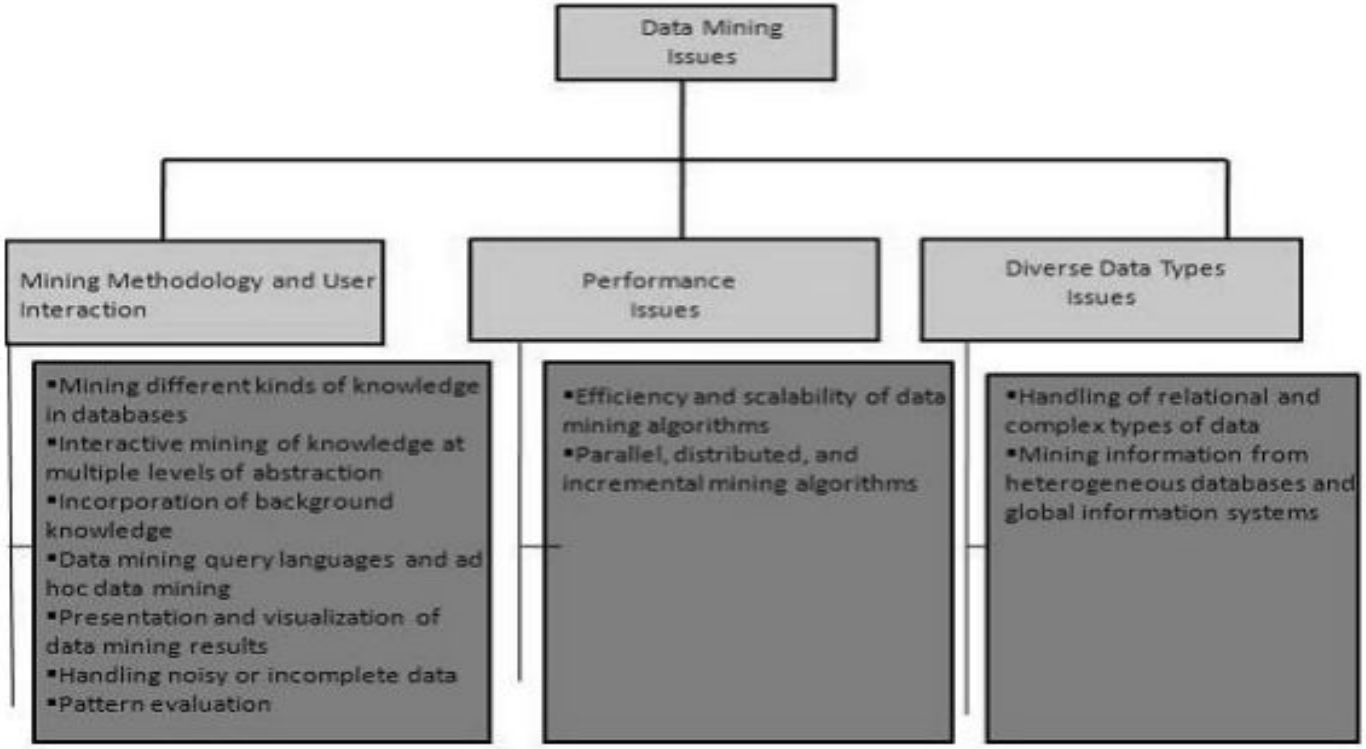
→ The major issues are

1) Mining Methodology and User Interaction

2) Performance Issues

3) Diverse Data Types Issues

Data Mining issues



Mining Methodology and User Interaction Issues

a) **Mining different kinds of knowledge in databases** - Different users may be interested in different kinds of knowledge. Therefore it is necessary for data mining to cover a broad range of knowledge discovery task.

b) **Interactive mining of knowledge at multiple levels of abstraction** - The data mining process needs to be interactive because it allows users to focus the search for patterns, providing and refining data mining requests based on the returned results.

c) **Incorporation of background knowledge** - To guide discovery process and to express the discovered patterns, the background knowledge can be used. Background knowledge may be used to express the discovered patterns not only in concise terms but at multiple levels of abstraction.

d) Data mining query languages and ad hoc data mining - Data Mining Query language that allows the user to describe ad hoc mining tasks, should be integrated with a data warehouse query language and optimized for efficient and flexible data mining.

Mining Methodology and User Interaction Issues

e) **Presentation and visualization of data mining results** - Once the patterns are discovered it needs to be expressed in high level languages,

and visual representations. These representations should be easily understandable.

f) **Handling noisy or incomplete data** - The data cleaning methods are required to handle the noise and incomplete objects while mining the data regularities. If the data cleaning methods are not there then the accuracy of the discovered patterns will be poor.

g) **Pattern evaluation** - The patterns discovered should be interesting because either they represent common knowledge or lack novelty

Performance Issues

a) **Efficiency and scalability of data mining algorithms** - In order to effectively extract the information from huge amount of data in databases, data mining algorithm must be efficient and scalable.

b) **Parallel, distributed, and incremental mining algorithms -** The factors such as huge size of databases, wide distribution of data, and complexity of data mining methods motivate the development of parallel and distributed data mining algorithms. These algorithms divide the data into partitions which is further processed in a parallel fashion. Then the results from the partitions is merged. The incremental algorithms, update databases without mining the data again from scratch.

Diverse Data Types Issues

a) **Handling of relational and complex types of data** - The database may contain complex data objects, multimedia data objects, spatial data, temporal data etc. It is not possible for one system to mine all these kind of data.

b) **Mining information from heterogeneous databases and global information systems** - The data is available at different data sources on LAN or WAN. These data source may be structured, semi structured or unstructured. Therefore mining the knowledge from them adds challenges to data mining.

END OF UNIT 1

Data Warehousing & Data Mining

BSC.CSIT, 8th Sem

HCOE

Unit: 2

Operational Database(OLTP) VS Data Warehouse (OLAP)

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

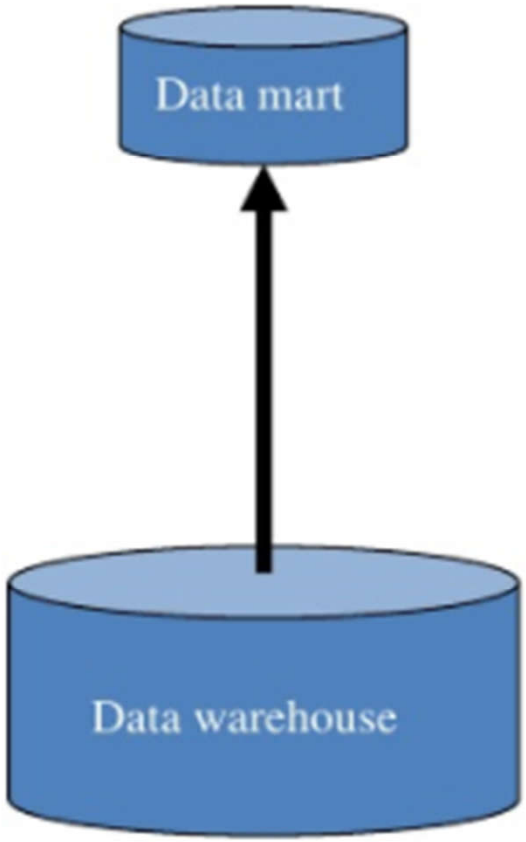
Data Marts

- A data mart is a scaled down version of a data warehouse that focuses on a particular subject area.
- A **data mart** is a subset of an organizational data store, usually oriented to a specific purpose or major data subject, that may be distributed to support business needs.
- Data marts are analytical data stores designed to focus on specific business functions for a specific community within an organization.
- Usually designed to support the unique business requirements of a specified department or business process
- Implemented as the first step in proving the usefulness of the technologies to solve business problems

Reasons for creating a data mart

- Easy access to frequently needed data
- Creates collective view by a group of users
- Improves end-user response time
- Ease of creation in less time
- Lower cost than implementing a full Data warehouse
- Potential users are more clearly defined than in a full Data warehouse

Characteristics of the Departmental Data Mart



- Small
- Flexible
- Customized by Department
- OLAP
- Source is departmentally structured data warehouse

Data Mart	Data Warehouse
Sometimes holds only one subject area	Holds several subject areas
Summarized data	Detailed information
Is easy to build	Is difficult to build
Department-wide data	Enterprise-wide data

"Hello, I'm a data warehouse."

"And I'm a data mart."

Metadata

→Metadata is data about data which helps to make the data findable and understandable.

→ Metadata is necessary for using, building and administrating data warehouse.

→The metadata can be

a) **Descriptive**- information about the content and context of the data.

b) **Structural-** information about the structure of the data.

c) **Administrative-** information about the file types, rights and management and preservation processes.

Why use Metadata?

The comprehensive metadata will:

1) Facilitate data discovery.
2) Helps users determine the applicability of the data.
3) Enable interpretation and reuse.
4) Allow any limitation to be understood.
5) Clarify ownership and restrictions on reuse.
6) provide interoperability.

Multidimensional Data Model

→A data warehouse is usually modeled by a multi-dimensional data structure called **data cube** which allows data to be modeled and viewed in a multiple dimension.

→Each data cube consists of **dimensions** which corresponds to an attribute or a set of attributes in the schema and a **Cell** stores the value of some aggregated measure.

→ The multidimensional data model consists of two types of table

Multidimensional data model contd…

a) Dimension Table

→ Consists of tuple of attributes of dimension.

→ It is simple primary key.

b) Fact Table

→ A fact table has a tuples, one per a recorded fact.

→ It is compound primary key.

Schemas for Multidimensional data model: star , snowflake and fact constellation or Data warehouse Schema

→ The most popular data model for a data warehouse is a multi dimensional data model which can exist in the form of a star schema, a snowflake schema or a fact constellation schema.

Star Schema

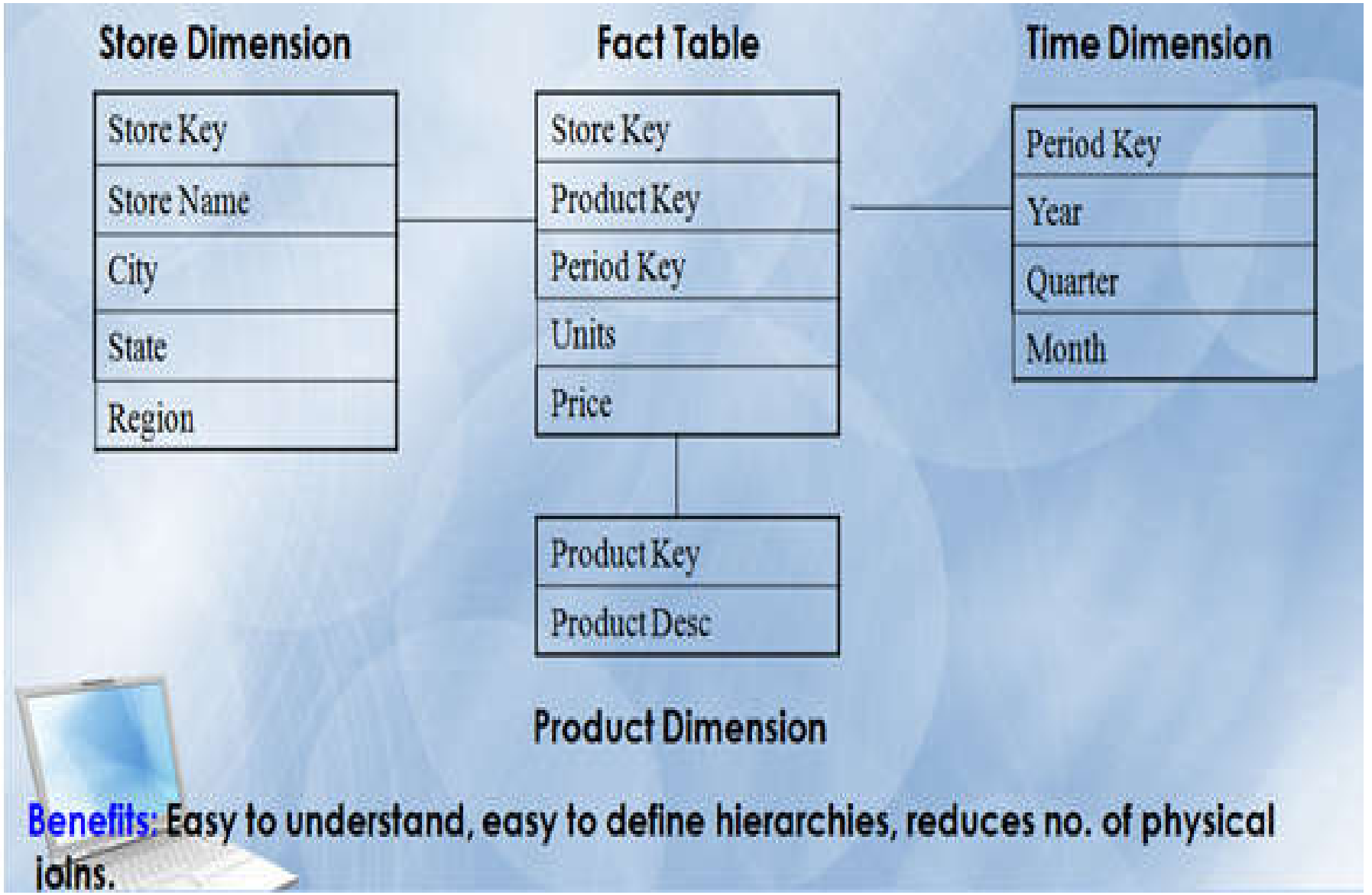
→ Star schema is the most common modeling paradigm in which a data warehouse contains

- a large central table called fact table which contains a large amount of data with no redundancy.
- a set of smaller table called dimension table which is one for each dimension.

→ Every fact table points to one tuple in each of the dimensions and also has additional attributes.

→ The dimension table is displayed in a radial pattern around the central fact table.

→ Example:


Store Dimension
Store Key
Store Name
City
State
Region
Fact Table
Store Key
Product Key
Period Key
Units
Price
Time Dimension
Period Key
Year
Quarter
Month
Product Key
Product Desc
Product Dimension
Benefits: Easy to understand, easy to define hierarchies, reduces no. of physical joins.

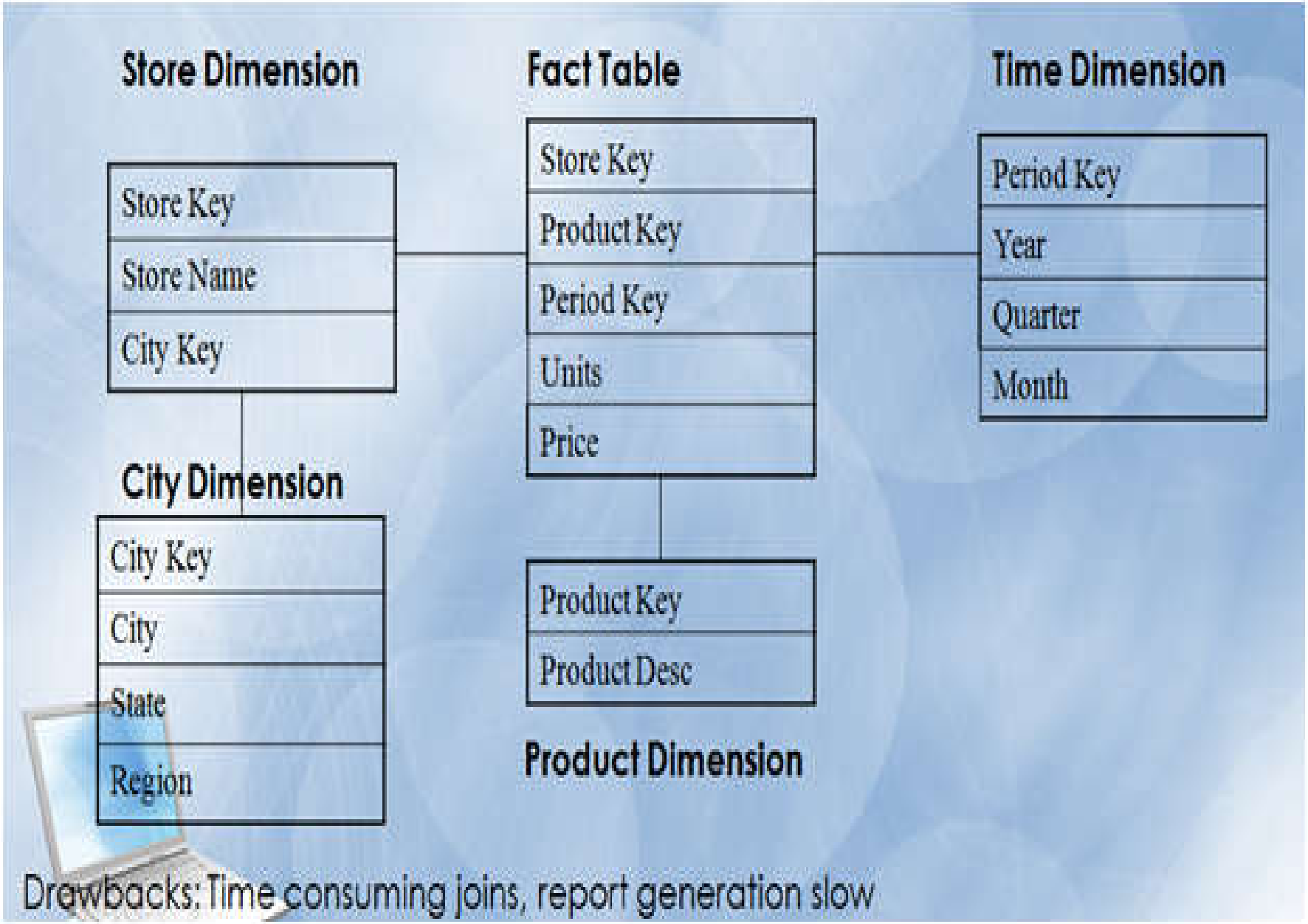
Snowflake Schema

→A snowflake schema is a variant of star schema model where some dimension tables are normalized and further splits data into additional tables

→A single, large and central fact table and one or more tables for each dimension in snowflake.

→ Snowflake schema is kept in normalized form to reduce redundancies and such a table is easy to maintain and saves storage space.

→Exmaple:


Store Dimension
Store Key
Store Name
City Key
City Dimension
City Key
City
State
Region
Fact Table
Store Key
Product Key
Period Key
Units
Price
Product Key
Product Desc
Product Dimension
Time Dimension
Period Key
Year
Quarter
Month
Drawbacks: Time consuming joins, report generation slow

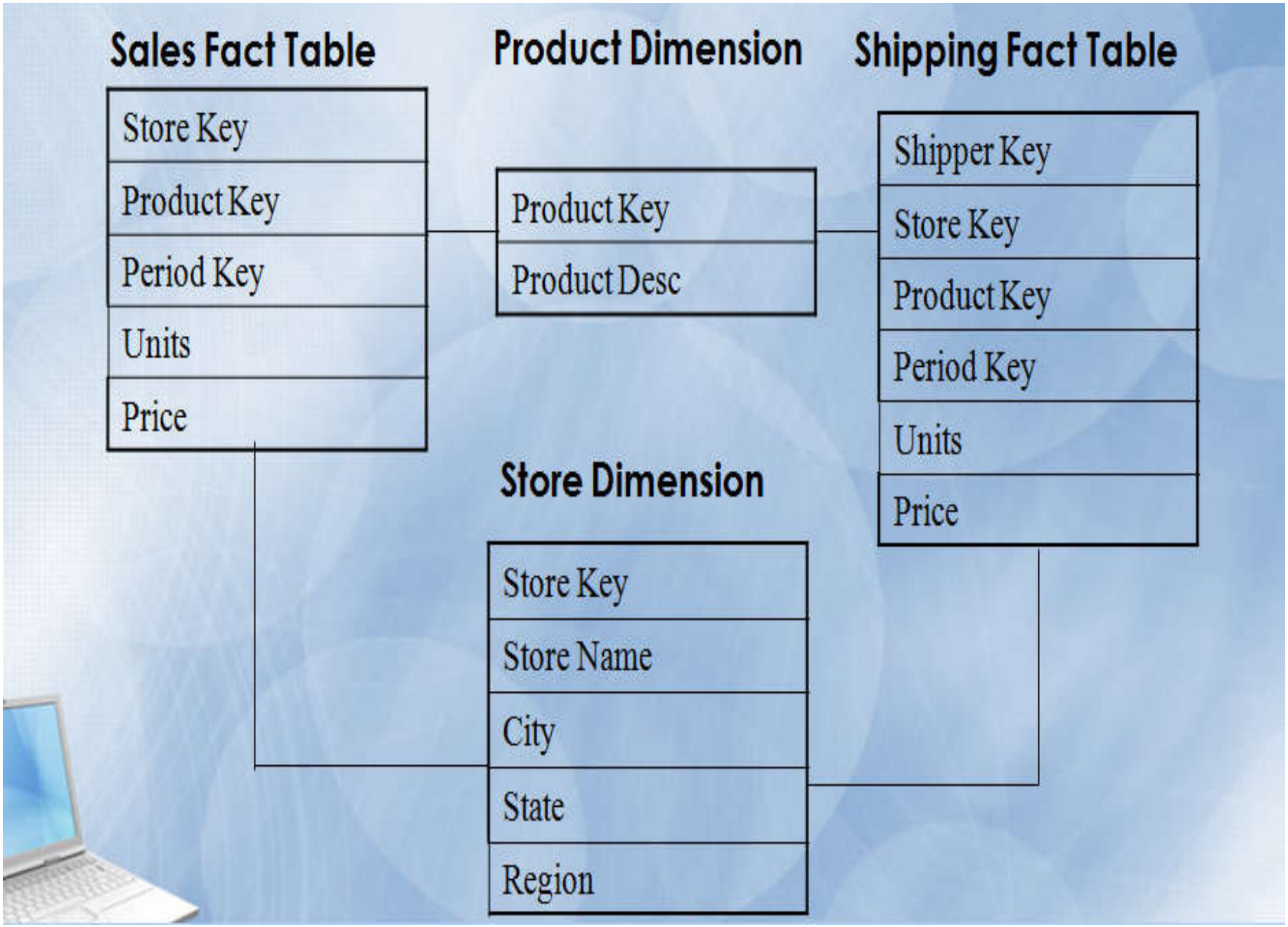
Fact Constellation Schema

→In fact constellation schema, multiple fact table share dimension tables.

→This schema is viewed as collection of stars hence called galaxy schema or fact constellation.

→Sophisticated application requires such schema.

→ Example:


Sales Fact Table
Store Key
Product Key
Period Key
Units
Price
Product Dimension
Product Key
Product Desc
Shipping Fact Table
Shipper Key
Store Key
Product Key
Period Key
Units
Price
Store Dimension
Store Key
Store Name
City
State
Region

Difference Between Star Schema and Snowflake Schema

S.No	Star Schema	Snowflake Schema
1	**Data Structure:** De-Normalized Data Structure	**Data Structure:** Normalized Data Structure
2	**Dimension:** Category wise Single Dimension Table	**Dimension:** Dimension table split into many pieces
3	**Data dependency & redundancy:** More data dependency and redundancy	**Data dependency & redundancy:** Less data dependency and No redundancy
4	**Join:** No need to use complicated join	**Join:** Complicated Join
5	**Query Result:** Query Results Faster	**Query Result:** Some delay in Query Processing
6	**Parent Table:** No Parent Table	**Parent Table:** It may contain Parent Table
7	**DB Structure:** Simple DB Structure	**DB Structure:** Complicated DB Structure

Snowflake schema	Fact constellation schema
A single, large and central fact table and one or more tables for each dimension.	Multiple fact tables share dimension tables.
Snowflake schema contain one star scheme at a time	Fact constellation schema contain multiple star scheme
It's not more complex then fact constellation because it contains single fact tables.	Constellation schema is more complex than star or snowflake, which is because it contains multiple fact tables.
Tables are easier to maintain.	Difficult to maintain due to multiple fact tables.
Saves the storage space.	Not save the space due to multiple fact table.
Normalize form of start schema.	Normalize for of start or snow flake schema.
Easy to Navigate between the tables due to less number of joins.	Difficult to Navigate between the tables due to maximum number of joins.
Snowflake schema can be defined using DMQL as follows:	Fact constellation schema can be defined using DMQL as follows:
define cube sales snowflake [time, item, branch, location]: dollars sold = sum(sales in dollars), units sold = count(*)	define cube sales [time, item, branch, location]: dollars sold = sum(sales in dollars), units sold = count(*)
Simple and complex query use to access data from database	More complex query use to access data from database

Class Work-1

Suppose that a data warehouse consists of the three dimensions *time*, *doctor*, and *patient*, and the two measures *count* and *charge*, where *charge* is the fee that a doctor charges a patient for a visit.

(a) Enumerate three classes of schemas that are popularly used for modeling data warehouses.

(b) Draw a schema diagram for the above data warehouse using one of the schema classes listed in (a).

END of UNIT-2

Data Warehousing & Data Mining

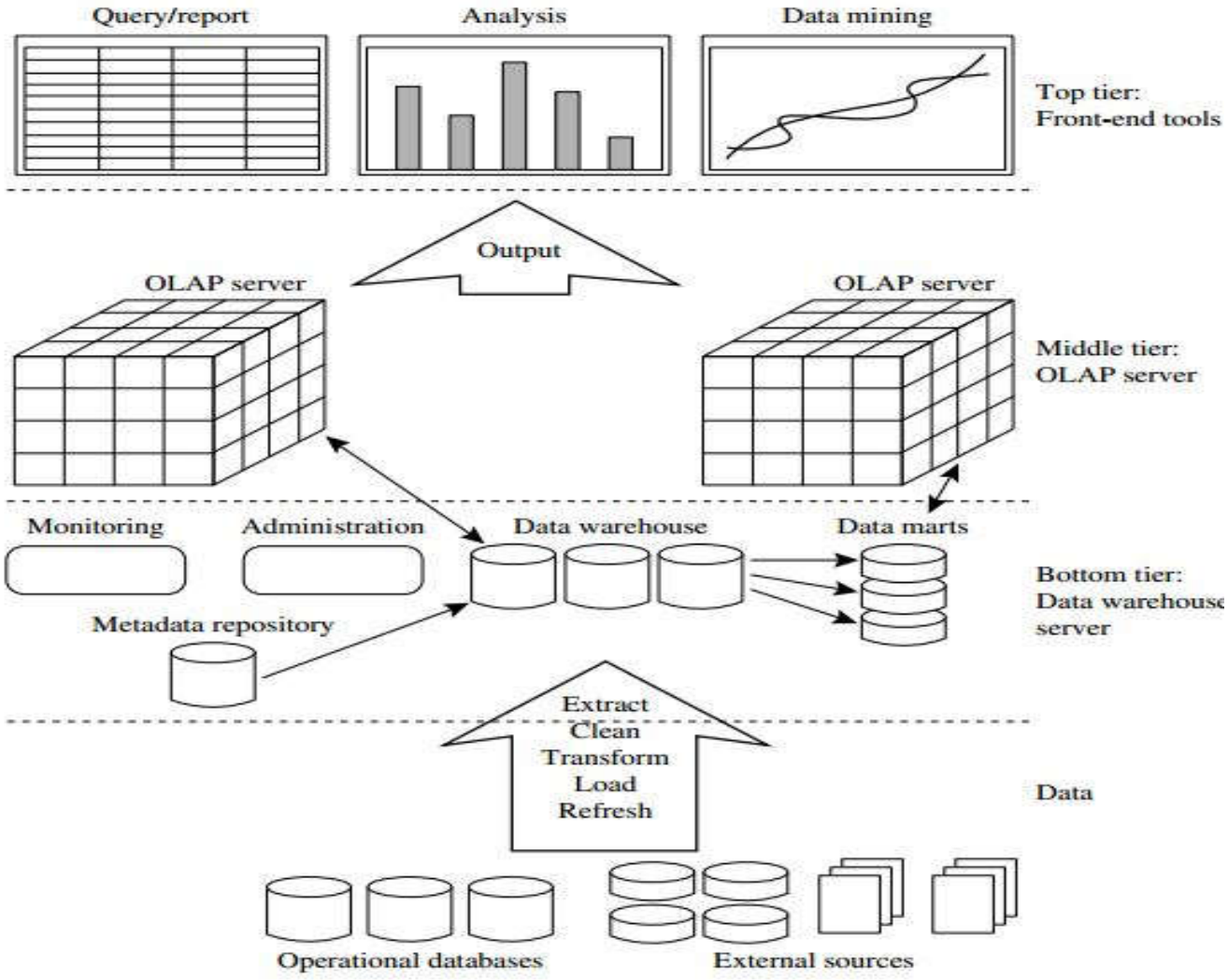
BSC.CSIT, 8th Sem

HCOE

Unit: 3

Compiled by: Narayan Dhamala

Data warehouse Architecture -A Multi-tired Architecture



DW 3-tier architecture contd..

1) Bottom tier:-The bottom tier is a warehouse database server that is always a relational database system.

- Back-end tools and utilities are used to feed data into the bottom tier from operational databases or other external sources. These tools and utilities perform data extraction,cleaning and transformation as well as load and refresh functions to update the data warehouse.
- The date extracted using application program interfaces known as gateways.
- Example of gateways are ODBC(open database connection)and OLEDB(Open Linking and embedding for database) by microsoft and jdbc(java database connecton).
- This tier also contains a metadata repository, which stores information about the data warehouse and its contents.

DW 3-tier architecture contd..

2.)Middle tier:- The middle tier is an OLAP server that is typically implemented using either:-

a) A relational OLAP (ROLAP) model that is,an extended relation DBMS that maps operations.Intermediate server b/w relational back-end server and client front end tools.

b) A multidimentional OLAP (MOLAP) model that is, a special purpose server that directly implements multidimentional data and operations. Supports multidimention views.

DW 3-tier architecture contd..

3.)Top tier:-The top tier is a front –end client layer ,which contains query and reporting tools ,analysis tools,and or data mining tools.

- Note:-

OLAP – Online Analytical Processing:

- This is the major task of Data Warehousing System.
- Useful for complex data analysis and decision making.
- Market oriented –used by managers,executives and data analyst.

Data warehouse back end tools and utilities

→ The DW uses back end tools and utilities to populate and refresh data.

→ The back end tools and utilities perform the following functions

a) Data extraction- gathers data from multiple, heterogeneous and external sources.

b) Data cleaning- Detect errors in data and correct them when possible.

c) Data transformation- converts data from legacy or host format to warehouse format.

d) Load- which sorts, summarizes ,checks integrity, and builds indices and partitions.

e) Refresh- which involves updating from data sources to the warehouse.

--

Why a Data Warehouse is Separated from Operational Databases

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

- An operational database is constructed for well-known tasks and workloads such as searching particular records, indexing, etc. In contrast, data warehouse queries are often complex and they present a general form of data.
- Operational databases support concurrent processing of multiple transactions. Concurrency control and recovery mechanisms are required for operational databases to ensure robustness and consistency of the database.
- An operational database query allows to read and modify operations, while an OLAP query needs only **read only** access of stored data.
- An operational database maintains current data. On the other hand, a data warehouse maintains historical data.

Data Warehouse(DW) Models:
Enterprise DW, Metadata, Virtual DW, Distributed DW

Enterprise Warehouse

- An enterprise warehouse collects all the information and the subjects spanning an entire organization.
- It provides enterprise wide data integration.
- The data is integrated from operational systems and external information providers.
- This information can vary form a few gigabytes to hundreds of gigabytes, terabytes or beyond.
- It may be implemented on traditional mainframes, super server computer or parallel architecture platform.
- Enterprise warehouse requires extensive business modeling and may takes years to design and build.

Data Warehouse(DW) Models:
Enterprise DW, Metadata, Virtual DW, Distributed DW

Virtual Warehouse

- When end-users access the “system of record” (the OLTP system) directly and generate “summarized data” reports and thereby given the feel of a “data warehouse”, such a data warehouse is known as a “**Virtual data warehouse**”.
- The virtual data warehouse is a subject oriented, current valued and detailed only collection of data in support of organization need for up to the second operational information.
- The view over an operational data warehouse is known as a virtual warehouse.
- Virtual warehouse is easy to build and building virtual warehouse requires excess capacity on operational database servers.
- Virtual warehouse collects data from a wide variety of sources like database (essentially business data base), data mart etc..

Advantages and Dis-advantages of virtual data warehosue

Advantages

- provides end-users with the most current corporate information.
- No data redundancy.
- No need for additional hardware to handle the analysis processing.

Dis-advantages

- The data in virtual data warehouse may be inconsistent or incomplete as it is derived directly from operational databases without any kind of integration.
- End user access times are unpredictable. They could request meaningless queries or scan all the data as they have complete access to entire detailed operational databases.
- The operational database is frequently not in the form of decision support system end user needs because it is designed and tuned to support OLTP operations rather than OLAP.

Distributed Warehouse

- In distributed warehouse, the components are distributed across several physical databases.

Advantages

- **Local autonomy:** Each site in a distributed data warehouse is autonomous. The local warehouse data is locally owned, managed and controlled even if it is accessible from other remote sites.
- **Performance:** Data in a distributed warehouse can be stored close to its normal point of use which reduces response time and communication cost.
- **Reliability and Availability:** Reliability means the system can continue to function despite failure of one or more sites. If a distributed data warehouse supports replicated data at more than one sites , a crash or failure of communication link at one or more of the sites does not necessarily make the warehouse data inaccessible.

Dis-advantages of Distributed data warehouse

- **Security:** A distributed data warehouse utilizes a network which introduces a weak security link.
- **Complexity:** Since data are stored at different sites access and management is more challenging.
- **Cost:** Each site must have people to maintain the system.

Data Warehouse Manager

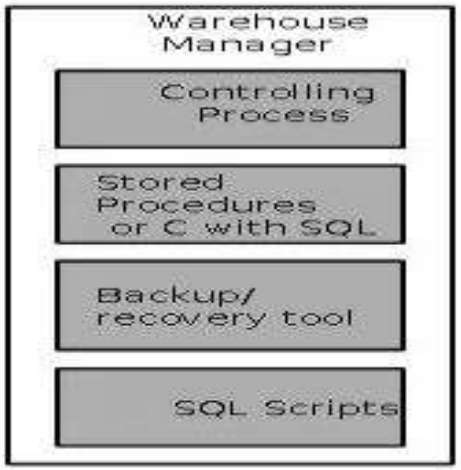
A warehouse manager is responsible for the warehouse management process. It consists of third-party system software, C programs, and shell scripts.

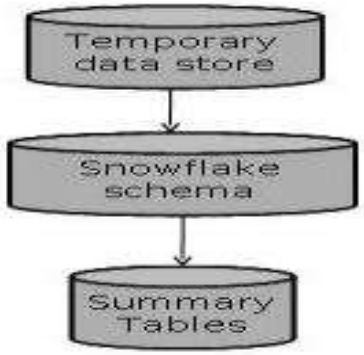
The size and complexity of warehouse managers varies between specific solutions.

Warehouse Manager Architecture

A warehouse manager includes the following:

- The controlling process
- Stored procedures or C with SQL
- Backup/Recovery tool
- SQL Scripts





Operations Performed by Warehouse Manager

- A warehouse manager analyzes the data to perform consistency and referential integrity checks.
- Creates indexes, business views, partition views against the base data.
- Generates new aggregations and updates existing aggregations.
- Generates normalizations.
- Transforms and merges the source data into the published data warehouse.
- Backup the data in the data warehouse.

OLAP Server and It's types

- OLAP server is based on multi-dimensional data model which allows fast, consistent and interactive access to information.
- The 3 different types of OLAP server are

a) **ROLAP (Relational OLAP)**

b) **MOLAP (Multidimensional OLAP)** and

c) **HOLAP (Hybrid OLAP)**

ROLAP server

- The OLAP server which is based on relational database system is called ROLAP server.

Advantages of ROLAP Server

- ROLAP servers can be easily used with existing RDBMS.
- Data can be stored efficiently, since no zero facts can be stored.
- ROLAP server offers highly scalability.
- ROLAP tools do not use pre-calculated data cubes.

Disadvantages of ROLAP Server

- Poor query performance.
- Some limitations of scalability depending on the technology architecture that is utilized.

MOLAP Server

- The OLAP Server which is based on multidimensional database system is called MOLAP Server.

Advantages of MOLAP Server

- MOLAP allows fastest indexing to the pre-computed summarized data.
- Helps the users connected to a network who need to analyze larger, less-defined data.
- Easier to use, therefore MOLAP is suitable for inexperienced users.

Disadvantages of MOLAP Server

- MOLAP are not capable of containing detailed data.
- The storage utilization may be low if the data set is sparse.

HOLAP Server

- The OLAP server which is the combination of ROLAP and MOLAP Server is called HOLAP server.
- HOLAP server combines the benefits of higher scalability of ROLAP and faster computation of MOLAP.
- HOLAP server allows to store large data volumes of detailed information.

MOLAP vs ROLAP

MOLAP	ROLAP
Information retrieval is fast.	Information retrieval is comparatively slow.
Uses sparse array to store datasets.	Uses relational table.
MOLAP is best suited for inexperienced users, since it is very easy to use.	ROLAP is best suited for experienced users.
Maintains a separate database for data cubes.	It may not require space other than available in the data warehouse.
DBMS facility is weak.	DBMS facility is strong.

End of Unit 3

Data Warehousing & Data Mining

BSC.CSIT, 8th Sem

HCOE

Unit: 4

Data Cube

→ A data cube is a multidimensional data model which allows data to be modeled and viewed in a multiple dimension.

→ Data cube is defined by dimensions and facts. The dimensions are the entities with respect to which an enterprise preserve the records.

→ Suppose a company wants to keep track of sales records with the help of sales data warehouse with respect to time, item, branch, and location. These dimensions allow to keep track of monthly sales and at which branch the items were sold. There is a table associated with each dimension. This table is known as dimension table. For example, "item" dimension table may have attributes such as item_name, item_type, and item_brand.

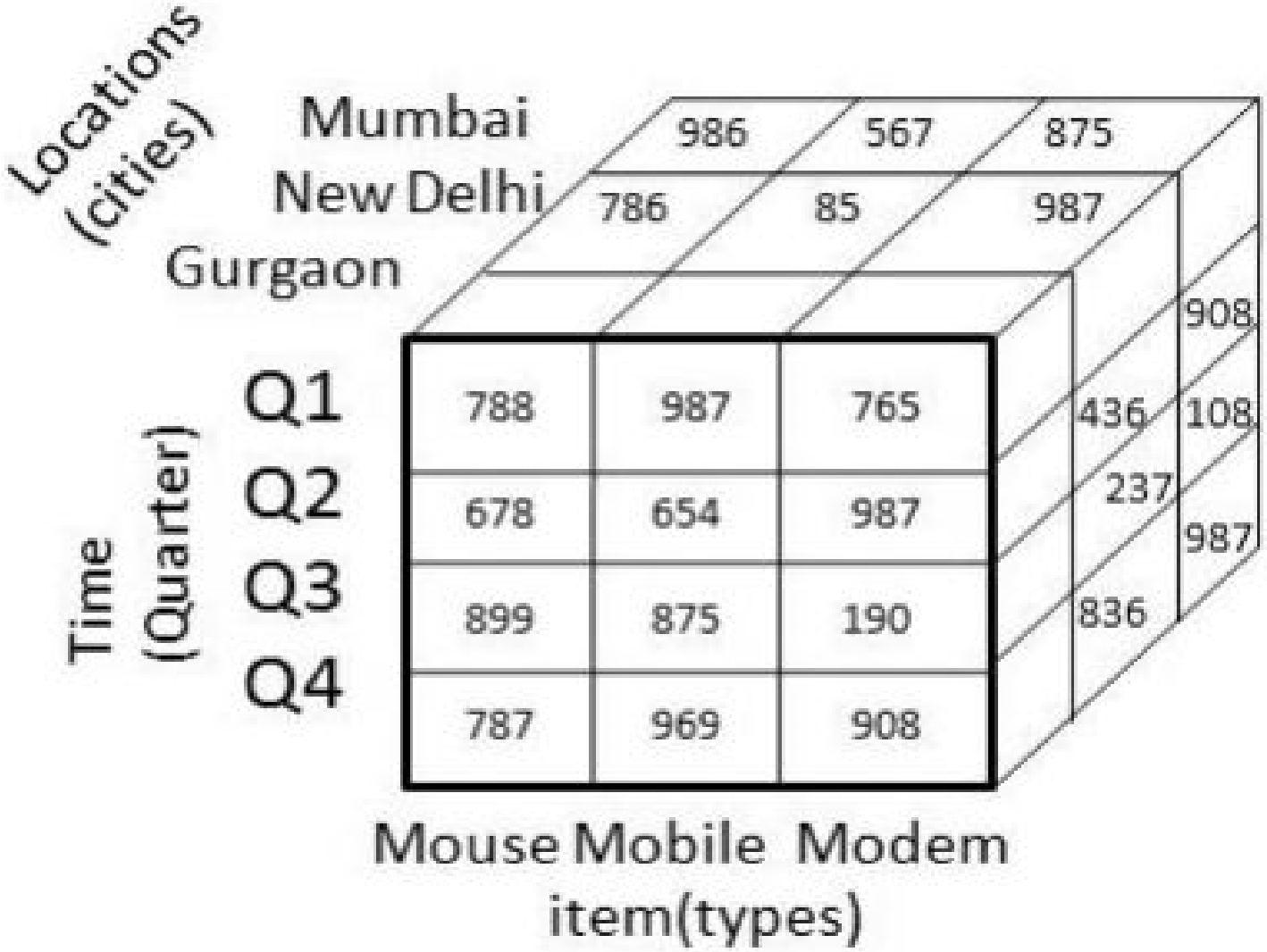
The following table represents the 2-D view of Sales Data for a company with respect to time, item, and location dimensions.

Location="New Delhi"				
Time(quarter)	Item(type)			
	Entertainment	Keyboard	Mobile	Locks
Q1	500	700	10	300
Q2	769	765	30	476
Q3	987	489	18	659
Q4	666	976	40	539

But here in this 2-D table, we have records with respect to time and item only. The sales for New Delhi are shown with respect to time, and item dimensions according to type of items sold. If we want to view the sales data with one more dimension, say, the location dimension, then the 3-D view would be useful. The 3-D view of the sales data with respect to time, item, and location is shown in the table below:

Time	Location=" Gurgaon"			Location="New Delhi"			Location="Mumbai"		
	Item			Item			Item		
	Mouse	Mobile	Modem	Mouse	Mobile	Modem	Mouse	Mobile	Modem
Q1	788	987	765	786	85	987	986	567	875
Q2	678	654	987	659	786	436	980	876	908
Q3	899	875	190	983	909	237	987	100	1089
Q4	787	969	908	537	567	836	837	926	987

→ The above 3-D table can be represented as 3-D data cube as shown in the following figure:


Locations (cities)
Mumbai
New Delhi
Gurgaon
986
567
875
786
85
987
908
436
108
237
987
836
Time (Quarter)
Q1
Q2
Q3
Q4
788
987
765
678
654
987
899
875
190
787
969
908
Mouse
Mobile
Modem
item(types)

OLAP Operations

→The following OLAP operations are used in multidimensional data.

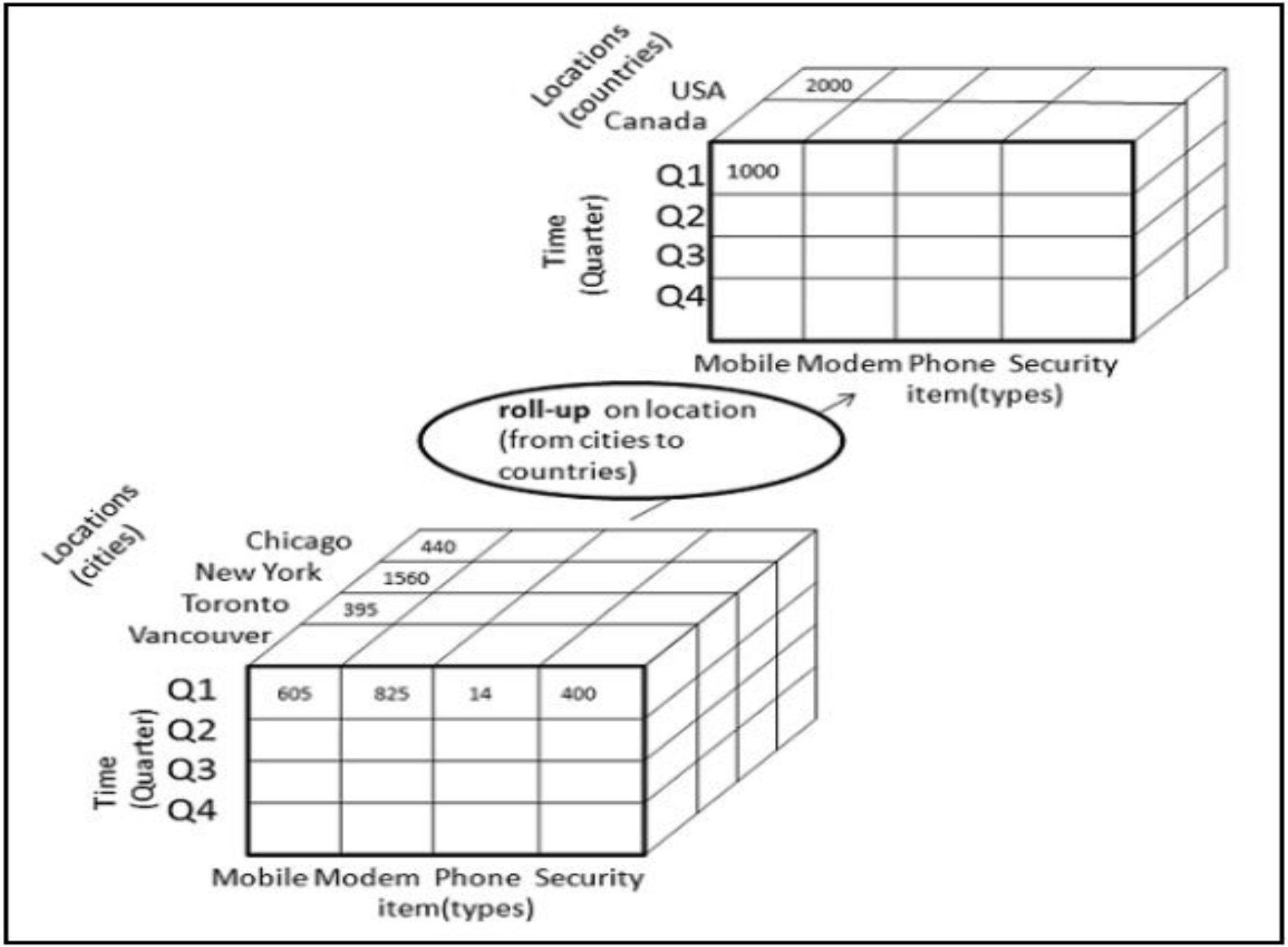
a) Roll up (Drill up)
b) Roll down (Drill down)
c) Slice
d) Dice
e) Pivot (Rotation)

Roll up

- The roll up operation performs aggregation operations on the data cube either

1) By climbing the concept hierarchy for a dimension or
2) By dimension reduction

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


Locations (countries)
USA
Canada
2000
Q1
Q2
Q3
Q4
1000
Time (Quarter)
Mobile Modem Phone Security
item(types)
roll-up on location (from cities to countries)
Locations (cities)
Chicago
New York
Toronto
Vancouver
440
1560
395
Q1
Q2
Q3
Q4
605
825
14
400
Time (Quarter)
Mobile Modem Phone Security
item(types)

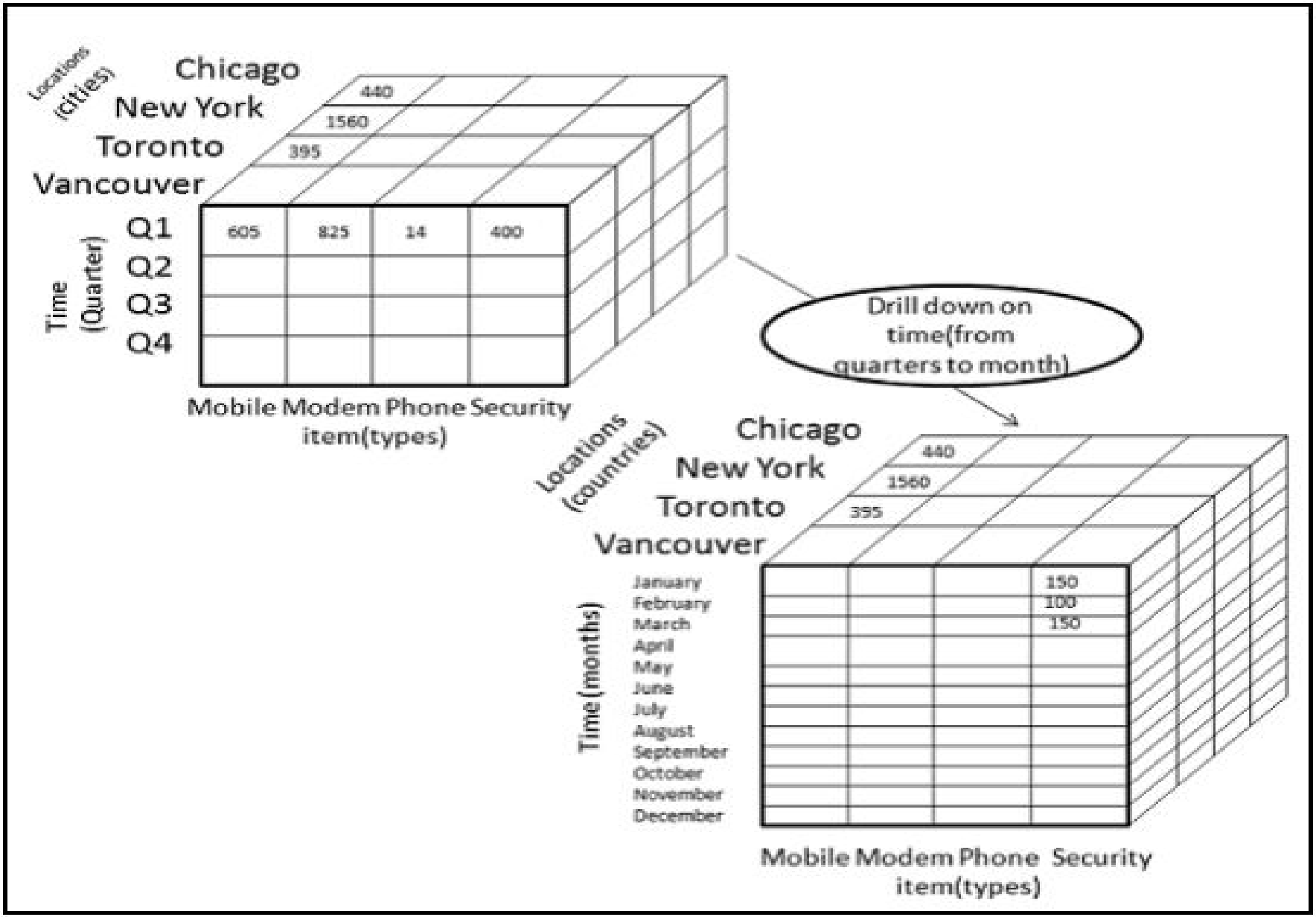
Roll down

- Drill-down is the reverse operation of roll-up. It is performed by either

  1) By stepping down a concept hierarchy for a dimension or

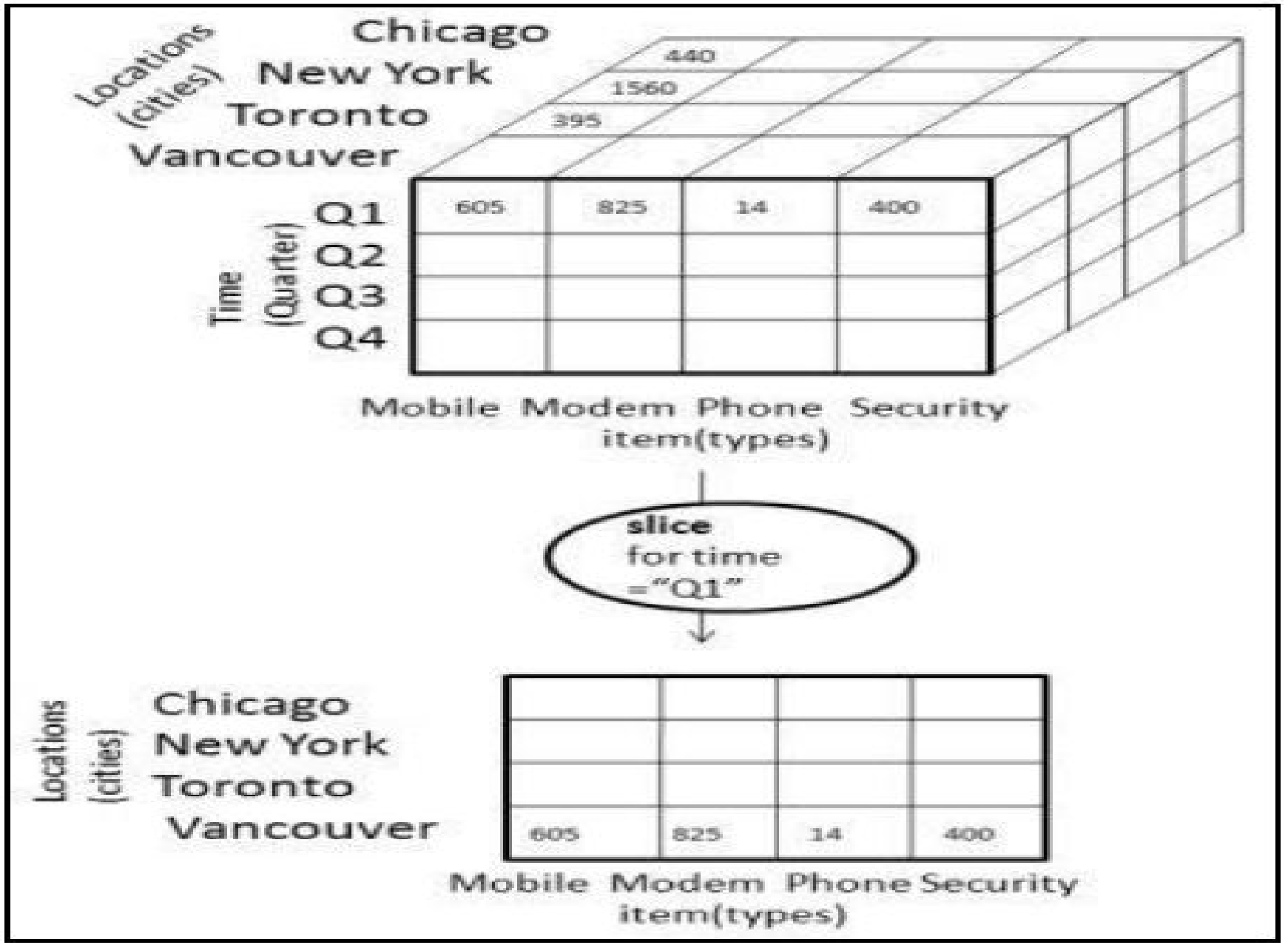
  2) By introducing a new dimension.

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


Locations (cities)
Chicago
New York
Toronto
Vancouver
440
1560
395
Time (Quarter)
Q1
Q2
Q3
Q4
605
825
14
400
Mobile Modem Phone Security
item(types)
Drill down on time(from quarters to month)
Locations (countries)
Chicago
New York
Toronto
Vancouver
440
1560
395
Time (months)
January
February
March
April
May
June
July
August
September
October
November
December
150
100
150
Mobile Modem Phone Security
item(types)

Slice

- The slice operation perform selection operation on one particular dimension from a given cube and provides a new sub cube.
- The Slice operation performed for the dimension "time" using the criterion time = "Q1“ is as follows.


Locations (cities)
Chicago
New York
Toronto
Vancouver
440
1560
395
Time (Quarter)
Q1
Q2
Q3
Q4
605
825
14
400
Mobile Modem Phone Security
item(types)
slice
for time
="Q1"
Locations (cities)
Chicago
New York
Toronto
Vancouver
605
825
14
400
Mobile Modem Phone Security
item(types)

Dice

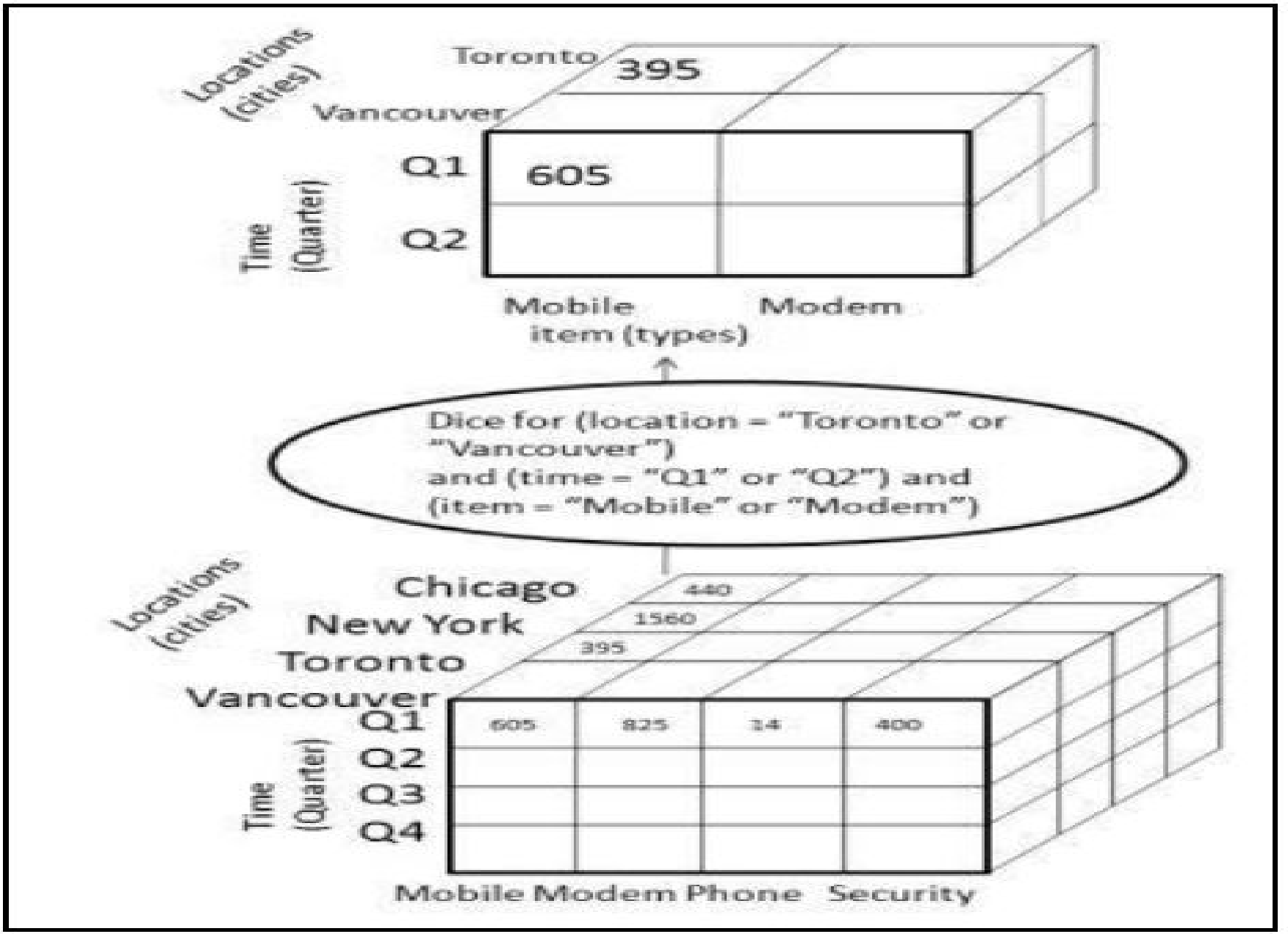
- The dice operation perform selection operation on two or more dimensions from a given cube and provides a new sub cube.
- The dice operation performs on the cube based on the following three dimensions.

  (location = "Toronto" or "Vancouver")

  (time = "Q1" or "Q2") and

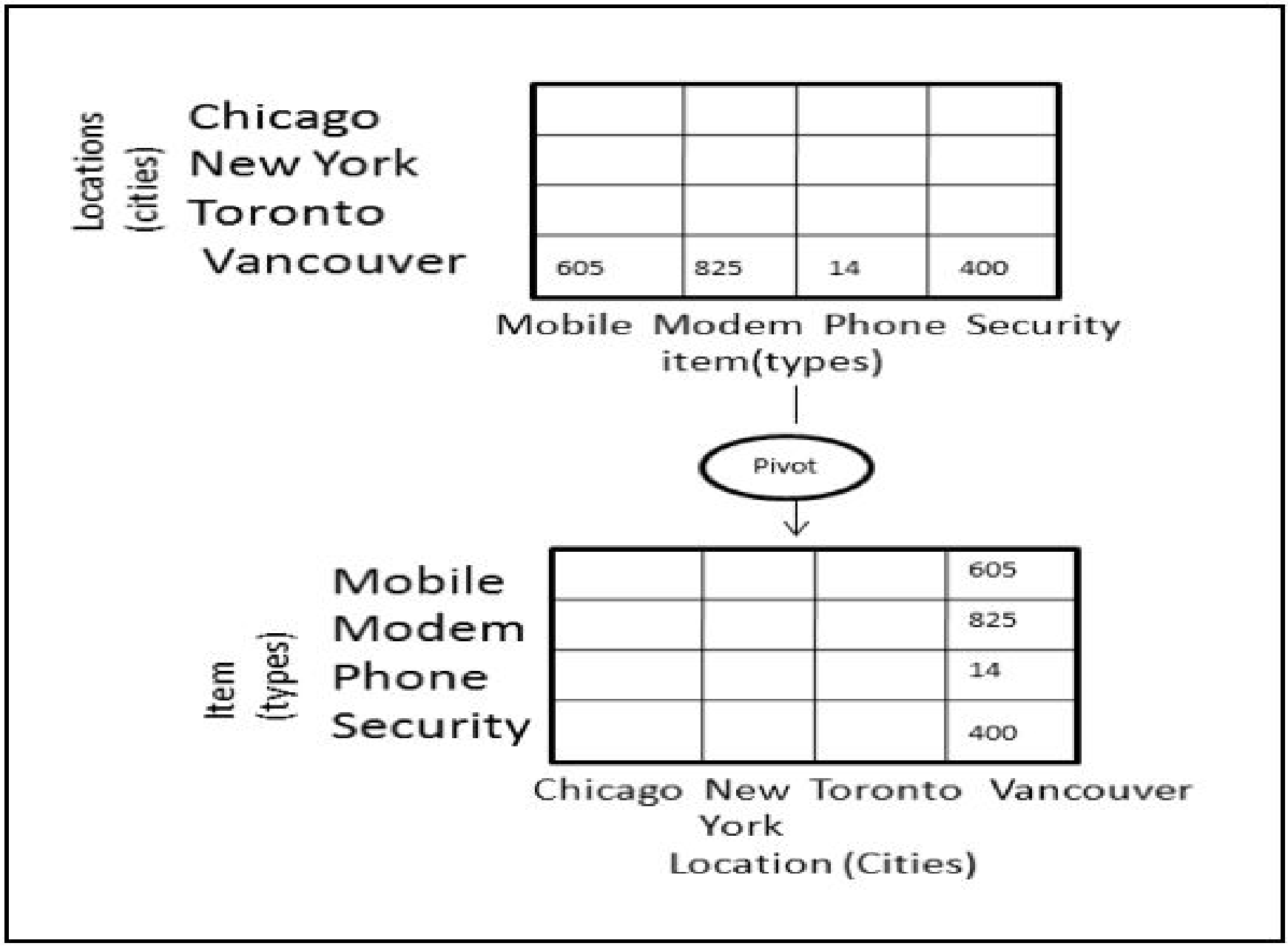
  (item =" Mobile" or "Modem")

  are as follows


Locations (cities)
Toronto
395
Vancouver
Q1
605
Q2
Time (Quarter)
Mobile
Modem
item (types)
Dice for (location = "Toronto" or "Vancouver") and (time = "Q1" or "Q2") and (item = "Mobile" or "Modem")
Locations (cities)
Chicago
440
New York
1560
Toronto
395
Vancouver
Q1
605
825
14
400
Q2
Q3
Q4
Time (Quarter)
Mobile
Modem
Phone
Security

Pivot(Rotation)

- The pivot operation is also known as rotation.
- The pivot operation rotates the data axes in view to provide an alternative presentation of the data.


Locations (cities)
Chicago
New York
Toronto
Vancouver
605
825
14
400
Mobile Modem Phone Security
item(types)
Pivot
Item (types)
Mobile
Modem
Phone
Security
605
825
14
400
Chicago New York Toronto Vancouver
Location (Cities)

OLAP data and queries

OLAP 101 – Queries example

Date	Country	City	OS	Browser	Sale
2012-05-21	USA	NY	Windows	FF	0.0
2012-05-21	USA	NY	Windows	FF	10.0
2012-05-22	USA	SF	OSX	Chrome	25.0
2012-05-22	Canada	Ontario	Linux	Chrome	0.0
2012-05-23	USA	Chicago	OSX	Safari	15.0
5 visits, 3 days	2 countries USA: 4 Canada: 1	4 cities: NY: 2 SF: 1	3 OS-es Win: 2 OSX: 2	3 browsers FF: 2 Chrome:2	50.0 3 sales

OLAP 101 – Queries example

- **Rolling up** to country level:

```
SELECT COUNT(visits), SUM(sales)
GROUP BY country
```

Country	visits	sales
USA	4	$50
Canada	1	0

- **"Slice"** by browser

```
SELECT COUNT(visits), SUM(sales)
GROUP BY country
HAVING browser = "FF"
```

Country	visits	sales
USA	2	$10
Canada	0	0

- **Top** browsers by sales

```
SELECT SUM(sales), COUNT(visits)
GROUP BY browser
ORDER BY sales
```

Browser	sales	visits
Chrome	$25	2
Safari	$15	1
FF	$10	2

Computation of Data Cube

Why Data cube computation is needed?

→To retrieve the information from the data cube in the most efficient way possible.

→ The queries run on the cube will be fast.

The Full cube

- The multi way array aggregation method computes full data cube by using a multidimensional array as its basic data structure
  1. Partition array into the chunks
  2. Compute aggregate by visiting (i.e. accessing the values at) cube cells

Advantage

the queries run on the cube will be very fast.

Disadvantage

pre-computed cube requires a lot of memory.

An Iceberg-Cube

- contains only those cells of the data cube that meet an aggregate condition.
- It is called an Iceberg-Cube because it contains only some of the cells of the full cube, like the tip of an iceberg.
- The purpose of the Iceberg-Cube is to identify and compute only those values that will most likely be required for decision support queries.
- The aggregate condition specifies which cube values are more meaningful and should therefore be stored.
- This is one solution to the problem of computing versus storing data cubes.

Advantage:

pre-compute only those cells in the cube which will most likely be used for decision support queries.

A Closed Cube

A closed cube is a data cube consisting of only closed cells

Shell Cube

we can choose to precompute only portions or fragments of the cube shell, based on cuboids of interest.

General strategies for data cube computation

1. Sorting hashing and grouping
2. Simultaneous aggregation and caching intermediate results
3. Aggregation from smallest child when there exist multiple child cuboid
4. The Apriori pruning method can be explored to compute iceberg cube efficiently

1. Sorting, hashing and grouping.

These operations facilitate aggregation, i.e. computation of the cells that share the same set of dimension values.

These techniques can also perform:

- shared-sorts: sharing sorting costs across multiple cuboids
- share-partitions: sharing partitioning costs across multiple cuboids

Example:

To compute total sales by branch, day, and item, it is more efficient to sort tuples or cells by branch, and then by day, and then group them according to the item name.

2. Simultaneous aggregation and caching intermediate results.

Reduce expensive disk I/O operations by computing higher-level group-bys from computed lower-level group-bys.

These techniques can also perform:

- Amortized-scans: computing as many cuboids as possible at the same time to reduce disk reads

Example:
To compute sales by branch, we can use the intermediate results derived from the computation of a lower-level cuboid, such as sales by branch and day.

3. Aggregation from the smallest child.

If a parent 'cuboid' has more than one child, it is efficient to compute it from the smallest previously computed child 'cuboid'.

Example:

To compute a sales cuboid, Cbranch, when there exist two previously computed cuboids, C{branch,year} and C{branch,tem}, it is obviously more efficient to compute Cbranch from the former than from the latter if there are many more distinct items than distinct years.

4. The Apriori pruning method can be explored to compute iceberg cube efficiently

The Apriori property, in the context of data cubes, states as follow:
If given cell does not satisfy minimum support, then no descendant (i.e. more specialized or detailed version) of the cell will satisfy minimum support either.
This property can be used to substantially reduce the computation of iceberg cubes.

Techniques for Cube computation

Top-down approach: Multi-way array aggregation

-> In multi-way array aggregation, the computations starts from the larger group-bys and proceeds towards the smallest group-bys as shown in below figure.

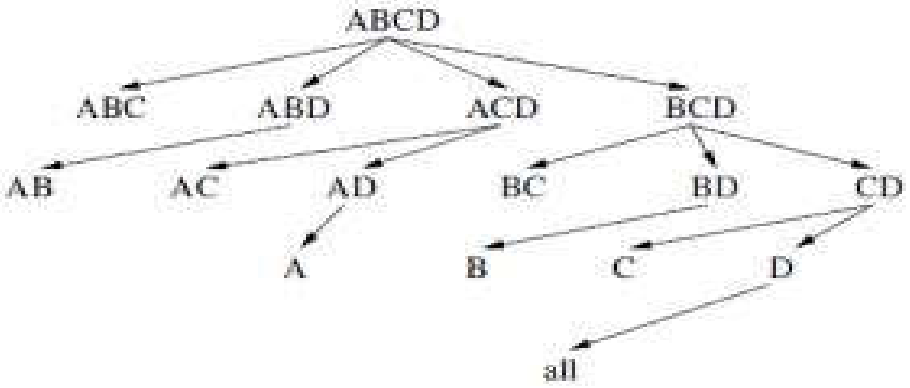


Fig 1: Top-Down Approach

Multi-way array aggregation

→A partition based algorithm is designed and implemented to convert a relational table or external load file to a chunked array.

→ There is no direct tuple comparision and it perform simultaneous aggragation on multiple dimensions.

→ In multi-way array aggregation, the intermediate aggregate values are re-used for computing ancestor cuboids.

→ It cannot do apriori pruning that means it can not perform iceberg cube optimization.

→ In Multi-Way array aggregation, It partition arrays into chunks (a small sub cube which fits in memory). It uses compressed sparse array addressing: (chunk_id, offset) and compute aggregates in —"multiway" by visiting cube cells in the order which minimizes the number of times to visit each cell, and reduces memory access and storage cost

→ **Limitation of the method**: computing well only for a small number of dimensions.

Bottom-up approach: Bottom-up computation (BUC)

→ BUC is an algorithm for sparse and iceberg cube computation which uses the bottom-up approach that allows to prune unnecessary computation by recurring to A-priori pruning strategy.

→ The A-priori pruning strategy states that if a given cell does not satisfy minsup,then no descendant will satisfy minsup either.

→ The iceberg cube computation problem is to compute all group-bys that satisfy an iceberg condition.

→ First, BUC partitions dataset on dimension A, producing partitions a1, a2, a3, a4.Then, it recurses on partition a1,the partition a1 is aggregated and BUC produces <a1,*,*,*>.

→ Next, it partitions a1 on dimension B. It produces <a1,b1,*,*> and recurses on partition a1,b1. Similarly, it produces <a1,b1,c1,*> and then <a1,b1,c1,d1>. Now, it returns from recursion and produces <a1,b1,c1,d2> etc.

BUC contd……

→After processing partition a1, BUC processes partition a2 and so on as shown in Fig.2 below

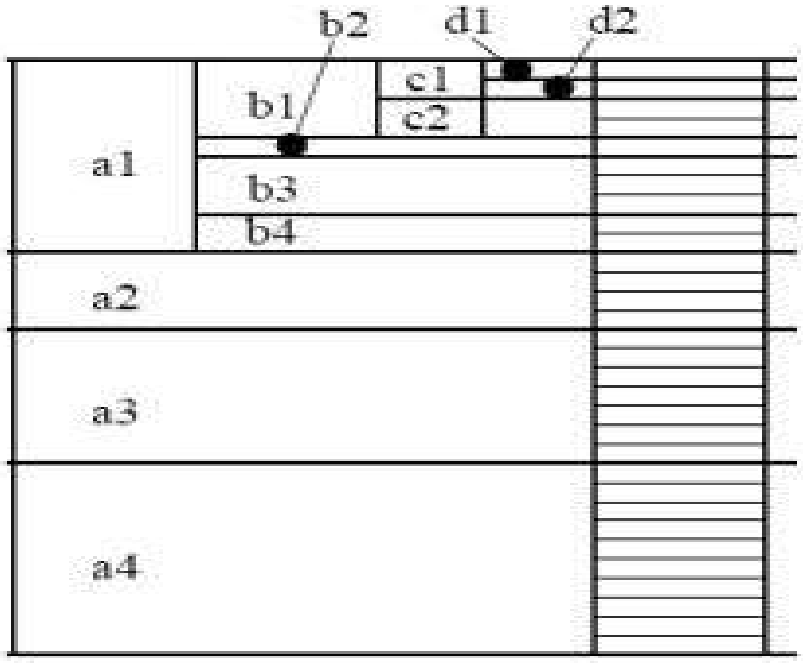


Fig 2 : BUC Partitioning

BUC Contd..

→BUC is sensitive to data skew and to the order of the dimensions processing and first most discriminating dimensions improves performance.

→It shares partitioning costs but does not share computation between parent and child cuboids.

Tuning and Testing of Data Warehouse

→**Self-study**

End of Unit 4

Data Warehousing & Data Mining

BSC.CSIT, 8th Sem

HCOE

Unit: 5

Data Mining Tools

ORACLE®

Microsoft®
SQL Server® 2012

WEKA

→**WEKA** (Waikato Environment for Knowledge Analysis) is a popular suite of machine learning software written in Java, developed at the University of Waikato, New Zealand.

→WEKA is free software available under the GNU General Public License.

Features:

- Written in JAVA
- Has graphical user interfaces
- Contains a collection of visualization tools and algorithms for data analysis and predictive modeling
- Supports standard data mining tasks like data preprocessing, clustering, classification, regression, visualization, and feature selection

Weka Contd..

Usage:

- Apply a learning method to a dataset & analyze the result
- Use a learned model to make predictions on new instances
- Apply different learners to a dataset & compare results

MS Excel

- In order to bridge the gap between the common user and the complex data mining process, Microsoft has introduced a new and efficient data mining tool ,the Microsoft SQL Server 2005 Data Mining Add-Ins for Office 2007 putting data mining within the reach of every user or desktop.
- The software pre-requisites for using the add-in are:
- Microsoft Office 2007 installed.
- Microsoft SQL Server 2005 or above installed.
  - Microsoft .NET 2.0 framework or higher (for SQL server 2008 only).
  - Microsoft PowerShell (for SQL server 2008 only)

Microsoft SQL Server

- →**Microsoft SQL Server** is a relational database server, developed by Microsoft: it is a software product whose primary function is to store and retrieve data as requested by other software applications, be it those on the same computer or those running on another computer across a network (including the Internet). Microsoft has introduced a wealth of new data mining features in Microsoft SQL Server 2008 that allow businesses to answer their concerns with data and mining for information in them.
- The current version of SQL Server, SQL Server 2008, (code-named "Katmai") aims to make data management self-tuning, self organizing, and self maintaining.
- SQL Server 2008 data mining features are integrated across all the SQL Server products, including SQL Server, SQL Server Integration Services, and Analysis Services.
- Accessing the data mining results is as simple as using an SQL-like language called Data Mining Extensions to SQL, or DMX.

Oracle

→The **Oracle Database** (commonly referred to as *Oracle RDBMS* or simply as *Oracle*) is an object-relational database management system (ORDBMS) produced and marketed by Oracle Corporation.

- **Oracle Data Mining (ODM)** is used to incorporate data mining with the Oracle database.
- ODM is used for both supervised (where a particular target value should be specified) and unsupervised (where patterns in data are observed) data mining.
- The results of Oracle Data Mining can be viewed by the Oracle Business Intelligence's reporting/publishing component.
- Oracle BI Standard Edition One is a product that is used to extract business information concealed in the data.
- **Oracle Warehouse Builder (OWB)** is used to create the logical and physical design of the data mart.

SPSS

→ SPSS (originally, Statistical Package for the Social Sciences) is a computer program used for survey authoring and deployment (**IBM SPSS Data Collection**), data mining (**IBM SPSS Modeler**), text analytics, statistical analysis, and collaboration and deployment (batch and automated scoring services).

Data Warehousing & Data Mining

BSC.CSIT, 8th Sem

HCOE

Unit: 6

DMQL (Data Mining Query Language)

→Data mining query language is a query language which is designed to support ad hoc and interactive data mining.

→The DMQL is actually based on SQL (Structure d Query Language).

→ The DMQL can work with databases and data warehouse and is used to define data mining tasks.

DMQL Contd…

→The DMQL provides the

1) Syntax for task-relevant data specification

- The data mining query language is used to specify the data that are relevant to particular subject of interest and helps in decision making process.

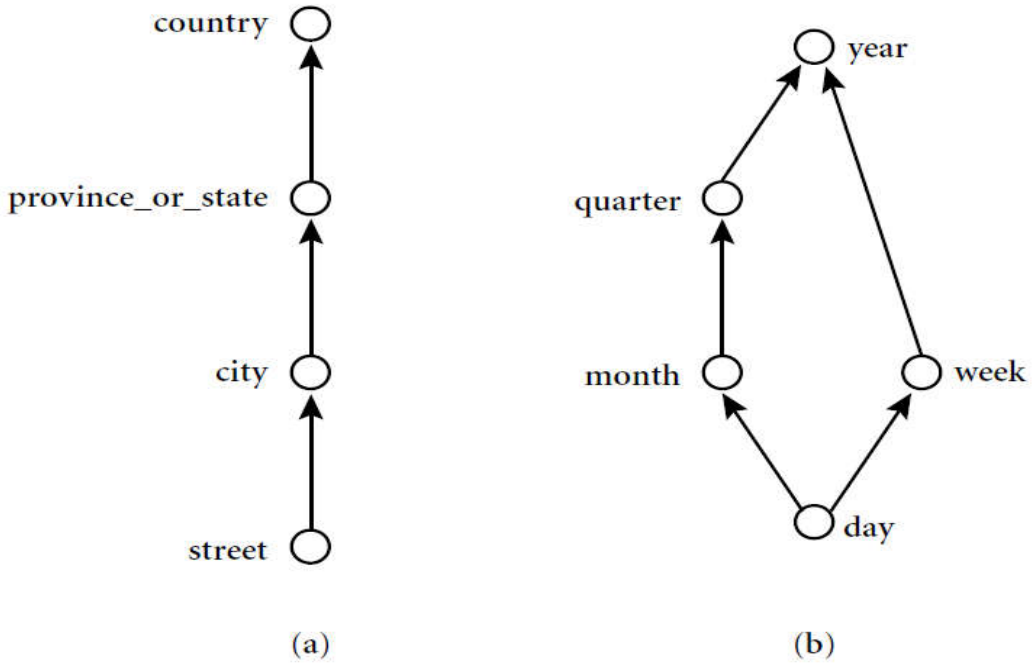
2) Syntax for specifying the kind of knowledge

- The DMQL specify the kind of knowledge by performing operations like : Characterization, Discrimination, Association, Classification and Prediction.

DMQL Contd…

3) Syntax for concept hierarchy specification

- The DMQL specify the concept hierarchy.
- In general, the concept hierarchy refers to mapping low level concepts to higher level concepts.



DMQL Contd…

4)**Syntax for Interestingness measure specification**

- DMQL provides the syntax for interestingness measure and thresholds which is specified by user.
- The threshold may be support threshold and confidence threshold.

5) Syntax for pattern presentation and visualization

- DMQL provides the syntax which allows user to specify the display of discovered patterns in one or more form.
- The discovered pattern can be presented and visulaized in the form of table,chart, diagram etc..

Data Mining Query Languages and Standardization

- Standardizing the Data Mining Languages will serve the following purposes:

  a) Helps systematic development of data mining solutions.

  b) Improves interoperability among multiple data mining systems and functions.

  c) Promotes education and rapid learning.

  d) Promotes the use of data mining systems in industry and society.

Multidimensional schema Definition using DMQL

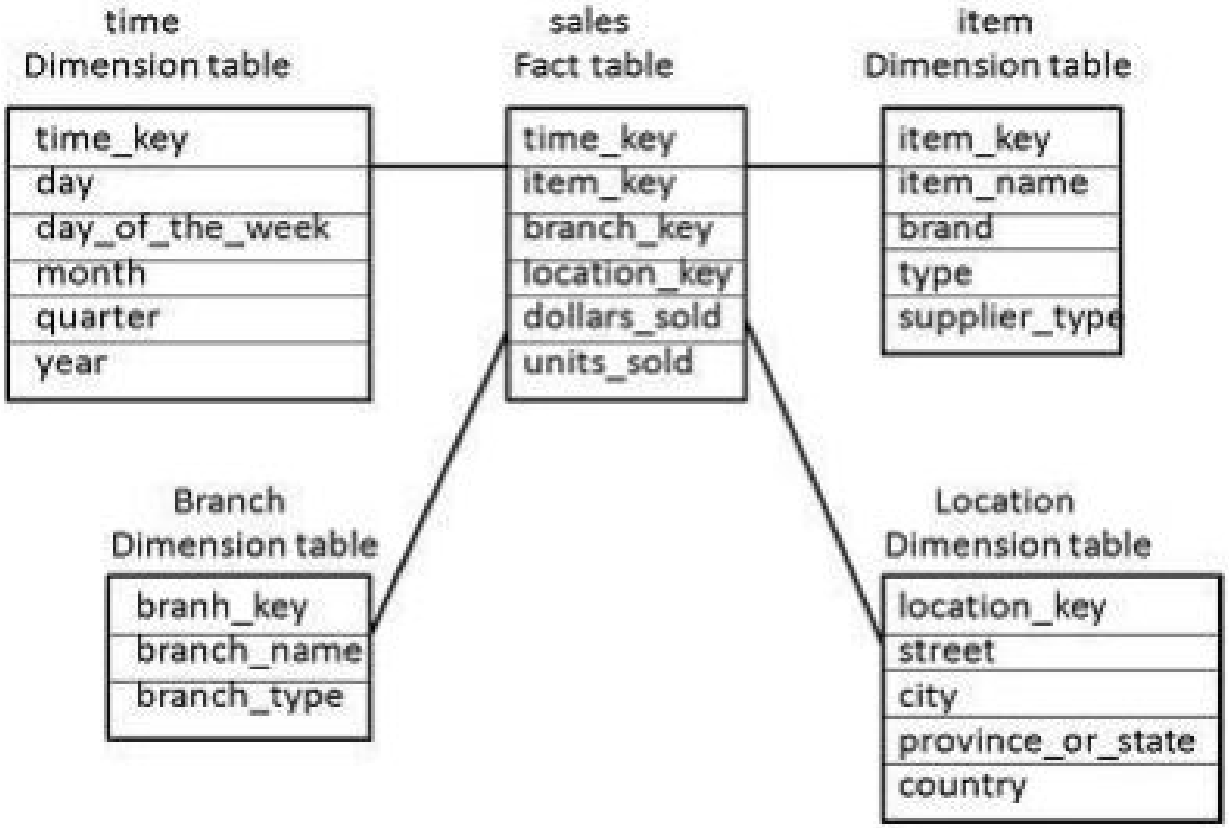
Syntax for Cube Definition

```
define cube < cube_name > [ < dimension-list > ]: < measure_list >
```

Syntax for Dimension Definition

```
define dimension < dimension_name > as ( < attribute_or_dimension_list > )
```

Star Schema


time
Dimension table
time_key
day
day_of_the_week
month
quarter
year
sales
Fact table
time_key
item_key
branch_key
location_key
dollars_sold
units_sold
item
Dimension table
item_key
item_name
brand
type
supplier_type
Branch
Dimension table
branh_key
branch_name
branch_type
Location
Dimension table
location_key
street
city
province_or_state
country

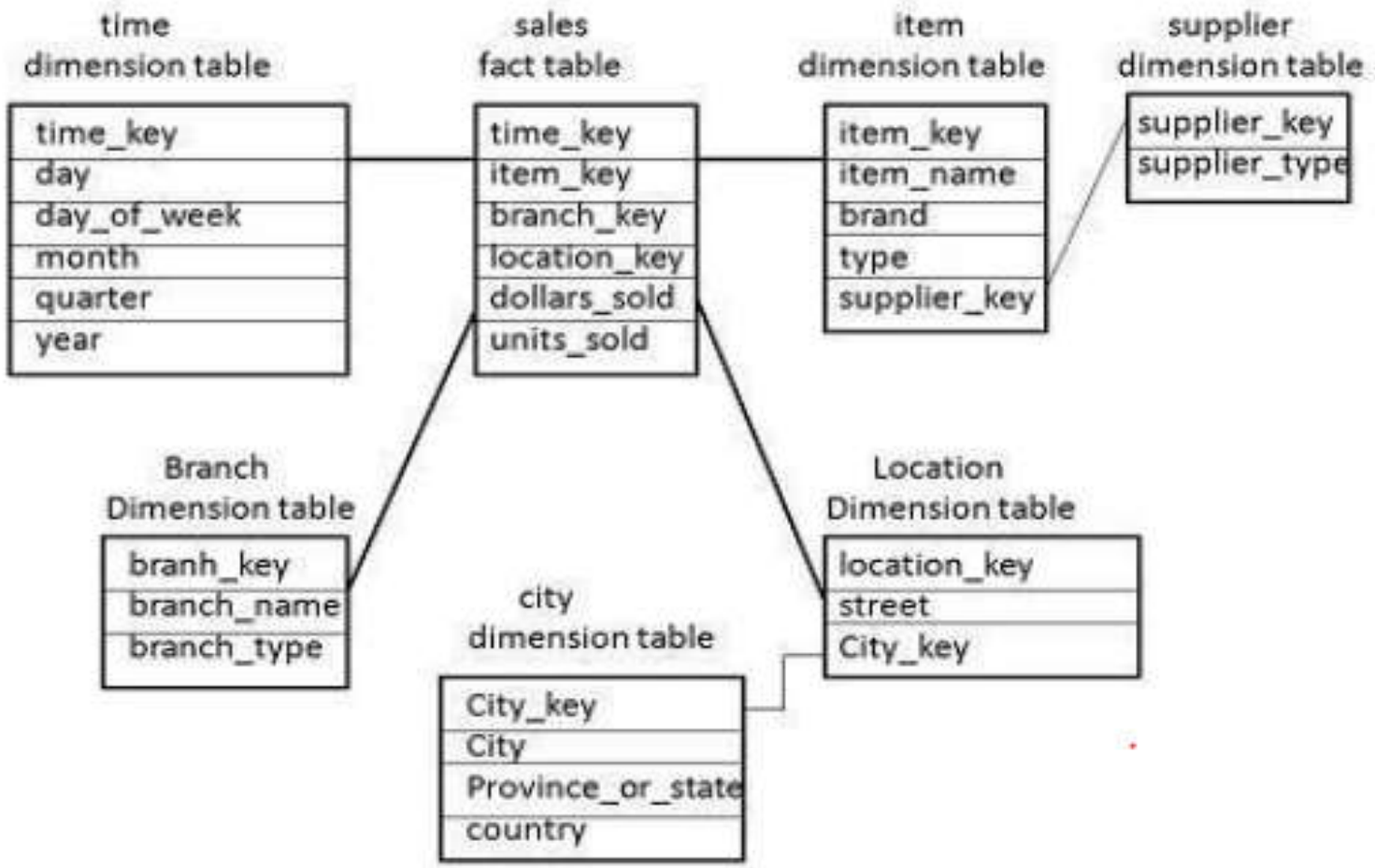
Star Schema Definition using DMQL

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

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

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

Snow flake Schema



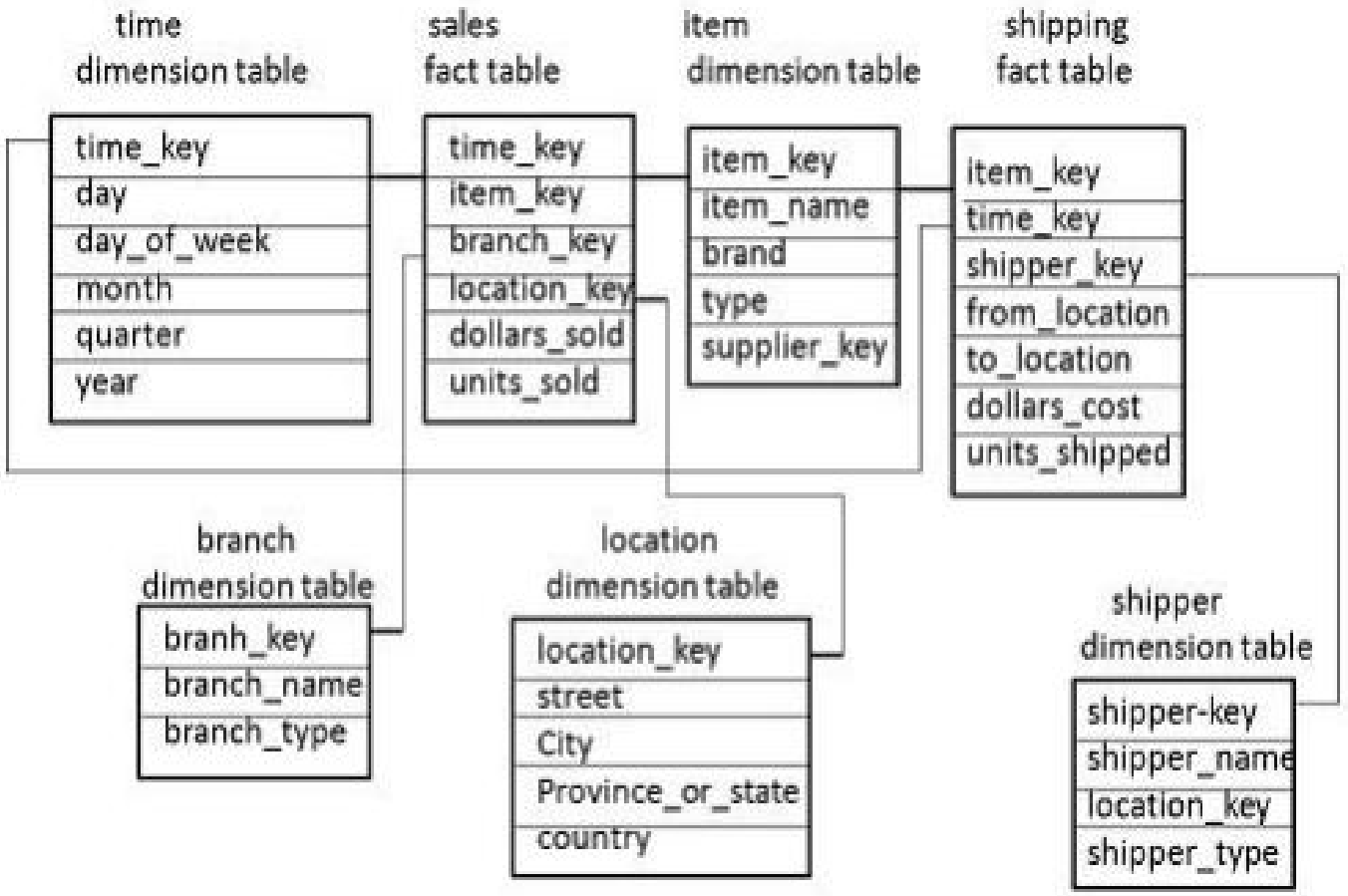
Snow Flake Schema Definition using DMQL

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

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

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

Fact Constellation Schema



Snow Flake schema definition using DMQL

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

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

define dimension time as (time key, day, day of week, month, quarter, year)
define dimension item as (item key, item name, brand, type, supplier type)
define dimension branch as (branch key, branch name, branch type)
define dimension location as (location key, street, city, province or
state, country)
define cube shipping [time, item, shipper, from location, to location]:

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

define dimension time as time in cube sales
define dimension item as item in cube sales
define dimension shipper as (shipper key, shipper name, location as
location in cube sales, shipper type)
define dimension from location as location in cube sales
define dimension to location as location in cube sales
```

End of Unit 6

Data Warehousing & Data Mining

BSC.CSIT, 8th Sem

HCOE

Unit: 7

Frequent Patterns

→Frequent patterns are patterns(i.e. item sets, subsequences or substructures) that occurs frequently in a data set.

For example: a set of items, such as milk and bread, that appear frequently together in a transaction data set is a frequent item set.

→ Finding frequent pattern plays an important role in mining associations, correlations and many other interesting relationship among data.

Market Basket Analysis

- → Market basket analysis is an example of frequent item set mining.
- → In market basket analysis, the market analyst analyzes customer buying habits by finding association between the different items that customers places in their shopping baskets.
- → The discovery of these associations helps the retailers to develop marketing strategies by gaining inside into which items are frequently purchased together by customer.
- → For instance, if customers are buying milk, how likely are they to also buy bread (and what kind of bread) on the same trip o the supermarket? This information can lead to increased sales by helping retailers do selective marketing and plan their shelf space.
- → Market basket analysis can also help retailers plan which items to put on sale at reduced prices. If customers tend to purchase computers and printers together, then having a sale on printers may encourage the sale of printers as well as computers.

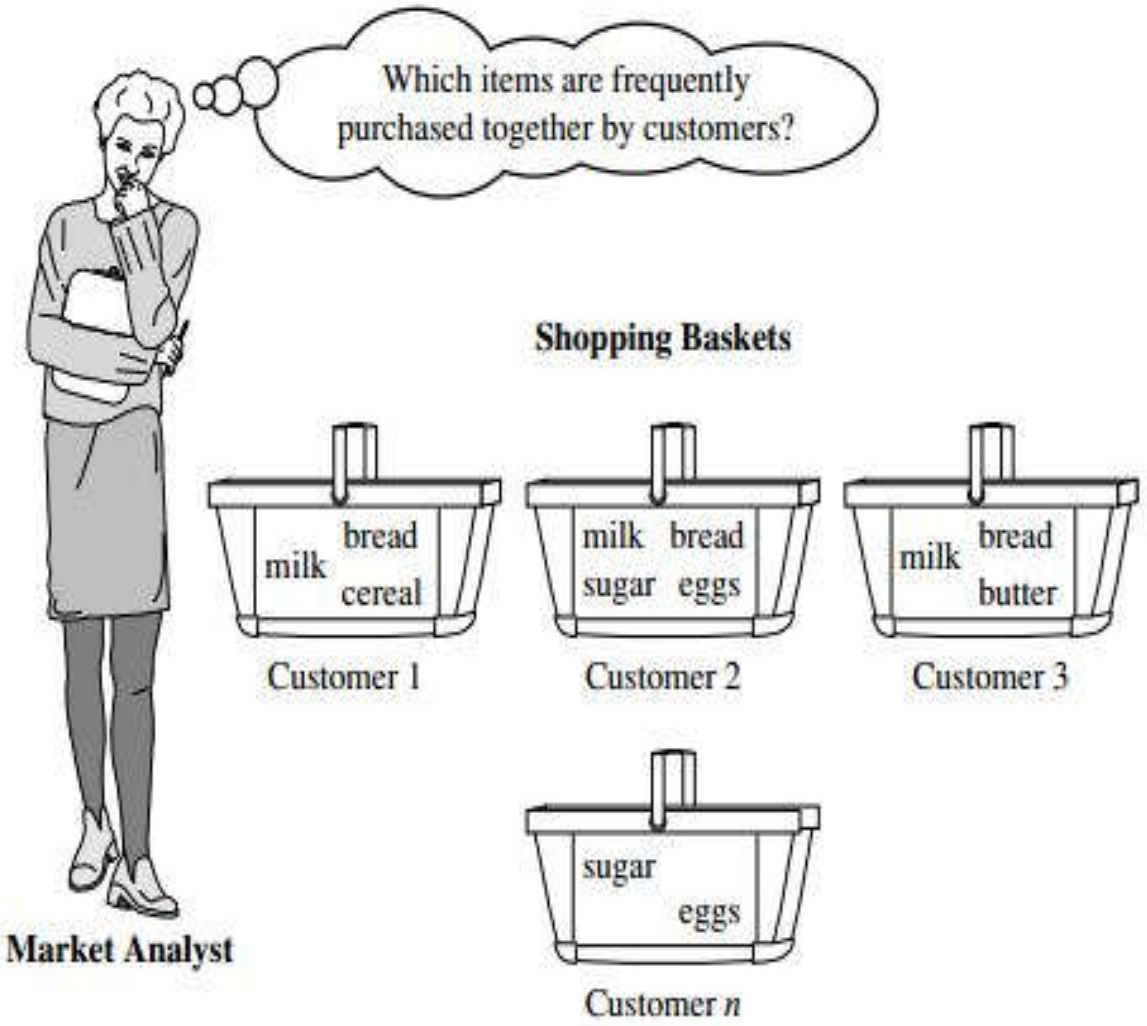


Figure 6.1 Market basket analysis.

Association Rule Mining (ARM)

- Association rule mining is the process of Finding frequent patterns, associations, correlations, or causal structures among sets of items or objects in transaction databases, relational databases, and other information repositories.
- An association rule can be in the form of implication, A → B, where A is called antecedent and B is called consequent. Both A and B are frequent item sets in a transactional database and A Ç B=∅ where (Ç=Intersecction).
- The rule A → B can be interpreted as "If item set A occurs in a transaction T, then item set B will also be there in the same transaction.
- To measure the rule interestingness in ARM Support and confidence are used .

ARM Contd…

- **Support**: Support is the probability of item or item sets in the given transactional database.
- Support(X) = n(X) / n where n is the total number of transactions in the database and n(X) is the number of transactions that contains the item set X .

  Therefore, support (X=>Y) = support (XUY)
- **Confidence**: confidence is conditional probability, for an association rule X=>Y confidence is defined as:

  Confidence(X=>Y) = support (XUY) / support(X).
- The association rules are considered to be interesting if they satisfy both minimum support and minimum confidence criteria. These criteria are specified by users or experts.
- The rules having support and confidence greater than or equal to the user specified criteria are extracted by association rule mining task.

ARM as two sub-processes

- Association rule mining consists of two sub-processes:- finding frequent item sets and generating association rules from those item sets.
- **Frequent itemset**: Frequent itemset is a set of items whose support is greater than the user specified minimum support. An itemset X in A (i.e.,X is a subset of A) is said to be a frequent itemsetin T with respect to σ, if $support(X)_T \geq \sigma$.
- **Association rule**: An association rule is an implication or if-then-rule which is supported by data and can be represented in the form X→Y. An association rule must satisfy user-set minimum support (min_sup) and minimum confidence (min_conf).The rule X→Y is called a strong association rule if support ≥ min_sup and confidence ≥ min_conf.

Application of ARM
(Why ARM is necessary?)

- To Find frequent patterns, associations, correlations, or causal structures among sets of items or objects in transaction databases, relational databases, and other information repositories.
- To discover relationships among various items in the database.
- To measure the interestingness of user based on support and confidence.
- Association rule mining can be used in application like: catalog design, market basket analysis, cross-marketing , clustering, classification etc.

  (**Note**: Explain market basket analysis in detail)

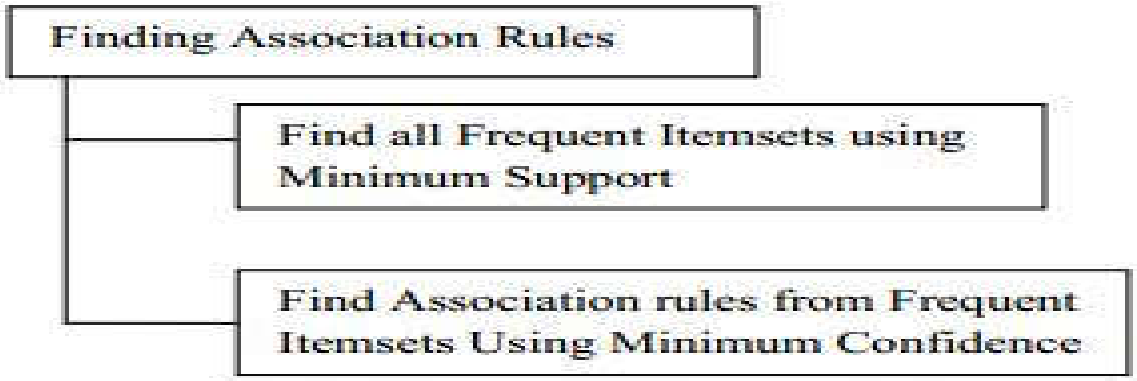


Figure 1: Generating Association Rules

Apriori algorithm

- Apriori pruning principle: If there is any itemset which is infrequent, its superset should not be generated/tested.
- Method:
  - Initially, scan DB once to get frequent 1-itemset
  - Generate length (k+1) candidate itemsets from length k frequent itemsets
  - Test the candidates against DB
  - Terminate when no frequent or candidate set can be generated

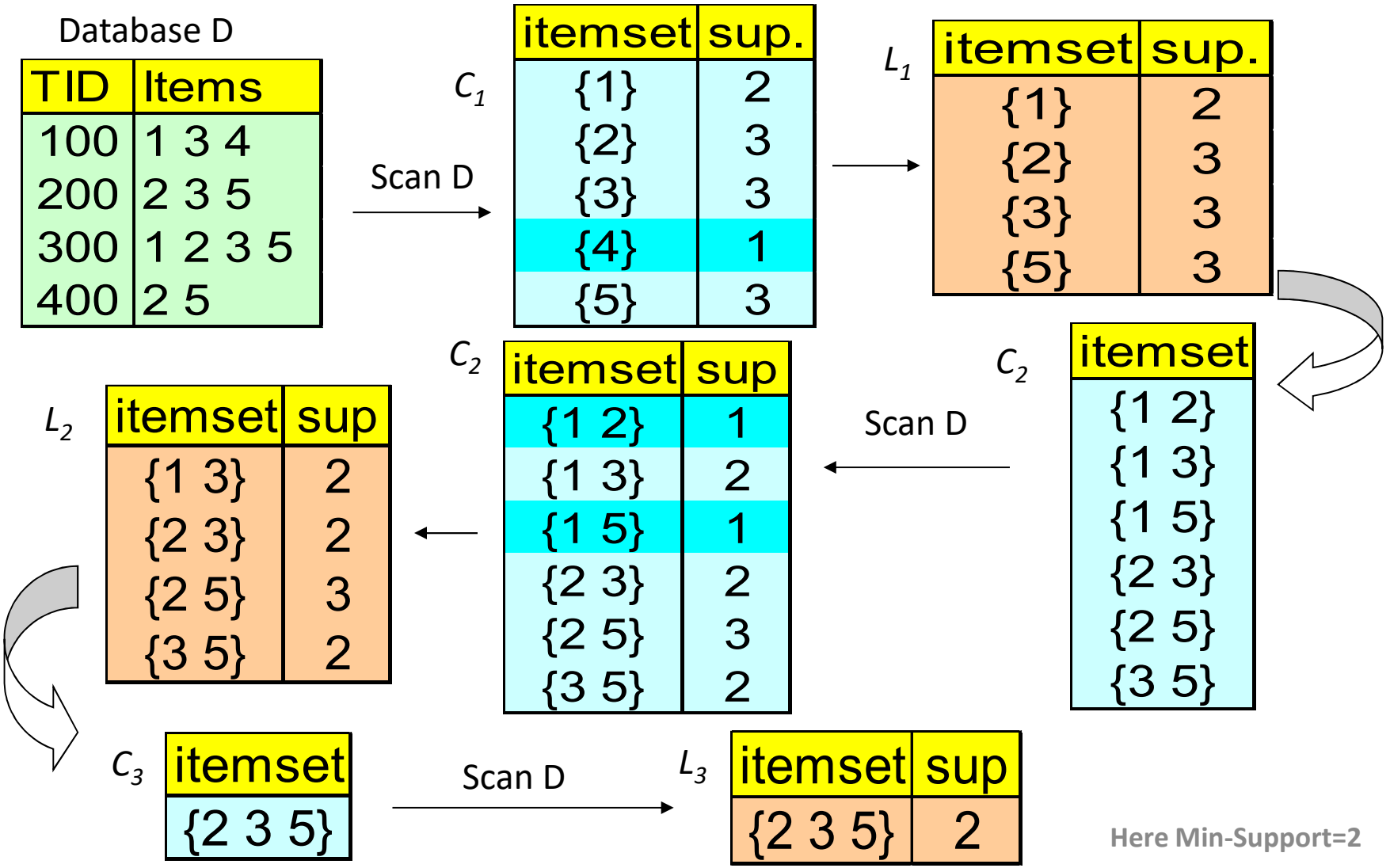
Apriori Algorithm

- Pseudo-code:

  C_k: Candidate itemset of size k
  L_k : frequent itemset of size k

  L_1 = {frequent items};
  for (k = 1; L_k !=∅; k++) **do begin**
  C_{k+1} = candidates generated from L_k;
  for each transaction t in database do
  increment the count of all candidates in C_{k+1} that are contained in t
  L_{k+1} = candidates in C_{k+1} with min_support
  end
  return $\cup_k L_k$;

Example


Database D
TID Items
100 1 3 4
200 2 3 5
300 1 2 3 5
400 2 5
Scan D
C1
itemset sup.
{1} 2
{2} 3
{3} 3
{4} 1
{5} 3
L1
itemset sup.
{1} 2
{2} 3
{3} 3
{5} 3
C2
itemset
{1 2}
{1 3}
{1 5}
{2 3}
{2 5}
{3 5}
Scan D
C2
itemset sup
{1 2} 1
{1 3} 2
{1 5} 1
{2 3} 2
{2 5} 3
{3 5} 2
L2
itemset sup
{1 3} 2
{2 3} 2
{2 5} 3
{3 5} 2
C3
itemset
{2 3 5}
Scan D
L3
itemset sup
{2 3 5} 2
Here Min-Support=2

Example Contd..

- Thus the generated association rules are

I2$\Rightarrow$I3 * I5

I3 $\Rightarrow$I2*I5

I5 $\Rightarrow$I3*I2

I2*I5 $\Rightarrow$I3

I3 *I2$\Rightarrow$I5

I3*I5 $\Rightarrow$I2

Class Work : Calculate Support and Confidence percent of above rule.(after studying next slide)

Mining Association Rules—An Example(Calculation of Support and Confidence)

Transaction ID	Items Bought
2000	A,B,C
1000	A,C
4000	A,D
5000	B,E,F

Min. support 50%
Min. confidence 50%

Frequent Itemset	Support
{A}	75%
{B}	50%
{C}	50%
{A,C}	50%

For rule $A \Rightarrow C$:

support = support({*A* union *C*}) = 50%

confidence = support({*A* union *C*})/support({*A*}) = 66.6%

Advantages and Disadvantages of Apriori Algorithm

Advantages

1. This algorithm has least memory consumption.
2 .Easy implementation.
3. It uses Apriori property for pruning therefore, itemsets left for further support checking remain less.

Disadvantages

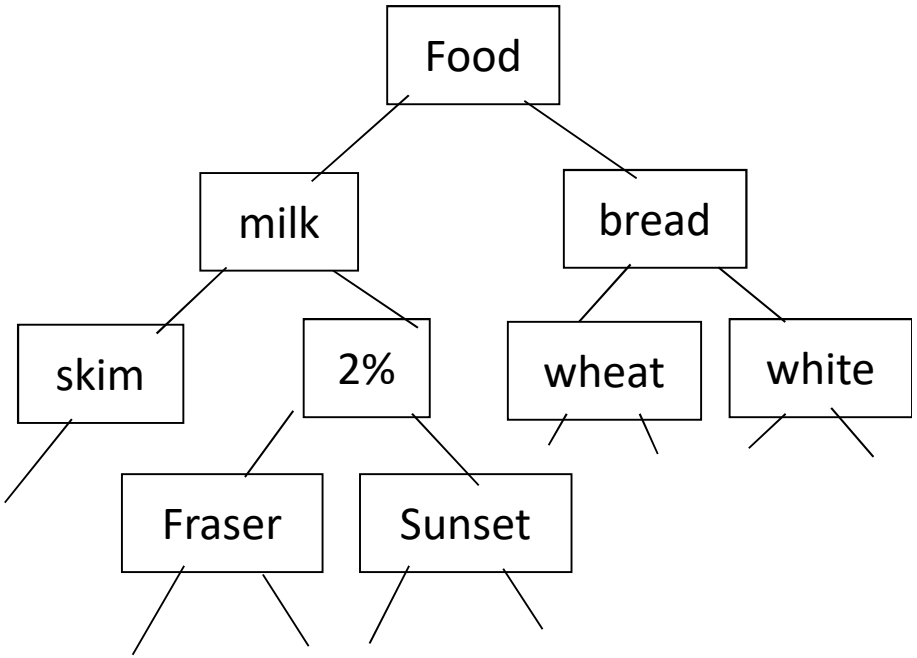
1. It requires many scans of database.
2. It allows only a single minimum support threshold.
3. It is favourable only for small database.
4. It explains only the presence or absence of an item in the database.
5. Obtaining non interesting rules
6. Huge number of discovered rules
7. Low algorithm performance

Types of Association Rules

- Boolean vs. quantitative associations (Based on the types of values handled)
  - buys(x, “SQLServer”) ^ buys(x, “DMBook”) → buys(x, “DBMiner”) [0.2%, 60%]
  - age(x, “30..39”) ^ income(x, “42..48K”) → buys(x, “PC”) [1%, 75%]
- Single dimension vs. multiple dimensional associations (each distinct predicate of a rule is a dimension)
- Single level vs. multiple-level analysis (consider multiple levels of abstraction)
  - What brands of beers are associated with what brands of diapers?

Multiple-Level Association Rules

- Items often form hierarchies.
- Items at the lower level are expected to have lower support.
- Rules regarding itemsets at appropriate levels could be quite useful.
- Transaction database can be encoded based on dimensions and levels
- We can explore shared multi-level mining



Multi-Dimensional Association: Concepts

- Single-dimensional rules:

  buys(X, “milk”) $\Rightarrow$ buys(X, “bread”)

- Multi-dimensional rules: 2 dimensions or predicates
  - Inter-dimension association rules (*no repeated predicates*)

    age(X,”19-25”) $\wedge$ occupation(X,“student”) $\Rightarrow$ buys(X,“coke”)
  - hybrid-dimension association rules (*repeated predicates*)

    age(X,”19-25”) $\wedge$ buys(X, “popcorn”) $\Rightarrow$ buys(X, “coke”)

End of Unit 7

Data Warehousing & Data Mining

BSC.CSIT, HCOE

8th Sem

Classification & Prediction

- There are two forms of data analysis that can be used for extracting models describing important classes or to predict future data trends
  1) Classification
  2) Prediction
- Classification models predict categorical class labels; and prediction models predict continuous valued functions.

For example: we can build a classification model to categorize bank loan applications as either safe or risky, or a prediction model to predict the expenditures in dollars of potential customers on computer equipment given their income and occupation

Classification vs. Prediction

- Classification
  - predicts categorical class labels
  - classifies data based on the training set and the values (class labels) in a classifying attribute and uses it in classifying new data
- Prediction
  - predicts unknown or missing values
- Typical applications
  - Credit approval
  - Target marketing
  - Medical diagnosis
  - Fraud detection

How Does Classification Work?

The Data Classification process includes two steps:

- Building the Classifier or Model
- Using Classifier for Classification

Building the Classifier or Model

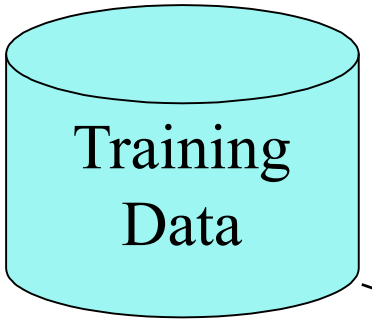
→ This step is the learning step or the learning phase.

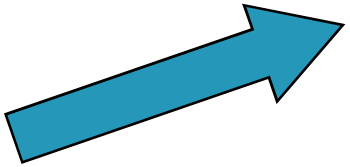
→In this step the classification algorithms build the classifier.

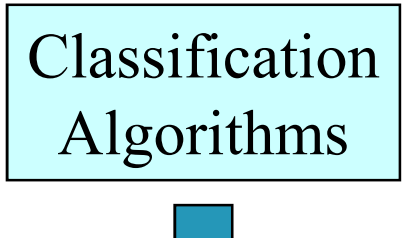
→The classifier is built from the training set made up of database tuples and their associated class labels.

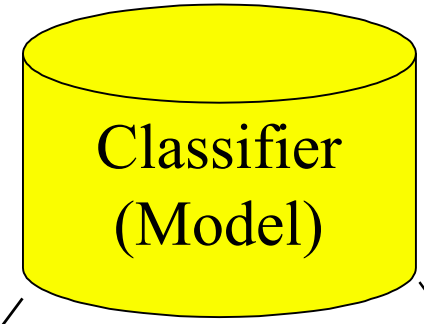
→Each tuple that constitutes the training set is referred to as a category or class. These tuples can also be referred to as sample, object or data points.

Building the Classifier or Model









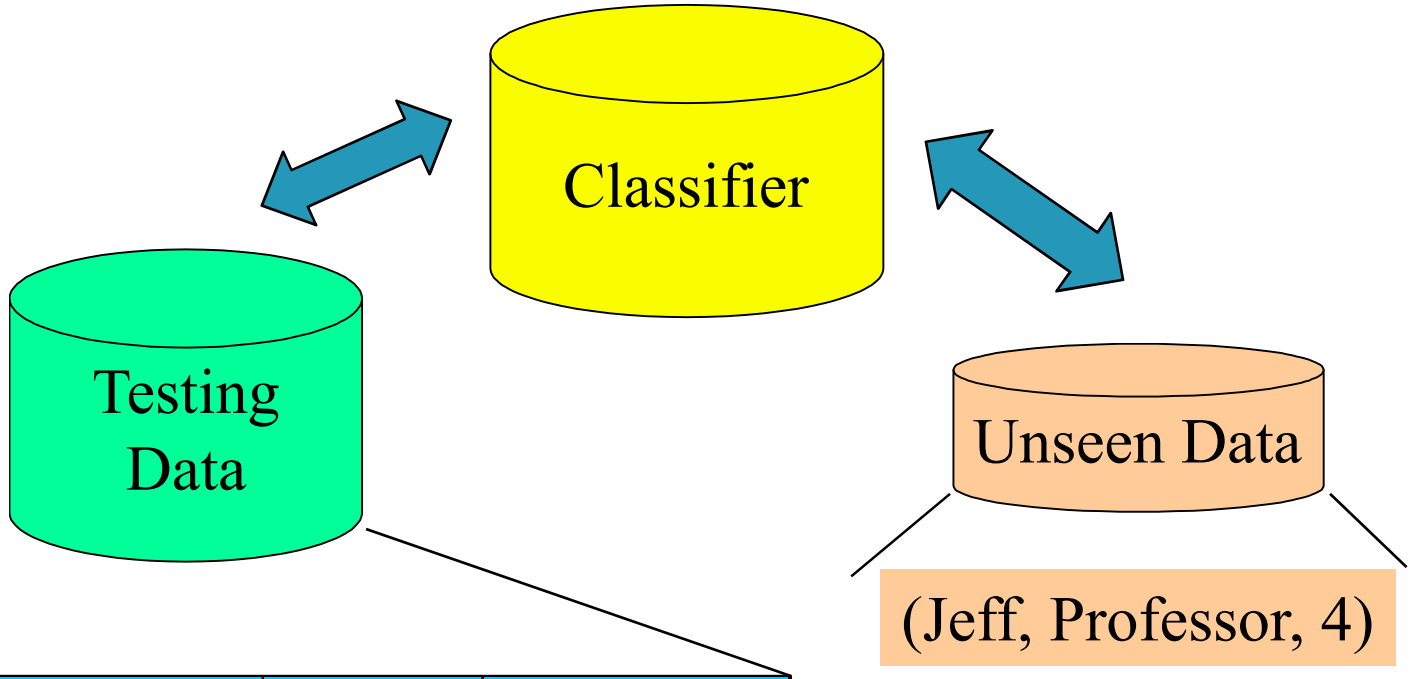
NAME	RANK	YEARS	TENURED
Mike	Assistant Prof	3	no
Mary	Assistant Prof	7	yes
Bill	Professor	2	yes
Jim	Associate Prof	7	yes
Dave	Assistant Prof	6	no
Anne	Associate Prof	3	no

IF rank = ‘professor’
OR years > 6
THEN tenured = ‘yes’

Using Classifier for Classification

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

Using Classifier for Classification



NAME	RANK	YEARS	TENURED
Tom	Assistant Prof	2	no
Merlisa	Associate Prof	7	no
George	Professor	5	yes
Joseph	Assistant Prof	7	yes

Classification and Prediction Issues

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

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

Note: Data can also be reduced by some other methods such as wavelet transformation, binning, histogram analysis, and clustering.

Comparison of Classification and Prediction Methods

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

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

CATEGORIZATION (Classification) ALGORITHM

- Baye's Theorem
- Decision Trees
- Back Propagation Algorithm

BAYESIAN METHODS

- Learning and classification methods based on probability theory.
- Baye's theorem plays a critical role in probabilistic learning and classification.
- Uses prior probability of each category given no information about an item.
- Categorization produces a posterior probability distribution over the possible categories given a description of an item.

$$P(A|B) = \frac{P(A) * P(B \mid A)}{P(B)}$$

BAYESIAN METHODS...

D =

Size	Color	Shape	Category
Small	Red	Circle	Positive
Large	Red	Circle	Positive
Small	Red	Triangle	Negative
Large	blue	Circle	Negative

Size<small, medium, large>

Color<red, blue, green>

Shape<circle, triangle, square>

Category<positive, negative>

AFTER TRAINING

Probability	Positive	Negative
P(Y)	0.5	0.5
P(small|Y)	0.5	0.5
P(medium|Y)	0.0	0.0
P(large|Y)	0.5	0.5
P(red|Y)	1.0	0.5
P(blue|Y)	0.0	0.5
P(green|Y)	0.0	0.0
P(square|Y)	0.0	0.5
P(triangle|Y)	0.0	0.5
P(circle|Y)	1.0	0.5

BAYESIAN METHODS...

- Testing Sample X: <medium, red, circle>

- $P(pos \mid X) = \frac{P(pos) * P(X \mid pos)}{P(X)}$

  $= P(pos) * P(medium \mid pos) * P(red \mid pos) * P(circle \mid pos)$

  $= 0.5 * 0.001 * 1.0 * 1.0$

  $= 0.0005$

- $P(neg \mid X) = \frac{P(neg) * P(X \mid neg)}{P(X)}$

  $= P(neg) * P(medium \mid neg) * P(red \mid neg) * P(circle \mid neg)$

  $= 0.5 * 0.001 * 0.5 * 0.5$

  $= 0.000125$

Naïve Bayesian Classifier: Training Dataset

Class:
C1:buys_computer = 'yes'
C2:buys_computer = 'no'

Data sample
X = (age <=30,
Income = medium,
Student = yes
Credit_rating = Fair)

age	income	student	credit_rating	comp
<=30	high	no	fair	no
<=30	high	no	excellent	no
31…40	high	no	fair	yes
>40	medium	no	fair	yes
>40	low	yes	fair	yes
>40	low	yes	excellent	no
31…40	low	yes	excellent	yes
<=30	medium	no	fair	no
<=30	low	yes	fair	yes
>40	medium	yes	fair	yes
<=30	medium	yes	excellent	yes
31…40	medium	no	excellent	yes
31…40	high	yes	fair	yes
>40	medium	no	excellent	no

Naïve Bayesian Classifier: An Example

- $P(C_i)$: P(buys_computer = "yes") = 9/14 = 0.643
  P(buys_computer = "no") = 5/14= 0.357

- Compute $P(X|C_i)$ for each class
  P(age = "<=30" | buys_computer = "yes") = 2/9 = 0.222
  P(age = "<= 30" | buys_computer = "no") = 3/5 = 0.6
  P(income = "medium" | buys_computer = "yes") = 4/9 = 0.444
  P(income = "medium" | buys_computer = "no") = 2/5 = 0.4
  P(student = "yes" | buys_computer = "yes) = 6/9 = 0.667
  P(student = "yes" | buys_computer = "no") = 1/5 = 0.2
  P(credit_rating = "fair" | buys_computer = "yes") = 6/9 = 0.667
  P(credit_rating = "fair" | buys_computer = "no") = 2/5 = 0.4

- **X = (age <= 30 , income = medium, student = yes, credit_rating = fair)**

$P(X|C_i)$: P(X|buys_computer = "yes") = 0.222 x 0.444 x 0.667 x 0.667 = 0.044
P(X|buys_computer = "no") = 0.6 x 0.4 x 0.2 x 0.4 = 0.019
$P(X|C_i)*P(C_i)$: P(X|buys_computer = "yes") * P(buys_computer = "yes") = 0.028
P(X|buys_computer = "no") * P(buys_computer = "no") = 0.007

Therefore, X belongs to class ("buys_computer = yes")

DECISION TREES

- Decision tree induction is the learning of decision trees from class labeled training tuples.
- A decision tree is a flowchart like tree structure where each internal node denotes a test on an attribute, each branch represents an outcome of the test and each node holds a class label.
- The benefits of having a decision tree are as follows:

  → It does not require any domain knowledge.

  → It is easy to comprehend.

  →The learning and classification steps of a decision tree are simple and fast.

Algorithm for Decision tree

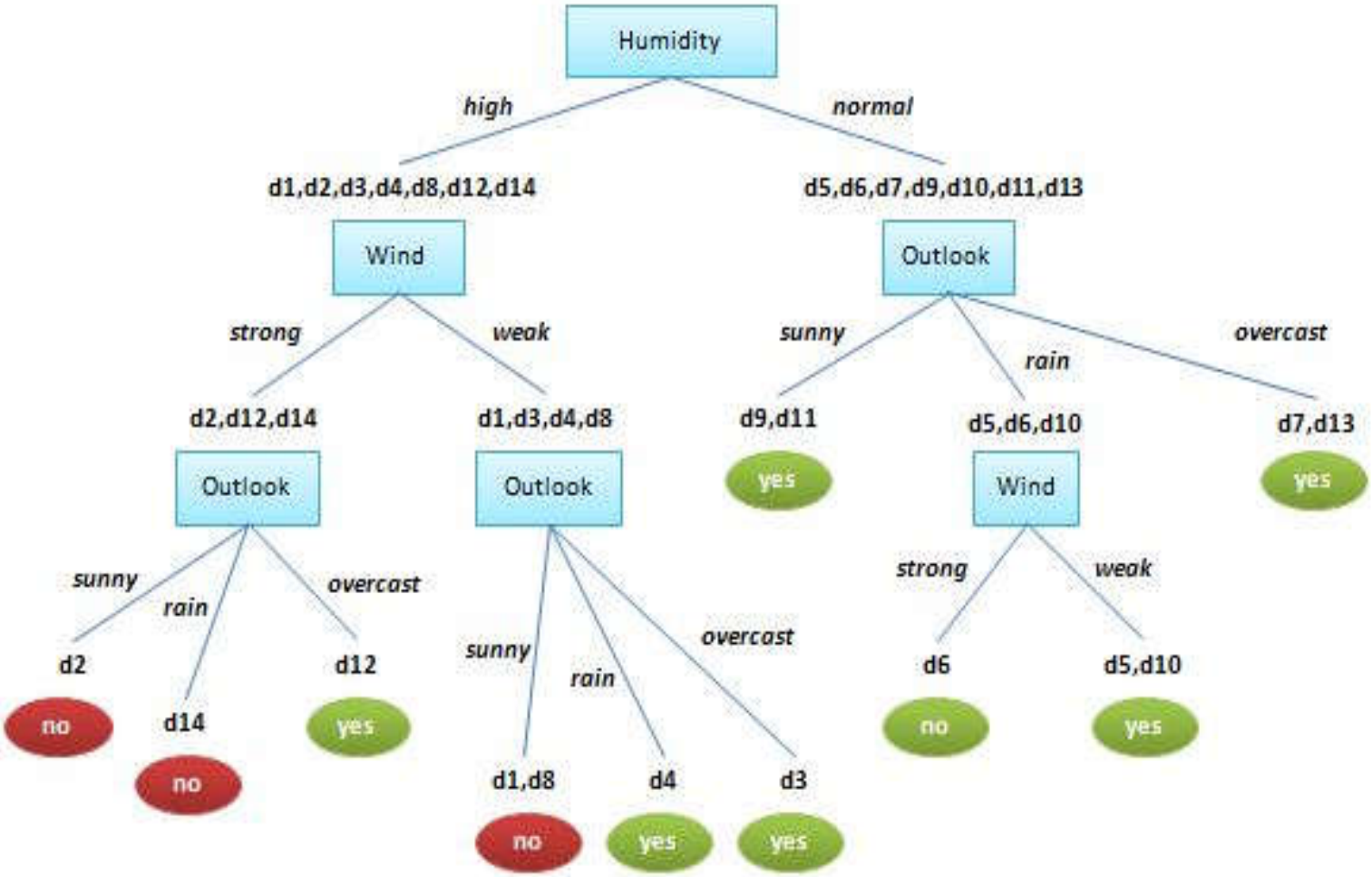
Input:
T *//Decision Tree*
D *//Input Database*
Output:
M *//Model Prediction*
DTProc Algorithm:
//Illustrates Prediction Technique using DT
for each $t \in D$ **do**
n = *root node of* T;
while *n not leaf node* **do**
Obtain answer to question on n *applied* t;
Identify arc from t *which contains correct answer;*
n = *node at end of this arc;*
Make prediction for t *based on labeling of* n;

DECISION TREES...(EXAMPLE)

Day	Outlook	Temperature	Humidity	Wind	Play Tennis
D1	SUNNY	HOT	HIGH	WEAK	NO
D2	SUNNY	HOT	HIGH	STRONG	NO
D3	OVERCAST	HOT	HIGH	WEAK	YES
D4	RAIN	MILD	HIGH	WEAK	YES
D5	RAIN	COOL	NORMAL	WEAK	YES
D6	RAIN	COOL	NORMAL	STRONG	NO
D7	OVERCAST	COOL	NORMAL	STRONG	YES
D8	SUNNY	MILD	HIGH	WEAK	NO
D9	SUNNY	COOL	NORMAL	WEAK	YES
D10	RAIN	MILD	NORMAL	WEAK	YES
D11	SUNNY	MILD	NORMAL	STRONG	YES
D12	OVERCAST	MILD	HIGH	STRONG	YES
D13	OVERCAST	HOT	NORMAL	WEAK	YES
D14	RAIN	MILD	HIGH	STRONG	NO

Outlook <sunny, overcast, rain>
Temperature <hot, mild, cool>
Humidity <high, normal>
Wind <weak, strong>
Play Tennis <no, yes>

Decision Tree (Example)



Testing : <sunny, hot, normal, weak> → YES

Information gain

- ID3 uses information gain as its attribute selection measure
- Based on the value or “information content” of messages
- Let node *N represent or hold the tuples of partition D*
- The attribute with the highest information gain is chosen as the splitting attribute for node *N*
- This attribute minimizes the information needed to classify the tuples in the resulting partitions

Information gain contd..

- Select the attribute with the highest information gain
- Let p_i be the probability that an arbitrary tuple in D belongs to class C_i, estimated by $|C_{i,D}|/|D|$
- Expected information (entropy) needed to classify a tuple in D:

$$Info(D) = -\sum_{i=1}^{m} p_i \log_2(p_i)$$

- Information needed (after using A to split D into v partitions) to classify D:

$$Info_A(D) = \sum_{j=1}^{v} \frac{|D_j|}{|D|} \times I(D_j)$$

- Information gained by branching on attribute A

$$Gain(A) = Info(D) - Info_A(D)$$

Attribute Selection: Information gain

- Class P: buys_computer = “yes”
- Class N: buys_computer = “no”

$$Info_{age}(D) = \frac{5}{14} I(2,3) + \frac{4}{14} I(4,0) + \frac{5}{14} I(3,2) = 0.694$$

age	p_i	n_i	$I(p_i, n_i)$
<=30	2	3	0.971
31…40	4	0	0
>40	3	2	0.971

means “age <=30” has 5 out of 14 samples, with 2 yes’es and 3 no’s. Hence

$$Gain(age) = Info(D) - Info_{age}(D) = 0.246$$

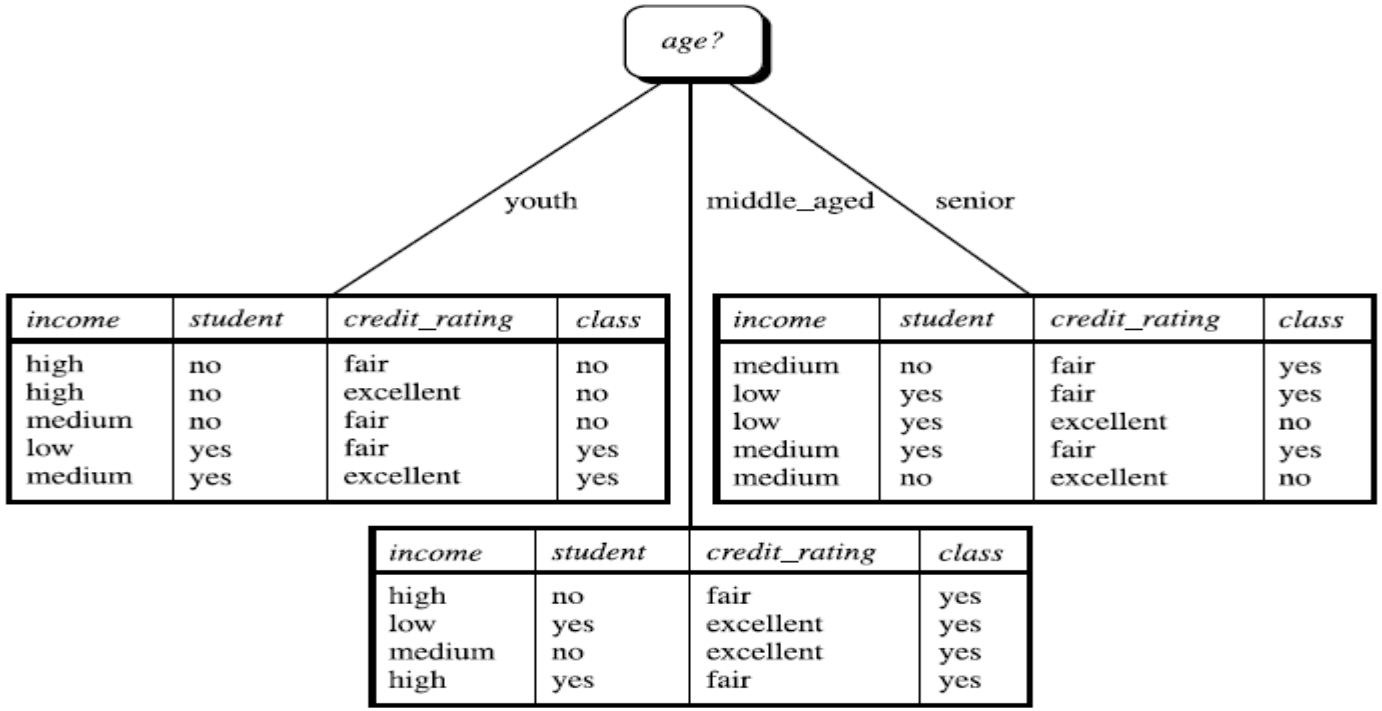
Similarly,

$$Gain(income) = 0.029$$
$$Gain(student) = 0.151$$
$$Gain(credit\_rating) = 0.048$$

age	income	student	credit_rating	buys_computer
<=30	high	no	fair	no
<=30	high	no	excellent	no
31…40	high	no	fair	yes
>40	medium	no	fair	yes
>40	low	yes	fair	yes
>40	low	yes	excellent	no
31…40	low	yes	excellent	yes
<=30	medium	no	fair	no
<=30	low	yes	fair	yes
>40	medium	yes	fair	yes
<=30	medium	yes	excellent	yes
31…40	medium	no	excellent	yes
31…40	high	yes	fair	yes
>40	medium	no	excellent	no

Information gain: Attribute Selection

- Because *age has the highest information gain among* the attributes, it is selected as the splitting attribute



income	*student*	*credit_rating*	*class*
high	no	fair	no
high	no	excellent	no
medium	no	fair	no
low	yes	fair	yes
medium	yes	excellent	yes

income	*student*	*credit_rating*	*class*
medium	no	fair	yes
low	yes	fair	yes
low	yes	excellent	no
medium	yes	fair	yes
medium	no	excellent	no

income	*student*	*credit_rating*	*class*
high	no	fair	yes
low	yes	excellent	yes
medium	no	excellent	yes
high	yes	fair	yes

ID3 Algorithm:

1.create a node N;
2. if tuples in D are all of the same class C then
3. return N as a leaf node labeled with the class C;
4. if attribute_list is empty then
5. return N as a leaf node labled with the majority class in D;
6.apply Attribute_selection_method(D, attribute_list) to find the "best"splitting_criterion;
label node N with with splitting_criterion;
7. if splitting_attribute is discrete-valued and
multiway splits allowed then //not restricted to binary trees
attribute_list (arrow mark) attribute _ list - splitting_attribute;
8. for each outcome j of splitting_criterion
9. let (symbol)be the set of data tuples in D satisfying outcome j; // partition
10.if (symbol) is empty then
attach a leaf labeled with the majority class in D to node N;
11. else attach the node returned by
Generate_decision_tree(symbol,attribute_list)to node N;
endfor
return N;

Advantages of using ID3

- Understandable prediction rules are created from the training data.
- Builds the fastest tree.
- Builds a short tree.
- Only need to test enough attributes until all data is classified.
- Finding leaf nodes enables test data to be pruned, reducing number of tests.
- Whole dataset is searched to create tree

Disadvantages of using ID3

- Data may be over-fitted or over-classified, if a small sample is tested.
- Only one attribute at a time is tested for making a decision.
- Classifying continuous data may be computationally expensive, as many trees must be generated to see where to break the continuum.

Pros and Cons of Decision Tree

Pros

- no distributional assumptions
- can handle real and nominal inputs
- speed and scalability
- robustness to outliers and missing values
- interpretability
- compactness of classification rules
- They are easy to use.
- Generated rules are easy to understand .
- Amenable to scaling and the database size.

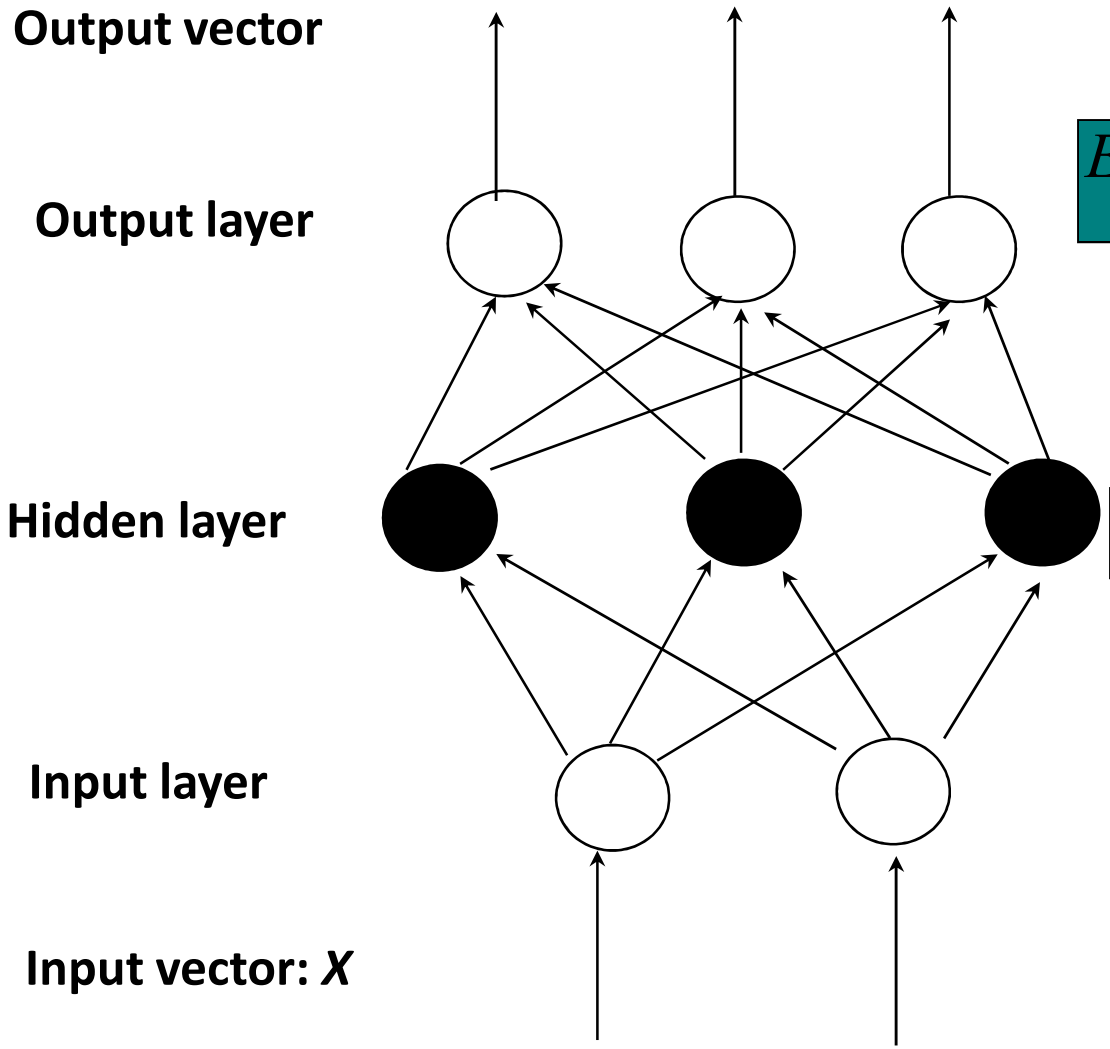
Cons

- several tuning parameters to set with little guidance
- decision boundary is non-continuous
- Cannot handle continuous data.
- Incapable of handling many problems which cannot be divided into attribute domains.
- Can lead to over-fitting as the trees are constructed from training data.

Classification by Backpropagation

- Backpropagation: A **neural network** learning algorithm
- Started by psychologists and neurobiologists to develop and test computational analogues of neurons
- A neural network: A set of connected input/output units where each connection has a **weight** associated with it
- During the learning phase, the **network learns by adjusting the weights** so as to be able to predict the correct class label of the input tuples
- Also referred to as **connectionist learning** due to the connections between units

A Multi-Layer Feed-Forward Neural Network



$$Err_j = O_j(1-O_j)\sum_k Err_k w_{jk}$$

$$\theta_j = \theta_j + (l)Err_j$$

$$w_{ij} = w_{ij} + (l)Err_j O_i$$

$$Err_j = O_j(1-O_j)(T_j - O_j)$$

w_{ij}

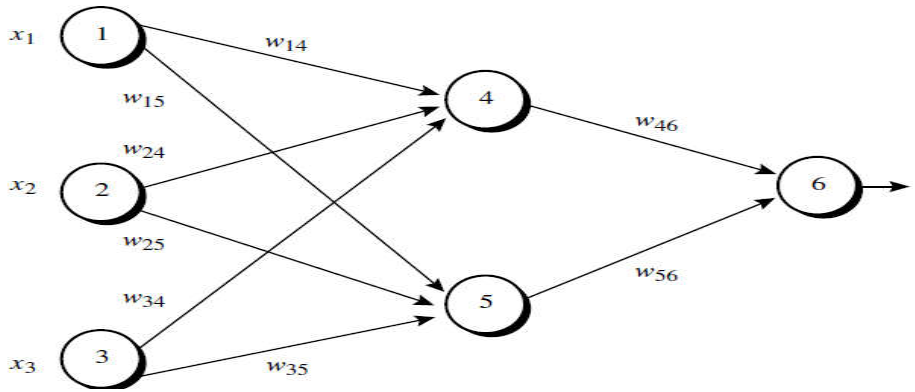
$$O_j = \frac{1}{1+e^{-I_j}}$$

$$I_j = \sum_i w_{ij} O_i + \theta_j$$

Algorithm

(1) Initialize all weights and biases in *network*;
(2) while terminating condition is not satisfied {
(3) for each training tuple X in D {
(4) // Propagate the inputs forward:
(5) for each input layer unit j {
(6) $O_j = I_j$; // output of an input unit is its actual input value
(7) for each hidden or output layer unit j {
(8) $I_j = \sum_i w_{ij} O_i + \theta_j$; //compute the net input of unit j with respect to the previous layer, i
(9) $O_j = \frac{1}{1+e^{-I_j}}$; } // compute the output of each unit j
(10) // Backpropagate the errors:
(11) for each unit j in the output layer
(12) $Err_j = O_j(1-O_j)(T_j - O_j)$; // compute the error
(13) for each unit j in the hidden layers, from the last to the first hidden layer
(14) $Err_j = O_j(1-O_j)\sum_k Err_k w_{jk}$; // compute the error with respect to the next higher layer, k
(15) for each weight w_{ij} in *network* {
(16) $\Delta w_{ij} = (l) Err_j O_i$; // weight increment
(17) $w_{ij} = w_{ij} + \Delta w_{ij}$; } // weight update
(18) for each bias θ_j in *network* {
(19) $\Delta \theta_j = (l) Err_j$; // bias increment
(20) $\theta_j = \theta_j + \Delta \theta_j$; } // bias update
(21) } }

Example:



Initial input, weight, and bias values.

x_1	x_2	x_3	w_{14}	w_{15}	w_{24}	w_{25}	w_{34}	w_{35}	w_{46}	w_{56}	θ_4	θ_5	θ_6
1	0	1	0.2	−0.3	0.4	0.1	−0.5	0.2	−0.3	−0.2	−0.4	0.2	0.1

The net input and output calculations.

Unit j	*Net input, I_j*	*Output, O_j*
4	$0.2+0-0.5-0.4=-0.7$	$1/(1+e^{0.7})=0.332$
5	$-0.3+0+0.2+0.2=0.1$	$1/(1+e^{-0.1})=0.525$
6	$(-0.3)(0.332)-(0.2)(0.525)+0.1=-0.105$	$1/(1+e^{0.105})=0.474$

Advantages of Neural Network

- prediction accuracy is generally high
- robust, works when training examples contain errors
- output may be discrete, real-valued, or a vector of several discrete or real-valued attributes
- fast evaluation of the learned target function
- High tolerance to noisy data
- Ability to classify untrained patterns
- Well-suited for continuous-valued inputs and outputs
- Successful on a wide array of real-world data
- Algorithms are inherently parallel
- Techniques have recently been developed for the extraction of rules from trained neural networks

Disadvantages of Neural Network

- long training time
- difficult to understand the learned function (weights)
- not easy to incorporate domain knowledge
- Require a number of parameters typically best determined empirically, e.g., the network topology or ``structure."
- *Poor interpretability:* Difficult to interpret the symbolic meaning behind the learned weights and of ``hidden units" in the network

Data Preprocessing
Unit 1

DW& DM, BSC.CSIT

8th sem

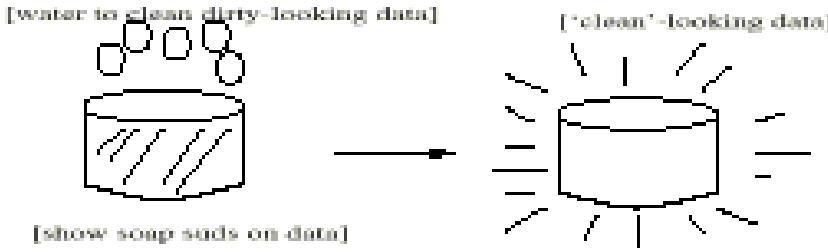
HCOE

Major Task in Data Preprocessing

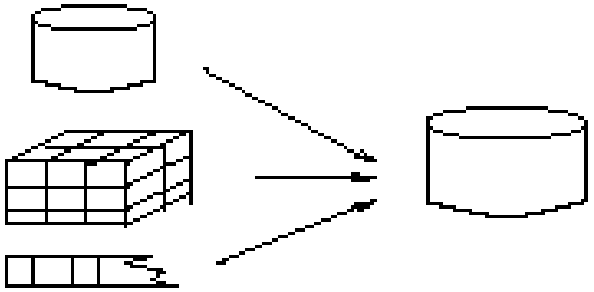
- Data cleaning
  - Fill in missing values, smooth noisy data, identify or remove outliers, and resolve inconsistencies
- Data integration
  - Integration of multiple databases, data cubes, or files
- Data transformation
  - Normalization and aggregation
- Data reduction
  - Obtains reduced representation in volume but produces the same or similar analytical results
- Data discretization
  - Part of data reduction but with particular importance, especially for numerical data

Forms of Data preprocessing

Data Cleaning



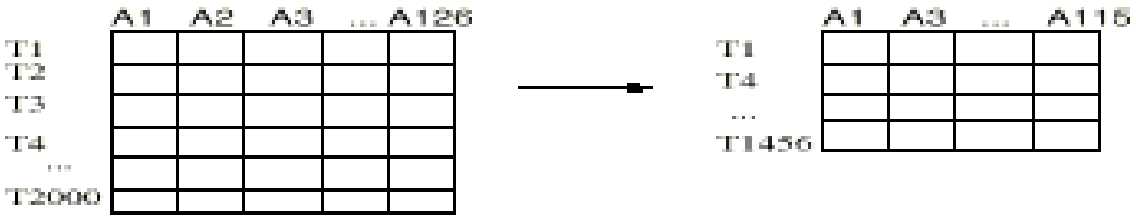
Data Integration



Data Transformation

-2, 32, 100, 59, 48 → -0.02, 0.32, 1.00, 0.59, 0.48

Data Reduction



Further study

- For more detail of Data preprocessing go through Kambler book of data mining

Clustering

By : Babu Ram Dawadi

Clustering

- cluster is a collection of data objects, in which the objects similar to one another within the same cluster and dissimilar to the objects in other clusters

- Cluster analysis is the process of finding similarities between data according to the characteristics found in the data and grouping similar data objects into clusters.

- Clustering: Given a database D = {t1, t2, .., tn}, a distance measure dis(ti, tj) defined between any two objects ti and tj, and an integer value k, the clustering problem is to define a mapping f: D -> {1, …, k} where each ti is assigned to one cluster Kj, 1<=j<=k. here ‘k’ is the number of clusters.

Examples of Clustering Applications

- Marketing: Help marketers discover distinct groups in their customer bases, and then use this knowledge to develop targeted marketing programs
- Land use: Identification of areas of similar land use in an earth observation database
- Insurance: Identifying groups of motor insurance policy holders with a high average claim cost
- City-planning: Identifying groups of houses according to their house type, value, and geographical location
- Earth-quake studies: Observed earth quake epicenters should be clustered along continent faults

Categories of Clustering

main categories (classes) of clustering methods

- Partition-based
- Hierarchical
- Density-based
- Grid-based
- Model-based

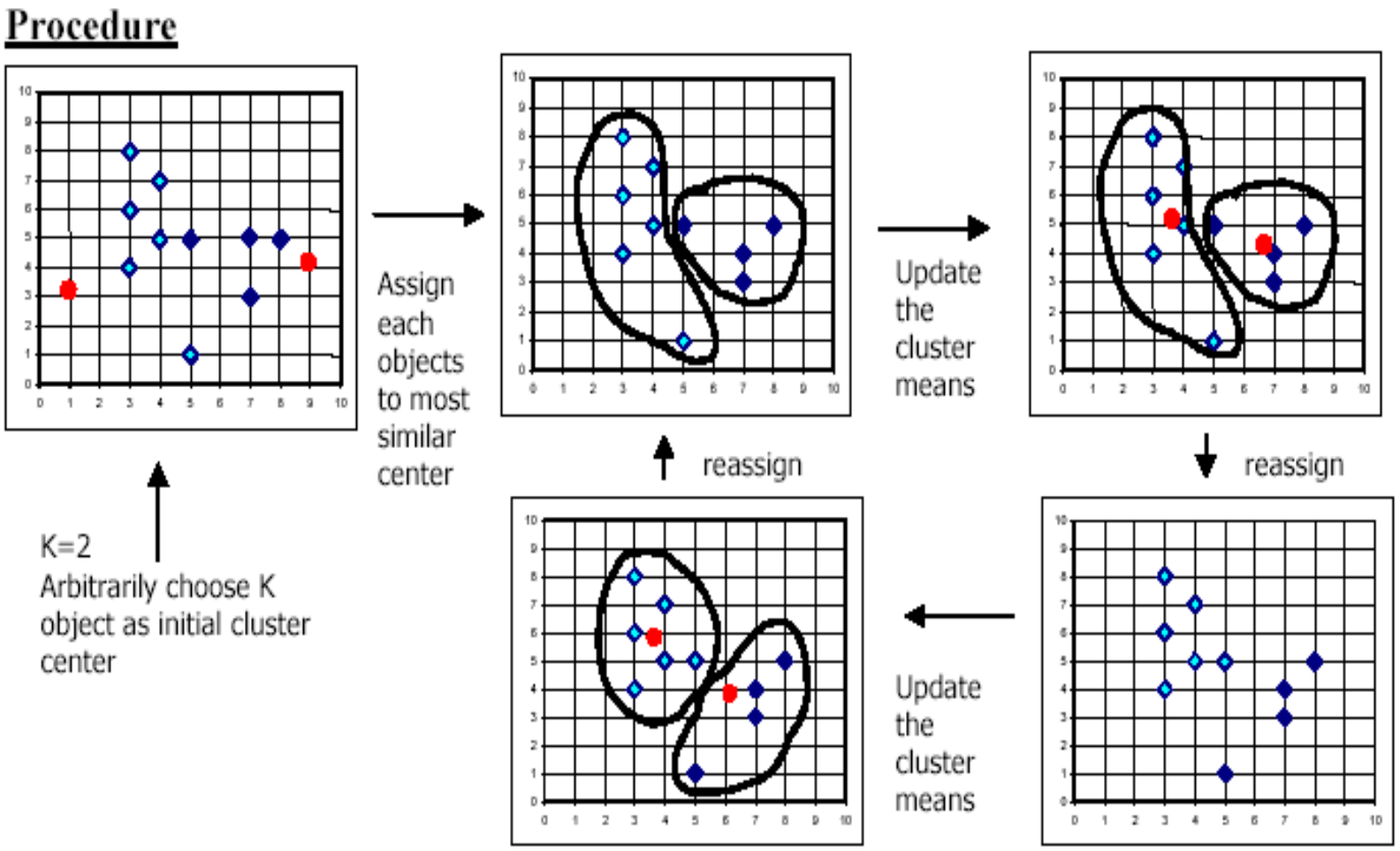
Partitioning Algorithms: Basic Concept

- Partitioning method: Construct a partition of a database ***D*** of ***n*** objects into a set of ***k*** clusters
- Given a *k*, find a partition of *k clusters* that optimizes the chosen partitioning criterion
  - Heuristic methods: *k-means* and *k-medoids* algorithms
  - *k-means* (MacQueen'67): Each cluster is represented by the center of the cluster
  - *k-medoids* or PAM (Partition around medoids): Each cluster is represented by one of the objects in the cluster

The *K-Means* Clustering Method

1. Chose k number of clusters to be determined
2. Chose k objects randomly as the initial cluster centers
3. Repeat
    1. Assign each object to their closest cluster center, using Euclidean distance
    2. Compute new cluster centers, calculate mean point
4. Until
    1. No change in cluster centers or
    2. No object change its clusters

The *K-Means* Clustering Method



K-Means Clustering

Example

Consider the following instances [in the two-dimensional form]

Instance	X	Y
1	1.0	1.5
2	1.0	4.5
3	2.0	1.5
4	2.0	3.5
5	3.0	2.5
6	5.0	6.1

1. If the objects are to be partitioned into 2 clusters then take K=2.
2. Next , chose two points at random representing initial cluster centers:
   Object 1 and 3 are chosen as cluster centers; i.e.
   C1:= (1.0, 1.5) and C2:= (2.0, 1.5) are chosen as the initial centroid
3. Euclidean distance between point i and j
   $D(i - j)=((Xi – Xj)^2 + (Yi – Yj)^2)^{1/2}$

- Initial cluster centers **C1:(1.0,1.5) C2:(2.0,1.5)**
  - For object '1'
    D(C1 – 1) = 0.00 D(C2 –1) = 1.00
    Since D(C1-1)<D(C2-1) the object '1' falls in cluster **C1**

K-Means Clustering

- For object '2'
  D(C1 – 2) = 3.00 D(C2 –2) = 3.16
  Since D(C1-2)<D(C2-2) the object '2' falls in cluster **C1**

- For object '3'
  D(C1 – 3) = 1.00 D(C2 –3) = 0.00
  Since D(C2-3)<D(C1-3) the object '3' falls in cluster **C2**

- For object '4'
  D(C1 – 4) = 2.24 D(C2 –4) = 2.00
  Since D(C2-4)<D(C1-4) the object '4' falls in cluster **C2**

- For object '5'
  D(C1 – 5) = 2.24 D(C2 –5) = 1.41
  Since D(C2-5)<D(C1-5) the object '5' falls in cluster **C2**

- For object '6'
  D(C1 – 6) = 6.02 D(C2 –6) = 5.41
  Since D(C2-6)<D(C1-6) the object '6' falls in cluster **C2**

- Then the cluster C1 and C2 contain the following objects respectively
  C1 : {1,2}
  C2 : {3.4.5.6}

K-Means Clustering

4. Recomputing cluster centers [taking the mean]
   a. for C1:

$$X_{C1} = (1.0+1.0)/2 = 1.0$$
$$Y_{C1} = (1.5+4.5)/2 = 3.0$$

   b. For C2:

$$X_{C2} = (2.0+2.0+3.0+5.0)/4 = 3.0$$
$$Y_{C2} = (1.5+3.5+2.5+6.0)/4 = 3.375$$

Thus the new cluster centers are C1(1.0,3.0) and C2(3.0,3.375)

5) As the cluster centers have changed the algorithm performs another iteration

- New cluster centers C1(1.0,3.0) and C2(3.0,3.375)
  - D(C1 – 1) = 1.50 D(C2 –1) = 2.74
    Object '1' falls in C1
  - D(C1 – 2) = 1.50 D(C2 –2) = 2.29
    Object '2' falls in C1
  - D(C1 – 3) = 1.80 D(C2 –3) = 2.13
    Object '3' falls in C1

K-Means Clustering

- D(C1 – 4) = 1.12 D(C2 –4) = 1.01
  Object '4' falls in C2

- D(C1 – 5) = 2.06 D(C2 –5) = 0.88
  Object '5' will be in C2

- D(C1 – 6) = 5.00 D(C2 –6) = 3.30
  Object '6' will be in C2

- Then the cluster C1 and C2 contain the following objects respectively
  C1 : {1,2.3}
  C2 : {4.5.6}

6. computing new cluster centers
   - For C1:
     $X_{C1} = (1.0+1.0+2.0)/3 = 1.33$
     $Y_{C1} = (1.5+4.5+1.5)/3 = 2.50$
   - For C2:
     $X_{C2} = (2.0+3.0+5.0)/3 = 3.33$
     $Y_{C2} = (3.5+2.5+6.0)/3 = 4.00$
- Thus the new cluster centers are C1(1.33,2.50) and C2(3.33,4.3.00)
- As the cluster centers have changed the algorithm performs another iteration

[repeat the process until there is no change in cluster centers or no object change its cluster]

Weakness of K-means

- Applicable only when *mean* is defined, then what about categorical data?
- Need to specify *K,* the *number* of clusters, in advance
  - run the algorithm with different K values
- Unable to handle noisy data and *outliers*
- Works best when clusters are of approximately of equal size

Hierarchical Clustering

Clustering comes in a form of a tree – *dendrogram* visualizing how data contribute to individual clusters

Clustering is realized in a successive manner through:

- **successive splits, or**
- **successive aggregations**

Hierarchical Clustering

Provides graphical illustration of relationships between the data in the form of *dendrogram*

Dendrogram is a binary tree

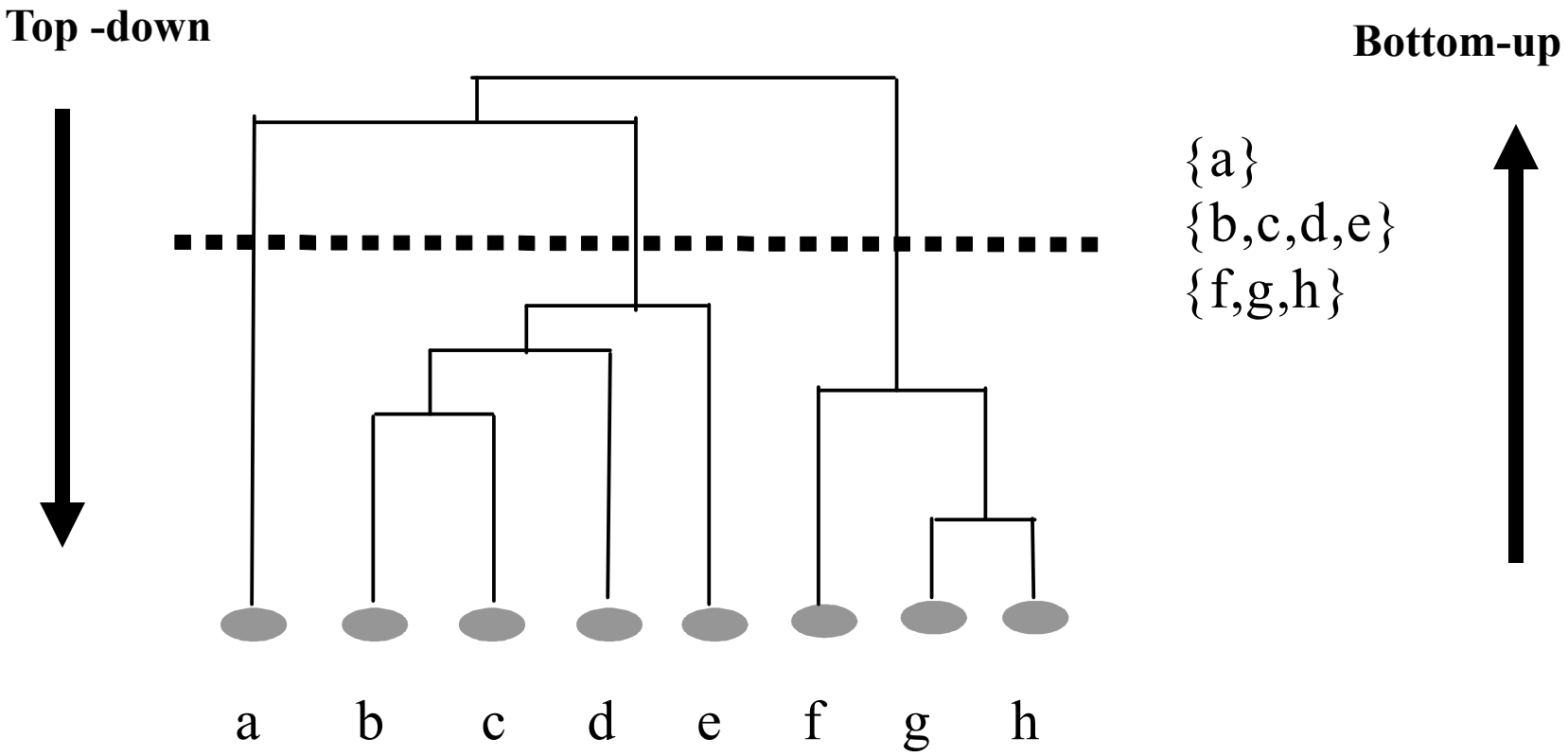
Two fundamental approaches:

- **Bottom – up (agglomerative approach)**
- **Top-down (divisive approach)**

Hierarchical Clustering: Types

- Agglomerative(**Bottom-up or agglomerative)**:
  - starts with as many clusters as there are records, with each cluster having only one record. Then pairs of clusters are successively merged until the number of clusters reduces to k.
  - at each stage, the pair of clusters are merged which are nearest to each other. If the merging is continued, it terminates in the hierarchy of clusters which is built with just a single cluster containing all the records.
- *Divisive algorithm (***Top-down or divisive):** takes the opposite approach from the agglomerative techniques. These starts with all the records in one cluster, and then try to split that clusters into smaller pieces.

Hierarchical Clustering



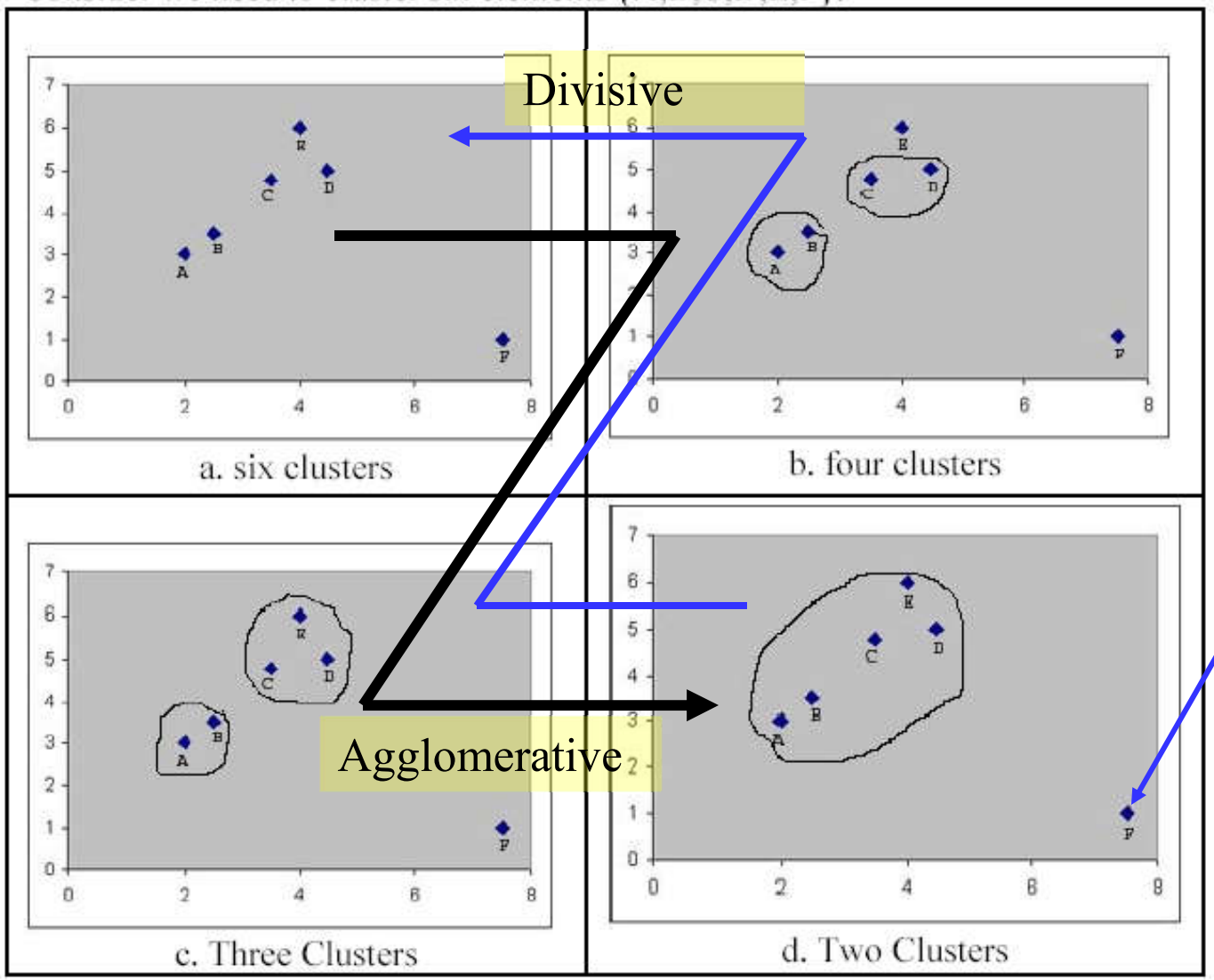
Hierarchical methods

- **Agglomerative methods start with each object in the data forming its own cluster, and then successively merge the clusters until one large cluster is formed (that encompasses the entire dataset)**

- **Divisive methods start by considering the entire data as one cluster and then split up the cluster(s) until each object forms one cluster**

Example of Hierarchical Clustering

Consider we need to cluster six elements {A,B,C,D,E,F}.



Density-Based Clustering Methods (DENCLUE)

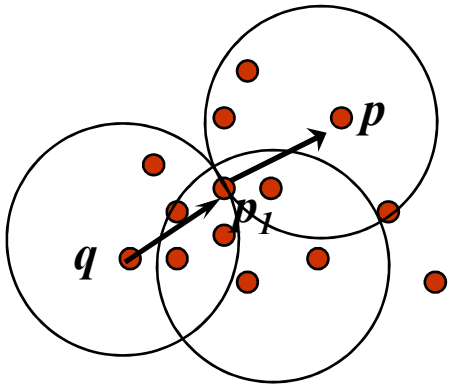
- Clustering based on density (local cluster criterion), such as density-connected points
- Major features:
  - Discover clusters of arbitrary shape
  - Handle noise
  - One scan
  - Need density parameters as termination condition
- Several interesting studies:
  - DBSCAN: Density Based Spatial Clustering of Application Noise
  - OPTICS: Ordering Points to Identify the Clustering Structures
  - CLIQUE : Clustering in Clique

Density-Based Clustering: Background

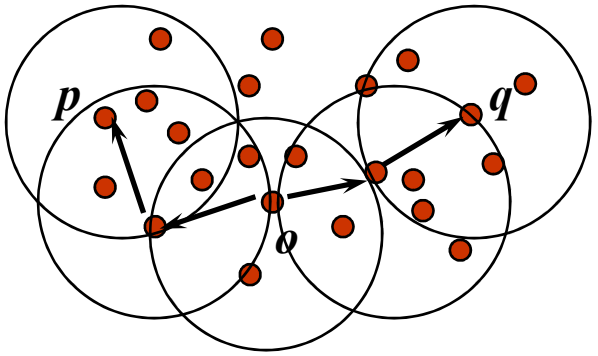
- **The basic terms**
  - **The neighbourhood of an object that enclosed in a circle with radius Eps is called Eps - neighbourhood of that object**
  - **Eps neighbourhood with minimum object points is called core object.**
  - **An object A from a dataset is directly density reachable from object B where A is the member of Eps-neighbourhood of B and B is a core object.**

Density-Based Clustering:

- Density-reachable:
  - A point *p* is density-reachable from a point *q* wrt. *Eps*, *MinPts* if there is a chain of points p_1, …, p_n, $p_1 = q$, $p_n = p$ such that p_{i+1} is directly density-reachable from p_i

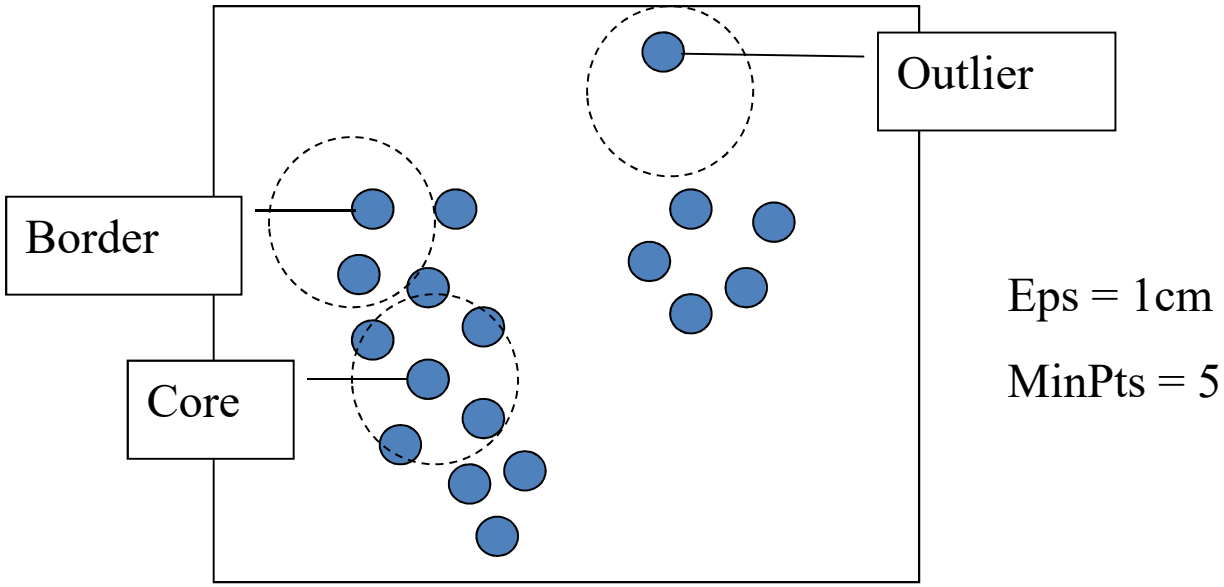


- Density-connected
  - A point *p* is density-connected to a point *q* wrt. *Eps*, *MinPts* if there is a point *o* such that both, *p* and *q* are density-reachable from *o* wrt. *Eps* and *MinPts*.



DBSCAN: Density Based Spatial Clustering of Applications with Noise

- Relies on a *density-based* notion of cluster: A *cluster* is defined as a maximal set of density-connected points
- Discovers clusters of arbitrary shape in spatial databases with noise



DBSCAN: The Algorithm

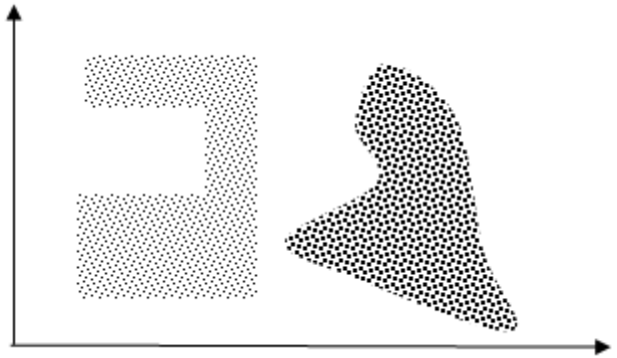
- Arbitrarilyy select a point ***p***
- Retrieve all points density-reachable from ***p*** wrt ***Eps*** and ***MinPts***.
- If ***p*** is a core point, a cluster is formed.
- If ***p*** is a border point, no points are density-reachable from ***p*** and DBSCAN visits the next point of the database.
- Continue the process until all of the points have been processed.

Grid-Based Clustering Method

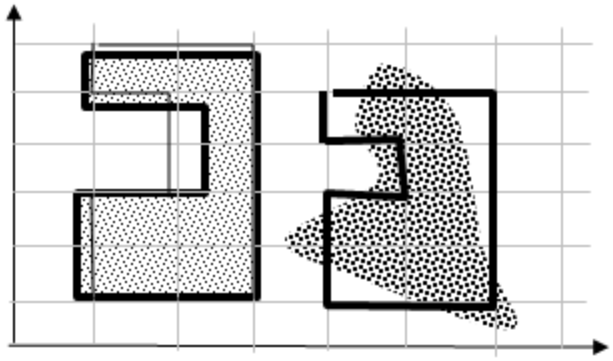
- Using multi-resolution grid data structure
- Several interesting methods
  - STING (a STatistical INformation Grid approach) by Wang, Yang and Muntz (1997)
  - WaveCluster by Sheikholeslami, Chatterjee, and Zhang (VLDB'98)
    - A multi-resolution clustering approach using wavelet method
  - CLIQUE: Agrawal, et al. (SIGMOD'98)

Grid-Based Clustering

Describe structure in data in the language of generic geometric constructs – *hyperboxes* and their combinations



Collection of clusters of different geometry



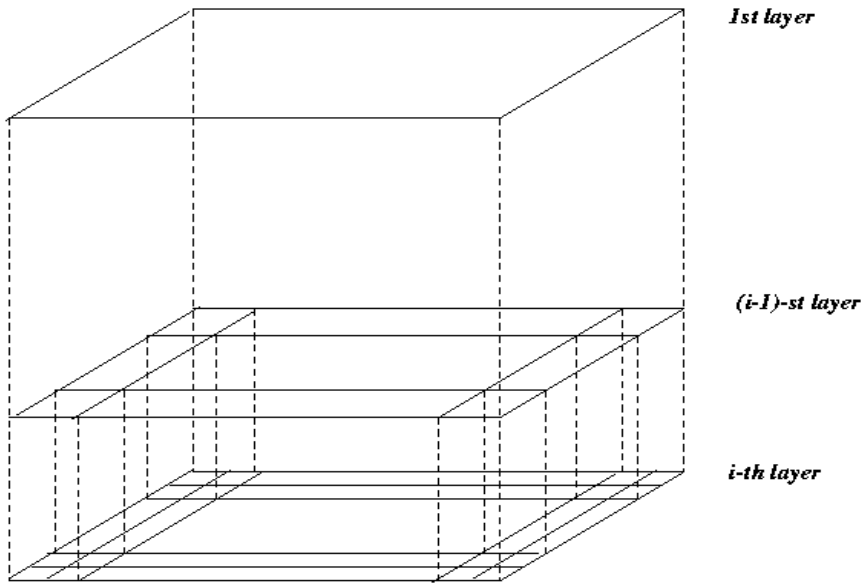
Formation of clusters by merging adjacent hyperboxes of the grid

Grid-Based Clustering Steps

- **Formation of the grid structure**
- **Insertion of data into the grid structure**
- **Computation of the density index of each hyperbox of the grid structure**
- **Sorting the hyperboxes with respect to the values of their density index**
- **Identification of cluster centers (viz. the hyperboxes of the highest density)**
- **Traversal of neighboring hyperboxes and merging process**
- **Choice of the grid:**
  - **too rough grid may not help capture the details of the structure in the data.**
  - **too detailed grid produces a significant computational overhead.**

STING: A Statistical Information Grid

- The spatial area is divided into rectangular cells
- There are several levels of cells corresponding to different levels of resolution



STING: A Statistical Information Grid

- Each cell at a high level is partitioned into a number of smaller cells in the next lower level
- Statistical info of each cell is calculated and stored beforehand and is used to answer queries
- Parameters of higher level cells can be easily calculated from parameters of lower level cell
  - *count*, *mean*, *s*, *min*, *max*
  - type of distribution—normal, *uniform*, etc.
- For each cell in the current level compute the confidence interval

STING: A Statistical Information Grid

- Remove the irrelevant cells from further consideration
- When finish examining the current layer, proceed to the next lower level
- Repeat this process until the bottom layer is reached

- Advantages:
  - Query-independent, easy to parallelize, incremental update
  - *O(K),* where *K* is the number of grid cells at the lowest level

- Disadvantages:
  - All the cluster boundaries are either horizontal or vertical, and no diagonal boundary is detected

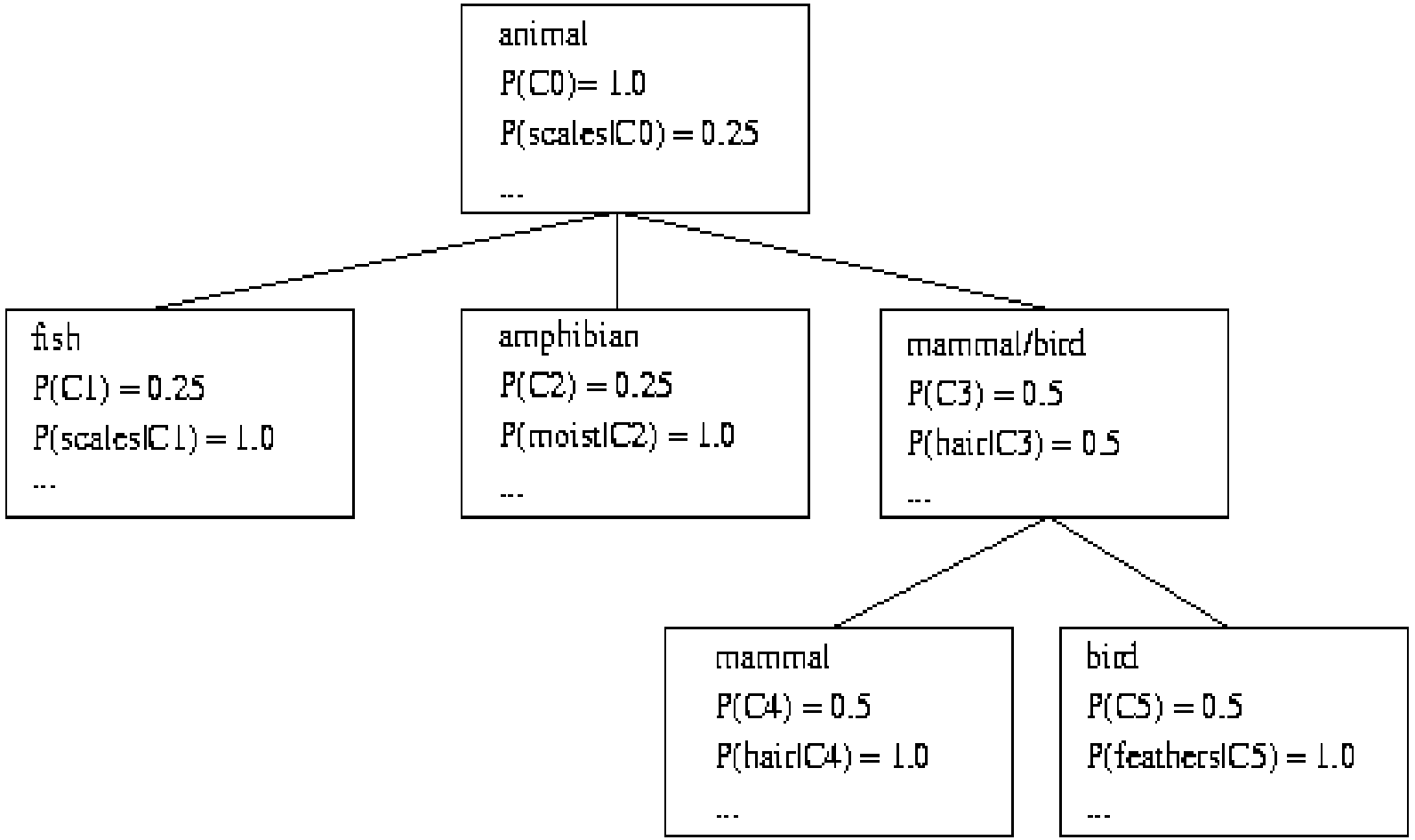
Model-Based Clustering Methods

- Attempt to optimize the fit between the data and some mathematical model

- Statistical and AI approach
  - COBWEB (Fisher'87)
    - A popular a simple method of incremental conceptual learning
    - Creates a hierarchical clustering in the form of a classification tree
    - Each node refers to a concept and contains a probabilistic description of that concept

COBWEB Clustering Method

A classification tree



Summary

- Cluster analysis groups objects based on their similarity and has wide applications

- Clustering algorithms can be categorized into partitioning methods, hierarchical methods, density-based methods, grid-based methods, and model-based methods

- Outlier detection and analysis are very useful for fraud detection, etc. and can be performed by statistical, distance-based or deviation-based approaches