Introduction To Data Mining

1.1 Motivation & Importance:

The information industry in recent years contains the huge amount of data. This huge amount of data must be transformed into useful information & knowledge. This knowledge & information used in many systems like market analysis, business Management, production & analysis, Science & Engineering etc., This is the motivation behind the data mining technology.

The evolution of database technology contains the functions like data collection & database creation, data management (store, Retrive & transaction process) & data analysis & understanding

The database evolution is shown in below diagram.

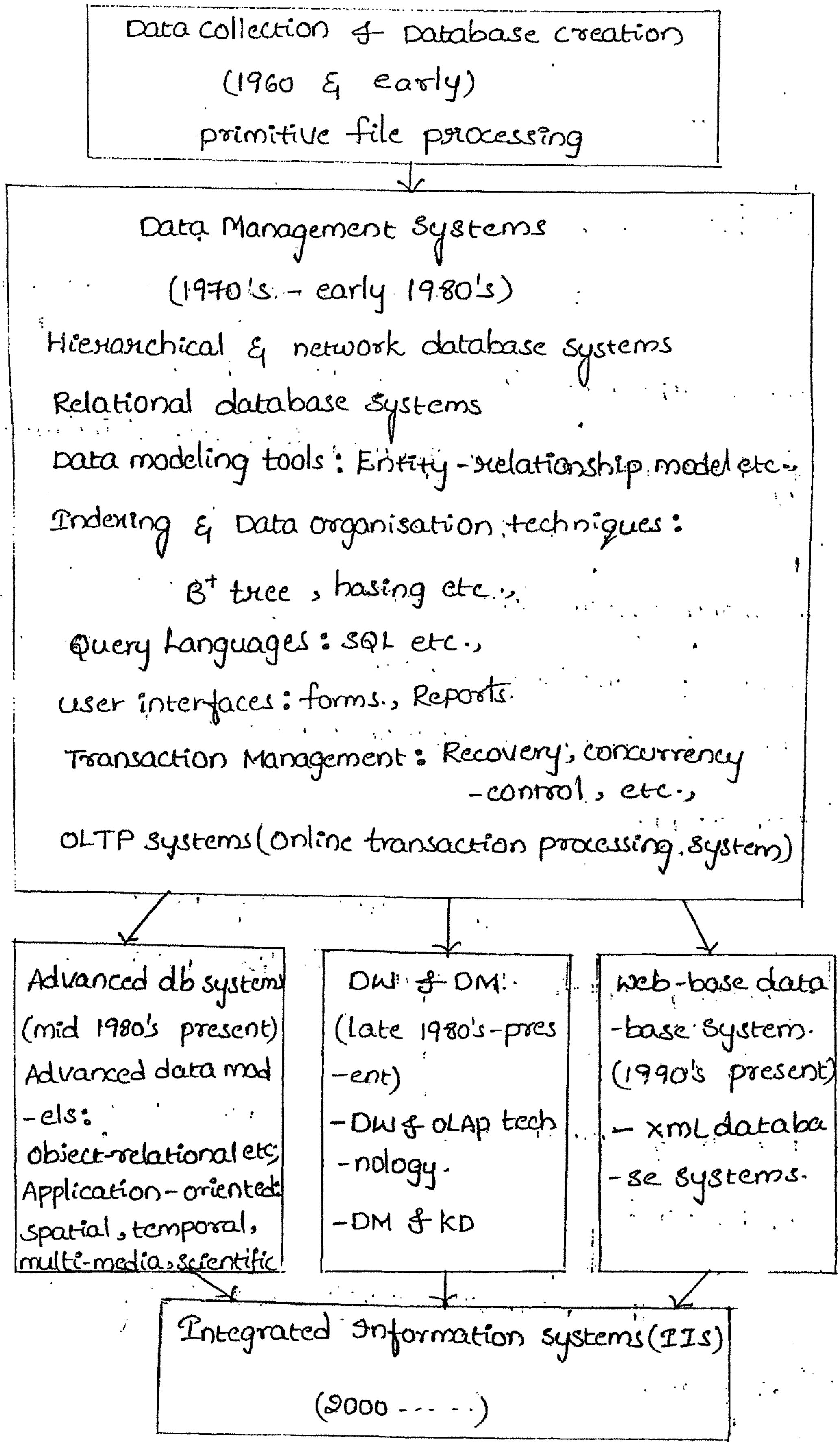


fig 1.1 The Evolution of database technology

In the year of 1960 & Early data is stored in the form of primitive file processing. This is used as a base of the rest of the database systems.

In the year of 1970, db systems are developed. The initial db systems are Hierarchical & network, later relational db systems, data modeling techniques, indexing & data organisation & query languages are developed. After that to provide Easy understanding to user several user interfaces, form reports & query processing & query optimization, & transaction management techniques are developed.

In the year of early 1980's OLTP (Online transaction processing systems are developed). In the year of mid 1980's advanced db systems are developed like advanced data model, application oriented db systems are developed.

In the year of late 1980's Data ware-housing DW has been developed. It is a reposit-ry of hetrogeneous data sources organised und

tc..

m)

data
em.
esent)
taba
ms.

a schema to support management decision.

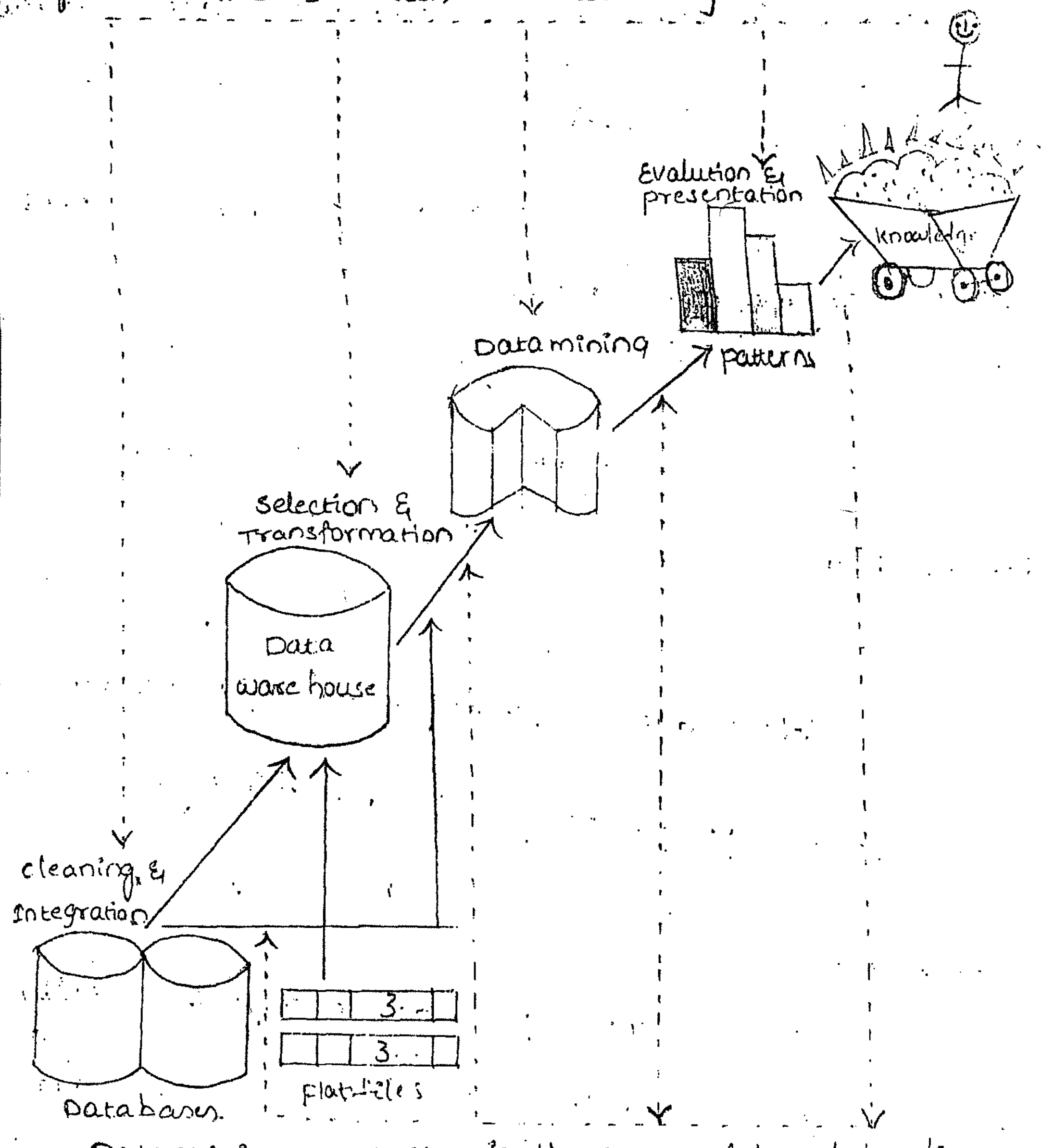
Data Warehouse contains the OLAP (online analytical processing) technology. OLAP contains the functions like summarization & aggregation. Later on data mining & knowledge discovery techni-ques are developed.

In the year of 1990 web based db systems are developed. By integrating above three systems we get integrated information systems. This systems developed in the year of 2000.

Therefore, the information industry contains the large amount of data & also data is stored in different sources that exceeds human ability. Therefor-e, we need data mining tools to analyse & search for interesting patterns or knowledge. This knowledge is used by the management to take Accurate deci-sions.

1.2 What is data mining:

Data mining refers to extracting or mining data from large databases some of the people think that data mining is a synonym of KDD knowledge discov-ery in databases & also some of the people think that data mining as a essential step in the the process of KDD. This is shown in below diagram.



Data mining as a step in the process of knowledge discove

1. Data cleaning:

The data from the large hetrogeneous databases are retrieved & noisy data or inconsistent data is removed before going to the next step.

2. Data Integration:

The data from the different sources are integrated & then this data is loaded into data warehouse

3. Data warehouse: (DWH)

It is a centralized repository it contains all the organizations data.

4. Data Selection:

The selected data is retrieved from the data warehouse.

5. Data Transformation:

Here data is transformed into a form that must be appropriate for the data mining.

6. Data Mining:

It is essential process. It contains intelligent techniques or methods like summarization, aggregation etc..

7. **pattern Evolution:**

Here interesting patterns are Evolutied.

8. **knowledge presentation:**

Here visualization techniques are applied to present mined knowledge to the user.

Based on this the typical data mining architecture contains the following components.

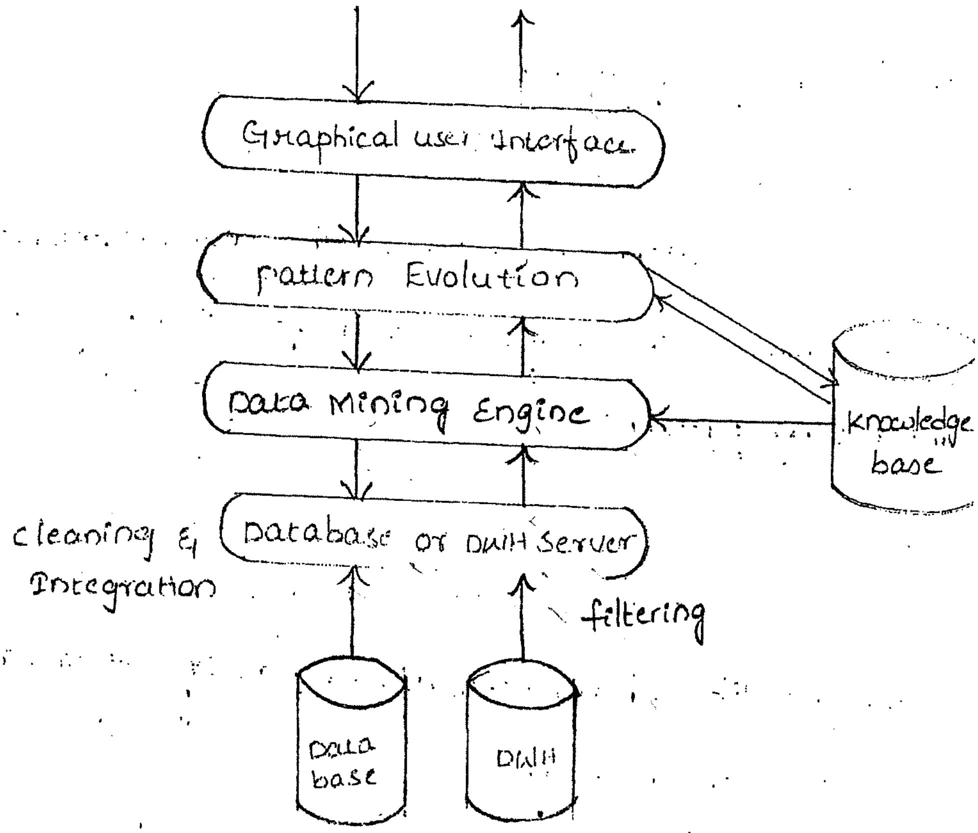


fig 1.3 : Typical datamining Architecture

1. Database, DWH or Any other information Repository:

Here data is retrieved from any data-base, or data ware house or any other information repository we apply data cleaning & integration technique before data is going to database ~~or~~ DWH server.

2. Database or DWH server:

It contains all the data according to the user specification.

3. knowledge base:

Here we retrieve the domain knowledge & this domain knowledge is used for searching the interesting patterns

4. Data Mining Engine:

It is a essential process. It contains technique like characterization, classification, association & cluster analysis etc.

5. pattern Evolution:

Here interesting patterns are evoluated.

itory:

y data

ation

on tech

ult server.

ling to

wledge

the

ins

, associa

4.

6. Graphical user interface:

It provides the communication between user & data mining system through this end user can present the queries to datamining system.

1.3 Data Mining - on what kind of data:

Data mining can be applicable to any information repository. This information repository may be a relational databases, DWH, transactional database advanced database systems, & Advanced database applications.

1.3.1 Relational Databases:

Relation database system consists of tables & also each table contains the unique name. Each table consists of set of attributes or columns & generally stores large set of data in the form of tuples or records or rows. Aswellas each row is uniquely accessed by using a key.

Example:

For example, relation database tables for All Electronics Company contains the tables like customer, item employee, purchases etc., Customer

cust-Id	name	address	age	income	....
C1	smith	123 Ham.st canada	21	$25,000	

Here customer table contains the attributes like cust-Id, name, address, age, income etc.,

Item

Item-Id	name	brand	category	type	place-made	cost
I3	high res-TV	Toshiba	high resolution	TV	Japan	$2000

Employee

emp_id	name	category	group	salary
E5	John	home, entertainment	manager	$10,000

Branch

Branch-id	name	address
b1	city square	123 main st, Tomato, canada

purchases

trans-id	cust-id	item-id	date	time	cost
T100	C1	I3	29/06/09	15:45	$100

It
fi
rel
etc
ava
In
us
1.3
un
load
-se
-ti
ALL

Item_Sold

Trans_id	Item_id	quantity
T100	I3	1

fig 1.4 Relational tables for AllElectronics database

Here some of the tables represents the relation between multiple tables like purchases, item_sold etc.,

The Relation database is a most popularly available & rich information repository in datamining. In Relation databases data modelling is done by using ER-model

1.3.2 Data Ware house (DWH):

It is a centralized repository organised under a schema to support management decision. we load data into "data warehouse" i.e., the data warehouse is constructed by data cleaning, data transformation, data integration & data load.

for example, typical data warehouse for ALL Electronics company.

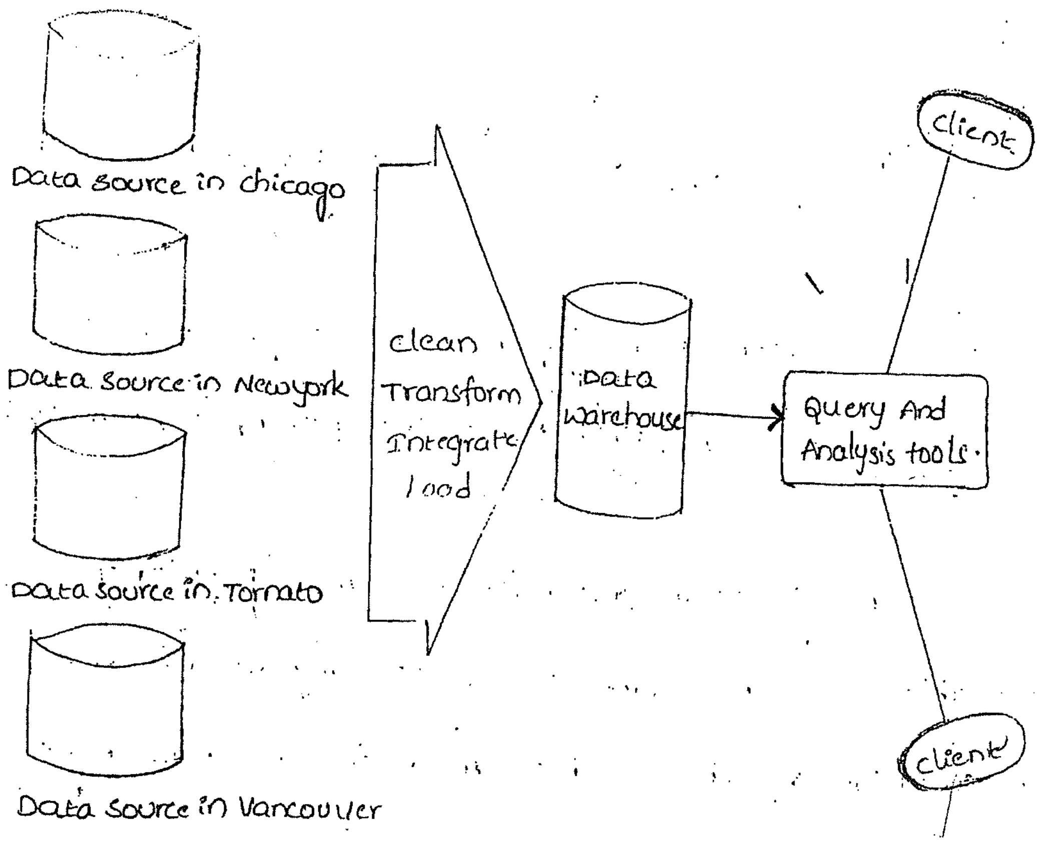


fig 1.5 Typical Architecture of Data Warehouse for All Electronics company.

Here AllElectronics company has branches All around the world. Therefore data from different branches first of all cleaned, transformed, integrated & finally loa-ded into data warehouse, then we apply query & analysis tools before going to present the data to the management.

Data warehouse contains the historical data i.e.) 5 to 10 years of data & used to support management decision. In data warehouse data modeling is done by using schema but the physical architecture of data warehouse contains the "data cube". Through this data cube "multi-dimensional view of data is presented."

Example

Data cube for summarized sales data of "ALLElectronics" is shown below. This data cube contains the three dimensions i.e., address (it contains city values Chicago, Newyork, Toronto, Vancouver), Time (Quater values Q_1, Q_2, Q_3 & Q_4) & item (it contains item typer like home entertainment, computer phones, security).

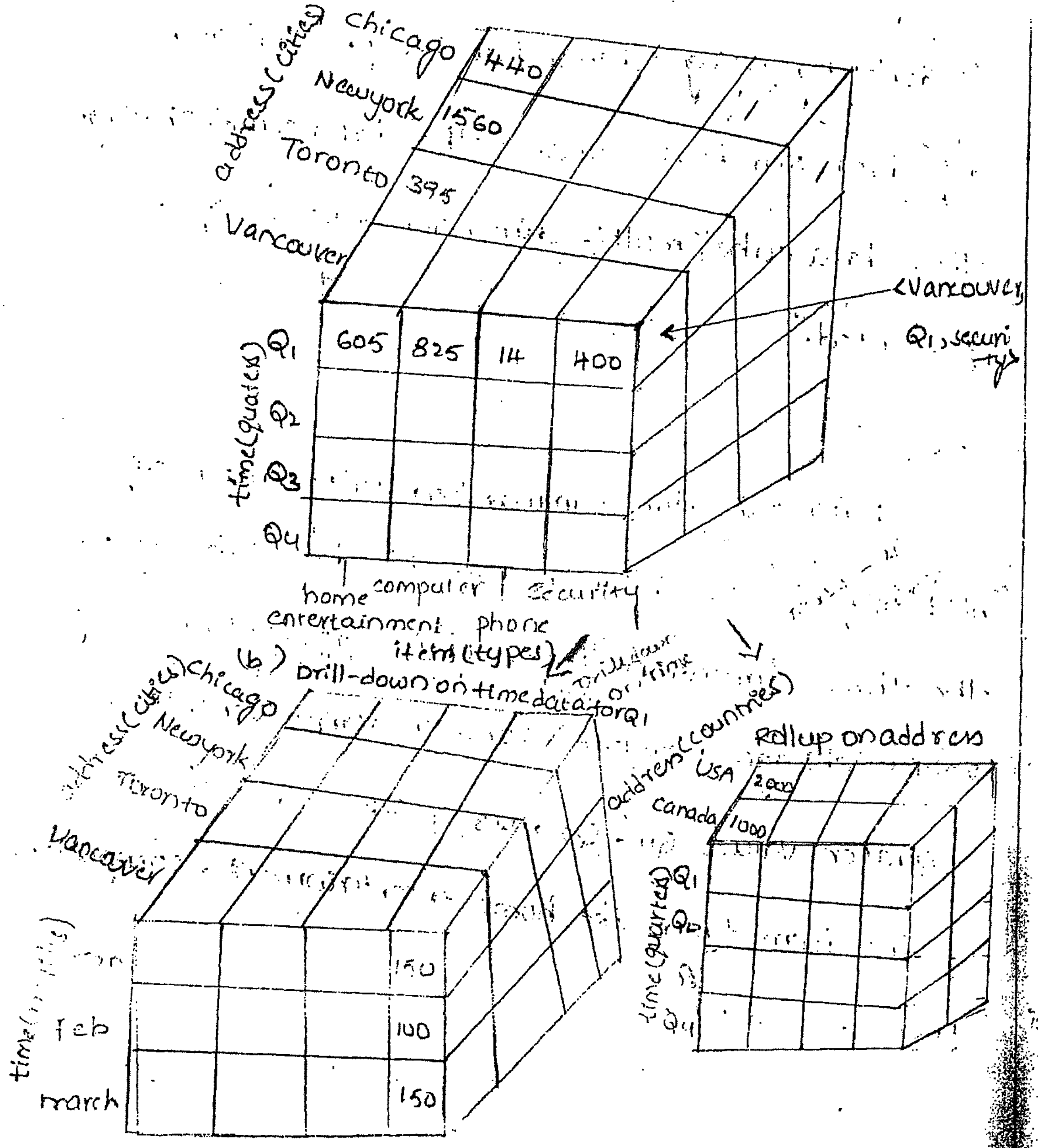


fig 1.6 multi-dimensional data cube for AllElectronics

1.6 (a) : summarized sales Data

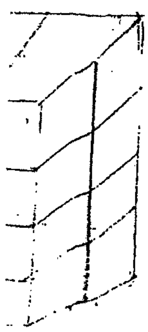
(b) : Drill Down & roll up operations

Here data warehouse contains the data about all the subiects & all the organizations. Therefore, its scope is enterprise-wide. On the other hand datamart, it is a subiect of data warehouse. It contains the single subiect area & its scope is department-wide.

<Vancouver, Q_1, security>

Data warehouse provides multi-dimensional view & summarized data. Therefore, data warehouse is well suited to OLAP (online Analytical processing).

address



OLAP contains the techniques like "drill-down" & "roll up".

In fig 1.6(b) drill-down on the time data for Q_1 by monthly & also rollup on address by countries.

Electronics

1.3.3 Transactional Databases:

Transactional databases contains the files where each record represents a transaction. A transaction uniquely identified by the transaction-i trans-id. These transactions are can be stored in tables & each record/row represents transaction.

The fragment of transactional database for All Electronics is shown in below.

Sales

Trans-id	list of item_id
T100	I1, I3, I8
....	....

fig 1.7 Fragment of Transactional database for sales at All Electronics.

Here the sales table contains the nested relation i.e., it contains the list of item-id's

1.3.4. Advanced Database systems & Advanced Database applications:

The new applications like spacial data (such as maps), Engineering & Design (such as designing buildings, Designing components, Integrating circuits), Text & multimedia (such as audio, video, images) Temporal & time related data (such as historical data & stock exchange data) & world wide web (widely distributed information repositories on the internet) etc.,

Ob

-ms

Her

the

-ap

(i)

(ii)

ano

(iii)

imp

Thi

thi

act

Ob

are

Object - oriented database systems :

These object oriented database systems mainly based on object oriented paradigm. Here each entity is treated as object, In this the data & the code related to object are Encapsulated into single unit.

Each object contains the following.

(i) set of variables that describes the object :

This is similar to attributes in ER-model

(ii) set of messages to communicate with one object to another object.

(iii) set of methods each method contains the code to implement message.

The similar objects are combined into class. This class is known as object class then each object of this class is an instance for that class.

Eg : employee class contains variables like name, address, sallary etc.,

Object - related database systems :

The object related database systems are constructed from object relational data model.

These models are extended from relational model & also new data types has been added to handle the complex object. The constructors also defined to handle this added data types in relational query languages.

In data mining system the objects oriented database systems & object related database systems share some similarties.

Spacial database systems:

Spacial database systems contains the data in the form of geographic maps & are represented in the form of Raster format with n-dimensional bit maps or pixel maps.

For example, 2D satellite image is represen-ted in raster format, where each pixel represents the rain fall in this specific area.

The maps are also represented in Vector form i.e., where buildings, roads etc., are represented in the form of points, polyline or polygon.

del &
the
to handle
nguages.
iects
latabase

Text & multimedia Database Systems:

Text database systems contains long sentences, paragraphs to specify product specification or error or bug specification or summary report specification etc., These text database systems are unstructured.

Multimedia databases contains the data in the form of audio, video, images. These are used in applications like voice message systems, video on demand systems, world wide web information systems etc.,

data
ented
ional

Temporal & Time-series Database systems:

These two database systems contains the set of data. This data continuously changes with time like stock exchange data.

represen
present

World Wide Web information systems:

These are the widely distributed information systems like yahoo or America online or online service. These are linked with objects to exchange the information through internet i.e., the end-

d in
, are
or

user can get these services through internet & also end user get attractive webpages. These web pages are un-structured i.e., doesn't contain any sche-ma or pattern.

1.4 Data Mining functionalities :

Data mining functionalities are used to specify the kind of patterns to be mined found in data mining tasks.

Data mining tasks mainly classified into

1. Descriptive
2. predictive

"Descriptive means it explains the general characteristics of a data in database".

"predictive means it provides conclusion on current data through prediction".

Here enduser can go for the search for ultiple patterns because end user does not know

Wh

mul-

-ng

hints

pat

int

or

cl

p

-d

or

(What kind of patterns are interesting.)

Therefore, data mining system provides the multiple search facility & also provides the interesting patterns are search in various granularities (levels).

Therefore, datamining system provides the hints to the end user to search for the interesting patterns.

The data mining functionalities mainly classified into

1.4.1 concept/ class description: characterization & Descrimination:

The data is associated with the concept or a class. for example in AllElectronics database class of items for sale contains the computers & printers.

similarly concept of customers include big spenders & budget spenders.

Therefore, the data associated with concept or class is called as concept/class description. These descriptions are derived via

1. Data characterization
2. Data discrimination
3. Both Data characterization & Data discrimination.

1. Data characterization:

It is the summarization of general characteristics or features of a target class.

Data characterization can be achieved through several techniques.

for ex, Rollup operation in OLAP.

once the data characterization is completed then data can be presented in various forms "bar charts, pie charts, multi-dimensional datacubes".

2. Data Descrimination:

It is the comparison of target class with one or more constructing classes. This is called as the "data descrimination".

Here End user specifies the targets classes & constructing classes & also End user can the data from database by using queries:

3. Data characterization & Data Descrimination:

Note: we have to write both (1) & (2) here

1.4.2 Association Analysis:

The association analysis is can be used to identify the association rule between each pair of attributes.

This association analysis mainly used in transactional data analysis.

The association rule $X \Rightarrow Y$ i.e.,

"$A_1 \wedge A_2 \ldots\ldots A_m \longrightarrow B_1 \wedge B_2 \ldots\ldots B_n$" where

A_i for $i \in \{1, \ldots m\}$ and B_j for $j \in \{1, \ldots n\}$ for pair of attributes.

For Example, AllElectronics database contains the association rule

$$age(X, \text{"20...29"}) \wedge income(X, 20k - 30k) \Rightarrow buys(X, \text{"CD player"})$$

[support = 2% confidence = 60%]

as mentioned in the above Ex, a person whos age between 20 to 29 & whose income will be 20k to 30k can buy or purchase a CD player at AllElectronics company. With support 2% & confid-ence 60%.

(or)

This association rule indicates that $x \Rightarrow y$ can be interpreted as customers 2% support are 20 to 29 years of age with income 20k to 30k buys or purchases a CD player at AllElectronics company.

There is a probability confidence 60% of this age group & this ~~co~~ income can buy CD player.

1.4.3 Classification & prediction:

The classification is the process of finding set of models or functions that describes class i.e., the classification mainly used to identify class labels. But many of the applications the end user search for the data. In most of the cases usually we search numerical data.

Therefore search for numerical data

in database is called "prediction".

1.4.4 Cluster Analysis:

Cluster Analysis is nothing but identifying the data object with respect to our requirement; we have to also examine the data object which are not relevant to our clan label.

By performing cluster analysis, we can easily identify similarity & dissimilarity of class label. so that we can group the similar objects & we can leave the non-similar objects which is represen-ted in below diagram.

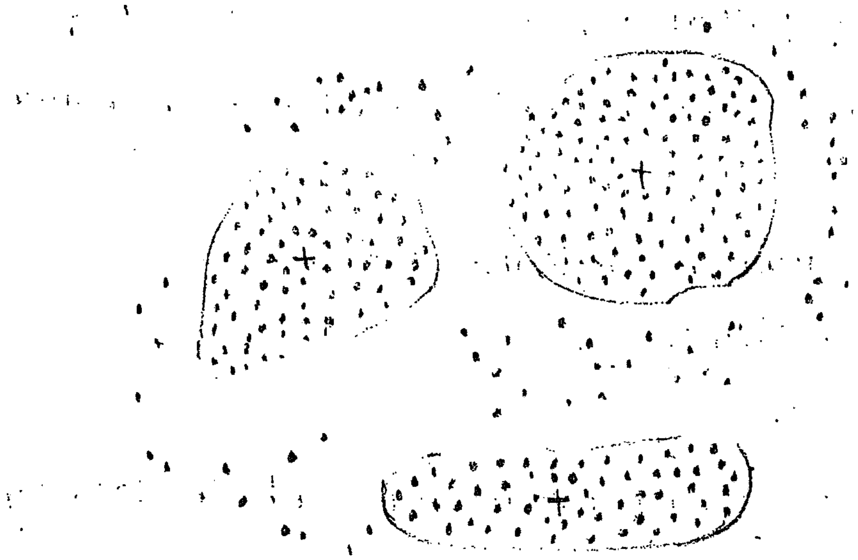


fig 1.8
A 2-D plot of customer data with respect to custom-er locations in a city, showing three data clusters. Each cluster "center" is marked with a "+"

1.4.5 Evolution Analysis:

The data evolution analysis describes data objects those are changing with time.

These data objects are evoluted through charaterization, descrimination, association analysis & cluster analysis.

1.5 Are All of the patterns Interesting?:

(or)

Interestingness of a pattern:

The datamining system capable of providing 1000's of patterns. A pattern is interesting if

* That must be understandable to the human beings.
* It must be applicable for new data.
* potentially useful.
* Novel.

Interesting pattern represents the knowledge. Several objective measures for interestingness of a pattern. In these association rule is one of the method. Therefore association rule $(X \Rightarrow Y)$ then support $(X \Rightarrow Y) = P(X \cup Y)$, where $X \cup Y$ indicates that a transaction contains both X & Y i.e., the union of item X & item Y.

Confidence $(X \Rightarrow Y) = P(Y/X)$ conditional probability i.e., the probability that a transaction containing X also contains Y.

1.6 Classification of datamining systems:

The classification of data mining system is an integration of multiple disciplines (fields). This is shown in below

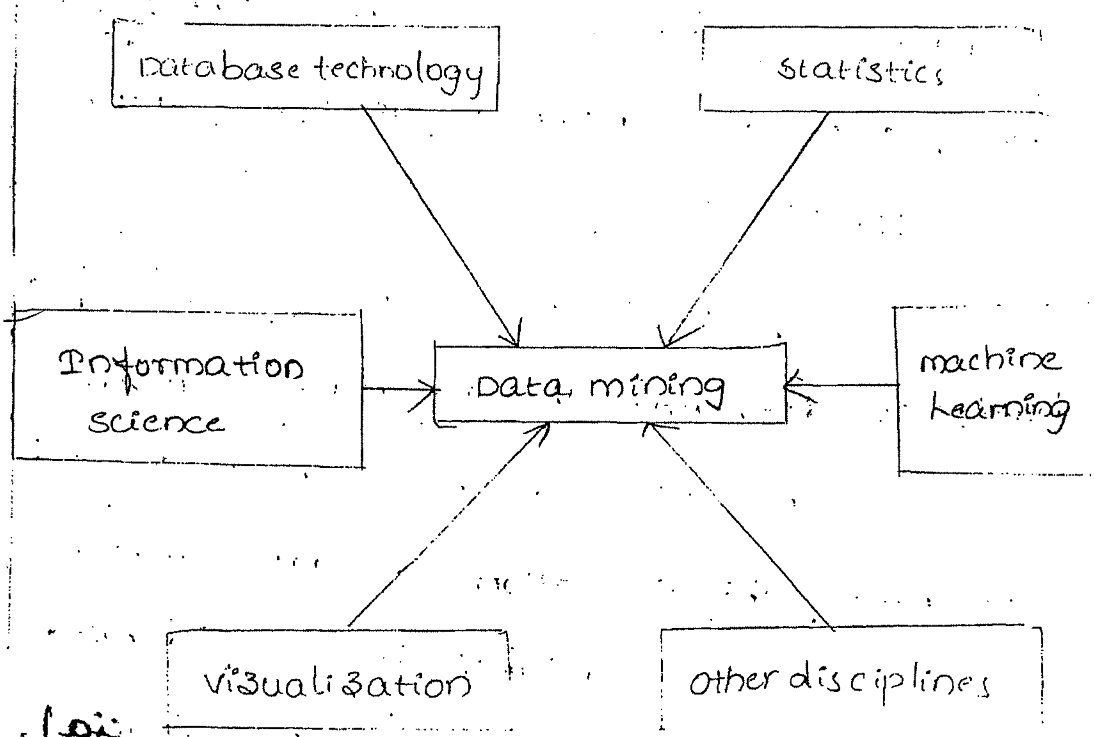


fig 1.9 Classification data mining systems

Here datamining system is integrated with database Technology, statistics, machine learning, vizualization, information science etc.,

Therefore datamining system is classified as following

1. Classification According to kind of database it is mined:

The data mining system is classified based on database mined. But the database technology is classified based on data model or type of data (i.e., application oriented).

If the database Technology is classified based on data model this may be a relational database, transactional database, object-oriented database, object relational or data warehouse.

If the database is classified based on the type of data then it may be a spacial, temporal & time series, text & multi-media etc., systems.

2. Classification according to the kind of knowledge it is mined :

Data mining system is classified based on knowledge mined i.e., we use the datamining functi-nalities like concept or class description, association analysis, cluster analysis & Evolution analysis.

Data mining system is also classif-ed based on knowledge mined at different levels. or top level or primitive level ~~to~~ (low level or raw level) or multiple levels.

3. Classification according to kind of techniques utilized :

The datamining system is classifed base on techniques utilized. These techniques describes the degree of user interaction involved.

For example query-driven systems, autonomo-us systems, interactive systems etc., or methods for data analysis.

4. classification according to kinds of applications adapted :

The datamining system is classified based on applications adapted like finance database system, tele communication database system, share market database system etc.,

1.7 Major issues of datamining:

These are mainly classified into

1. mining methodology & user interaction issues.
2. performance issues
3. Issues related to database types.

1. mining methodology & user interaction issues:

These specifies the kinds of knowledge mined, knowledge mined at different levels, use of domain knowledge & knowledge presentation.

Mining knowledge from Databases:

Different users require different

kinds of knowledge. Therefore, datamining system must provide wide range of data analysis & also knowledge discovery through different datamining techniques, like data characterization, data discri-mination, association analysis, cluster analysis & Evolution analysis.

1.2 mining knowledge at different levels:

Here enduser interact with the data mining system & use the different OLAP (online analytical processing) tech-niques like drill down & rollup, through this techniques end user mined the knowledge at different levels.

1.3 Background knowledge:

This background knowledge guides the discovery process if End user has a background know-edge about the database like the constraints or association rules or conditions then this datamini-g process is speedup.

1.5 presentation & visualization:

The mined knowledge is presented to the enduser & this must be understanble to the enduser. Therefore, we use the several vizualization techniques like trees, graphs, charts, matrix etc.,

1.4 Data mining query languages for datamining Tasks:

Like the relational query language, for example SQL. In SQL we present. Adhoc (temporary query) Query & get the data, similarly datamining query langu-ages has to be developed & it must support end user Adhoc datamining tasks, like data analy-sis, get the domain knowledge, understand the constraints & conditions etc.,

1.6 Handling noisy (or) incomplete data:

Generally databases contains the noisy or incomplete data. Therefore, we use the datamining techniques like data cleaning &

data analysis to avoid this noisy or incomplet-ed data.

1.7 Pattern Evolution:

Still the data mining system uncover thousands of patterns & also many of the discov-everd patterns uninteresting to the user. Therefo-re, end user has to specify the measures or cons-traints to reduce the search criteria.

2. Performance Issues:

These issues contains the efficiency, scalability & parallelization of data mining algorithms.

2.1 Efficiency & scalability of datamining Algorithms:

We have to retrieve the data from large databases. Therefore, the data mining algo-rithms must be efficient & scalable.

In datamining system knowledge discovery, Efficiency & scalability are the main key terms.

2.2 parallel, distributed & Incremental data mining algorithms:

End user has to access the data parallely from different data sources & also data is distributed to different data sources. These two are done by using parallel & distributed datamining algorithms. In these algorithms data is divided into partitioned then partition processed paralley & then result of this partitions are merged.

The incremental datamining algorithms are mainly used to reduce the cost of data mining process & also update the data in database without performing the search from the begining of DataBase.

3. Issues related to database types:

3.1 Handling relational & complex data:

The relational data is handled

mining

xarallely

distribu

, are

nining

ded into

ey & then

g algori

of data

databases

ining of

efficiently by using relational database syste data warehouse systems.

The complex data like spatial data, tex & multi-media data, time series data. These complex data is handled by using systems spac -al db systems, text & multi-media systems, time series db systems etc.,

3.2 Mining data from heterogeneous databases

Mining the data like LAN systems, WAN systems, distributed systems & Hetrogeneous database systems is only possible through data minin System. This datamining system also provides & improves the information exchange & interoperatabi -ty in hetrogeneous databases.

Chapter - 2

Data Warehouse And OLAP Technology for Data Mining

Basic concepts of data warehouse :-

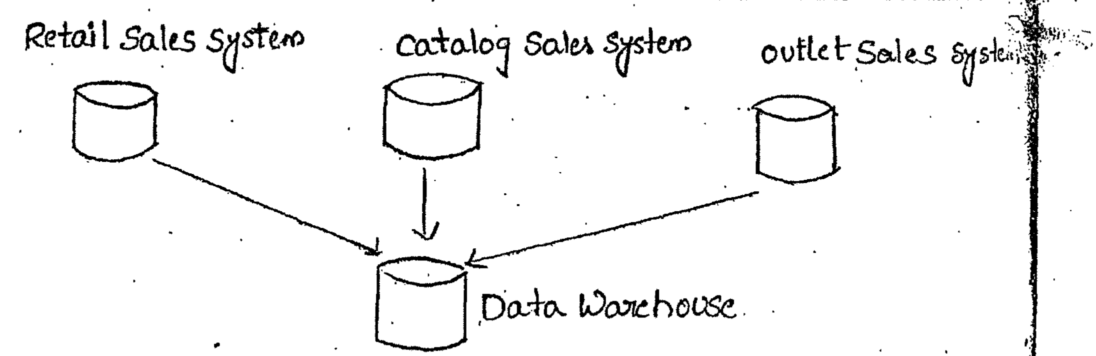
(i) What is Data Warehouse

According to warehouse Inman, "A Data - Warehouse is a Subject Oriented, Integrated, non - Volatile, Time - Variant collection of data, to support management decisions".

Subject Oriented:

For Example, consider the retailer, these retailer contains the Database systems like Retail Database System, Catalog Database system and Outlet Database Systems. These database systems individually Support for different queries. But a user want to run a query in all the sales. This is only possible through data - Ware house. therefore Data warehouse Organizes Subject areas like customer, Products, suppliers, Sales etc. Here the subject area ~~is "sales"~~. This is shown in the following diagram.

V. Vijaya Lakshmi



Sales for Data Warehouse

fig (2.1.1) Subj-Oriented Sales information

Integrated :

Here, different data Sources data are integrated. Then this data loaded it into Data-Warehouse. These sources may be relationa databases, flat files, Excel Sheets, Online Transaction records, etc. Before the data is loaded it into DWH, we apply data cleaning and data analysis.

In the above example, 3 data Source, data are integrated then we get the unique key. Using ~~this unique~~ key, we uniquely identify the data record in dataware. This is shown in the following diagram.

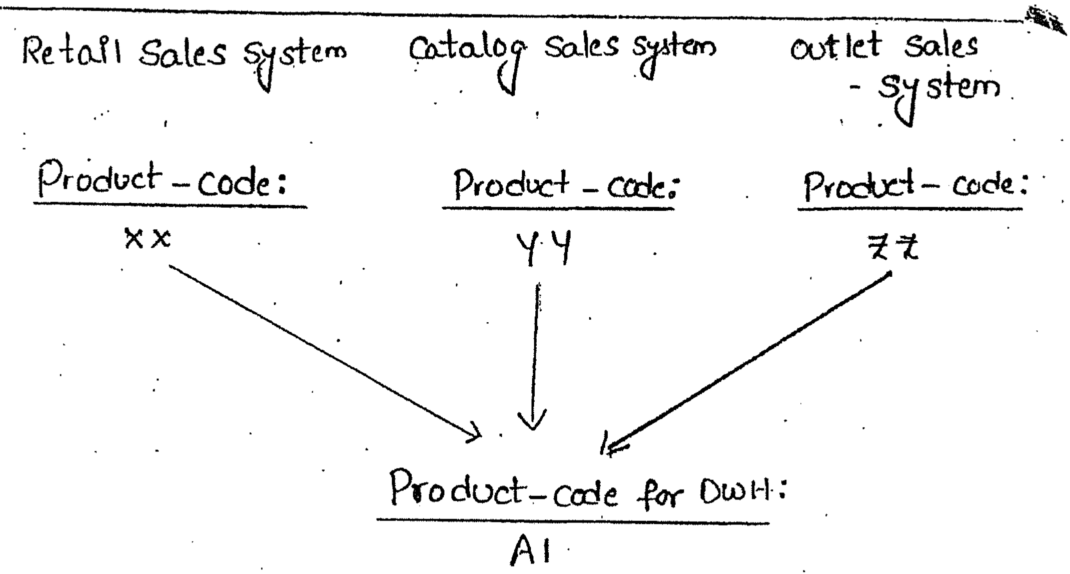


fig (2.1.2). Integrated information for Sales

Non-Volatile:

The non-Volatile means read only. The DWH User always read the data. But the OLTP user can read the data as well as write the data. This is shown in below.

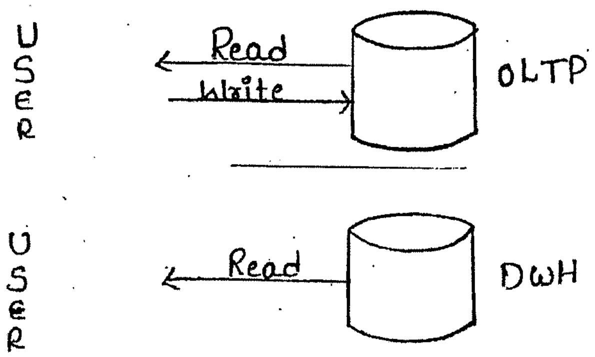


fig (2.1.3) Non-Volatile

Time - Variant:

The DWH contains the historical data i.e; 5 to, 10 years of data. But the OLTP System contains the current data.

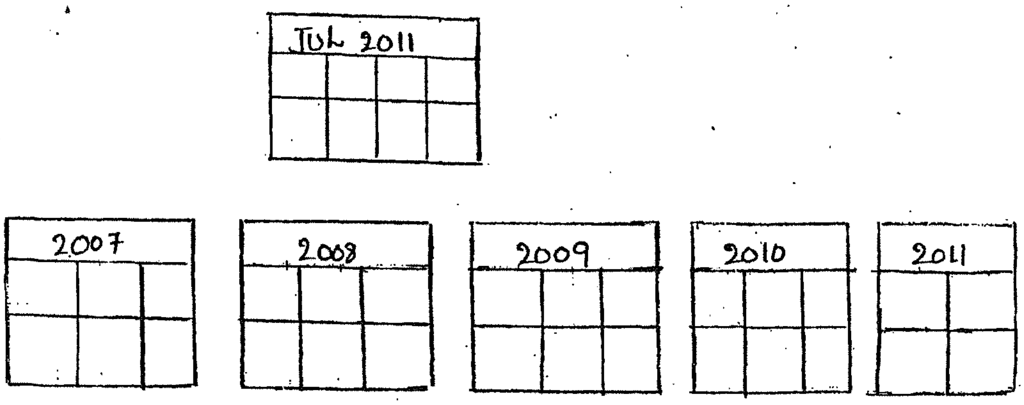


fig (2.1.4). Time Variant

Finally we extract the data from the DWH & then we apply the different Visualization Techniques like Trees, Graphs, Charts, Tables, etc. to present the data to management. The management using this data they take the accurate decision.

2.1.1 Differences between operational database systems & Data warehouses:-

The online operational database systems mainly Used for Online transactions & Query processing. These include the transactions like sales, payrole, marketing, manufacturing, etc. These type of systems are called as "online Transaction processing" (OLTP) systems.

These systems mainly used for day to day Transactions.

The DWH contains the historical data. This is used for data modelling and data analysis. This is used for decision making. This type of systems are called as "online Analytical processing (OLAP) systems.

Advantages of Data Warehouse:-

1. It makes the data permenant.
2. It makes the data accessable.
3. It identifies hidden business operations.
4. It improves the customer Relationship.
5. It provides the security.

Differences between OLTP and OLAP:-

1. Users and System Orientation:

The OLTP system is customer orientation to support the Online Transaction.

The OLAP system is market orientation to support decision making.

2. Data: The OLTP systems contains the current data. this data does not be used for decision making.

The OLAP system contains the historical data and supports functions like summerization & aggregation and also data stores and manages at different levels.

3. Database design: In OLTP systems data modelling is done by using ER-model.

In OLAP systems data modelling is done by star (or) snowflake scheme.

4. View: Using OLTP systems we extract the data within a Enterprise & department.

In OLAP systems we acess the data from different organization's & different data sources.

5. Access Pattern: The OLTP systems contains the data. This data used for online Transac-tion's..

The OLAP systems contains the historical data. This data is used for data Analysis.

DWH → OLAP
OpDBS → OLTP.

Users and system orientation
Data.
Database design.
View
Access patter.

Comparision between OLTP and OLAP:

Feature	OLTP	OLAP
Characteristic	Transactional - processing	Informational - processing.
Users	Clerk, DBA, DB - professional	HR, manager, Analyst, executive.
Functions	Day - to - Day Transa -ctions.	Decision making.
Orientation	Online Transaction - orientation	Analysis Orientation
Data	current data	Historical data
Database Design	ER - model	star (or) Snowflake Schema
View	Relational (2-D)	Multi Dimensional.
Units of work	Simple Transactions	Complex Queries.
Acess pattern	Read / write	Read only
number of users	thousands	Hundreds
Number of data - records accessed	Tens	millions
Data storage	100MB to GB	100 GB - TB

fig (2.1.5). comparision between OLTP & OLAP

Data Warehouse Modeling : Data cube & OLAP

2.2. Multi-Dimensional Data model:

The Data Warehouse (or) OLAP tools contains the data in the form of multi-dimensional model. i.e; It contains the Data cube.

2.2.1. From Tables and Excel Sheets to Data cubes:

· Using Data cube we can View the data (or) analyze the data in the form of multi-dimension -al model. It is defined by Dimensions and Facts.

The Dimension is nothing but the Entity. Using this Organizations store the data. for Example, sales DWH Contains the dimensions like Time, Item, location, supplier etc. Using this dimensions we can analyze the things like monthly sales of a item, branches & locations at which the item were sold.

This Dimension information is stored in a Table. This is called as "Dimension Table". Each Dimension Table contains the Set of Attributes. for Example, item - Dimension contains the attributes like item-id, name, category, brand, type, etc.

The multiDimensional data model the Entire data typically organized under the central Theme like Sales. This "Central Theme" is called "Fact Table".

The facts are numerical Entities (or) numerical measures like dollars-sold (sales amount in dollars), units-sold (Total units sold).

For example, consider the table for sales DWH for All electronic's company is shown in below. It contains the dimension's like Time, items, and location.

Location = "Vancouver"				
Time (quarter)	item (types)			
	Home Entertainment	Computers	phones	Security
Q_1	600	700	500	400
Q_2	700	800	600	700
Q_3	500	600	400	700
Q_4	600	700	800	900

fig (2.2.1.1). Table for Sales DWH, it contains Dimensions, time, item & location

The table information is represented in 3D-cube the 3D-cube is shown in below.

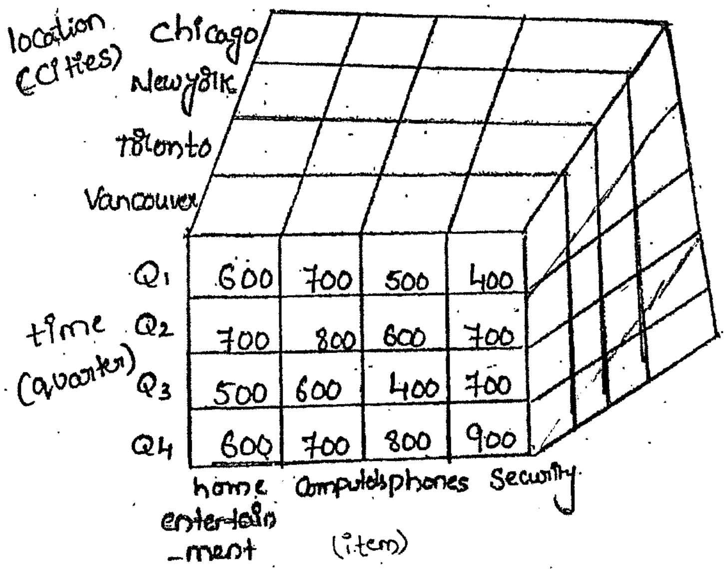


fig (2.2.1.2): A 3-D data cube for sales, it contains dimensions time, items & locations.

The Datacube contains the n-dimensions. The above Data cube may contains the set of dimensions. using This set of Dimensions we can construct the relation lattice of cuboids. Each cuboid gives the ~~different~~ level of summerization. Once the cuboids are completed we make the data cube. This is shown in the following diagram.

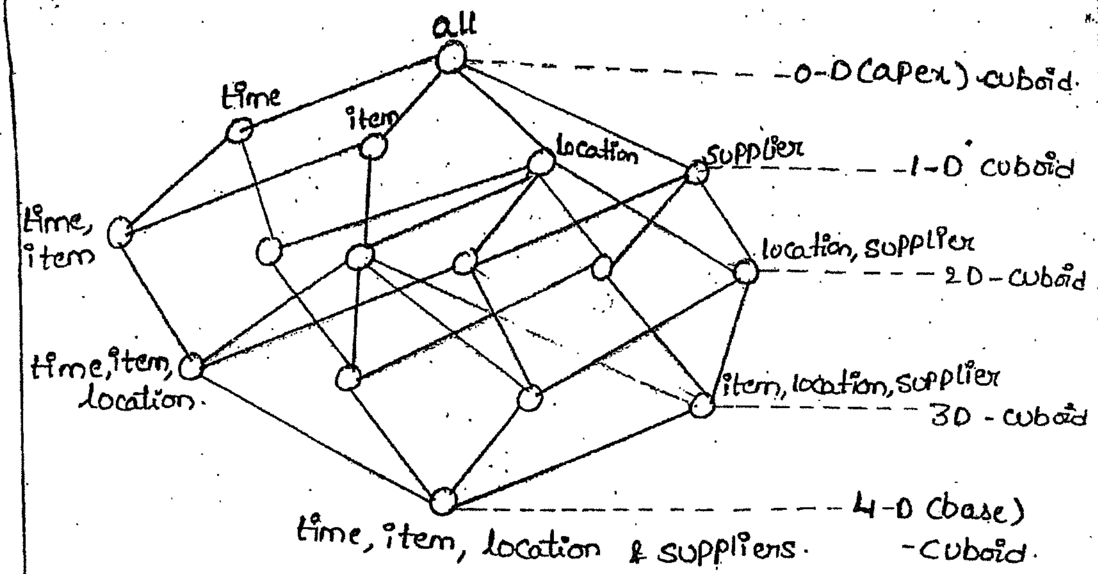


fig (2.2.1.3): A lattice of cuboids makes 4D Data cubes. It contains Dimensions time, item, location & suppliers.

In the above cuboid, 4D cuboid is called as The "Base cuboid". it gives the lowest level of summerization. The 0-D cuboid is called as "apex" cuboid. It gives the highest level of summerization. It is represented by the keyword "all".

2.2.2. Star, Snow-flake and fact constellations Schemas for multidimensional databases:-

In Relational database the data is loaded by using entity Relationship model. This is best suitable for the online Transactions. But the Data warehouse contains the subject areas like customers, products, sales, suppliers, etc. This is used for data analysis.

Therefore to load the data into DWH we use multi-dimensional Data model. Such model exists (or) form of star schema (or) snow-flake schema (or) fact constellation schema.

Star Schema:

That is the most popular data model to load the data into DWH. It contains 2 tables.

1. Fact Table (It contains large amount of data without any duplication).
2. Dimension Tables (one Table for each dimension

This schema look like the star Because of that it is named as "Star schema". This is shown in below.

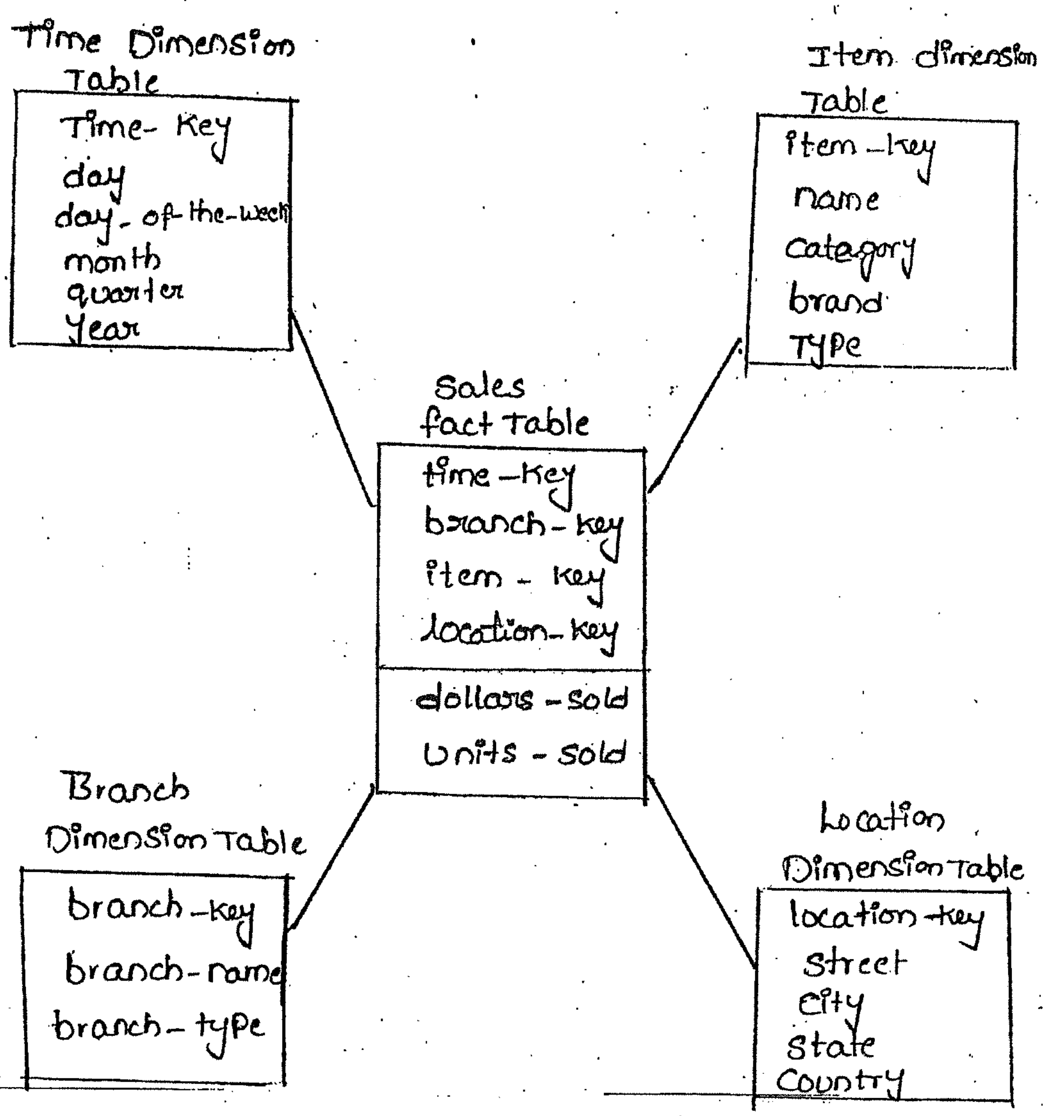


fig (2.2.2.1): Star Schema for All Electronics Sales DWH

Star Schema for fact Table contains 2 Parts.

(1) Key for Each dimension Table.

(2) Measures that is to be analyzed. i.e; like dollars-sold, units-sold.

Snowflake Schema:

In Star schema the dimensions tables are not normalized. But in snowflake schema the dimension tables are normalized. i.e; The tables are further splitted it into more tables. This is shown in below.

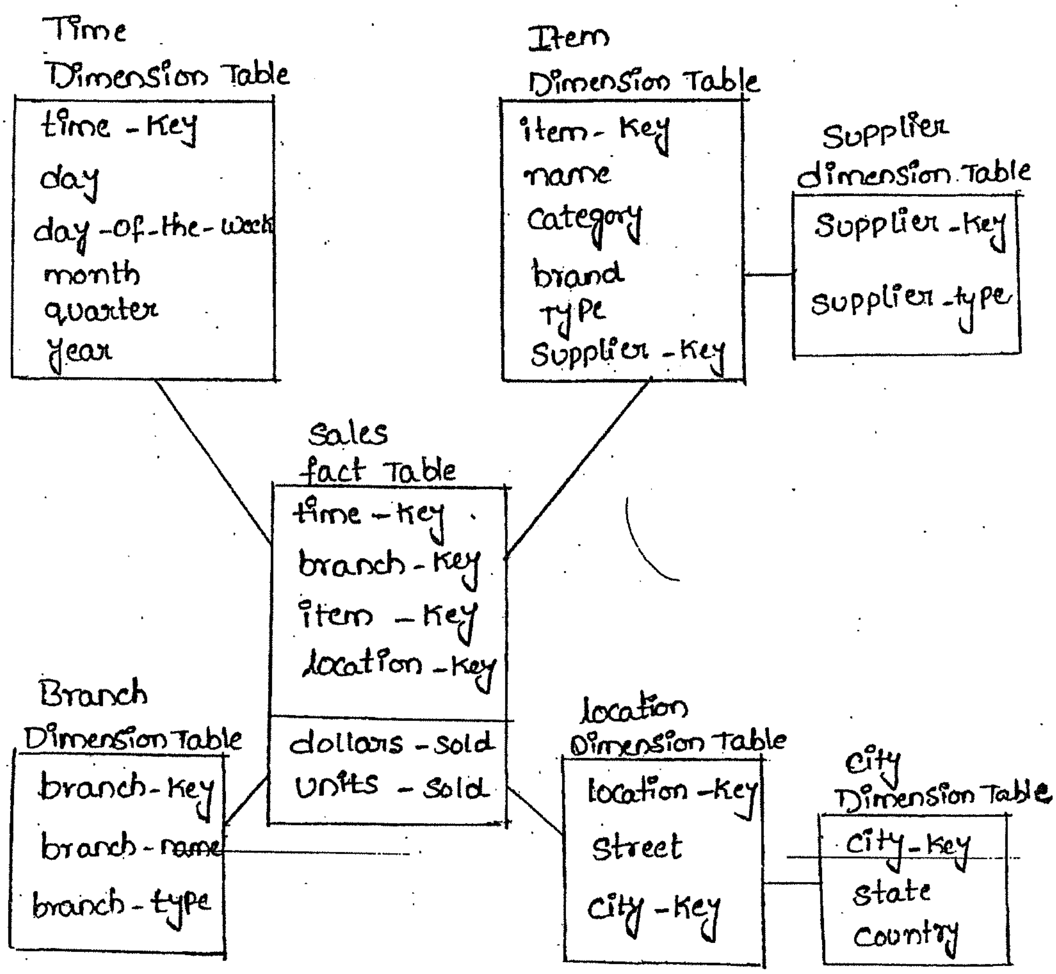


fig (2.2.2.2) Snowflake Schema for AllElectronics Sales DWH

Fact Constellation Schema:

In some of the applications it requires the multiple fact tables to represent the Dimension Tables. Such a Schema is called as the "fact Conste-llation schema".

the DWH contains the information about the Entire the organization .... Its scope is enterprise wide. But the Data mart is subset of DWH and it contains the single subject area ~~is~~ it's scope is department wide. If it is a DWH (or) Data Mart. It contains 2 Definition's.

(i) Cube Definition:

Syn: define cube < cube-name > [< dimension - list>]: <measures - list>

(2) Dimension Definition:

Syn: define dimension < dimension - name> as <attributes_list>

2.2.3. Measures:-

These are the numeric values that is to be analyzed. we find the measure Value at any time by aggregating the data with corresponding dimensions.

The measures are mainly classified into 3.

(1) Distributive:

If it is a distributive measure, then it Contains the function's like sum(), max(), min(), count(), etc....

(2) Algebric: If the measure is algebric means it contains the function's like avg(). The average is

$$avg() = \frac{sum()}{count()}$$

Here sum() & count() are distributive measures.

(3) Holistic:

If the measure is holistic means it contains the functions like mean(), median(), mode().

2.2.4. Introduction to Concept Hierarchies:-

The concept hierarchies defines sequence ~~to~~ of mappings from lowest level to highest level. The concept hierarchy for the location is shown in the following diagram.

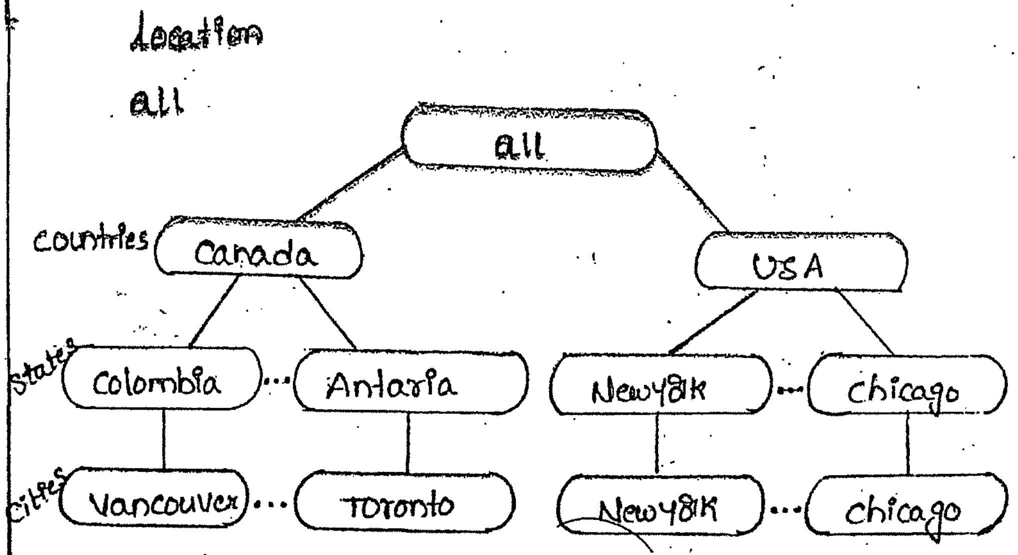


fig (2.2.4.1): Concept hierarchies for location Dimension

The concept of hierarchy for location dimension contains the attributes street, city, state and country. Using this attributes we can define the concept hierarchy.

" Street < city < State < country ".

These attributes are ~~organized in~~ partial order then we get the "lattice". The lattice for location Dimension.

fig (2-2-4-2). lattice for location Dimension

We can also define the lattice for time Dimension.

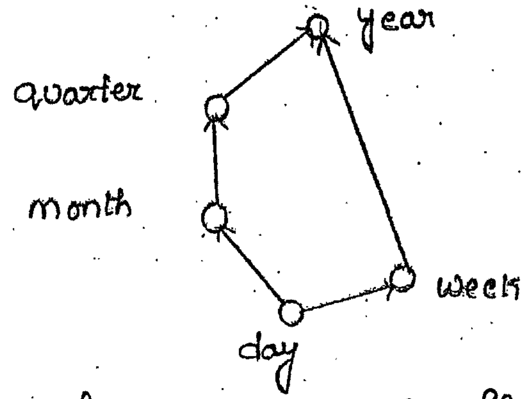


fig (2.2.4.3). Lattice for Time Dimension

The concept hierarchy is also defined for grouping of values then it is called as "Set - grouping hierarchy". The Set Grouping hierarchy for price Dimension is shown in below.

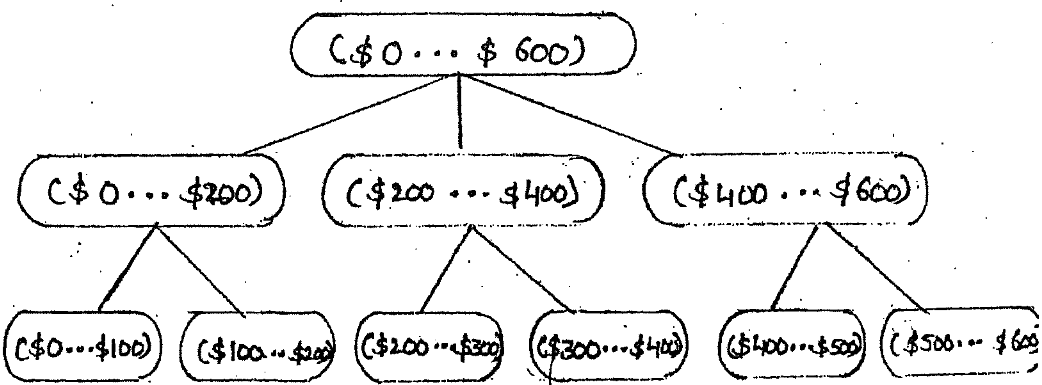


fig (2.2.4.4). Set Grouping hierarchy for price dimension

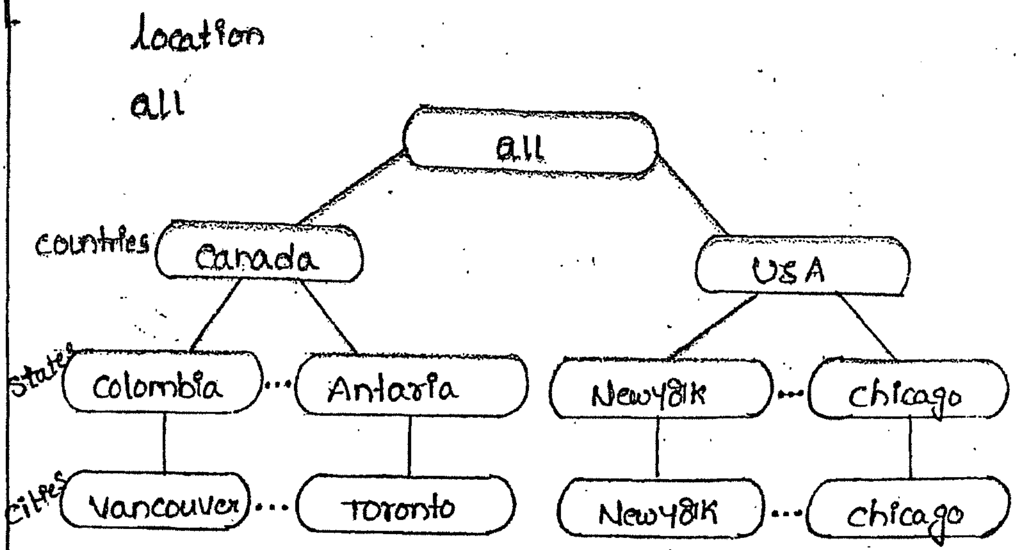


fig (2.2.4.1): Concept hierarchies for location Dimension

The concept of hierarchy for location dimension contains the attributes street, city, state and country. Using this attributes we can define the concept hierarchy.

" Street < city < State < country ".

These attributes are organized in ~~partial order~~ then we get the "lattice". The lattice for location Dimension.

fig (2.2.4.2). Lattice for location Dimension

We can also define the lattice for time Dimension:

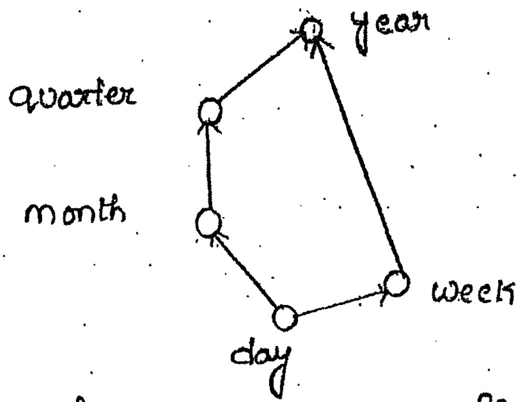


fig (2.2.4.3). Lattice for Time Dimension

The concept hierarchy is also defined for grouping of values then it is called as "Set - grouping hierarchy". The Set Grouping hierarchy for price Dimension is shown in below.

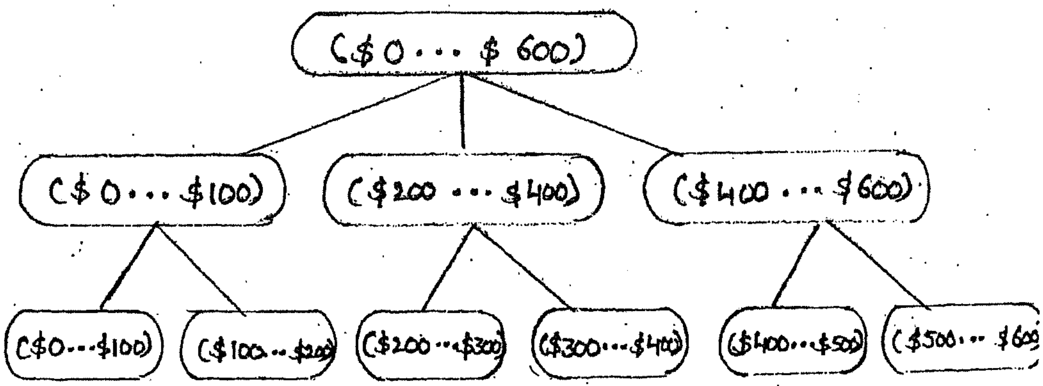


fig (2.2.4.4). Set Grouping hierarchy for price dimension

2.2.5 OLAP operations in multiDimensional Data Model:-

The multiDimensional Data model allows the data is organized in multiple Dimensions and also Each dimension contains the set of levels as defined in concept hierarchy. The OLAP Operations are shown in the following diagram.

combination of data-mart is going to form

DWH

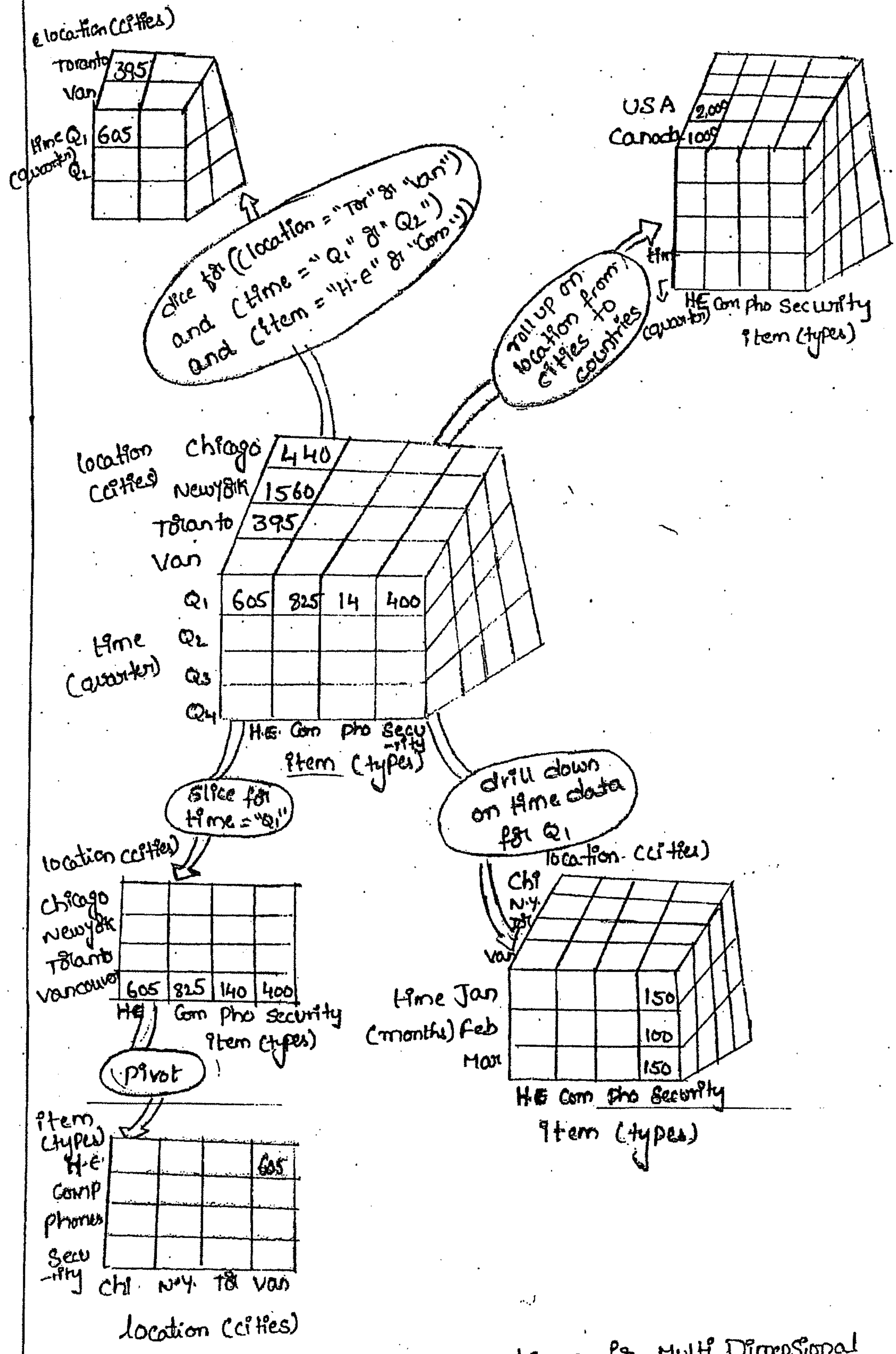


fig (2.2.5.1): OLAP operations for Multi Dimensional Data Model

(1) Roll-Up:

It is also called as Drill-up. It specified by claimbing up a concept hierarchy for a given dimension. Here roll-up operation is specified for location (from cities to countries).

(2) Drill-Down: It is the reverse of roll-up operation. Here Drill-Down operation is specified for time data on Q_1. Drill-Down means it is the stepping down the concept hierarchy for the given dimension.

(3) Slice and Dice: In slice operation we select only one Dimension to get the sub cube. Here slice operation is specified for time data = Q_1.

In Dice operation we select the 2 or more Dimensions to create the sub cube. Here Dice operation is specified for location = "Toronto" or "Vancouver", Time = "Q_1" or "Q_2", item = "home entertainment" or "computer".

Pivot: It is the Visualization operation. Here we change the data acass to present the another view of data. pivot is also called as "Rotate".

In Some of the OLAP applications additionally contains two operations.

(1) Drill Across: This uses the relational db requires to access the data from multiple fact tables.

2, Drill-Through: It also uses the relational db ~~requir~~ queries to drill lowest level of data cube. i.e; down to its back end relational table

2.2.6. Starnet Query Model for querying multi-dimensional Database's:

The starnet query model consists of radial lines originating from the centre and each radial line represent concept hierarchy for given dimension and also each level in concept hierarchy is called as "foot print".

The starnet query model for All Electronics database.

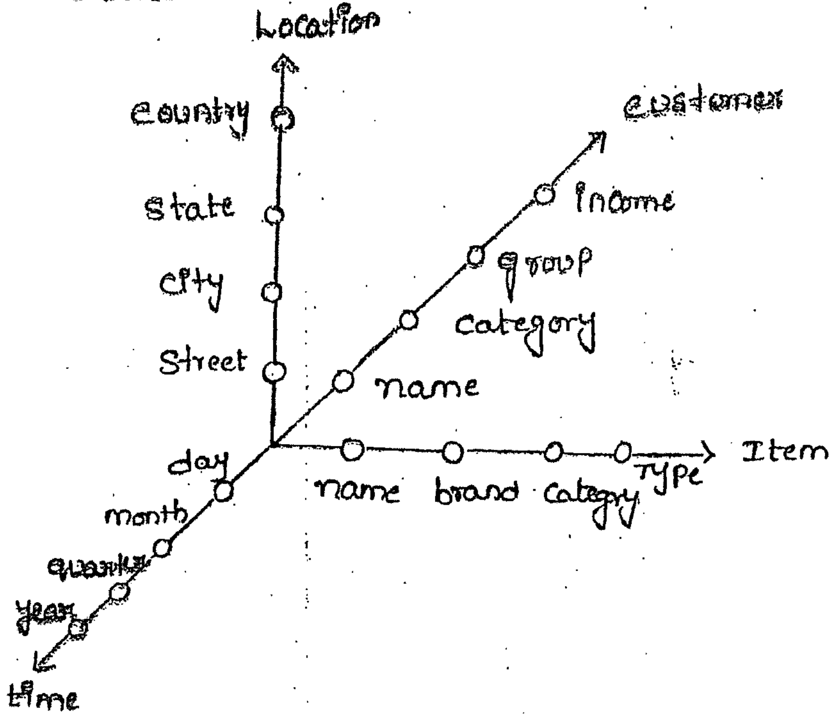


fig (2.2.6.1): Starnet query model for All Electronics Database

39. ## 2.3 Data Warehouse Architecture :-

2.3.1. The steps for designing and Construction of DWH :-

Reasons for DWH Design :-

1, The DWH provides the competative advantage. i.e, it contains the live Data.
2, using DWH we can extend our bussiness.
3, It provides the security.
4, It improves the Customer Relationship.
5, It Reduces the cost by using Relational DB querries to design the DWH generally we follow the 4 views.

a, Top-Down view.
b, Data source view.
c, Data ware house view.
d, Business process view.

a, **Top-Down view :-**

In this view we select the essential data before going to the DWH Design.

b, **Data source view :-**

In this view it exposes the data stored & Managed by the operational Data bases.

c, **DWH View :-**

In this view Data is used stored in the form of fact & Dimensional Tables.

V. Vijaya Lakshmi

d, **Business process view :-**

Here Data is extracted from the DWH and it presents to the end user accor--ding to the opinion of the End User.

The process of DWH Design :-

To Design the DWH generally we follow the 3 approaches :

1. Top-down Approach.
2. Bottom-up Approach.
3. Combined Approach.

1. **Top-down Approach :-**

In this approach we start from the strategic planning and Design. This approach is best suitable for the Technology is known & business problems are solved.

2. **Bottom-up Approach :-**

In this approach we always perform the Experimentation and here the system is completed quickly.

3. **Combined Approach :-**

In this approach it contains the 2 advantages that is strategic planning from top-down design & Rapid implementation from bottom-up approach.

Generally to design the DWH we follow the any one of the 2 Methods. i.e, Water fall Method (or) spiral Method.

Waterfall Model :-

In this small Model we construct systematic and structured analysis for each step and before going to the next step this is just like the waterfalling from one step to next step.

Spiral Model :-

In this Model updation is easy i.e., it can be adapted to any new Technology or any Model. Because of this it is best suitable for DWH & Data Marts.

The steps for Designing DWH process :-

The steps are

1. choose a bussiness process to Model. i.e., If we require entire data and its scope is enterprise wide then go for the DWH Model.

   The Data scope is limitted to with in the department then go for the Data ~~Most~~ Mart Model.

2. We have to define the level for each dimension.

3. We have to define the dimension tables that must be included in fact Table.

4. We have to define the Measures that is to be analyzed like dollars - sold, units - sold.

Three – Tier DWH Architecture :-

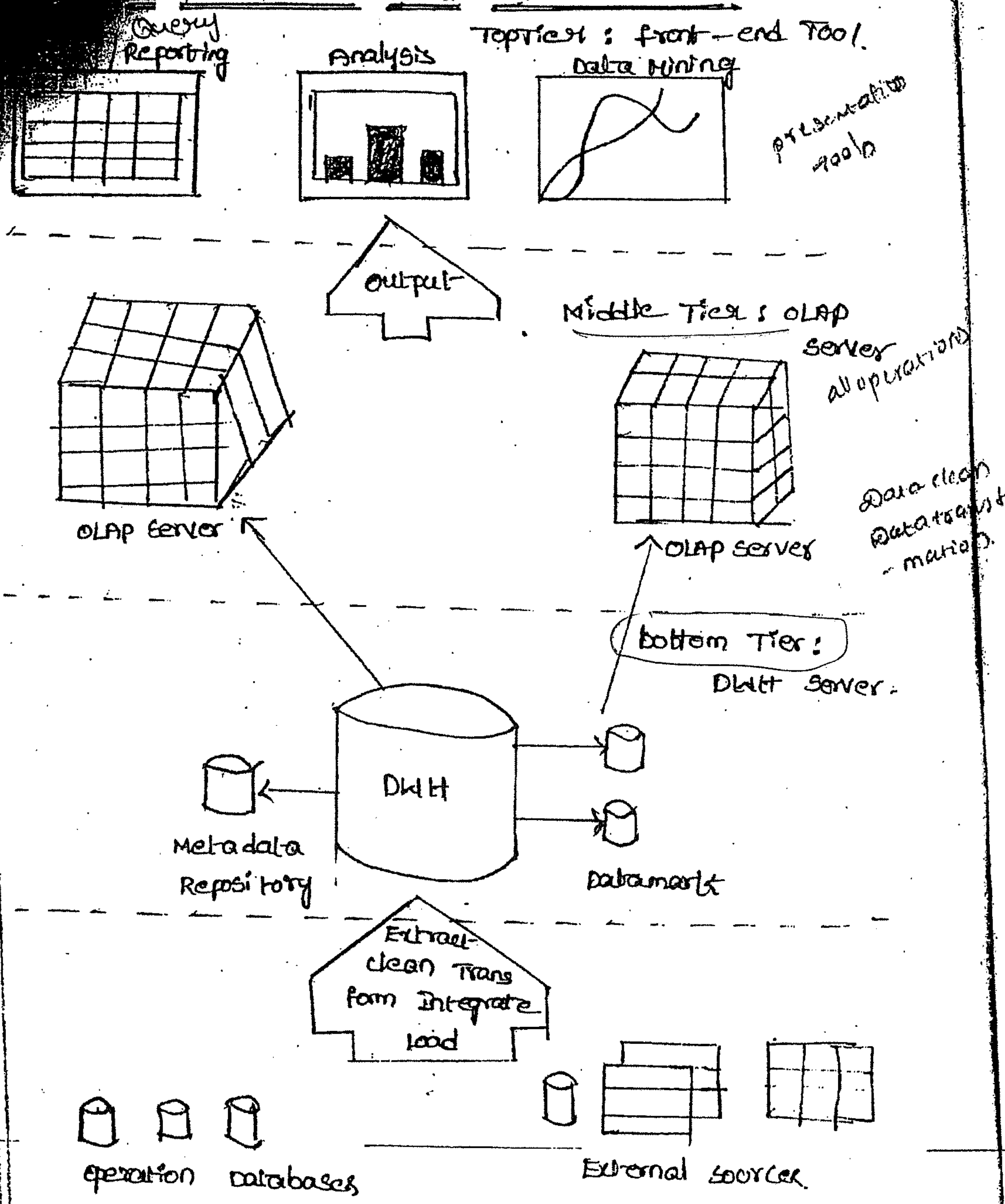


fig (2.3.2) A Three _ Tier DWH Architecture.

This architecture contains the 3 Tires.

1, The Bottom tier contains the DWH Server - Here data is extracted from the operation databases (or) external sources like Text-file, excel-sheet online-Data record etc. Then we apply data cleaning & data Transformation then this entire data is integrated and finally this is loaded it into DWH. To extract the data we use the application programs. These application programs are called as gateways.

2, The Middle Tier contains the OLAP servers. These servers are constructed directly from DWH data (or) Data Marts data. To construct these servers we use 2 Models.

i, ROLAP Model (Relational OLAP) → 3 schemas star, snowflakes, factconstellation

ii, MOLAP Model (Multi-Dimensional OLAP).

ROLAP Model, it is the Relational Db system. It maps the Multi-Dimensional data to Relational Db systems.

MOLAP Model, These are the special servers. These serves provide the multi-Dimensional view

3, The Top Tier contains the front end Tools like query and reporting tools, analysis Tools, Data Mining Tools. using these Tools the data is presented to Management.

Recommended Architecture for DWH :-

The Recommended architecture Contains the 3 steps.

Step 1 :

We have to define the high level enterprise Data Model. This model must be consist Realible and it must integrate the all the subject areas data.

Step 2 :

In step 2, We design independent data marts at the same time we have to develop the enterprise DWH.

Step 3 :

In step 3, first of all we develop the distributed data marts. These are used to integrate the different data marts data. Then finally we Construct the multi-Tier architecture for DWH.

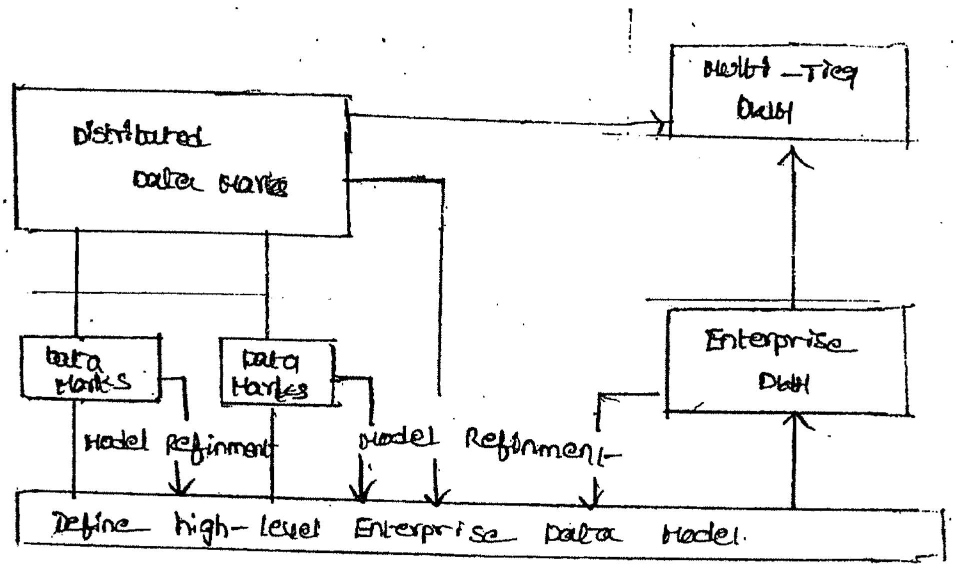


fig (2.3.2.2) A Recommended Architecture for DWH (two-Tier).

2.3.3 Types of OLAP Servers : ROLAP, MOLAP & HOLAP :-

The Types of OLAP servers Mainly classified into 3.

1. ROLAP (Relation OLAP).
2. MOLAP (Multidimension OLAP).
3. HOLAP (Hybrid OLAP).

These servers provides the Multidimensional View to the end user & the data is accessed from DWH & data marks.

1. **ROLAP :- (Relation OLAP)**

: This is the extensions of relational db Management system. It is also called as Relational db Management system. It contains the advantage of greater scalability and these ROLAP servers placed in b/w DWH & front End Tools.

2. **MOLAP (Multidimensional OLAP) :-**

These servers store the data in the form of Multidimensional arrays and provides the Multi-dimensional view to the end user. These ~~user~~ servers contains the advantage of efficient space utilization.

3. **HOLAP (Hybrid OLAP) :-**

It is the combination of ROLAP & MOLAP. It contains the advantage of scalability from ROLAP & efficient space utilization from MOLAP.

2.4 Implementation of DWH: using Data cubes and OLAP

The DWH contains the large amount of data. Therefore It is a critical for Data Mining system to provide efficient cube computation Techniques & query process Techniques.

2.4.1 Efficient computation of Data Cubes :-

In Multidimensional view cube computation extends SQL and includes "compute cube operator". For example sales data cube for all electronics contains 3 dimensions and one measure. i.e., item product year and measure is sales in dollars. This data is analyzed by using the following
"compute sum of sales, group by item, year"
Ily "compute sum of sales. group by year"
"compute sum of sales group by item".

The above cube contains the total no.of cuboids (or) group by's are $= 2^3 = 8$ i.e.,
{ (item, product, year), (item, year), (item, product)
(product, year), (item), (product), (year), () }

This above cube is represented in the form of Lattice of cuboids. This is shown in below.

() +1 is represented as all

$2^3 = 8$ (7+1) item

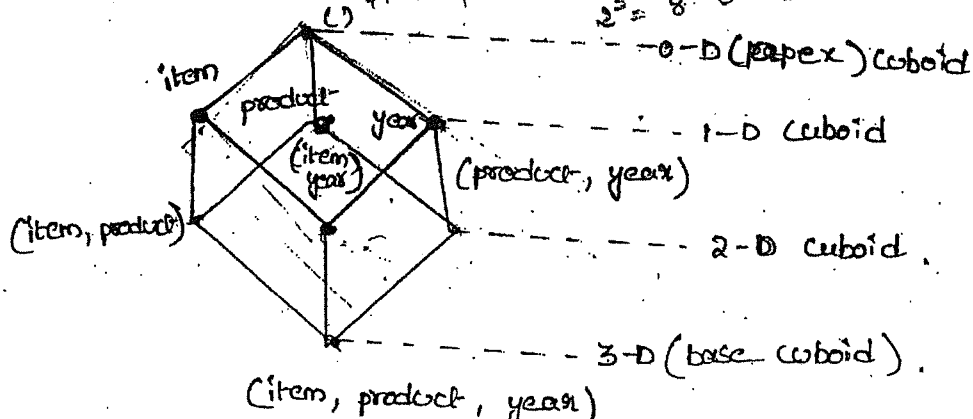


fig (2.4.1) : Lattice of Cuboids makes the 3-D (Database) Datacube, the Dimensions are item, product, & year.

Compute sales groupby item(or) product (or) year

Here, apex cuboid is represented by () or all It specify group by is Empty that is total sum of all the sales. This is computed by using the query.

"Computed sum of total sales". Similarly one dimension operation is specified by using query

"compute sum of sales group by item".

Ily 2-D operation is specified by using query

"Compute sum of sales group by item and year."

finally 3-D operation is specified by using query.

"Compute sum of sales group by item, product and year".

The above data cube is defined by using the DMQL.

DMQL → Data Mining Query Language

Define cube sales {item, product, yearly}: sum (sales_in_dollars). A cube with n dimension contains the Total Coboids = 2^n, if there is no hierarchy for each dimension. The above contain the 3 dimensions. Therefore Total coboids = $2^3 = 8$. But if the each dimension contains the level of hierarchy like time dimension i.e., day < month < quater < year; then total coboids $= \prod_{i=1}^{n} (L_i + 1)$. Where n is the no. of dimensions L_i means levels associated with the dimension i.

Here '1' specifies the high level summa-rization i.e., represented by R all, or, ( )

For example a cube contains 10 dimensions and each dimension contains the 4 levels then approximately. The total no. of coboids is 5^{10}. Therefore it contains the large no. of coboids and also each coboid requires the large amount of space. This is avoided by using concept hierarchy. - [illegible] level [illegible] level [illegible] compute [illegible]

Eg canada → countries → [illegible]

3. The aggregates are computed from the previously computed aggregate rather than taking from fact table.

MOLAP cube computation Techniques :-

In MOLAP the data is stored in the form of multidimensional array. Here each row is divided it into chunk. The chunk is a subcube and small enough to fit in the memory available for cube. The n-dimensional array is divided it into n-chunks. This is called as chunking. Each chunk is stored as a object on the disk. Each chunk is identified by "chunkID + offset".

For example 3-Dimension array contains 3 dimensions. i.e., A, B & C, then the chunks are shown in below.

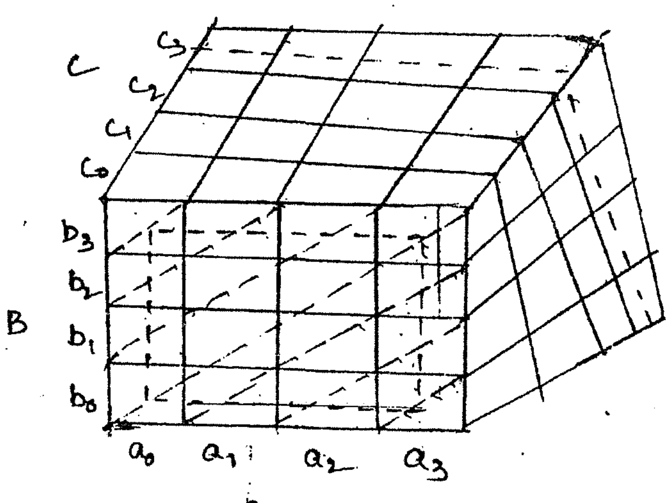


fig (2.4.2): 3-D array contains 3 dimensions A, B & C organized into 64 chunks.

partial materialization :-

The data cube materialization mainly classified into 3.

1, No materialization (eg - item)
2, full materialization
3, partial materialization (2 no 3 dimension & value 3 specific ...)

In those 3rd one is best option i.e, partial materialization. partial materialization means selected cuboids are materialized. i.e, selected cuboids are analyzed. i.e, The selected cuboids data is represented by using Data cubes.

The partial materialization contains the 3 steps.

1. Identify the ~~(subjects)~~ subsets of cuboids that is to be materialized.
2. Analyze cuboids during the query processing
3. update the cuboids during the data load process.

Multiway Array Aggregation in the computation of Data cubes :-

To Analyze the data ROLAP contains tuple & tables these two are the basic data structures in ROLAP.

ROLAP cube computation Techniques :

The ROLAP uses the

1, Sorting, Hashing & grouping operations on dimension attributes to recorder the data.
2, The ROLAP uses the grouping for sub queries to fastly the processing of sub querries.

The Datacube contains the following.

1, The highlevel summerization is represented by () or all.

2, The 2 dimension i.e., AB, AC & BC is calculated by grouping AB, AC, BC.

3, The one dimension i.e., A, B & C is calculated by grouping dimensions individually.

4, The 3-D (base) cuboid is represented by ABC.

Two arrangements for Multiway array aggregation for 3-D cube concept hierachy of sis values & el.

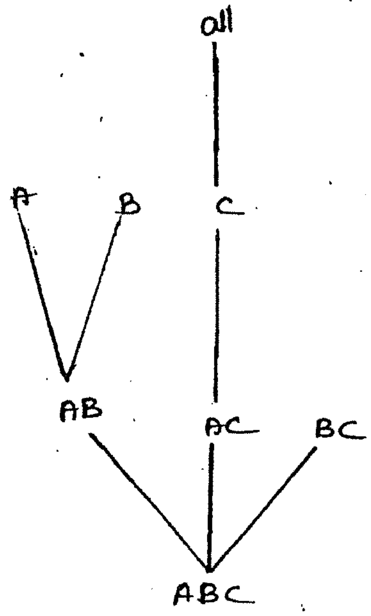
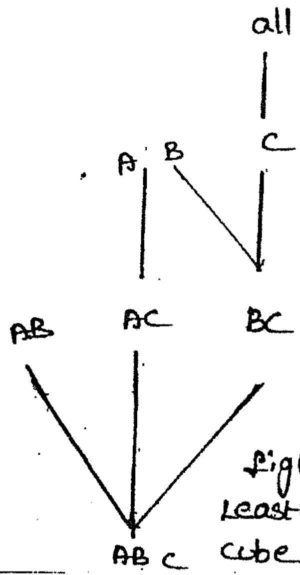


fig (a) Best used Cube Computing



fig(b). Leastly used cube computat

for ex, the Memory units for dimension A, B & C are 40, 400 & 4000 Memory units. then largest space required for 2-Dimension. i.e., BC (for units = 400×4000 = 1600000 MU) then 2nd largest space is required for 2-D.

i.e., AC (for units = 40×4000 = 160000 MU). The smallest space is required for 2-D.
i.e., AB (for units = 40×4000 = 16000 MU)

∴ In fig (a), Minimum space required.
= AB (for whole AB) + AC (for one row for AC)
+ BC (for one chunk for BC)

= 40×4000 + 10×4000 + 100×1000
= 16000 + 40000 + 100000 = 1,56,000 M.U.

In fig (b), Minimum space required.
= BC (for whole BC) + AC (for one row for AC)
+ AB (one chunk for AB).

= 400 × 4000 + 40 × 1000 + 10×100
= 1600000 + 40000 + 1000
= 16,41,000 MU.

2.4.2 : Indexing OLAP data :

To provide efficient data access most of the DWH systems contains index structure and materialization. The Materialization is provided through data cubes. The index structure is provided by using bit map indexing ~~and~~ join indexing.

Bit map indexing :-

This is the most popular data accessing Technique in OLAP. Using this we find the data quickly in data cube. This is the alternative Technique for Record-Id (RID) lists. In Bit-Map indexing. Each attribute value consists of distinct bit vector values. If the attribute value 'v' for the given row in base table then its value is set to '1' corresponding to the row in the bit-map index, all the remaining bits are set to '0' in the corresponding row at bit-map index.

for example consider the all electronics db.

Base Table

RID	item	city
R_1	H	V
R_2	C	V
R_3	P	V
R_4	S	T
R_5	H	T

Item Bit-map Index Table

RID	H	C	P	S
R_1	1	0	0	0
R_2	0	1	0	0
R_3	0	0	1	0
R_4	0	0	0	1
R_5	1	0	0	0

City Bit-map Index Table

RID	V	T
R_1	1	0
R_2	1	0
R_3	1	0
R_4	0	1
R_5	0	1

fig (2.4.2.1): OLAP index by using Bit-map Indexing

Advantages of Bitmap indexing :-

(1) It is the easy as well as the simple compared with the indexing & hashing.

(2) It requires the less amount of accessing time because the value is represented by using only 1 bit.

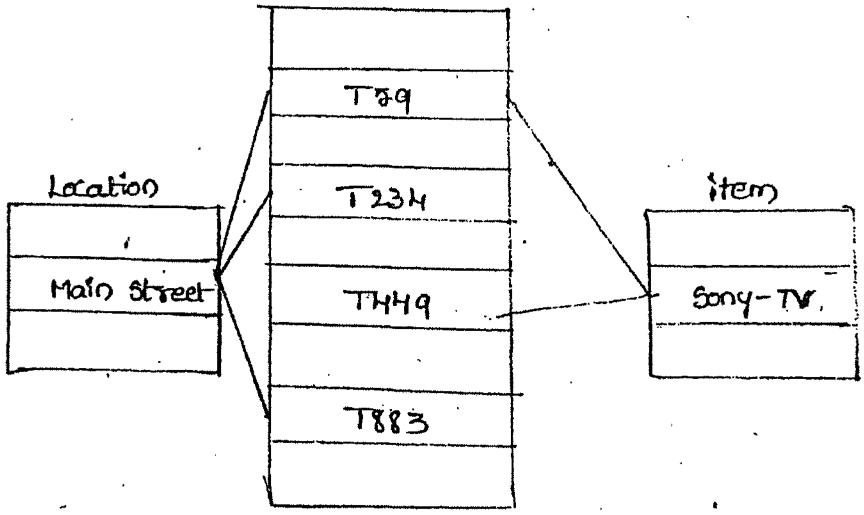
(3) It requires less amount of space.

Join indexing :-

It is mainly used in relational db by using this we easily find the joinable tuples.

for ex, Two relations are R(RID, A) and S(B, SID) join on attributes A and B. The join index record pair (RID, SID) where RID and SID are the record identifiers for relation R and S.

For example consider the sales fact table and 2D Tables i.e., location and item. The relation b/w the sales fact table and dimension tables are shown in below.



fig(4.2.2): Relation b/w sales fact table and 2D tables i.e., location and item.

Here, main street value in location dimension joins the tuples with T59, T234 & T883 of the sales fact tables similarly the value Sony TV of item dimension joins the Tuples T59 and T449 of the sales fact table.

The join index Tables are shown in below:

join index Table
Location/ Sales

Location	Sales -key
-----	-----
Main_street	T59
Main_street	T237
Main_street	T883
-----	-----

join index Table
item / Sales

item	Sales - item
---	---
Sony-TV	T59
Sony-TV	T449
---	---

join Index Table for Two Dimensions Location / item / Sales.

Location	item	Sales-key
Main-street	Sony TV	T59

fig (2.4.2.3) join Index Tables.

This join index Tables mainly used to maintain the relation b/w primary key and foreign key. This is mainly used in star schema to maintain the relation b/w foreign key of the fact table with primary key of dimension tables.

2.4.3 Metadata Repository :-

Metadata is nothing but data about the data. The Metadata repository contains the following steps.

1. The structure of DWH which includes schemas views, dimensions, hierarchies and data definitions.
2. The operational repository which includes data usage and monitoring information.
3. It also includes the algorithms for summarization.
4. To map data sources to data ware house which includes data cleaning, data transformation data integration and security Techniques.
5. The bussiness repository which includes bussiness policy, conditions, terms and business definitions

2.5 Datacube Technology :-

Here initially we write about the data cube in multidimensional data model

It is mainly classified into.

2.5.1 "Discovery Driven Exploration" of Data cubes:-

Here analyst or user search for the interest patterns by using OLAP operations i.e., drill-down, drill-up, slice and dice & pivot. OLAP opera-tion imp

If the discovery process is not automatted then end user search for the intersting patterns manually this is difficult. This is avoided by using the method discovery driven exploration. In this discovery driven exploration we identify the interes-ting pattern and it is marked. To provide the visualization to the end user.

Discovery Driven Exploration :-

In this method previously computed measures indicates the data exceptions after that these measures treated as exception indicators and using this endoser identifies the interesting pattern easily. ∴ These measures provide the "degree of Suprise" i.e., w.r.to existing value in the cell with expected value.

These measures are calculated by using statical analysis or grouping functions. These measures mainly classified into 3.

Chapter-3

Data preprocessing

Database consists of massive volume of data which is collected from heterogeneous sources due to this heterogenity, real world data tends to be inconsistent and noisy. If data is inconsistent, then there is a possibility that mining process can lead to confusion which results inaccurate.

Need for preprocessing the Data:

Incomplete, noisy and inconsistent data is common in large real world databases and datawarehouse

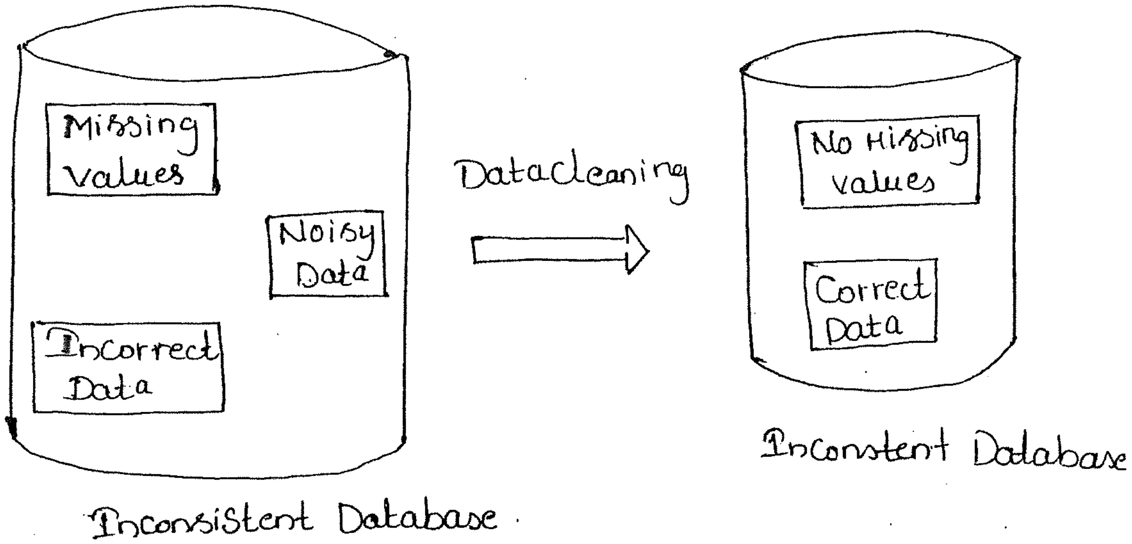
Incompleted data can occur for a no. of reasons.

1. Attributes of interest may not always be available.
2. Relevant data may not be recorded due to a misunderstanding.
3. Data that is inconsistent with other recorded data might be deleted.
4. The data collection instruments used may be faulty.
5. There may have been human or computer errors occuring at data entry.
6. Errors in data transmission can also occur.

- To overcome the above problems the following data preprocessing techniques are required.

1. Data cleaning
2. Data Integration
3. Data Transformation
4. Data Reduction.
5. Data Discretization.

1. <u>Data Cleaning</u>: when data is collected from datasources then there are chances that the data can be inconsistent, incompleted and noisy. Data cleaning is a Process of removing unnecessary and inconsistent data from the databases. The main purpose of data cleaning is to improve the quality of data by filling missing values, reconfiguring the data to make sure that data is in consistent format.



fig(1): DataCleaning

1.1: <u>Missing Values</u>:

The missing values consists the following techniques and corrected befor applying DMQL.

a) Ignore the tupple:

This technique is simple. In this just we avoid the tupple which doesn't contain values. But it's not recommended.

b.) Fill the missing values manually:

Here we manually fill the missing values. But this technique is not recommended for large database which contains important values.

c). Using Global Constant values:

Here the missing values has been filled with the global constant such as unknown (or) infinity. But this technique not succeeded because it deviates the Datamining Process.

d) Using attribute mean value:

Here each missing value has been filled by mean value i.e, In All Electronics customer average income is 2800 \$. Then the customer record i.e the income attribute's missing value is filled with 2800 \$.

e) Use the most portable values:

Here we find out the most portable values by using different techniques like Bayesian classification and decision Tree induction etc.

1.2 Noise data:

Here the noise data has been smoothing out by comparing neighbouring values. In this technique datavalues has been distributed into different wings.

1. Binning Method :

a) : <u>Smoothing by Bin-by-mean</u>

Here we find out the mean value for each Bin & missing values are replaced with that mean value.

b). <u>Smoothing by Bin Bounding</u> :

Here we find out min and max values for each Bin then each Bin's value is replaced with the closest value to the min or max.

Ex: price in dollars of an item is as follows.

4, 8, 15, 21, 21, 24, 25, 28, 34.

Bin 1 : 4, 8, 15

Bin2 : 21, 21, 24

Bin3 : 25, 28, 34

Smoothing by Bin

Bin 1 : 9 9 9

Bin 2 : 22 22 22

Bin 3 : 29 29 29.

smoothing by Bin Boundary value;

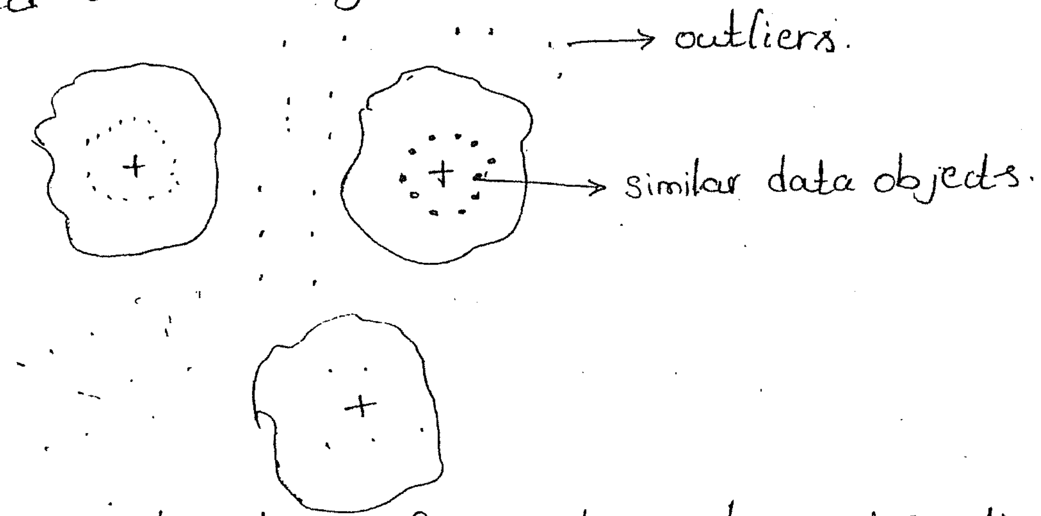
Bin 1 : 4 4 15

Bin 2 : 21 21 24

Bin 3 : 25 25 34

In Bin boundary we identify the data if it is near to the min value then we replace that value with min I or min values otherwise if it is near to the max value then we replace that value with the max value.

<u>Clustering</u>: In this we find out outliers i.e, clustering identifies Similar data objects those are placed in one cluster & also identifies dis-similar data objects and those are called outliers, in other words grouping the similar data objects into 1 place is called as clustering.



<u>Using combination of computers & human inspection</u>:

In this outliers can be identified through a combination of computer & human inspection i.e, we need to identify any algorithemic approa-ch to find out the outliers and clusters in our data set along with this we have to take the help of manual procedure to identify clusters & outliers.

<u>Inconsistent data</u>:

Inconsistent data means the data with

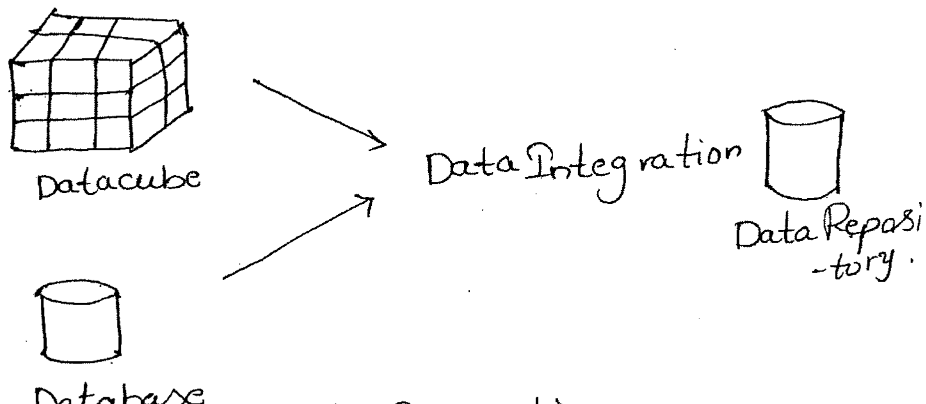
duplicated values. For Eg; ItemId is used to categor-ize items in AllElectronics. If the item-id entered wrong or missing then we can say that Item-id is inconsistent data. Then by applying different smoothing techniques Bin by mean, Bin by Boundary etc. to convert the data into consistent data.

<u>Data Integration & Data Transformation</u>:

Here the data from different data sources is collected and placed into the DWH, but it contains different difficulties to make unique identification or unique representation for the data of different datasources.

Data mining requires data integration. Data integ-ration is the merging of data from multiple data stores into a coherent data store as in DWH. These sources may include multiple databases, datacubes or flat files.

<u>Data Integration</u>: Data integration is a process of combining data from heterogeneous data sources such as different data bases, flatfiles etc, to form a single consistent data repository.



Data Integration.

For eg: In one data Source customer-id is entered as cust-id where as in another data source it entered as Customer Number.

The integration contains different problems while collecting the data from multiple tables. This can be avoided with an analysis called as correlation analysis. For Eg: The correlation analysis or relation b/w two attributes can be measured as,

$$= \frac{\Sigma(A-\bar{A})(B-\bar{B})}{(n-1)\,\overline{\sigma A}\,\overline{\sigma B}}$$

In the above Equation 'n' represents no.of tupples $\bar{A}$ & $\bar{B}$ represents mean values of A, B ΣA & ΣB represents standards deviations of A, B.

we can find the mean value of A as $\bar{A} = \frac{\Sigma A}{n}$

Similarly we can find out the standard deviation of A as

$$\overline{\sigma A} = \sqrt{\frac{(A-\bar{A})^2}{n-1}}$$

If the result of correlation analysis is greater than 0 (zero) then two attributes are positively correlated i.e, if we increase the value of attribute A then the value of attribute B will also increase.

If the correlation analysis result is '0' then two attributes are independent to each other.

If the correlation analysis result is less than 0 then the 2 attributes are negatively correlated ie If the value of A increases then the value of B will decrease.

Data Transformation:

Data Transformation is nothing but converting diff data sources into a format that must be acceptable for the datamining system.

5, 48, 99, 35, 81 $\xrightarrow{\text{Data Transformation}}$ 0.005, 0.048, 0.099, 0.035, 0.081.

1. Smoothing:

Here the noise data is subjected for Smoothing by using different smoothing techniques, Binning, clustering etc,.

2. Aggregation:

Here we use some aggregated functions to perform data transformation. For Eg: using daily sales we can compute monthly sales. In the same way, using monthly sales we can compute yearly or Annual sales.

3. ~~Generl~~ Generalization:

Here lower level value is replaced with higher level value by using Concept hierarchy. For eg: The dimension location contains concept hierarchy street, city, State & Country instead of placing the street value would be more generalized to place Country value; the data mining System.

4. Normalization:

In Normalization we force each value should in specific range. -1.0 to +1.0

5. Attribute Construction: In this new attribute

will be constructed by integrating attributes from different sources.

In the above 5 methods the Best one is "Normalization".

Normalization:

It contains following techniques.

1. Min-Max Normalization:

Here min_A max_A are the minimum & maximum values of the attribute A. Then the value ~~vof~~ A is transformed into v'. By using new range of new min_A & new max_A.

we compute new values i.e., v' By using below formulae.

$$v' = \frac{v - min_A}{max_A - min_A}(new\ max_A - new\ min_A) + new\ min_A$$

min max Normalization maintains same relationship of original datavalues. For Eg: The income attribute contains the min & max values of \$ 12000 & \$ 98000 then the income attribute changed into new range (0,1).

By using min, using min max Normalization the original value is \$ 73600 will be Transformed, into

$$v' = \frac{73600 - 1200}{98000 - 12000}(1-0) + 0.$$

$$v' = 0.716$$

2. Z-Score Normalization:

Z stands for zero mean. So this normalization is also called as zero mean normalization. In this attribute is find out by using mean & standard deviation value.

The value v of A will be transfomed by using the formulae.

$$v' = \frac{v - \bar{A}}{\sigma A}$$

Here $\bar{A}$ is called as mean value σA is called as standard deviation. For Eg: The attribute 'A' mean & standard deviation values are \$ 54000 \$ 16,000 then we can Z-Score the Normalization F_1 value v \$ 73600 will transformed into

$$v' = \frac{73600 - 54000}{16000}$$

$$= 1.225$$

Normalization by decimal scaling:

In this technique we move the decimal value based on absolute value of an attribute. By using this technique the original B of A is transformed into B' by using.

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

where j is the smallest integer based on the attribute value.

for Eg The attribute A Contains values between -986 to +917. Then we find out the absolute value as 986 by using the decimal scaling we can find out the new value v' by dividing it with 1000

i.e j=3

$$J = 3$$

$$v' = \frac{-986}{10^3}$$

$$= -0.986$$

Data Reduction:

Here we apply aggregation operations, Redundant attributes are removed from the data. Then we can apply data compression Technique on data.

In data reduction, we reduce the size of the data in such a way that essential features does not effect. It contains several techniques.

i) Data aggregation:

In Data aggregation we will apply some aggregation operation on the data to construct the data cube & we can also identify some clustered data to reduce the size of the data.

For eg. In AllElectronics database, the data is stored quarterly based for 3 years i.e, for the year 1997, 1998 & 1999 which is shown in the below diagram.

Year	1999

Year	1998

Year	1997
Q_1	\$ 234000
Q_2	\$ 368000
Q_3	\$ 246000
Q_4	\$ 157000

$\Longrightarrow$

Year -	Sales.
1997	x
1998	y
1999	z

we can apply different aggregation operations for the annual sales to represent the data in yearly format rather than storing it into Quarterly wise.

In the reduction, we can also represent the data in data cube format to represent multiple dimensions which is represented in the below diagram.

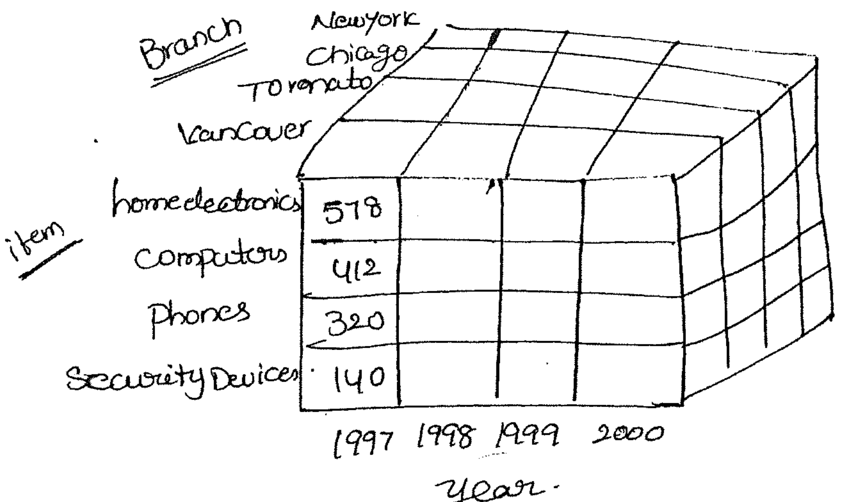


fig: Sales datacube for all electronics.

ii) Dimensions Reduction: Here the redundant attribu or dimension is identified & it is removed. It contai -ns several methods to reduce dimensions. In those methods, the main aim is we need to identify th min no. of attributes to represent the dataset. Among the available Techniques, the greedy method is important. In this method, it enable us to find out the best attribute or worst attribute based on decision tree analysis. It contains the following techniques.

1. Forward Selection: For Eg: Consider the following decision tree.

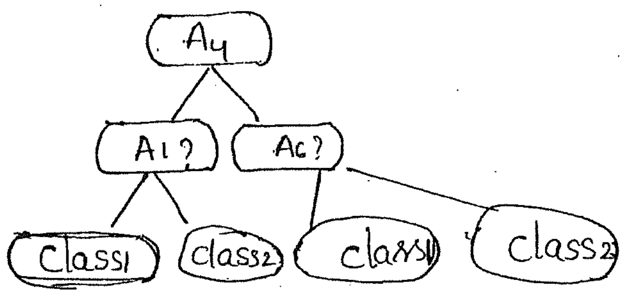


fig: Decision Tree data.

In forward selection we will start the identification with the empty set. Consider the data set as $\{A_1, A_2, A_3, A_4, A_5, A_6\}$ initial set is $\{\}$ (Empty set) In the order the first attribute to consider these $\{A_1\}$. The next dataset we consider is $\{A_1, A_4\}$ & the next data set we consider is $\{A_1, A_4, A_6\}$ we leave other attributes which are not important.

2. The Backward Eliminations:

Here we start our Searching from the complete

Dataset. In each step we eliminate worst attribute & that is removed from the dataset. This process will contained until all the dataset is complete. Here the dataset is $\{A_1, A_2, A_3, A_4, A_5, A_6\}$.

So we start over search from this only. In first step, we eliminate the attribute $\{A_2\}$ so the dataset becomes.

$$\{A_1, A_3, A_4, A_5, A_6\}.$$
$$\Downarrow$$
$$\{A_1, A_4, A_5, A_6\}$$
$$\Downarrow$$
$$\{A_1, A_4, A_6\}.$$

3. Combination of forward selection & Backward elimination: It is the integration of above 2 techniques. This is basically applicable for large Dataset. In this we need to identify the best attribute as well as worst attribute based on our requirement. we will add the best attribute to the dataset at the same time we remove the ~~A~~ worst attribute from the Dataset.

To perform all these redundant techniques we need to use decision tree induction. In decision tree induction, we perform test on attributes & we identify the results based on the classifications. Each classification is represent~~ate~~ed by "class labels"

In the last example, the total dataset i.e $[A_1, A_2, A_3, A_4, A_5, A_6]$ is reduced as $[A_1, A_4, A_6\}$ & final result will fall into either class 1 or class 2.

* Definition of Data Reduction :

Data reduction is a process of compressing massive volume of data into limited data set without sacrificing data integrity.

Year 1998	
Sem	Total
1	850
2	800

Year 1999	
Sem	Total
1	825
2	830

Data Reduction →

Year	Total
1998	1650
1999	1655
2000	1600

Year

Year 2000	
Sem	Total
1	800
2	800

iii Data Compression

It means we reduce the data in such a way that it must be in smaller size i.e., we reconstruct the data from the data in the form of compressed. If we compress the data without any loss of information that Data Compression technique is called as "Loss less Data Compression" otherwise it is called as "Loss Data Compression".

This Data compression contains two techniques

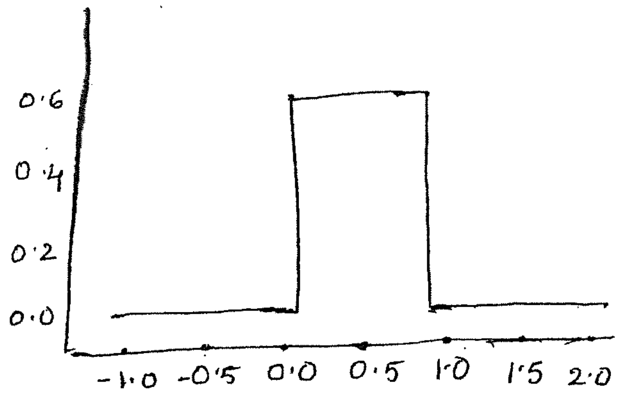
1. Discrete wavelet Transform (DWT)
2. Principle Components Analysis (PCA).

1. Discrete wavelet transform :

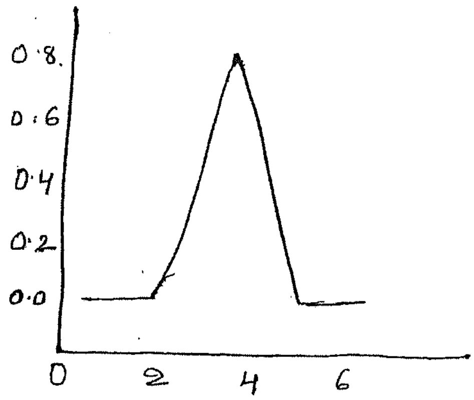
In this technique, the data Vector-D is transformed into D' of wavelet co-efficient. In this technique, we transform any value into new value by considering the coefficients with specific Range. This technique is derived from DFT (Discrete Fourier Transform). This DWT contains several wavlet Transform techniques. But familiar ones are.

(a) Haar-2
(b) Daubenchies 4

which are represented in the belao diagram.



(a) Haar-2



(b) Daubechies-4

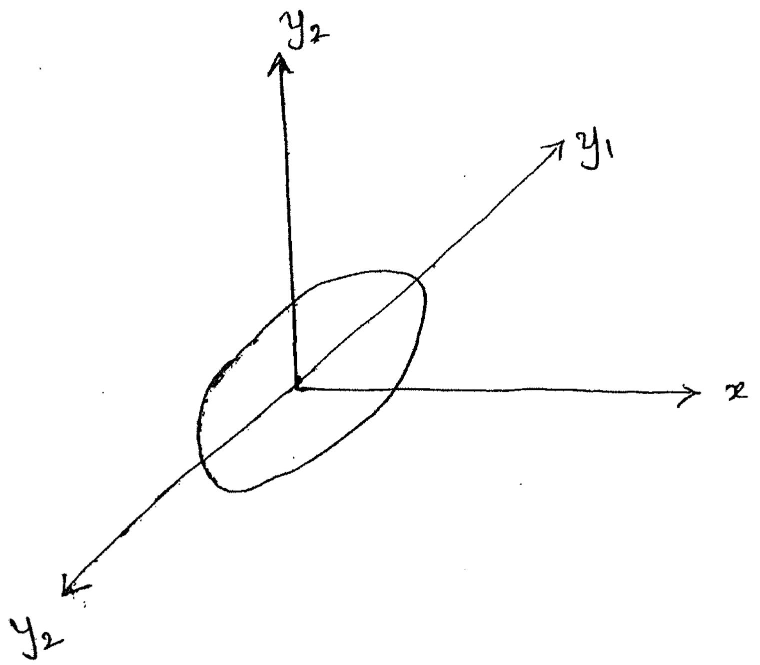
In Haar -2 technique we will consider the data based on the specific range co-efficient r. In the above diagram, the range for considerable values is from 0 to r. Then we consider the class labels in that specific rare by leaving other one.

Daubechies 4 also follows same technique but represent in the form of pyramid. The only difference is in Daubechies 4, the dataset length L is represented in the form of even values or in the form of L integer power of 2. Here we apply two functions to compress the data. In that initially we apply smoothing techniques such as sum & average and we findout the differences. If the data set is too large, then we can divide the dataset consider its length as L/2.

The procedure will repeat until the length of the dataset becomes 2. Then we will get compressed data which can be placed into the Datamining System.

2. Principle Components Analysis:

Here we compress the data of N-tuples or N Data vectors reduced into K-dimensional orthogonal vectors. By using this technique we can represent the data into compressed vectors. This technique is called as KL (Karhunen-Loébe) technique.



In principle component of analysis we apply the normalization of the dataset to represent the entire values in specific range we identify k-dimensi-onal orthogonal vectors & then we compute orthogonal vectors c with new range suchthat $c <= k$.

Here we construct new access to represents the compressed dataset. According to the above diagram x_1 & x_2 are original axis and y_1 & y are new axis for the original data which represents the compressed data from the original data.

In this method, final higher principle compon-ents are integrated to represent the original data.

iv **Numerosity Reduction :**

Here we reduce the data by selecting the smaller group of values. It is classified into.

i) parametric

ii) Nonparametric

parametric : In parametric Reduction techniques, the data is reduced by using

i, Linear regression

ii, Log-Linear regression models.

→ In Linear Regression model, we find the best line to be fit b/w the pair of attributes. If 1 value is given then we find the another value. Here we use the formula.

$$y = \alpha + \beta x$$

Where α, β are the regression coefficients.

x is the predictor value and

y is the Responsive value.

→ In Log Linear Model we predict the cell value of base cuboid. Using this we find the high-er level cuboids values. i.e By using low level cuboid values we can find high level cuboid values.

Non - parametric :

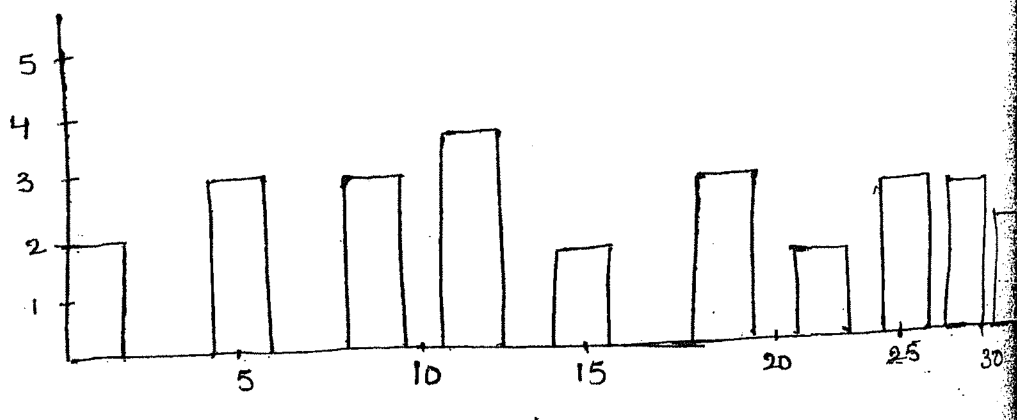
Here data is reduced by using the several methods. The methods are.

1. Histograms
2. Clustering
3. Sampling

1. Histograms :

Here we use the several smoothing techniques. The histogram of an attribute 'A'. is partitioned into disjoint parts. These are called as buckets. These buckets are represented in horizontal axis and the bucket count i.e, frequency is represented in vertical axis

For example, consider the all electronics sales data for price is specified in dollars i.e ; 1, 1, 5, 5, 8, 8, 8, 10, 10, 10, 10, 15, 15, 18, 18, 18, 21, 21, 25, 25, 25, 28, 28, 28, 30, 30. Then the histogram for the above data is show in below.

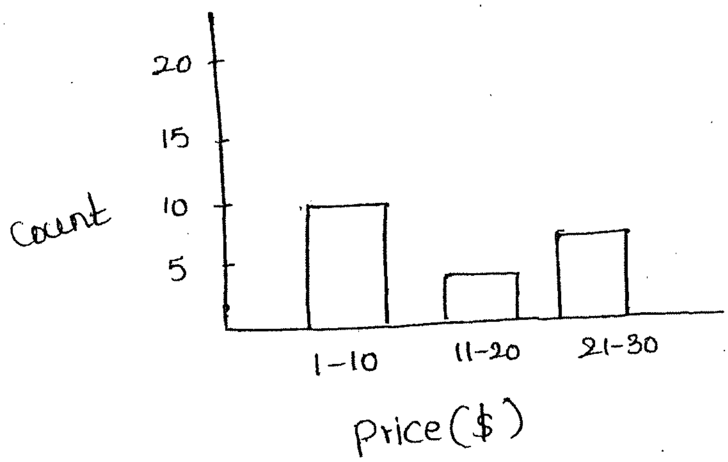


[illegible] Histogram for price according to singleton buckets

Here, each bucket contains only one value then those buckets are called as singleton buckets to partition the data we use several partitioning techniques. In those familiar one's are;

i. <u>Equi width</u>:

The equiwidth histogram contains equal width or constant width. The Equiwidth histogram for the above data.



ii <u>Equidepth</u>

The equidepth histogram contains the constant frequency that is, each bucket approximately contains the equal no. of samples.

iii) <u>V-optimal</u>:

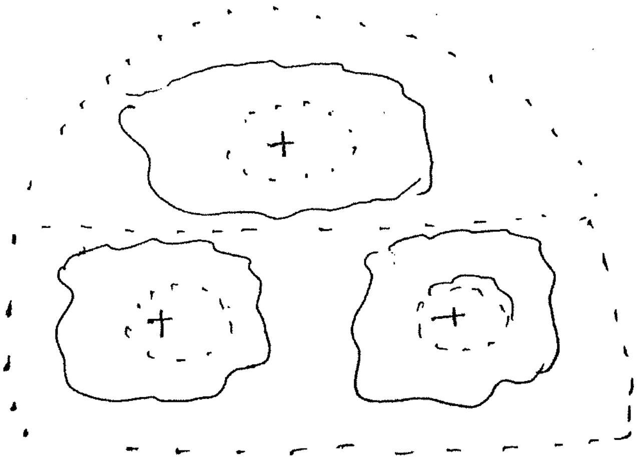
The V-optimal histogram contains the low variance. The histogram variance is nothing but sum of values of the bucket.

iv) Max-Diff :

The Max-Diff histogram contains the maximum difference of each pair of adjacent values. Here, we use the formula "$\beta-1$". Here 'β' is the maximum difference. this is specified by the end-user.

2. clustering :

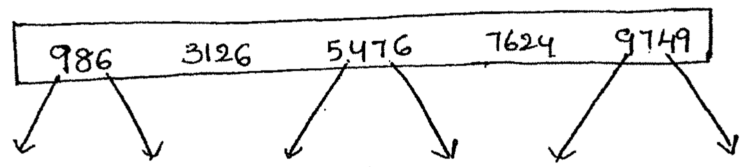
In clustering similar data objects are stored in one cluster and dissimilar data objects are stored in another cluster. for example, 2D customer data w.r.to customer locations in city is shown in below.



fig(a): 2D customer Data w.r.to custom location in city.

Here center of the cluster is marked with '+'. But in data reduction requires the original values to store the data in efficient way and access the data in efficient way. We use multidimensional index Tree structure. B+ tree organization.

The B^+ Tree organization is shown in below.



fig(b): B^+ Tree organization for data set.

For ex consider the large database. This datab-ase contains the 10,000 tuples. These are represented by using the keys 1 to 9999. We partitioned this data by using the equidepth Histogram. i.e. Each bucket approximately contains the equal no. of values. Then we get the 5 keys. These are ranging from 1 to 985, 986 to 3125, 3126 to 5475, 5476 to 7623, 7624 to 9748, 9749 to 9999.

Therefore, Each bucket approximately contains the 10,000 / 6 and also the keys are again divided it into Subkeys.

3. <u>Sampling</u>:

In sampling we reduce the data i.e Large amount of data is replaced with small random sample. For eg. the data set "D" and it contains the 'N' no. of tuples. Then the samplings are:

i) <u>Simple Random Sample without Replacement (SRSWR) for size n</u>:

Here We select the 'n' tuples where $n < N$. The Probability of drawing the tuple from D is $1/N$. Here we draw the tuple but, that tuple never be rewritten.

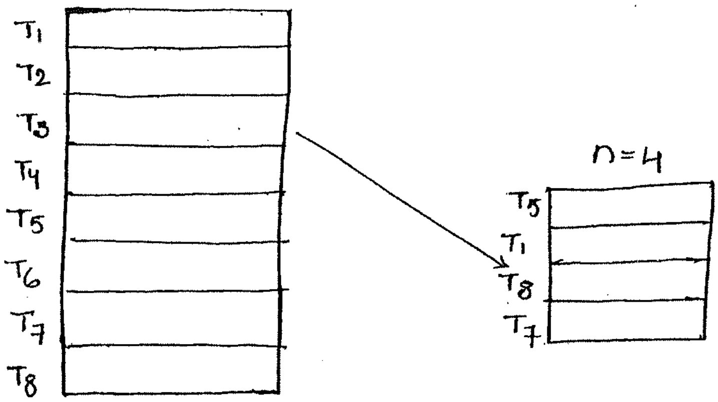


fig (a) : SRSWOR for n=4

ii) <u>Simple Random sample With Replacement</u> (SRSWR).
<u>for size n:</u>

Here, we access the tuple, recorded and rewritten. This technique allows the end user. The tuple is accessed again.

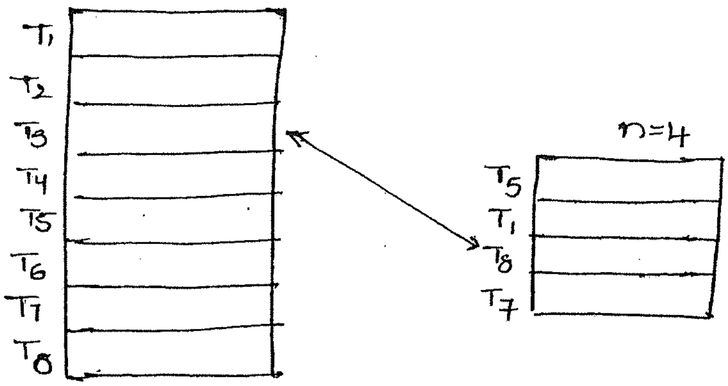
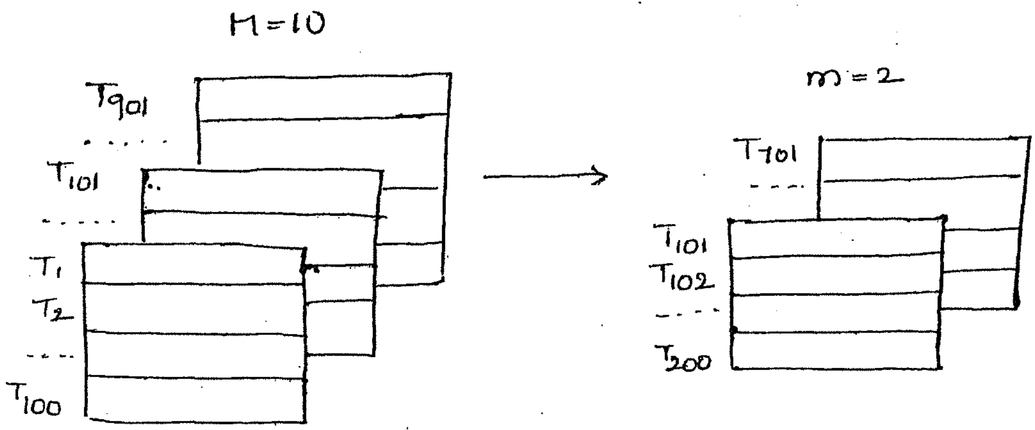


fig (b) : SRSWR for n=4.

iii) <u>cluster sample:</u>

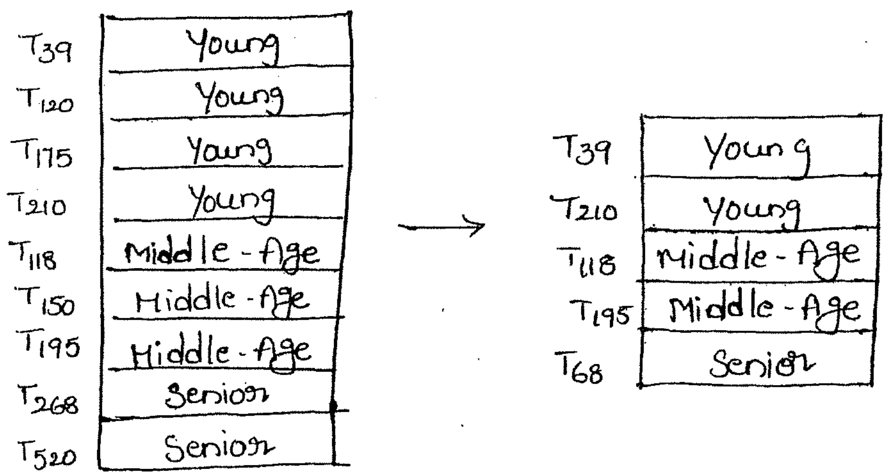
Here the tuples in 'D' grouped into 'M' mutually disjoint clusters. Then we select 'm' clusters. This is shown in below.



fig(c): cluster sample with m=2.

iv) <u>Stratified</u> <u>Sample</u>:

Here, large tuples of 'D' partitioned into mutually disjoint parts. These parts are called as strata. Then we generate the stratified samples of D for each stratum. This is shown in below.

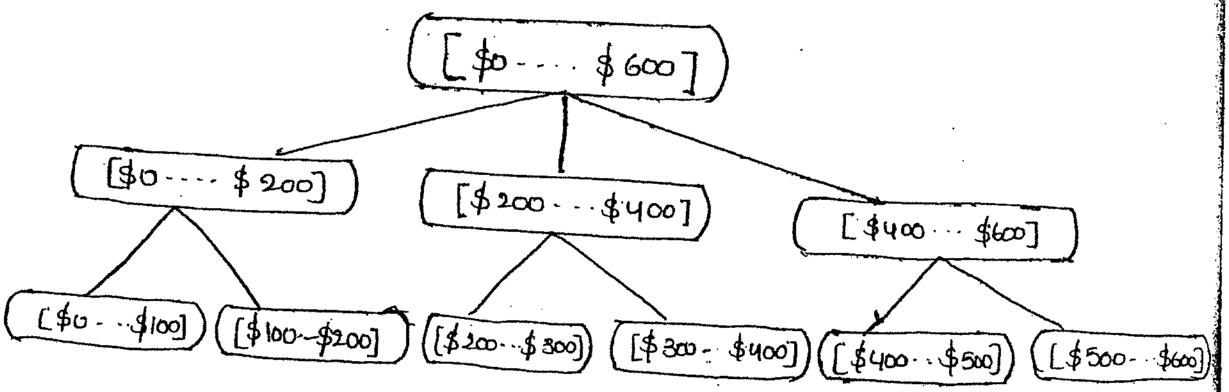


fig(d): stratified sample according to Age.

<u>Discretization & concept Hierarchy Generation</u>;

The Discretization means we reduce the data value of an continuous attribute by dividing it into intervals. In this best method is concept Hierarchy. In concept Hierarchy we collect and replaced with low level Hierarchies. For ex, Age attribute contains the values young, middle and seniors.

Then the Age attribute is replaced with one of these 3 values. Concept Hierarchy for price is shown in below.



fig(a): Concept Hierarchy for price.

Discretization and Concept Hierarchy Generation for Numeric data:

We apply the concept Hierarchy easily for Numeric data by using the method distributed attribute analysis. This method contains the 5 techniques.

1. Binning
2. Histogram Analysis
3. Cluster Analysis
4. Entropy - Based Discretization
5. Data Segmentation by natural partitioning

1. Binning:

Here data is partitioned and distributed it into different buckets or bins. Here we use the several Smoothing techniques.

- Smoothing by bin Mean:

Here, we find the mean Value for each bin and that is replaced with each value in bin.

ii. <u>Smoothing</u> <u>by</u> <u>bin</u> <u>Boundary</u>:

Here we find the maximum value, minimum value. Then each bin value is replaced with closest boundary For ex, Consider the price in dollars are 4, 8, 15, 21, 21, 24, 25, 28, 34 Partitioned with equidepth i.e, 3.

bin 1: 4 8 15

bin 2: 21 21 24

bin 3: 25 28 34

Smoothing by bin Mean:

bin 1: 9 9 9

bin 2: 22 22 22

bin 3: 29 29 29

Smoothing by bin boundary:

bin 1: 4 4 15

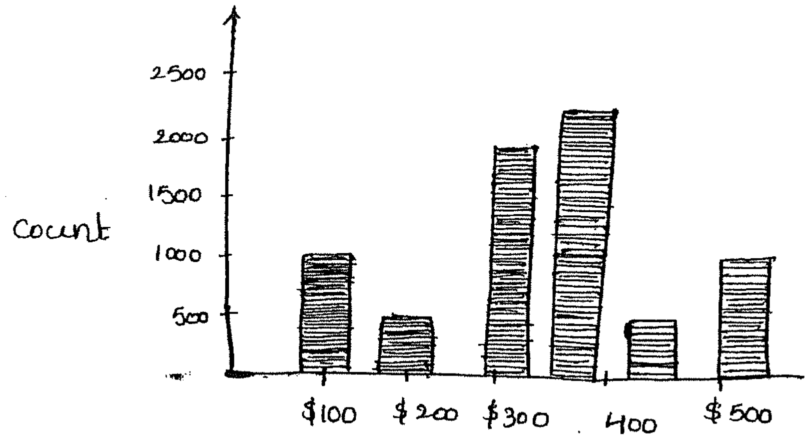
bin 2: 21 21 24

bin 3: 25 25 34

2. <u>Histogram</u> <u>Analysis</u>:

The Histogram for an attribute A is composed by partitioning data into disjoint bucket. These buckets are represented in horizontal axis and the bucket count i.e frequency is represented in vertical axis. For ex, Consider the most frequency range for price is \$ 300 to \$ 350 and this data is partitioned by using Equiwidth partition i.e. Each bucket contains the

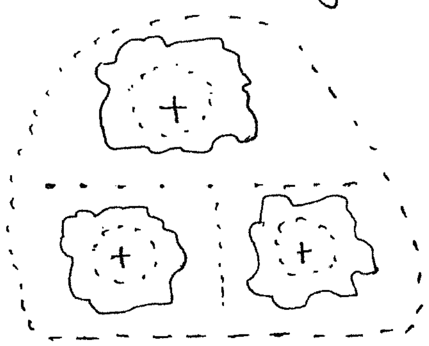
Equiwidth



fig(a): Histogram Analysis for price ($).

3. Cluster analysis:

In cluster Analysis, similar data objects are stored in one cluster and dissimilar data objects are stored in another cluster. The 2D customer data with respect to customer Locations in city is shown in below.



fig(a): 2D customer Data w.r to customer locations in city.

Here center of the cluster is marked with '+' and the cluster again divided it into smaller clusters.

4. Entropy Based Discretization:

The information based measure is called as Entropy. This is applied recursively for the numeric data of Attribute A.

Let us consider 'm' classes. The sample 'S' contains s_i samples in class C_i for $i = 1, 2, 3, \cdots m$. Then expected information for the given sample.

$$I(S_1, \cdots, S_m) = -\sum_{i=1}^{m} \frac{S_i}{S} \log_2 {}^{S_i/S} \rightarrow (1)$$

Then we can calculate Entropy for an attribute A as.

$$E_A = \sum_{j=1}^{v} \frac{(S_{1j} + \cdots + S_{mj})}{S} \times I(S_{1j}, \cdots, S_{mj}) \rightarrow (2)$$

Finally we can calculate information gain as follows

$$\text{gain}(A) = I(S_1, \cdots S_m) - E(A)$$

Based on the information gain, we can identify strongly relevant attributes and weakly relevant attributes.

5. <u>Data segmentation by Natural partitioning</u>:

Many of the users prefer the data is distributed uniformly. The uniform distribution allows to the end-user. the end-user read the data easily. For ex, Annual Salaries of a particular company ranging from [\$ 50,000 .... \$ 60,000] rather than specifying. [\$ 51,252.50 ..... \$ 61,252.612].

In this we use the method 3-4-5 i.e, the data is partitioned into Equiwidth of 3,4 or intervales. This method contains the following steps.

<u>Step-1</u>: If the most significant bit is 3,6,7 or 9 then the data is partitioned it into 3 equiwidth intervals.

<u>Step-2</u>: If the most significant bit is 2,4 or 8 then the data is partitioned it into 4-Equiwidth intervals

Step 3: If the most significant bit is 1, 5 or 10 then the data is partitioned it into 5 Equiwidth intervals.

For ex, consider the profit at all branches for all electronics company ranging from - \$ 351, 986 ... \$ 4,700, 876.50 and also 5% & 95% percentiles are - \$ 159, 896 and \$ 1,838,761. It contains the following steps.

Step-1: The given data MIN = -\$ 351, 986 , MAX = \$ 4,700, 876.50. Then the 5% is treated as low and 95% Percentile value is treated as high value.

Step-2: If we examine low and the high most significant bit is million dollar bit position i.e, Most significant bit = \$ 1,000,000 (msd) most significant Digit. Rounding low down to significant bit i.e' million dollar bit position

we get low' = -\$ 1,000,000

Similarly, Rounding high upto million dollar bit position we get high' = \$ 2,000,000

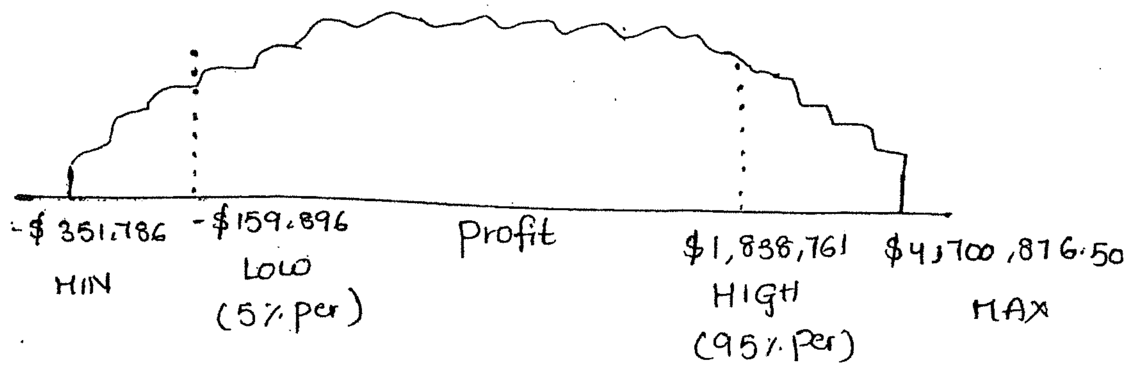
$$\Rightarrow (high' - low')/msd = \frac{\$2000000 - (-1000000)}{1000000}$$

Then we get the new range. this range is partitioned into 3 equiwidth.

Step-3: If we examine MIN and MAX then we get the MIN' = -\$ 4,000,000 and MAX' = \$ 5,000,000. Then we get the new range. This new range is partitioned into 4 Equiwidth intervals.

Step-4: These partitions again partitioned into sub partitions to define the partition hierarchy.

The entire process is shown below.



Step 2: msd = $ 1,000,000

Low' = $ 1,000,000 HIGH = $ 2,000,000.

Step 3: [-$ 1,000,000 . . . . . $ 2,000,000]

[-$ 1,000,000 ... $0] [$0 ... $1,000,000] [$1,000,000 ... $2,000,000]

Step-4: [-$4,000,000 . . . . $ 5,000,000]

[-$4,000,000 ... $0] [$0 ... $1,000,000] [$1,000,000 ... $2,000,000] [$2,000,000 ... $ 5,000,000]

Step-5:

- [-$4,00,000 ... -$3,00,000]
- [-$3,00,000 ... -$2,00,000]
- [-$2,00,000 ... -$1,00,000]
- [-$1,00,000 ... $0]

- [$0 ... $2,00,000]
- [$2,00,000 ... $4,00,000]
- [$4,00,000 ... $6,00,000]
- [$6,00,000 ... $8,00,000]
- [$8,00,000 ... $1,000,000]

- [$1,000,000 ... $1,200,000]
- [$1,200,000 ... $1,400,000]
- [$1,4000,000 ... $1,6000,000]
- [$1,6000,000 ... $1,8000,000]
- [$1,8000,000 ... $2,000,000]

- [$2,000,000 ... $3,000,000]
- [$3,000,000 ... $4,000,000]
- [$4,000,000 ... $5,000,000]

fig(a): 1.

fig(a): Concept Hierarchy for Profit based on 3-4-5 rule

Concept Hierarchy Generation for Categorical Data:

The Categorical data means discrete. It contains the finite no. of values for ex, job category, item types etc. It is mainly classified into 2.

1. specification of partial ordering of attributes at the schema level by user:

Here we specify the partial ordering of concept Hierarchy at the Schema level itself. For ex, Location Dimension Contains attributes Street, city, State, country. Then the Partial.

chapter-4

Getting to know your Data

4.1 Data objects and Attribute types

Data sets are made up of data objects. A data object represents an entity.

For ex:- In Sales DB, the objects may be customer store items, sales;

medical DB the objects may be Patients.

Data objects are typically described by attributes. Data objects can also be referred to as samples, examples, instances, data points or objects. If the data objects are stored in a data base they are data tuples. i.e, the rows of a DB correspond to the data objects, & columns corresponds to the attributes.

What is an Attribute?

An attribute is a data field, representing a characteristic or feature of a data object. The nouns attribute, dimension, feature & variable are often used interchangeably in the literature. The term dimension is commonly used in DWH. In Machine learning we use the term feature, while statisticians Refer the term variable.

Observed values for a given attribute are known as observations. A set of attributes used to describe a given object is called an attribute vector.

Types of attributes :-

The type of an attribute is determined by the set of possible values. they are

1) nominal
2) binary
3) ordinal
4) numeric.

1) Nominal Attributes :-

Nominal means "relating to names". the values of a nominal attribute are symbols or names of things. Each value represents some kind of category, code (or) state. Nominal attributes are also reffered to as categorical. the values do not have any meaningful order. In computer science the values also known as enumerations.

Ex1 — suppose that hair_color & marital_status are two attributes describing person objects.

possible values for hair_color are black, brown, red, gray white.

the attribute marital - status can take on the values single, married, divorced & widowed.

2, Binary Attributes :-

Abinary att: is a nominal att: with only two categories or states: 0 (or) 1, where 0 is represents absent of an attribute, & 1 represents present of an att: Binary att: are referred to as Boolean if the two states correspond to true & false.

for ex :- Given the attribute Smoker describing a patient object, 1 indicates that the patient smokes, while 0 indicates that the patient does not.

A binary attribute is symmetric if both of its states are equally valuable and carry & carry the same weight; i.e, there is no preference on which outcome should be coded as ~~or~~ 0 or 1.

A binary attribute is Asymmetric, if the outcomes of the states are not equally important. such as the positive & negative outcomes of a medical test for HIV. By convention, we code the most important outcome, which is usually the rarest one, by 1 & the other by 0.

3, ordinal Attributes :-

An ordinal att is an att with possible values that have a meaningful order or ranking among them, but the magnitude b/w successive values is not known.

Ex:- ordinal att's are useful for registering subjective assessments of qualities that cannot be measured objectively. ordinal attributes are often used in surveys for rating.

Ex:- In one survey, participants were asked to rate how satisfied they were as customer. customer satisfication had the following ordinal categories.

0: very dissatisfied 1: Some what dissatisfied
2: neutral 3: satisfied & 4: very satisfied.

4, Numeric Attributes :-

A numeric attribute is quantitative; ie, it is a measurable quantity, represented in integer or real values. Numeric att's can be interval-scaled or ratio-scaled.

Interval-scaled Attributes

Interval-scaled att's are measured on a scale of equal-size units. The values of interval-scaled attributes have order and can be positive, 0, or -ve. In addition to providing a ranking of values, such att:s allow us to compare & quantify the diff. b/w values.

Ex:

1) Celsius Temperature
2) IQ (Intelligence scale
3) Time on a clock with hands.

ively

Ratio - Scaled Attributes:-

A ratio - scaled Att: is a numeric att: with an inherent zero point. i.e, if a measurement is ratio scaled, we can speak of a value as being a multiple of another value. In addition, we can also compute the diff: b/w values, as well as mean, median & mode.

Ex:-

1) Age, weight, Height.
2) Ruler Measurements
3) Years of education.

5). Discrete versus Continous Attributes :-

In we have organized att:s into nominal, binary, ordinal, & numeric types. there are many ways to organize att: types. the types are not mutually exclusive.

A discrete attribute :-

It has a finite or countably infinite set of values, which may or may not be represented as integers. the attri hair-color, smoker, medical-test each have a finite number of values, such as 0 or 1 for binary atts and are discrete.

Continuous Attribute :-

If an attri is not discrete, then it is continous. The terms numeric attribute & continous att: are often used interchangeably in literature. Real values are represented using a finite number of digits. Continous attributes are typically represented as floating point variables.

2.2. Basic Statistical Description :-

Basic statistical descriptions can be used to identify properties of the data and highlight which data values should be treated as noise or outliers. In this we discusses three areas of basic statistical descriptions. measures of Central Tendency, measures of dispersion & graphic Displays.

2.2.1 Measuring the central Tendency : mean, median and Mode:

A measure of central tendency is a single value that describes the way in which a group of data cluster arround a central value. In other words, it is a way to describe the centre of a data set. They there are three measures of central tendency:

1) Mean
2) Median
3) Mode.

Mean :-

The mean is preferred measure of central tendency because it consider all the values of dataset. In order to calculate the mean, data must be numerical. You cannot use the mean for nominal data, which is data on characteristics like gender.

The mean for set of values is

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

Ex:- Suppose we have the following values for salary, shown in increasing order:

30, 36, 47, 50, 52, 52, 56, 60, 63, 70, 70, 110.

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

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

$$= 58$$

The mean salary is \$58,000.

Sometimes, each value x_i in a set may be associated with a weight w_i for $i = 1, 2, \ldots N$. The weights reflect the significance.

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

This is called as weighted average or weighted Arithmetic mean (AM).

Median :-

The median is the middle value for the given data, that has been arranged in order of magnitude.

If we have 2 middle value, we can find the the median mean value b/w the 2 values, & take it as a median.

The median is expensive to compute when we have a large number of observations.

$$\text{Median} = l + \left(\frac{\frac{n}{2} - cf}{f} \right) \times h.$$

l = lowest values (lowest interval)

$\frac{n}{2}$ = $\frac{\text{no. of observations}}{2}$

c_f = highest frequency
c_f = above the f
h = class interval

Model

A statistical term that refers to the most frequently occurring number found in a set of numbers. The mode is found by collecting and organizing the data in order to count the frequency of each result.

Ex: Suppose we have the following values
30, 36, 47, 50, 52, 52, 56, 60, 63, 70, 70, ~~70~~ 110

mode = 70, 52. This is bimodal.

For unimodal numeric data that are moderately skewed, we have the following expression.

$$\text{mean} - \text{mode} \approx 3 \times (\text{mean} - \text{median})$$

i.e, the mode for unimodal frequency curves that are moderately skewed can easily be approximated if the mean & median values are known.

Mid range :-
~~Range~~:-

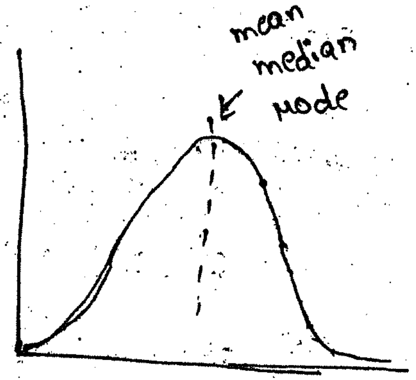
(The range of a set of data is the diff^s b/w largest & smallest values)

For the above example the midrage is as follows

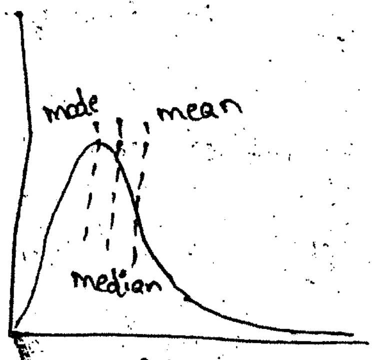
$$\frac{30,000 + 110,000}{2} = 70,000.$$

In a unimodal frequency curve with perfect symmetric data distribution the mean, median & mode are all at the same center value.

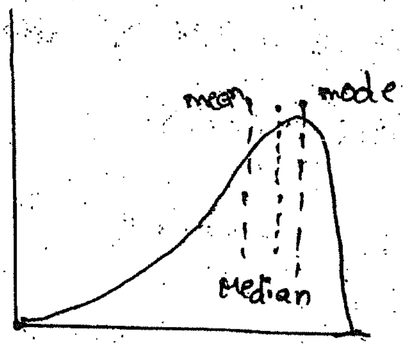
Data is most real application are not symmetric. they may instead be either "positively skewed", where the mode occurs at a value that is, smaller than median, or negatively skewed, where the mode occurs at a value greater than the median.



(a) Symmetric data



(b) positively skewed data.



(c) Negatively skewed data.

fig(4.1) Mean, median, mode of symmetric v/s positively & negatively skewed data.

4.2 Measuring the Dispersion of Data :-

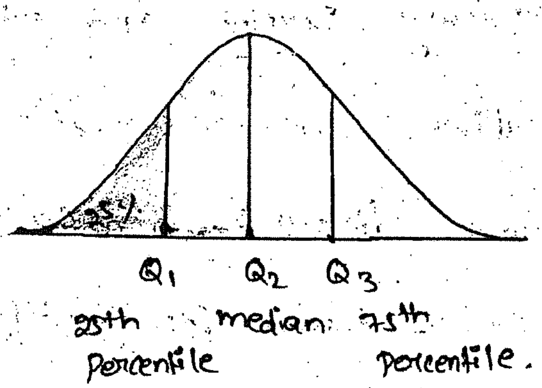
A measure of spread, or measure of dispersion is used to describe the variability in a Sma Sample or population. The measures include range, quantiles, quartiles, percentiles and the Interquartile range. The five number summary, which can be displayed as a boxplot, is usefull to identifying outliers.

Range, Quartiles, and Interquarties Range

let $x_1, x_2 \ldots x_N$ be a set of observations for some numeric attribute, X. The range of the set is the difference b/w the large & small value range = max(x) − min(x).

Quartiles tell us about the spread of the data set by breaking the data set into quarters, just like median breaks it in half.

The first quartile(Q_1) lies b/w the 25^{th} & 26^{th} observation, the second quartile (Q_2) b/w 50^{th} & 51^{th} observation & the third quartile (Q_3) b/w the 75^{th} & 76^{th} observation.



The distance b/w the first & third quartiles is simple measure of spread that gives the range covered by the middle half of the data. this distance is called "Interquartile range"(IQR)

$$IQR = Q_3 - Q_1$$

Ex: suppose we have have the following values for salary is

30, 36, 47, 50, 52, 52, 56, 60, 63, 70, 70, 110.

It contains 12 observations, already stored in increasing order. thus, the quartails are for the data are 3rd, 6th & 9th values respectively.

$\therefore$ $Q_1 = 47,000$

$Q_2 = 52,000$

$Q_3 = 63000$

$IQR = 63 - 47 = 16,000.$

five-number Summary, Boxplots and outliers:-

The outlier is an observation that lies an abnormal distance from other values. A common rule of thumb for identifying suspected outliers is to single value falling at least $1.5 \times IQR$ above the 3rd Quartile or below the 1st quartile.

The five-number summary of a distribution consists of the median (Q_2), the quartiles Q_1 & Q_3, and the smallest & largest individual observations written in the order of min, Q_1, median, Q_3, max.

Boxplots are a popular way of visualizing a distribution. A boxplot incorporates the five number summary as follows:

* The ends of the box are the the quartiles, so the box lenght is interquartile range.
* The median is marked by a line within the box.
* Two lines (called whiskers) outside the box extend to the Smallest & largest Observations.

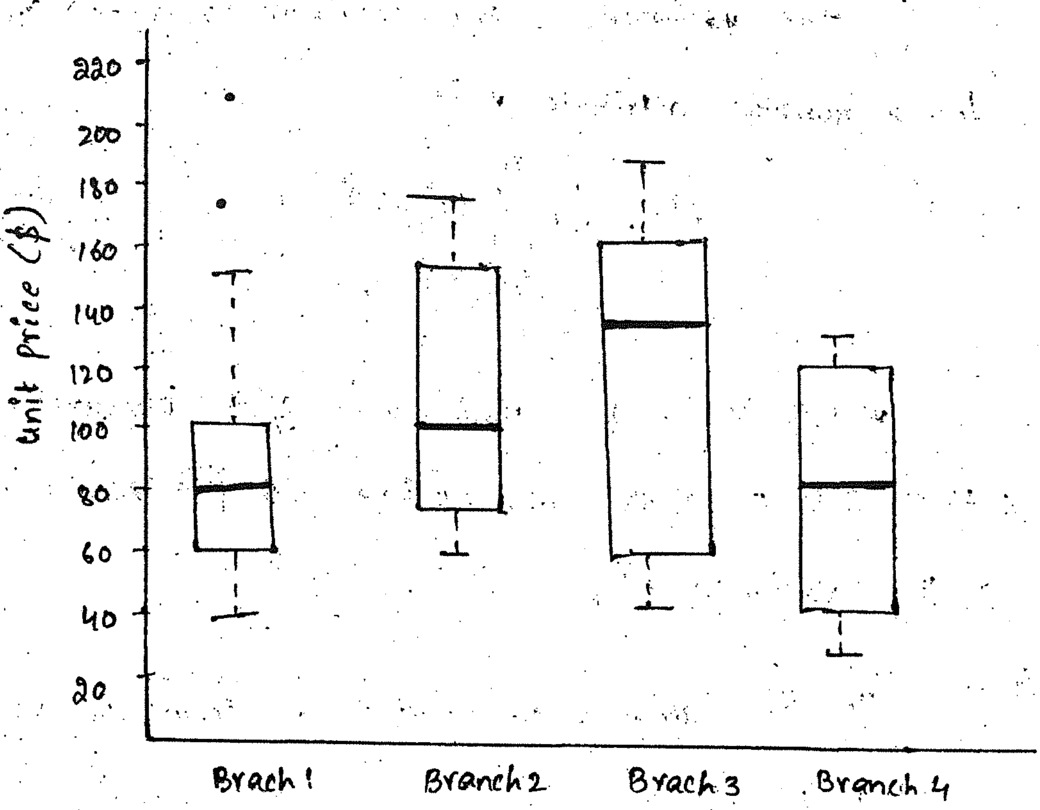


Fig. 3 Box plot for the unit price data for items sold at four branches of all Electronics during a given time period.

Variance and Standard Deviation:

Variance and standard deviation are measures of data dispersion. They indicate how spread out a data distribution is.

A low standard deviation means that the data observations tend to be very close to the mean, while high S.D indicates that the data are spread out over a large range of values.

The variance of N observations $x_1, x_2, \ldots, x_N$ for a numeric attribute x is

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

4.

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

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

& the S.D σ, of the observations is the square root of the variance σ^2.

Ex: In the above observation we found $\bar{x} = 58{,}000$ and $N = 12$.

$$\sigma^2 = \frac{1}{12}(30^2 + 36^2 + \cdots + 110^2) - 58^2$$

$$= 379.17$$

$\sigma \doteq \sqrt{379.17}$

$\simeq 19.47$.

The basic properties of S.D σ, as a measure of spread are as follows:

1, σ measures spread about the mean and should be considered only when the mean is chosen as the measure of center.

2, $\sigma = 0$ only, when there is no spread, i.e, when all observations have same value. other wise $\sigma > 0$.

4.2.3 Graphic Displays of Basic Statistical Descriptions of Data :-

The graphic displays of basic statistical descriptions include quantile plots, quantile - quantile plots, histograms & scatter plots. Such graphs are useful for the visual inspection of data, which is useful for data preprocessing. first three of these show univariate distributions, and while scatter plots show bivariate distributions.

Quantile plot:-

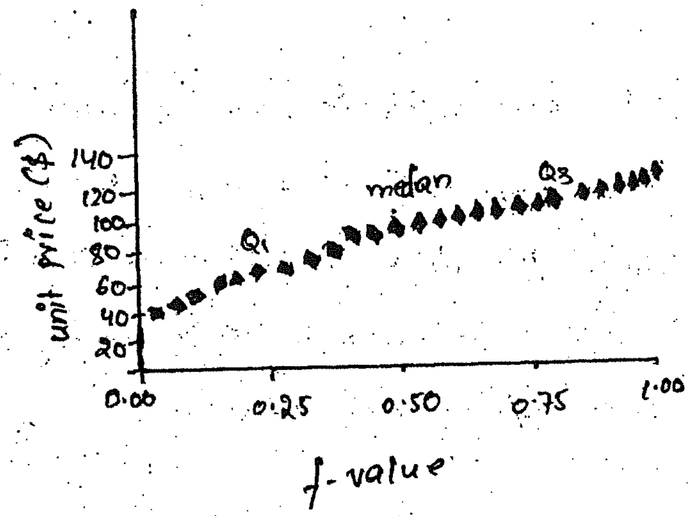
A quantile plot is a simple, we cover and effective way to have a first look at a univariate data distribution. It displays all of the data for the given attribute. then, it plots quantile information.

Ex! let x_i, for $i = 1$ to N, be the data sorted in increasing order so that x_1 is the smallest element & x_N is the largest for some ordinal or numeric attribute X.

A set of unit price Data for Items Sold at a Branch of All Electronics

unit price ($)	count of Items sold
40	275
43	300
47	250
–	–
74	360
75	515
78	540
–	–
115	320
117	270
120	350

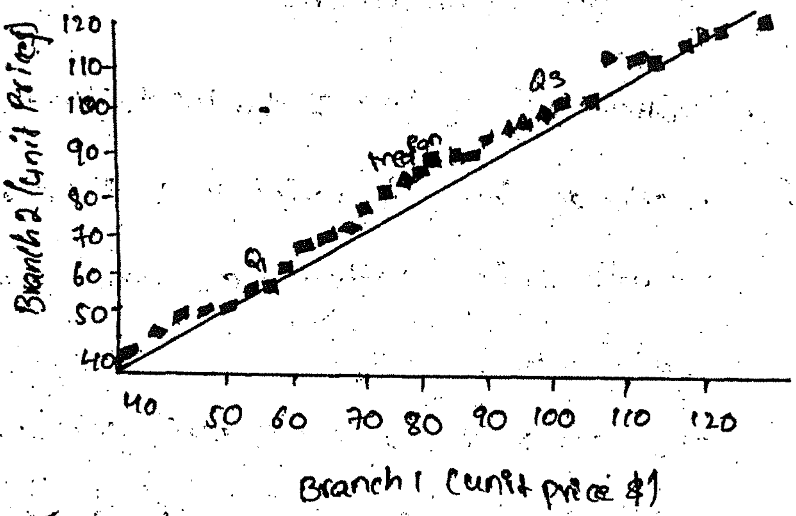
Table 2.1



A quantile plot for the unit price data of Table 2.1.

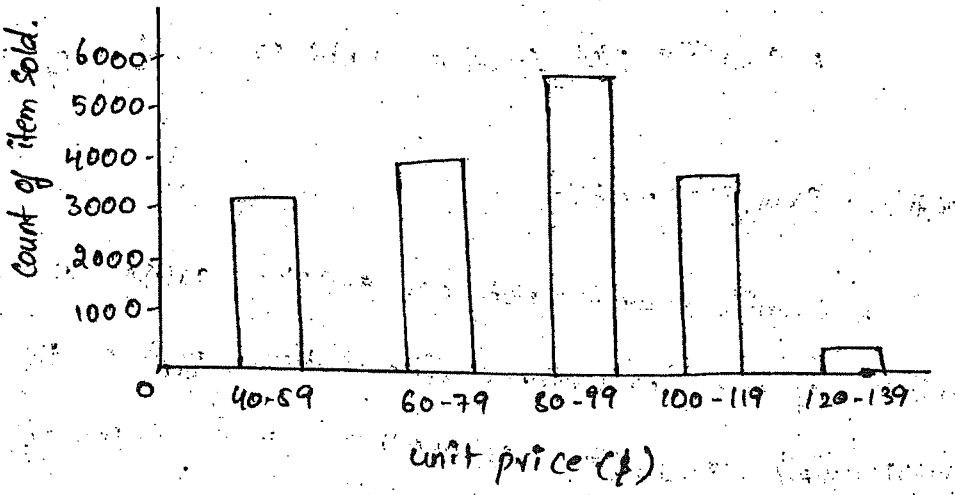
Quantile - Quantile plot :-

A quantile - quantile plot, or q-q plot, graphs the quantiles of one univariate distribution against the corresponding quantiles of another. It is a powerful visualization tool in that it allows the user to view whether there is a shift in going from one distribution to another.



Histograms:-

A histogram is an accurate representation of numerical data. To construct a histogram for an attribute x, partitions the data distributions of x into buckets. Each bucket is represents only a single attribute of value/frequency pair.

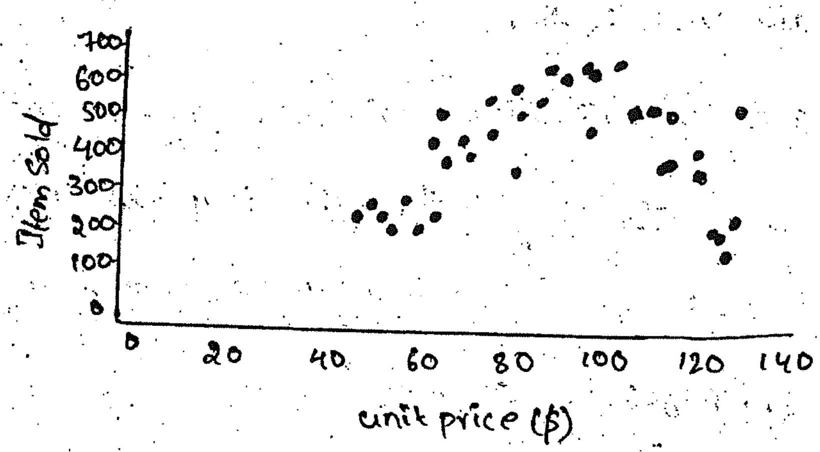


A histogram for table 4.1

Scatter plots and Data Correlation:-

A scatter plot is one of the most effective graphical methods for determining if there appere to be a relationship, pattern or trend b/w two numeric attributes.

To construct a scatter plot, each pair of value is treated as a pair of coordinates in an algebraic and plotted as points in the plane.



scatterplot for the table 2.1

The scatter plot is useful to represent the bivariate data to clusters of points & outliers, or to explore the possibility of correlation relations. Two attributes, X, and Y, are correlated if one attribute implies to other. correlation can be +ve, -ve or null.

(a)
Positive

(b)
negative.

Scatter plots can be used to find (a) positive (or) (b) negative b/w attributes.

4.3 Data Visualization

Data visualization aims to communicate data clearly and effectively through graphical represents. Data visualization has been used extensively in many applications.

for ex: at work for reporting, managing business operations, and tracking progress of tasks.

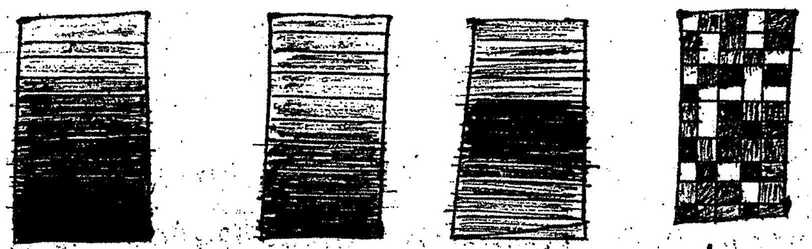
More popularly, we can take advantage of visualization techniques to discover data relationships that are otherwise not easily observable by looking at the raw data.

4.3.1 pixel oriented visualization Techniques :-

A simple way to visualize the value of dimension is use a pixel where the colors of the pixel reflects the dimension's value. For the data set of m dimensions, pixel-oriented technique creates m windows on the screen, one for each dimension

Ex: AllElectronics maintains a customer info table which consists of income, credit limit, transaction volume and age. we can analyze correlation b/w income and the other attributes by visualization?

we can sort all the customers in income ascending order, & used this order to layout the customer data in four visualization windows, as follows.



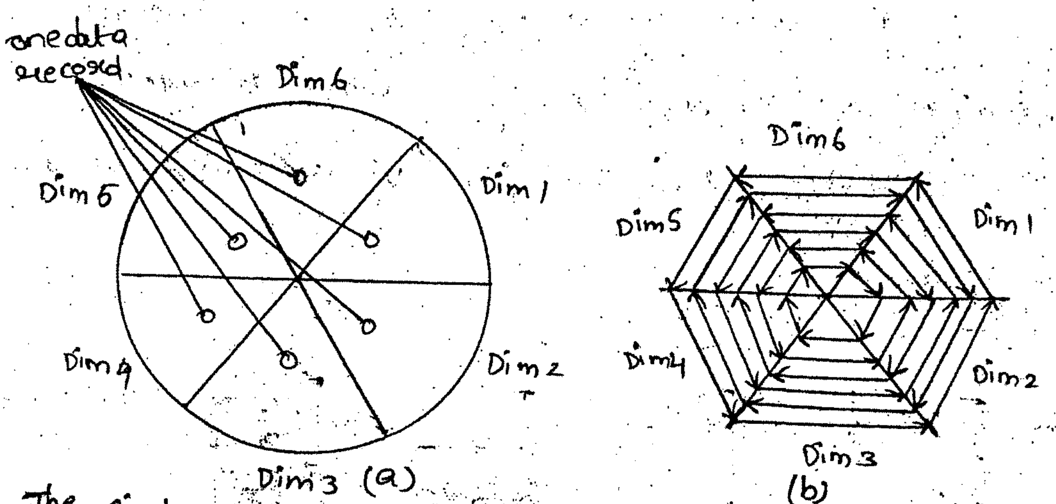
(a) income (b) credit_limit (c) transaction_volume (d) age.

pixel oriented visualization of four att.

Credit_limit increases as income increases; whose income is in the middle range are more likely to purchase more from AllElectronics; there is no clear correlation b/w income & age.

Filling a window by laying out the data records in a linear way may not work well for a wide window. we can lay out the data records in a space-filling curve to fill the windows.

A It is a curve with a range that covers the entire n-dimensional unit hyper cube. Since the visualization windows are 2-D, we can use any 2-D space filling curve as follows.

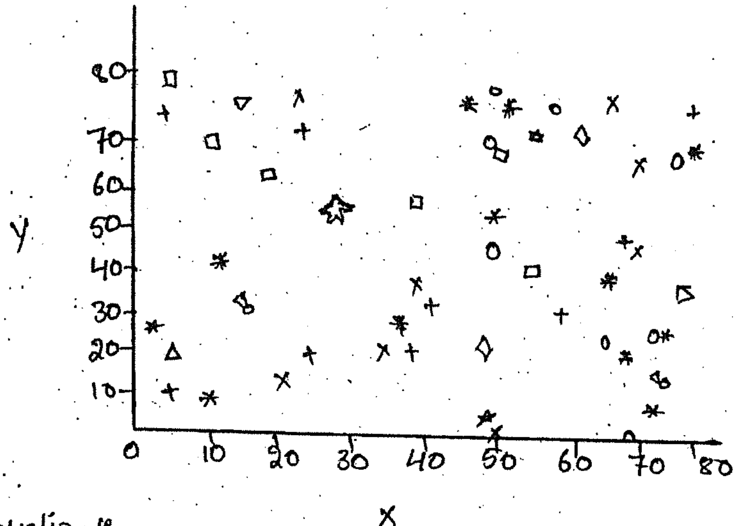


The circle segment technique (a) Representing a data record in circle segments (b) laying out pixels in circle segments.

4.3.2 Geometric projection visualization Techniques :-

Geometric projection techniques help users find interesting projections of multidimensional data sets. The central challenge the geometric projection techniques try to address is how to visualize a high diem dimensional space on a 2-D Display.

A Scatter plot displays 2-D data points using Cartesian coordinates. A third dimension can be added using different colors or shapes to shapes to represent diff. data points.



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

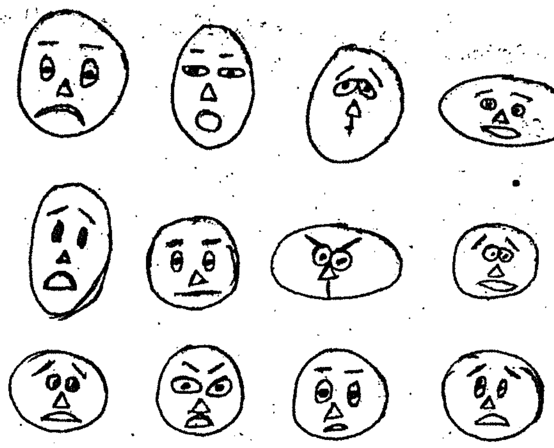
where x and y are two spatial attributes and the third dim: is represented by different shapes. Through this visualization, we can see that the points of type "+" and "x".

A scatter plot matrix technique is a useful extension to the scatter plot. For an n dimensional data set, a scatter plot matrix is an n×n grid of 2-D scatter plots that provides a visualization of each dim: with every other dim: The scatter plot matrix becomes less effective as the dimensionality increases.

4.3.3 :- Icon Based Visualization Techniques

Icon-based visualization techniques use small icons to represent multidimensional data values. There are two popular icon-based techniques chernoff faces and stick figures.

Chernoff faces were introduced in 1973 by statistician Herman chernoff. They display multi dimensional data of up to 18 variables as a cartoon human face.



Chernoff faces. Each face represents an n-dimensional data point $(n \leq 18)$

4.3.4 Hierarchical Visualization Techniques :-

The visualization techniques discussed so far focus on visualizing multiple dimensions simultaneously. However, for a large data set of high dimensionality, it would be difficult to visualize all dimensions at the same time. Hierarchial visualization techniques partition all dimensions into subsets. The subsets are visualized in a hierarchical manner.

all ~~words~~ "worlds-within-worlds" also known as n-vision, is a representative hierarchical visualization method. Suppose we want to visualize a 6-D data set, where the dimensions are $F, X_1 \cdots X_5$. we can first fix the values of dimesions X_3, X_4, X_5 to some selected values, say C_3, C_4, C_5. we can then visualize F, X_1, X_2 using 3-D plot.

2.3.5 Visualizing Complex Data and Relations:-

Visualization techniques were mainly for numeric data. Recently, more and more non-numeric data, such as text and social networks, have become available.

for ex; many, people on the web tag various objects such as pictures, blog entries and product reviews.

sonal

L.

4.4 Measuring Data Similarity and Dissimilarity :-

In data mining applications, such as clustering, outliers analysis, and nearest-neighbour classification, we need ways to assess how ~~dike~~ alike or unalike objects are in comparison to one another.

for ex; a store may want to search for clusters of customer objects, resulting in group of customers with similar characteristics. (income, are of residence, age). Such information can then be used for marketing.

A cluster is a collection of data objects such
a cluster are similar to one
the objects.

so employs clustering-based
potential outliers as objects
to others.

ct similarities can also be used
ation schemes where a given
label based on its similarity
the model.

[Total No. of Printed Pages-2

[45113]

M.C.A. DEGREE EXAMINATIONS

FOURTH SEMESTER

Paper - III : DATA MINING CONCEPTS AND TECHNIQUES

(W.E.F 2019-20 Admitted Batch)

Time : 3 Hours Maximum : 75 Marks

Answer all questions.

SECTION - A (4×15=60)

1. a. What is multi - dimensional analysis? How it is implemented through OLAP? Explain various types of OLAP systems used for multi - dim...

4.4.1 Data Matrix versus Dissimilarity Matrix :-

we studying the central tendency, dispersion & Spread of observed values for some attribute X. That are one-dimensional, i.e, described by a single attribute. we talk about objects described by multiple attributes. There fore, we need a change in notation.

Suppose that we have n objects (Ex; persons, items or courses) described by p attributes (age, height, weight or gender). The objects are $x_1 = (x_{11}, x_{12}, \dots x_{1p})$, $x_2 = (x_{21}, x_{22}, \dots x_{2p})$ and so on, where x_{ij} is the value for object x_i of the j^{th} attribute.

Main memory-based clustering and nearest-neighbour algorithms typically operate on eigther of the following two data structures.

1) Data matrix (object-by-attribute structure) :—

This is also called as object-by-attribute structure. This structure stores the n data objects in form of relational table, or n x p matrix (n objects x p attributes).

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

of

2) <u>Dissimilarity matrix :-</u>

This is also called as object-by-object structure. This structure stores a collection of proximities that are available for all pairs of n objects. It is also represented as n x n table.

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

where $d(i,j)$ is the measured dissimilarity (or) difference b/w objects i & j. $d(i,j)$ is a non-ve number that is close to 0 when objects i & j are highly similar or near each other.

$d(i,i) = 0$ i.e, the difference b/w an object and itself is 0.

Measure of similarity can be expressed as a function of measures of dissimilarity.

Ex :- for nominal data

$$sim(i,j) = 1 - d(i,j)$$

where $sim(i,j)$ is the similarity b/w objects i and j.

A data matrix is made up of two entities they are rows (for objects) and columns (for attributes) therefore, the data matrix is often called as "two-mode" matrix. The dissimilarity matrix contains one kind of entity & so is called as "one-mode" matrix.

4.4.2 Proximity Measures for Nominal Attributes:-

A nominal attribute can take on two or more states. For example: map-color is a nominal attribute that may have say five states: red, yellow, green, pink and blue.

let the no. of states of a nominal attribute be M. The states can be denoted by letters, symbols or a set of integers, such as 1, 2, ... M.

dissimilarity b/w objects described by nominal attributes

The dissimilarity b/w two objects i and j can be computed based on the ratio of mismatches:

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

where m is the no. of matches i.e., no. of attributes for which i & j are in the same state. and p is the total no. of attributes describing the objects.

)

e"

rix

M.

Ex: Suppose that we have the sample data of as follows, expect that only the object-identifier and the attribute test-1 are available; where test-1 is nomial.

Table 4.2. A simple Data Table Containing Attributes of Mixed type.

Object Identifier	test-1 (nominal)	test-2 (ordinal)	test-3 (numeric)
1	code A	excellent	45
2	code B	fair	22
3	code C	good	64
4	code A	excellent	28

lets compute the dissimilarity matrix.

$$\begin{bmatrix} 0 & & & \\ d(2,1) & 0 & & \\ d(3,1) & d(3,2) & 0 & \\ d(4,1) & d(4,2) & d(4,3) & 0 \end{bmatrix}$$

we set p=1, sothat $d(i,j)$ evaluates to 0 if the i&j are match, & ~~if~~ 1 if the Objects differ. Thus, we get

$$\begin{bmatrix} 0 & & & \\ 1 & 0 & & \\ 1 & 1 & 0 & \\ 0 & 1 & 1 & 0 \end{bmatrix}$$

we see all the objects are dissimilar except objects 1 and 4 i.e $d(4,1) = 0$.

Alternatively, similarity can be computed as

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

4.4.8:- Proximity Measure for Binary Attributes:-

A binary attribute has only one of two states: 0 or 1 where 0 means that the attribute is absent, and 1 means that is present. ~~To~~ Treating binary attributes as if they are numeric can be misleading. Therefore, methods ~~to~~ specific to binary data are necessary for computing dissimilarity.

One approach involves computing a dissimilarity matrix from the given binary data. If all binary attrs are thought of a having the same weight, we have the 2x2 contingency table as follows

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

Contingency Table for Binary Attributes.

where 'q' is the number of attributes that equal to 1 for both objects i & j, 'r' is the no. of attributes that equal '1' for object i, but equal '0' for object j, s is the number of attributes that equal '1' for object 'i' but equal 0 for object j, and 't' is the no. of attributes that equal '0' for both objects i & j. The total no. of attributes is p

$$p = q + r + s + t.$$

For symmetric binary attributes, each state is equally valluable. Dissimilarity that is based on symmetric binary attributes is called symmetric binary dissimilarity. If objects i and j are described by symmetric binary attributes, then the dissimilarity b/w i & j is

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

For asymmetric binary attributes, the two states are not equally important, given two asymmetric binary attributes, the agreement of two 1's is then considered more significant than that of two 0's. Therefore, such binary attributes are often considered "monary" (having one state). The dissimilarity based on these attributes is called asymmetric binary dissimilarity

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

where the no. of negative matches t, is considered unimportant and is thus ignored.

when both Symmetric & asymmetric binary attributes occur in the same data set, the mixed attributes approach can be applied.

Ex:- Suppose the patient record table contains the attribute name, gender, fever, cough, test-1, test-2, test-3, & test-4.

where name is an object identifier, gender is a symmetric attribute, and the remaining attributes are asymmetric binary.

name	gender	fever	cough	test-1	test-2	test-3	test-4
Jack	M	Y	N	P	N	N	N
Jim	M	Y	Y	N	N	N	N
Mary	F	Y	N	P	N	P	N
:	:	:	:	:	:	:	:

fig.4 Relational Table where patients are Described by Binary Attributes.

4 u a c

For Asymmetric attribute values, the values Y and P be set to 1, and the value N (no or negative) be set to '0'.

The distance b/w each pair of the three patients Jack, Mary & Jim is

$$d(\text{Jack}, \text{Jim}) = \frac{1+1}{1+1+1} = 0.67$$

$$d(\text{Jack}, \text{Mary}) = \frac{0+1}{2+0+1} = 0.33$$

$$d(\text{Jim}, \text{Mary}) = \frac{1+2}{1+1+2} = 0.75.$$

These measurements suggest that Jim & mary are unlikely to have a similar disease. and Jack and mary are the most likely to have a similar disease.

4.4.4 Dissimilarity of Numeric Data: Minkowski Distance

We describe distance measures that are commonly use for computing the dissimilarity of objects by numeric attributes. these measures include the Euclidean, Manhattan and minkowski distances.

In some cases, the data are normilized before applying distance calculations. This involves transforming the data to fall within a smaller or common range such as [-1, 1] or [0.0, 1.0].

The most popular distance measure is "Euclidean distance". let $i = (x_{i1}, x_{i2} \cdots x_{ip})$ and $j = (x_{j1}, x_{j2} \cdots x_{jp})$ be two objects described by p numeric attributes. The Euclidean distance b/w objects i & j is

$$d(i,j) = \sqrt{(x_{i1} - x_{j1})^2 + (x_{i2} - x_{j2})^2 + \cdots + (x_{ip} - x_{jp})^2}$$

Ex: let $x_1 = (1,2)$ & $x_2 = (3,5)$, the Euclidean distance b/w two objects is

$$d(x_1, x_2) = \sqrt{(1-3)^2 + (2-5)^2} = \sqrt{2^2 + 3^2} = 3.61.$$

Another well-known measure is the Manhattan distance or city block distance, name so because it is the distance in blocks b/w any two points in a city.

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

Ex: Manhattan distance for the above example

$$d(x_1, x_2) = |1-3| + |2-5| = 2+3 = 5.$$

Both the Euclidean and the Manhattan distance satisfy the following mathematical properties.

1) Non-negativity: $d(i,j) \geq 0$. : Distance is a non-ve number.
2) Identity of indiscernibles: $d(i,i) = 0$. The distance of an object to itself is '0'.

3) Symmetry: $d(i,j) = d(j,i)$: Distance is a symmetric function.

x_{jp}

ii) Triangle inequality: $d(i,j) \le d(i,k) + d(k,j)$.

"Minkowski distance" is a generalization of the Euclidean and Manhattan distances.

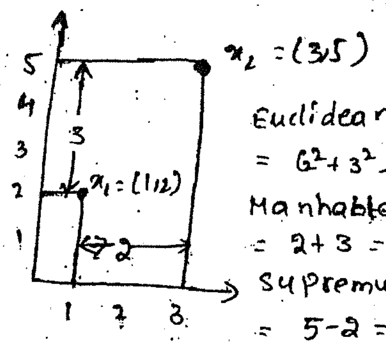
$$d(i,j) = \sqrt[h]{|x_{i1}-x_{j1}|^h + |x_{i2}-x_{j2}|^h + \cdots + |x_{ip}-x_{jp}|^h}$$

where h is a real number such that $h \ge 1$. It represent the manhattan distance when $h=1$ and Euclidean distance when $h=2$.

The supremum distance is a generalization of the Minkowski distance for $h \to \infty$. To compute it, we find the att. f that gives the maximum difference in values b/w two objects.

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

$$= \max_{f}^{p} |x_{if} - x_{jf}|$$



Euclidean distance
$= (2^2 + 3^2)^{1/2} = 3.61$
Manhattan distance
$= 2 + 3 = 5$
supremum distance
$= 5 - 2 = 3$.

Euclidean, Manhattan, supremum distance between two objects

4.4.5 proximity Measure for ordinal Attributes :-

The values of an ordinal attributes have a meaningful order or ranking. The treatment of ordinal attributes is quite similar to that of numeric attributes when computing dissimilarity b/w two objects.

suppose f is an attribute from the set of ordinal attributes describing n objects. The dissimilarity computation w.r.to f involves the following steps:

1, The value of 'f' for ith object is x_{if}, and f has M_f ordered states, representing the ranking $1, \ldots M_f$ replace each x_{if} by its corresponding rank $r_{if} \in \{1 \ldots M_f\}$

2, We perform data normalization by replacing the rank r_{if} of the ith object in the fth attribute by

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

3, dissimilarity can be computed using any of the distance measures described in numeric attributes, using z_{if} to represent the f value for ith object.

4.4 to ec c ((

gfu?
es

4.4.6 Dissimilarity for Attributes of Mixed Types :-

One approach is to group each type of attribute together, performing seperate data mining analysis for each type eg: clustering. this is feasible if these analyses derive compatible results.

suppose that the data set contains p atts of mixed type. The dissimilarity b/w i & j objects is defiend as

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

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

1) x_{if} or x_{jf} is missing (or)
2) $x_{ij} = x_{jf} = 0$ and attribute f is asymmetric binary.

Other wise, $\delta_{ij}^{(f)} = 1$.

The contribution of attribute f to the dissimilarity b/w i & j i.e, $d_{ij}^{(f)}$ is computed dependent on its type:

* If f is numeric $d_{ij}^{(f)} = \frac{|x_{if} - x_{jf}|}{max_h x_{hf} - min_h x_{hf}}$,

where h runs over all nonmissing objects for att: f.

* If f is nominal or binary : $d_{ij}^{(f)} = 0$ if $x_{if} = x_{jf}$; otherwise $d_{ij}^{(f)} = 1$.

* If f is ordinal: compute the ranks r_{if} and $z_{if} = \frac{r_{if} - 1}{M_f - 1}$, and treat z_{if} as numeric.

4.4.7 Cosine Similarity :-

A document can be represented by thousands of attributes, each recording the frequency of a particular word or phrase in the document. Thus, each document is an object represented by a "term-frequency vector".

Cosine Similarity is a measure of similarity that can be used to compare document (or) give a ranking of documents w.r.to a given vector of query words. Let x & y be two vectors for comparison. using cosine measure as a similarity function, we have

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

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

Ex: - suppose that x & y are the first two term-frequency vectors in the below table. i.e $x = (5,0,3,0,2,0,0,2,0,0)$ & $y = (3,0,2,0,1,1,0,1,0,1)$

Document Vector or Term-Frequency Vector

Document	team	coach	hockey	baseball	soccer	penalty	score	win	loss	season
Document 1	5	0	3	0	2	0	0	2	0	0
Document 2	3	0	2	0	1	1	0	1	0	1
Document 3	0	7	0	2	1	0	0	3	0	0
Document 4	0	1	0	0	1	2	2	0	3	0

compute the cosine similarity b/w 2 vectors x & y

$$x^t \cdot y = 5\times3 + 0\times0 + 3\times2 + 0\times0 + 2\times1 + 0\times1 + 0\times0 + 2\times1 + 0\times0 + 0\times1 = 25$$

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

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

$$sim(x,y) = \frac{25}{6.48 \times 4.12} = 0.94.$$

Therefore, if we were using the cosine similarity measure to compare these documents, they would be considered quite similar.

5th unit

Architecture of Data mining System :-

To get the efficient architecture of DM System : The DM System must be integrated with db (or) DWH. If the DM System integrated with db or DWH then, we have to consider the following coupling schemas.

① No coupling
② loosly coupling
③ Semitight Coupling
④ Tight Coupling.

① **No coupling :-** Here, DM system doesnot integrate with db or DWH. These types of systems contains the following drawbacks.

① If the DM System doesn't integrate with the db then it doesn't provide any Performance, Scaliability etc..,

② If the DM System doesn't integrate with the DWH then it doesnot provide data cleaning, data transformation, Data integration etc..,

These Systems extract the data from the files because these Systems doesn't contain any data Structures or any algorithms & the results are send to file.

∴ No coupling means poor design.

2. Loosely Couplings:

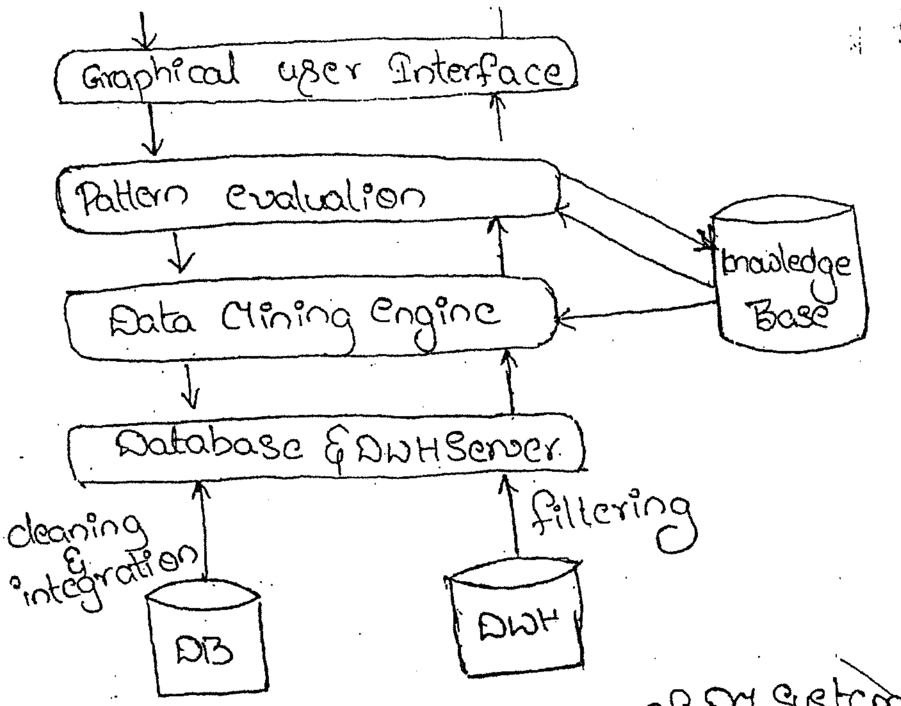
Here, DM system extract the essential features from db or DWH. These systems extract the data from centralized depository & the results are send to any file or db or DWH. These systems far better than the no of coupling systems.

3. Semitight Couplings:

Here, DM Systems integrate with db or DWH to get the DM primitives i.e..., store, retrieve, histogram analysis, join indexing etc..,

4. Tight couplings:

Here, DM Systems fully integrated with db or DWH to get all the features & to provide the integrated information enviro to provide the integrated architecture. finally meet. This is the desired architecture. Using this tight coupling we design the typical architecture of DM.



fig(a): Typical Architecture of DM System

The typical architecture of DM System contains the following Components:

① DB, DWH (or) Any information Repository:

Here, we extract the data from any information Repository db (or) DWH or any then we apply the data cleaning, data transformation, data integration & finally data load. This data is loaded it into db & DWH Server.

② DB (or) DWH Server:

It contains the data according to user Specification.

③ Knowledge Base :-

Here, enduser can get the knowledge from the different data Sources. If the end user as the knowledge then the DM process is Simplified.

④ Data mining engine :-

Here, we apply the DM functions.

⑤ Pattern evolution :-

Here, interesting patterns are evaluated.

⑥ GUI :-

It provides the communication b/w end user & DM System. i.e., through this enduser Present the queries to the DM system. and also discovered Patterns are visualized to the enduser by using different visualization Techniques.

Concept Description: characte~~ristic~~ization & Comparison :-

The DM is mainly classified into two

1. Descriptive mining
2. Predictive mining

The descriptive mining analyses the collected data set in the form of summeris-ation. i.e.., it explains the general charact-eristics of data.

The Predictive mining analyses the data in order to construct one or more models using this models we can predict the behaviour of a new data set.

The descriptive mining is called as concept description.

what is Concept description :-

The concept is nothing but the collec-tion of data. for eg.., frequently-buys, graduate-students etc..., The concept description summerizes and explains the general characteristics of data. The concept description is also called as class description. It contains the two types.

1. class characterization
2. class comparison

The class characterization explains the characteristics of collected data.

The class comparison compares the two or more collections of data.

Comparison b/w OLAP & Concept Descriptions

a) Aggrigate functions vs complex datas.

The DWH & OLAP Systems the data is stored in the form of multi dimensional data model. i.e.., Data cubes. The data cubes contains the dimensions & measures. The measures are analysed by using the aggregate functions, i.e.., Sum(), Avg() etc.., But the concept description in DB handle the complex data types like non numeric, text, Special, multi media etc...,

b) Manual vs automations.

The DWH and OLAP Systems we have to Specify the dimensions & also we have to Specify the OLAP operations. But the Concept description in db. we select the dimensions & also we select the data at multiple levels.

Data Generalization & Summerization Based characterization:-

For eg.., the data is store in large db. then, this data generalization provides the data abstraction from low

Conceptual level to high conceptual

The data generalization mainly classified into two .

(1) Data cube approach - we have to see the 2nd chapter .

(2) Attribute Oriented Induction (AOI) Approach .

(a) **Attribute Oriented Induction (AOI) Approach :-**

This is introduced in the year of 1989 . This is the best Approach for data generalization & Summerization based data generalization & characterization . It contains the two steps .

1. Select Task Relevant data .
2. Data Generalization .

1. **Select Task Relevant data :-**

for eg.... end user analyse the characteristics of graduate students from the Big university - DB & the attributes are name , gender , major , Birth - Place , Birth - date , Residence & Phone number .

Note :- The data cube approach is based on DWH orientation . But the AOI approach based on Relational Database .

on, this is specified in DMQL

Syntax :-

use Big-university -DB
mine characteristics as "science-students"
in relevance to name, gender, major,
birth-place, birth date
residence, phone #
from student
where status in "graduates".

This is transformed into relation query

Syntax :

use Big-university - DB
select name, gender, major, birth-place, birth-date,
residence, phone #.
from student
where status in {"MCA", "MBA", "MA", "Mcom"}.

Once this query is executed. we get the table. This table is called as Task relevant data or initial working relation. This is shown in below.

name	gender	major	birth-place	birth-date	residence	phone #
John	M	Science	Hyd, AP India	12/07/76	123 Bhilk Hyd	234576
Laaa	F	Engineering	vancouver BC, Canada	10/08/70	Mainst vancouver	997-86

Data Generalisation:

Again it contains two steps.

(1) Attribute Removal

(2) Attribute Generalization.

(1) Attribute Removal:

It contains the two rules.

Rule 1: If the attribute as large no of distinct values and no hierarchy is defined then remove that attribute from the relation.

Rule 2: If the attribute as high concept hierarchy is defined then remove the low concept hierarcy & replaced with high concept hierarchy.

(2) Attribute Generalization:

If the attribute as the large no of distinct values & concept hierarchy is defined then we select the generalization threshold value. After applying the AOI rules in Table 1.1 we get the following steps.

(1) name: If contains the large no of distinct values & no concept hierarchy is defined. ∴ Remove that attribute from the working relation.

(2) Gender: It contains 2 values and it remains the same.

(3) major: It contains 3 values i.e., Science engineering, & business. It remains

the same.

④ birth-Place :

It's concept hierarchy is defined city, state & country then this is replace with birth country

⑤ birth-date :

This attribute is transformed into age & age is transformed into age-range.

⑥ Residence : It's concept hierarchy is defined street to city. This attribute is replaced with residence city.

⑦ Phone # : It contains the large no of distinct values & no concept hie-rarchy is defined.

∴ remove it from the relation.

Here, similar tuples are merged & specified by using the count attribute Then, finally we get the table 1.2

gender	major	birth-country	age-range	residence-city	count
M	Science	India	20...30	Hyd	20
F	Engineering	Canada	30....40	Vancouver	30

Table (1.2) After applying AOI Rules, the final Relational Table.

* **Efficient Implementation of attribute oriented Induction:**

For efficient implementation of AOI we use the following algorithm.

Algorithm:- Attribute . oriented . Induction.

i/p :- i, DB, a Relational db

ii, DMQueries , A DM Query.

iii, A-list, list of Attributes

iv, Gen(ai), It is the Generalization operators . for each ai.

v, A.thersh. value (ai), It is the Generalization Threshold value for each ai.

o/p :. P ← Prime relation

methods :. W ← task- relevant-data (DB- DM Query);

Here 'W' is the initial working relatio

① P ← prepare . for . generalization (W)

(a) If the attribute ai contains the larg no of distinct values & no Concept hierarchy is defined then remove that attribute from the working relati

(b) If the attribute 'ai' high level conce hierarchy is defined then remove the low level Concept hierarchy & replace with high level concept hierarchy.

bb⑤ P ← Generalization(W);

(a) If the attribute contains the large no of distinct values & concept hierarchy is defined. then we select the Threshold range.

(b) If the generalization contains the similar tuples these are merged by using the count attribute.

Simple AOI Algorithm.

* Presentation of Derived Generalization:

we apply the AOI on Relational DB then we get the generalization Relation. This is Presented at end user. by using several visualization techniques i.e., cross Table, Barcharts, Piecharts etc.,

for eg., consider the sales table for all electronics is shown in below.

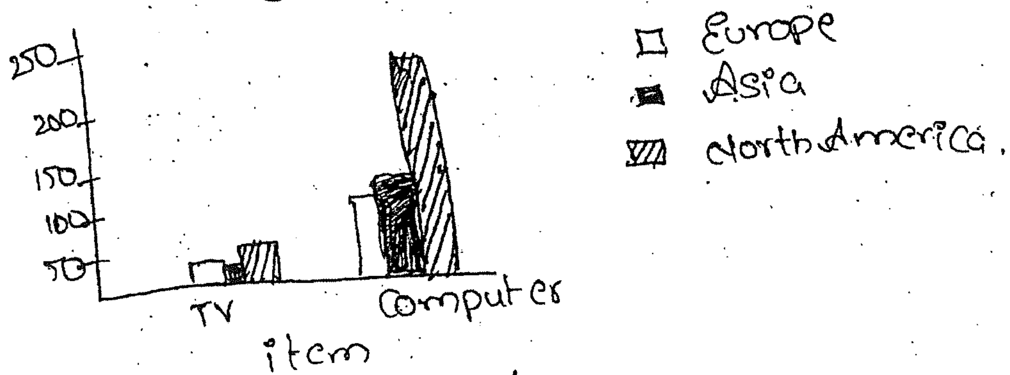
location	item	Sales (in dollars)	count (in thousands)
Europe	TV	15	200
Asia	TV	12	350
North America	TV	28	400
Europe	Computer	120	1000
Asia	"	130	1200
North America	"	250	1800

Sales (a) sales of all electronics

This table information is represented in cross table.

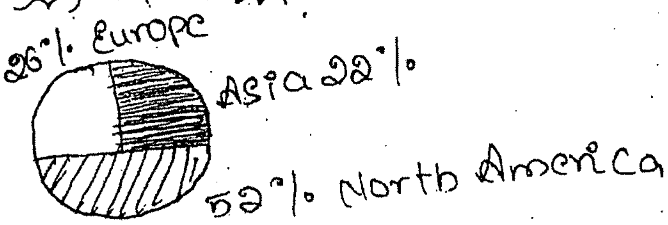
location/item	TV Sales	TV Count	Computer Sales	Computer Count
Europe	15	200	120	1000
Asia	12	350	130	1200
North America	18	400	850	1800

fig(b) . cross Table.

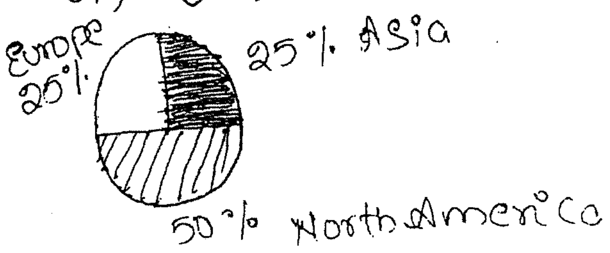


fig(c) Bar chart.

Pie chart for TV sales :



Pie chart for Computer Sales :



* Analytical Characterization :-

using this attribute relevance we find the weekly relevant & irrelevant attributes These are removed from Concept description using this we also find the most relevant attributes these are included in concept description.

why attribute relevance is required :-

The DWH & OLAP systems contains the drawbacks Of enduser has to specify the dimensions & also he has to specify the high conceptual level. This is specified by using stmt " generalize dimension location to the country level".

The enduser does not know the attrib-ute required to achieve the DM task then he specify all the attributes by using the stmt in relevance to *'. But, this does not give the accurate data -:. To identify mostly relevant attributes & delete weekly relevant attributes. we require the attribute relevance analysis.

Methods for attribute Relevance Analysis:-

In this method we integrate the information gain analysis with data dimension analysis

Finally, we find the information gain for each attribute & then the highest information gain attribute is added in concept description & remove the lowest information gain attribute from the concept description.

For eg.., Consider the training samples 'S' & also each training sample class label must be node. To identify the class label we use one attribute for eg.., Consider the attribute status using this attribute we find the whether he is graduate student or under graduate student.

Let us consider the 'm' classes the sample 's' contains 'si' samples in class C_i for $i = 1, 2, \ldots m$ then the sample belon to class 'C_i' with the probability $\frac{s_i}{s}$ where s - Total no of samples then the expected information needed to classify the given sample.

$$I(s_1, \ldots s_m) = -\sum_{i=1}^{m} \frac{s_i}{s} \log_s^{s_i/s} \longrightarrow (1)$$

Let us consider the attribute partition values $\{a_1, a_2, \ldots a_v\}$ then th attribute partition the given set 'S' into $\{S_1, \ldots S_v\}$ then Let us consider the S_j falling C_{ij} then entrophy of A.

$$E(A) = \sum_{j=1}^{v} \frac{(S_{ij} + \ldots + S_{mj})}{S} \times I(S_{ij}, \ldots, S_{mj}) \quad (2)$$

finally, information gain$(A) = I(S_1, \ldots S_m) - E(A)$

This method containg the following steps.

1. **Data Collection:-**

we collect the data for target class & contrasting class. The target class Containg the data that is to be characterized. The contrasting the class Contains the comparitive data ~~that~~ is ~~to be characterized. The C~~

2. **Apply AOI Rules:-**

Here, we apply the AOI Rules.

3. **Remove irrelevant & weakly Relevant Attributes:**

It containg three steps

Step 1:- Comput the expected information gain

Step 2:- Comput the entrophy value.

Step 3:- Comput the information gain. The highest information gain attributes is added in Concept description.

Example for class characterization:-

It containg the following steps

Step 1:- collect the data for target class. i.e.., graduate Students. This is shown in below.

gender	major	birth-Country	age-range	Count
M	Science	CANADA	20.......25	24
F	Science	CANADA	20.......25	30
M	Science	USA	20.......25	30
M	Engineer	CANADA	25.......30	20
F	Engineer	USA	25.......30	16

fig(a) Data for target class i.e.., graduate Students.

Step 2:- collect the data for Contrasting class i.e..., under graduate Students.

gender	major	birth-country	age-range	Count
M	Science	CANADA	20.......25	20
F	Science	USA	20.......25	22
M	Engineer	CANADA	25.......30	20
F	Engineer	USA	25.......30	26
M	Business	CANADA	30.......35	20
F	Business	USA	30.......35	22

fig(b) Data for Contrasting class i.e.., under graduate Students.

we already apply the AOI Rules & Remove the name, phone # because they contains the large no of distinct values & no concept hierarchy is defined.

The graduate student class is represented by 's_1' & under graduate student class is represented by 's_2'. The 's_1' contains the 120 tuples & 's_2' contains the 130 tuples. ∴ Expected information needed $I(s_1, s_2)$

$$I(s_1,s_2) = -\frac{120}{250}\log_2^{120/250} - \frac{130}{250}\log_2^{130/250}$$

$$= 0.99$$

Then consider the attribute major, for example.., major = "science".

$s_{11} = 84$, $s_{21} = 42$ then total - 126

expected information needed

$$I(s_{11}, s_{21}) = -\frac{84}{126}\log_2^{84/126} - \frac{42}{126}\log_2^{42/126}$$

$$= 0.78$$

major = "Engineer".

$s_{12} = 36$, $s_{22} = 46$ then total 82

expected information needed

$$I(s_{12}, s_{22}) = -\frac{36}{82}\log_2^{36/82} - \frac{46}{82}\log_2^{46/82}$$

major = "Business"

$s_{13} = 0$ $s_{23} = 42$ then

expected information needed.

$$I(s_{13}, s_{23}) = 0$$

Then, entropy value for major

$$E(major) = \frac{126}{250} \times I(s_{11}, s_{21}) + \frac{82}{250} I(s_{12}, s_{22}) +$$

$$\frac{42}{250} I(s_{13}, s_{23})$$

$$= \frac{126}{250} \times 0.78 + \frac{82}{250} \times 0.98 + \frac{42}{250} \times 0$$

$$= 0.77$$

∴ information gain for major.

$$G(\text{major}) = I(s_1, s_2) - E(\text{major})$$

$$= 0.99 - 0.77 = 0.22.$$

Similarly, we can find the information gain for other attributes.

i.e..., $G(\text{gender}) = 0.003$

$G(\text{birth-country}) = 0.004$

$G(\text{age-range}) = 0.59$

for eg..., Consider the attribute relevance Threshold is "0.1" then less than or equal to this value. Consider as the weakly relevant attributes.

∴ The attributes major & age-range is consider as the most relevant attributes these are included in Concept description. Similarly, the attributes gender & birth-Country are the weakly relevant attributes these are removed from Concept description.

* Class Comparison : Discrimination b/w different class :-

The end user always have interest on mining the data from one or more Contrasting classes. This is called as class Comparison or class Discrimination.

Rather than extracting the data from single class. i.e.., class characterization. The target class & contrasting class must have the similarities. for eg..., Person & item class not Comparable. But the sales of last 3years all Comparable & also graduate students vs under graduate students are Comparable.

* **Methods for class Comparison:**

It contains the following steps.

Step1:- Collect the data for target class & Contrasting class through DB queries.

Step2:- Apply the attribute relevance analysis.

Step3:- Apply the Generalization. i.e.., we select the generalization threshold range for specific attribute. Then this is applied on target class & Contrasting class to get the Prime relations.

Step4:- The derived class comparisons are Presented to enduser by using several visualization techniques.

step1:- for eg..., consider the big-university DB & the attributes are name, gender,

major, birth-place, birth-date, residence
& phone # to get the
use Big-university-DB
mine Comparison as "grade-vs-undergrade-students"
in relevance to name, gender, major, birth-place, birthdate, residence, phone #
for "graduate-students"
where status in "graduate"
versus "undergraduate-students"
where status in "undergraduate"
analyse count %
from student.

Once the query is executed, we get the data for target class i.e.., graduate students.

Name	gender	major	birth-place	birth-date	residence	phon
Mahi	M	Science	Hyd, AP INDIA	12-07-76	St-stree Hyd	234-5
Lee	F	Engineering	Vancover ABC, CANADA	12-07-70	ABS-St, vancouver	599-8

fig(a) Data for Target class i.e.., graduate Studer

Name	gender	major	birth-place	birth-date	residence	Phone#
RAM	M	Science	N.Y. ABC, USA	12-07-80	Main St, N.Y	123456
Anu	F	busine-ss	vancouve, XYZ, CANADA	10-09-77	SR-St, vancouver	234-57

Fig(b) Data for contrasting class i.e.., undergrad. Students

Step2: Apply the attribute relevance analysis after applying the attribute relevance analysis the attribute name, gender, birth-place, residence & Phone # are removed from the relational table.

Step3: Apply the generalization i.e., the attribute birth-place is transformed into age-range. The Prime relations are shown in below table.

major	age-range	count %
Science	20.......25	5.32 %
Engineering	25.......30	4.12 %

fig(a) Prime Relation for graduate Students.

major	age-range	Count %
Science	20.......25	3.32 %
Business	30.......35	2.14 %

fig(b). Prime Relation for undergraduate Students.

Step4:

These Prime relations are presented to enduser by using visualization techniques. In visualization technique we analyzed the

mec wit Pre use Sto cl d k d

measure is count %. for eg..., 5.32% of students with the age 20......25 & major in Science.

Presentation of class Comparisions:-

~~To Present the class Comparisons we~~ use the method. This method is called as Statistics discrimination (or) d. weight value.

for eg..., let 'S_A' be the any tuple & the target class 'C_j' then d-weight value of 'S_A' in 'C_j' is defined as the tuples containing target class by using total no of tuples.

$$d\cdot weight = count(S_A \in C_j) \Big/ \sum_{i=1}^{m} count(S_A \in C_i)$$

Here, m containing the total no of tuples i.e..., target class & contrasing class. for eg..., consider the data for graduate students and under graduate students.

Status	major	age. range	count
graduate	Science	20......25	210
undergraduate	Engineering	25......30	90

fig(a): Data for graduate & undergraduate Students

The d.weight value for target class i.e..., graduate students

$$d\cdot weight = \cancel{210}^{7} \Big/ \cancel{300}_{10} = 0.7 = 70\%$$

d-weight value for contrasting class.
i.e.., undergraduate students

d-weight = 90/300 = 0.3 = 30%

finally, if the student in major in science & with age 20.....25 then, there is a probability is that he is the 70% of graduate students.

* Class Description:-

The class description means it contains the 2. i.e.., class characterization & class comparison. for eg.., Consider the data shown below.

Location	TV	Computer
Europe	90	150
North America	210	520

fig(a): class table data for 2 locations.

Then d-weight value i.e.., TV

Europe (d-weight) = 90/300 = 0.3 = 30%

North America (d-weight) = 210/300 = 0.7 = 70%

d-weight value i.e.., Computer

Europe (d-weight) = 150/670 = 0.22 = 22%

North America (d-weight) = 520/670 = 0.78 = 78%

These d.weight values are shown in below.

Location / item	TV		Computer	
	count	d.weight	count	d.weight
Europe	90	30%	150	22%
North America	210	70%	520	78%

fig (b) d.weight value for fig(a)

from the fig(b) we can conclude that the north America Sales are very good

* Mining Statistical class Description from the large Database :-

Upto now we analyze the characteristics of a relational DB by using the Predefined aggregate functions like sum(), COUNT() etc But many of the applications end user want to analyize the characteristics of a large DB by using the measures mean, median, mode. These 3 measures are called as "Central tendency".

* Measuring the Central tendency :-

The most popular method to find the Centre value of a given Set is mean. for eg... Data items $x_1, \ldots, x_n$ then mean $= \frac{1}{n} \sum_{i=1}^{n} x_i$

lity
ts.
it
on
ta
%
%
=78%

This mean is equal to avg in SQL.

$$\text{average} = \frac{\text{sum}}{\text{count}}$$

Sometimes the x_i contains the bytes i.e.,, w_i for $i=1$ to n. Then, weighted arithematic mean.

$$= \frac{\sum_{i=1}^{n} x_i w_i}{\sum_{i=1}^{m} w_i}$$

we can find the median value by using a formula:

$$\boxed{\text{median} = L_1 + \left(\frac{N/2 - (\Sigma f)_l}{f\text{median}}\right) c}$$

where 'L_1' is the lowest class boundary

'c' is the class interval.

$(\Sigma f)_l$ is the cummulative frequency < fmedian

fmedian is the median frequency

'N' is the no of samples.

Eg:- for eg.,, Consider the below table.

class interval 'c'	frequency fmedian	Cummulative frequency $(\Sigma f)_l$
1-3	2	2
4-6	$\boxed{3}$	5
7-9	5	10

0. Mining Association Rules in large Data Bases

Association Rule Mining:-

The association rule Mining Searches the interesting relationships between items in the given data set.

An typical Example of association rule of Market Basket Analysis.

Market Basket Analysis:-

This Market basket analysis analyse buying behaviour of customer by using association rul between the items.

For Eg: consider the three customers, then Ma Basket Analysis is shown below.

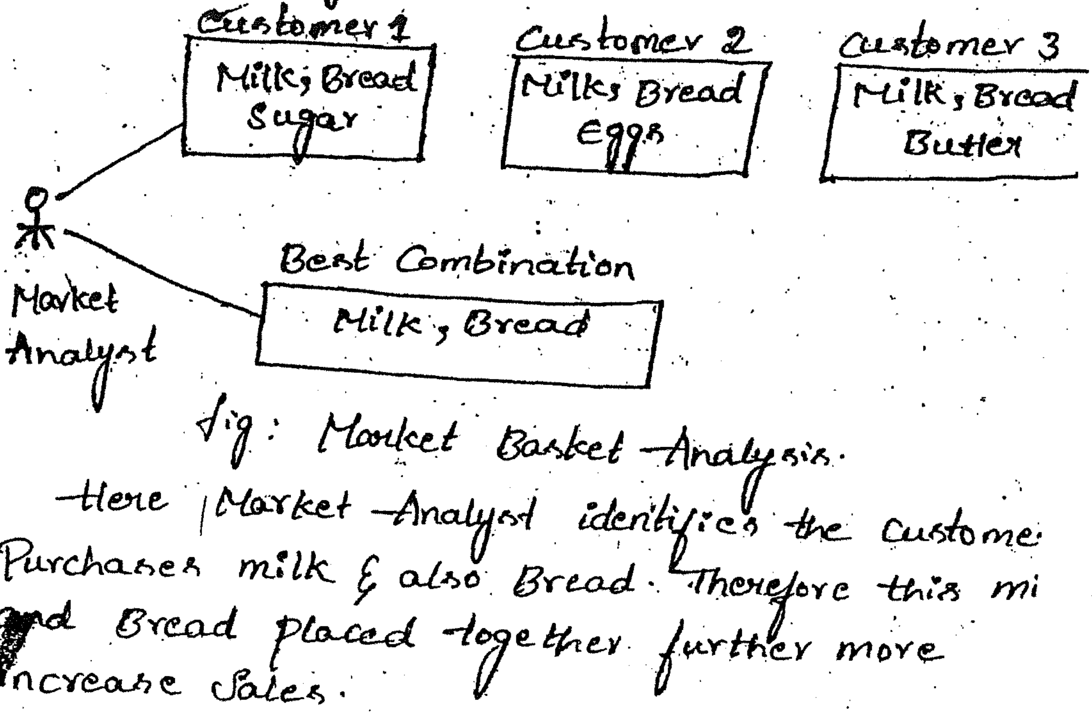


fig: Market Basket Analysis.

Here Market Analyst identifies the custome Purchases milk & also Bread. Therefore this mi and Bread placed together further more increase Sales.

Therefore, as well as using the results Market Basket Analysis the ...

This market Basket Analysis uses the associati rule.

For Eg: Customer purchases a Computer and also Purchases financial Mgmt Software. Then the association rule,

Computer $\Rightarrow$ financial - Management - Software [Support 2% & Confidence = 60%]

Here Confidence 60% means 60% of customers Purchase Computers & also purchases financial Mgmt software & support is 2%.

Basic Concepts:-

Consider set of items, $I = \{i_1, i_2 --- i_m\}$ & task relevant data set 'D'. we get this task relevant data set by using set of transactions. Each transaction is represented by T_i $T \subseteq I$.

As well as each transaction is uniquely identified by using TID [Transaction ID].

For Eg, item $A \subset T$, then $A \Rightarrow B$, $A \subset I$, $B \subset I$ & $A \cap B = \phi$

Then association rule $A \Rightarrow B$ with Support 's', where 's' is the percentage of transactions in 'D' Contains A & B i.e.,

$$D(A \cup B)$$

The association rule $A \Rightarrow B$ has Confidence 'c', where 'c' is the Percentage of transactions in 'D' Contains A & also Contains B i.e.,

$P(B/A)$

$\therefore$ Support $(A \Rightarrow B) = P(A \cup B)$ i.e, Transactio Contains 'A' & 'B'.

Confidence $(A \Rightarrow B) = P(B/A)$ i.e, Transaction Contains 'A' & also contains 'B'.

Here, support & confidence is represented in the form of percentage & ranges in betwe 0% to 100%.

Association Rule Mining : A Road Map:-

The market - Basket analysis i.e, one o the association rule. The association rule classified based on the following criteries.

i, Based on the type of Value used in rule:-

The association rule does not contain any item or attribute or dimension, then that association rule is called as boolean associa rule.

Ex:- Computer $\Rightarrow$ financial - management - Software

If the association rule is described b/w quantitative Values, then that association rule is called as quantitative association rule.

Ex:- age(x, "20 --- 29") $\wedge$ income(x, "30k --- 39K") $\Rightarrow$ buys (x, "VCR")

ii, Based on the type of dimensions used in the r

For Ex:- The association is defined for one dimension then that association rule is called as single dimension association rule.

Ex: buys (X, "computer") => buys (X, "financial-Mgnt-S/w")

iii) Based on the different types of Abstractions in the Rule:-

Here the association rule contains the different types of data abstractions. i.e. using this association rule we abstract data at different levels.

Here the association rule contains the different types of data levels.

Ex:- age (X, "30 --- 39") => buys ("computer")
age (X, "30 ---- 39") => buys ("laptops")

** Mining Single dimensional Boolean Association Rule from Transactional DbS:-

Here we use the 'Apriori' algorithm.

Apriori Algorithm:- Finding frequency Itemsets by using candidate sets.

1) The Apriori Algorithm is used to find frequent Item sets.
2) These are used in Boolean association rule.
3) Apriori means recursive approach, i.e., level-wise-search.

4) First of all we find frequent Itemset. This is denoted by 'L_1'. Then using 'L_1' we find L_2 i.e., frequent 2 - itemset.

5) This is used to find, L_3 and so on; until we can not get the no frequent set. i.e., the Apriori algorithm uses prior knowledge of frequent item set and for each L_K we have to scan entire databas i.e., D.

To simplify the Apriori algorithm, it uses apriori property. Using this Apriori Property, we reduce search length.

For Ex, item set is represented by 'I' & it does not satisfying the minimum support threshold i.e., sup-min, then I is not a frequent itemset, then

$$P(I) < Sup\text{-}min$$

If we add another item 'A' then it results

$I \cup A$,

It is also not a frequent itemset

$$P(P(I)) < Sup\text{-}min$$

This apriori property contains the term steps

1) The Join Step:-

To calculate L_K, a, Candidate k itemset are joined, L_{K-1} to itself. The candiate set is represented by C_K. Consider l_1 & l_2 iten in L_{K-1}. In general $x_1[j]$ means j value of L_1, then we join $L_{K-1} \bowtie L_{K-1}$, it requires the first K-2 items are Equal. Therefore, to join l_1 and l_2 of L_{K-1} and if

$$[(l_1[1]=l_2[1]) \wedge (l_1[2]=l_2[2]) \wedge \dots \wedge (l_1[K-2]=l_2[K-2]]$$

$$\wedge (l_1[K-1] < l_2[K-1])]$$

The last conditions

$l_1[K-1] < l_2[K-1]$ is used to avoid the duplicate values.

2) The <u>Prune</u> Step:-

The prune step, all non-Empty subset of frequent items is also frequent.

<u>Ex</u>:- Consider transactional database of 'all Electronics' This is represented by 'D' & it contains of transactions.

∴ we use the Aprori algorithm to find frequent item & have minimum count is '2'

TID	List of item-IDS
T100	I_1, I_2, I_5
T200	I_2, I_4
T300	I_2, I_3
T400	I_1, I_2, I_4
T500	I_1, I_3
T600	I_2, I_3
T700	I_1, I_3
T800	I_1, I_2, I_3, I_5

Scan D for count of candidates →

C_1

itemset	Support Count
$\{I_1\}$	6
$\{I_2\}$	7
$\{I_3\}$	6
$\{I_4\}$	2
$\{I_5\}$	2

dig : Transactional ... for all ...

Compare candidate sup-count with min sup-count $\longrightarrow$

L_1

Item set	Sup-count
{I1}	6
{I2}	7
{I3}	6
{I4}	2
{I5}	2

$C_2 = L_1 \bowtie L_2$

Generate C_2 candidate using L_1 $\longrightarrow$

Itemset	Sup-count
{I1, I2}	4
{I1, I3}	4
{I1, I4}	1
{I1, I5}	2
{I2, I3}	4
{I2, I4}	2
{I2, I5}	2
{I3, I4}	0
{I3, I5}	1
{I4, I5}	0

Compare candidate sup-count with min sup-count $\longrightarrow$

L_2

Itemset	Sup-Co
{I1, I2}	4
{I1, I3}	4
{I1, I5}	2
{I2, I3}	4
{I2, I4}	2
{I2, I5}	2

$C_3 = L_2 \bowtie L_2$

Generate C_3 candidate using L_2

Itemset	Sup-count
{I1, I2, I3}	2
{I1, I2, I5}	2

Compare candidate c.c sup-count with min sup-count $\longrightarrow$

L_3

itemset	S c
{I1, I2, I3}	
{I1, I2, I5}	2

Using Apriori algorithm, we find candidate ... & most frequently item set with min Support Count

Steps for finding candidate item set & frequent itemset

1) This algorithm scans all the transcations of database to find total count of each item set. This itemset is called as candidate 1-itemset, C_1.

2) Suppose consider minimum support threshold is $2\ (2/9 = 0.22 = 22\%)$

3) Candidate support count is compared with minimum support count. Then we get frequent itemset L_1.

4) To find C_2, the L_1 joined itself i.e,

Join : $C_2 : L_1 \bowtie L_1$

Then we find L_2 by comparing candidate support count with minimum support count.

5) Join $C_3 : L_2 \bowtie L_2$

$$\{\{I_1, I_2\}, \{I_1, I_3\}, \{I_1, I_5\}, \{I_2, I_3\}, \{I_2, I_4\}, \{I_2, I_5\}\} \bowtie \{\{I_1, I_2\}, \{I_1, I_3\}, \{I_1, I_5\}, \{I_2, I_3\}, \{I_2, I_4\}, \{I_2, I_5\}\}.$$

$$= \{\{I_1, I_2, I_3\}, \{I_1, I_2, I_5\}, \{I_1, I_3, I_5\}, \{I_2, I_3, I_4\}, \{I_2, I_3, I_5\}, \{I_2, I_4, I_5\}\}.$$

Prune Step:-

All non-Empty frequent Subsets are also frequent.

1. 2-Item sets for $\{I_1, I_2, I_3\}$ are $\{I_1, I_2\}$, $\{I_1, I_3\}$ and $\{I_2, I_3\}$ Then $\{I_1, I_2, I_3\}$ Add to C_3 because all 2-item sets present in L_2.
2. 2-Item sets for $\{I_1, I_2, I_5\}$ are $\{I_1, I_2\}$, $\{I_2, I_5\}$ and $\{I_1, I_5\}$ there all 2-item set Present in L_2 Then $\{I_1, I_2, I_5\}$ add to C_3.
3. 2-Item sets for $\{I_1, I_3, I_5\}$ are $\{I_1, I_3\}$, $\{I_3, I_5\}$ and $\{I_1, I_5\}$ Then $\{I_3, I_5\}$ is n in L_2 then $\{I_1, I_3, I_5\}$ is Removed from C_3
4. 2-Item sets for $\{I_2, I_3, I_4\}$ are $\{I_2, I_3\}$, $\{I_3,$ and $\{I_2, I_4\}$ then $\{I_3, I_4\}$ is not in L_2. Therefore, it is not a frequent set. So $\{I_2, I_3$ is removed from C_3.
5. 2-Item sets for $\{I_2, I_3, I_5\}$ are $\{I_2, I_3\}$ $\{I_3, I_5\}$ and $\{I_2, I_5\}$ Then $\{I_2, I_3, I_5$ is removed from C_3.
6. 2-Itemsets for $\{I_2, I_4, I_5\}$ are $\{I_2, I_4\}$, $\{I_4, I_5\}$ and $\{I_2, I_5\}$ then $\{I_4, I_5\}$ is in L_2. Therefore it is not frequent set. Then $\{I_2, I_4, I_5\}$ is removed from C_3.

7. Then $C_3 = \{\{I_1, I_2, I_3\}, \{I_1, I_2, I_5\}\}$

8. Again we scan all the transcations to find the count value for 3-itemsets. Then we get L_3.

Item set	sup-count
$\{I_1, I_2, I_3\}$	2
$\{I_1, I_2, I_5\}$	2

Apriori Algorithm:-

Find frequent set using recursive level-wise approach based on candidate item-set.

Input: Database D, op-transcation; minimum support threshold, min-sup

Output: L, frequent itemset.

Method: L_1 = first-frequent-1-itemset;

for ($K=2$; $L_{K-1} \neq \phi$; $K{+}{+}$)

{

C_K = apriori-gen(L_{K-1}, min-sup);

for each transaction $t \in D$

{

C_t = subset(count

for each $c \in C_K$

c.count++;

}

$L_K = \{c \in C_K \,/\, c.count \geq \text{min-sup}\}$

}
return L;
}

Procedure apriori-gen (L_{K-1}: frequent k-1 itemset
min-sup: mim. support threshold)
{
Consider $l_1, l_2, l_1 \in L_{K-1}, l_2 \in L_{K-1}$
if $l_1[1] = l_2[1] \wedge \ldots \wedge (l_1[K-2] = l_2[K-2])$
$(l_1[K-1] < l_2[K-1])$ then $c = l_1 \bowtie l_2$;
if has-infrequent-Subset (c_K, L_{K-1}) the
delete c;
else
add c to Ck;
return Ck;
}

Procedure has-infrequent-subset (c: candidate
item set, L_{K-1}: frequent k-itemse
{
if $c \notin L_{K-1}$ then
return TRUE
else
return FALSE
}

Alg:-

Apriori Algorithm for finding frequent iter
i.e, used in boolean association rule.

Generating Association rules from frequent ...

Using Apriori alg, we find frequent item sets then we derive strong association rules

The strong association rule means it must include Support Count & Confidence Count. Then we use the following Equation for the Confidence to derive strong association rules where Conditional Probability is represented in the form of the Support Count.

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

where Support-Count $(A \cup B)$ represents transactions Containing A & B.

Support-Count(A) represents transaction Containing 'A'.

Using this we derive 2 associations rules for Each frequent item sets l, must Contain all non-Empty Subsets of 'l'.

For Each Subset 's' of 'l', the o/p rule

"$s \Rightarrow (l-s)$" if $\dfrac{\text{support-count}(l)}{\text{support-Count}(s)} \geq$ min-Confidence

For Ex, frequent itemset,

$l = \{I_1, I_2, I_5\}$ then Subsets are

$\{I_1, I_2\}$ $\{I_1, I_5\}$ $\{I_2, I_5\}$ $\{I_1\}$ $\{I_2\}$ & $\{I_5\}$

Then the association rules are

$I_1 \cap I_2 \Rightarrow I_5$, Confidence $= 2/4 = 50\%$. —— ①

$I_1 \cap I_5 \Rightarrow I_2$, Confidence $= 2/2 = 100\%$. —— ②

$I_2 \cap I_5 \Rightarrow I_1$, Confidence $= 2/2 = 100\%$. —— ③

$I_1 \Rightarrow I_2 \cap I_5$, Confidence $= 2/6 = 33\%$. —— ④

$I_2 \Rightarrow I_1 \cap I_5$, Confidence $= 2/7 = 28\%$. —— ⑤

$I_3 \Rightarrow I_1 \cap I_2$, Confidence $= 2/2 = 100\%$. —— ⑥

Consider min confidence is 70% , then the o/p is ②, ③, ⑥ only [we get] strong associati

Improve the Efficiency of Apriori:-

we use following techniques to improve the Efficiency of Apriori alg.

1. Hash based Technique:-

using this, we reduce the size of candidate -k itemset. we create both hash table for 'H by using the hash function.

$$\boxed{H(x,y) = ((\text{order of } x) \times 10 + (\text{order of } y)) \bmod 7}$$

we consider transaction db 6-1 then

Hash table H_2 for Candidate-2 itemset

Hash address	0	1	2	3	4	5	6
Hash count	2	2	4	2	2	4	4
Hash Contents	$\{I_4, I_4\}$ $\{I_2, I_5\}$	$\{I_1, I_5\}$ $\{I_1, I_5\}$	$\{I_2, I_3\}$ $\{I_2, I_3\}$ $\{I_2, I_3\}$ $\{I_2, I_3\}$	$\{I_2, I_4\}$ $\{I_2, I_4\}$	$\{I_2, I_5\}$ $\{I_2, I_5\}$	$\{I_1, I_2\}$ $\{I_1, I_2\}$ $\{I_1, I_2\}$ $\{I_1, I_2\}$	{I {I {I_1 {I_1

$\{I_1, I_2, I_5\}$

$h(I_1, I_2) = 10+2 \% 7 = 5$

$h(I_1, I_5) = 10+5 \% 7 = 1$

$h(I_2, I_5) = 20+5 \% 7 = 4$

$h(I_2, I_4) = 20+4 \% 7 = 3$

$h(I_2, I_3) = 20+3 \% 7 = 2$

$h(I_1, I_4) = 10+4 \% 7 = 0$

$\{I_1, I_2, I_4\} \Rightarrow h(I_1, I_4) = 10+4 \% 7 = 0$

$h(I_1 I_3) = 10+3 \% 7 = 6$

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

$h(I_3 I_5) = 30+5 \% 7 = 0$

Consider min-support Count is '3'. Then the addresses 0, 1, 3, 4 does not be considered for C_2.

2. Transaction Reduction:-

- If the transaction does not contain frequent k-itemset, then it also does not contain the frequent k+1 itemset then those transactions must be removed.

3. Partitioning:-

In this, it requires only 2 database scans & also it consists 2 phases.

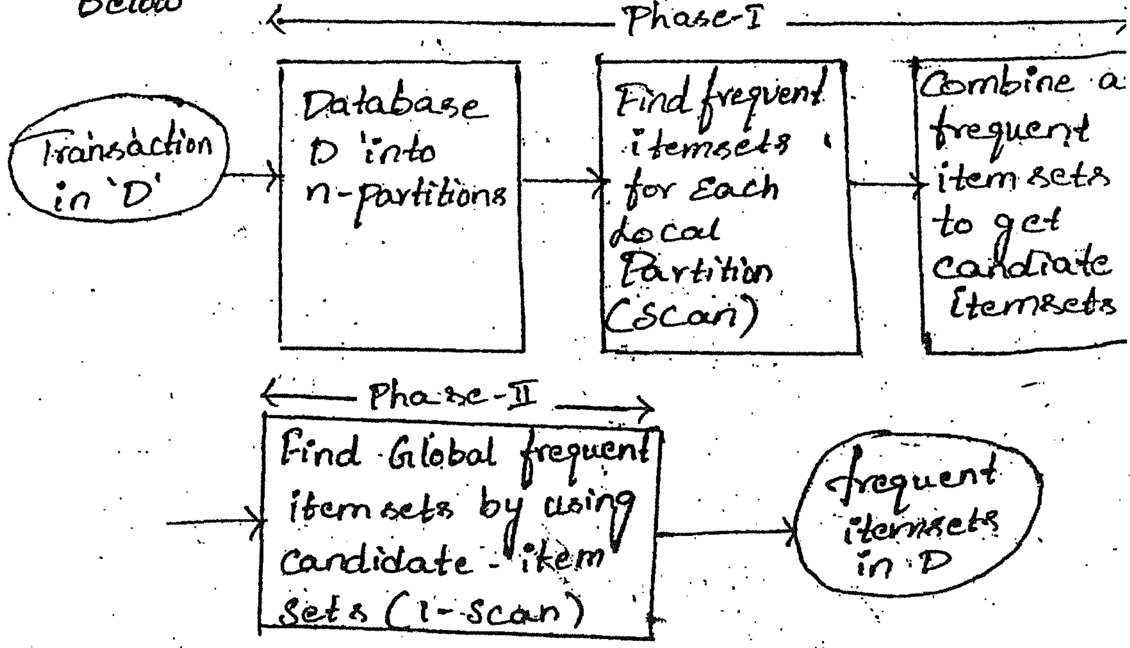
Phase-I: It Contains following steps.

1. Divide db 'D' into 'n' Partitions.
2. Find frequent itemset for Each local Partition

3. Combine these ... Candidate item sets.

Phase-II

1. Find global frequent item set. This is shown below.



Partition data Method.

4. Sampling:-

Consider the random sample 's' in database. Then we find frequent itemsets in 's' rather than 'D'.

The size of 's' in such a way that it must be fit in primary memory i.e., in single scan all the transactions of 's' avoided.

Here we are finding frequent itemset in 's'. Therefore, we may miss some frequent itemsets. Therefore we consider lower support threshold rather than min support threshold.

Generate Frequent itemsets without using Candidate itemsets (or) FP growth Method

In Apriori algorithm, we find Candidate itemsets. Using this Candidate itemsets, we find frequent itemsets. But the Apriori Algorithm Contains the following drawbacks.

i) It generates large No. of Candidate itemsets For Ex, 10^4 frequent-itemset. Then apriori alg generates approximately 10^7 Candidate-2 itemsets.

ii) It requires the large No. of database Scans to identify the Value of candidate itemsets.

To avoid these drawbacks, we go for the frequent Pattern-growth Method.

This is simply Called as FP-growth method. It follows the Divide-and-Conquer technique. It contains three steps.

1. Compress the db representing itemsets into frequent pattern-tree or FP-tree following the itemset Information.
2. Transform Compressed db into Conditional DB.
3. Generate frequent Patterns.

Consider the transaction db fig:

TID	List of item-IDs
T100	I_1, I_2, I_5
T200	I_2, I_4
T300	I_2, I_3
T400	I_1, I_2, I_4
T500	I_1, I_3
T600	I_2, I_3
T700	I_1, I_3
T800	I_1, I_2, I_3, I_5
T900	I_1, I_2, I_3

After the first scan of db, we get L_1, it contains the item set & support-count & we represent this in descending order of support count & it is represented by 'L'

$$L = [I_2:7, I_1:6, I_3:6, I_4:2, I_5:2]$$

Consider support-count is '2', Then <u>FP-tree Construction</u> is

after the second scan of db, the first transaction is

"T100 : I_1, I_2, I_5".

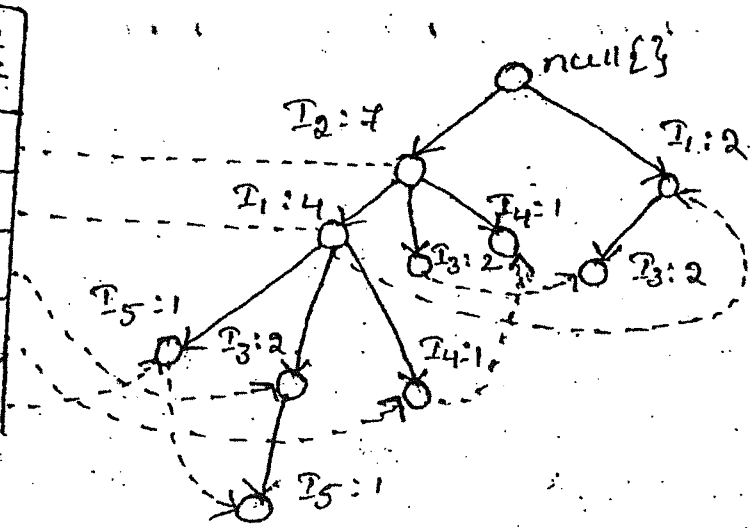
& these items are arranged based on the order of 'L'. Then we get (I_2, I_1, I_5) & o these items are occured in first time, the ($I_2:1$) is treated as the root & ($I_1:1$) is attached to I_2. & ($I_5:1$) is attached to I_1 & then second transaction.

$T_{200}: I_2, I_4$

then, I_2 is already root, so increment the count value of 'I_2' & $(I_4:1)$ is attached to I_2. This is shown below.

FP-Tree Construction start with the root, this root is labelled as null {}

Item Id	Support Count	Node Link
I_2	7	
I_1	6	
I_3	6	
I_4	2	
I_5	2	



FP-tree based on Transaction Database.

<u>Construction</u> of <u>Conditional db</u> by <u>Mining</u> FP-Tree:-

Start with the item 'I_5' it contains the 2 branches. These branches are identified by verifying Node link. Then the paths are

$I_2\ I_1\ I_5 : 1 \qquad I_2\ I_1\ I_3\ I_5 : 1$

Then I_5 is considered as suffix. Then we

1. Conditional patterns how i.e, as link paths of I_5 are $(I_2\ I_1 : 1)$ and $(I_2 I_1 I_3 : 1)$

Then Conditional FP-tree contains only one path, then we merge above 2 paths & we get

<I2 - x3:1> ...

we cannot include I_3 because its support count is '1' & it is less than the support count. Then this single path generates the frequent itemsets by concatinating suffix.

∴ Frequent patterns are

$I_2 I_5 : 2$ $\quad$ $I_1 I_5 : 2$

This is shown below.

Item	Conditional pattern Base	Conditional FP-tree	Frequent Pattern Generation
I_5	$\{\langle I_2 I_1 : 1\rangle \langle I_2 I_1 I_3 : 1\rangle\}$	$\langle I_2 : 2, I_1 : 2\rangle$	$I_2 I_5 : 2$ $I_1 I_5 : 2$ $I_2 I_1 I_5 : 2$
I_4	$\{\langle I_2 I_1 : 1\rangle, \langle I_2 : 1\rangle\}$	$\langle I_2 : 2\rangle$	$I_2 I_4 : 2$
I_3	$\{\langle I_2, I_1 : 2\rangle, \langle I_2 : 2\rangle \langle I_1 : 2\rangle\}$	$\langle I_2 : 4, I_1 : 2\rangle, \langle I_1 : 2\rangle$	$I_2 I_3 : 4, I_1 I_3 : 2, I_2 I_1 I_3 : 2$
I_1	$\{\langle I_2 : 4\rangle\}$	$\langle I_2 : 4\rangle$	$I_2 I_1 : 4$

Conditional db by Mining FP-tree

Finally, using this FP-growth method large Pattern is divided into smaller pattern by using suffix & then suffix is concatenated to get the frequency patterns.

This alg. reduces the search cost & also this is more faster than the apriori algorithm.

ICEBERG QUERIES:-

The Apriori alg is used to improve the Efficiency of ICE-BERG Query. This ice-berg query is mainly used in DM preferably Market

The ICE-BERG Query Computes aggregation function over an attribute, to find out which aggregate values over the specified threshold.

Consider the relation 'R' with attributes a-1, a-2, ---- a-n, & b and aggregate function agg-f. Then the ice-berg query looks like the following.

Select R.a-1, R.a-2, ----R.a-n, agg-f(R.b)
from relation R.
group by R.a-1, R.a-2 --- R.a-n
having agg-f(b) >= threshold.

The above ICE-BERG query accepts the large no of i/p tuples, but it gives very small No. of o/p tuples, because the o/p tuples only displays that satisfy the threshold values.

MINING MULTILEVEL ASSOCIATION RULES FROM TRANSACTION DATABASES:-

Mining multilevel association rules means rules are included at different levels of abstraction.

Multilevel Association Rules:-

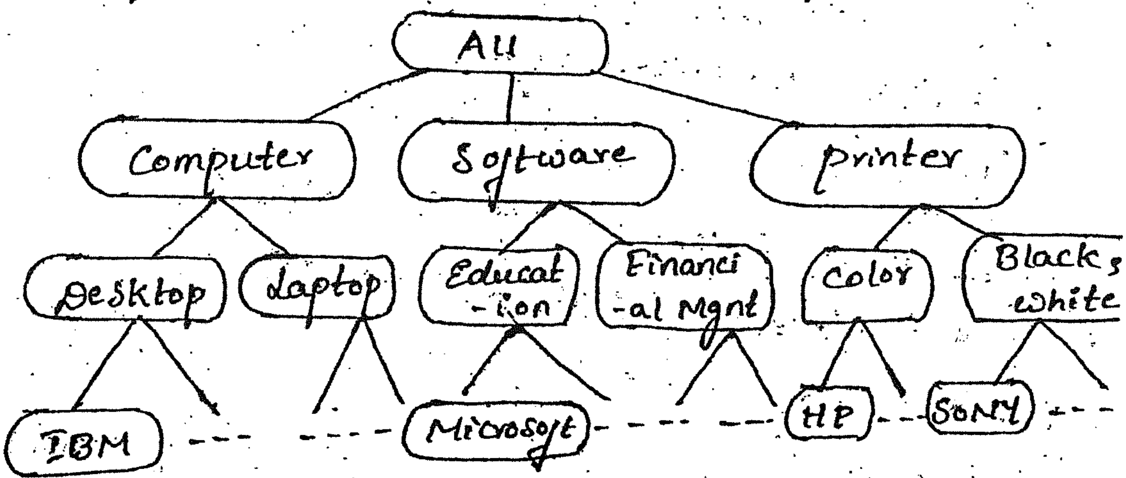
It is difficult to define strong association rules at low level or primitive level.

Consider the task-relevant data set as specified below.

TID	Items Purchased
T1	IBM, Desktop Computer, Sony, B/W printer,
T2	Microsoft Education S/w, HP Color printe
T3	Microsoft financial Mgmt S/w,
T4	IBM laptop Computer
⋮	⋮

Task Relevant Data set

The above Task relevant data set is represented in Concept hierarchy as.



The concept hierarchy defines set of mapping from low level to high level.

Here it contains 4 levels. i.e, levels 0, 1, 2 {

The top level is level '0' i.e., represented by using keyword 'all'.

The level '1' contains Computer, S/w & printer.

The level 2 contains Desktop Computer, laptop Computer & so on.

The level 3 contains IBM Desktop Computer & so on.

The lowest level is level 0 & in this level ... difficult to identify interesting patterns i.e.,

Consider "IBM Desktop Computer" & "Sony B&W Printer" occur in very few.

∴ Define Generalization like "IBM Desktop Computers" to "Computer" & "Sony B&W printer" to "Printer".

∴ The Combination {Computer, Printer}, as many People frequently purchased.

∴ using this multilevel Concept hierarchies, we Easily define the interesting patterns.

Approaches to Mining Association Rules:-

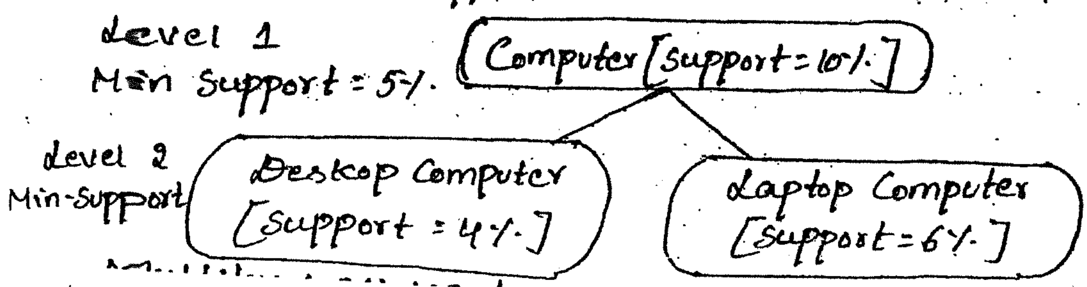
Generally we follow the top down approach, i.e, with level 1 & find frequent item. There are no frequent item sets. Then we go for level 2 & so on.

we use the following approaches for mining association rules.

Uniform Support:-

Here we define Same minimum support Threshold for all the levels.

For Ex, Min. Support Threshold is 5%, then



Multilevel ... uniform Support.

Therefore's compu

frequent patterns & desktop Computer or not.

This approach Contains the advantages of I. Searching process is Simplified i.e, here we Verify only those items satisfy Min-support II. Here Enduser has to specify only one Min-support Threshold value.

This technique Contains the drawbacks of

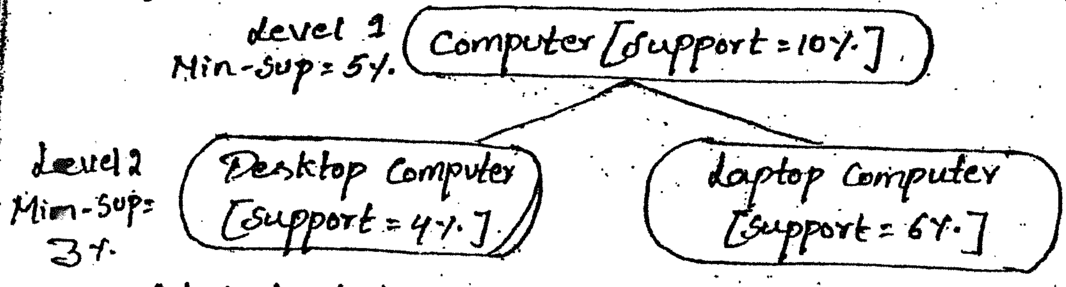
i) we already see the low level item is not frequen as upper level item. Therefore, we Cannot define Min-Support Threshold.

ii) if we specify very large support threshold the the low level important patterns may be missed

iii) If we define Very small support threshold then it Contains the large No-of unimportant patterns

ii) <u>Reduced Support</u>:-

Here Each level has its own minimum support threshold. Generally higher levels contains the big values than the low levels. Consider Min-support threshold for level 1 & 2 are 5%. & 3%. This is shown in below.



Multilevel Mining with reduced Support.

Here computer, ...

are frequent patterns.

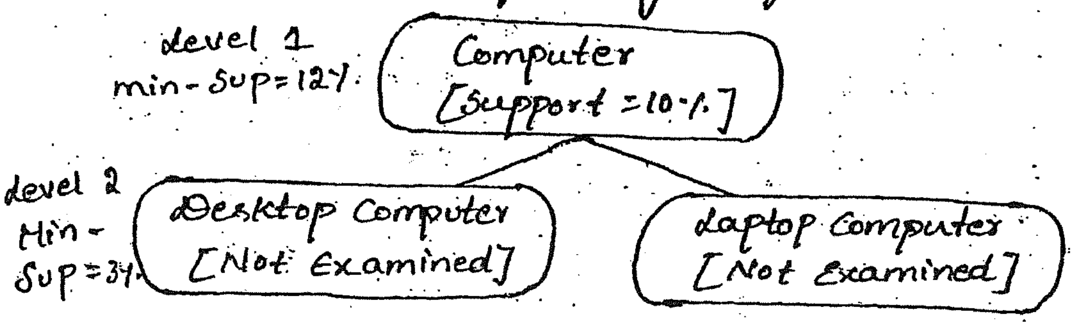
iii) Level-by-Level Independent:-

Here Each node is Examined independently i.e., regardless of its parent is frequent or not.

iv) Level Cross Filtering by Single item:-

Here 'item with 'ith' level is Examined if & only iff its parent $(i-1)^{th}$ level is frequent.

Consider the following Diagram.



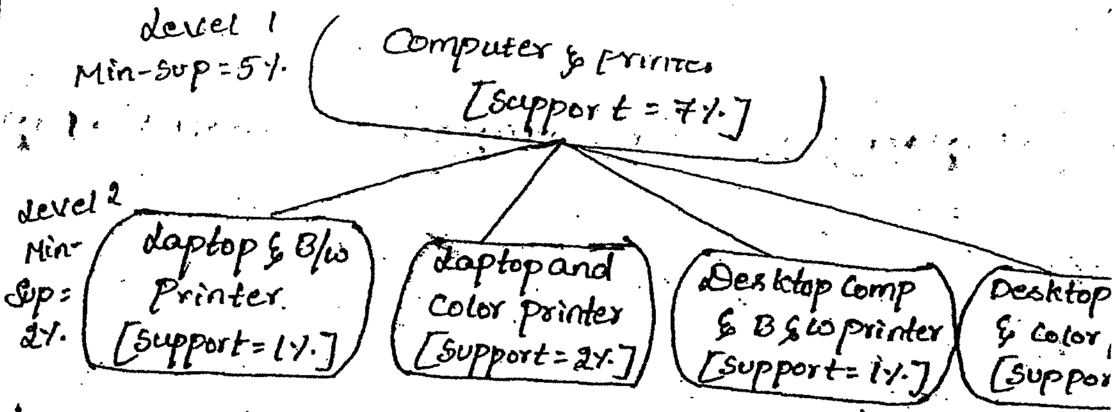
Multilevel Mining with reduced support using level cross filtering by single item.

Here Desktop Computer & Laptop Computer is not Examined because its parent 'Computer' is not a frequent.

v) Level Cross Filtering by k-item Set:-

Here k-item Set is Examined [k-itemset with i^{th} level] if & only iff Parent k-itemset of $(i-1)^{th}$ level is frequent.

Consider the below diagram.



Multilevel Mining with Reduced support using level-cross filtering by k-item set, here k=2

Here {Laptop Computer, B/w printer}
{Laptop Computer, Color "}
{ " " , B&w " }
{ " " , Color " } are Exami

because 'Computer' & 'Printer' are frequent.

Checking redundant Multilevel Association Ru

Using concept hierarchy we define association rules at different levels of abstraction. This is Called as Multilevel association rule. when we are deriving multilevel association rules we ma Contain redundant rules. when we are deriving multilevel association rules we may contain redundant rules by "ancestor" property Exist different items.

For Ex Consider the concept hierarchy fig 6.4 Contains the ancestor property of 'Desktop Compu is the ancestor of 'IBM'.

For Ex, Consider the following association rules.

Desktop Computer ⇒ B&W printer —— ①

IBM " " ⇒ B&W printer —— ②

Here rule ② does'nt give any ad

Let 'R_1' is the ancestor of 'R_2' then 'R_1' replaces 'R_2'.

Mining MultiDimensional Association rules from relational DB & DWH (or) Pattern Evolution Method:-

- Here we mine association rules more than one dimension or predicates.

Multi Dimensional Association Rules:-

For Ex, Consider the 'ALL ELECTRONICS' DB. The Boolean Association rule is i.e., it doesnt Contain any dimension.

IBM Desktop Computer ⇒ Sony B&W printer.

Using this we define single dimension association rule as

buys (x, "IBM Desktop Computer") ⇒
buys (x, "Sony B&W Printer).

- Here we are using one predicate "buys" we can also define multidimensional Association rule as

age (x, "20 --- 29") ∧ occupation (x, "Student")
⇒ buys (x, "Laptop").

Here, it contains o ...

occupation & buys.

Here No predication is Repeated. Then it is called as Inter Dimensional Association Rule.

If a predicate is repeated, then called as hybrid Dimensional Association. This is shown below:

age(x, "20---29") ∧ buys (x, "laptop") ⇒ buys(x, "B&w

Generally, attributes in database quantitative.

Here 'categorical' means it contains finite No. values & this values does contain any ordering

Quantitative means it contains any numerica values & this values are ordered.

Mining Multidimensional Association using Stati

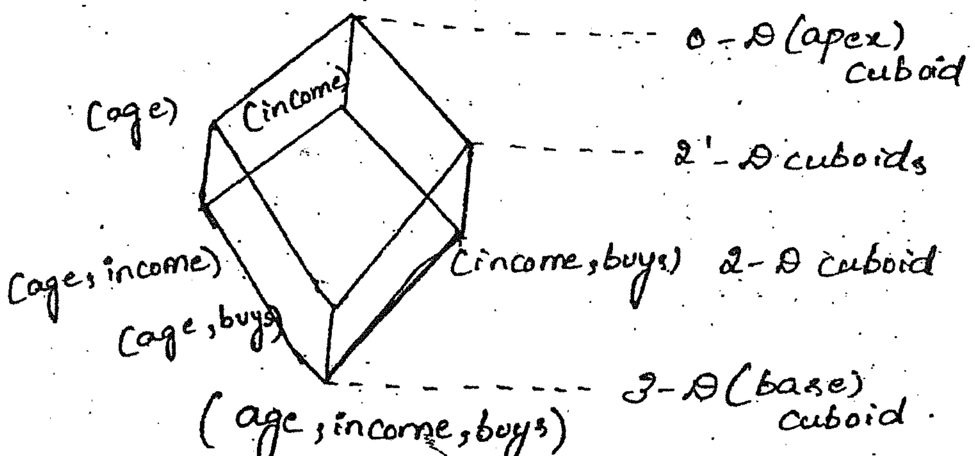
Discretization of Quantitativeutes:-

Discretization is performed before the D. Consider the concept hierarchy for in is replaced with numerical range of num values such as "20k --- 40k", "41k --- so on.

Then the discretization is static & pre-defined. This range of values consider as categories. We find the values to fall in Each Category. Then we find frequent item set. using this freq item set, we derive association rules.

the task relevant data cube. Data cube may Contains the multidimensions & also using the datacube, we Easily derive multidimensional Association rules.

The data cube Consists of lattice of cuboids. These are multidimensional data Structures & these Contains task relevant data & also Contains aggregated & grouping information. This is Shown below.



Lattice of cuboids make 3-D Data cube

Here the base cuboid aggregates task relevant data by age, income, buys.

2-D cuboid i.e, (age, income), it aggregates task relevant data age & income.

1-D cuboid Specifies task relevant data of 1 Dimension.

The apex cuboid specifies total No. of transcations in task relevant data.

Mining Quantitative Association Rules

Quantitative association rules are multi dimensional association rules but they contain numerical attributes. These are derived from discretization.

In Quantitative association rules, the left side it contains 2 Quantitative attributes & in right side it contains Categorical attributes. These is shown below:

$$A_{quan1} \wedge A_{quan2} \Rightarrow A_{cat}$$

Here, A_{quan1}, A_{quan2} are quantitative Numerical range values.

'A_{cat}' is the categorical attribute task relevant data.

For Ex, 2-D Quantitative association rule is

age(X, "30---39") ∧ income(X, "29K---30K) ⇒ buys (X, "TV").

These quantitative association rules derived from ARCS (Association Rules Clustering System)

This ARCS approach contains the following steps.

1) Binning:- The Quantitave attributes generally contain wide range of values. To smooth the values, we use following binning methods.

a) Equiwidth:- Here it contains interval size each bin is equal.

c) Equidepth: here it contains equal no. of
values in each bin.

2). Find Frequent Set:-

To find frequent set, we construct 2-dimensional array containing count values for each category. These values are scanned to get frequent set. These frequent set also satisfy min. support & min. confidence.

3) Clustering association rules:-

For Ex, consider the association rules below.

age(x, 34) ∧ income(x, "31K --- 40K") => buys(x, "TV") — 1

age(x, 35) ∧ income(x, "31K --- 40K") => buys(x, "TV") — 2

age(x, 34) ∧ income(x, "41K --- 50K") => buys(x, "TV") — 3

age(x, 35) ∧ income(x, "41K --- 50K") => buys(x, "TV") — 4

These rules are represented in 2D Grid as below

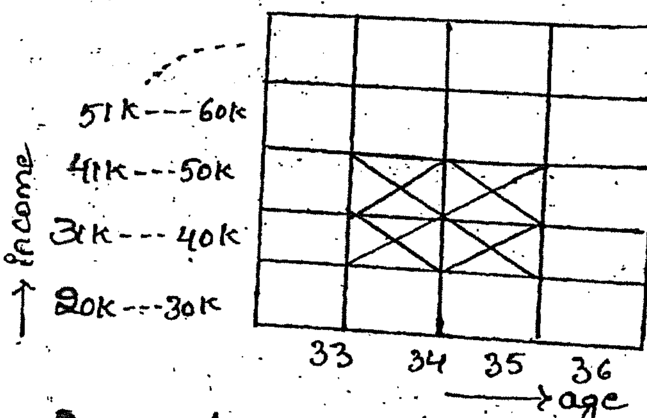


fig:- grid represents customer who buys "TV"

The above 4 association rules are simply represented in:

age(x, "34---35") ∧ income(x, "30K---50K") => buys(x, "TV").

Mining Distance Based Association

For Ex, the data for the 'price' partitioned b using Equiwidth & Equidepth & this is Compared with distance-based partition & this is shown below.

Prices($)	Equiwidth (width $10)	Equidepth (deph=2)	Distance-based
7	[0,10]	[7, 20]	[7, 7]
20	[11, 20]	[22, 50]	[20, 22]
22	[21, 30]	[51, 53]	[50, 53]
50	[31, 40]		
51	[41, 50]		
53	[51, 60]		

fig: Data is partitioned by using Equiwidth & Equidepth.

Here, we Examined the above table. The distance based partition is the Efficient c bcoz it groups the values in such a way the are closer & also closer to specified interval.

Ex. [20, 22]

If u Consider, Equidepth partition, it group 2 distinct values i.e., [22, 50]

If U Consider, Equiwidth partition, it conta Equal Size of intervals but Some of these inter does'nt contain any values. i.e, [31, 40]

∴ Distance based partition is the best one be this distance-based association contains th Drawback. i.e., attribute.

For Ex, consider the association rule,

item-type (x, "electronic") ∧ manufacturer (x, "Apple") ⟹ price (x, $200)

Here, 'x' is the item, In reality, we always prefer 'Apple' Electronic items are approximately '$200' rather than Exactly '$200'

∴ To define approximate values of an attribute & also derive distance based association rules, we use 2 phase algorithm.

Phase-I :-

Here we define interval for Each cluster.

Phase-II

Each cluster is verified to identify frequent set.

Using this frequent set, we derive association rules.

From Association Mining to Correlation-Analysis :-

Even the strong association rules also uninteresting & gives the wrong information. Therefore we require additional measures that contains statistical analysis.

∴ we go for the Correlation-Analysis.

Strong-Association Rules are not interesting :-

Consider the "All Electronics" database. we analyze 10,000 transaction. In those 10,000 transactions , 6000 ...

Purchase the computer games , 7,500 are videos and 4000 transactions contains the both.

Consider support count is 30% & confidence is 60%.

buys(x, "Computer games") => buys(x, "videos") — [support = 40% , confidence = 66%]

$$\text{Support}(A \Rightarrow B) = \frac{\#\text{-tuples-containing-}A\text{-}\&\text{-}B}{\text{Total-}\#\text{-of-tuples}}$$

$$\text{Confidence}(A \Rightarrow B) = \frac{\#\text{-tuples-containing-}A\text{-}\&\text{-}B}{\#\text{-of-tuples-containing}}$$

$$= \frac{4000}{6000} = 2/3 = 0.66 = 66\%.$$

∴ Rule 1 is strong association rule, but the probability of purchasing videos is

$$\frac{7500}{10,000} = 0.75 = 75\%.$$

∴ It is greater than 66%.

So, computer games & videos are inversely associated. i.e, customer purchase the computer games that decreases the videos.

From Association Analysis to Correlation Analysis

The above association rule A => B is uninteresting & it gives the wrong information. So we for the Correlation Analysis.

Correlation Analysis [illegible] is
represented as

$$Corr_{A,B} = \frac{P(A \cup B)}{P(A)P(B)} \quad \text{--- ②}$$

If the result is >1, then 'A' & 'B' are +ve Correlated. i.e., if we increase 'A', 'B' also increases.

If the result is less than '1', then 'A' & 'B' are -vely Correlated i.e., if we increase 'A' that decreases 'B'.

If the result is Equal to '1' then 'A' & 'B' are independent. i.e., in between 'A' & 'B' there is no Correlation.

Consider the above Ex, then

Computer games = 6000

videos = 7500

Computer games & videos = 4000

The probability of purchasing 'computer games' is $P(\{games\}) = \frac{6000}{10000} = 0.60$

This is probability of purchasing 'videos' is $P(\{videos\}) = \frac{7500}{10000} = 0.75$

The probability of purchasing both 'Computer games' and 'videos' is

$$P(\{games, videos\}) = \frac{4000}{10000} = 0.40$$

$\therefore$ Then Correlation between Computer games &

Videos is

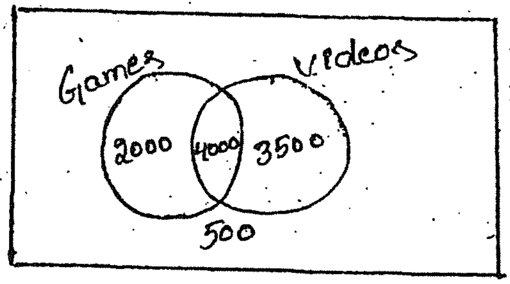
$$\text{Corr}_{\text{games, videos}} = \frac{P(\{\text{games, videos}\})}{P(\{\text{games}\}) \times P(\{\text{videos}\})}$$

$$= \frac{0.40}{0.60 \times 0.75}$$

$$= 0.89$$

∴ Therefore the computer games & videos are very correlated.

The transactions in DB are summarized by Contingency Table. This is shown below.



	Games	$\overline{\text{Games}}$	Σ_{row}
Videos	4000	3500	7500
$\overline{\text{Videos}}$	2000	500	2500
Σ_{col}	6000	4000	10000

fig: 2x2 Contingency Table, Summarizing Transcation Purchased

Here '$\overline{\text{games}}$' represents dont contain computer games

$\overline{\text{videos}}$ represents dont contain computer vid

Constraint Based Association Mining:-

Many of the DM association rules are uncovered by DM System & also discovered rules are unimportant to the End user. Therefore we go for the Constraint based association Mining.

In Constraint based association Mining. we mine association rules based on Constraints these Constraints are Specified by End user.

The Constraints are.

i) Knowledge Constraints:-

These Constraints Specifies the kind of knowledge to be mined.

ii) Data Constraints:-

The Constraints Specifies the task relevant data sets.

iii) Level Constraints:-

These Constraints Specifies the total No. of levels in concept hierarchy.

iv) Threshold Constraints:-

These Constraints Specifies the Support & Confidence Values.

v) Rule Constraints:-

These constraints Specifies what type of association rules are to be mined & also Specifies the no. of predicates in Each rule.

Mata Rule for Mining Association rules:-

Many of the rules are uninteresting & the wrong information. But in Meta rules allows the user syntactic representation of association rules.

Sometimes these meta rules may contain the Constraints to improve the Efficiency of DM System.

The meta rule looks like the following.

$P_1(X, Y) \wedge P_2(X, Z) \Rightarrow$ buys$(X,$ "Educational S/w"$)$ —

Here, P_1 & P_2 are predicates

$X \longrightarrow$ Customer

$Y, Z \longrightarrow$ Numeric Values related to Predicates P_1 & P_2.

Whenever we issue this meta rule, the DM system searches for the rule that is similar to specified meta rule. Then the DM system ge the similarity & return the following association rule.

age$(X,$ "30 - - 39"$) \wedge$ income$(X,$ "30k - - - 39k"$) \Rightarrow$ buys$(X,$ "Educational S/w"$)$ —— 2

This meta rule is mainly used for guiding the DM process.

For Ex we want to mine inter-dimensional association rules. Then the meta rule is Specified as

$P_1 \wedge P_2 \wedge --- \wedge P_n \Rightarrow Q_1 \wedge Q_2 \wedge --- \wedge Q_n$

Here in this meta rule, we are using total Predicates $P = l+r$

& also P_i where $i = 1 --- l$ &

Q_j where $j = 1 --- r$

Additional Constraint Rules:-

This additional Constraint rules Contains set of relations. These relations are specified by the user by using aggregate functions. These rules are mainly used in multidimensional association rule Mining.

Consider the "All Electronics" multidimensional Sales db. This Contains the following relations

Sales(cust-Name, item-Name, trans-id)

lives(cust-Name, city, Street)

item(item-Name, price)

transaction(trans-id, day, month, year)

Here lives, item & transaction tables are dimensional tables & fact table is Sales. Here we specify the relation by using 3.

i.e., cust-Name, item-Name, trans-id.

Ex. we want to mine association query i.e, find the Sales of cheap items (sum of price < $100) & also find Sales of Expensive items (min of price $500) for Vancouver

Customer in 1999.

This query is specified in DMQL as

mine association as lives (c, "Vancouver")

from Sales

where S. year = 1999 and T. year = 1999 group

by C.

having sum (I. price) and min (I. price) will

Support threshold = 1%.

with Confidence threshold = 50%.

IV unit

7. Classification & Prediction

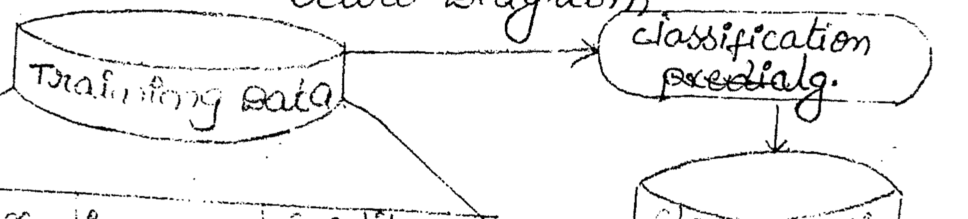
7.1 What is classification & what is prediction:

The data " contains 2 steps.

i) **construct the model by using predefined dataset**: This model uses the tuples in the db & also each tuple must fall in class. This is determined by using attribute. This attribute is called as class label attribute. Then the training data is collected & analyzed by using classification algorithm i.e., learned knowledge is represented into classification rule.

ii) **classification:**

In this, classification, first verify the accuracy of the " rule, its accuracy is ok. Then it will be applied for new data. This is shown in below diagram.



Name	Age	Income	Credit-rating
James	>30	low	fair
Lee	31...40	high	excellent
Bill	>40	Med	fair
Corn	>40	high	excellent

if age = "31... 40"
and income = high
then
credit-rating = excellent

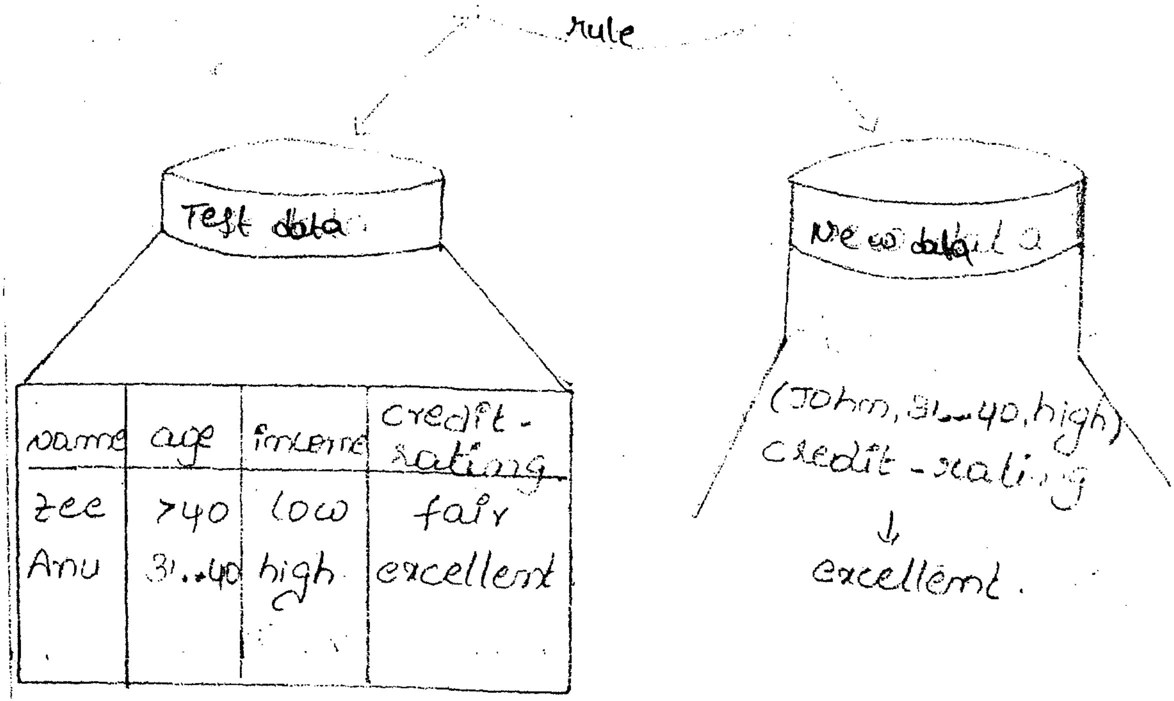


fig: 7.1 The Data classification.

a) Learning:

Training data is analyzed by using classification algorithm & here class label attr is credit-rating. Then learning data is transformed into classification rule.

b) classification:

The classification rule is applied on te data, to find its accuracy. If it is accurate i.e., applied for newdata.

In fig 7.1 (a) it analyzes the existing data of customers & using this class label of ne customers is predicted.

The prediction is different from classification. The prediction allows construct the model & use the model to find the class label of unlabelled sample. The prediction uses the classification & regression.

1. **classification:**

using classification we predict the discrete values.

2. **Regression:**

using this, we predict continuous values.

7.2 Issues Regarding classification & Predictions

we use the following issues.

7.2.1. Preparing the data for classification & prediction:

Here, we initially examine the data to improve the efficiency, accuracy, scalability of classification & prediction.

i) **Data cleaning:**

Here noisy or missing data values are smoothed by using several smoothing techniques.

ii) **Relevant Analysis:**

Here, irrelevant attributes are removed from the data before applying classification & prediction.

iii) Data Transformation:

we already know higher levels contains the more frequent items rather than the low level. This is achieved by using concept hierarchy. The data transformation specifies generalization levels to concept hierarchy.

23/9 7.2.2 Comparing classification & Prediction Methods:

The classification & prediction methods are compared by using following methods.

Sai Ram 1. predictive Accuracy:

This specifies how we use the model to predict class label of new data. As well as it also, specifies the prediction must be accurate.

2. Speed:

It specifies, to predict the class label it must take the less no. of comparisons.

3. Robustness:

This specifies that we predict the class label, if the model contains some noisy data or missing values.

4. Scalability:

It specifies model is constructed from large db.

5. Interpretability

It specifies level of understanding &

also specifies how we interpret the model

7.3. classification by Decision Tree Induction:

Decision Tree means flow chart like tree structure. The internal nodes specifies test on attribute.

The branch is the outcome of the test result. The leaf node represents class. The top node of the tree is considered as a root node. The decision tree for concept hierarchy buys-computer is shown below.

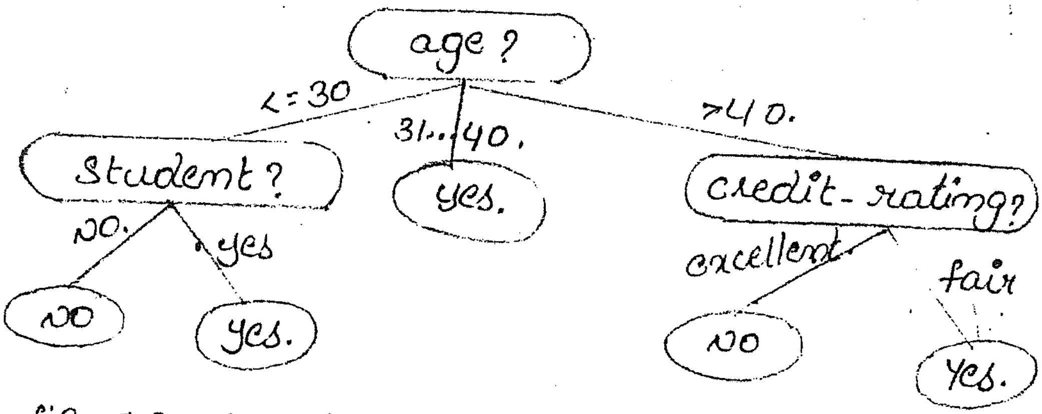


fig: 7.3. Decision Tree for concept buys-computer indicates whether the customer buys computer or not.

The internal nodes represents test on attribute. The leaf node class (buys-computer = no or buys-computer = yes).

This decision tree contains the adv. of we classify unknown sample & also this decision tree directly converted into association i.e basic decision tree alg

Algorithm: generate-decision-tree.

Input: The training samples.

Output: A decision tree.

Method:

1. create a node 'N'.
2. if all the samples are same class 'c' then
3. return 'N' as a leaf labelled with class 'c'
4. if attribute-list is empty
5. return N as a leaf labelled with most common class.
6. Select test-attribute from attribute-list with highest inf. gain.
7. Label node 'N'
8. Each known value 'a_i' of test-attribute
9. grow branch from node 'N'.
10. Let 'S_i' be the samples in test-attribute $= a_i$.
11. if S_i is empty then
12. attach the leaf labelled with most common class.
13. else attach node & recall generate-decision-tree.

7.3.1 Basic Decision Tree Induction Algorithm

7.3.1. Decision Tree Induction:

The above decision tree alg. follows the top down approach & divide & conquer approach.

The alg. is explained by using the following steps create a model (step 1)

- if the samples are same class, then return all the class labels related to that class (step 2,3)
- otherwise select the test-attribute by using entrophy based inf. gain.
- For each known value of test-attribute, create branch
- if the samples of test-attribute are empty, then return most common class labels. otherwise again recall the decision tree alg.

Attribute Selection:

Let 'S' be the training samples. It contains the 'm'-classes. Let 's_i' be any sample in 'S' with class label 'c_i' for $i = 1 \ldots m$. then expected inf. is needed to classify the given sample

$$I(s_1, \ldots s_m) = -\sum_{i=1}^{m} p_i \log_2(p_i)$$

Here 'p_i' is the probability of given sample fall in class label 'c_i'. It is calculated by $\frac{s_i}{s}$.

consider an attribute 'A' with values $\{a_1, a_2 \ldots a_v\}$. This attrib. 'A' partition the sample 'S' into subsets $\{S_1, S_2 \ldots S_v\}$

Let 'S_j' contains 's_{ij}' samples of class 'c_i'. then entrophy of 'A'

$$E(A) = \sum_{j=1} \frac{S_{ij} + \cdots + S_{mj}}{S} \cdot I(S_{ij}, \ldots, S_{mj})$$

∴ The inf. gain through this partition 'A' as

$$G(A) = I(S_1 \ldots S_m) - E(A).$$

The attibute with the highest inf. gain is treated as the most relevant attrib. that is used to construct the decision tree.

24/9

Ex. consider the data tuples from 'All Electronics' database.

Table: 7.3. Data tuples from All Electronics DataBase.

RID.	age	income	Student	credit-rating	class: buys-computer.
1.	<=30	low	no	fair.	no.
2.	<=30	high	no	excellent	no
3.	31...40	low	no	fair	yes.
4.	>40	medium	yes	excellent	no.
5.	>40	high.	no	fair.	yes.
6.	>40	low	no	fair	yes.
7.	31..40	high.	yes	fair	yes
8.	31..40	low	no	excellent	yes
9	31..40	medium	yes	fair	yes.
10.	<=30	"	yes	fair	yes.
11	<=30	low	"	excellent	yes.
12.	<=30	high	no	fair	no
13.	>40	low	no	excellent	no
14	>40	high.	[illegible]	fair	yes.

Here class label attribute is 'buys-computer'. It contains 2 values {yes, no}. The class 'C_1' is represented by samples of 'yes', the class 'C_2' is " " " " 'no'.

The above table contains '9' samples of 'yes' & '5' samples of 'no'. Therefore, the expected inf. is needed to classify the given sample.

$$I(S_1, S_2) = -\frac{9}{14}\log_2 9/14 - \frac{5}{14}\log_2 5/14$$

$$= 0.94.$$

Then we have to calculate the entropy value for each attribute starting with attribute 'age'. This attribute contains the different branches & fall in 2 classes {yes or no}.

for age = "<=30"

$S_{11} = 2.$ $S_{21} = 3.$ $\Rightarrow$ total 5

$$I(S_{11}, S_{21}) = -\frac{2}{5}\log_2 2/5 - \frac{3}{5}\log_2 3/5$$

$$= 0.97.$$

for age = "31...40"

$S_{12} = 4,$ $S_{22} = 0.$ $\Rightarrow$ total $\Rightarrow$ 4.

$I(S_{12}, S_{21}) = 0.$

for age = ">40" total $\rightarrow$ 5

$S_{13} = 3.$ $S_{23} = 2.$

$$I(S_{13}, S_{23}) = -\frac{3}{5}\log_2 3/5 - \frac{2}{5}\log_2 2/5$$

$0.97.$

∴ Entrophy of age is

$$E(age) = \frac{5}{14} I(S_{11}, S_{21}) + \frac{4}{14} I(S_{12}, S_{22}) + \frac{5}{14} I(S_{13}, S_{23})$$

$$= \frac{5}{14}(0.97) + \frac{4}{14}(0) + \frac{5}{14}(0.97)$$

$$= \frac{10}{14}(0.97)$$

$$= \frac{9.7}{14}$$

$$= 0.69.$$

$$Gain(age) = I(S_1, S_2) - E(age)$$

$$= 0.94 - 0.69$$

$$\boxed{= 0.25.}$$

Similarly, we can calculate the inf. gain for income is

gain (income) = 0.02

gain (Student) = 0.15

gain (credit-rating) = 0.04

→ gain (class: buys-computer)

Therefore, age contains the highest inf. gain. it is the most relevant attribute. Therefore, it is selected as the test attribute. Using this we construct the decision tree.

age?

<30

income	student	credit-rating	class.
low	no	fair	no
high	no	excellent	no
Medium	yes	fair	yes
low	yes	excellent	yes
high	no	fair	no.

>40

income	student	credit-rating	class.
Medium	yes	excellent	no
high	no	fair	yes
low	no	fair	yes.
Low	no	excellent	no
high	yes	fair	yes.

31..40

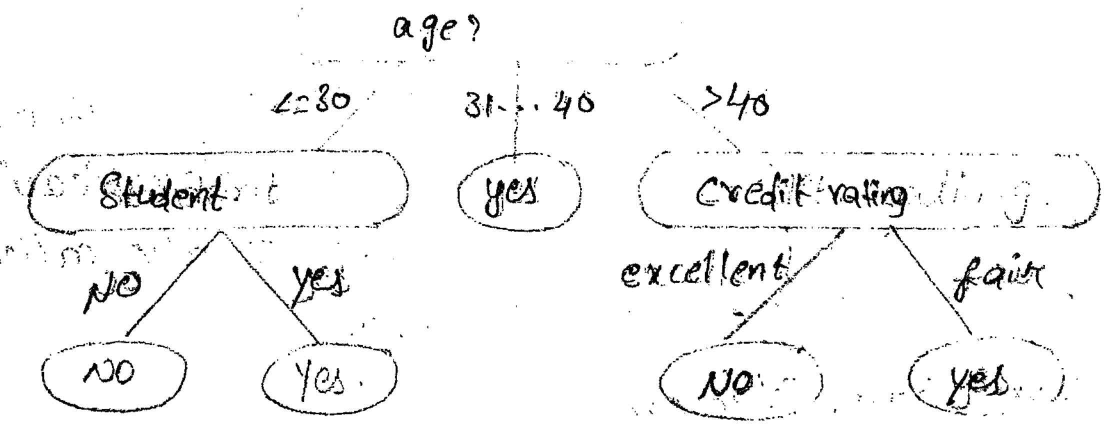
income	Student	credit-rating	class.
low	no	fair	yes
high	yes	fair	yes.
low	no	excellent	yes
Medium	yes	fair	yes.

fig. 7.3. The 'age' attribute is the highest inf. gain. ∴ it is the root node & branches grow with 'age' values.

Here the age 31..40 contains same samples i.e., class 's'.

The final decision tree based on the alg. is shown below.

fig: Decision tree for concept buys-computer.



7.3.2 Extracting classification rules from decision trees:

The knowledge represented in decision tree extracted & represented in the form of classification IF-THEN rule. One rule is derived for one path starting from root node to leaf node.

Consider the decision tree fig. 7.3.

The classification rules are,

IF age = "<=30" and Student = "no" THEN buys-computer = "no".

IF age = "<=30" and Student = "yes" THEN buys-computer = "yes".

IF age = "31...40" THEN buys-computer = "yes".

IF age = ">40" and credit-rating = "excellent" THEN buys-computer = "no"

IF age = ">40" and credit-rating = "fair" THEN buys-computer = "yes".

7.4. Bayesian Classification.

It is used to predict the class label of unknown sample. It is based on Bayes Theorem & then we apply simple Bayesian classification. The simple Bayesian classification is also called as naive Bayesian classification.

7.4.1. Bayes Theorem:

Let 'x' be an unknown sample & let 'H' be the hypothesis. Using this 'H', we predict the class label of 'x'. Finally we determine $P(H/x)$. This is called as posterior probability i.e., probability of 'H' on condition 'x'.

For ex, consider the data samples as 'fruits'. These are described by using 'colour' & 'shape'. Let 'x' is the red colour & round shape. Then 'H' gives, x is apple. Then

$P(H/x)$ gives the confidence of x is Apple because its colour is red & it is round shape.

To determine $P(H/x)$ it requires the additional inf. i.e., $P(H)$ gives the prior probability of H; $P(x)$ gives the prior probability of 'x' and $P(x/H)$ i.e., $P(x)$ on condition H.

Therefore using this Bayes Theorem we define the

$$P(H/x) = \frac{P(x/H) \cdot P(H)}{P(x)} \quad \text{--- (1)}$$

7.4.2 Naive Bayesian classification:

" " " or simple, Bayesian classification contains the following steps.

Step 1:

Consider the 'm' class i.e, $c_1, c_2 \ldots c_m$. Let 'x' be the unknown sample. Then according to naive Bayesian classification, the unknown sample 'x' is assigned to class 'c_i' if & only if $P(c_i/x) > P(c_j/x)$ for $1 \leq j \leq m$; $j \neq i$.

Thus we maximize probability of $P(c_i/x)$

But according to Bayes Theorem,

$$P(c_i/x) = \frac{P(x/c_i)\,P(c_i)}{P(x)} \quad \text{--- (2)}.$$

Step 2:

The $P(x)$ is same for all the classes. Then we maximize $P(x/c_i) \cdot P(c_i)$

For ex, prior probabilities are not known then all the classes are equal i.e,

$$P(c_1) = P(c_2) = \ldots = P(c_m).$$

Then we maximize $P(x/c_i)$ otherwise, we maximize $P(x/c_i)\,P(c_i)$

$$P(c_i) = S_i/S.$$

S_i = class c_i

S : Total no. of samples.

Step 3:

If there is no dependent relationship among the attributes. Then

$$P(X|c_i) = \prod_{k=1}^{n} P(x_k|c_i)$$

Here, we find the $P(x_1|c_i)$; $P(x_2|c_i)$.... $P(x_n|c_i)$

a) If A_k is categorical, then

$$P(x_k|c_i) = \frac{S_{ik}}{S_i}$$

where 'S_{ik}' is the training samples of class 'c_i' having value 'x_k' for 'A_k'.

b) If A_k is continuous-valued, then we use the gaussian distribution.

$$\therefore P(x_k|c_i) = g(x_k, \overset{\text{mean}}{\mu_{c_i}}, \overset{\text{SD}}{\sigma_{c_i}})$$

$$= \frac{1}{\sqrt{2\pi}\,\sigma_{c_i}} e^{-(x_k-\mu_{c_i})^2/2\sigma_{c_i}^2}$$

where μ_{c_i}, σ_{c_i} are the mean & standard deviation of A_k.

Step 4:

The unknown sample 'x' is classified by determining $P(X|c_i).P(c_i)$ for each class 'c_i'.

we assign unknown sample 'x' to class 'c_i' if & only if $P(X|c_i).P(c_i) > P(X|c_j).P(c_j)$

for $1 \leq j \leq m, j \neq i$.

Example:

consider the table 4.3.1 in that the data samples are represented by attributes age, income, student, credit-rating. The class label is buys-computer.

Buys-computer contains 2 classes {yes, no}.

The class 'c_1' is represented by buys-computer="yes"

" " c_2 " " " " " " = "no".

The unknown sample x,

x = (age = "<=30", income = "medium",
Student = "yes" credit-rating = "fair").

we have to maximize $P(x/c_i) \cdot P(c_i)$ for $i = 1, 2$.

The prior probabilities are calculated by

p(buys-computer = "yes") = 9/14 = 0.64.

p(buys-computer = "no") = 5/14 = 0.36.

To calculate $P(x/c_i)$ for $i = 1, 2$, we first of all calculate the conditional probabilities.

p(age = "<=30" | buys-computer = "yes") = $\frac{2}{9}$ = 0.22.

p(age = "<=30" | buys-computer = "no") = $\frac{3}{5}$ = 0.6

P(income = "medium" | buys-computer = "yes") = $\frac{2}{9}$ = 0.22

p(income = "medium" | buys-computer = "no") = $\frac{1}{5}$ = 0.2.

p(student = "yes" | buys-computer = "yes") = $\frac{5}{9}$ = 0.56.

P(student = "yes" | buys-computer = "no") = $\frac{1}{5}$ = 0.2.

$P($credit-rating = "fair" / buys-computer = "yes")
$= 7/9 = 0.78.$

$P($credit-rating = "fair" | buys-computer = "no")
$= 2/5 = 0.4.$

$\therefore$ $P(x$ | buys-computer = "yes") $= 0.22 \times 0.22 \times 0.56 \times 0.78$
$= 0.02$

$P(x$ | buys-computer = "no") $= 0.6 \times 0.2 \times 0.2 \times 0.4$
$= 0.0096.$

$P(x|c_i)$ $\qquad$ $P(c_i)$

$P(x$ | buys-computer = "yes"). $P($buys-computer = "yes")
$= 0.02 \times 0.64$
$= 0.0128.$

$P(x$ | buys-computer = "no"). $P($buys-computer = "no")
$= 0.0096 \times 0.36$
$= 0.003456.$

Therefore,

$P(x$ | buys-computer = "yes"). $P($buys-computer = "yes")
$>$ $P(x$ | buys-computer = "no"). $P($buys-computer = "no")

i.e., $c_1 > c_2$.

Then, unknown sample 'x' is assigned to buys-computer = "yes".

3/W Classification by Backpropagation :

The Back propagation based on the neural n/w Learning algorithm.

Here neural means set of input & output lines are connected & also each line assigns the weighted value. Then in learning process, the n/w learns by Modi-fying the weights of each connection. Then this is used to identify correct class labels of i/p samples.

The back propagation mainly applies on multilayer Feed-Forward neural n/w. & then we define the n/w topology. Finally, we apply the Back propagation.

7.5.1. Multilayer Feed-Forward neural n/w :

This " " " " " is shown below.

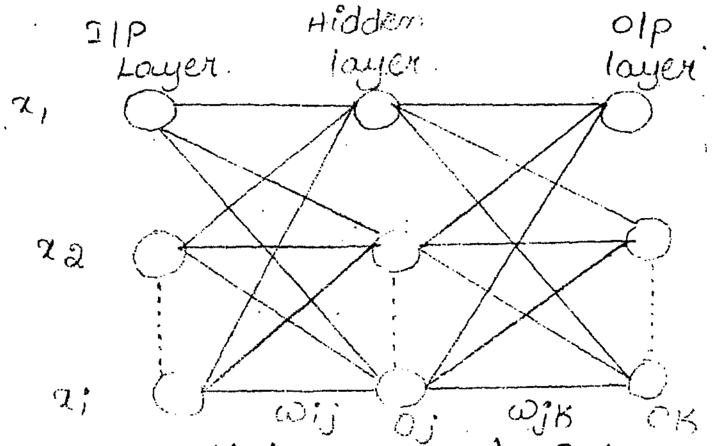


Fig. 7.5. A multi-layer Feed-Forward neural n/w

The training sample $x = \{x_1, x_2 \dots x_i\}$ form the I/p Layer.

each pair of adjacent layers. it contains the weighted connection.

For ex. w_{ij} is the weight from i to j.

The multilayer means it contains only one hidden layer. Feed-Forward means, the weighted connections never come back.

The o/ps of I/p layer are the I/ps of hidden layer & so on.

7.5.2 Defining a n/w Topology:

Here end user has to define the n/w topology by specifying no. of units in i/p layer, no. of units in hidden layer, no. of units in o/p layer.

7.5.3 Back Propagation:

In this, the samples are repeated continuously. For each data sample, we modify the weights to minimize the error b/w n/w prediction & well-known class label.

The back propagation contains the following steps.

i. Initialize the weights:

Here the weighted connection b/w each pair of adjacent layers, we assign small random value.

for ex: this value ranging from
(-1 to 1 (or) -0.5 to 0.5)

ii) Foward the Selected inputs:

In this step, we find net i/p & o/p for each unit & also each layer.

For ex. unit is 'j' in hidden or o/p. layer. then its net i/p is calculated by using a formula.

$$I_j = \sum_i w_{ij} O_i + \theta_j \quad \text{--- (1)}$$

Here w_{ij} – is weight from i to j

O_i – o/p of unit 'i'.

θ_j – o/p of Bias. using this we differentiate the diff. i/p units.

I_j – net i/p for unit j.

These i/p s are used by hidden or o/p layers. Then these layers apply activation function. This is shown in below.

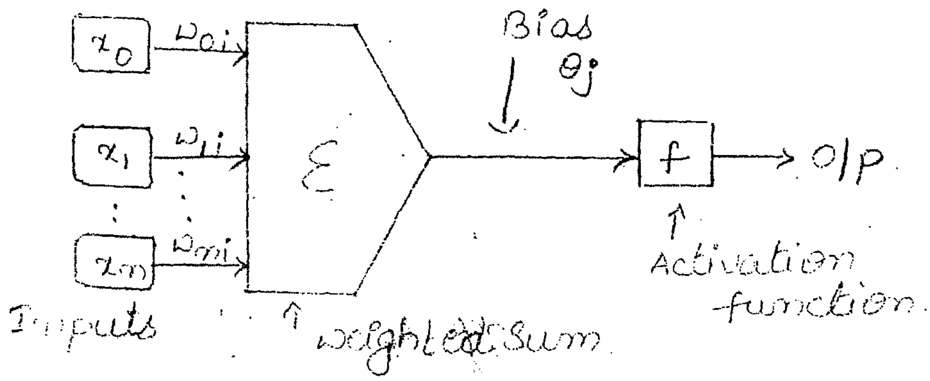


fig 7.5.1 Activation function

Let unit 'j' & its net I/p I_j. Then O_j is the o/p for unit 'j'.

$$O_j = \frac{1}{1+e^{-I_j}} \quad \text{— ②}$$

7/10 iii) Propagate error Backward ~~forward~~:

To propagate the error Backward ~~forward~~, We ~~update~~ modify the weights & Bias values.

For unit 'j' the error 'Err_j' for o/p layer is calculated by using a formula

$$Err_j = O_j(1-O_j)(T_j - O_j) \quad \text{— ③}$$

Here O_j - o/p of unit 'j'.

T_j - o/p of true value for known class label.

The error at hidden layer for unit 'j' is calculated by using the formula,

$$Err_j = O_j(1-O_j)\sum_K Err_K\, w_{jK} \quad \text{— ④}$$

The weights are updated by using the formula.

$$\Delta w_{ij} = (l)\, Err_j\, O_i$$

$$w_{ij} = w_{ij} + \Delta w_{ij} \quad \text{— ⑤}$$

Here Δw_{ij} - change of weight w_{ij}.

l - linear rate. it is a constant value & ranging b/w 0.0 to 1.0.

IIly Bias are updated by using the formula

$$\Delta\theta_j = (l)\, Err_j$$

$$\theta_j = \theta_j + \Delta\theta_j \quad \text{—— (6).}$$

Ex: consider the following multilayer feed forward neural n/w

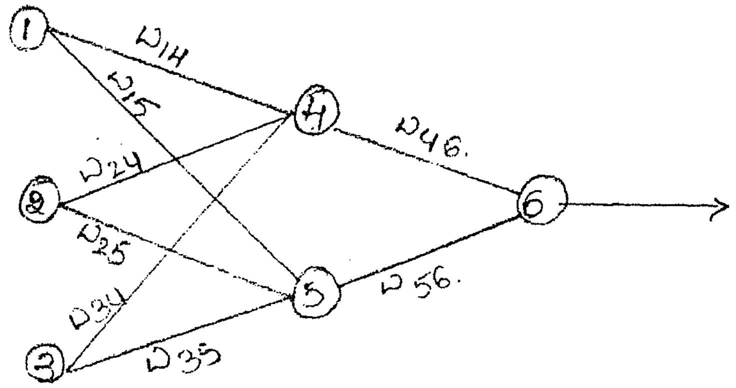


fig. 7.6. Multilayer Feed-Forward neural n/w.

The training sample, $x = (1, 0, 1)$ whose class label is '1' & linear rate is 0.9. The initial i/ps, weights & Bias values are as shown in below.

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

Table: 7.7. Initial I/p weight & Bias values.

calculation of net I/p & o/p values:

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.	$(-0.3)(0.332) - (0.2)(0.525) = -0.105$	$1/(1+e^{0.105}) = 0.474$

Calculation of Error:

unit j	Error
6	(0·474) (1-0·474) (1-0·474) = 0·1311
5	(0·525) (1-0·525) (0·1311) (-0·2) = -0·0065
4	(0·332) (1-0·332) (0·1311) (-0·3) = -0·0087

Calculation of weights & Bias updates:

weight / bias	New value
w_{46}	(-0·3) + (0·9) (0·1311) (0·332) = -0·260
w_{56}	(-0·2) + (0·9) (0·1311) (0·525) = -0·138
w_{14}	0·2 + (0·9) (-0·0087) (1) = 0·192
w_{15}	-0·3 + (0·9) (-0·0065) (1) = -0·3058
w_{24}	0·4 + (0·9) (-0·0087) (0) = 0·4
w_{25}	0·1 + (0·9) (-0·0086) (0) = 0·1
w_{34}	-0·5 + (0·9) (-0·0087) (1) = 0·5078
w_{35}	0·2 + (0·9) (-0·0065) (1) = 0·194
θ_6	0·1 + (0·9) (0·1311) = 0·2179
θ_5	0·2 + (0·9) (-0·0065) = 0·194
θ_6	-0·4 + (0·9) (-0·0087) = -0·407

Termination condition:

Δw_{ij} becomes the small i.e., it is less than the specified Threshold range.

8/10

8. Cluster Analysis.

8.1. What is cluster Analysis?

The process of storing similar objects in one cluster & dissimilar objects in another cluster. The data clusters are shown in below.

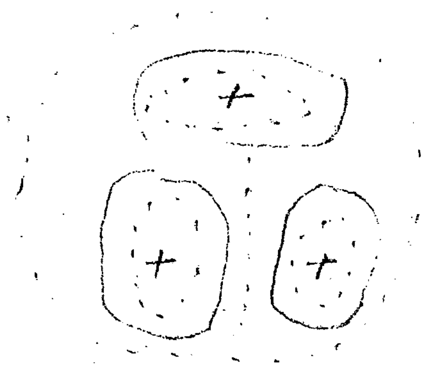


fig. 8.22-D customer Data w.r to customer locations in city. Here Three data clusters are specified & centre of cluster is marked with '+'.

The following are the critical requirements for cluster in data Mining.

(1) Scalability:

The clustering algorithms are efficient for low data, but DM system contains large volume of data. to hande this large volume of data, the clustering ~~analysis~~ alg.s must be highly scalable i.e., efficient.

8.2 Partitioning Methods.

The simplest and most fundamental version of cluster analysis is partitioning, which organizes the objects of a set into several exclusive groups or clusters.

Given a data set D, of n objects and K, the no. of clusters to form, a partitioning algorithm organizes the objects into K partitions $K \le n$, where each partition represents a cluster.

Partitioning methods :-

1) K-means : A centroid - Based Technique.
2, K- medoids: A Representative Object - Based Technique.

1) K-means: A Centroid-Based Technique :-

K-means clustering intends to partition n objects into K clusters in which each object belongs to the cluster with the nearest mean. This method produces exactly K different clusters of greatest possible distinction.

The best number of cluster K leading to the greatest separation is not known as a prior and must be computed from the data. The objective of K-means clustering is to minimize total intra-cluster variance or the squared error function:

$$E = \sum_{i=1}^{k} \sum_{p \in C_i} dis(p, c_i)^2$$

$$E = \sum_{i=1}^{k} \sum_{p \in C_i} dist(p, c_i)^2,$$

where E is the sum of the squared error.

P is the point in space representing a given object.

c_i is the centroid of cluster C_i.

Algorithm:

K-means algorithm for partitioning, where each clusters is represented by the mean value of the objects in the cluster.

Input:

K: the no of clusters,

D: a data set containing n objects.

Output:- A set of K clusters.

Method:-

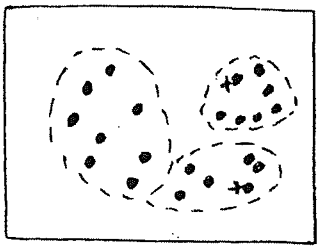
1) arbitarily choose K objects from D as the initial cluster centers;

2) repeat

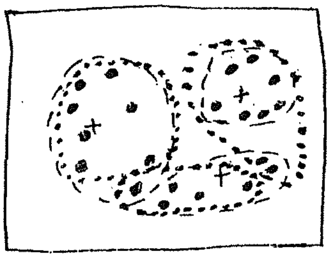
3) assign each object to the cluster to which the object is the most similar, based on the mean value of the objects in the cluster;

4) update the cluster means, i.e, calculate the mean value of the objects for each cluster;

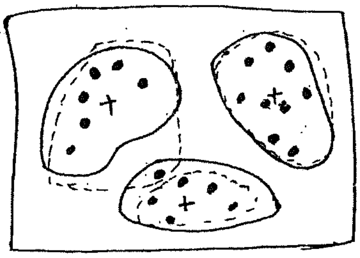
5) until no change;



(a) Initial clustering.



(b) Iterate.



(c) Final clustering.

clustering of a set of objects using the k-means

Consider a set of objects located in 2-D space, as depicted in fig (a). Let $k=3$, i.e, the objects to be partitioned into 3 clusters.

According to the algorithm, we arbitrarily choose 3 objects as 3 initial cluster centers, cluster centers are marked by '+'. Each object is assigned to a cluster based on the cluster center to which it is the nearest.

Next the cluster centers are updated i.e, the mean value of each cluster is recalculated based on the current object in the cluster.

$E = \sum_{i=1}^{k} \sum_{p \in o_i} |p_i - o_i|$

using this new cluster centers, the objects are redistributed to the cluster based on which cluster center is the nearest. The process of iteratively reassigning object to clusters to improve the Partitioning is referred to as iterative velocation.

The time complexity of the k-means algorithm is $O(nkt)$, where n is the total no. of objects, k is the no. of clusters, & t is the no. of iterations. therefore, the method is relatively scalable & efficient in Processing large data sets.

2, K-Medoids: A Representative Object-Based Technique:-

The K-means algorithm is Sentive to outliers because such objects are far away from the majority of the data, they can dramitically distroy the mean value of the cluster. This affects the assignment of other objects to clusters. This effects is particularly due to the use of the squared error function.

The K-medoids is an iterative clustering algorithm which iterates until each representative object is the mediad or most centraly located object of its cluster.

$E = \sum_{i=1}^{k} \sum |p - o_i|$

The most common realisation of K-medoid clustering is Partitioning Around Medoids (PAM)

K-Medoids Algorithm (PAM) :-

Input :

K: the no. of clusters

D: a dataset containing n objects

Output :-

A set of K clusters

method :-

1) Arbitrary choose K objects from D as representative objects.

2, Repeat

3, Assign each remaining object to the cluster with the nearest representative object

4, For each representative object oj

5, Randomly select a non representative object Orandom.

6, Compute the total cost S of swapping representative object Oj with Orandom.

7, if S<0 then replace Oj with Orandom

8, Until no change.

$O(k(n-k)^2)$.

The complexity of each iteration is $O(k(n-k)^2)$. For large values of n and k, such computation becomes very costly.

Advanges

Advantages:—

* K-Medoids method is more robust than K-Means in Presence of noise and outliers.

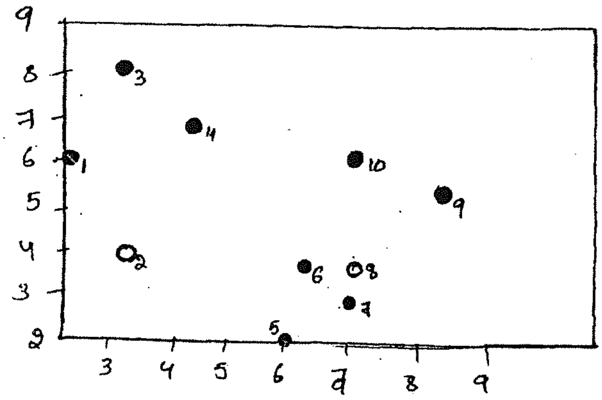
Disadvantages:—

* K-medoids is more costly that the K-Means Method
* Like K-means, K-medoids requires the user to Specify K
* It does not Scale well for large data Sets.

Example:—

Data objects

	A_1	A_2
O_1	2	6
O_2	3	4
O_3	3	8
O_4	4	7
O_5	6	2
O_6	6	4
O_7	7	3
O_8	7	4
O_9	8	5
O_{10}	7	6

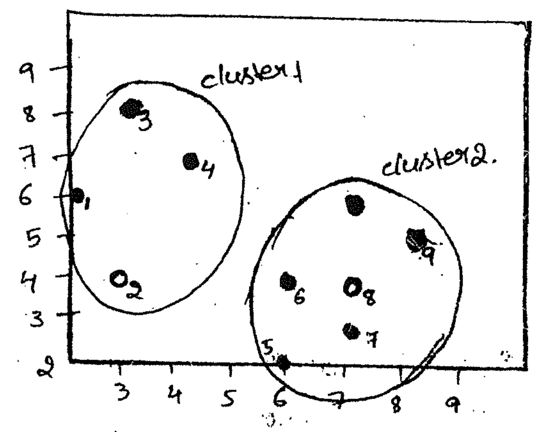


Goal: Create two clusters

choose randomly two medoids

$O_2 = (3,4)$

$O_8 = (7,4)$



* The absolute error/criterion (for the set of medoids (O_2, O_8))
* Assign each object to the closest representative object
* using L1 Metric (Manhattan), we form the following clusters

cluster1 = $\{O_1, O_2, O_3, O_4\}$
cluster2 = $\{O_5, O_6, O_7, O_8, O_9, O_{10}\}$

* Compute the absolute error criterion [for the set of Medoids (O_2, O_8)]

$$E = \sum_{i=1}^{k} \sum_{p \in C_i} |p - o_i| = |O_1 - O_2| + |O_3 - O_2| + |O_4 - O_2| +$$
$$|O_5 - O_8| + |O_6 - O_8| + |O_7 - O_8| + |O_9 - O_8| +$$
$$|O_{10} - O_8|.$$

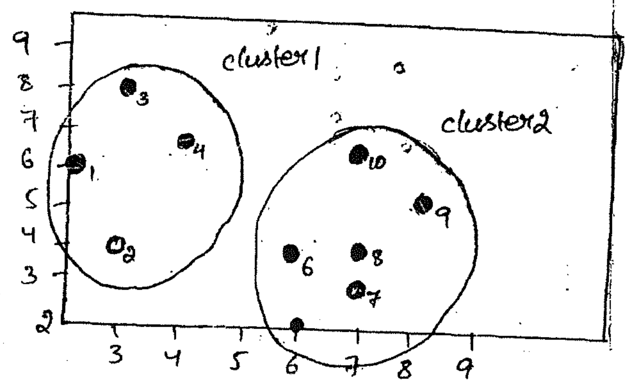
* The absolute error criterion [for the set of Medoids (O_2, O_8)]

$$E = (3+4+4) + (3+1+1+2+2) = 20$$

* choose a random object O_7
* Swap O_8 and O_7

* Compute the absolute error criterion [for the set of medoids (O_2, O_7)]

$E = (3+4+4) + (2+2+1+3+3) = 22.$



* Compute the cost function

Absolute error $[O_2, O_7]$ − Absolute error $[O_2, O_8]$

$S = 22 - 20$

$S > 0 \Rightarrow$ It is a bad idea to replace O_8 by O_7

8.3 Hierarchical Methods:-

A hierarchical clustering method creates a hierarchical structure from data objects. The hierarchical clustering methods can be classified into two types. They are,

1) Agglomerative (bottom-up)
2) Divisive (top-down).

Agglomerative Hierarchical Clustering:-

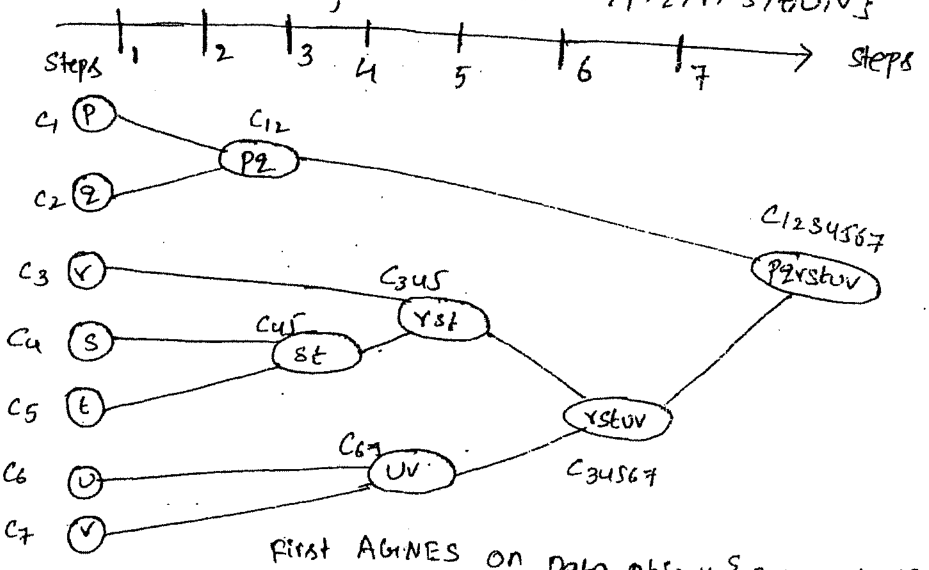
This involves merging of data objects. It initiates with each object forming its own group/cluster. For every pair of cluster, some value of dissimilarity is computed, then the clusters are merged into larger and larger clusters until all the clusters are merged into one largest cluster or until a termination condition is reached. The merging of clusters is carried-out based on the Euclidean distance b/w any two objects from different clusters. The user can set the termination criteria by fixing the desired no. of clusters.

Divisive Hierarchical clustering:-

This clustering method is also known as top-down method. This involves division of objects cluster into smaller parts. It initiates with all the objects in the same cluster. This single cluster is splitted into smaller clusers, until each object is one cluster or until a termination condition is reached.

Example for AGNES and DIANA :- AGNES (Agglomerative Nesting)

A dataset of seven objects {p, q, r, s, t, u, v}



First AGNES on Data objects {p, q, r, s, t, u, v} connec

alg: (

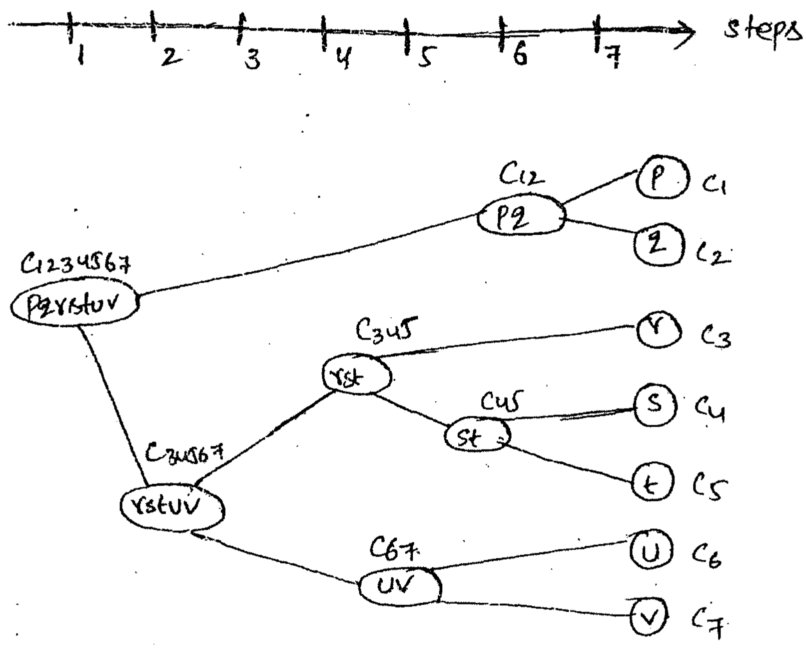
with

dens

of

NO'

be



1, (DIANA) Divisive Analysis on Data Objects {p, q, r, s, t, u, v}

In 'AGNES', all the objects are placed into individual clusters ($C_1 - C_7$), from the second step onwards, the objects are merged into larger objects based on the minimum Euclidean distance b/w any two objects from diff. clusters this process is continues until a single cluster containing all the objects is obtained.

2,

In DIANA, all the data objects are first placed into a single cluster ($C_{1234567}$). This single cluster is then divid into smaller clusters based on maximum Euclidean distance b/w closest neighbouring objects in the cluster. This proce continues until each objects is placed in an individual clu

8.3.2 Distance Measures in Algorithmic Methods :-

whether using an agglomerative method or a divisive method, a core need is to measure the distance b/w 2 clusters, where each cluster is generally a set of objects.

Four widely used measures for distance b/w clusters are as follows.

minimum distance : $dist_{min}(C_i, C_j) = \min_{p \in C_i,\, p' \in C_j} \{|p - p'|\}$

Maximum distance : $dist_{max}(C_i, C_j) = \max_{p \in C_i,\, p' \in C_j} \{|p - p'|\}$

mean distance : $dist_{mean}(C_i, C_j) = |m_i - m_j|$

Average distance : $dist_{avg}(C_i, C_j) = \frac{1}{n_j \cdot n_i} \sum_{p \in C_i,\, p' \in C_j} |p - p'|$.

where $|p - p'|$ is the distance b/w 2 objects or points

m_i is the mean for cluster C_i,

n_i is the no. of objects in C_i

They are also known as linkage measures. when an algorithm uses the minimum distance, $d_{min}(C_i, C_j)$ to measure the distance b/w clusters, it is called as "nearest - neighbor clustering algorithm".

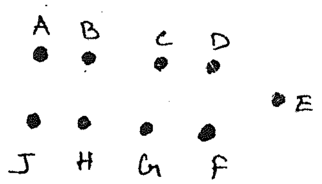
An agglomerative hierarchical clustering algorithm uses the minimum distance measure is also called a "minimal spanning tree algorithm". The minimal spaning tree is the one with the least sum of edge weights.

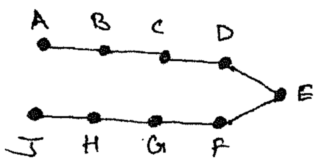
when an algorithm uses the maximum distance $d_{max}(c_i, c_j)$, to measure the distance b/w clusters. it is called as "farther-neighbor clustering algorithm". 8.3

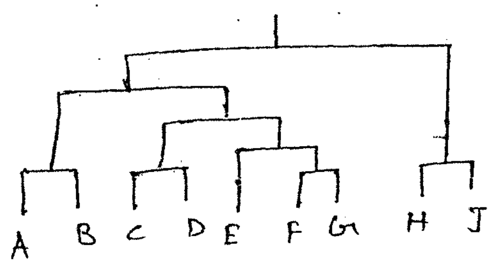
The use of mean or average distance is a compromise b/w the minimum & maximum distance and overcomes the outliers sensitivity problem.

<u>Example :-</u>

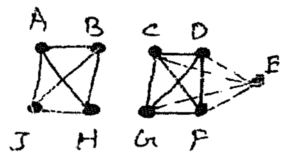
let us apply the hierarchical clustering to the data set {A, B, C, D, E, F, G, H, J}

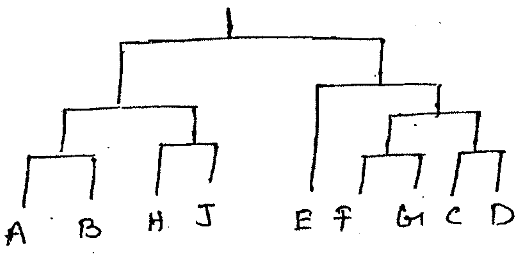






clustering using single linkage





clustering using complete linkage.

8.3.3 BIRCH : Multiphase Hierarchical clustering using Clustering Feature Trees

BIRCH (Balanced Iterative Reducing and Clustering using Hierarchies) method integrates both hierarchical clustering methods, in initial stage and iterative partioning clustering method at final stages. In other words, the o/p generated from hierarchical methods serve as I/p or preprocessing step for iterative partitioning.

Advantages of BIRCH when Compared to Hierarchical Methods

1) BIRCH method improves the Scalability of clustering.
2) It has the ability to undo the mistakes which have occured in previous phases.

In BIRCH method, for an input data points, the following three factors are calculated.

1) Centroid (C)

It determines the center of the cluster. It is

Calculated using the formula, $G = \frac{\sum_{i=1}^{n} z_i}{n}$

2, <u>Radius (r)</u>

It determines the average distances of the I/p Point from the centroid.

$$r = \sqrt{\frac{\sum_{i=1}^{n} (z_i - G)^2}{n(n-1)}}$$

3, <u>Diameter (D)</u>

It determines the average pair wise distance in a Cluster.

$$D = \sqrt{\frac{\sum_{i,j=1}^{n} (z_i - z_j)^2}{n(n-1)}}$$

The radius and diameter factor determines the level of association of the cluster around the centroid.

<u>Clustering Feature and Clustering Feature Tree</u>:-

In BIRCH method, clustering Feature (CF) is defined as a three- dimensional vector. which contains the info about the clusters of objects that have been discarded or compressed in a summarized form.

The 'CF' for 'n' data points spread over d-dimensions can be represented as,

$CF = (n, L_{sum}, S_{sum})$

where

$n \rightarrow$ no. of data points (or) objects.

$L_{sum} \rightarrow$ linear sum of data points,

$S_{sum} \rightarrow$ Square sum of data points.

Clustering Feature Tree (CFT) is height-balanced tree which is used to represent clustering features of individual clusters in an hierarchical fashion.

The non leaf nodes maintain the info. about the sums of the clustering features of their child nodes and thereby summarizing the clustering info. of these child nodes.

The size of a clustering feature tree is dependent on two factors.

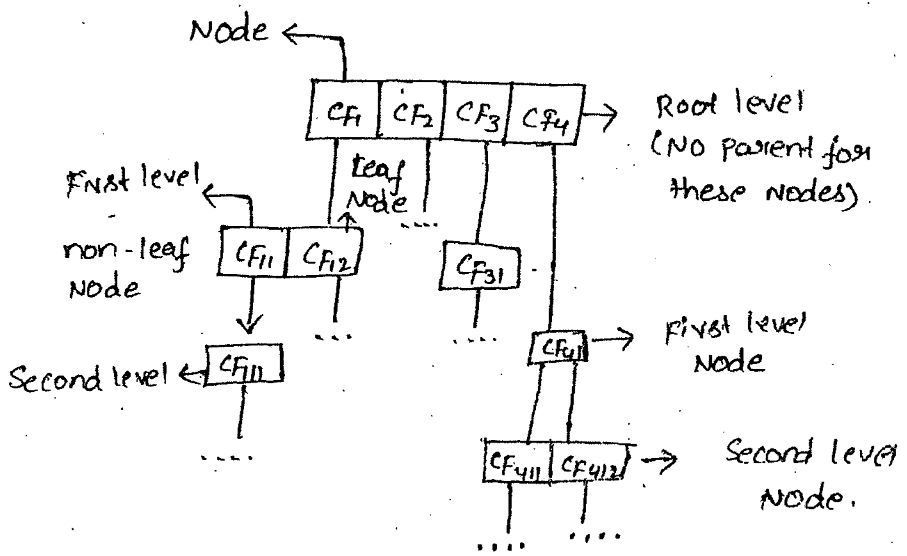
1, Branching factor (B)

This factor decides the maximum no. of child nodes for a non leaf node.

2, Threshold (T)

Threshold decides the maximum diameter that a sub cluster i.e, a collection of non-leaf and its child nodes.

Example of CF Tree



BIRCH method is used to generate efficient clusters using those resources that are available. This method applies multiphase clustering technique which generates good clustering with minimum intervention from I/o and minimum amount of main memory for storage.

8.3.4 Chameleon: A Hierarchical clustering Algorithm using Dynamic Modeling :-

This method uses dynamic modeling. It basically constructs a sparse k-nearest neighbor graph, then partitions the graph into pieces and then clusters the pieces together. produces more natural clusters than DBSCAN. uses density measurements to determine the k-nearest neighbor.

tures:-

dapt to the characteristics of the data set to find the
atural clusters.

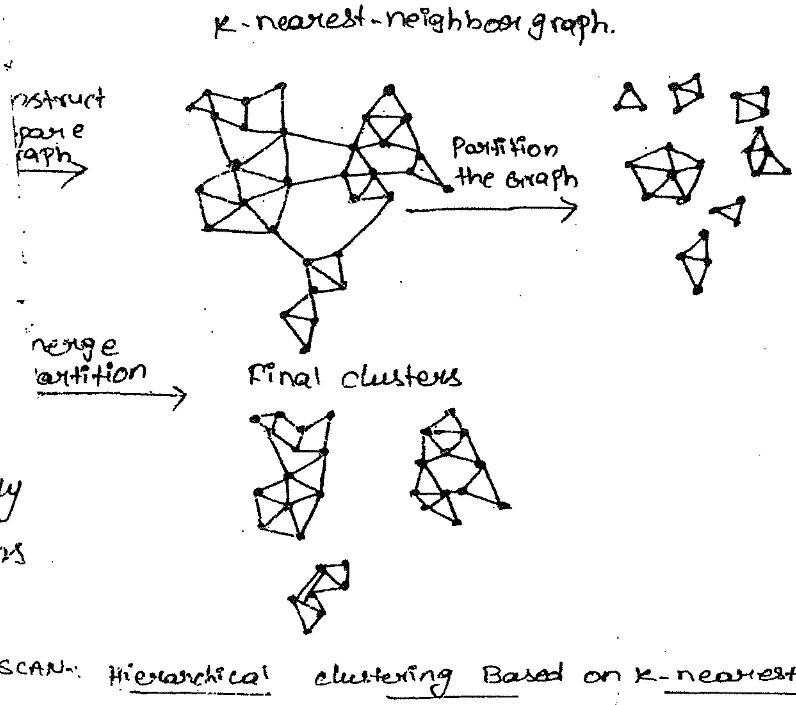
se a dynamic model to measure the similarity
w clusters.

main property is the relative closeness and relative
er connectivity of the cluster.

Two clusters are combined if the resulting cluster
ares certain properties with the constituent cluste

The merging scheme preserves self-similarity

e of the areas of application is spatial data.



Hierarchical clustering Based on k-nearest Neighbours
mic Modeling.

level
parent for
se nodes)

1st level
node

Second level
Node.

efficient clusters
lable. This method
ue which generates
tervention from I/o
mory for storage.

ustering Algorithm

modeling. It basically
r graph. then partitions
clusters the pieces
ural clusters than DBSCAN.
determine the k-nearest

Chameleon Steps :-

Preprocessing :-

Represent the data by a graph

1, Given a set of points, construct the k-nearest neighbor (k-NN) graph to capture the relationship b/w a point and its k-nearest neighbours.

2, concept of neighborhood is captured dynamically.

Phase 1 :-

use a multilevel graph partitioning algorithm on the graph to find a large no. of clusters of well-connected vertices.

Each cluster should contain mostly points from one "true" cluster i.e, is a sub-cluster of a "real" cluster.

Phase 2 :-

use hierarchical agglomerative clustering to merge sub-clusters.

1, Two clusters are combined if the resulting cluster shares certain properties with the constituent clusters.

2, There are two key properties used to model cluster similarity.

* Relative Interconnectivity :-

Absolute interconnectivity of two clusters normalized by the internal connectivity of the clusters.

* **Relative closeness :-**

Absolute closeness of two clusters normalized by the internal closeness of the clusters.

4 Density-Based Methods :-

Density based clustering methods have been developed to determine clusters with arbitary space. clustering based on density, such as denisity-connected points.

Major Features of Density-Based Methods :-

1) Discover clusters of arbitorary shape.
2) Handle noise.
3) One scan.
4) Need density parameters as termination condition.

There are three different methods of density-based clustering. they are.

1) DBSCAN
2) OPTICS
3) DENCLUE

8.4.1 DB SCAN :-

Density-Based Spatial Clustering of Application with Noise (DBSCAN) is a density-based clustering algorithm. In DBSCAN a cluster is a set of maximum density connected points DBSCAN finds arbitrary - shaped clustering alg: and finds arbitrary - shaped clusters in spatial DBs with noise and expands regions having enough high density into clusters. To better understand the concept of DBSCAN method we first need to know some definitions.

Noise

Noise is defined as the set of objects in N which do not belong to any cluster.

1, ε- neighborhood of an object:-

The ε- neighborhood of an object is defined as for a given positive radius, the neighborhood of an object must be with in the radius(ε).

2, Core object:-

An object is said to be a core object, if the ε- neighborhood of an object contains at least a minimum no. of objects (Minpnts).

3, Directly - Density Reachable:-

An object 'e' belonging to a set of objects, N is directly density- reachable from an object 'f', if 'f' is a core object and e is with in ε- neighborhood of f.

4) <u>Density - Reachable</u> :-

Given a neighborhood ε and minipnts in a group of objects N and when there is a series of objects $e_1, e_2, \ldots e_n$ where $e_1 = f$ and $e_n = e$ ∋ e_{a+1} is directly density reachable from e_n with respect to ε and minipnts, for $1 \le a \le n$, $e_a \in N$ then an object e is density reachable from object f.

5) <u>Density - connected</u> :-

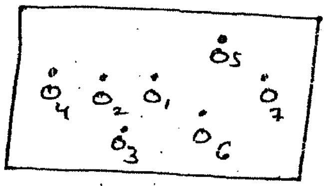
Given ε and minipnts in a group of objects N and when there is an object 'g' belonging to N ∋ both e & f are density - reachable from 'g' with respect to ε and minpnts. then an object 'e' is density - connected to object f.

6) <u>Density clusters</u> :-

A cluster with respect to ε and minpnts is a non-empty subset of N containing maximum no. of density connected objects with respect to density - reachability.

<u>Example</u> :-

let us assume that minpnts = 5 and ε = 2 cms.



objects before clustering.

<u>steps for finding clusters</u> :-

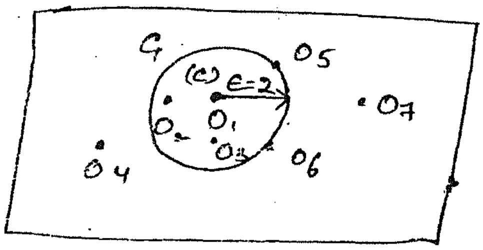
1) Select an unclassified object O_1. The objects that are in the neighborhood of O_1 along with the radius 2 are,

$N_{\varepsilon}(O_1) = \{O_1, O_2, O_3, O_5, O_6\}$

since, $N_{\varepsilon}(O_1) \geq$ minpnts i.e, $5 \geq 5$

O_1 is considered as a core object (c)

Thus these set of points together form a cluster, which is assigned a cluster-id (C_1). this is shown in below.



O_1 Neighboring Objects

2, Now, select the object O_2 from the set. the neighboring Objects of O_2 are

$N_{\varepsilon}(O_2) = \{O_1, O_2, O_3, O_4\}$

since $N_{\varepsilon}(O_1) <$ minpts i.e, $4 < 5$, the object O_2 is considered as a 'Noise' object (N) and therefore no cluster is formed.

3, select the object O_3, O_4, and O_5 from set. The neighboring objects of O_3, O_4, O_5 are

$N_{\varepsilon}(O_3) = \{O_1, O_2, O_3, O_6\}$, $N_{\varepsilon}(O_4) = \{O_2, O_4\}$

$N_{\varepsilon}(O_5) = \{O_1, O_5, O_6, O_7\}$

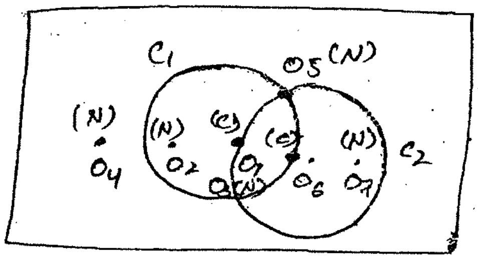
Since $N_{\varepsilon}(O_3) <$ minpnts, $N_{\varepsilon}(O_4) <$ minpnts, $N_{\varepsilon}(O_5) <$ minpnts;

so, these are consider as Nosie objects (N). and there fore no cluster is formed.

(iv) Finally, select the object O_6 from the set. The neighboring objects of O_6 are

$$N_\epsilon(O_6) = \{O_1, O_3, O_5, O_6, O_7\}$$

Since, $N_\epsilon(O_6) \geq$ minpnts i.e, $5 \geq 5$ the object O_6 is considered as core object. Thus, these set of points form a cluster, & assigned a cluster-id (C_2).



O_1 & O_6 objects with their neighbors forming clusters C_1 & C_2.

The resultant DB SCAN clustering in which there are two clusters as,

$C_1 = \{O_1, O_2, O_3, O_5, O_6\}$ &

$C_2 = \{O_1, O_3, O_5, O_6, O_7\}$

where O_1, O_6 are core objects and O_2, O_3, O_4, O_5, O_7 are noise objects.

8.4(d) OPTICS :-

Algorithm:—

DBSCAN : a density-based clustering algorithm.

Input:

D: a dataset containing n objects,

ϵ: the radius parameter &

MinPts: the neighborhood density threshold.

Output:- A set of density-based clusters;

Method:-

1, Mark all objects as unvisited;

do

randomly select an unvisited object p;

mark p as visited;

if the ϵ-neighborhood of p has at least Minpnts objects

Create a new cluster C and add p to C;

let N be the set of objects in the ϵ-neighborhood of p;

for each point p' in N

if p' is unvisited

mark p' as visited;

if the ϵ-neighborhood of p' has at least minpnts points

add those points to N;

if p' is not yet a member of any cluster, add p' to C;

end for

Output C;

else mark p as noise;

until no object is unvisited;

classification according to the database mined.
class accor to the knowle mined.
class accord to the techniques utilized.
class accor to the applications adopted.