UNIT 1

INTRODUCTORY CONCEPTS

Que 1: What is a data warehouse? What are its characteristics?

A decision support database that is maintained separately from the organization's operational database and Supports information processing by providing a solid platform of consolidated, historical data for analysis.

(Or)

A repository of multiple heterogeneous data sources organized under a unified schema at a single site in order to facilitate management decision making.

(Or)

"A data warehouse is a subject-oriented, integrated, time-variant, and nonvolatile collection of data in support of management's decision-making process."

Subject-oriented: data warehouses typically provide a simple and concise view around particular subject issues by excluding data that are not useful in the decision support process. Such as customer,supplier, product, and sales etc..

Integrated: A data warehouse is usually constructed by integrating multiple heterogeneous sources, such as relational databases, flat files, and on-line transaction records.

Time-variant: Data are stored to provide information from a historical perspective (e.g., the past 5–10 years).

Nonvolatile: A data warehouse is always a physically separate store of data transformed from the application data found in the operational environment. It usually requires only two operations in data accessing: initial loading of data and access of data.

Que 2: what are the differences between operational database systems and data warehouses?

	OLTP	OLAP
users	clerk, IT professional	knowledge worker
function	day to day operations	decision support
DB design	application-oriented	subject-oriented
data	current, up-to-date detailed, flat relational isolated	historical, summarized, multidimensional integrated, consolidated
usage	repetitive	ad-hoc
access	read/write index/hash on prim. key	lots of scans
unit of work	short, simple transaction	complex query
# records accessed	tens	millions
#users	thousands	hundreds
DB size	100MB-GB	100GB-TB
metric	transaction throughput	query throughput, response

Que 3: Why separate data warehouse is preferable?
(Or)
Why have separate data warehouse?

High performance of both systems • An operational database is designed to perform Predefined or known tasks workloads, such as indexing and hashing using primary keys, searching for particular records. • On the other hand, data warehouse queries are often complex. They are designed to analyze data at different levels of abstraction. And deal with multidimensional views.
Concurrency Control • An operational database supports Concurrency control, deadlock management, to ensure consistency, and have robustness of transactions. • An OLAP query often needs read-only access of data records for summarization and aggregation. Concurrency control when applied in OLAP operations reduces the throughput of OLTP system.
Date Needed • In case of OLAP, Decision support requires historical data. • Whereas operational databases do not typically maintain historical data.

A MULTIDIMENSIONAL DATA MODEL

Que 4: write a short note on.

1) Dimension and Dimension table
2) Fact table
3) Data Cube
4) Cuboids

Dimension: Dimensions are entities with respect to which an organization wants to keep records. For example, if we consider the sales of a company, **time, item** and **location** are dimensions.

Dimension table: Every dimension has some attributes associated with it. These attributes describe the dimensions. For example, dimension table for **item** include, attributes **item name, brand,** and **type.**

Fact table: Facts are quantities by which we want to analyze relationships between dimensions

Fact table has two components,
1) **Measures** (For Example: Number of items sold)
2) **Keys to each of the related dimension tables** (For example: Item_key, Time_key, Location_key)

Data cube: A data warehouse is based on a multidimensional data model which views data in the form of a **data cube**

Cuboids: Each possible subset of a given dimensions.
For example, we have 3 dimensions **time**, **item** and **location.** Then the cuboids will be {{item, time}, {time, location}, {time} etc…}

Que 5: What is a multidimensional data model?

In data warehousing the data cube is *n*-dimensional. To gain a better understanding of data cubes and the multidimensional data model, let's start by looking at a simple 2-D data cube that is, in fact, a table or spreadsheet for sales data.

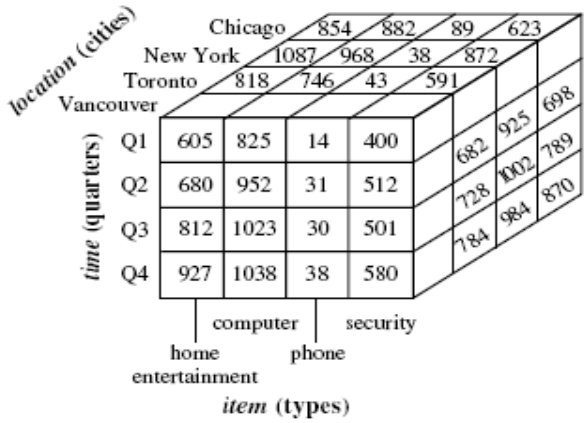
A 2-D view of sales data for *AllElectronics* according to the dimensions *time* and *item*, where the sales are from branches located in the city of Vancouver. The measure displayed is *dollars_sold* (in thousands).

	location = "Vancouver"			
	item (type)			
time (quarter)	home entertainment	computer	phone	security
Q1	605	825	14	400
Q2	680	952	31	512
Q3	812	1023	30	501
Q4	927	1038	38	580

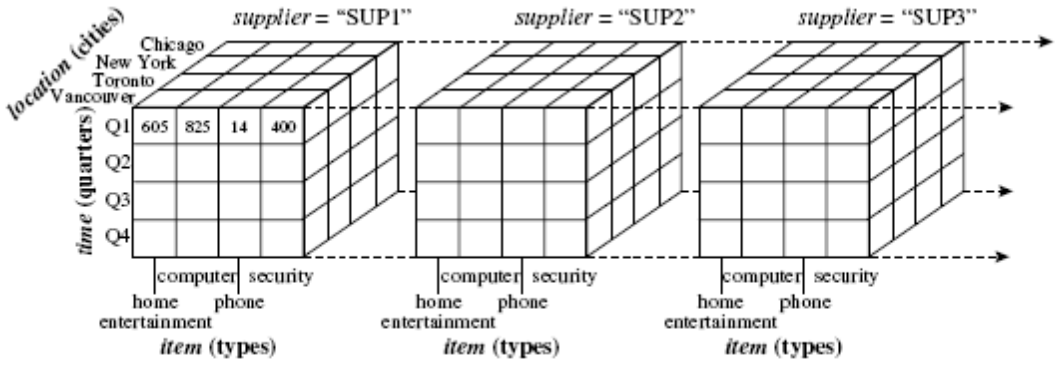
- Now, suppose that we would like to view the sales data with a third dimension. For instance, suppose we would like to view the data according to *time* and *item*, as well as *location* for the cities Chicago, New York, Toronto, and Vancouver. These 3-D data are shown in Table. The 3-D data of Table are represented as a series of 2-D tables.

Table 3.3 A 3-D view of sales data for *AllElectronics*, according to the dimensions *time*, *item*, and *location*. The measure displayed is *dollars_sold* (in thousands).

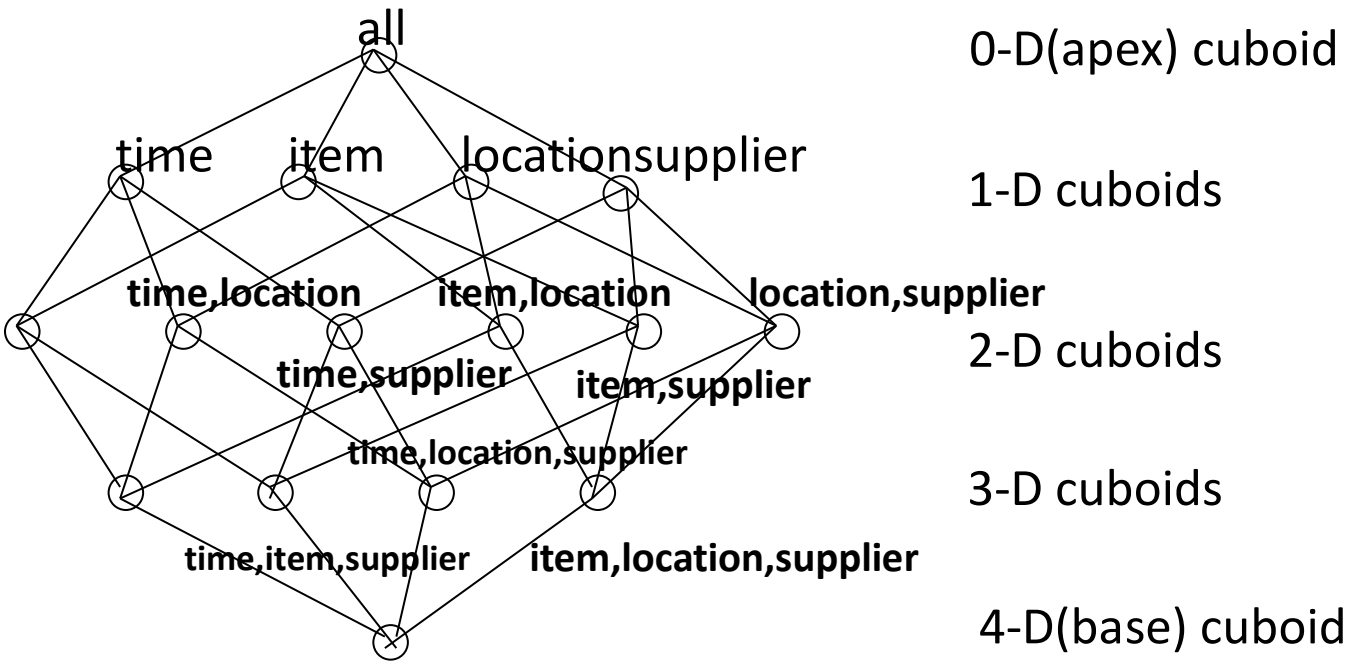
	location = "Chicago"				**location** = "New York"				**location** = "Toronto"				**location** = "Vancouver"			
	item				**item**				**item**				**item**			
time	home ent.	comp.	phone	sec.	home ent.	comp.	phone	sec.	home ent.	comp.	phone	sec.	home ent.	comp.	phone	sec.
Q1	854	882	89	623	1087	968	38	872	818	746	43	591	605	825	14	400
Q2	943	890	64	698	1130	1024	41	925	894	769	52	682	680	952	31	512
Q3	1032	924	59	789	1034	1048	45	1002	940	795	58	728	812	1023	30	501
Q4	1129	992	63	870	1142	1091	54	984	978	864	59	784	927	1038	38	580



- Suppose that we would now like to view our sales data with an additional fourth dimension, such as *supplier*. Viewing things in 4-D becomes tricky. However, we can think of a 4-D cube as being a series of 3-D cubes.



- The 0-D cuboid, which holds the highest level of summarization, is called the **apex cuboid**.
- The cuboid that holds the lowest level of summarization is called the **base cuboid**



Que 6: What do you mean by star, snowflake, and fact constellation schemas for multidimensional databases.

- The entity-relationship data model is commonly used in the design of relational databases.
- In the same way multi dimensional model is used for designing data warehouse. Such a model can exist in the form of a star schema, a snowflake schema, or a fact constellation schema

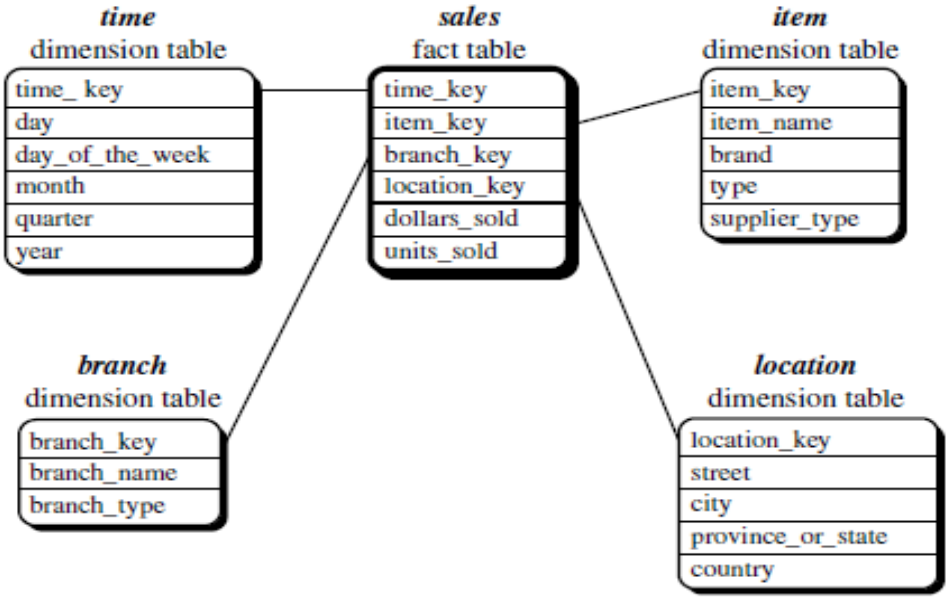
Star schema:

The most common modeling paradigm is the star schema, in which the data warehouse contains.
(1) A large central table (fact table) containing the bulk of the data and
(2) A set of smaller attendant tables (dimension tables), one for each dimension.

Syntax for defining star schema in DMQL (Data mining query language)

define cube sales_star [time, item, branch, location]:
dollars_sold = sum(sales_in_dollars), avg_sales = avg(sales_in_dollars),
units_sold = count(*)

define dimension time **as** (time_key, day, day_of_week, month, quarter, year)
define dimension item **as** (item_key, item_name, brand, type, supplier_type)
define dimension branch **as** (branch_key, branch_name, branch_type)
define dimension location **as** (location_key, street, city, province_or_state, country**)**



Snowflake schema:

The snowflake schema is a variant of the star schema model, where some dimension tables are *normalized*, thereby further splitting the data into additional tables.

Advantages of snowflake schema over star schema:

- In Snowflake schema, data has less redundancy than star schema. This is because, data in snowflake schema is normalized.

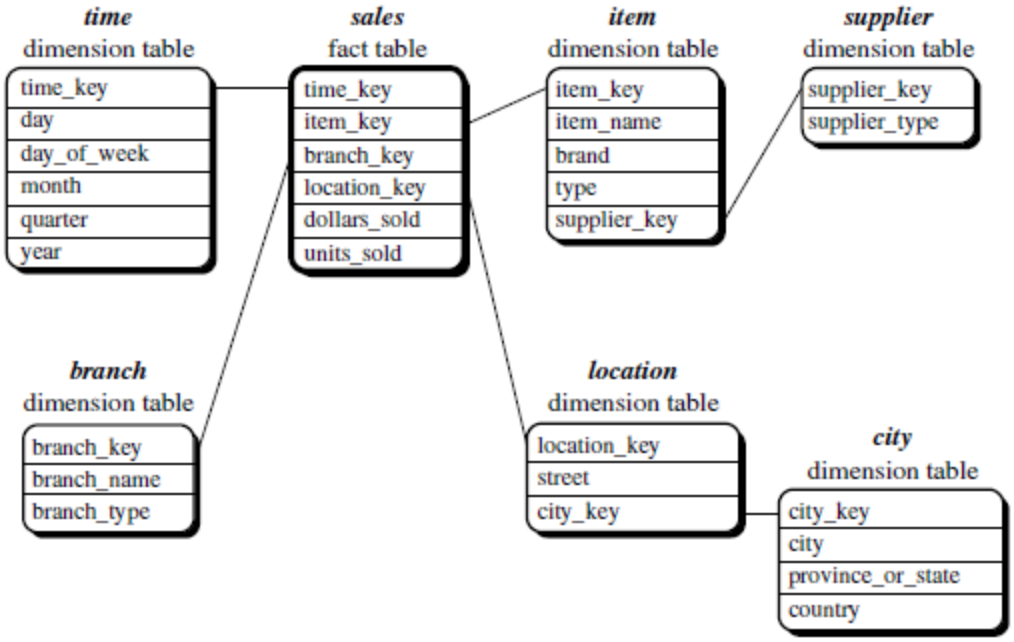
Disadvantages of Snowflake schema over star schema:

- Performance is very less in snowflake schema due to more number of joins as compared to star schema.

Syntax for defining snow flake schema in DMQL (Data mining query language)

define cube sales_snowflake [time, item, branch, location]:
dollars_sold = sum(sales_in_dollars), avg_sales = avg(sales_in_dollars),
units_sold = count(*)

define dimension time **as** (time_key, day, day_of_week, month, quarter, year)
define dimension item **as** (item_key, item_name, brand, type, supplier(supplier_key, supplier_type))
define dimension branch **as** (branch_key, branch_name, branch_type)
define dimension location **as** (location_key, street, city(city_key, province_or_state, country))



Fact constellation:

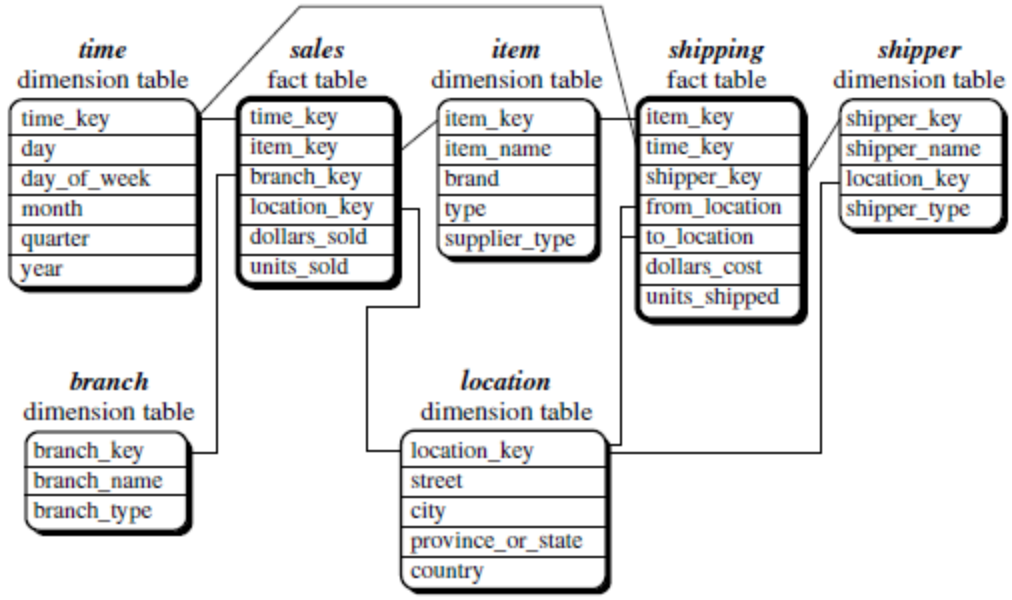
Sophisticated applications may require multiple fact tables to *share* dimension tables. This kind of schema can be viewed as a collection of stars, and hence is called a galaxy schema or a fact constellation.

define cube sales [time, item, branch, location]:
dollars_sold = sum(sales_in_dollars), avg_sales = avg(sales_in_dollars),
units_sold = count(*)

define dimension time **as** (time_key, day, day_of_week, month, quarter, year)
define dimension item **as** (item_key, item_name, brand, type, supplier_type)
define dimension branch **as** (branch_key, branch_name, branch_type)
define dimension location **as** (location_key, street, city, province_or_state, country)

define cube shipping [time, item, shipper, from_location, to_location]:
dollar_cost = sum(cost_in_dollars), unit_shipped = count(*)

define dimension time **as** time in cube sales
define dimension item **as** item in cube sales
define dimension shipper **as** (shipper_key, shipper_name, location as location in cube sales, shipper_type)
define dimension from_location **as** location in cube sales
define dimension to_location **as** location in cube sales



In general, Star and snowflake schema are used to model data marts. And Fact constellation schema is used to model data warehouse.

Note: basic syntax

- Cube Definition (Fact Table)
  define cube <cube_name> [<dimension_list>]: <measure_list>
- Dimension Definition (Dimension Table)
  define dimension <dimension_name> as
  (<attribute_or_subdimension_list>)
- Special Case (Shared Dimension Tables)
  define dimension <dimension_name> as
  <dimension_name_first_time> in cube <cube_name_first_time>

Que 7: What do you mean by measures of data cube? What are three categories of measures?

Measure: A data cube measure is a numerical function that can be evaluated at each point in the data cube space. (It is the value in each cell in a data cube)

- **Distributive:** if the result derived by applying the function to *n* aggregate values is the same as that derived by applying the function on all the data without partitioning E.g., count(), sum(), min(), max()

- **Algebraic:** if it can be computed by an algebraic function with *M* arguments (where *M* is a bounded integer), each of which is obtained by applying a distributive aggregate function E.g., avg(), min_N(), standard_deviation()

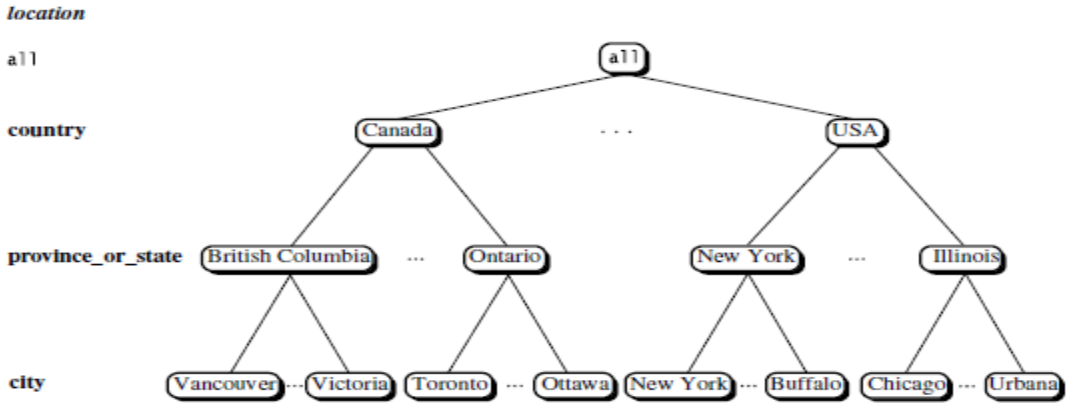
- **Holistic:** if there is no constant bound on the storage size needed to describe sub aggregate. E.g., rank()

Que 8: What do you mean by concept hierarchy? What are the basic types of concept hierarchies?

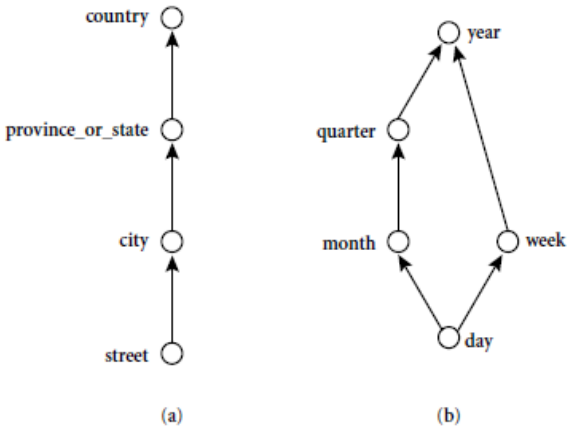
A concept hierarchy defines a sequence of mappings from a set of low-level concepts to higher-level, more general concepts.

Schema hierarchy: A concept hierarchy that is a total or partial order among attributes in a database schema is called a schema hierarchy. Schema hierarchy may formally express existing relationship between attributes.

.

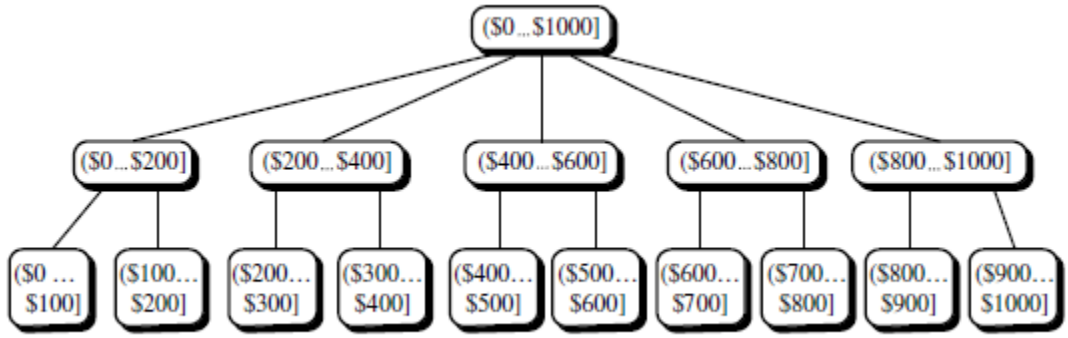


A concept hierarchy for the dimension *location*



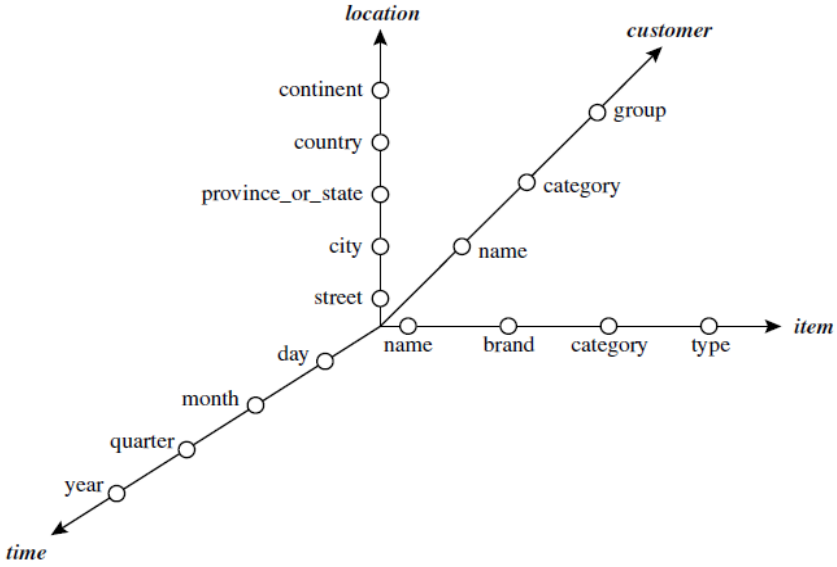
Hierarchical and lattice structures of attributes in warehouse dimensions: (a) a hierarchy for *location*; (b) a lattice for *time*.

Set-grouping hierarchy: Concept hierarchies may also be defined by discretizing or grouping values for a given dimension or attribute, resulting in a set-grouping hierarchy. A total or partial order can be defined among groups of values.



Que 9: Explain a starnet query model for querying multidimensional databases

A starnet model consists of radial lines emanating from a central point, where each line represents a concept hierarchy for a dimension. Each abstraction level in the hierarchy is called a footprint. These represent the granularities available for use by OLAP operations such as drill-down and roll-up.



Que 10: what are the OLAP operations in the multi-dimensional data model?

Roll-up: The roll-up operation (also called the *drill-up* operation by some vendors) performs aggregation on a data cube, either by *climbing up a concept hierarchy* for a dimension or by *dimension reduction.*

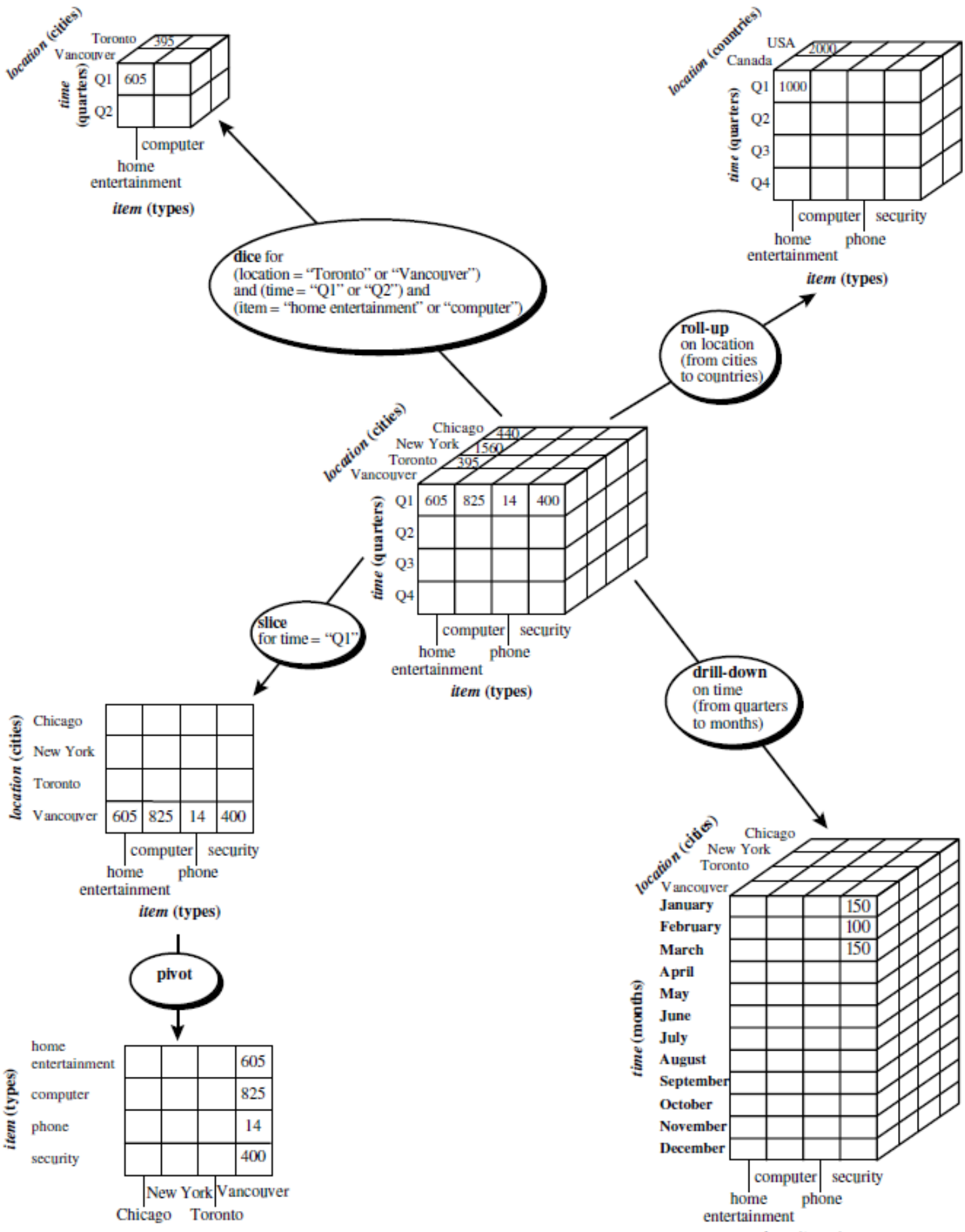
- Ex:roll-up operation aggregates data by ascending the location hierarchy from the level of city to the level of country

Drill-down : It is the reverse of roll-up. It navigates from less detailed data to more detailed data. Drill-down can be realized by either *stepping down a concept hierarchy* for a dimension or *introducing additional dimensions*.

- Ex: drill-down for time
- day<month<quarter<year
- form the level of quarter to the more detailed level of month

Slice: a selection on one dimension of the cube resulting in subcube
Ex: sale data are selected for dimension time using time =Q1

Dice: defines a subcube by performing a selection on two or more dimensions
Ex: a dice opp. Based on
location="toronto" or "vencover" and
time =Q1 or Q2 and
item = "home entertainment" or "computer"


location (cities)
Toronto
Vancouver
395
time (quarters)
Q1
605
Q2
computer
home entertainment
item (types)
location (countries)
USA
Canada
2000
Q1
1000
Q2
Q3
Q4
time (quarters)
computer
security
home entertainment
phone
item (types)
dice for
(location = "Toronto" or "Vancouver")
and (time = "Q1" or "Q2") and
(item = "home entertainment" or "computer")
roll-up
on location
(from cities
to countries)
location (cities)
Chicago
New York
Toronto
Vancouver
440
1560
395
Q1
605
825
14
400
Q2
Q3
Q4
time (quarters)
computer
security
home entertainment
phone
item (types)
slice
for time = "Q1"
drill-down
on time
(from quarters
to months)
location (cities)
Chicago
New York
Toronto
Vancouver
605
825
14
400
computer
security
home entertainment
phone
item (types)
pivot
item (types)
home entertainment
computer
phone
security
605
825
14
400
New York
Vancouver
Chicago
Toronto
location (cities)
Chicago
New York
Toronto
Vancouver
January
February
March
April
May
June
July
August
September
October
November
December
150
100
150
time (months)
computer
security
home entertainment
phone

DATA WAREHOUSE ARCHITECTURE

Que 11: Explain various steps in design and construction of data warehouse.

Design of Data Warehouse: A Business Analysis Framework

Four views regarding the design of a data warehouse:

- **Top-down view**: Allows selection of the relevant information necessary for the data warehouse
- **Data source view** :Exposes the information being captured, stored, and managed by operational systems
- **Data warehouse view**: Includes fact tables and dimension tables. It represents the information that is stored inside the data warehouse, including pre-calculated totals and counts, as well as information regarding the source, date, and time of origin
- **Business query view** : Sees the perspectives of data in the warehouse from the view of end-user

Data Warehouse Design Process

- Top-down, bottom-up approaches or a combination of both
  - Top-down: Starts with overall design and planning (mature)
  - Bottom-up: Starts with experiments and prototypes (rapid)
- From software engineering point of view
  - Waterfall: structured and systematic analysis at each step before proceeding to the next
  - Spiral: rapid generation of increasingly functional systems, short turnaround time, quick turn around

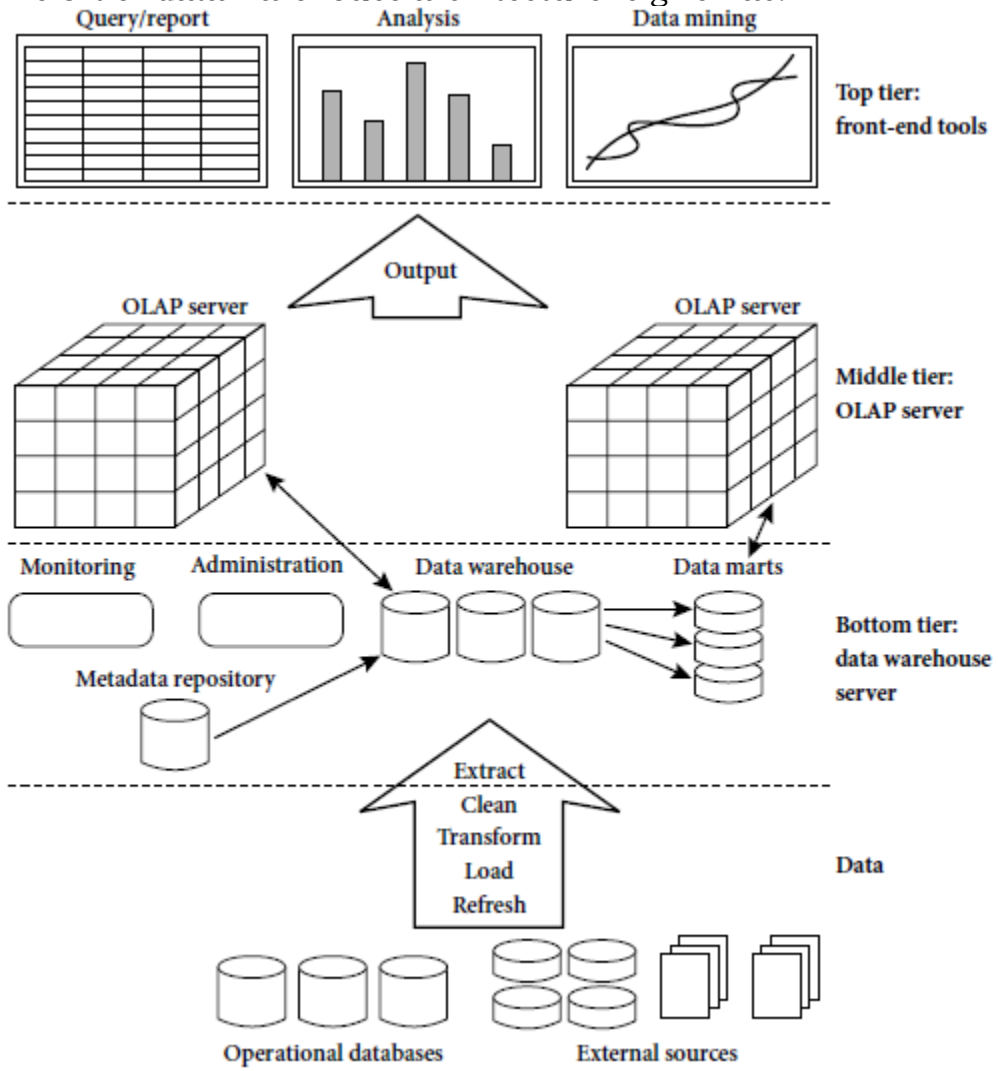
Typical data warehouse design process

- Choose a business process to model, e.g., orders, invoices, etc.
- Choose the *grain* (*atomic level of data*) of the business process
- Choose the dimensions that will apply to each fact table record
- Choose the measure that will populate each fact table record

Que 12: Explain three-tier data warehouse architecture in detail.

Note: This question has very big explanation because, this topic has been asked as essay type question (16 marks) many times.

The 3-tier data warehouse architecture is given as:



Initially data is collected either from tables, flat files or spread sheets. After gathering the data, we apply various backend tools.

Back end tools and utilities

i) Data extraction: It typically gathers data from multiple, heterogeneous, and external sources.
ii) Data cleaning: It is used to remove noise and inconsistent data from raw data.
iii) Data transformation: It converts data from host format to warehouse format.

iv) Load: It sorts, summarizes, consolidates, computes views, checks integrity, and builds indices and partitions.
v) Refresh: It propagates the updates from the data sources to the warehouse

Bottom tier

The bottom tier is a warehouse database server that is almost always a relational database system. The data are extracted using application program interfaces known as gateways. Examples of gateways include ODBC JDBC. This tier also contains a metadata repository, which stores information about the data warehouse and its contents.

From the architecture point of view, there are three **data warehouse models**. They are:

i) Enterprise warehouse: It collects all the information about subjects spanning the entire organization. It contains corporate wide data. It typically contains detailed data as well as summarized data and the size may vary from few GB's to 100's of GB's, terabytes and so on.

ii) Data mart: A data mart contains a subset of corporate wide data that is of value to a specific group of users.

Independent data marts are sourced from data captured from one or more operational systems or external information providers, or from data generated locally within a particular department or geographic area. ***Dependent*** data marts are sourced directly from enterprise data warehouses.

iii) Virtual warehouse: It is a set of views over operational databases.

Meta Data Repository
Meta data are data about data. When used in data warehouse, metadata are the data that defines warehouse objects. A metadata repository should contain the following information:

i) A description of data warehouse structure, which includes data warehouse schema, dimensions, data definitions, data mart locations and contents.
ii) It should also contain whether the data is historical or current data.
iii) The algorithms used for summarization, which includes measure and dimension algorithm definitions.
iv) Business meta data, which includes business terms and definitions.

Middle tier

The middle tier is an OLAP server that is typically implemented using either (1) a relational OLAP (ROLAP) model, that is, an extended relational DBMS that maps operations on multidimensional data to standard relational operations; or

(2) a multidimensional OLAP (MOLAP) model, that is, a special-purpose server that directly implements multidimensional data and operations

There are three types of **OLAP servers.** They are:

i) ROLAP (Relational) server: These are the intermediate servers that stand in between a back-end server ad front end tools. ROLAP technology tends to have high **scalability** than MOLAP.

ii) MOLAP (Multidimensional) server: It provides efficient storage utilization. MOLAP servers have a two-level storage:

a) Denser data sets: They are identified and stored as array structures.

b) Sparse data sets: It employs compression technology for efficient storage utilization (ie) it stores only non-zero values.

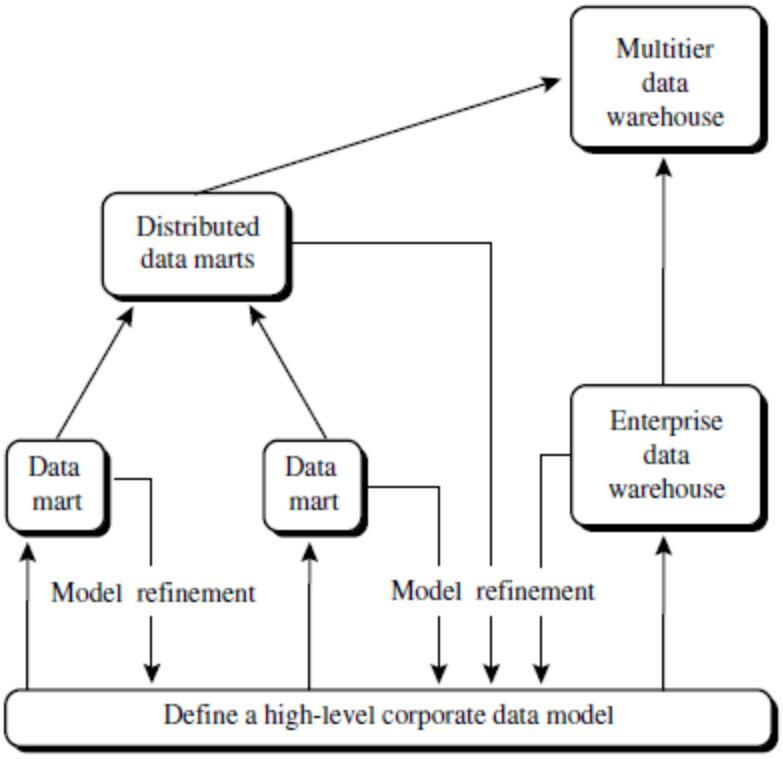
iii) HOLAP (Hybrid) server: It is a combination of both ROLAP and MOLAP. It takes the advantage of scalability in ROLAP and storage utilization from MOLAP. The Microsoft SQL Server 2000 supports a hybrid OLAP server.

iv) Specialized SQL servers: To meet the growing demand of OLAP processing in relational databases, some database system vendors implement specialized SQL servers that provide advanced query language and query processing support for SQL queries over star and snowflake schemas in a read-only environment

Top tier

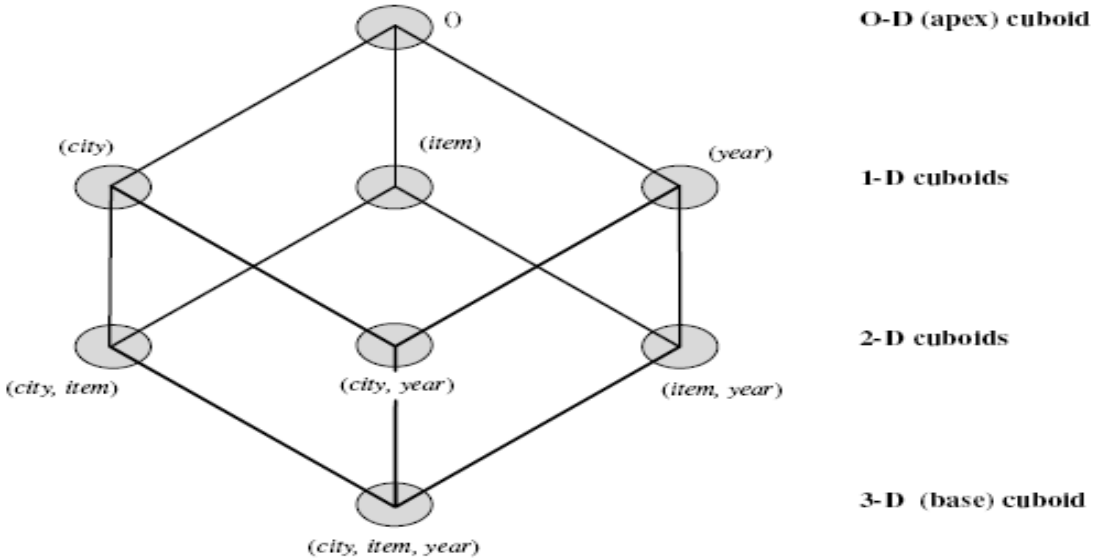
The top tier is a front-end client layer, which contains query and reporting tools, analysis tools, and/or data mining tools.

//optional
First, a high-level corporate data model is defined within a reasonably short period Second, independent data marts can be implemented in parallel with the enterprise warehouse based on the same corporate data model set as above. Third, distributed data marts can be constructed to integrate different data marts via hub servers. Finally, a multitier data warehouse is constructed where the enterprise warehouse is the sole custodian of all warehouse data, which is then distributed to the various dependent data marts.


Multitier data warehouse
Distributed data marts
Data mart
Data mart
Enterprise data warehouse
Model refinement
Model refinement
Define a high-level corporate data model

DATA WAREHOUSE IMPLEMENTATION

Que13 : How to efficiently compute a data cube.



- Taking the three attributes, ***city, item***, and ***year***, as the dimensions for the data cube, and ***sales in dollars*** as the measure, the total number of cuboids, or group by's, that can be computed for this data cube is $2^3 = 8$. The possible group-by's are the following: **(*city, item, year*), (*city, item*), (*city, year*), (*item, year*), (*city*), (*item*), (*year*), ()**, where () means that the group-by is empty (i.e., the dimensions are not grouped).

- The **apex cuboid**, or 0-D cuboid, refers to the case where the group-by is empty. It contains the total sum of all sales.

- The **base cuboid** is the least generalized (most specific) of the cuboids

- If there were no hierarchies associated with each dimension, then the total number of cuboids for an *n*-dimensional data cube, is 2^n. However, in practice, many dimensions do have hierarchies. In this case the numbers of cuboids are:

$$T = \prod_{i=1}^{n} (L_i + 1)$$

Where *Li* is the number of levels associated with dimension *i*. And One is added to *Li* in Equation to include the *virtual* top level, all.

There are three choices for data cube materialization given a base cuboid:

No materialization: Do not precompute any of the "nonbase" cuboids. This leads to computing expensive multidimensional aggregates on the fly, which can be extremely slow.
Full materialization: Precompute all of the cuboids. The resulting lattice of computed cuboids is referred to as the *full cube*. This choice typically requires huge amounts of memory space in order to store all of the precomputed cuboids.
Partial materialization: Selectively compute a proper subset of the whole set of possible cuboids. Partial materialization represents an interesting trade-off between storage space and response time.

Que 14: How to index OLAP Data?

To facilitate efficient data accessing, most data warehouse systems support index structures and materialized views

There are 2 indexing techniques:

- Bit map Indexing
- Join Indexing

In **Bit map Indexing**, The attribute values are converted into attributes. It is especially useful for low-cardinality domains because comparison, join, and aggregation operations are then reduced to bit arithmetic, which substantially reduces the processing time. Bitmap indexing leads to significant reductions in space and I/O.

Example of bit map Indexing:

Base table

RID	*item*	*city*
R1	H	V
R2	C	V
R3	P	V
R4	S	V
R5	H	T
R6	C	T
R7	P	T
R8	S	T

Item bitmap index table

RID	H	C	P	S
R1	1	0	0	0
R2	0	1	0	0
R3	0	0	1	0
R4	0	0	0	1
R5	1	0	0	0
R6	0	1	0	0
R7	0	0	1	0
R8	0	0	0	1

City bitmap index table

RID	V	T
R1	1	0
R2	1	0
R3	1	0
R4	1	0
R5	0	1
R6	0	1
R7	0	1
R8	0	1

Note: H for "home entertainment," C for "computer," P for "phone," S for "security," V for "Vancouver," T for "Toronto."

Join indexing keeps track of the joinable rows of two relations from a relational database without performing costly join operations. Join indexing is especially useful for maintaining the relationship between a foreign key and its matching primary keys, from the joinable relation.

For example, if two relations R (RID, A) and S(B, SID) join on the attributes A and B, then the join index record contains the pair (RID, SID), where RID and SID are record identifiers from the R and S relations, respectively

Join index table for *location/sales*

location	*sales_key*
. . .	. . .
Main Street	T57
Main Street	T238
Main Street	T884
. . .	. . .

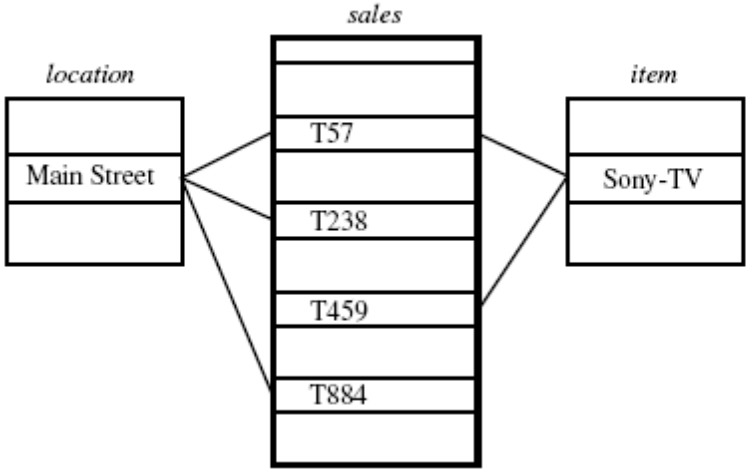
Join index table for *item/sales*

item	*sales_key*
. . .	. . .
Sony-TV	T57
Sony-TV	T459
. . .	. . .

Join index table linking two dimensions *location/item/sales*

location	*item*	*sales_key*
. . .	. . .	. . .
Main Street	Sony-TV	T57
. . .	. . .	. . .

For example, if two relations R (RID, A) and S(B, SID) join on the attributes A and B, then the join index record contains the pair (RID, SID), where RID and SID are record identifiers from the R and S relations, respectively



Que 15: Write a short note on "Efficient processing of OLAP queries".

The purpose of materializing cuboids and constructing OLAP index structures is to speed up query processing in data cubes.

1. **Determine which operations should be performed on the available cuboids:** This involves transforming any selection, projection, roll-up (group-by), and drill-down operations specified in the query into corresponding SQL and/or OLAP operations.

2. **Determine to which materialized cuboid(s) the relevant operations should be applied**: This involves identifying all of the materialized cuboids that may potentially be used to answer the query and estimating the costs of using the remaining materialized cuboids, and selecting the cuboid with the least cost.

Let the query to be processed be on {brand, province_or_state} with the condition "year = 2004", and there are 4 materialized cuboids available:

1) {year, item_name, city}
2) {year, brand, country}
3) {year, brand, province_or_state}
4) {item_name, province_or_state} where year = 2004

Which should be selected to process the query?

If Indexing is good then cuboid 3 may be good and if indexing is not good cuboid 4 should be selected.

Data Mining

Que 16. Explain the applications of data warehouse

There are three kinds of data warehouse applications

Information processing: supports querying, basic statistical analysis, and reporting using crosstabs, tables, charts, or graphs.

Analytical processing: supports basic OLAP operations, including slice-and-dice, drill-down, roll-up, and pivoting

Data mining: supports knowledge discovery by finding hidden patterns and associations, performing classification and prediction etc..

Que 17: Difference between OLAP and Data Mining

Data Mining	OLAP
Data mining allows the automated discovery of patterns and interesting knowledge hidden in large amounts of data.	OLAP is a data summarization/aggregation tool that helps simplify data analysis,
Data mining tools is to automate as much of the process as possible	OLAP tools are targeted toward simplifying and supporting interactive data analysis,
Data mining covers association, classification, prediction, clustering, time-series analysis, and other data analysis tasks.	OLAP systems can present general descriptions of data from data warehouses,
Data mining analyze transactional, spatial, textual, and multimedia data that are difficult to model with current multidimensional database technology	Can handle data in multidimensional databases.

Que 18: What is OLAM? How does it integrate with OLAP?
(Or)
What are the advantages of using Data in Data warehouse for data mining?

On-line analytical mining (OLAM) integrates on-line analytical processing (OLAP) with data mining and mining knowledge in multidimensional databases.

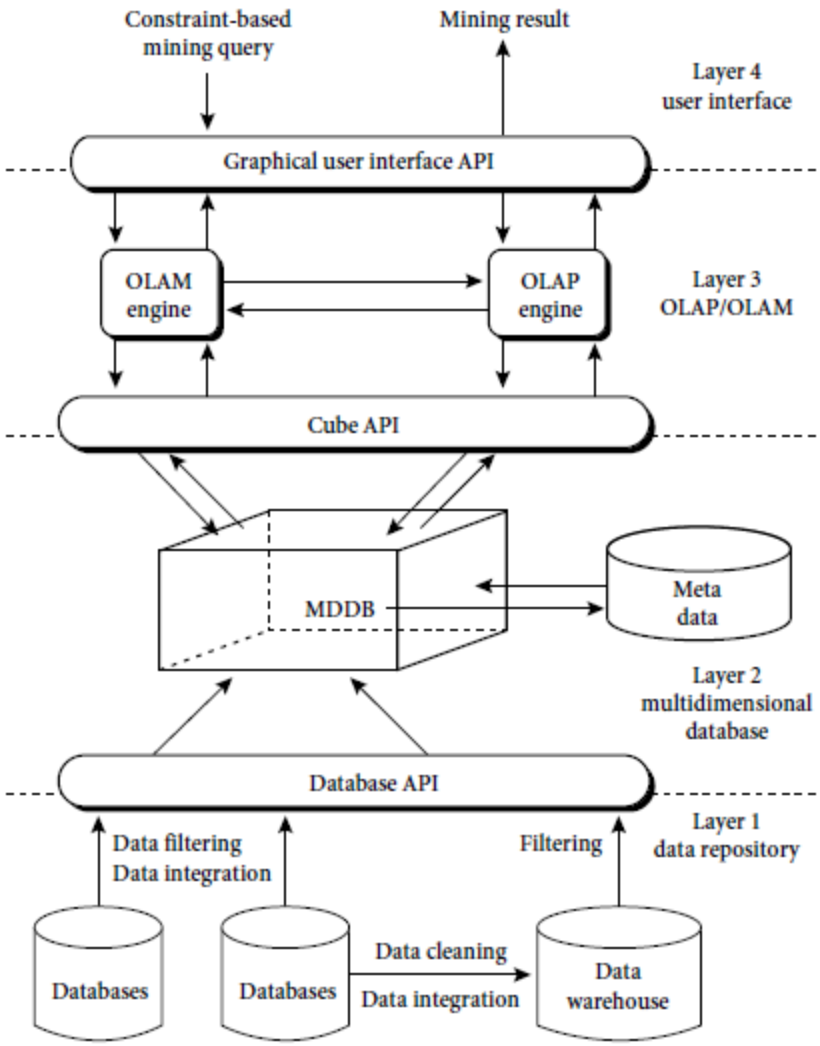
High quality of data in data warehouses: Most data mining tools need to work on integrated, consistent, and cleaned data, which requires costly data cleaning, data integration and data transformation as preprocessing steps. A data warehouse constructed by such preprocessing serves as a valuable source of high quality data for OLAP as well as for data mining

Available information processing infrastructure surrounding data warehouses: Comprehensive information processing and data analysis infrastructures have been surrounding data warehouses, which include accessing, integration, consolidation, and transformation of multiple heterogeneous databases. It is good to make the best use of the available infrastructures rather than constructing everything from scratch.

OLAP-based exploratory data analysis On-line analytical mining provides facilities for data mining on different subsets of data and at different levels of abstraction, by drilling, pivoting, filtering, dicing, and slicing on a data cube and on some intermediate data mining results. This, together with data/knowledge visualization tools, will greatly enhance the power and flexibility of exploratory data mining.

On-line selection of data mining functions: Provides users with the flexibility to select desired data mining functions and swap data mining tasks dynamically.

Que 19: Explain OLAM Architecture.



OLAM and OLAP servers both accept user on-line queries (or commands) via a graphical user interface API and work with the data cube in the data analysis via a cube API. A metadata directory is used to guide the access of the data cube. The data cube can be constructed by accessing and/or integrating multiple databases via an MDDB API and/or by filtering a data warehouse via a database API that may support OLE DB or ODBC connections.

Unit 2

What is data mining? What are its applications?

Ans: Data mining is a process of discovering useful knowledge from large data stores.

Or

It is a technique used for extraction of new and useful knowledge from large data store for decision making purposes.

Applications of data mining:

Business:

Modern business uses data mining to perform market analysis to identify new product bundles, finding the root cause of manufacturing problems, to prevent customer wearing away and acquire new customers, cross-sell to existing customers, to determine sales trends etc.…

Data mining also contributes in customer relationship management Rather than randomly contacting a prospect or customer through a call center or sending mail, a company can concentrate its efforts on predicting customers who may be interested to use the offers.

Medicine, Science and engineering:

Many satellites gather huge amount of information on weather conditions. Data mining may be used to answer many question as volcano detection, earthquake detection, relation between ocean's surface temperature and land temperature etc…

Data mining is also used in micro biology, for example it aims to find out how the changes in an individual's DNA sequence affects the risks of developing common diseases such as cancer, which is of great importance to improving methods of diagnosing, preventing, and treating these diseases.

Another example of data mining in science and engineering is found in educational research, where data mining has been used to study the factors leading students to choose to engage in behaviors which reduce their learning, and to understand factors influencing university student retention.

Data mining is used in many more applications like banking, traffic analysis, computer network analysis etc…

Motivating challenges of data mining

Ans: The following are the motivating challenges of the specific challenges that motivated the development of data mining.

1) **Scalability**: "The amount of time required to extract the knowledge proportional to the amount of data". Data mining algorithm need to handle huge amount of data. It is required that data mining should be scalable. Data mining algorithms use new data structures to handle different kinds of data. In some cases DM uses out of memory techniques to handle large data.

2) **High Dimensionality**: Traditional data analysis techniques cannot handle high dimensional data. "Dimensions may be considered as attributes or properties of a data in some cases." It is now common to deal with hundreds of attributes. Computational cost also increases with respect to the number of dimensions.

3) **Heterogeneous and complex data**: Traditional method often deals with data of similar type. But complex data are often encountered in today's world. For example, web pages contain semi-structured information. Climatic data contain time series data of various earth locations. Data mining should be able to deal with heterogeneous and complex data by considering relationship among data, such as parent child relationships etc...

4) **Data ownerships and distribution**: Data is stored in different locations and owned by different people. Distributed data mining deals with following challenges.
   A. Reduction of communication cost to perform distributed mining.
   B. Consolidation of DM results from various sources.
   C. Security issues.

5) **Nontraditional analysis**: In traditional statistics, a hypothesis was proposed and an experiment is designed to gather the data and then the data is analyzed with respect to hypothesis. But, in data mining many hypothesis are generated and data mining techniques are used to automate the process of hypothesis generation.

Origin of Data Mining

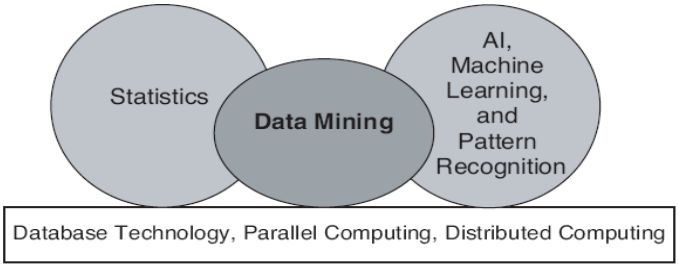


Figure 1.2. Data mining as a confluence of many disciplines.

Data mining draws idea such as

1) Sampling, estimation and hypothesis testing from **statistics**
2) Search algorithms, modeling techniques and learning from **Artificial intelligence, machine learning, pattern recognition.**
3) Data mining has adopted ideas from **evolutionary computing, information theory, signal processing, visualization, information retrieval etc...**
4) **DBMS** to provide storage, query processing and indexing.
5) **Parallel computing** to provide addressing massive size of data.
6) **Distributed computing** is essential when data cannot be gathered in a single place.

Data mining tasks

There are two categories of data mining:

Predictive tasks: Predicting the value of an attribute based on values of another attribute is predictive task. The value to be predicted is known as **target variable**. And, the values used for making predictions are known as **explanatory variables**.

Descriptive tasks: It characterizes the general properties of a target class of data in database.

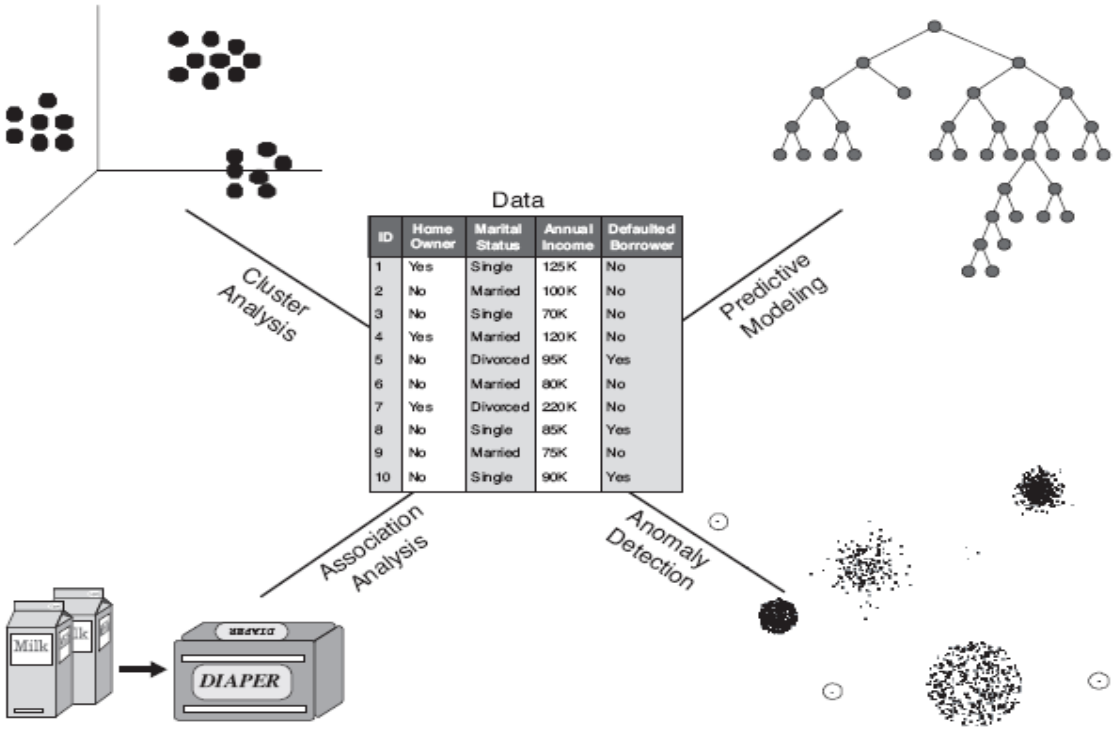
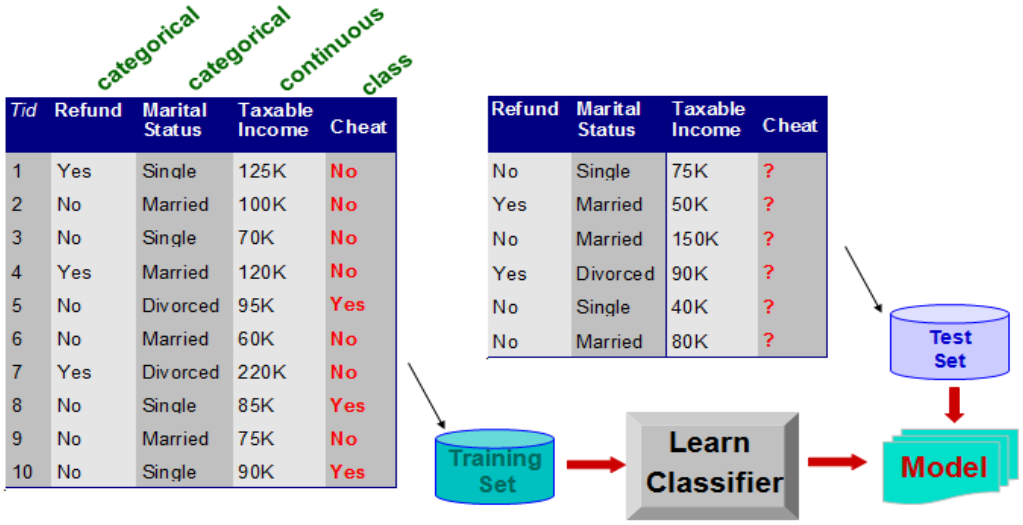


Figure 1.3. Four of the core data mining tasks.

1) **Predictive modeling**: Process of building a model for the target variable based on explanatory variable. There are 2 types of predictive modeling tasks 1) classification 2) Regression.

   Classification: Classification is the process of finding a set of models which are used to predict the class of object whose class label is unknown. It is used for discreet target variables. Example: Predicting whether web user will buy a product online or not.



Regression: Regression is often used to predict the missing values rather than class labels. It is used for continues target variables. Example: Predicting the price of a gold or stock.

Example: Suppose, as sales manager of AllElectronics, you would like to classify a large set of items in the store, based on three kinds of responses to a sales campaign: good response, mild response, and no response. You would like to derive a model for each of these three classes based on the descriptive features of the items, such as price, brand, place made, type, and category. The resulting classification should maximally distinguish each class from the others, presenting an organized picture of the data set.

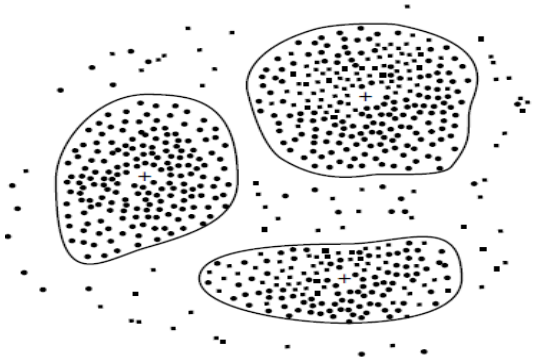
2) **Association analysis**: It is used to discover strongly associated features in data. The discovered patterns are represented in the form of association rules. Applications of association analysis include, identifying web pages accessed together, identifying items bought together in a super market etc…

TID	*Items*
1	Bread, Coke, Milk
2	Beer, Bread
3	Beer, Coke, Diaper, Milk
4	Beer, Bread, Diaper, Milk
5	Coke, Diaper, Milk

Example: Consider the supermarket billing database. Association analysis can be used to find the list of items bought together in a supermarket. For example coke and milk are bought together which is represented as (coke milk)

3) **Cluster analysis:** It is a process of grouping data into classes or clusters so that objects within a cluster have high similarity in comparison to one another, but are very dissimilar to objects in other clusters.

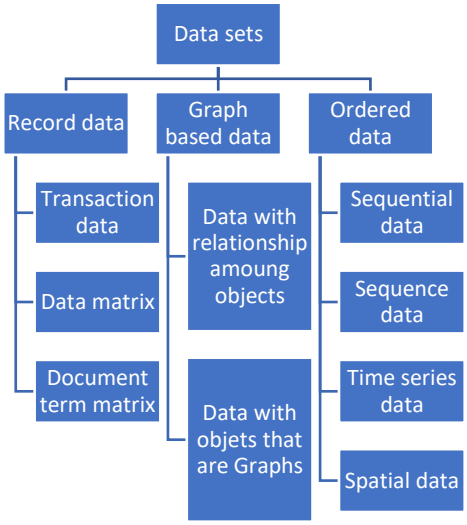
Cluster analysis can be performed on *AllElectronics* customer data in order to identify homogeneous subpopulations of customers. These clusters may represent individual target groups for marketing. Figure shows a 2-D plot of customers with respect to customer locations in a city. Three clusters of data points are evident



4) **Anomaly detection**: Detection of observations that are different from rest of the data is known as anomaly detection or outliers.

Example: Anomaly detection may uncover fraudulent usage of credit cards by detecting purchases of extremely large amounts for a given account number in comparison to regular charges incurred by the same account. Outlier values may also be detected with respect to the location and type of purchase, or the purchase frequency.

Types of data sets



:

Record data:

Record data is a collection of records each of which consists of a fixed set of attributes. Record data is usually stored in flat files or relational databases.

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

1) **Transaction data**: A special type of record data, where each record transaction) involves a set of items. For example, consider a grocery store. The set of items purchased by a customer during one shopping trip constitute a transaction. This type of data is known as market basket analysis.

TID	*Items*
1	Bread, Coke, Milk
2	Beer, Bread
3	Beer, Coke, Diaper, Milk
4	Beer, Bread, Diaper, Milk
5	Coke, Diaper, Milk

2) **Data matrix**: If data objects have the same fixed set of numeric attributes, then the data objects can be thought of as points in a multi-dimensional space. Such data set can be represented by an m by n matrix, where there are m rows, and n columns, one for each attribute. This type of matrix is known as data matrix. Standard matrix operation may be applied on it.

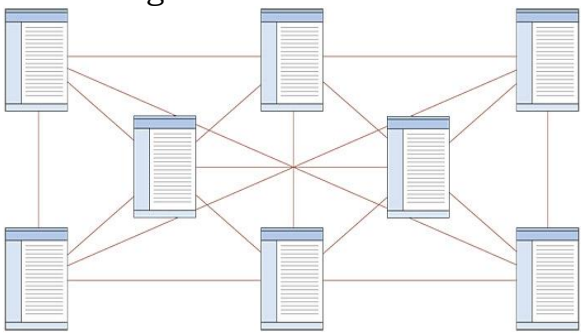
Projection of x Load	Projection of y load	Distance	Load	Thickness
10.23	5.27	15.22	2.7	1.2
12.65	6.25	16.22	2.2	1.1

3) **Sparse data matrix:** It's a special case of data matrix where only non zero values are important. Consider the below example where each document is represented as term vector. And term is attributes. In document 1 ,word (team) is found 3 times, Word(play) is found 5 times and so on.

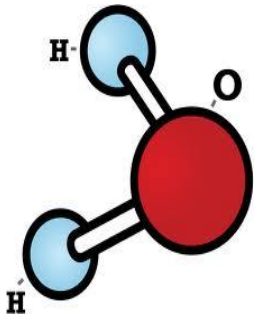
	team	coach	play	ball	score	game	win	lost	timeout	season
Document 1	3	0	5	0	2	6	0	2	0	2
Document 2	0	7	0	2	1	0	0	3	0	0
Document 3	0	1	0	0	1	2	2	0	3	0

Graph based data:

1) **Data with relationships among data objects:** Graphs are used to represent relationship between data objects. The data object can be considered as node and relationship between the nodes can be represented by links. Consider an example of WWW where web pages are interlinked with other pages. Links provide good information to search and extract relevant WebPages.



2) **Data with objects that are graphs**. If objects contain subobjects that have relationship among them, then objects are represented by graphs. For example consider water molecule(object) which can be represented as:



Ordered data

When attributes have relationships in terms of time or space, then that data is known as ordered data.

Sequential data (temporal data): It is an extension of record data with time associated with each record.

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

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

Consider the customers data maintained by an online store. We can find some interesting patterns from the above data, For example: customers who bought a mobile phone are more likely to buy mobile case,& screen guard in near future

Sequence data: It is a data set which contains a sequence of actions, such as sequence of words or letters. For Example: The DNA sequence of two people may be analyzed to know the relationship between them. Below is the human genetic code using four nucleotides.

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

Time series data: It is used to store sequence of data that changes over time. For example the value of stock of a company which changes with respect to time

Spatial data: It contains spatial related information like geographic maps data, VLSI chip design data, Medical and satellite image data etc…Its main application is in weather data. For example climate of mountain areas located at various altitude. An important aspect of spatial data is spatial auto correlation; that is two points on earth that are close to each other usually have similar temperature and rainfall.

Data Quality

Ans: Quality of data plays an important role in data mining. We can get quality knowledge from a good quality data. The quality of data is not perfect due to following reasons:

Human errors during data entry

Hardware limitations

Flaws in data collection process

Values or entire object may be missing

There may be duplicate objects.

Inconsistency

Etc…

Definitions:

Error: For continuous attribute, the numerical difference of measured and true value is called error.

Measurement Errors: It refers to any problem resulting from the measurement process.

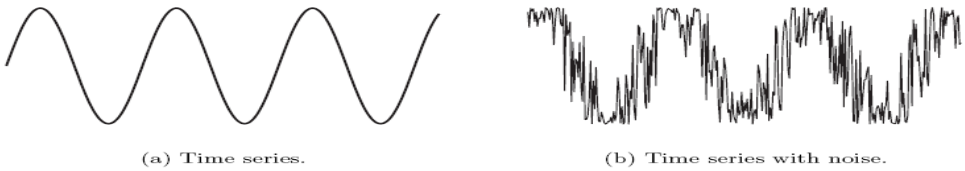
Data collection error: It refers to errors such as omitting data objects or attribute values.

VARIOUS ASPECTS OF DATA QUALITY RELATED TO DATA MEASUREMENT AND COLLECTION:

Noise and artifacts:

Noise is a random error or variance in a measured variable. It may involve the distortion of a value or the addition of false object. Noise is generally associated with temporal and time series data. Image processing techniques are used to reduced the noise (Removing noise is difficult). Data mining algorithms should be robust (tolerate noise) to deal with noise.

Artifacts: Deterministic distortion of data is often referred as artifacts.
Or
A substance or structure not naturally present in the data being observe d but formed by artificial means. For example: a mark on a set of photographs.



(a) Time series. (b) Time series with noise.

Figure 2.5. Noise in a time series context.

(a) Three groups of points. (b) With noise points (+) added.

Figure 2.6. Noise in a spatial context.

Precision, Bias and accuracy:

Precision: The closeness of repeated measurements (of the same quantity) to one another. The precision is generally measured by standard deviation

Bias: A systematic variation of measurements from quantity being measured. The bias is generally measured by mean.

Accuracy: The closeness of measurements to the true value of the quantity being measured.

Significant digit: The Nearest acceptable precision value of the digit is known as significant digit.+_ 0.05.

Outliers: A data set may contain data objects that do not comply with the general behavior or model of the data. This data objects are outliers. Most data mining algorithm methods discard outliers as noise or exceptions. However, in some applications such as fraud detection, the rare event is more important.

Missing values

Missing values are often encountered in data sets. This may due to incomplete information collection.

Missing values are handled in the following ways:

Eliminate data objects or attributes: This method is not very effective, unless the tuple contains several attributes with missing values. A Related strategy is to eliminate attributes that have missing values.

Estimate missing values: In some cases the missing values can be estimated my using the other values. For continuous attributes, missing values can be estimated by using the average of the remaining values. For categorical attribute, missing value can also be estimated using the most common values.

Ignore missing values during analysis: Data mining algorithms should be robust and be able to deal with missing values. In large data sets having some missing values may not affect data mining results.

Inconsistent values (conflicting):

There may be inconsistencies in the data recorded for some transaction. For example: while entering some personal data the pin code may not match with name of the city. Height of a person may be negative. Height of person does not match with his weight. This type of inconsistency may be reduced by using some checks. But removing inconsistency may need heavy data instead.

Duplicate data:

There may be many duplicate values in a data. There are two main issues. First, same data object may be represented by two different objects (different e-mail address for same person).Second, accidentally combining two different objects.(2 people may have same name).The term duplication is often used to refer to the process of dealing with this issues.

Data Preprocessing

What is data preprocessing?

It is a process of converting the data into appropriate and clear form for data mining. The input data should be of good quality, because the quality of knowledge provided by data mining is proportional to the input data.

Aggregation

Ans: Combining of 2 or more objects is known as aggregation. The large data set Quantitative attributes may be aggregated by either taking sum or mean of attribute values.

Advantages of aggregation

- Aggregation makes data smaller, smaller data sets require less processing time.
- It can provide higher view of data i.e. Data can be viewed at higher level of abstraction (summarized data can be viewed).
- Attribute at higher level of abstraction have less variability than attribute at lower level of abstraction.

Consider an example:

Below example records the sales in 3 branches of a supermarket (day wise report).

TID	Item	Store location	Date	price
1001	Colgate tooth brush	Bhimavaram	1/3/2013	30
1002	Amul Butter	Tanuku	2/4/2013	20
1003	hp Computer mouse	Bhimavaram	2/4/2013	200
1004	Pepsodent toothpaste	Tanuku	3/4/2013	50
1005	Night lamp	Tanuku	3/3/2013	100
1006	Thums up	Bhimavaram	4/4/2013	65

Fig:1

Location	Month	Sales
Bhimavaram	March	98.3
Tanuku	April	56.6

Fig :2

Fig1 represents the original data set and fig 2 represents the aggregated data set. TID and Items are removed and daily sales are converted into monthly sales.

Sampling Technique

Ans: This technique is inherited by data mining from statistics.

It is very expensive and time consuming to process all the data. Sampling technique can be used to consider only a subset of data inserted of whole data for analysis.

A Sample is said to be repetitive if it has the same properties as original data.

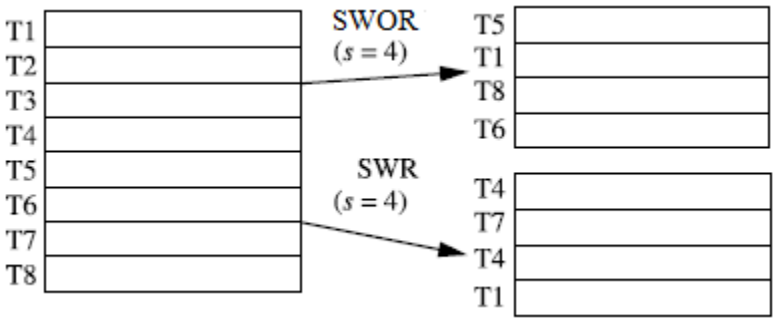
Sampling Approaches

There are mainly three techniques for sampling:

- Sampling without replacement (SWOR)
- Sampling with replacement(SWR)
- Stratified sampling

Sampling without replacement (SWOR): An item once selected from the data set, it is removed from the original population (data set).Consider the below example: No item is sampled more than once.

Sampling with replacement (SWR): An item once selected from the data set, it is again kept in the same place constituting original population. Means, a same item may be sampled more than once. Consider the example below. Item T4 is sampled 2 times.



Stratified sampling: The number of objects drawn from each group is proportional to the size of the group in original population. Consider the below example. The ratio of age groups in the original population is same as that of the samples. (4:8:2) is proportional to (2:4:1).

Stratified sample
(according to *age*)

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

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

Problems with sampling:

- If the sample size is less then, some important patterns may be missed. And if, sample size is more then they eliminate the advantages of sampling (i.e. less time consuming and less storage space)

Progressive sample

Progressive sample method is used to determine the sufficient size of the sample. This approach starts with a small sample, and then increase the sample size until a sample of sufficient size has been obtained.

The correct sample size can be found in the following way:

The accuracy of predictive model increases with respect to the size of sample. At a point the accuracy doesn't increase. This point is known as leveling off point. Another sample is considered from the same original data and increase the small sample to the same size. Now, the closeness of this with the leveling point is measured.

Dimensionality reduction

Ans: Dimensionality reduction refers to the creation of new attributes that are combination of the old attributes.

Advantages of dimensionality reduction

1) Dimensionality reduction eliminates irrelevant attributes and reduces noise in the data
2) Many Data mining algorithm work better with data having less number of dimensions (attributes).
3) Reduction of dimensions leads to a more understandable data model.
4) Reduction of dimensions allows data to be visualized easily.
5) Time and memory required by the data mining algorithm is reduced with reduction in dimensionality.

Curse of dimensionality (disadvantages of having more dimensions in data)

1) Data analysis becomes difficult as the dimensionality of data increases.
2) If data is having more number of dimensions, it is very difficult to create a classification model due to fewer objects.
3) In the case of clustering, the density and distance between the objects would be more. This makes clustering difficult.

Feature subset selection

Ans: In feature subset selection, only a subset of all dimensions is used. This is specially used when there are large numbers of redundant and irrelevant dimensions in the dataset.

Note:

Redundant features: Has same information (same values) in more than one dimension.

Irrelevant features: Has dimensions irrelevant to data mining tasks.

Techniques for eliminating irrelevant and redundant features:

1) **Common sense**: Some irrelevant and redundant dimensions can be removed using common sense or having a sound command on domain.

2) **Brute-force approach**: Try all possible feature subsets as input to data mining algorithm and select the subset which produces best results. But this method will not work if numbers of attributes (features) are more.

3) **Embedded approaches**: Feature selection occurs naturally as part of the data mining algorithm. The algorithm will decide which features to include and which to ignore.
4) **Filter approaches:** attributes (features) are selected before data mining algorithm is run using some independent approaches.
5) **Wrapper approaches**: The data mining algorithm itself is used to determine the attribute subset.

Feature Creation

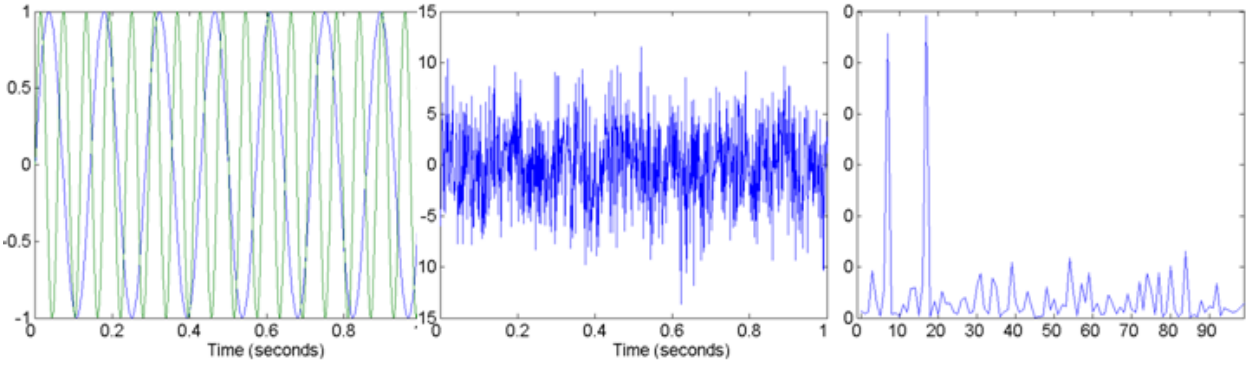
Ans: Feature creation is a process of creating new set of attributes that can capture information more efficiently than the original attributes.

The three methodologies for creating new attributes are:

1) Feature extraction.
2) Mapping the data to a new space
3) Feature construction.

Feature extraction: The creation of new attributes from original raw data is known as Feature extraction. This method is mainly used in image processing. Consider a set of images stored in the form of pixels, we want to classify them as containing human faces or not. If new attributes are created containing information about certain edges and colors, then this attributes helps us to better classify these images.

Mapping the data to a new space: It is the process of viewing the data in different angles to reveal important and interesting features in it. This method is mainly used in Time series data. Consider the below example. Fig 1 contains two time series data with out noise. And fig (2) contains two time series data with noise. By Appling Fourier transformation, time series data in fig (2) is converted into frequency information presented in fig (3).



Fig(1) Two Sine Waves

Fig(2)Two Sine Waves + Noise

Fig(3) Frequency

Feature construction: Sometimes the attributes in the original data sets have the necessary information, but still these attributes are not suitable for data mining algorithm. In this situation, New attributes are constructed from the original attributes which as more suitable for data mining algorithm.

Example: Consider an original data set containing mass and volume information of various metals. In this case it is more meaningful to construct density information (density =mass/volume) rather than mass and volume.

Metal	Mass	volume
Metal A	16.23	15.68
Metal B	17.89	14.34
Metal C	18.33	13.67

Fig A

Metal	Density
Metal A	1.0350
Metal B	1.2475
Metal C	1.3408

Fig B

Discretization and Binarization

Binarization: The process of converting continuous and discrete attributes into binary attributes is known as binarization.

The two techniques for binarization:

1) If there are m categorical values, then uniquely assign each ordinal value to an integer (0,m-1). Then convert these integers into binary numbers

In the below example there are 5 categorical values (awful, poor, ok, good, great), the uniquely assign each value to an integer (awful=0, poor=1, ok=2, good=3, great=4). Then convert these integers into binary numbers.

Categorical value	Integer value	X1	X2	X3
Awful	0	0	0	0
Poor	1	0	0	1
OK	2	0	1	0
Good	3	0	1	1
Great	4	1	0	0

Note: If the categorical values are ordinal then values should be stored in a sequence.

2) In this technique only the presence of item is considered. Here the number of binary attributes is equal to the number of categorical values.

Categorical value	Integer value	X1	X2	X3	X4	X5
Awful	0	1	0	0	0	0
Poor	1	0	1	0	0	0
OK	2	0	0	1	0	0
Good	3	0	0	0	1	0
Great	4	0	0	0	0	1

Discretization:

Process of converting continuous attributes into categorical attributes is known as Discretization. This technique is used for the data used for classification and association analysis. Discretization involves two subtasks:

1) How many categorical values to include.
2) How to map continuous attributes to categorical attributes.

For example consider a student table sorted in percentages order

Student name	percentages		grade
A	45.8	→	Second class
B	47.9		
C	66.7	→	First class
D	65.6		
E	62.5		
F	77.8	→	destination
G	80.6		

There are two types of discretization 1) Unsupervised discretization
2) Supervised discretization

Unsupervised discretization: In this type of discretization, domain knowledge (class information) is not used to convert continuous attributes into categorical attributes. Rather, they are discretized using techniques like 1) Equal width 2) Equal depth 3) K means etc...

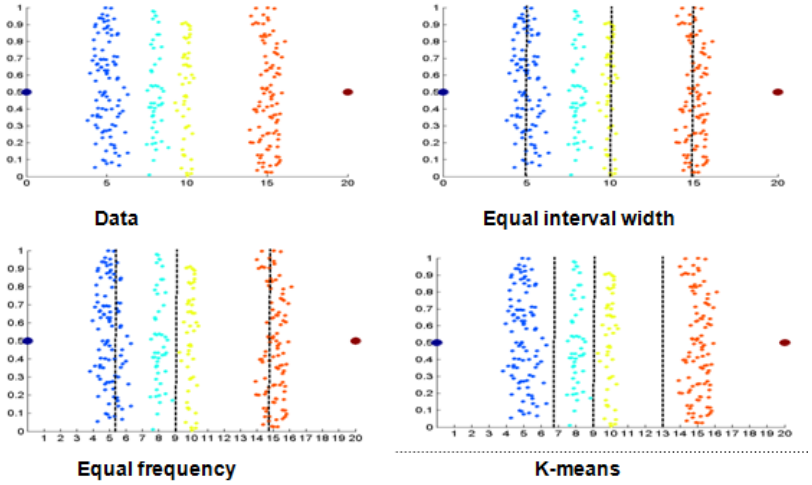
Equal width: It divides the attribute values into equal intervals. Consider a set of attribute values(2,3,4,9,8,15,16,23,26,28,21,22)

Bin1(1-10)-2,3,4,9,8
Bin 2(11-20)-15, 16
Bin3(20-30)-23,26,28,21,22

Equal depth: It divides the attribute values into equal parts. First assign them in an order(2,3,4,8,9,15,16,21,22,23,26,28)

Bin1: 2,3,4,8
Bin 2: 9,15,16,21
Bin3: 22,23,26,28

In the below example, dark dotes are outliers. K-means performs the best



Supervised Discretization:

Discretization methods which use class labels are known as supervised discretization. Supervised discretization produces better results than unsupervised discretization.
Entropy measure is a common method of discretization.

Entropy = $e_{i=}$

$$e_i = \sum_{i=1}^{k} p_{ij} \log_2 p_{ij},$$

Where, k=number of class labels, and i=(1,2,3..)
m_{ij}= number of values of class j in i^{th} interval
And, $p_{ij}=m_{ij}/m_{i.}$
The total entropy is denoted by

$$e = \sum_{i=1}^{n} w_i e_i,$$

Variable Transformation or attribute transformation

When transformation is performed on all values of a attribute, then it is known as variable transformation. This technique is mainly used when the attribute values are very large, or when the magnitude is more

Uses of variable transformation
1) It is used to transform larger values into smaller values
2) It is used to reduce the dependency of values on its units
3) It is also used when the magnitude of the values are important then the values.

Simple functions

In this, simple mathematical function is applied to all values of the variable (attribute) individually. For example consider a variable(x) it is converted by taking its $\sqrt{X}$ or e^x or 1/X or sin X or |x| or log X etc..

Weight (x)	Transformed weight
1000	31.66
100	10
350	18.70
245	15.65

This technique is mainly used when the value range is very large.
Note: In some cases the transformations changes the nature of the variable. For example the values {1, 2, 3} if transformed into {1/1, 1/2, 1/3} then the order changes i.e. (1<2<3) but (1/1> 1/2> 1/3).

Normalization or standardization

In this, formulas like mean and standard deviation are used. These methods are mainly used to reduce the dependency of values on its units. For example weight of people can be measured in kg or pounds. In order to consider weights irrespective of measures these techniques are helpful.
If $\bar{X}$ the mean of attribute values and S_x standard deviation then, $X' = (X-\bar{X})/S_X$ Creates a new variable.
The mean and standard deviation are affected by outliers. In this case median and absolute standard deviation can be used.

Absolute standard deviation= $\sum_{i=1}^{m} |Xi - \mu|$

Xi= value of the variable
μ= median
m= number of values of an attribute.

Measuring Data Similarity and Dissimilarity

In data mining applications, such as clustering, outlier analysis, and nearest-neighbor classification, we need ways to assess how alike or unalike objects are in comparison to one another.
For example, a store may want to search for clusters of customer objects, resulting in groups of customers with similar characteristics (e.g., similar income, area of residence, and age).

A **similarity measure** for two objects, i and j, will typically return the value 0 if the objects are unalike. Value is higher when objects are more alike

A **dissimilarity measure** works the opposite way. It returns a value of 0 if the objects are the same (and therefore, far from being dissimilar). The higher the dissimilarity value, the more dissimilar the two objects are.

Note: Proximity refers to a similarity or dissimilarity

1) Data Matrix versus Dissimilarity Matrix

Data Matrix: (or object-by-attribute structure): This structure stores the n data objects in the form of a relational table, or n-by-p matrix (n objects ×p attributes).

$$\begin{bmatrix} x11 & \cdots & x1p \\ \vdots & \ddots & \vdots \\ xn1 & \cdots & xnp \end{bmatrix}$$

A data matrix is made up of two entities or “things,” namely rows (for objects) and columns (for attributes). Therefore, the data matrix is often called a two-mode matrix.

Dissimilarity Matrix: Dissimilarity matrix (or object-by-object structure): This structure stores a collection of proximities that are available for all pairs of n objects. It is often represented by an n-by-n table.

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

For example, d(2,1) represents the distance between 2nd object and 1st object. The bigger the value of (2,1) represent the larger the difference between them.

The dissimilarity matrix contains one kind of entity (dissimilarities) and so is called a one-mode matrix.

2) Proximity Measures for Nominal Attributes

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

The proximity is calculated using following formula

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

In the above formula, Let **m** be total number of matches between two-point attributes and **p** be total number of attributes.

Example:

Roll No	**Marks**	**Grades**
1	96	A
2	87	B
3	83	B
4	96	A

Now, we apply the formula (described above) for finding the proximity of nominal attributes:

Note: p is total number of attributes. In our case p is 2 (Marks, Grades).

– d(1,1)= (p-m)/p = (2-2)/2 = 0

– d(2,1)= (p-m)/p = (2-0)/2 = 1

– d(3,1)= (p-m)/p = (2-2)/2 = 1

– d(4,1)= (p-m)/p = (2-2)/2 = 0

– d(4,3)= (p-m)/p = (2-0)/2 = 1

– d(2,2)= (p-m)/p = (2-2)/2 = 0

– d(3,2)= (p-m)/p = (2-1)/2 = 0.5
(One attribute i.e Grades is matching so, m is 1)

– d(4,2)= (p-m)/p = (2-0)/2 = 1

– d(3,3)= (p-m)/p = (2-2)/2 = 0

– d(4,4)= (p-m)/p = (2-2)/2 = 0

As seen from the calculation, we observe that the similarity between an object with itself is 1, which seems intuitively correct.

The proximity matrix is

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

$$\begin{bmatrix} 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 1 & 0.5 & 0 & 0 \\ 0 & 1 & 1 & 0 \end{bmatrix}$$

3) Proximity Measures for Binary Attributes

Similarity Measures for Binary Data are called similarity coefficients and typically have values between 0 and 1. The comparison between two binary objects is done using the following four quantities:

Name	Test 1	Test 2	Test 3	Test 4
Suresh	1	0	0	0
Ramesh	1	0	1	0
Rajesh	0	0	0	0

M_{11}= is the number of attributes that equal 1 for both objects i and j,
M_{10}= is the number of attributes that equal 1 for object i but equal 0 for object j
M_{01}= is the number of attributes that equal 0 for object i but equal 1 for object j,
M_{00}= is the number of attributes that equal 0 for both objects i and j.

a) Dissimilarity between i and j *in case of Binary Symmetric attributes are*

$$d(i,j) = \frac{M_{10} + M_{01}}{M_{11} + M_{10} + M_{01} + M_{00}}$$

Let's calculate dissimilarity between Ramesh, Suresh and Rajesh in case of binary symmetry attributes.

d(Ramesh, Suresh)=$\frac{0+1}{1+0+1+2} = \frac{1}{4} = 0.25$

d(Ramesh, Rajesh)= $\frac{2+0}{0+2+0+2} = \frac{2}{4} = 0.5$

d(Suresh, Rajesh)= $\frac{1+0}{0+1+0+3} = \frac{1}{4} = 0.25$

Dissimilarity matrix in case of binary symmetry attributes is:

$$\begin{bmatrix} \boldsymbol{Matrix} & Suresh & Ramesh & Rajesh \\ Suresh & 0 & - & - \\ Ramesh & 0.25 & 0 & - \\ Rajesh & 0.25 & 0.5 & 0 \end{bmatrix}$$

b) Dissimilarity between *i* and *j in case of Binary asymmetric attributes are*

$$d(i,j) = \frac{M_{10} + M_{01}}{M_{11} + M_{10} + M_{01}}$$

Let's calculate dissimilarity between Ramesh, Suresh and Rajesh in case of binary asymmetry attributes.

d(Ramesh, Suresh)=$\frac{0+1}{1+0+1} = \frac{1}{2} = 0.5$

d(Ramesh, Rajesh)= $\frac{2+0}{0+2+0} = \frac{2}{2} = 1$

d(Suresh, Rajesh)= $\frac{1+0}{0+1+0} = \frac{1}{1} = 1$

Dissimilarity matrix in case of binary asymmetry attributes is:

$$\begin{bmatrix} \boldsymbol{Matrix} & Suresh & Ramesh & Rajesh \\ Suresh & 0 & - & - \\ Ramesh & 0.5 & 0 & - \\ Rajesh & 1 & 1 & 0 \end{bmatrix}$$

c) Symmetric binary similarity between i and j are

$$d(i,j) = \frac{M_{11} + M_{00}}{M_{11} + M_{10} + M_{01} + M_{00}}$$

Let's calculate similarity between Ramesh, Suresh and Rajesh in case of symmetric binary attribute.

d(Ramesh, Suresh)=$\frac{1+2}{1+0+1+2} = \frac{3}{4} = 0.75$

d(Ramesh, Rajesh)= $\frac{0+2}{0+2+0+2} = \frac{2}{4} = 0.5$

d(Suresh, Rajesh)= $\frac{0+3}{0+1+0+3} = \frac{3}{4} = 0.75$

$$\begin{bmatrix} \boldsymbol{Matrix} & Suresh & Ramesh & Rajesh \\ Suresh & 0 & - & - \\ Ramesh & 0.75 & 0 & - \\ Rajesh & 0.75 & 0.5 & 0 \end{bmatrix}$$

d) Asymmetric binary similarity also known as Jaccard coefficient between i and j are

$$d(i,j) = \frac{M_{11}}{M_{11} + M_{10} + M_{01}}$$

Let's calculate similarity between Ramesh, Suresh and Rajesh in case of asymmetric binary attribute.

S(Ramesh, Suresh)=$\frac{1}{1+0+1} = \frac{1}{2} = 0.5$

S(Ramesh, Rajesh)= $\frac{0}{0+2+0} = 0$

S(Suresh, Rajesh)= $\frac{0}{0+1+0} = 0$

$$\begin{bmatrix} \boldsymbol{Matrix} & Suresh & Ramesh & Rajesh \\ Suresh & 0 & - & - \\ Ramesh & 0.5 & 0 & - \\ Rajesh & 0 & 0 & 0 \end{bmatrix}$$

4) Dissimilarity of Numeric Data

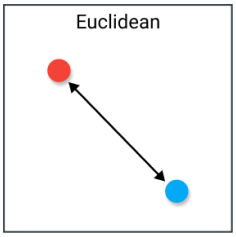
All attributes may not be of same scale. So, data may be normalized before applying these distance formulas. Normalizations is performed using Z-Score normalization.

Let's consider a dataset

Name	Test 1(out of 5)	Test 2 (Out of 5)
Rama	1	2
Mohan	3	5

a) Euclidean distance:

It is a distance measure that best can be explained as the length of a segment connecting two points. Euclidean distance is also called L2 norm.



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

$$d = \sqrt{(3-1)^2 + (5-2)^2}$$

$$d = \sqrt{(2)^2 + (3)^2}$$

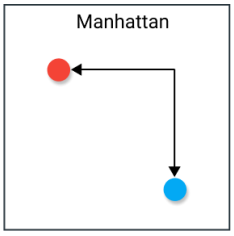
$$d = \sqrt{4+9}$$

$$d = \sqrt{13}$$

$$d = 3.605551$$

b) Manhattan distance

The Manhattan distance, often called Taxicab distance or City Block distance, calculates the distance between real-valued vectors. Imagine vectors that describe objects on a uniform grid such as a chessboard. Manhattan distance is also called L1 norm.



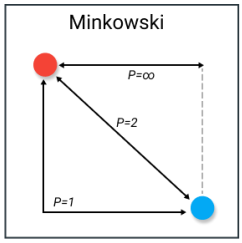
$$d(i,j) = |x_{i1} - x_{j1}| + |x_{i2} - x_{j2}| + \cdots + |x_{ip} - x_{jp}|.$$

d=|3-1|+|5-2|

d=5

c) Minkowski distance

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



It is defined as

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

$$\sqrt[2]{|3-1|^2+|5-2|^2}$$

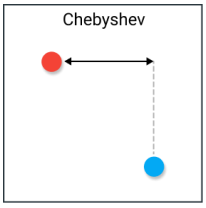
d=3.605

where h is a real number such that h >= 1. (Such a distance is also called Lp norm in some literature, where the symbol p refers to our notation of h.

It represents the Manhattan distance when h = 1 (i.e., L1 norm) and Euclidean distance when h = 2

d) Supremum distance

Chebyshev distance is defined as the greatest of difference between two vectors along any coordinate dimension. In other words, it is simply the maximum distance along one axis.



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

The second attribute gives the greatest difference between values for the objects, which is 5-2 = 3. This is the supremum distance between both objects.

5) Proximity Measures for Ordinal Attributes

The values of an ordinal attribute have a meaningful order or ranking about them, yet the magnitude between successive values is unknown. An ordinal variable can be discrete or continuous. An example includes the sequence **small, medium, large** for a size attribute. Ordinal attributes may also be obtained from the discretization of numeric attributes by splitting the value range into a finite number of categories. These categories are organized into ranks.

Sl.No	**Test (Ordinal)**

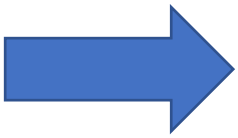
1	Excellent
2	Fair
3	Good
4	Excellent

Consider the above table, the proximity is calculated in 4 steps.

Step 1: Find the value of Mf (Count the number of states. In our case Mf is 3 (Excellent, Good, Fair).

Step 2: Replace each x_{if} by its corresponding rank, $r_{if} \{ f_1, : : : , M_f \}$.

Sl.No	**Test (Ordinal)**
1	Excellent
2	Fair
3	Good
4	Excellent



Sl.No	**Test (Ordinal)**
1	3
2	1
3	2
4	3

Replace as: Excellent-3, Good-2, Fair-1.

Step 3: Normalize the ranks: Since each ordinal attribute can have a different number of states, it is often necessary to map the range of each attribute onto [0.0, 1.0] so that each attribute has equal weight. We perform data normalization for this.

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

Where,

r_{if} = i^{th} object in the f^{th} attribute

M_f = Number of unique states=3

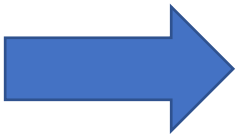
Fair (1) = $\frac{1-1}{3-1} = 0$

Good (2) = $\frac{2-1}{3-1} = 0.5$

Excellent (3) = $\frac{3-1}{3-1} = 1$

The above data is modified as,

Sl.No	Test (Ordinal)
1	3
2	1
3	2
4	3



Sl.No	Test (Ordinal)
1	1
2	0
3	0.5
4	1

Step 4: Dissimilarity can then be computed using any of the distance measures

Let's use Manhattan distance:

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

d(2,1)=|0-1|=1

d(2,3)=|0.5-1|=0.5

d(2,4)=|0-1|=1

d(3-4)=|0.5-1|=0.5

d(1,3)=|1-0.5|=0.5

d(1,4)=|1-1|=0

$$\begin{bmatrix} 0 & - & - & - \\ 1 & 0 & - & - \\ 0.5 & 0.5 & 0 & - \\ 0 & 1 & 0.5 & 0 \end{bmatrix}$$

Therefore, objects 1 and 2 are the most dissimilar, as are objects 2 and 4 (i.e., d(2,1)= 1.0 and d(4,2)= 1.0). This makes intuitive sense since objects 1 and 4 are both excellent.

Similarity values for ordinal attributes can be interpreted from dissimilarity as **sim(i, j)= 1-d(i, j).**

6) Proximity Measures of Mixed Attributes

There are two approaches to compute the dissimilarity between objects of mixed attribute types.

1) One approach is to group each type of attribute together, performing separate data mining (e.g., clustering) analysis for each type. This is feasible if these analyses derive compatible results. However, in real applications, it is unlikely that a separate analysis per attribute type will generate compatible results.

2) A more preferable approach is to process all attribute types together, performing a single analysis. One such technique combines the different attributes into a single dissimilarity matrix, bringing all of the meaningful attributes onto a common scale of the interval [0.0,1.0].

Suppose that the data set contains p attributes of mixed type. The dissimilarity

d(i , j) between objects i and j is defined as

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

Where $\delta_{ij}^{(f)}$ =0 if,

i. x_{if} or x_{jf} is missing,

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

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

$d_{ij}^{(f)}$ depends on type of attribute and can be calculated as with help of following formulas:

(1) if f is numeric $d_{ij}^{(f)} = \frac{|x_{if}-x_{jf}|}{max-min}$

(2) if f is nominal or binary $d_{ij}^{(f)}$=0 if $x_{if} = x_{jf}$, Otherwise $d_{ij}^{(f)}$=1

(3) if f is ordinal, compute the ranks $z_{if} = \frac{r_{if}-1}{M_f-1}$ and threat z_{if} as numeric.

Example: Compute the dissimilarity between objects of mixed attribute types given in Table

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

a) Dissimilarity Matrix for Attribute Test-1 (Nominal)

Find the dissimilarity matrix between objects for attribute **test-1** which is of type nominal using formula no 2. Dissimilarity is 1 if two objects have different value for attribute Test-1 otherwise 0.

Dissimilarity Matrix between objects for attribute test-1 (Nominal).

Object identifier	1	2	3	4
1	0			
2	1	0		
3	1	1	0	
4	0	1	1	0

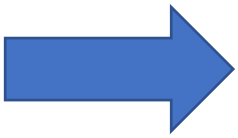
b) Dissimilarity Matrix for Attribute Test-2 (Ordinal)

Consider the above table, the proximity is calculated in 4 steps.

Step 1: Find the value of Mf (Count the number of states. In our case Mf is 3 (Excellent, Good, Fair).

Step 2: Replace each x_{if} by its corresponding rank, $r_{if} \{ f_1, : : : , M_f \}$.

Sl.No	Test (Ordinal)
1	Excellent
2	Fair
3	Good
4	Excellent



Sl.No	Test (Ordinal)
1	3
2	1
3	2
4	3

Replace as: Excellent-3, Good-2, Fair-1.

Step 3: Normalize the ranks: Since each ordinal attribute can have a different number of states, it is often necessary to map the range of each attribute onto [0.0, 1.0] so that each attribute has equal weight. We perform data normalization for this.

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

Where,

r_{if} = i^{th} object in the f^{th} attribute

M_f = Number of unique states=3

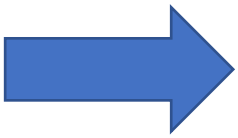
Fair (1) = $\frac{1-1}{3-1} = 0$

Good (2) = $\frac{2-1}{3-1} = 0.5$

Excellent (3) = $\frac{3-1}{3-1} = 1$

The above data is modified as,

Sl.No	Test (Ordinal)
1	3
2	1
3	2
4	3



Sl.No	Test (Ordinal)
1	1
2	0
3	0.5
4	1

Step 4: Dissimilarity can then be computed using any of the distance measures

Let's use Manhattan distance:

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

d(2,1)=|0-1|=1

d(2,3)=|0.5-1|=0.5

d(2,4)=|0-1|=1

d(3-4)=|0.5-1|=0.5

d(1,3)=|1-0.5|=0.5

d(1,4)=|1-1|=0

$$\begin{bmatrix} 0 & - & - & - \\ 1 & 0 & - & - \\ 0.5 & 0.5 & 0 & - \\ 0 & 1 & 0.5 & 0 \end{bmatrix}$$

c) Dissimilarity Matrix for Attribute Test-3 (Numerical)

Find the dissimilarity matrix between objects for attribute test-3 which is of type numerical using formula no 1. normalize the values so that can be mapped between [0,1] using formula

$$d_{ij}^{(f)} = \frac{|x_{if} - x_{jf}|}{max - min}$$

Consider the data below.

Object identifier	test-3 (numeric)
1	45
2	22
3	64
4	28

$$d_{1,2} = \frac{\left|x_{1f} - x_{2f}\right|}{64 - 22} = \frac{45 - 22}{42} = 0.548$$

$$\mathrm{d}_{1,3} = \frac{|45 - 64|}{42} = 0.452$$

$$\mathrm{d}_{1,4} = \frac{|45 - 28|}{42} = 0.405$$

$$\mathrm{d}_{2,3} = \frac{|22 - 64|}{42} = 1$$

$$\mathrm{d}_{2,4} = \frac{|22 - 28|}{42} = 0.143$$

$$\mathrm{d}_{3,4} = \frac{|64 - 28|}{42} = 0.857$$

Dissimilarity Matrix between objects for attribute Test-3 (Numerical).

Object identifier	1	2	3	4
1	0			
2	0.548	0		
3	0.452	1	0	
4	0.405	0.143	0.857	0

Single dissimilarity matrix between all objects for all attributes

Now, combines the different attributes into a single dissimilarity matrix. The dissimilarity d (i , j) between objects i and j is defined as

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

Where $\delta_{ij}^{(f)}$ =0 if,

i. x_{if} or x_{jf} is missing,

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

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

As we can see that there is no any missing value and also no any asymmetric binary attribute, so $\delta_{ij}^{(f)}$=1 for all data points. And p=3 as there 3 attributes.

Let's keep all dissimilarly matrix here.

Test 1	1	2	3	4
1	0			
2	1	0		
3	1	1	0	
4	0	1	1	0

Test 2	1	2	3	4
1	0			
2	1	0		
3	0.5	0.5	0	
4	0	1	0.5	0

Test 3	1	2	3	4
1	0			
2	0.548	0		
3	0.452	1	0	
4	0.405	0.143	0.857	0

Dissimilarity between Object 1 and Object 2

$$d(1,2)=\frac{\sum_{f=1}^{p=3} \delta_{1,2}^{(f)} d_{1,2}^{(f)}}{\sum_{f=1}^{p=3} \delta_{1,2}^{(f)}}=\frac{(\delta_{1,2}^{(Test-1)} d_{1,2}^{(Test-1)})+(\delta_{1,2}^{(Test-2)} d_{1,2}^{(Test-2)})+(\delta_{1,2}^{(Test-3)} d_{1,2}^{(Test-3)})}{\delta_{1,2}^{(Test-1)}+\delta_{1,2}^{(Test-2)}+\delta_{1,2}^{(Test-3)}}$$

$$d(1,2)=\frac{(1*1)+(1*1)+(1*0.548)}{1+1+1}=\frac{2.548}{3}=0.849$$

Similarly,

$$d(1,3)=\frac{(1*1)+(1*0.5)+(1*0.405)}{1+1+1}=\frac{1.905}{3}=0.635$$

$$d(1,4)=\frac{(1*0)+(1*0)+(1*0.405)}{1+1+1}=\frac{0.405}{3}=0.135$$

$$d(2,3)=\frac{(1*1)+(1*0.5)+(1*1)}{1+1+1}=\frac{2.5}{3}=0.8$$

$$d(2,4)=\frac{(1*1)+(1*1)+(1*0.143)}{1+1+1}=\frac{2.143}{3}=0.714$$

$$d(3,4)=\frac{(1*1)+(1*0.5)+(1*0.857)}{1+1+1}=\frac{2.357}{3}=0.952$$

Dissimilarity Matrix between objects for attributes (Test-1, Test-2, Test-3)

Object identifier	**1**	**2**	**3**	**4**
1	0			
2	0.849	0		
3	0.635	0.8	0	
4	0.135	0.714	0.952	0

Cosine similarity

Cosine similarity is a measure of similarity that can be used to compare documents or, say, give a ranking of documents with respect to a given vector of query words. Let x and y be two vectors for comparison. Using the cosine measure as a similarity function, we have

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

For example, in the below table, we see that Document1 contains five instances of the word team, while hockey occurs three times. The word coach is absent from the entire document, as indicated by a count value of 0. Such data can be highly asymmetric.

Document	*team*	*coach*	*hockey*	*baseball*	*soccer*	*penalty*	*score*	*win*	*loss*	*season*
Document1	5	0	3	0	2	0	0	2	0	0
Document2	3	0	2	0	1	1	0	1	0	1
Document3	0	7	0	2	1	0	0	3	0	0
Document4	0	1	0	0	1	2	2	0	3	0

Cosine similarity is mainly used in information retrieval, biologic taxonomy, gene feature mapping

$Document_1$ = (5, 0, 3, 0, 2, 0, 0, 2, 0, 0)

$Document_2$ = (3, 0, 2, 0, 1, 1, 0, 1, 0, 1)

$Document_1 \bullet Document_2 = 5*3+0*0+3*2+0*0+2*1+0*1+0*1+2*1+0*0+0*1 = 25$

$||Document_1|| = (5*5+0*0+3*3+0*0+2*2+0*0+0*0+2*2+0*0+0*0)^{0.5}=(42)^{0.5} = 6.481$

$||Document_2|| = (3*3+0*0+2*2+0*0+1*1+1*1+0*0+1*1+0*0+1*1)^{0.5}=(17)^{0.5} = 4.12$

$\cos(Document_1, Document_2) = 0.94$

UNIT 4

PART 1

BASIC CONCEPTS

Que 1: What is classification? What is purpose of classification?

Classification is the process of finding a model (or function) using training data that describes and distinguishes data classes or concepts, for the purpose of being able to use the model to predict the class of objects whose class label is unknown.

Purpose of classification: **1) Descriptive modeling 2) Predictive modeling**

Descriptive modeling is a classification model used for summarizing the data in terms of classes and attributes as shown below

Name	Body Temperature	Skin Cover	Gives Birth	Aquatic Creature	Aerial Creature	Has Legs	Hiber-nates	Class Label
human	warm-blooded	hair	yes	no	no	yes	no	mammal
python	cold-blooded	scales	no	no	no	no	yes	reptile
salmon	cold-blooded	scales	no	yes	no	no	no	fish
whale	warm-blooded	hair	yes	yes	no	no	no	mammal
frog	cold-blooded	none	no	semi	no	yes	yes	amphibian
komodo dragon	cold-blooded	scales	no	no	no	yes	no	reptile
bat	warm-blooded	hair	yes	no	yes	yes	yes	mammal
pigeon	warm-blooded	feathers	no	no	yes	yes	no	bird
cat	warm-blooded	fur	yes	no	no	yes	no	mammal
leopard	cold-blooded	scales	yes	yes	no	no	no	fish

Predictive modeling is a classification model used to predict the class label of unknown records.

Name	Body Temperature	Skin Cover	Gives Birth	Aquatic Creature	Aerial Creature	Has Legs	Hiber-nates	Class Label
gila monster	cold-blooded	scales	no	no	no	yes	yes	?

Que2: What are the applications of Classification?

- **Customer Target Marketing**: Predict buying interests on the basis of previous training examples. The target variable may encode the buying interest of the customer.
- **Medical Disease Diagnosis**: The features may be extracted from the medical records, and the class labels correspond to whether or not a patient may pick up a disease in the future. In these cases, it is desirable to make disease predictions with the use of such information.
- **Biological Data Analysis:** Biological data can be used to build a model which can be used to predict a new virsus, bacteria etc…
- **Document Categorization and Filtering**: Many applications, such as newswire services, require the classification of large numbers of documents in real time. This application is referred to as document categorization.
- **Social Network Analysis:** For predicting useful properties of actors/person in a social network.
- **Detecting spam email messages** based upon the message header and content.
- **Classifying galaxies** based upon their shapes.

GENERAL APPROACH TO SOLVING A CLASSIFICATION PROBLEM

Que 3: What is the General approach to solve a classification problem?

Training data: Set of sample whose class labels are known and are used for building a model

Test set: Set of samples whose class labels are hided. The model is applied on test set to predict class labels. This predicted class labels are compared with actual class labels to know the accuracy of model.

Learning algorithm: Algorithm that learns from training data and create set of rules. This model can be used to predict class label of samples whose class label is unknown.

Model: It is set of rules to predict class labels.

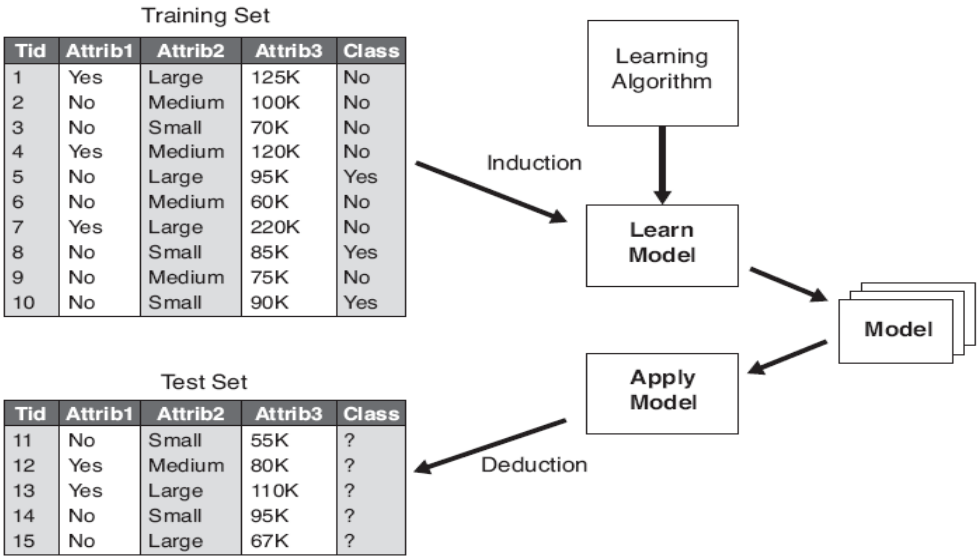


Figure 4.3. General approach for building a classification model.

1) First partition the historical data into training set and test set.
2) Use a classification algorithm like **decision tree classifiers**, to build a model from training data (Model is a set of rules learned from training set)
3) Apply this model on test set to check accuracy of model.

   Accuracy of model $= \frac{\text{number of test samples correctly predicted}}{\text{total number of samples in test set}}$
4) If accuracy is good, then apply the model on new /unknown sample whose class label is unknown and predict the class label.

Que 4: Write a short note on:

a. Confusion matrix

b. Performance metric:

i. Accuracy

ii. Error rate

Confusion matrix: It is a matrix having count of number of class labels correctly predicted and number of class labels incorrectly predicted by model.

Confusion matrix representation:

		Predicted Class	
		$Class = 1$	$Class = 0$
Actual Class	$Class = 1$	f_{11}	f_{10}
	$Class = 0$	f_{01}	f_{00}

In the above table f_{11} and f_{00} are correct predictions by model and f_{01} and f_{10} are incorrect predictions of model.

Accuracy: Defined as the ratio of samples correctly classified

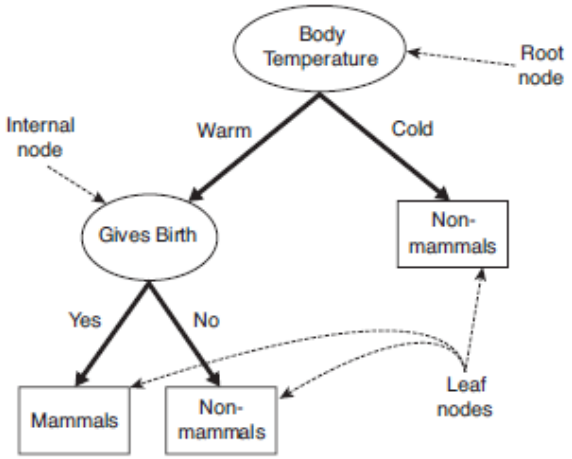
$$\text{Accuracy} = \frac{\text{Number of correct predictions}}{\text{Total number of predictions}} = \frac{f_{11} + f_{00}}{f_{11} + f_{10} + f_{01} + f_{00}}.$$

Error Rate: Defined as the ratio of samples correctly classified

$$\text{Error rate} = \frac{\text{Number of wrong predictions}}{\text{Total number of predictions}} = \frac{f_{10} + f_{01}}{f_{11} + f_{10} + f_{01} + f_{00}}.$$

DECISION TREE INDUCTION

Que 5: Explain the working of a decision tree.



The tree has three types of nodes.

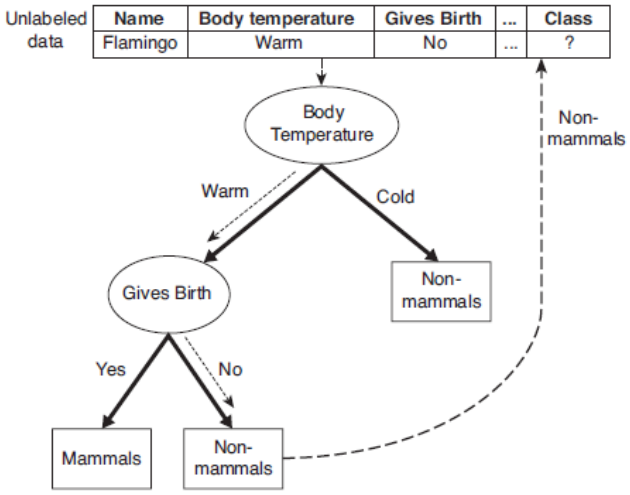
i) A **root node** has no incoming edges and zero or more outgoing edges.

ii) **Internal nodes**, each of which has exactly one incoming edge and two or more outgoing edges.

iii) **Leaf** or **terminal nodes**, each of which has exactly one incoming edge and no outgoing edges.

In terms of decision tree induction, root and internal nodes are called attribute test conditions. And, leaf (or) terminal nodes are class labels.

Once a decision tree is build, we apply *attribute test condition* on records and follow appropriate branches based on outcome of test condition. This will either lead to an internal *branch node* where another *attribute test condition* is applied or a *leaf node* where class label is assigned.



BUILDING A DECISION TREE

Que 6 : Explain Hunt's algorithm for building decision trees

There are various methods for building a decision tree

1) **Hunt'algorithm**
2) ID3
3) C4.5
4) CART

Hunt's algorithm

In Hunt's algorithm, a decision tree is grown in a recursive fashion by partitioning the training records into subsets.

Let D_t be a set of training records that are associated with node t and $y=\{y_1,y_2,\ldots,y_c\}$ be the class labels.

The recursive procedure for hunt's algorithm is as follows:

STEP 1

If all the records in D_t belong to same class y_t, then t is a leaf node labeled as y_t.

STEP 2

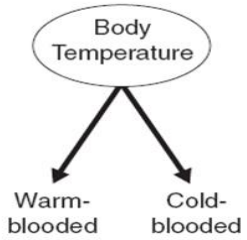
If D_t contains records that belong to more than one class, an attribute test condition is selected to partition the records into smaller subsets. A child node is created for each outcome and the records in D_t are distributed based on the outcomes. The algorithm is then recursively applied for each node.

METHODS FOR EXPRESSING AN ATTRIBUTE TEST CONDITIONS

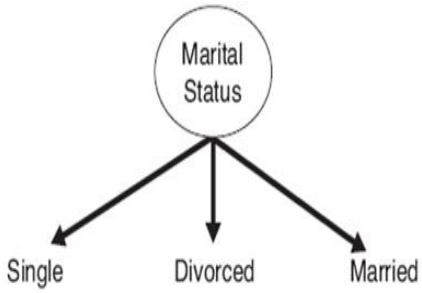
Que 7: What are the Methods for expressing attribute test conditions

The following are the methods for expressing attribute test conditions. They are:

i) **Binary attribute:** The test condition for binary attribute generate two outcomes as shown below:

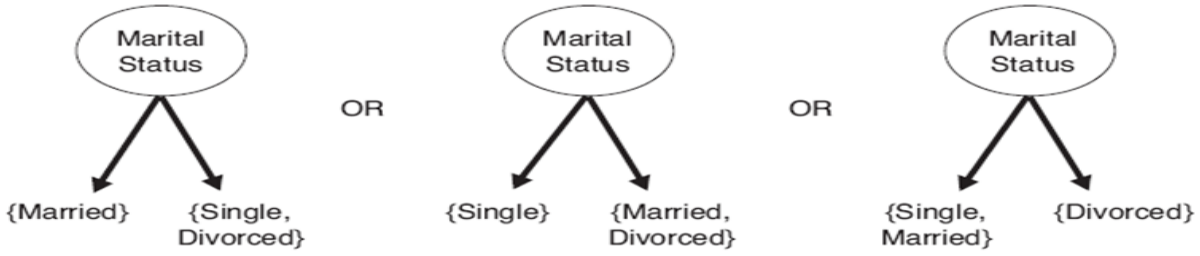


ii) **Nominal attributes:** since a nominal attribute can have many values, its test condition can be expressed in two ways as shown below:



(a) Multiway split

For a multi way split, the number of outcomes depends on the number of distinct values for the corresponding attribute.



(b) Binary split {by grouping attribute values}

iii) Ordinal attribute: It can also produce binary or multi way splits. It is same as nominal attributes but the attribute values have order among them.

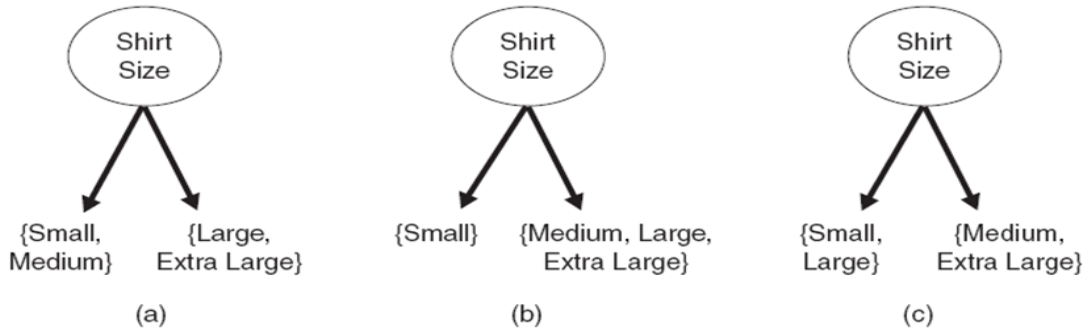


Figure 4.10. Different ways of grouping ordinal attribute values.

In the above example, condition 'a' and condition 'b' satisfies order but condition 'c' violates the order property.

iv) Continuous attributes: The test condition can be expressed as a comparison test with binary outcomes, or a range with many outcomes.

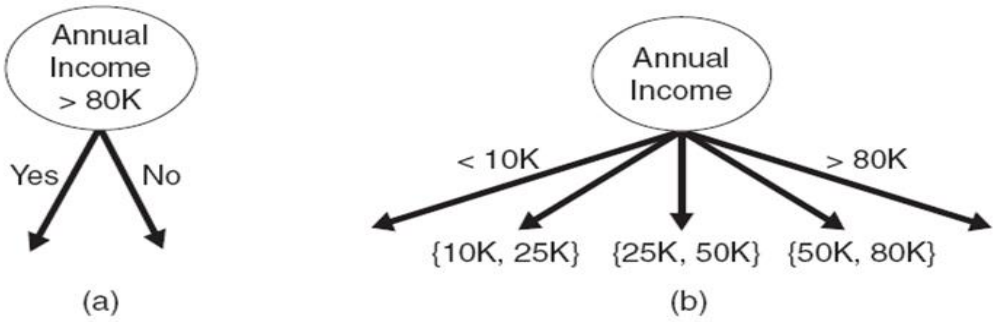


Figure 4.11. Test condition for continuous attributes.

MEASURES FOR SELECTING THE BEST SPLIT

Que 8: what are the measures for selecting the best split

Three main methods for selecting the best split

1) Entropy
2) Gini Index
3) Classification error

Let P(i|t) denote the fraction of records belonging to class i at a node t. the measures for selecting the best split are often based on the degree of impurity of the child nodes. The smaller the degree of impurity, the more skewed the class distribution. For example, a node with class distribution (0,1) has zero impurity, whereas a node with uniform class distribution (0.5,0.5) has the highest impurity.

Examples of impurity measures include: (Note: C is number of classes)

$$\text{Entropy}(t) = -\sum_{i=0}^{c-1} p(i|t)\log_2 p(i|t),$$

$$\text{Gini}(t) = 1 - \sum_{i=0}^{c-1}[p(i|t)]^2,$$

$$\text{Classification error}(t) = 1 - \max_i[p(i|t)],$$

Node N_1	Count
Class=0	0
Class=1	6

$\text{Gini} = 1 - (0/6)^2 - (6/6)^2 = 0$
$\text{Entropy} = -(0/6)\log_2(0/6) - (6/6)\log_2(6/6) = 0$
$\text{Error} = 1 - \max[0/6, 6/6] = 0$

Node N_2	Count
Class=0	1
Class=1	5

$\text{Gini} = 1 - (1/6)^2 - (5/6)^2 = 0.278$
$\text{Entropy} = -(1/6)\log_2(1/6) - (5/6)\log_2(5/6) = 0.650$
$\text{Error} = 1 - \max[1/6, 5/6] = 0.167$

Node N_3	Count
Class=0	3
Class=1	3

$\text{Gini} = 1 - (3/6)^2 - (3/6)^2 = 0.5$
$\text{Entropy} = -(3/6)\log_2(3/6) - (3/6)\log_2(3/6) = 1$
$\text{Error} = 1 - \max[3/6, 3/6] = 0.5$

The 3 measures attain maximum values when the class distribution is uniform and minimum when all the records belong to same class.

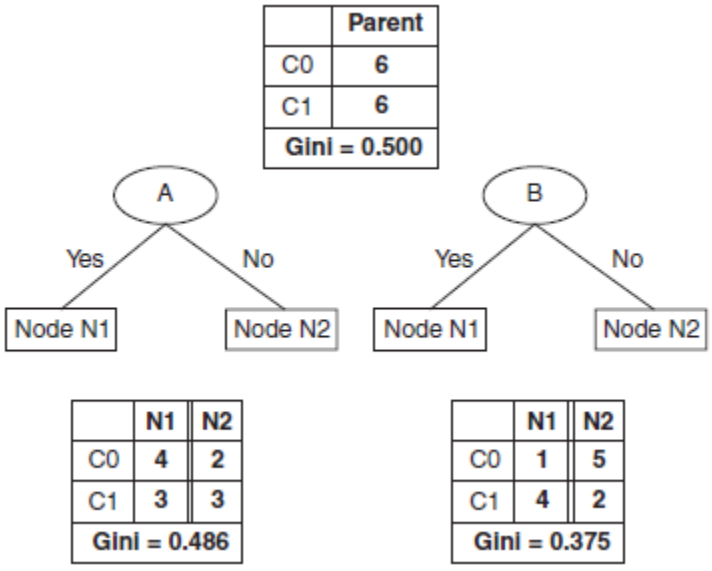
//Optional Compare the degree of impurity of the parent node with the degree of impurity of the child node. The larger their difference, the better the test condition. The gain, Δ, is a criterion that can be used to determine the goodness of a split.

$$\Delta = I(\text{parent}) - \sum_{j=1}^{k} \frac{N(v_j)}{N} I(v_j),$$

Where I(.) is the impurity measure of a given node, N is the total number of records at the parent node, k is the attribute values and $N(v_j)$ is the number of records associated with node v_j. when entropy is used as impurity measure the difference in entropy is known as information gain, Δ_{info}. **//**

Que 9: what are the methods for Splitting of binary attributes?

Suppose there are two ways to split the data into smaller subsets, say, A and B. before splitting the GINI index is 0.5 since there are equal number of records from both the classes.



For attribute A,

For node N1, the GINI index is $1-[(4/7)^2+(3/7)^2]=0.4898$

For node N2, the GINI index is $1-[(2/5)^2+(3/5)^2]=0.48$

The average weighted GINI index is $(7/12)(0.4898)+(5/12)(0.48)=0.486$

For attribute B, the average weighted GINI index is 0.375, since the subsets for attribute B have smaller GINI index than A, attribute B is preferable.

Que 10: What are the methods for Splitting of nominal attributes?

A nominal attribute can produce either binary or multi way split.

	Car Type	
	{Sports, Luxury}	{Family}
C0	9	1
C1	7	3
Gini	0.468	

	Car Type	
	{Sports}	{Family, Luxury}
C0	8	2
C1	0	10
Gini	0.167	

(a) Binary split

	Car Type		
	Family	Sports	Luxury
C0	1	8	1
C1	3	0	7
Gini	0.163		

(b) Multiway split

The computation of GINI index is same as for binary attributes. The smaller the average GINI index is the best split. In our example, multi way split has the lowest GINI index, so it is the best split.

Consider 3rd table,

The Gini index for Family cars: $1-(1/4)^2-(3/4)^2 = 0.375$

The Gini index for Sports cars: $1-(8/8)^2-(0/8)^2= 0$

The Gini index for Luxury cars: $1-(7/8)^2-(1/8)^2=0.21938$

$(4/20)*0.375 + (8/20)*0+ (8/20)*0.21938=0.075+0.0877=0.163$

Que 11: What are the methods for Splitting of continuous attributes?

Consider continuous attribute Annual income from below table.

	binary	categorical	continuous	class
Tid	Home Owner	Marital Status	Annual Income	Defaulted Borrower
1	Yes	Single	125K	No
2	No	Married	100K	No
3	No	Single	70K	No
4	Yes	Married	120K	No
5	No	Divorced	95K	Yes
6	No	Married	60K	No
7	Yes	Divorced	220K	No
8	No	Single	85K	Yes
9	No	Married	75K	No
10	No	Single	90K	Yes

- Attribute values of annual income and corresponding class label are arranged in horizontal way as shown below. These are called **sorted values**
- Calculate the mean between every two adjacent values and plot below the sorted values. Theses mean values are called **split positions.**

Class	No		No		No		Yes		Yes		Yes		No		No		No		No		No	
	Annual Income																					
Sorted Values →	60		70		75		85		90		95		100		120		125		220			
Split Positions →	55		65		72		80		87		92		97		110		122		172		230	
	<=	>	<=	>	<=	>	<=	>	<=	>	<=	>	<=	>	<=	>	<=	>	<=	>	<=	>
Yes	0	3	0	3	0	3	0	3	1	2	2	1	3	0	3	0	3	0	3	0	3	0
No	0	7	1	6	2	5	3	4	3	4	3	4	3	4	4	3	5	2	6	1	7	0
Gini	0.420		0.400		0.375		0.343		0.417		0.400		*0.300*		0.343		0.375		0.400		0.420	

Example:

Consider mean value 97,

Number of 'Yes' class labels <= 97 is 3

Number of 'No' class labels <=97 is 3

Number of 'No' class labels >97 is 4

Number of 'yes' class labels >97 is 0

Calculate Gini Index for less than 97:

$1-(3/6)^2-(3/6)^2= 0.5$

$1-(0/4)^2-(4/4)^2=0$

Average Gini index for 97 is

$(6/10)* 0.5 + (4/10)*0=0.3$

In the same way, Calculate the GINI index for every split position and the smaller GINI index split position can be chosen as the best split for the attribute.

ALGORITHM FOR DECISION TREE INDUCTION.

Que:12 Write the Algorithm for decision tree induction

Algorithm 4.1 A skeleton decision tree induction algorithm.

```
TreeGrowth (E, F)
 1: if stopping_cond(E,F) = true then
 2:    leaf = createNode().
 3:    leaf.label = Classify(E).
 4:    return leaf.
 5: else
 6:    root = createNode().
 7:    root.test_cond = find_best_split(E, F).
 8:    let V = {v|v is a possible outcome of root.test_cond }.
 9:    for each v ∈ V do
10:       E_v = {e | root.test_cond(e) = v and e ∈ E}.
11:       child = TreeGrowth(E_v, F).
12:       add child as descendent of root and label the edge (root → child) as v.
13:    end for
14: end if
15: return root.
```

i) The **createNode()** function extends the decision tree by creating a new node. A node in the decision tree has either a test condition, denoted as node.test_cond, or a class label, denoted as node.label.

ii) The **find.best_split ()** function determines which attribute should be selected as the test condition for splitting the training records.

iii) The **classify()** function determines the class label to be assigned to a leaf node.

iv) The **stopping_cond()** function is used to terminate the tree-growing process by testing whether all the records are classified or not.

Que 13: Explain a decision tree with an example

Consider the below table:

A	B	Class Label
T	F	+
T	T	+
T	T	+
T	F	−
T	T	+
F	F	−
F	F	−
F	F	−
T	T	−
T	F	−

The contingency tables after splitting on attributes A and B are:

	$A=T$	$A=F$
+	4	0
−	3	3

	$B=T$	$B=F$
+	3	1
−	1	5

The overall entropy before splitting is:

$$E_{orig} = -0.4 \log 0.4 - 0.6 \log 0.6 = 0.9710$$

The information gain after splitting on A is:

$$\begin{aligned} E_{A=T} &= -\frac{4}{7}\log\frac{4}{7} - \frac{3}{7}\log\frac{3}{7} = 0.9852 \\ E_{A=F} &= -\frac{3}{3}\log\frac{3}{3} - \frac{0}{3}\log\frac{0}{3} = 0 \\ \Delta &= E_{orig} - 7/10 E_{A=T} - 3/10 E_{A=F} = 0.2813 \end{aligned}$$

The information gain after splitting on B is:

$$\begin{aligned} E_{B=T} &= -\frac{3}{4}\log\frac{3}{4} - \frac{1}{4}\log\frac{1}{4} = 0.8113 \\ E_{B=F} &= -\frac{1}{6}\log\frac{1}{6} - \frac{5}{6}\log\frac{5}{6} = 0.6500 \\ \Delta &= E_{orig} - 4/10 E_{B=T} - 6/10 E_{B=F} = 0.2565 \end{aligned}$$

Therefore, attribute A will be chosen to split the node.

Que 14: What are the characteristics of Decision tree induction?

1) No previous assumptions and thresholds are needed prior to construction of decision tree induction.
2) Finding an optimal tree is heuristic
3) Constructing a decision tree is computationally less expensive. And, classifying an unknown sample is extremely fast.
4) Robust to presence of noise.
5) Redundant attributes do not adversely affect the accuracy of decision tree
6) Irrelevant attribute affect decision tree at large. Say, choosing an irrelevant attribute as test condition may result in irrelevant tree.
7) Data fragmentation problem: In some situations, number of training records become small as me move down the decision tree. At a stage, these samples may be insufficient to label the leaf node.
8) Tree replication problem: A same set of sub tree is replicated may times in a tree. This problem occurs when there are few attributes.

Que 15 : What do you mean by tree pruning? Why pruning is useful for decision tree induction. What is drawback of a separate set of tuples to evaluate pruning?

Tree pruning: Removing of branches (or) Trimming a tree in order to improve generalization capability of a decision tree is called tree pruning.

(or)

Trimming a tree to reduce generalization errors is called tree pruning. Tree pruning is mainly used to reduce over fitting of data

Techniques of tree pruning

1) Pre-pruning
2) Post-pruning

Pre-pruning: Tree growing on a training data is restricted by keeping certain thresholds fixed by user. Means, a leaf node is restricted to expand when its error crosses (Training and generalization errors) certain thresholds. But, large thresholds may lead to under fitting and low thresholds may lead to over fitting. And deciding correct thresholds is very tricky.

Post-pruning: Initially, a tree is fully grown on a training data. Then, the unnecessary branches are removed from the tree. Post-pruning is better than pre-pruning. However unnecessary components (leafs and branches) should be grown which are computationally very expensive.

The decision tree built may overfit the training data. There could be too many branches, some of which may reflect anomalies in the training data due to noise or outliers. Tree pruning addresses this issue of overfitting the data by removing the least reliable branches (using statistical measures). This generally results in a more compact and reliable decision tree that is faster and more accurate in its classification of data.

The drawback of using a separate set of tuples to evaluate pruning is that it may not be representative of the training tuples used to create the original decision tree. If the separate set of tuples are biased or distorted, then using them to evaluate the pruned tree would not be a good indicator of the pruned tree's classification accuracy. Furthermore, using a separate set of tuples to evaluate pruning means there are less tuples to use for creation and testing of the tree. While this is considered a drawback in machine learning, it may not be so in data mining due to the availability of larger data sets.

Que 16. Write a short note on

1) precision
2) Recall

3) F-Measure

Precision: It tells how many of the returned samples are correct

$$\textbf{Precision} = TP/(TP+FP)$$

Recall: It tells how many of the positive samples does the model return.

$$\textbf{Recall} = TP/(TP+FN)$$

F Measure = (**2*TP**)/ (**2*TP+FP+FN**)

= (**2 * (*PRECISION* * *RECALL*)**)/ (***PRECISION*** + ***RECALL***)

F Measure represents harmonic mean between recall and precision.

TP: True positive: Corresponds to number of positive samples correctly predicted by the classification model
FN: false negative: Corresponds to number of positive samples wrongly predicted as negative by the classification model
FP: false positive: Corresponds to number of negative samples wrongly predicted as positive by the classification model.
TN: True negative: Corresponds to number of negative examples correctly predicted by the classification model.

Que 17: Explain issues regarding classification and prediction:

The major issue of Classification and Prediction are:

- **Data Cleaning** – Data cleaning involves removing the noise and treatment of missing values. The noise is removed by applying smoothing techniques and the problem of missing values is solved by replacing a missing value with most commonly occurring value for that attribute.

- **Relevance Analysis** – Database may also have the irrelevant attributes. Correlation analysis is used to know whether any two given attributes are related.

- **Data Transformation and reduction** – The data can be transformed by any of the following methods.
    - **Normalization** – The data is transformed using normalization. Normalization involves scaling all values for given attribute in order to make them fall within a small specified range. Normalization is used when in the learning step, the neural networks or the methods involving measurements are used.
    - **Generalization** – The data can also be transformed by generalizing it to the higher concept. For this purpose we can use the concept hierarchies.
- **Accuracy** – Accuracy of classifier refers to the ability of classifier to predicts the class label correctly. and the accuracy of the predictor refers to how well a given predictor can guess the value of predicted attribute for a new data.
- **Speed** – This refers to the computational cost in generating and using the classifier or predictor.
- **Robustness** – It refers to the ability of classifier or predictor to make correct predictions from given noisy data.
- **Scalability** – Scalability refers to the ability to construct the classifier or predictor efficiently; given large amount of data.

Que 18: What do you mean by eager and lazy learner? What are advantages and disadvantages?

Eager learner: Eager learners, when given a set of training tuples, will construct a generalization (i.e., classification) model before receiving new (e.g., test) tuples to classify. We can think of the learned model as being ready and eager to classify previously unseen tuples.

Lazy learner: Lazy learner waits until the last minute before doing any model construction in order to classify a given test tuple. That is, when given a training tuple, a lazy learner simply stores it (or does only a little minor processing) and waits until it is given a test tuple. It performs generalization after it encounters a new test tuples. Lazy learner as also called instance based learner.

Eager learner	lazy learner
1. Simply stores training data (or only minor processing) and waits until it is given a test tuple	**1.** Given a set of training tuples, constructs a classification model before receiving new (e.g., test) data to classify
2. More time for training but less time for predicting	**2.** Less time in training but more time in predicting
3. Decision tree and naïve bayes are examples of eager learner classifier	**3.** Case Base approach and K-nearest neighbor are examples of lazy learners
4. must commit to a single hypothesis (i.e model) that covers the entire instance space	**4.** Lazy method effectively uses a richer hypothesis space since it uses many local linear functions to predict the output of target function
5. Requires less storage as compared to lazy learners	**5.** Requires efficient storage techniques and more efficient indexing is required.
6. Computationally less expensive as compared to lazy learner	**6.** Computationally expensive
7. Parallel computation is not required.	**7.** Requires parallel computation
8. Doesn't support incremental learning	**8.** Supports incremental learning

PART 2

MODEL OVER FITTING

Que 12: Write a short note on the following:

1) **Training error**
2) **Generalization error**
3) **Model under-fitting.**
4) **Model Over-fitting.**

Training errors is the number of misclassification errors committed on training records. Means, A Model may not correctly predict the class label of unknown samples which is same as training data.

Generalization errors: It is the expected error of the model on previously unseen records. Means, A Model may not correctly predict the class label of unknown samples which is different from training data.

Model Under-fitting: The training and generalization error rates are large when the size of the tree is very small. This situation is known as **model underfitting**.

Model Over-fitting: When the tree becomes large, the test error rate increases and training error rate decreases. This situation is known as **model over-fitting**.

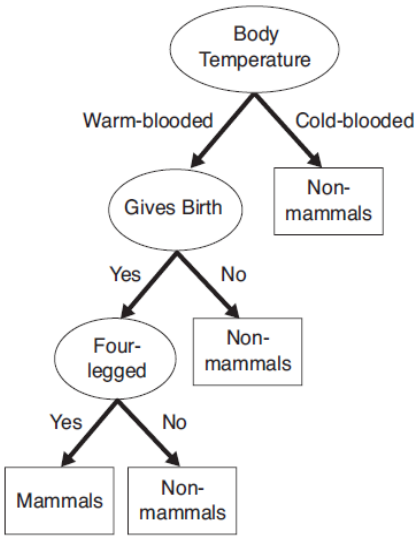
Que 13: What do you mean by over fitting due to present of noise?

- Consider the training and test sets for the mammal classification problem. Two of the ten records are mislabeled. Bats and whales are classified as non mammals instead of mammals.

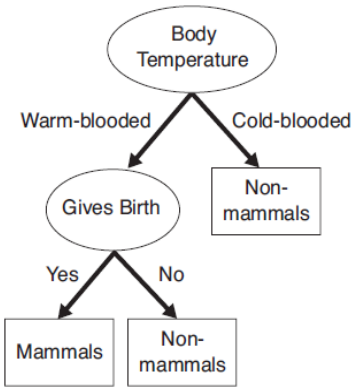
Name	Body Temperature	Gives Birth	Four-legged	Hibernates	Class Label
porcupine	warm-blooded	yes	yes	yes	yes
cat	warm-blooded	yes	yes	no	yes
bat	warm-blooded	yes	no	yes	no*
whale	warm-blooded	yes	no	no	no*
salamander	cold-blooded	no	yes	yes	no
komodo dragon	cold-blooded	no	yes	no	no
python	cold-blooded	no	no	yes	no
salmon	cold-blooded	no	no	no	no
eagle	warm-blooded	no	no	no	no
guppy	cold-blooded	yes	no	no	no

- The class label for {name='human', body-temperature='warm-blooded', gives berth='yes', four-legged='no', hibernates='no'} is non-mammals from above decision tree. But humans are mammals. The prediction is wrong due to presence of noise in data.

- Even the training error is zero the model has generalization errors.



(a) Model M1



(b) Model M2

Model M1 is Over trained (Over fitted) which wrongly classifies human as non-mammal .Model M2 (which is not over trained) correctly predicts human as mammal.

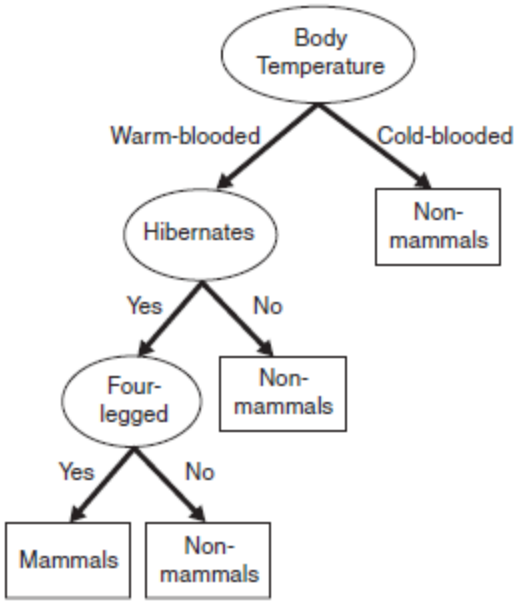
Que14: what do you mean by over fitting due to lack of representative samples?

When a model is built on very few training data, Over-fitting occurs. Means, lack of representative sample in training data may lead to a tree which wrongly predicts the class label of unknown sample.

For Example, Consider below table:

Name	Body Temperature	Gives Birth	Four-legged	Hibernates	Class Label
salamander	cold-blooded	no	yes	yes	no
guppy	cold-blooded	yes	no	no	no
eagle	warm-blooded	no	no	no	no
poorwill	warm-blooded	no	no	yes	no
platypus	warm-blooded	no	yes	yes	yes

The corresponding model is,



From above example, humans {*name='human', body-temperature='warm-blooded', gives berth='yes', four-legged='no', hibernates='no'*} are wrongly classified as non mammals.

EVALUATING THE PERFORMANCE OF CLASSIFIER

Que 15: What are the methods for evaluating performance of classifier?

1) **Holdout Method**
2) **Random Sampling**
3) **Cross-Validation**
    a. **Two fold cross validation**
    b. **K-fold cross validation**
    c. **Leave-one-out approach**
4) **Bootstrap**

- **Holdout Method**

  In this method the original data sets is divided into two parts, 50% or 2/3rd of original data is considered as training sets and another 50% or 1/3rd of original data as test sets respectively. Now, the classification model is trained on training tests and then applied on test sets. The performance of the classification algorithm is based on number of correct predictions made on the test set.

  Limitations

  1) Less number of samples for training(since the original samples are spitted)
  2) The model is highly dependent on the composition of the training and test sets

- **Random Sampling**

  Multiple repetition of holdout method is known as random sampling. Here the original data is divided randomly into training sets and test sets and the accuracy is calculated as in holdout method. This random sampling is then repeated k times and the accuracy is calculated for each time. The overall accuracy is:

$$acc_{\text{sub}} = \sum_{i=1}^{k} acc_i / k.$$

  Here acc_i is the model accuracy during i th iteration

  Limitations

  1) Less number of samples for training(since the original samples are spitted)
  2) A record may be used more than once in training and test tests.

- **Cross-Validation**

  There are three variations of cross-validation approach

  a) Two fold cross validation

  In this approach data is partitioned into two parts. The first part is considered as training set and the second part as test set. Now they are

swapped and the first part is considered as test set and second one as training set. The total error is the sum of both the errors.

b) K-fold cross validation

In this approach the data is partitioned into k subsets. One of the partitions is considered as test set and remaining sets are considered as training set. This process is repeated k times and the total error is the sum of all the k runs.

c) Leave-one-out approach

In this approach one record is considered as test set and rest of the samples are considered as training set. This process is repeated k times (k= number of records) and the total error is the sum of all the k runs. But this process is computationally very expensive.

- **Bootstrap**

  In this approach a record may be sampled more than once. Means a record when sampled is again kept back in the original data. So it is likely that the record may be sampled again and again. Consider original data of size N. The probability of a record to be chosen as bootstrap sample is $1-(1-1/N)^N$ .When the Size of N is very large then the probability is $1-e^{-1}$. The sampling is repeated B times to generate b bootstrap samples.

Annexure

These questions may not be in syllabus but some of these questions were asked in previous question papers.

Que 16: What do you mean by Bayesian classifier and bayes theorem? Explain how Bayesian classifier can be used for classification.

Bayesian classifiers

Bayesian classifier is the classification technique which uses **bayes theorem** for identifying unknown class label. Bayesian classifier can be implemented in two ways. They are:

i) Naïve Bayes Classifier

ii) Bayesian belief networks

Bayes Theorem

Bayes Theorem can be given as:

$$P(Y|X) = \frac{P(X|Y)*P(Y)}{P(X)}$$

Example: A conversation was held in a train with a person about long hair. There are 50% men and 50% women. In most of the cases, this conversation will be initiated by women. Suppose that 75% of women have long hair and 15% of men have long hair. We need to identify who initiated the conversation (men or women),

Bayes theorem can be written as:

$$P(W|L) = \frac{P(L|W)*P(W)}{P(L)}$$

where P (W|L) represents probability of women with long hair initiating the conversation.

P (L|W) represents probability of women with long hair. (75% of women have long hair. So, the probability is 0.75)

P (W) represents probability of women population. (there are 50% women. So, the probability is 0.50)

P (L) represents total probability of long hair.

Where P(L)= P(L|W)* P(W) + P(L|M)* P(M)

$$P(W|L) = \frac{0.75*0.50}{0.75*0.50+0.15*0.50} = \frac{5}{6}$$

$$=0.83\ (83\%)$$

The probability that a woman initiating the conversation was 83% and men initiating the conversation was 17%

Using the Bayes Theorem for classification

Let **X** denotes the attribute set and **Y** denote the class label or variable.

$$P(Y|X) = \frac{P(X|Y)*P(Y)}{P(X)}$$

Where **P (Y|X)** is the conditional probability also known as posterior probability for **Y**

P(Y) is the priori probability

During the training phase, we need to learn the posterior probabilities **P (Y|X)** for every combination of **X** and **Y** based on information gathered in the training data.

Consider the task of predicting whether a loan borrower will default on their payments. The below table is a training set with following attributes: Home Owner, Marital Status and Annual Income. Loan borrowers who paid the loan are classified as **no** and who defaulted is classified as **yes.**

Let us consider a test record **X= (Home Owner=No, Marital Status=Married, Annual Income=120K)**

In order to classify the record, we need to compute P (Yes |X) and P (No |X). If P (Yes |X) > P (No |X), then the record is classified as Yes, else, No.

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

Que 17: Explain Naïve Bayes Classifier with an example.

A Naïve Bayes Classifier estimates the class conditional probability by assuming that the attributes are conditionally independent, given the class label y. The conditional independence assumption can be formally stated as follows:

$$P(X \mid Y=y) = \prod_{i=1}^{d} P(Xi \mid Y = y)$$

To classify a test record, the naïve bayes classifier computes the posterior probability for class label Y as follows:

$$P(Y \mid X) = \frac{\prod_{i=1}^{d} P(Xi \mid Y=y) * P(Y)}{P(X)}$$

P(X) is fixed for every Y.

Estimating conditional probability for categorical attribute

$P(X_i = x_i \mid Y=y)$ is estimated according to the fraction of training instances in class y

Example

The conditional probability for P (Home Owner=Yes | No) = $\frac{3}{7}$ (7 represents the total number of samples with class label=No and 3 represents the total number of samples with Home Owner=Yes and class label =No)

Estimating conditional probability for continuous attribute

$$P(X_i=x_i \mid Y=y) = \frac{1}{\sqrt{2\Pi\sigma ij}} exp^{\frac{-(xi-\mu ij)2}{2\sigma ij2}}$$

Where μ_{ij} or $\bar{x}$ is the mean of values in a continuous attribute

$$\bar{x} = \frac{\sum_{i=1}^{n} xi}{n}$$

σ_{ij}^2 or s^2 is the variance or standard deviation

$$S^2 = \frac{\sum_{i=1}^{n}(xi-\bar{x})2}{n(n-1)}$$

Example

Consider the below loan data set and predict the class label of a test record

X= (Home owner=No, Marital Status=married, annual income=120K)

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

There are three records that belong to the class **Yes** and seven records that belong to class **No** $P(Y)$ can be calculated by

Therefore, P (Yes) = $\frac{3}{10}$

P (No) = $\frac{7}{10}$

Now calculate $P(\text{Xi}|Y = y)$,

For this we need to compute the posterior probabilities P (No| X) and P (Yes| X) Home owner and marital status are categorical attributes. The conditional probability can be calculated as follows:

P (Home owner=No| No) = $\frac{4}{7}$

P (Home owner=No| Yes) = $\frac{3}{3}$ =1

P (Marital status= Married| No) = $\frac{4}{7}$

P (Marital status= Married| Yes) = $\frac{0}{3}$ =0

But annual income is a continuous attribute, so, we need to use the below formula for calculating the conditional probability for (annual income= 120K)

$$P(X_i=x_i \mid Y=y) = \frac{1}{\sqrt{2\Pi\sigma ij}} exp^{\frac{-(xi-\mu ij)2}{2\sigma ij2}}$$

P (annual income= 120K| No) = $\frac{1}{\sqrt{2\Pi\sigma ij}} exp^{\frac{-(xi-\mu ij)2}{2\sigma ij2}}$

$\mu ij = \frac{125+100+70+120+60+220+75}{7} = 110$

$\sigma_{ij}^2 = \frac{(110-125)2+(110-100)2+(110-70)2+(110-120)2+(110-60)2+(110-220)2+(110-75)2}{7(7-1)}$

$= 420.23$

$\sigma_{ij} = \sqrt{420.23} = 20.49$

Substituting the values of μij, σ_{ij}^2 and σ_{ij} in the above equation, we get the value of P (annual income= 120K| No) = 0.017

P (annual income= 120K| Yes) = $\frac{1}{\sqrt{2\Pi\sigma ij}} exp^{\frac{-(xi-\mu ij)2}{2\sigma ij2}}$

$\mu ij = \frac{95+85+90}{3} = 90$

$\sigma_{ij}^2 = \frac{(90-95)2+(90-85)2+(90-90)2}{3(3-1)} = 8.12$

$\sigma_{ij} = \sqrt{8.12} = 2.84$

Substituting the values of μij, σ_{ij}^2 and σ_{ij} in the above equation, we get the value of P (annual income= 120K| Yes) = 0.014

Therefore,

P (X| No) = P (Home owner=No| No) * P (Marital status= Married| No) *
P (annual income= 120K| No)

$= \frac{4}{7} * \frac{4}{7} * 0.017 = 0.0055$

P (X| Yes) = P (Home owner=No| Yes) * P (Marital status= Married| Yes) *
P (annual income= 120K| Yes)

$= 1 * 0 * 0.014 = 0$

The posterior probability for class No is P (No| X) $= \alpha * \frac{7}{10} * 0.0055 = 0.00385\alpha$

The posterior probability for class Yes is P (Yes| X) $= \alpha * \frac{3}{10} * 0 = 0$

Where $\alpha = \frac{1}{P(X)}$ is a constant term

Since P (No| X) > P (Yes| X), the record is classified as **No**

Que 18: What are the characteristics of Naïve bayes classifier? Explain how correlated attributes affect the performance of a naïve bayes classifier.

1. They are robust to isolated noise points because such points are averages out when estimating conditional probabilities from data.
2. It can also handle missing values by ignoring the example during model building and classification.
3. They are robust to irrelevant attributes.
4. Correlated attributes can degrade the performance of naïve bayes classifiers because the conditional independence assumption no longer holds for such attributes.

Example

Consider the following probabilities:

P (A=0 | Y=0) =0.4, P (A=1 | Y=0) =0.6,

P (A=0 | Y=1) =0.6, P (A=1 | Y=1) =0.4

where A is a binary attribute and Y is a binary class variable. Suppose there is another attribute B which is perfectly correlated with A when Y=0, but is independent of A when Y=1. Let us assume that class conditional probabilities for B are same as for A.

Given a record (A=0, B=0), the posterior probabilities are calculated as follows:

$$P(Y=0 \mid A=0, B=0) = \frac{P(A=0|Y=0)P(B=0|Y=0)P(Y=0)}{P(A=0)P(B=0)}$$

$$= \frac{0.16 * P(Y=0)}{P(A=0)P(B=0)}$$

$$P(Y=1 \mid A=0, B=0) = \frac{P(A=0|Y=1)P(B=0|Y=1)P(Y=1)}{P(A=0)P(B=0)}$$

$$= \frac{0.36 * P(Y=1)}{P(A=0)P(B=0)}$$

If P(Y=0) =P(Y=1), then the record is classified to class 1. However the truth is,

P (A=0, B=0 | Y=0) = P (A=0 | Y=0) =0.4

Because A and B are perfectly correlated when Y=0. The result for Y=0 is

$$P(Y=0 \mid A=0, B=0) = \frac{P(A=0, B=0|Y=0)P(Y=0)}{P(A=0)P(B=0)}$$

$$= \frac{0.4 * P(Y=0)}{P(A=0)P(B=0)}$$

Which is larger than that for y=1. The record should have been classified as class 0.

Que 19: Write a short note on Bayes Error Rate

Suppose we know the true probability distribution that governs P(X|Y). The Bayesian classification method allows us to determine the ideal decision boundary for the classification task.

Example

Consider the task of identifying alligators and crocodiles based on their lengths. The average length of an adult crocodile is about 15 feet and alligator is 12 feet. Assuming that their length x follows a Gaussian distribution with a standard deviation equal to 2 feet, we can express their class conditional probabilities as follows:

$$P(X|\text{ crocodile}) = \frac{1}{\sqrt{2\Pi}\,2} exp\,\frac{-1}{2}^{(\frac{x-15}{2})2}$$

$$P(X|\text{ alligator}) = \frac{1}{\sqrt{2\Pi}\,2} exp\,\frac{-1}{2}^{(\frac{x-12}{2})2}$$

Assuming that the prior probabilities are same

$P(X=\hat{x}\ |\text{Crocodile}) = P(X=\hat{x}\ |\text{alligator})$

Using above equations w obtain:

$$(\frac{x-15}{2})^2 = (\frac{x-12}{2})^2$$

Which can be solved to yield x=13.5 i.e., the length< 13.5 are alligators and length> 13.5 are crocodiles.

Que 20: Explain briefly about Bayesian Belief Networks

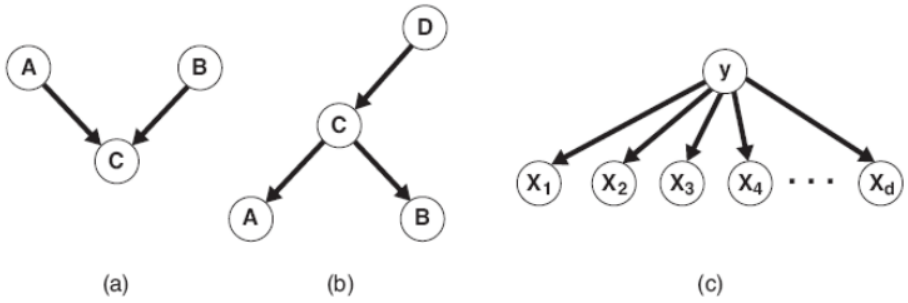
There are some problems with correlated attributes when calculating conditional probabilities using naïve bayes classifier. These problems can be rectified using Bayesian belief networks.

Model Representation

A Bayesian Belief Networks (BBN) provides a graphical representation of the probabilistic relationships among a set of random variables. There are two key elements of a Bayesian network. They are:

1. A directed acyclic graph (DAG) encoding the relationship between the set of variables.
2. A probability table associating each node to its immediate parent nodes.

Consider three random variables A, B and C, in which A and B are independent variables but have direct influence with C as shown below:



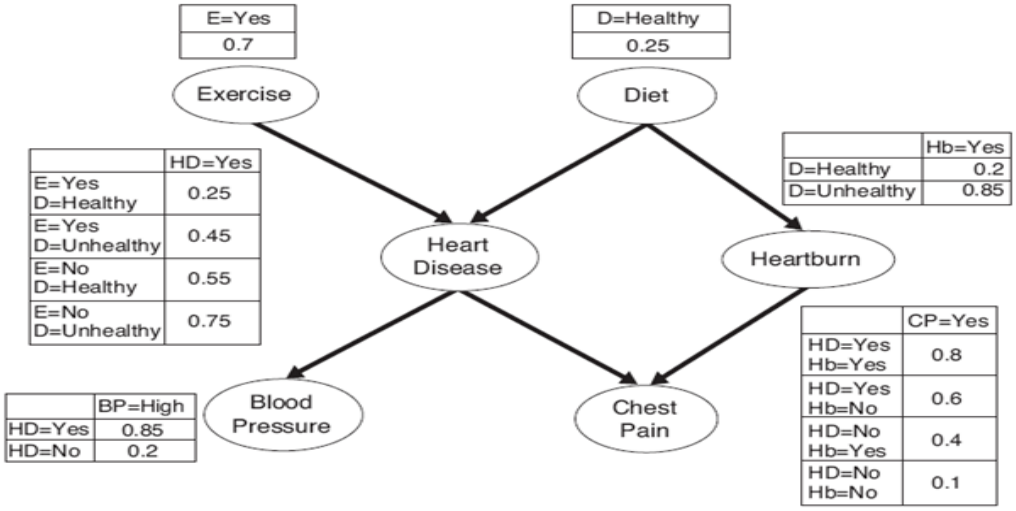
Representing probabilistic relationships using directed acyclic graphs.

If there is a directed arc from X to Y, then X is the **parent** of Y and Y is the **child** of X. If there is a directed path from X to Z, then X is the **ancestor** of Z and Z is a **descendant** of X.

Here **nodes** represent **variables or attributes** and **arcs** represent the **relationship** between those variables or attributes.

Each node is associated with a probability table basing on the following conditions:

1. If a node X does not have any parents, then the table contains only the prior information P(X)
2. If a node X has only one parent, then the table contains the conditional probability P(X|Y)
3. If a node X has multiple parents, then the table contains the conditional probability P (X|Y1,Y2,….., Y_k)



A Bayesian belief network for detecting heart disease and heartburn in patients.

The above figure is an example of a Bayesian network for modeling patients with heart disease and heart burn. Each variable in the diagram is assumed to be a binary valued.

Conditions

a) Exercise and diet are risk factors of heart disease.
b) Blood pressure and chest pain are the symptoms of heart disease.
c) Diet is the risk factor for heart burn.
d) Chest pain is the symptom for heart burn.

The nodes exercise and diet has only prior probability since it does not have any parent node. The nodes heart disease and heart burn and their symptoms contain conditional probabilities.

Example

P (Heart Disease=No | Exercise=No, Diet=healthy)

=1-P (Heart Disease=Yes | Exercise=No, Diet=healthy)

=1-0.55 =0.45

Model building

Model building in Bayesian networks involves two steps:

1) Creating the structure of the network.
2) Estimating the probability values in the tables associated with each node.

Algorithm for generating a Bayesian Network

1: let $T=(X_1,X_2,\ldots,X_n)$ denote a total order of variables.

2: **for** j=1 to d **do**

3: let $X_{T(j)}$ denote the jth highest order variable in T

4: let $\square(X_{T(j)})= \{X_{T(1)},X_{T(2)},\ldots,X_{T(j-1)}\}$ denote the set of variables preceding $X_{T(j)}$.

5: Remove the variables from $\square(X_{T(j)})$ that do not affect X_j (using prior knowledge).

6: Create an arc between $\square(X_{T(j)})$ and the remaining variables in $\square(X_{T(j)})$.

7: **end for**.

Example

Consider the variables as shown in the above figure. After performing the first step, let us assume that the variables are ordered in the following way: (E, D, HD, HB, CP, BP)

- P(D|E) is simplified to P(D). //since D and E are independent variables
- P(HD|E, D) cannot be simplified. //both E and D are dependent on HD
- P(HB|E, D, HD) is simplified to P(HB|D) //since E and HD are not dependent on HB
- P(CP|E, D, HD, HB) is simplified to P(CP|HB, HD) //since E and D are not dependent on CP
- P(BP|E, D, HD, HB, CP) is simplified to P(BP|HD) //since D, E, HB, CP are not dependent on CP

Basing on these conditional probabilities, we can create arcs between the nodes (E, HD), (D, HD), (D, HB), (HD, CP), (HB, CP) and (HD, BP). By connecting these arcs a network is formed as above fig.

Characteristics of BBN

1. BBN provides an approach for capturing the prior knowledge of a particular domain using a graphical model. The network can also be used to encode casual dependencies among variables.
2. Constructing the network can be time consuming and requires a large amount of effort. However, once the structure of the network has been determined, adding a new variable is quite straightforward.
3. Bayesian networks are well suite to dealing with incomplete data. Instances with missing values can be handled by summing or integrating the probabilities over all possible values of the attribute.
4. Because the data is combined probabilistically with prior knowledge, the method is quite robust to model over fitting.

UNIT 4

Association Analysis: Basic Concepts and Algorithms: Problem Definition, Frequent Item Set Generation, Apriori Principle, Apriori Algorithm, Rule Generation, Compact Representation of Frequent Itemsets, FP- Growth Algorithm. (Tan & Vipin)

INTRODUCTION

Que 1: What is association analysis?

Association analysis is used to discover interesting relationships hidden in large data sets.

The relationship hidden between data elements are represented in the form of association rules.

TID	Items
1	{Bread, Milk}
2	{Bread, Diapers, Beer, Eggs}
3	{Milk, Diapers, Beer, Cola}
4	{Bread, Milk, Diapers, Beer}
5	{Bread, Milk, Diapers, Cola}

Consider the dataset above the corresponding association rule is:

{milk} →{Bread} [s=2%,c=50%]

This type of analysis is called market basket analysis.

Applications of Association Analysis:

1) Market basket analysis: Buying trends of customers can be analyzed and retailers can make an offer between set of items to increase sales.
2) Earth science: Association between ocean, land, pollution and effect on temperature can be analyzed.
3) Medical diagnosis: Relation between food habits, exercise and decease can be analyzed.

Que 2: Write short note on:

a) **Binary representation:**
b) **Item set**
c) **Transaction width**
d) **support count**
e) **Support**
f) **Confidence**
g) **Frequent Itemset**
h) **Anti monotone property.**
i) **Support based pruning**
j) **Apriori principle**

a) Binary representation of association data:

TID	Bread	Milk	Diapers	Beer	Eggs	Cola
1	1	1	0	0	0	0
2	1	0	1	1	1	0
3	0	1	1	1	0	1
4	1	1	1	1	0	0
5	1	1	1	0	0	1

Each transaction is represented as '1's and'0's. Where '1' represents presence of an item in the transaction and '0' represent absence of an item in the transaction.

b) Item set: Consider I be set of items. In association rule, collection of item(s) forms an item-set.

{}→ Null item set

{brush, paste)→ 2 Itemset

c) Transaction width: Number of items in itemset is called transaction width.

d) Support count: Number of times the itemset is found in the transaction set. (Or) Number of transactions containing the itemset.

Consider the table below,

TID	Items
1	{Bread, Milk}
2	{Bread, Diapers, Beer, Eggs}
3	{Milk, Diapers, Beer, Cola}
4	{Bread, Milk, Diapers, Beer}
5	{Bread, Milk, Diapers, Cola}

Support count of itemset {Bread, Milk} is 3.
Because, {bread, Milk} is present in 3 transactions (1, 4 and 5).

e) Support: Support represents how often a rule is applicable to a dataset.

$$\text{Support, } s(X \longrightarrow Y) \quad = \quad \frac{\sigma(X \cup Y)}{N};$$

Where X and Y are items and N represents total number of transactions. Means, Out of all transactions (N) how many transactions contains both X and Y.

f) Confidence: Confidence represents, how often Y appear in transaction containing X.

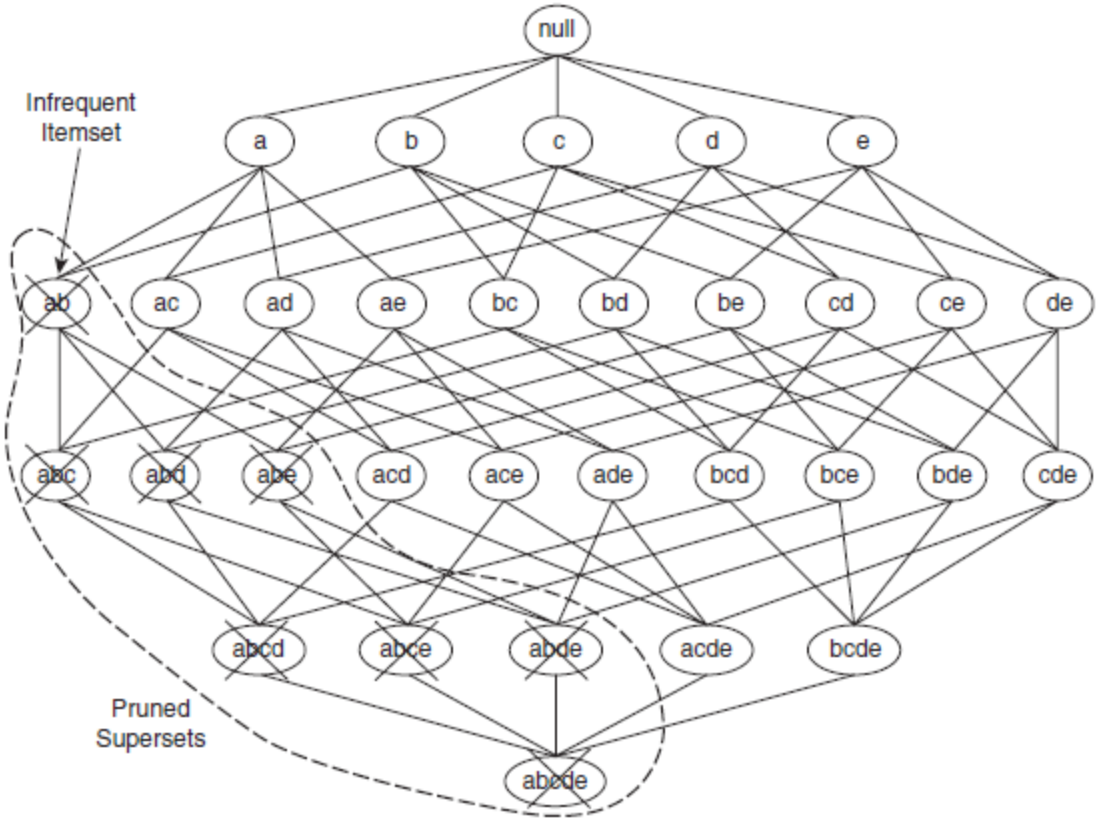
$$\text{Confidence, } c(X \longrightarrow Y) \quad = \quad \frac{\sigma(X \cup Y)}{\sigma(X)}.$$

Out of all transactions containing X, how many transactions contains Y.

g) Frequent itemset: An itemset is called frequent itemset if has minimum support and minimum confidence defined by the user.

h) Anti monotone property: Support of an itemset never exceeds its subset. For example, {brush, toothpaste} may have support count of 5. The item set (brush, toothpaste, bread} will never exceed 5.

i) Support based pruning: If an itemset is infrequent then all supersets containing the itemsets should also be infrequent.



If an item set {ab} is infrequent itemset, then all supersets containing both 'a' and 'b' are also infrequent and can be pruned(removed).

j) Apriori principle: If an itemset is frequent then, all its subsets are also frequent. Say for example, {c, d, e} is frequent itemset, then its subsets {c}, {d}, {e},{cd}, {de},{ce} should also be frequent.

Que 3: Explain Apriori algorithm in detail with example.
(Or)
How to find frequent item sets using candidate generation?

Algorithm 6.1 Frequent itemset generation of the *Apriori* algorithm.

1: $k = 1$.
2: $F_k = \{ i \mid i \in I \wedge \sigma(\{i\}) \geq N \times minsup\}$. {Find all frequent 1-itemsets}
3: **repeat**
4: $k = k + 1$.
5: $C_k = \text{apriori-gen}(F_{k-1})$. {Generate candidate itemsets}
6: **for** each transaction $t \in T$ **do**
7: $C_t = \text{subset}(C_k, t)$. {Identify all candidates that belong to t}
8: **for** each candidate itemset $c \in C_t$ **do**
9: $\sigma(c) = \sigma(c) + 1$. {Increment support count}
10: **end for**
11: **end for**
12: $F_k = \{ c \mid c \in C_k \wedge \sigma(c) \geq N \times minsup\}$. {Extract the frequent k-itemsets}
13: **until** $F_k = \emptyset$
14: Result $= \bigcup F_k$.

In the above algorithm,

K= Size of itemset

F_k= Frequent k^{th} item set.

I= Set of items (In below example, I={ I1,I2,I3,I4,I5})

i= Each element in I

σ{i}= Support count of items

Ck= Candidate set

T= Transaction database

t= individual transaction

Ct= candidates in Ck present in transaction data.

c= each candidate (or) each itemset in the candidate set.

σc = Support count of each candidate (or) Itemset.

Apriori_gen= A function call to generate candidate sets. (For example 2 item candidate set can be generated from one item frequent set).

Algorithm Explanation Step by step: //Optional

1. K is the size of itemset. Its value is initially set to 1 because 1-itemset should be generated first.
2. The support count of 1-itemset should be derived and the itemsets satisfying minimum support are kept and rests of them are deleted (F_1 is generated here).
    a. *Note: Here I is set of items (In below example, I={ I1,I2,I3,I4,I5})*
    b. *And i is individual item*
    c. *σ{i} is support count of items (Number of times the item i is found in our dataset)*
3. Repeat the process
4. Increment value of K to generate $k+1^{th}$ candidate set
5. After generating 1- item frequent set F_1, in step 2, Generate 2-Item candidate set using function call apriori_gen on F_1 .
    2- Item candidate set C_2 is generated from 1-item frequent set using one of the following approaches.
        i. *Brute force approach*
        ii. *F_{k-1}*F1 approach*
        iii. *F_{k-1} *F_{k-1} approach*
6. Consider each individual transaction (t) from our transaction dataset(T) and
7. Use function call 'subset' on C_k to generate itemsets present in our Transaction dataset. Let these itemsets be C_t.
    Note: delete itemsets from 2 item candidate sets whose support count is zero. And call rest of them as C_t
8. Out of all individual itemsets 'c' in 'C_t'
9. Find the support count of each candidate c in Ct. By scanning each transaction t and incrementing support count. $\sigma c = \sigma c+1$
10. End loop
11. End loop
12. Repeat step 4 to step 9 until biggest frequent item set is generated.
    a. *Support count for each candidate c is measured (σc). And candidates' c whose support count is greater than minimum support count is retained. This forms F_k*
13. This process is repeated until support count of all itemsets fall below minimum support count.
14. At the end, we have biggest frequent item sets.

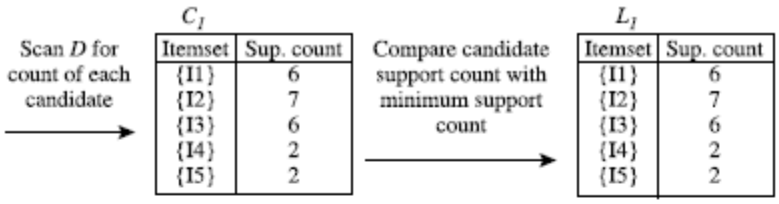
//

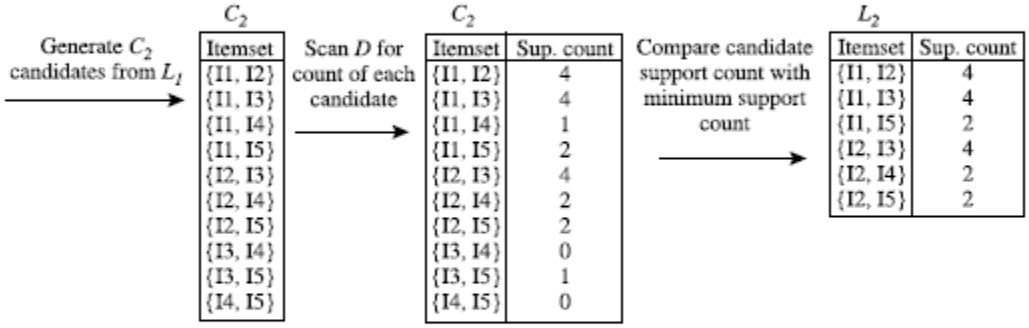
Example of apriori algorithm

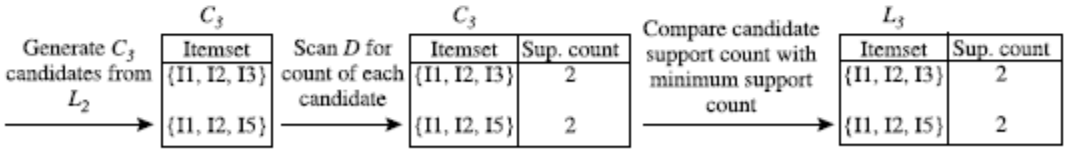
Consider a Transaction dataset T

TID	List of item_IDs
T100	I1, I2, I5
T200	I2, I4
T300	I2, I3
T400	I1, I2, I4
T500	I1, I3
T600	I2, I3
T700	I1, I3
T800	I1, I2, I3, I5
T900	I1, I2, I3

Apply apriori algorithm







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

Que 4: How to generate association rules from frequent item sets in apriori. (Or) What is confidence based pruning?

Once frequent itemsets are generated, then strong association rule can be generated by using confidence of the rule.

$$confidence(A \Rightarrow B) = P(B|A) = \frac{support\_count(A \cup B)}{support\_count(A)}.$$

Where,

support_count(AUB): Number of transactions containing both A & B together.

support_count(A): Number of transactions containing A.

Association rule generation is a two step process.

1) Generate all non empty subsets of a frequent itemset
2) For every subset calculate and find

$$\frac{\text{support\_count of frequent itemset}}{\text{support\_count of subset}} > Minimum\ confidence$$

TID	List of item_IDs
T100	I1, I2, I5
T200	I2, I4
T300	I2, I3
T400	I1, I2, I4
T500	I1, I3
T600	I2, I3
T700	I1, I3
T800	I1, I2, I3, I5
T900	I1, I2, I3

For example,

Consider a frequent itemset {I1, I2, I5}

possible subsets are:

{I1, I2}, {I1, I5}, {I2, I5}, {I1}, {I2}, {I5} and the possible association rules are:

$$I1 \wedge I2 \Rightarrow I5, \quad confidence = 2/4 = 50\%$$
$$I1 \wedge I5 \Rightarrow I2, \quad confidence = 2/2 = 100\%$$
$$I2 \wedge I5 \Rightarrow I1, \quad confidence = 2/2 = 100\%$$
$$I1 \Rightarrow I2 \wedge I5, \quad confidence = 2/6 = 33\%$$
$$I2 \Rightarrow I1 \wedge I5, \quad confidence = 2/7 = 29\%$$
$$I5 \Rightarrow I1 \wedge I2, \quad confidence = 2/2 = 100\%$$

Here, I1^I2 =>I5 Means,

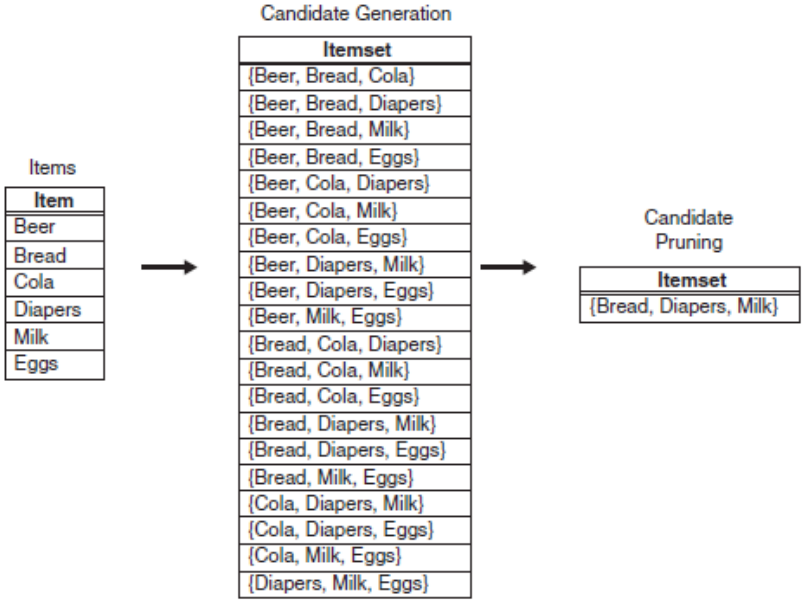
$$\frac{\text{Number of transaction containing I1,I2,I5}}{\text{Number of transactions containing I1,I2}} = 2/4 = 50\%$$

If, Minimum confidence is say, 70%, then 2nd, 3rd and 6th association rules will be retained.

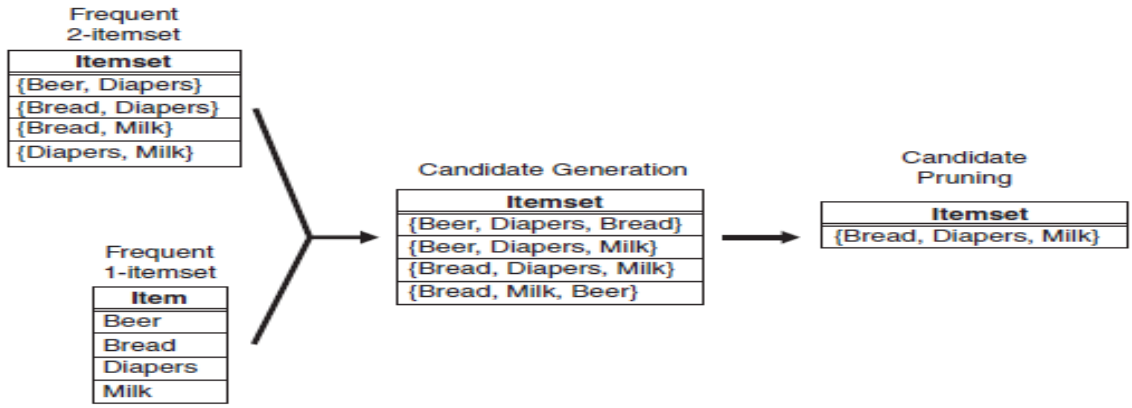
Que 5: What are the procedures for generating candidates in apriori algorithm?
(Or)
What are the techniques used in apriori_gen?

Ans:

Brute force method: All possible K- Itemsets are generated from k-1th item sets and infrequent k-item candidates are deleted.



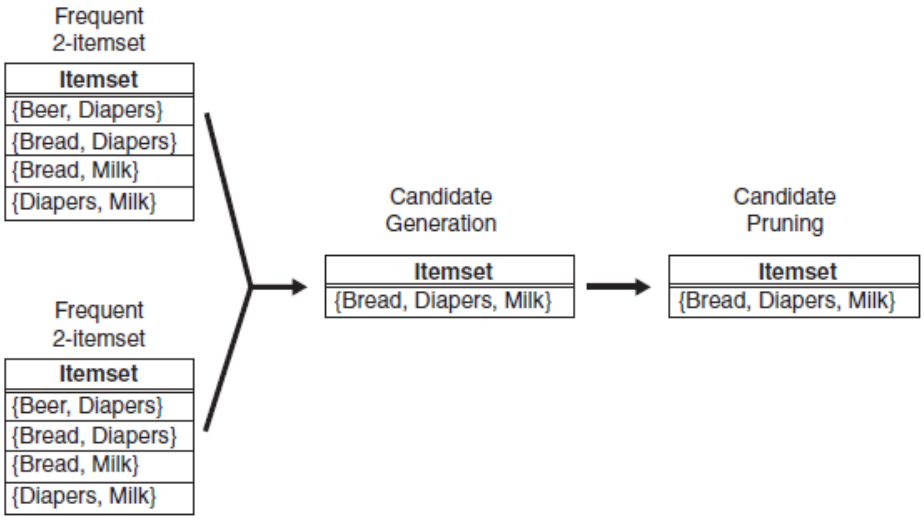
F_{k-1} * F_K method: As the name suggests, frequent 'K-1' item set is combined with 1 itemset to generate 'K' item-candidate set.



$F_{k-1} * F_{k-1}$ method: Frequent K-1th itemset is combined with itself to generate C_k.

$$\begin{aligned}\text{Join: } C_3 = L_2 \bowtie L_2 &= \{\{I1, I2\}, \{I1, I3\}, \{I1, I5\}, \{I2, I3\}, \{I2, I4\}, \{I2, I5\}\} \bowtie \\ &\quad \{\{I1, I2\}, \{I1, I3\}, \{I1, I5\}, \{I2, I3\}, \{I2, I4\}, \{I2, I5\}\} \\ &= \{\{I1, I2, I3\}, \{I1, I2, I5\}, \{I1, I3, I5\}, \{I2, I3, I4\}, \{I2, I3, I5\}, \{I2, I4, I5\}\}.\end{aligned}$$

In the above example two item sets are combined if starting elements in itemset are same and only last element in both item sets are different.



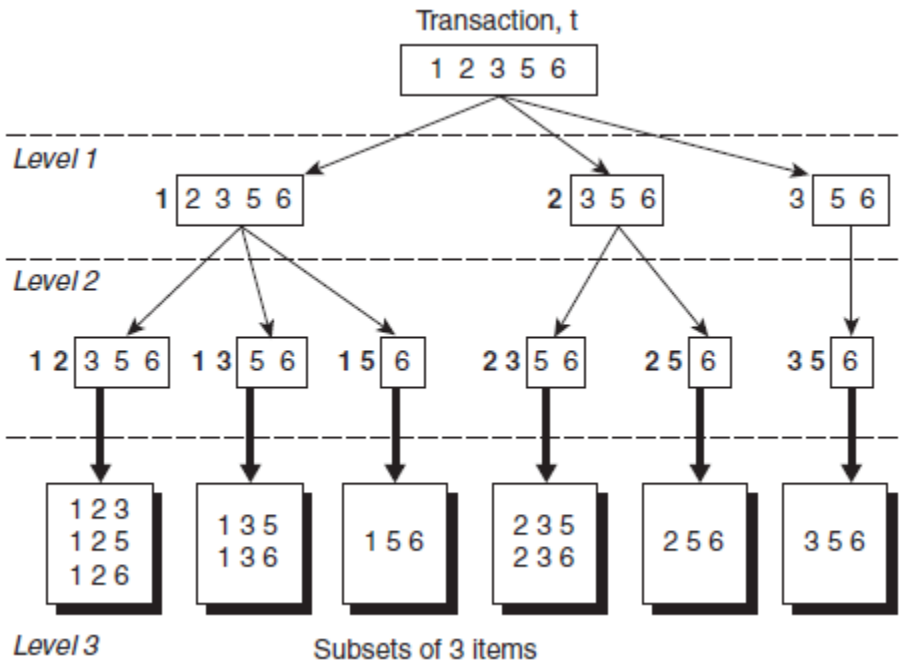
Que 6: What are the methods for generating support counts for candidates?
(Or)
Discuss about support counting using Hash Tree.

3 methods for generating support counts for candidates.

1) **Compare itemsets against each transaction**
2) **Enumerate the itemsets**
3) **Support counting using a hash tree.**

1) **Compare itemsets against each transaction:** Compare itemsets against each transaction and increment support count of the candidates contained in the transaction.

2) **Enumerate the itemsets:**

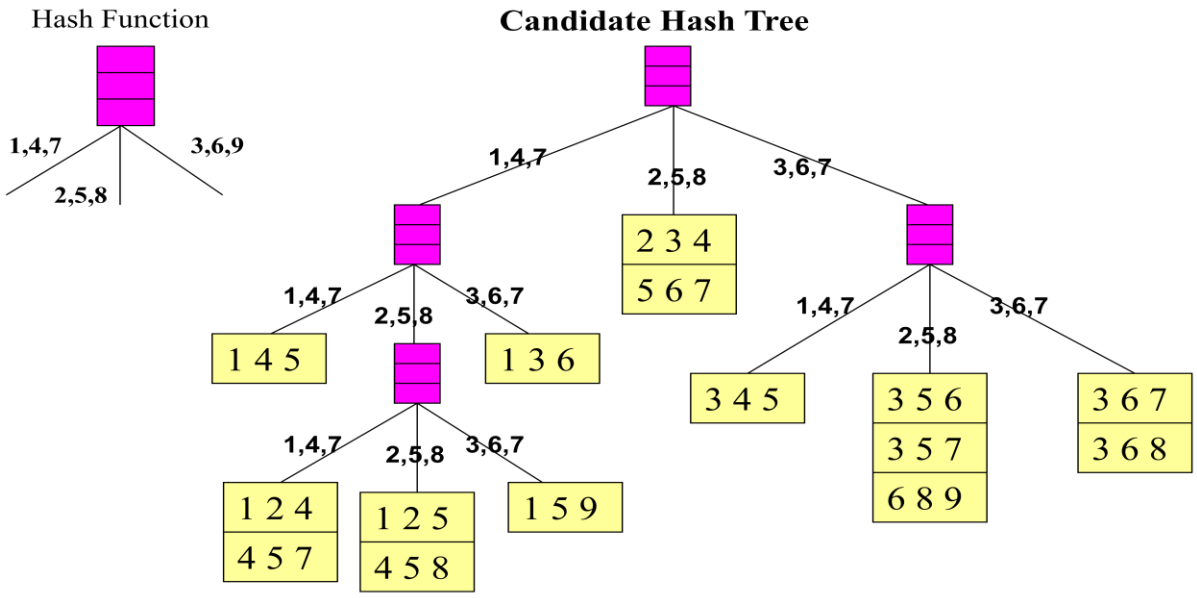


Arrange elements in increasing order of their occurrence in the transaction database.

In level 1, Take 1, 2, and 3 as prefix. And in level 2, take each set from level one and take the next elements are prefixes. This process repeats until 3 itemsets are generated.

3) Support counting using a hash tree:

{1 4 5}, {1 2 4}, {4 5 7}, {1 2 5}, {4 5 8}, {1 5 9}, {1 3 6}, {2 3 4}, {5 6 7}, {3 4 5}, {3 5 6}, {3 5 7}, {6 8 9}, {3 6 7}, {3 6 8}



Here, Items are stored in different buckets called as hash tables.

In hash tree, user needs to specify two things:

1) The hash function: here it is h(p)=p mod 3
2) Max leaf size: max number of itemsets stored in a leaf node (if number of candidate itemsets exceeds max leaf size, split the node)

For example,

1 mod 3=1	2 mod 3=2	3 mod 3=0
4 mod 3=1	5 mod 3=2	6 mod 3=0
7 mod 3=1	8 mod 3=2	9 mod 3=0

- In level 1
  - The itemsets containing 1, 4 and 7 as first elements are mapped to left side of the tree.

    - The itemsets containing 2, 5, and 8 as first elements are mapped to middle of the tree.
    - The itemsets containing 3, 6, and 9 as first element are mapped to right side of the tree.

- In level 2
    - Consider the leftmost sub tree, and apply the same hash function applied at level one.
    - The itemsets containing 1, 4 and 7 as second elements are mapped to left side of the tree.
    - The itemsets containing 2, 5, and 8 as second elements are mapped to middle of the tree.
    - The itemsets containing 3, 6, and 9 as second element are mapped to right side of the tree.

Repeat the same process for every sub tree until 3 itemsets are generated. Now, the candidate itemsets stored at leaf node are compared against each transaction. If a candidate is subset of transaction, its support count is incremented.

Que 7: Write the computational complexity of apriori algorithm.

- Choice of minimum support threshold
    - lowering support threshold results in more frequent itemsets
    - this may increase number of candidates and max length of frequent itemsets
- Dimensionality (number of items) of the data set
    - more space is needed to store support count of each item
    - if number of frequent items also increases, both computation and I/O costs may also increase
- Size of database

  - since Apriori makes multiple passes, run time of algorithm may increase with number of transactions
- Average transaction width
  - transaction width increases with denser data sets
  - This may increase max length of frequent itemsets and traversals of hash tree (number of subsets in a transaction increases with its width)

Que 8: What are the advantages and disadvantages of Apriori algorithm?

Disadvantages of Apriori

- The candidate generation could be extremely slow (pairs, triplets, etc.).
- The counting method iterates through all of the transactions each time.
- Constant items make the algorithm a lot heavier.
- Huge memory consumption

Advantages of Apriori

The Apriori Algorithm calculates more sets of frequent items.

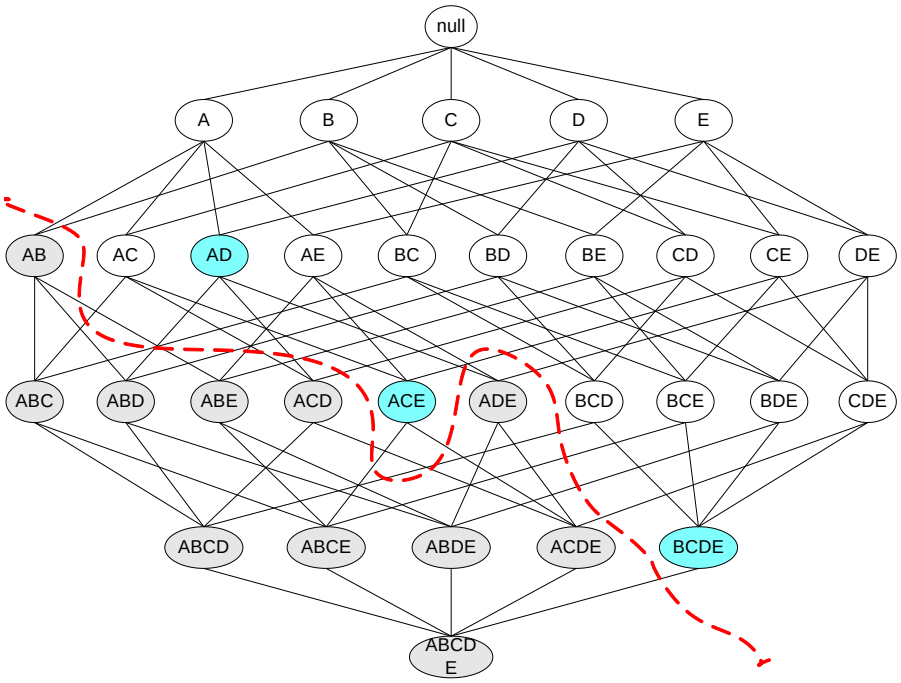
Que 9: Write a short note on:

1) Maximal frequent itemset

2) Closed itemset

3) Closed frequent itemset

Maximal frequent itemset: An itemset is maximal frequent if none of its immediate supersets is frequent.



In the above figure, consider 'AD' as frequent itemset, then if all supersets containing 'AD' are infrequent then 'AD' is called maximal frequent itemset.

Closed itemset: An itemset is closed if none of its immediate supersets has the same support as the itemset

Itemset	Support
{A}	4
{B}	5
{C}	3
{D}	4
{A,B}	4
{A,C}	2
{A,D}	3
{B,C}	3
{B,D}	4
{C,D}	3

- Consider itemset {A},
    - {A} has the support count of 4
    - And, {A,B} has also the support count of 4
    - So, {A} is not a closed itemset.

Closed frequent itemset: If an itemset is closed and its support count is greater than min_Sup, then it is called closed frequent item set.

Que: How to generate frequent itemsets without candidates?
(Or)
Explain FP Growth algorithm

Apriori uses a generate-and-test approach – generates candidate itemsets and tests if they are frequent

1. Generation of candidate itemsets is expensive (in both space and time.
2. Support counting is expensive
    - Subset checking (computationally expensive)
    - Multiple Database scans (I/O)

FP-Growth: allows frequent itemset discovery without candidate itemset generation. This is a two step approach:

- Build a compact data structure called the FP-tree
- Extracts frequent itemsets directly from the FP-tree

Algorithm for FP-Tree

Step 1:

- Scan DB once, find frequent 1-itemset
- Sort frequent items in frequency descending order, f-list
- Scan DB again, construct FP-tree
    - FP-Growth reads 1 transaction at a time and maps it to a path
    - Fixed order is used, so paths can overlap when transactions share items

Step 2:

- Generating conditional pattern base
    - Starting at the frequent item header table in the FP-tree
    - Traverse the FP-tree by following the link of each frequent item
    - Accumulate all of *transformed prefix paths* of the item to form *its* conditional pattern base
- For each conditional pattern-base

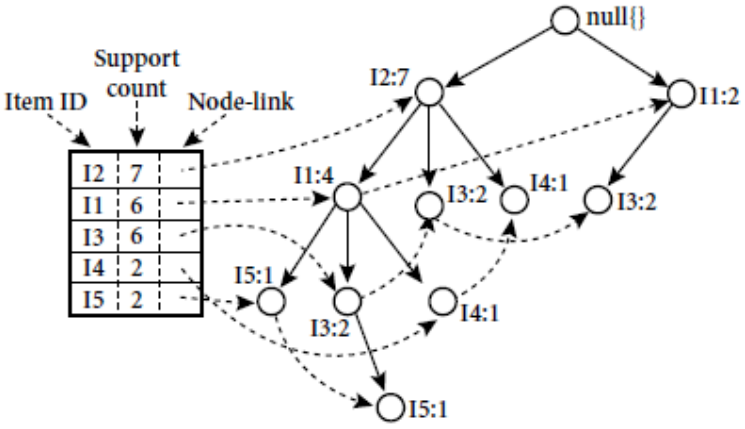
- Accumulate the count for each item in the conditional pattern base
- Construct the FP-tree for the frequent items of the pattern base

- Frequent itemsets is, all the combinations of its sub-paths, each of which is a frequent pattern

Example of FP Growth

TID	List of item_IDs
T100	I1, I2, I5
T200	I2, I4
T300	I2, I3
T400	I1, I2, I4
T500	I1, I3
T600	I2, I3
T700	I1, I3
T800	I1, I2, I3, I5
T900	I1, I2, I3

The first scan of the database is the same as Apriori, which derives the set of frequent items (1-itemsets) and their support counts (frequencies). Let the minimum support count be 2. The set of frequent items is sorted in the order of descending support count. This resulting set or *list* is denoted *L*. Thus, we have *L* = {{I2: 7}, {I1: 6}, {I3: 6}, {I4: 2}, {I5: 2}}.



An FP-tree registers compressed, frequent pattern information.

Mining the FP-tree by creating conditional (sub-)pattern bases.

Item	Conditional Pattern Base	Conditional FP-tree	Frequent Patterns Generated
I5	{{I2, I1: 1}, {I2, I1, I3: 1}}	⟨I2: 2, I1: 2⟩	{I2, I5: 2}, {I1, I5: 2}, {I2, I1, I5: 2}
I4	{{I2, I1: 1}, {I2: 1}}	⟨I2: 2⟩	{I2, I4: 2}
I3	{{I2, I1: 2}, {I2: 2}, {I1: 2}}	⟨I2: 4, I1: 2⟩, ⟨I1: 2⟩	{I2, I3: 4}, {I1, I3: 4}, {I2, I1, I3: 2}
I1	{{I2: 4}}	⟨I2: 4⟩	{I2, I1: 4}

Mining of the FP-tree is summarized in the above table.

//Optional

- We first consider I5, which is the last item in *L*, rather than the first. I5 occurs in two branches of the FP-tree. The paths formed by these branches are {(I2, I1, I5: 1) and (I2, I1, I3, I5: 1)}. Therefore, considering I5 as a suffix, its corresponding two prefix paths are (I2, I1: 1) and (I2, I1, I3: 1), which form its conditional pattern base. Its conditional FP-tree contains only a single path, {I2: 2, I1: 2}; I3 is not included because its support count of 1 is less than the minimum support count. The single path generates all the combinations of frequent patterns: {I2, I5: 2}, {I1, I5: 2}, {I2, I1, I5: 2}.

- For I4, its two prefix paths form the conditional pattern base, {{I2 I1: 1}, {I2: 1}}, which generates a single-node conditional FP-tree, {I2: 2}, and derives one frequent pattern, {I2, I4: 2}.

- Similar to the above analysis, I3's conditional pattern base is {{I2, I1: 2}, {I2: 2}, {I1: 2}}. its conditional FP-tree has two branches, (I2: 4, I1: 2) and (I1: 2), as shown in Figure, which generates the set of patterns, {{I2, I3: 4}, {I1, I3: 4}, {I2, I1, I3: 2}}.

- Finally, I1's conditional pattern base is {{I2: 4}}, whose FP-tree contains only one node, (I2: 4), which generates one frequent pattern, {I2, I1: 4}.

Another version of algorithm (As in Kamber text book, students can learn any one of the two algorithm)

Algorithm: **FP_growth.** Mine frequent itemsets using an FP-tree by pattern fragment growth.

Input:

- D, a transaction database;
- *min_sup*, the minimum support count threshold.

Output: The complete set of frequent patterns.

Method:

1. The FP-tree is constructed in the following steps:
   (a) Scan the transaction database D once. Collect F, the set of frequent items, and their support counts. Sort F in support count descending order as L, the *list* of frequent items.
   (b) Create the root of an FP-tree, and label it as "null." For each transaction *Trans* in D do the following. Select and sort the frequent items in *Trans* according to the order of L. Let the sorted frequent item list in *Trans* be $[p|P]$, where p is the first element and P is the remaining list. Call insert_tree$([p|P], T)$, which is performed as follows. If T has a child N such that $N.item\text{-}name = p.item\text{-}name$, then increment N's count by 1; else create a new node N, and let its count be 1, its parent link be linked to T, and its node-link to the nodes with the same *item-name* via the node-link structure. If P is nonempty, call insert_tree(P, N) recursively.
2. The FP-tree is mined by calling FP_growth$(FP_tree, null)$, which is implemented as follows.

```
procedure FP_growth(Tree, α)
(1)  if Tree contains a single path P then
(2)    for each combination (denoted as β) of the nodes in the path P
(3)      generate pattern β∪α with support_count = minimum support count of nodes in β;
(4)  else for each a_i in the header of Tree {
(5)    generate pattern β = a_i ∪ α with support_count = a_i.support_count;
(6)    construct β's conditional pattern base and then β's conditional FP_tree Tree_β;
(7)    if Tree_β ≠ ∅ then
(8)      call FP_growth(Tree_β, β); }
```

The FP-growth algorithm for discovering frequent itemsets without candidate generation.

Que: What are the advantages and disadvantages of FP-Tree?

Advantages:

- no candidate generation, no candidate test
- compressed database: FP-tree structure
- no repeated scan of entire database
- basic ops—counting local freq items and building sub FP-tree, no pattern search and matching
- leads to focused search of smaller databases

Disadvantage:

Parallelization of FP Growth technique may end with bottlenecks in shared memory.

Que: Compare and contrast Apriori and FP Growth.

Algorithm	Technique	Runtime	Memory usage	Parallelizability
Apriori	Generate singletons, pairs, triplets, etc.	Candidate generation is extremely slow. Runtime increases exponentially depending on the number of different items.	Saves singletons, pairs, triplets, etc.	Candidate generation is very parallelizable
FP-Growth	Insert sorted items by frequency into a pattern tree	Runtime increases linearly, depending on the number of transactions and items	Stores a compact version of the database.	Data are very inter-dependent, each node needs the root.

UNIT 5

Cluster Analysis: Basic Concepts and Algorithms: Overview, What Is Cluster Analysis? Different Types of Clustering, Different Types of Clusters; K-means: The Basic K-means Algorithm, K-means Additional Issues, Bisecting K-means, Strengths and Weaknesses; Agglomerative Hierarchical Clustering: Basic Agglomerative Hierarchical Clustering

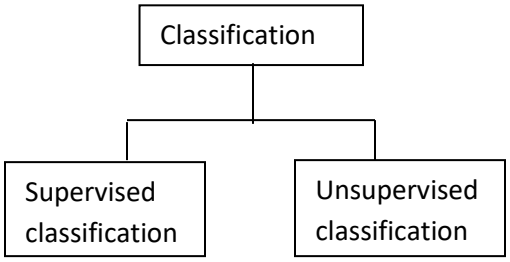
Algorithm DBSCAN: Traditional Density Center-Based Approach, DBSCAN Algorithm, Strengths and Weaknesses. (Tan &Vipin)

PART 1

Introduction

Que 1: What is Cluster Analysis? What are the applications of cluster analysis?

Classification is divided into two categories:



Supervised classification is simply known as classification in which the unknown class labels are identified.

Unsupervised classification is known as cluster analysis.

Clustering

Grouping of similar data items together is known as clustering. This is mainly used for summarization of data.

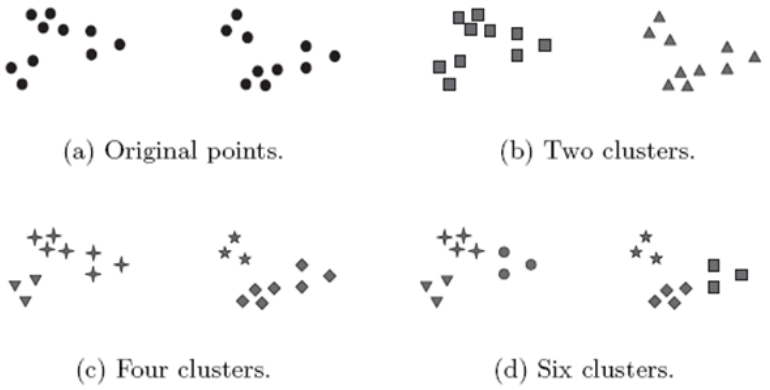


Fig: Different ways of clustering same set of points

Cluster analysis

The process of grouping data basing on the information found is known as cluster analysis. The main goal of cluster analysis is that the objects within a group be similar to one another and different from the objects in other groups.

Applications of clustering

1. **Information retrieval**
   Clustering can be used to group search results into a smaller number of clusters. For example, app like news in shots can cluster articles like crime, politics, prime minister.
2. **Climate**
   Cluster analysis has been applied to find patterns about related climatic conditions. And cluster the locations based on climate (like, volcanic area, earthquake areas, and Avalanche areas)
3. **Medicine**
   Clustering is used in medical to identify the diseases basing on the symptoms. And, patients can be clustered based on their diseases.
4. **Biology**
   Clustering is used in biology to analyze the large amount of genetic information. And, virus and bacteria can be grouped based on their behavior.
5. **Business**
   Clustering can be used to group the list of customers basing on the buying trends and sales of various products.

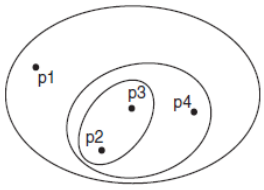
Que 2 what are different types of clustering techniques?

An entire collection of clusters is commonly referred to as **clustering**.

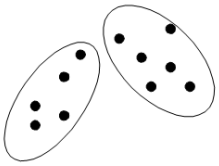
The various types of clusterings are as follows:

1. **Hierarchical versus partitional clustering**

   Hierarchical clustering: In this type of clustering, the set of clusters are nested clusters that are organized in the form of a tree known as dendrogram. Each node in the tree is the union of its children and root of the tree is the cluster containing all the objects.



   Partitional clustering: In this type of clustering, the set of clusters are unnested clusters. It is simply a division of the set of data objects into non-overlapping subsets such that each data object is assigned only to one subset.



2. **Exclusive versus overlapping versus fuzzy clustering**

   Exclusive clustering: In this type of clustering, each data object is exactly assigned to only one cluster.

   Overlapping clustering: There are some cases where one data object is assigned to more than one cluster. Such situations can be handled by overlapping clustering also known as **non-exclusive clustering** i.e. a data object can be assigned to more than one cluster.

   Fuzzy clustering: In this type of clustering, every data object can be assigned to every cluster, but basing on the membership function or weight which lies between 0 and 1. The value 0 implies that it doesn't belong to any cluster and 1 implies that it belongs to a cluster.

3. Complete versus partial clustering

Complete clustering: In this type of clustering, every data object is assigned to a cluster even if the data has some outliers or noise.

Partial clustering: The data object is assigned to a cluster only if it doesn't contain any noise or outliers.

Que3: What are different types of clusters?

1. Well-Separated clusters

A cluster is a set of objects in which each object is closer to every other object in the cluster than to any object not in the cluster i.e. the distance between any two points in different groups is larger than the distance between points in a group.

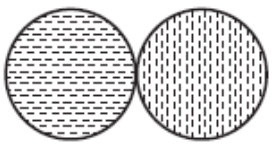
(a) Well-separated clusters. Each point is closer to all of the points in its cluster than to any point in another cluster.

2. Prototype based cluster

A cluster is a set of objects in which each object is closer to the prototype that defines the cluster than to the prototype of any other cluster. Prototype based cluster is also known as **center based cluster.**

For continuous attribute, the prototype of a cluster is centroid, i.e., the average of all points in the cluster.

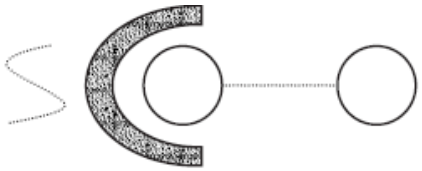
For categorical attribute, the prototype is medoid, i.e., the most representative point of a cluster.



(b) Center-based clusters. Each point is closer to the center of its cluster than to the center of any other cluster.

3. Graph based cluster

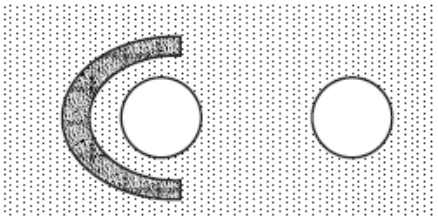
If the data is represented as a graph, where the nodes are objects and the links represent connections among objects, then a cluster can be defined as a connected component; i.e., group of objects that are connected to one another, but that have no connections to objects outside the group. **Contiguity based cluster** is an example of graph based cluster.



(c) Contiguity-based clusters. Each point is closer to at least one point in its cluster than to any point in another cluster.

4. Density based clustering

A cluster is a dense region of points, which is separated by low-density regions, from other regions of high density. It is used when the clusters are irregular or intertwined, and when noise and outliers are present.



(d) Density-based clusters. Clusters are regions of high density separated by regions of low density.

K-Means

Que 4: Explain K-Means partitioning Technique with example.

(Or)

What is partitioning method? Describe any one partition based clustering algorithm.

(Or)

With a suitable example, explain K-Means Clustering algorithm.

(Or)

Consider five points {A1, A2, A3, A4, and A5} with the following coordinates as a two dimensional sample for clustering:

A1= (0.5, 2.5); A2= (0, 0); A3= (1.5, 1); A4= (5, 1); A5= (6, 2);

Illustrate the K-means partitioning algorithms using the above data set.

Ans:

k- Means defines a prototype in terms of a centroid, which is usually the mean of a group of points and is applied to objects in a continuous n-dimensional space.

Algorithm 8.1 Basic K-means algorithm.

1: Select K points as initial centroids.
2: **repeat**
3: Form K clusters by assigning each point to its closest centroid.
4: Recompute the centroid of each cluster.
5: **until** Centroids do not change.

Example1:

Consider five points {X_1,X_2,X_3,X_4,X_5} with the following coordinates as a two dimensional sample for clustering :

A_1= (0.5, 2.5); A_2= (0, 0); A_3= (1.5, 1); A_4= (5, 1); A_5= (6, 2)

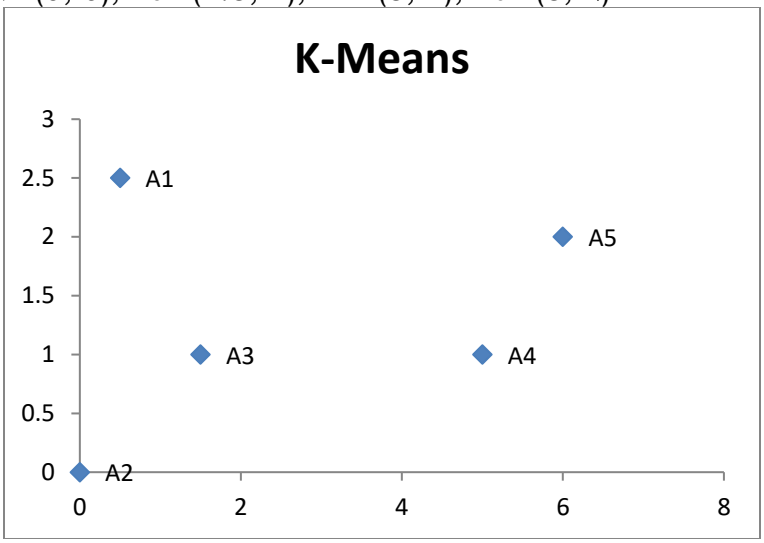


Figure 1

- Let us consider two random centroids 'A1' and 'A5' from above 5 points:
- Now measure distance from all elements to the two centroids. This can be done by using the formula $\sqrt{(\mathbf{xb}-\mathbf{xa})\mathbf{2}+(\mathbf{yb}-\mathbf{ya})\mathbf{2}}$

Calculate distance from all points to A1

Dist(A2,A1)= $\sqrt{(0-0.5)2+(0-2.5)2}$=2.54

Dist (A3,A1)= $\sqrt{(1.5-0.5)2+(1-2.5)2}$=1.802

Dist (A4,A1)= $\sqrt{(5-0.5)2+(1-2.5)2}$=4.743

Dist(A5,A1)= $\sqrt{(6-0.5)2+(2-2.5)2}$=5.52

Calculate distance from all points to A5

Dist(A1,A5)= $\sqrt{(0.5-6)2+(2.5-2)2}$=5.52

Dist (A2,A5)= $\sqrt{(0-6)2+(0-2)2}$=6.324

Dist (A3,A5)= $\sqrt{(1.5-6)2+(1-2)2}$=4.60

Dist (A4,A5)= $\sqrt{(5-6)2+(1-2)2}$=1.414

	A1	A5	Cluster to map
A1	0	5.52	A1
A2	2.54	6.324	A1
A3	1.802	4.60	A1
A4	4.743	1.414	A5
A5	5.52	0	A5

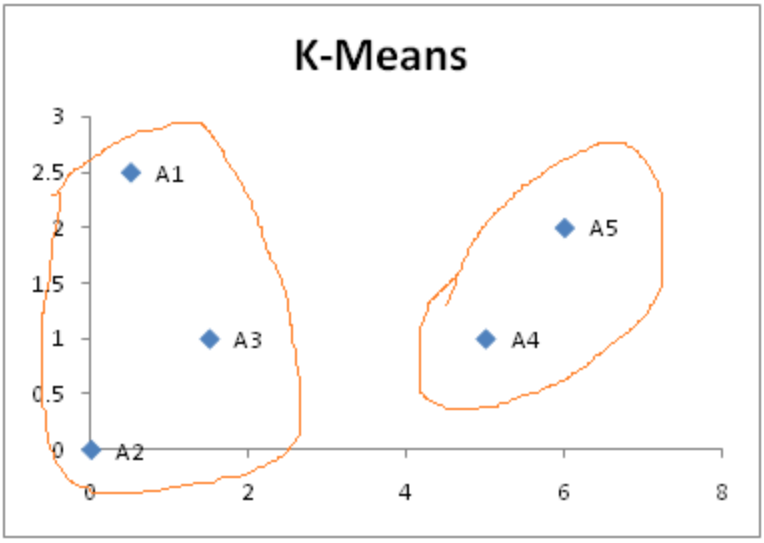


Figure 2

- Now 'A1','A2','A3' belongs to one cluster and 'A4' and 'A5' belongs to another cluster.
- Now, Calculate the average of A1, A2, and A3

  X-Axis= (0.5+0+1.5)/3=0.666

  Y-Axis = (2.5+0+1)/3=1.16

  Similarly calculate the average of 'A4' and 'A5' =

  X-Axis= (5+6)/2=5.5

  Y-Axis= (2+1)/2=1.5

  New centroids are,

  P= (0.66, 1.16)

  Q= (5.5, 1.5)
- Repeat same process, Calculate distance from all points to 'P' and 'Q':

  Calculate distance from all points to 'P'

  Dist(A1,P)= $\sqrt{(0.5-0.66)2+(2.5-1.16)2)}$=1.349

  Dist (A2,P)= $\sqrt{(0-0.66)2+(0-1.16)2)}$=1.33

  Dist(A3,P)= $\sqrt{(1.5-0.66)2+(1-1.16)2)}$=0.85

  Dist (A4,P)= $\sqrt{(5-0.66)2+(1-1.16)2)}$=4.34

  Dist (A5,P)= $\sqrt{(6-0.66)2+(2-1.16)2)}$=5.40

  Calculate distance from all points to Q.

  Dist(A1,Q)= $\sqrt{(0.5-5.5)2+(2.5-1.5)2)}$ =5.09

  Dist(A2,Q)= $\sqrt{(0-5.5)2+(0-1.5)2)}$=5.70

  Dist(A3,Q)= $\sqrt{(1.5-5.5)2+(1-1.5)2)}$=4.031

  Dist(A4,Q)= $\sqrt{(5-5.5)2+(1-1.5)2)}$=0.707

  Dist(A5,Q)= $\sqrt{(6-5.5)2+(2-1.5)2)}$=0.707

	P	Q	Cluster to map
A1	1.349	5.09	P
A2	1.33	5.70	P
A3	0.85	4.031	P
A4	4.34	0.707	Q
A5	5.40	0.707	Q

- Since ‘A1’,’A2’ and ‘A3’ belongs to one cluster and ‘A4’ and ‘A5’ belongs to another cluster. The process of clustering can be halted.

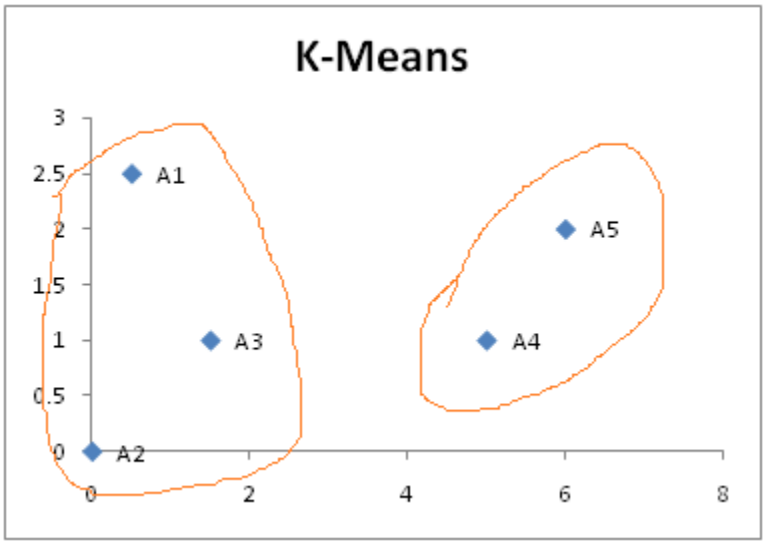


Figure 3

- Note: The process of making cluster is halted because we got the same type of cluster in 1st iteration and 2nd iteration. And there is no movement between elements in clusters.

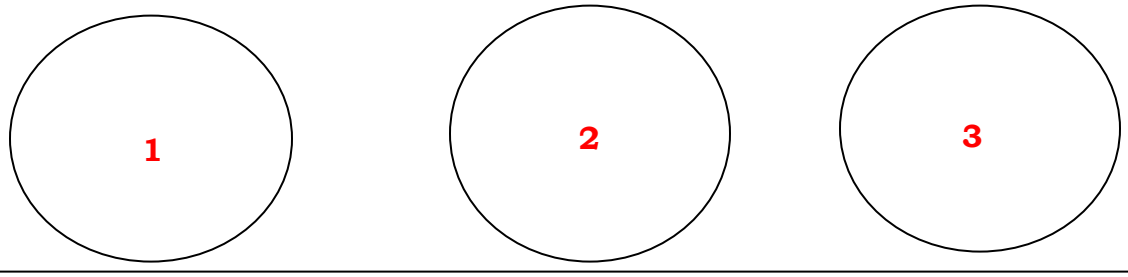
Example 2

(I recommend students to write this example in exam only if you have very-very less time)

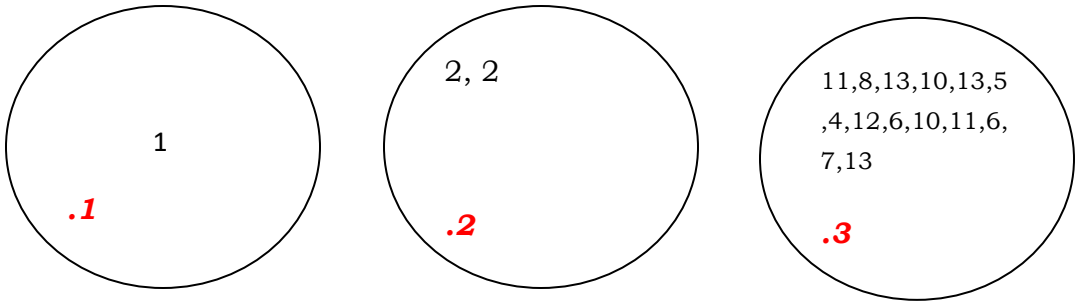
Consider samples

11,8,13,10,13,5,4,12,6,10,11,3,6,7,13,2,2,2,1,1

Consider 3 random centroids 1, 2, 3



Now, Move all elements to nearest centroid



Mean: 1 **Mean: 2** **Mean 3: 8.8**

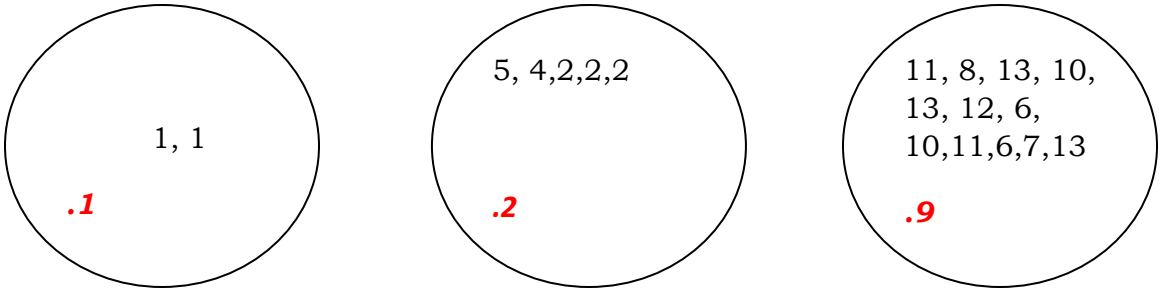
Now calculate the average of all clusters to find the new centroids:

Calculate the mean of 1st, 1=1=new centroid

Calculate the mean 2nd centroid 2, 2=2=new centroid

Calculate the mean 3rd centroid: 11,8,13,10,13,5,4,12,6,10,11,6,7,13,3=9=new centroid

Again, Move the elements to new centroids



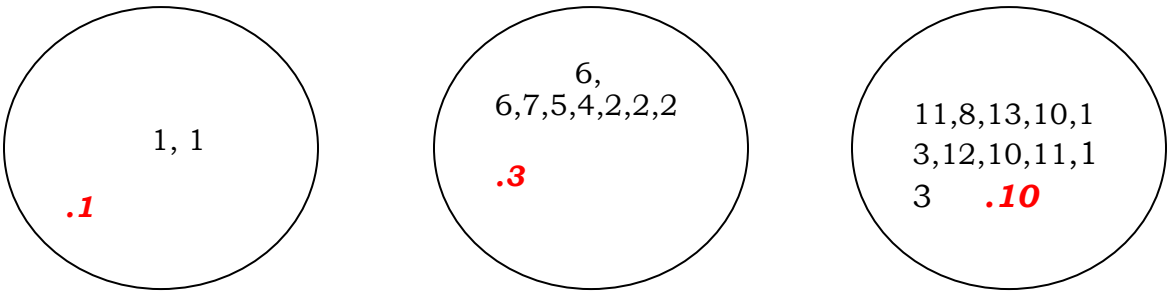
Mean: 1 **Mean: 3** **Mean: 10**

Centroid 1: 1, 1=1

Centroid 2: 5, 4, 2, 2, 2=3

Centroid 3: 11, 8, 13, 10, 13, 12, 6, 10, 11, 6, 7, 13=10

Again, Move the elements to new centroid



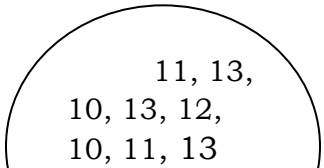
Mean: 1 **Mean: 4.25** **Mean: 11.22**

Centroid 1: 1, 1=1

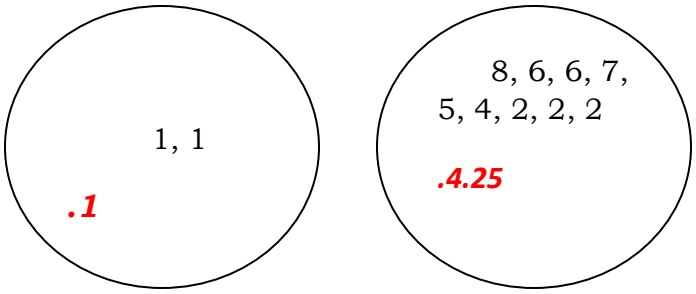
Centroid 2: 6, 6,7,5,4,2,2,2 =4.25

Centroid 3: 11,8,13,10,13,12,10,11,13 =11.22

Again, Move the elements to nearest centroid



EP



Centroid 1: 1, 1=1

Centroid 2: 8, 6, 6, 7, 5, 4, 2, 2, 2=4.66

Centroid 3: 11, 13, 10, 13, 12, 10, 11, 13=11.62

Since, all elements are in nearest centroid. So, Clustering can be stopped.

Que 5: What are the additional Issues of K-Means?

K-means: Additional issues

1. **Handling empty clusters**

   There are some cases where a cluster has only a single point i.e.; its centroids. In such case we either need to replace the centroids or eliminate it.

2. **Outliers**

   There are two ways to handle outliers. They are:

   a. **Preprocessing**

   b. **Post processing**

   In preprocessing technique, we need to first identify the outliers, remove the outliers and perform clustering. There are some cases where outliers are most important, such as, in fraud detections. In such cases, we use post processing techniques, i.e.; first cluster the data objects and later on remove the outliers.

3. **Reducing the SSE with post processing**

   SSE stands for sum of squared error.

$$\text{SSE} = \sum_{i=1}^{K} \sum_{\mathbf{x} \in C_i} dist(\mathbf{c}_i, \mathbf{x})^2$$

$$\mathbf{c}_i = \frac{1}{m_i} \sum_{\mathbf{x} \in C_i} \mathbf{x}$$

Low SSE of clusters means good clusters. There are some techniques to reduce SSE, they are as follows:

a. **Decreasing SSE by increasing number of clusters**
    i. **Split a cluster:** The cluster with the largest SSE is usually chosen and further splitted.
    ii. **Introduce a new cluster centroid:** The SSE can be decreased by replacing an old centroid with the new centroid. Or, remove an element from cluster and make it as new centroid. Now map the elements in the cluster to new centroid and make a new cluster.
b. **Decreasing SSE by decreasing number of clusters**
    i. **Disperse a cluster:** a cluster with high SSE can be dispersed and those points can be reassigned to other clusters.
    ii. **Merge two clusters:** The clusters with closest centroids are merged in order to reduce SSE.

4. Updating Centroids incrementally

In the basic K-means algorithm, centroids are updated after all points are assigned to a centroid. An alternative is to update the centroids after each assignment (incremental approach)

a. Each assignment updates centroids
b. More expensive
c. Introduces an order dependency
d. Never get an empty cluster
e. Can use “weights” to change the impact

Que 6: What is bisecting K-Means? Explain with Algorithm.

Please note: Here, there are two algorithms. 1st one is from text book and second one is a simplified version. Students can read which ever algorithm they fell easy.

This is an extension to k- means. To obtain k-clusters, split the set of all points into two clusters, select one of these clusters with large SSE(Sum of square error) to split until k-clusters have been produced. The procedure for splitting into clusters is same for k- Means and bisecting k-Means.

The algorithm for bisecting k-Means is as follows:

Algorithm 3 Bisecting K-means Algorithm.

1: Initialize the list of clusters to contain the cluster containing all points.
2: **repeat**
3: Select a cluster from the list of clusters
4: **for** $i = 1$ to *number_of_iterations* **do**
5: Bisect the selected cluster using basic K-means
6: **end for**
7: Add the two clusters from the bisection with the lowest SSE to the list of clusters.
8: **until** Until the list of clusters contains K clusters

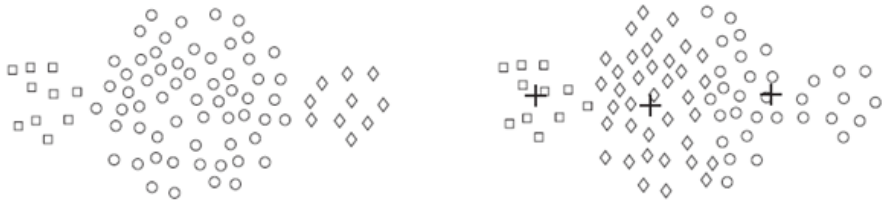
1. Repeat
2. Choose the parent cluster to be split C.
3. Repeat
4. Select two centroids at random from C.
5. Map all elements in the cluster C to nearest centroid
6. Recompute centroids by calculating mean of all elements of cluster and have new centroid assignment. (Apply K-Means on two centroids)
7. Calculate SSE for the 2 sub clusters. Choose the subclusters with maximum SSE
8. Consider the cluster with large SSE as new parent.
9. Again split the new parent
10. Until K Clusters are obtained

Que7 : Write a brief note on K-means and Different Types of Clusters.

k- Means has some difficulties when clustering have non-spherical shapes, different sizes and densities.

1. K- Means with clusters of different sizes

In the below fig, one of the cluster is much larger than the other two clusters, hence larger cluster is broken and combined with one of the smaller cluster.

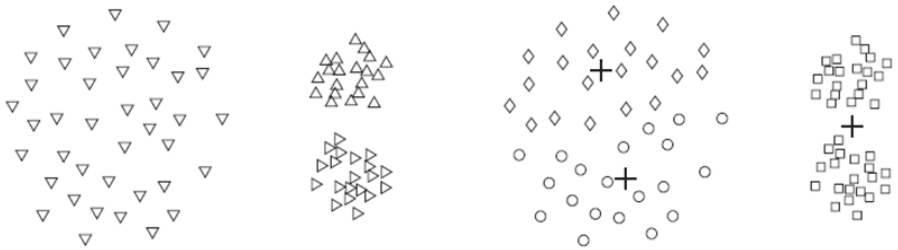


(a) Original points. (b) Three K-means clusters.

Fig: K-Means with clusters of different sizes

2. K- Means with clusters of different density

k- Means fails to find three natural clusters since the smaller cluster are much denser than the larger cluster. Therefore, larger cluster is divided into two smaller clusters.

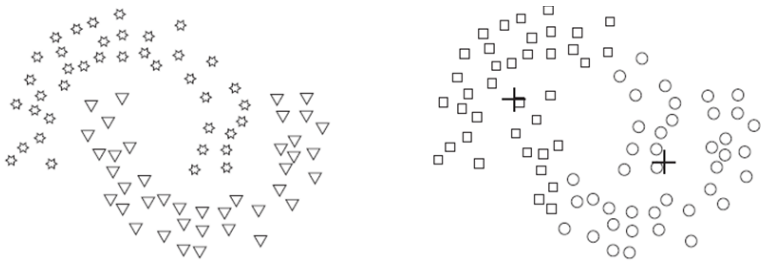


(a) Original points. (b) Three K-means clusters.

Fig: K- Means with clusters of different density

3. K- Means with non-globular clusters

In the above fig, the two natural clusters are combined, since the shapes of clusters are non- globular.



(a) Original points. (b) Two K-means clusters.

Fig: K- Means with non-globular clusters

Que 8: Write the Strengths, Weaknesses, Time and space complexity of K-Means?

Strengths

1. It is simple and useful for all varieties of data types.
2. Even though multiple iterations are performed, it is quite efficient.
3. Bisecting k- means is also efficient.

Weakness

1. It cannot handle non-globular clusters, clusters of different sizes and densities even though it can find pure sub-clusters.
2. Outliers cannot be clustered.
3. K-means is restricted for the notion of centroid.

Time complexity

O(I*K*m*n)

Where,

I- Number of Iteration for making perfect clusters

K- Number of clusters

m- Number of attributes

n- Number of elements

Space complexity

O((m+K)n)

Que 9 : Write the procedure to handle document data for clustering.

K-Means is not restricted to points and numbers. K-Means can be used to handle documents. The documents can be represented as document term matrix as shown below.

	T1	T2	T3	T4	T5	T6	T7	T8
Doc1	2	0	4	3	0	1	0	2
Doc2	0	2	4	0	2	3	0	0
Doc3	4	0	1	3	0	1	0	1
Doc4	0	1	0	2	0	0	1	0
Doc5	0	0	2	0	0	4	0	0
Doc6	1	1	0	2	0	1	1	3
Doc7	2	1	3	4	0	2	0	2

Here, DOC = Document and

T1, T2,...., T8 are terms

For example, DOC5 has term T6 4 times.

The main objective of document clustering is to group similar documents. For example NEWS portal organizes article in crime related, politics, social cause, sports etc. The quality of a document cluster is represented by cohesion which is represented as:

$$\text{Total Cohesion} = \sum_{i=1}^{K} \sum_{\mathbf{x} \in C_i} cosine(\mathbf{x}, \mathbf{c}_i)$$

Here,

- K-Number of clusters
- Ci= Cluster and
- ci= Centroid of cluster
- x= document

PART 2

Agglomerative Hierarchical Clustering

Que 10: What do you mean by Hierarchical clustering? What are different types of hierarchal clustering? How hierarchal clusters are represented?

Hierarchical clustering

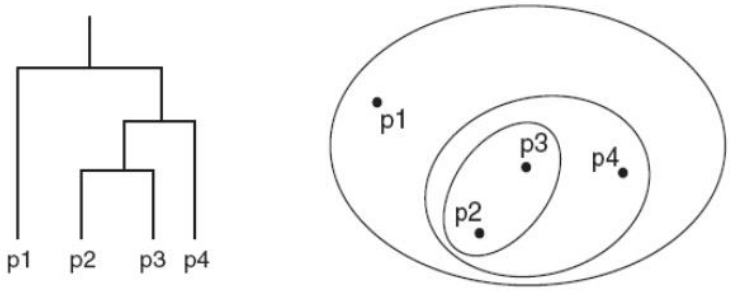
In this type of clustering, the set of clusters are nested clusters that are organized in the form of a tree known as dendrogram. Each node in the tree is the union of its children and root of the tree is the cluster containing all the objects.

There are two types of hierarchical clustering. They are:

1. Agglomerative hierarchical clustering
2. Divisive hierarchical clustering

Agglomerative hierarchical clustering

Start with the points as individual clusters and at each step, merge the closest pair of clusters.

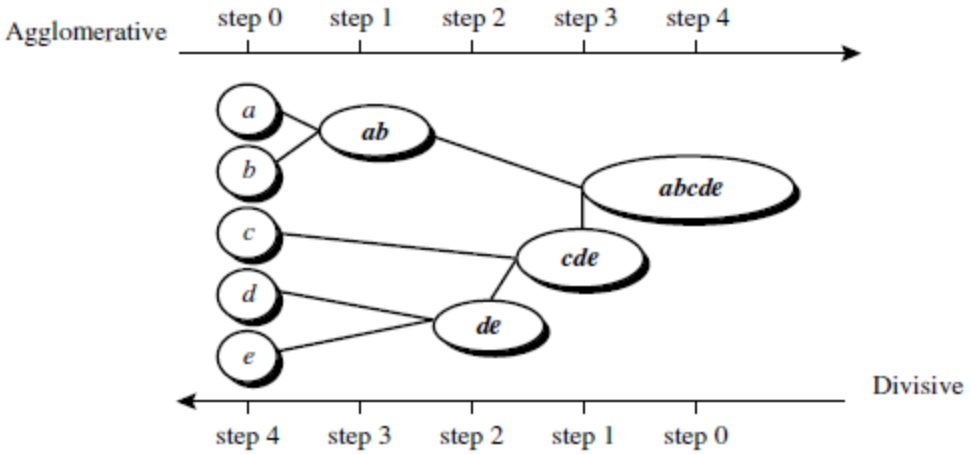


(a) Dendrogram. (b) Nested cluster diagram.

A hierarchical clustering of four points shown as a dendrogram and as nested clusters.

Divisive hierarchical clustering

Start with one i.e.; group all data objects into a single cluster, at each step, split a cluster until only single cluster of individual points remains.



Basic agglomerative hierarchical clustering algorithm

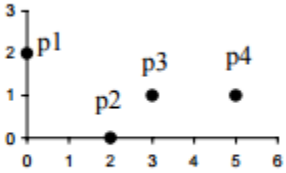
Starting with initial points as clusters, successively merge the two closest clusters until only one cluster remains.

Algorithm 8.3 Basic agglomerative hierarchical clustering algorithm.

1: Compute the proximity matrix, if necessary.
2: **repeat**
3: Merge the closest two clusters.
4: Update the proximity matrix to reflect the proximity between the new cluster and the original clusters.
5: **until** Only one cluster remains.

Que 11: What is meant by cluster proximity? Explain

Cluster proximity is defined as similarity or dissimilarity between elements or clusters. They are generally defined by proximity matrix.



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

	p1	p2	p3	p4
p1	0.000	2.828	3.162	5.099
p2	2.828	0.000	1.414	3.162
p3	3.162	1.414	0.000	2.000
p4	5.099	3.162	2.000	0.000

Four points and their corresponding data and proximity (distance) matrices.

Proximity matrix is a matrix containing distance between elements.

Que12: What are the methods for defining the proximity between the clusters?

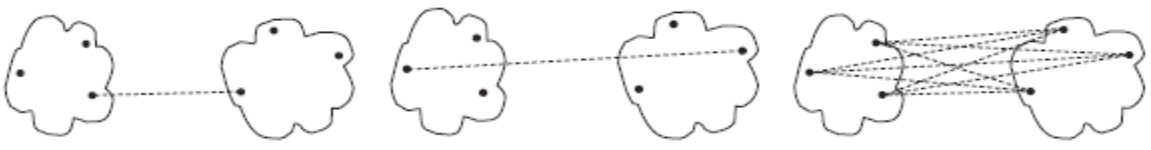
Proximity between clusters can be defined in three ways. They are:

1. Max
2. Min
3. Group average

Min defines cluster proximity between the closest two points that are in different clusters. This type of technique is also known as **single link technique.**

Max defines cluster proximity between the farthest two points that are in different clusters. This type of technique is also known as **complete link technique.**

Group average defines cluster proximity to be the average proximities of all pairs of points from different clusters.



(a) MIN (single link.) (b) MAX (complete link.) (c) Group average.

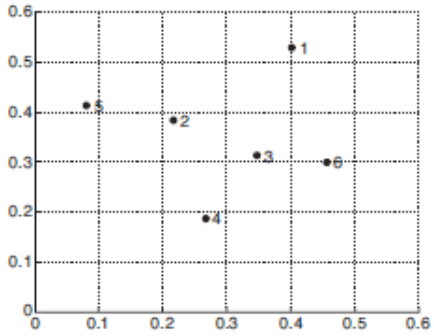
Graph-based definitions of cluster proximity

Que 13: How to define proximity between two clusters using MIN technique?

(Or)

Explain Single- Link hierarchical clustering with example?

Let us consider a data set with 6 data points.



Set of 6 two-dimensional points.

Point	x Coordinate	y Coordinate
p1	0.40	0.53
p2	0.22	0.38
p3	0.35	0.32
p4	0.26	0.19
p5	0.08	0.41
p6	0.45	0.30

xy coordinates of 6 points.

	p1	p2	p3	p4	p5	p6
p1	0.00	0.24	0.22	0.37	0.34	0.23
p2	0.24	0.00	0.15	0.20	0.14	0.25
p3	0.22	0.15	0.00	0.15	0.28	0.11
p4	0.37	0.20	0.15	0.00	0.29	0.22
p5	0.34	0.14	0.28	0.29	0.00	0.39
p6	0.23	0.25	0.11	0.22	0.39	0.00

Euclidean distance matrix for 6 points.

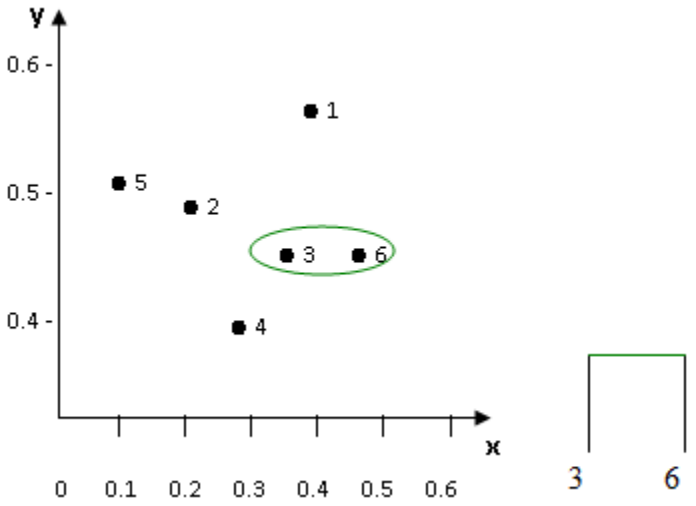
Min or single link technique

The proximity of two clusters is defined as the minimum of the distance between any two points I the two different clusters. The single link technique is good at handling non-elliptical shapes, but is sensitive to noise and outliers.

	P1	P2	P3	P4	P5	P6
P1	0	0.24	0.22	0.37	0.34	0.23
P2	0.24	0	0.15	0.20	0.14	0.25
P3	0.22	0.15	0	0.15	0.28	0.11
P4	0.37	0.20	0.15	0	0.29	0.22
P5	0.34	0.14	0.28	0.29	0	0.39
P6	0.23	0.25	0.11	0.22	0.39	0

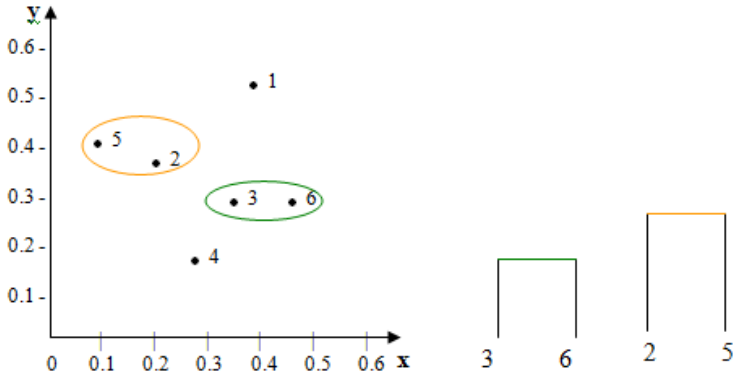
In the above table, p3, p6 is having the lowest distance, so merge the data points into a single cluster.

	P1	P2	P3P6	P4	P5
P1	0	0.24	0.22	0.37	0.34
P2	0.24	0	0.15	0.20	0.14
P3P6	0.22	0.15	0	0.15	0.28
P4	0.37	0.20	0.15	0	0.29
P5	0.34	0.14	0.28	0.29	0



The next lowest distance is for P2P5, so merge those two data points into a single cluster. The matrix obtains as follows:

	P1	P2P5	P3P6	P4
P1	0	0.24	0.22	0.37
P2P5	0.24	0	0.15	0.20
P3P6	0.22	0.15	0	0.15
P4	0.37	0.20	0.15	0



The distance between (p3, p6) and (p2, p5) would be calculated as follows:

dist((p3, p6), (p2, p5)) =MIN (*dist*(p3, p2) , *dist*(p6, p2), *dist*(p3, p5), *dist*(p6, p5))

=MIN (0.15, 0.25, 0.28, 0.39)

=0.15

dist((p3, p6), (p1)) =MIN (*dist*(p3, p1) , *dist*(p6, p1)

=MIN (0.22, 0.23)

=0.22

dist((p3, p6), (p4)) =MIN (*dist*(p3, p4) , *dist*(p6, p4)

=MIN (0.15, 0.22)

=0.15

dist((p2, p5), (p1)) =MIN (*dist*(p2, p1) , *dist*(p5, p1)

=MIN (0.24, 0.34)

=0.24

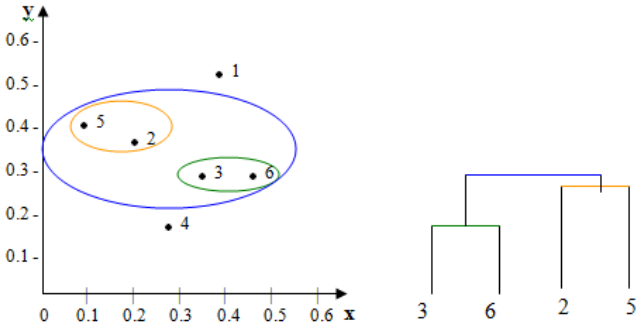
dist((p2, p5), (p4)) =MIN (*dist*(p2, p4) , *dist*(p5, p4)

=MIN (0.20, 0.29)

=0.20

So, looking at the last distance matrix above, we see that (p2, p5) and (p3, p6) have the smallest distance from all - 0.15. We also notice that p4 and (p3, p6) have the same distance - 0.15. In that case, we can pick either one. We choose (p2, p5) and (p3, p6). So, we merge those two in a single cluster, and re-compute the distance matrix.

	P1	P2P5P3P6	P4
P1	0	0.22	0.37
P2P5P3P6	0.22	0	0.15
P4	0.37	0.20	0



The distance between (P2, P5, P3, P6) and P1 would be calculated as follows:

dist((P2, P5, P3, P6), (p1)) =MIN (*dist*(p2, p1) , *dist*(p5, p1), *dist*(p3, p1) , *dist*(p6, p1))

=MIN (0.24, 0.34, 0.22, 0.23)

=0.22

The distance between (P2, P5, P3, P6) and P4 would be calculated as follows:

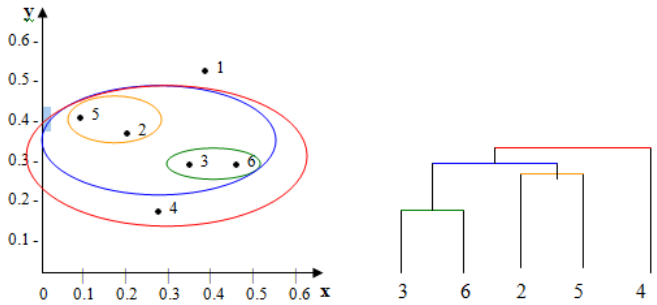
dist((P2, P5, P3, P6), (p4)) =MIN (*dist*(p2, p4) , *dist*(p5, p4), *dist*(p3, p4) , *dist*(p6, p4))

=MIN (0.20, 0.29, 0.15, 0.22)

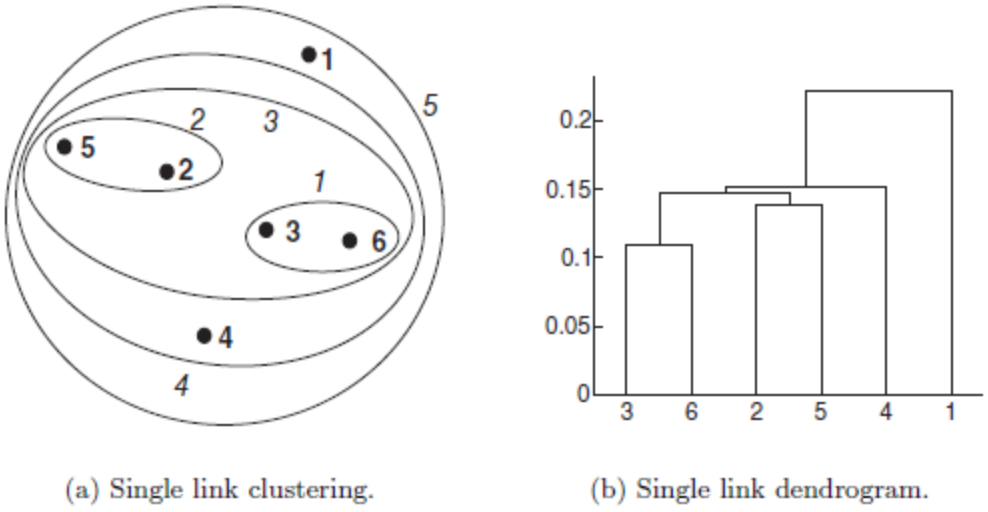
=0.15

So, looking at the last distance matrix above, we see that (p2, p5, p3, p6) and p4 have the smallest distance from all - 0.15. So, we merge those two in a single cluster, and re-compute the distance matrix.

	P1	P2P5P3P6P4
P1	0	0.22
P2P5P3P6P4	0.22	0



Finally merge clusters (P2, P5, P3, P6, P4) and P1. The clusters and dendrogram are formed as follows:



(a) Single link clustering. (b) Single link dendrogram.

Fig: single link clustering and the dendrogram of the 6 data points

Que 14: How to define proximity between two clusters using MAX technique?

(Or)

Explain complete Link hierarchical clustering with example?

(Or)

Explain Clique technique for clustering.

The proximity of two clusters is defined as the maximum of the distance between any two points in the two different clusters. Complete link is less susceptible to noise and outliers, but it can break large clusters and it favors globular shapes.

The procedure is same as Min but the difference is max distance is considered while combing to closest clusters.

For Example,

After the clusters (P2, P5) and (P3, P6) are formed, the distances are calculated as follows:

dist((p3, p6), (p2, p5)) =MAX (*dist*(p3, p2) , *dist*(p6, p2), *dist*(p3, p5),

dist(p6, p5))

=MAX (0.15, 0.25, 0.28, 0.39)

=0.39

dist((p3, p6), (p1)) =MAX (*dist*(p3, p1) , *dist*(p6, p1)

=MAX (0.22, 0.23)

=0.23

dist((p3, p6), (p4)) =MAX (*dist*(p3, p4) , *dist*(p6, p4)

=MAX (0.15, 0.22)

=0.22

dist((p2, p5), (p1)) =MAX (*dist*(p2, p1) , *dist*(p5, p1)

=MAX (0.24, 0.34)

=0.34

dist((p2, p5), (p4)) =MAX (*dist*(p2, p4) , *dist*(p5, p4)

=MAX (0.20, 0.29)

=0.29

Here the proximity is defined as maximum of distance but minimum of similarity. The lowest among all the distances is 0.22. So, merge clusters (P3, P6) and P4.

The distance between (P3, P6, P4) and remaining points are calculated as follows:

dist((p3, p6, p4), (p2, p5)) =MAX (*dist*(p3, p2) , *dist*(p6, p2), *dist*(p4, p2), *dist*(p3, p5) , *dist*(p6, p5), *dist*(p4, p5))

=MAX (0.22, 0.23, 0.37, 0.28, 0.39, 0.29)

=0.39

dist((p3, p6, p4), (p1)) =MAX (*dist*(p3, p1) , *dist*(p6, p1), *dist*(p4, p1))

=MAX (0.22, 0.23, 0.37)

=0.37

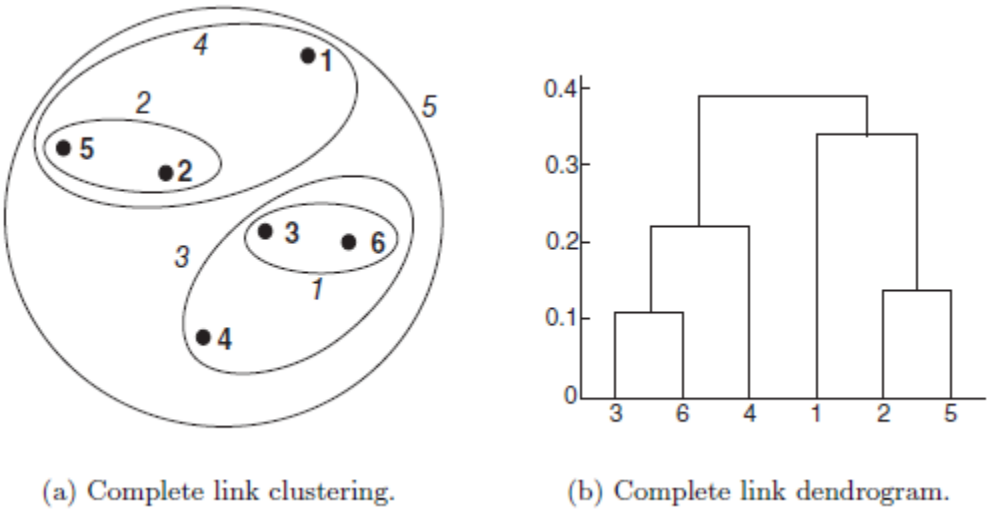
dist((p2, p5), (p1)) =MAX (*dist*(p2, p1) , *dist*(p5, p1)

=MAX (0.24, 0.34)

=0.34

The lowest among all the distances is 0.34. So, merge clusters (P2, P5) and P1.

Now merge clusters (P2, P5, P1) and (P3, P6, P4). The final clusters and dendrogram are formed as follows:



(a) Complete link clustering. (b) Complete link dendrogram.

Fig: Complete link clustering and the dendrogram of the 6 data points

Que 15: How to define proximity between two clusters using Group Average approach.

Group average technique

The proximity of two clusters is defined as the average pairwise proximity among all pairs of points in the different clusters. This is an intermediate approach between the single and complete link approaches. The cluster proximity (c_i, c_j) of clusters c_i,c_j, which are of size m_i and m_j, respectively, is expressed by the following equation:

$$proximity(C_i, C_j) = \frac{\sum_{\substack{\mathbf{x} \in C_i \\ \mathbf{y} \in C_j}} proximity(\mathbf{x}, \mathbf{y})}{m_i * m_j}.$$

Initially the clustering is same as for single link and complete link. But the distance is calculated using the above equation.

$dist(\{p3, p6\}, \{p2, p5\}) = (0.15 + 0.28 + 0.25 + 0.39)/(2 * 2) = 0.26$

$dist(\{p3, p6\}, \{p1\}) = (0.22 + 0.23)/(2 * 1) = 0.225$

$dist(\{p3, p6\}, \{p4\}) = (0.15 + 0.22)/(2 * 1) = 0.185$

$dist(\{p2, p5\}, \{p1\}) = (0.24 + 0.34)/(2 * 1) = 0.29$

$dist(\{p2, p5\}, \{p4\}) = (0.20 + 0.29)/(2 * 1) = 0.245$

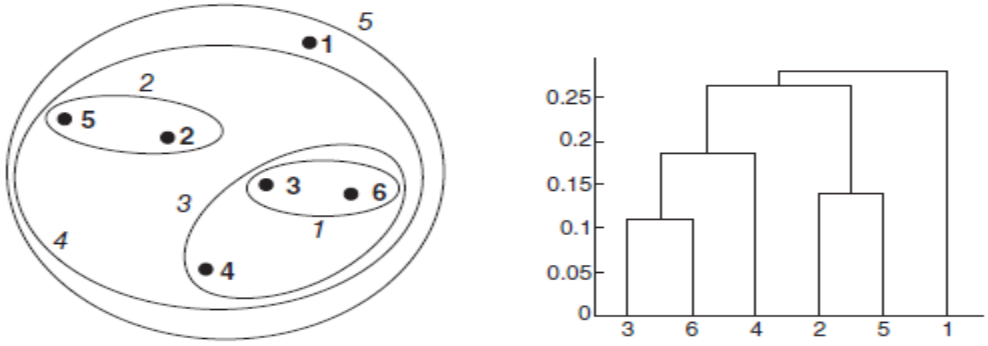
The lowest among all these distances is(P3, P6) and P4. therefore, the two clusters are merged.

The distance between (P3, P6, P4) and other data points are calculated as follows:

$dist(\{p3, p6, p4\}, \{p2, p5\}) = (0.22 + 0.28 + 0.25 + 0.39 + 0.20 + 0.29)/(3 * 2) = 0.271$

$dist(\{p3, p6, p4\}, \{p1\}) = (0.22 + 0.23 + 0.37)/(3 * 1) = 0.273$

The lowest among these two distances is 0.271, the two clusters (p3, p6, p4), (p2, p5) are merged and finally merged with P1. The final cluster and dendrogram are represented as follows:



(a) Group average clustering. (b) Group average dendrogram.

Fig: Group average clustering and the dendrogram of the 6 data points

Que 16: How to define proximity between two clusters using ward's approach.

The proximity between two clusters is defined as the increase in SSE that results after when two clusters are merged. Thus this method is similar to K-Means. This method makes ward's method distinct from other clustering techniques.

For example, Cluster A's initial SSE (Sum of square error) be 0. Let there are 3 clusters 'B', 'C', 'D'

SSE with nesting 'A' with 'B'=25.6

SSE with nesting 'A' with 'C'=10.6

SSE with nesting 'A' with 'D'=5.2

Since SSE with A and D is less, they are merged.

This method is similar to K-means because of SSE usage.

Disadvantage of Ward's Technique:

- Possibility of inversions: Two clusters that were merged may be more similar to clusters merged in previous step.

Que 17: what are the Key issues in hierarchical clustering?

1. **Lack of global objective function**
   Unlike K-Means which have some object function to measure the distance between elements and centroids. In hierarchical clustering, nearest neighbors (local merging) are merged.

2. **Ability to handle different cluster sizes**
   There are two approaches for handling clusters of different sizes. They are:
   a. **Weighted approach**, which treats all clusters equally.
   b. **Unweighted approach**, which takes the number of points in a cluster into account.

3. **Merging decisions are final**
   In this type of clustering, once two clusters are nested, Reverting back is near impossible. To address this issue, it's better to start with making clusters using K-means and then go with hierarchical clustering.

Que 18: Write the strengths, weakness, time and space complexity of hierarchical clustering?

Strengths

1. Produces better quality cluster.

2. Helps in specialized applications like creation of taxonomy (Dealing with hierarchical data).

Weaknesses

1. It is expensive.

2. Since the merges are final, it might cause some trouble to noisy and high dimensional data.

Time Complexity

O ($m^2 \log m$)

Where, 'm' is number of elements.

Space complexity

O (m^2)

DENSITY BASED CLUSTERING

Que 19: Explain center based approach for density based clustering?

In the center based approach, density is estimated for a particular point in the dataset as the number of points within the EPS (radius) of that point.

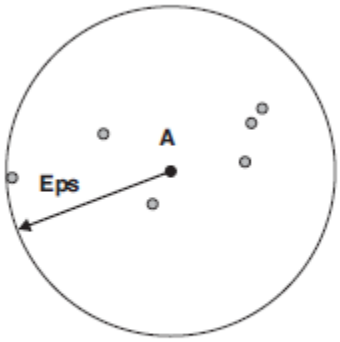
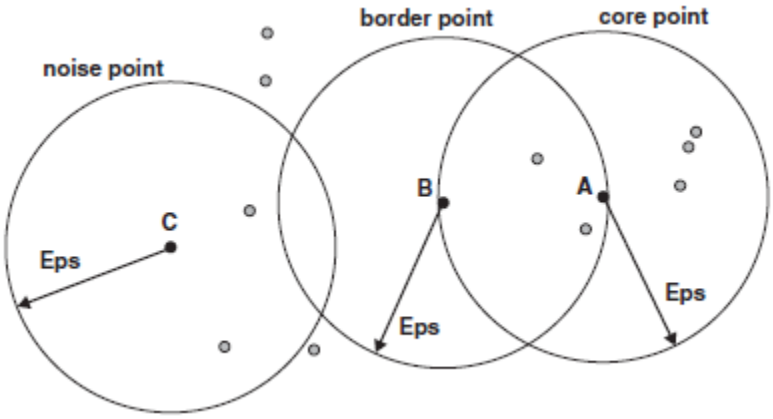


Figure 8.20. Center-based density.

A cluster is called denser, if it has minimum points (defined by Min_Pts) within the specified EPS(radius).

In density based clustering, the points are classified as core points, border points, and noise points.

1. **Core point:** These points are in the interior of the density based cluster. In the figure below, A is core point.
2. **Border point:** A border point is not a core point, but falls within the neighborhood of core point. In the figure below, B is border point.
3. **Noise point:** A noise point is any point that is neither a core point nor a border point which lies outside a density based cluster. In the figure below, C is noise point.
4. **Core object:** If the e-neighborhood of an object contains at least a minimum number, *MinPts*, of objects, then the object is called a core object.



Que 20 : Explain DBSCAN algorithm in detail.

DBSCAN (Density-Based Spatial Clustering of Applications with Noise) is a density based clustering algorithm. The algorithm grows regions with sufficiently high density into clusters and discovers clusters of arbitrary shape in spatial databases with noise.

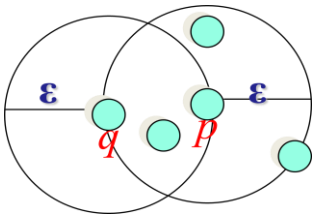
Algorithm:

Algorithm 8.4 DBSCAN algorithm.

1: Label all points as core, border, or noise points.
2: Eliminate noise points.
3: Put an edge between all core points that are within *Eps* of each other.
4: Make each group of connected core points into a separate cluster.
5: Assign each border point to one of the clusters of its associated core points.

The points in the cluster space can be classified as follows:

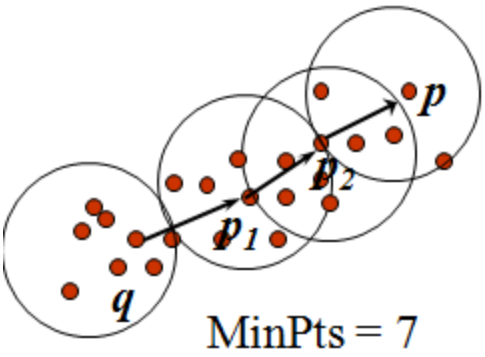
i. **Directly density reachable:** Given a set of objects, *D*, we say that an object ***p*** is directly density-reachable from object ***q*** if ***p*** is within the e-neighborhood of ***q***, and ***q*** is a core object. (An object q is directly density-reachable from object p if p is a core object and q is in p's ε-neighborhood.)



MinPts = 4

- q is directly density-reachable from p
- p is not directly density- reachable from q
- Density-reachability is asymmetric.

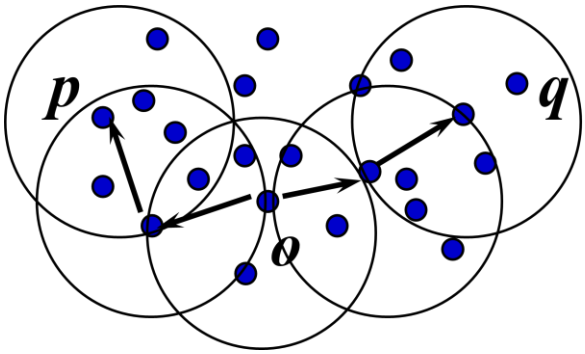
ii. **Density reachable:** An object ***p*** is density-reachable from object ***q*** with respect to e and *MinPts* in a set of objects, *D*, if there is a chain of objects ***p*1**, : : : , ***pn***, where ***p*1** = ***q*** and ***pn*** = ***p*** such that ***pi*+1** is directly density-reachable from ***pi*** with respect to e and *MinPts*, for 1 _ *i* _ *n*, ***pi*** □□***D***.



(A point p is directly density-reachable from p2; p2 is directly density-reachable from p1; p1 is directly density-reachable from q; p←p2←p1←q form a chain.)

- p is (indirectly) density-reachable from q
- q is not density- reachable from p

iii. **Density connected:** An object ***p*** is density-connected to object ***q*** with respect to e and *MinPts* in a set of objects, *D*, if there is an object **o** □*D* such that both ***p*** and ***q*** are density-reachable from ***o*** with respect to □ and *MinPts*.



- A pair of point's p and q is density-connected if they are commonly density-reachable from a point o. Density-connectivity is symmetric.

A density-based cluster is a set of density-connected objects that is maximal with respect to density-reachability. Every object not contained in any cluster is considered to be *noise*.

Que 21 : Write the strengths, weakness, time and space complexity of DBSCAN?

Ans:

Strengths:

1) Resistant to noise
2) Can handle clusters of arbitrary size.

Weakness:

1) Trouble when clusters have varying density.
2) Hard to handle high dimensional data.

Time complexity:

O(m* time to find points in EPS-Neighborhood)

Where, m is number of points.

Space complexity:

$O(m^2)$

Que 22: What is the difference between classification and clustering?

Classification	Clustering
1) Supervised learning	1) Unsupervised learning
2) Class labels are needed to learn the data	2) No Need of Class labels
3) Mostly do predictive tasks	3) Mostly do descriptive tasks
4) Algorithms: Decision tree induction, Naïve bayes approach	4) K-Means, DBSCAN
5) Mostly used to predict class label of new sample	5) Mostly used group data based on a property
6) Example application: To predict whether a customer will buy a computer or not	6) Example application: To cluster customers are premium customers, average spenders, low spenders.