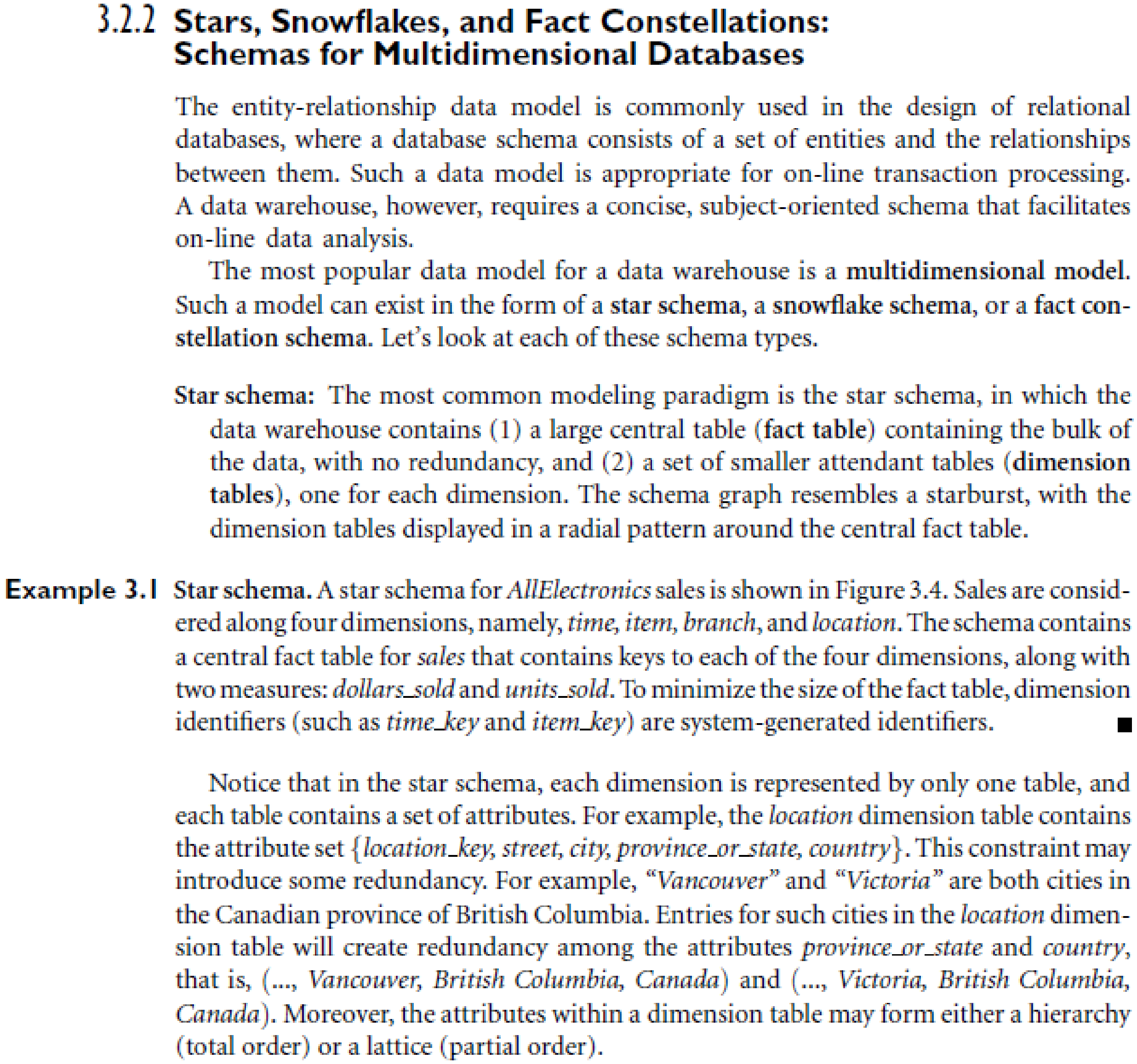
1a) Explain the three types of schemas in multi dimensional data model with example. 8M
Ans.



3.2.2 Stars, Snowflakes, and Fact Constellations: Schemas for Multidimensional Databases

The entity-relationship data model is commonly used in the design of relational databases, where a database schema consists of a set of entities and the relationships between them. Such a data model is appropriate for on-line transaction processing. A data warehouse, however, requires a concise, subject-oriented schema that facilitates on-line data analysis.

The most popular data model for a data warehouse is a **multidimensional model**. Such a model can exist in the form of a **star schema**, a **snowflake schema**, or a **fact constellation schema**. Let's look at each of these schema types.

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, with no redundancy, and (2) a set of smaller attendant tables (**dimension tables**), one for each dimension. The schema graph resembles a starburst, with the dimension tables displayed in a radial pattern around the central fact table.

Example 3.1 **Star schema.** A star schema for *AllElectronics* sales is shown in Figure 3.4. Sales are considered along four dimensions, namely, *time, item, branch*, and *location*. The schema contains a central fact table for *sales* that contains keys to each of the four dimensions, along with two measures: *dollars_sold* and *units_sold*. To minimize the size of the fact table, dimension identifiers (such as *time_key* and *item_key*) are system-generated identifiers. ■

Notice that in the star schema, each dimension is represented by only one table, and each table contains a set of attributes. For example, the *location* dimension table contains the attribute set {*location_key, street, city, province_or_state, country*}. This constraint may introduce some redundancy. For example, "*Vancouver*" and "*Victoria*" are both cities in the Canadian province of British Columbia. Entries for such cities in the *location* dimension table will create redundancy among the attributes *province_or_state* and *country*, that is, (..., *Vancouver, British Columbia, Canada*) and (..., *Victoria, British Columbia, Canada*). Moreover, the attributes within a dimension table may form either a hierarchy (total order) or a lattice (partial order).

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. The resulting schema graph forms a shape similar to a snowflake.

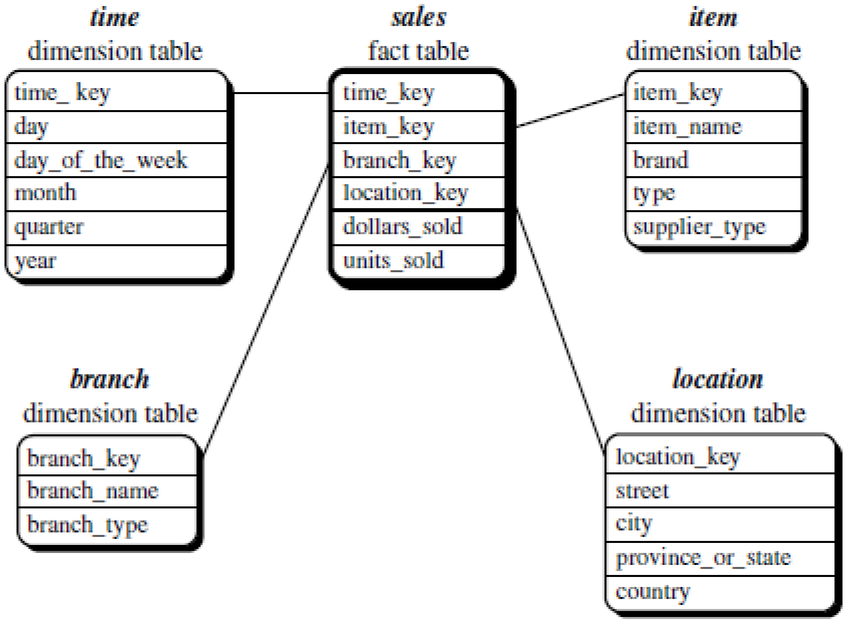


Figure 3.4 Star schema of a data warehouse for sales.

The major difference between the snowflake and star schema models is that the dimension tables of the snowflake model may be kept in normalized form to reduce redundancies. Such a table is easy to maintain and saves storage space. However, this saving of space is negligible in comparison to the typical magnitude of the fact table. Furthermore, the snowflake structure can reduce the effectiveness of browsing, since more joins will be needed to execute a query. Consequently, the system performance may be adversely impacted. Hence, although the snowflake schema reduces redundancy, it is not as popular as the star schema in data warehouse design.

Example 3.2 **Snowflake schema.** A snowflake schema for *AllElectronics* sales is given in Figure 3.5. Here, the *sales* fact table is identical to that of the star schema in Figure 3.4. The main difference between the two schemas is in the definition of dimension tables. The single dimension table for *item* in the star schema is normalized in the snowflake schema, resulting in new *item* and *supplier* tables. For example, the *item* dimension table now contains the attributes *item_key, item_name, brand, type*, and *supplier_key*, where *supplier_key* is linked to the *supplier* dimension table, containing *supplier_key* and *supplier_type* information. Similarly, the single dimension table for *location* in the star schema can be normalized into two new tables: *location* and *city*. The *city_key* in the new *location* table links to the *city* dimension. Notice that further normalization can be performed on *province_or_state* and *country* in the snowflake schema shown in Figure 3.5, when desirable. ■

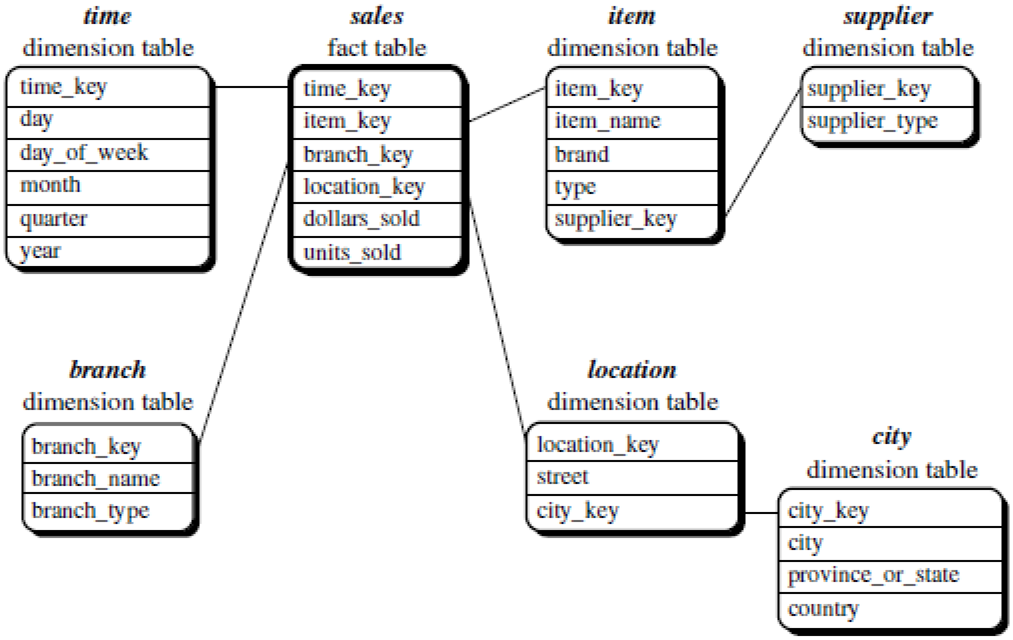


Figure 3.5 Snowflake schema of a data warehouse for sales.

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.**

Example 3.3 **Fact constellation.** A fact constellation schema is shown in Figure 3.6. This schema specifies two fact tables, *sales* and *shipping*. The *sales* table definition is identical to that of the star schema (Figure 3.4). The *shipping* table has five dimensions, or keys: *item_key, time_key, shipper_key, from_location*, and *to_location*, and two measures: *dollars_cost* and *units_shipped*. A fact constellation schema allows dimension tables to be shared between fact tables. For example, the dimensions tables for *time, item*, and *location* are shared between both the *sales* and *shipping* fact tables. ■

In data warehousing, there is a distinction between a data warehouse and a data mart. A data warehouse collects information about subjects that span the *entire organization*, such as *customers, items, sales, assets*, and *personnel*, and thus its scope is *enterprise-wide*. For data warehouses, the fact constellation schema is commonly used, since it can model multiple, interrelated subjects. A **data mart**, on the other hand, is a department subset of the data warehouse that focuses on selected subjects, and thus its scope is *department-wide*. For data marts, the *star* or *snowflake* schema are commonly used, since both are geared toward modeling single subjects, although the star schema is more popular and efficient.

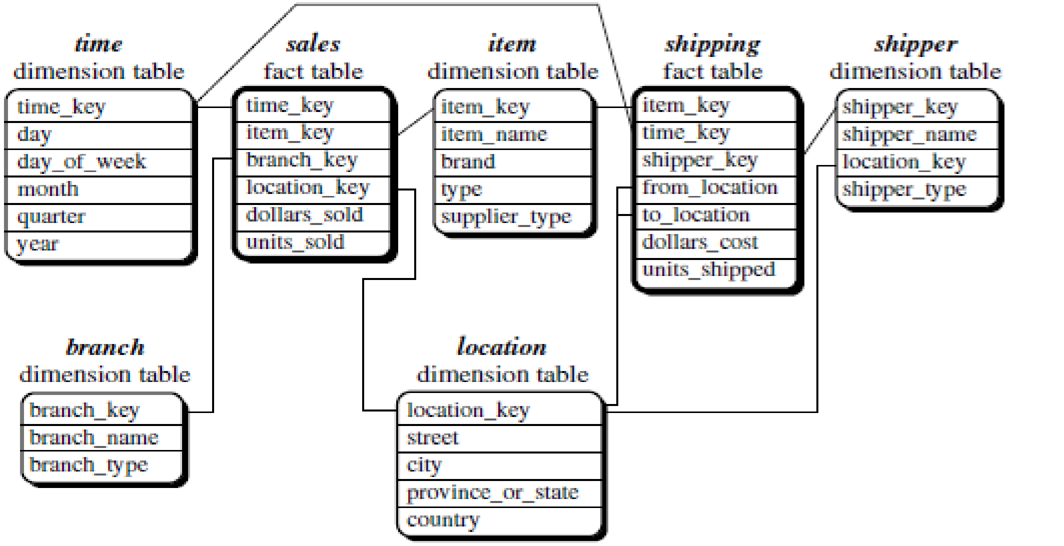
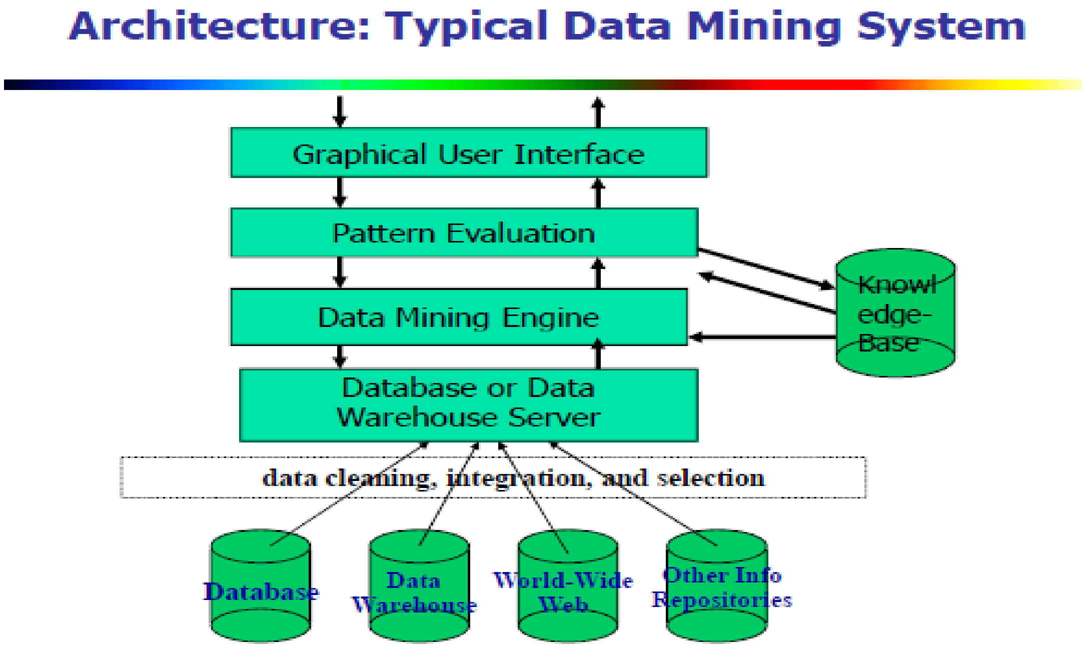


Figure 3.6 Fact constellation schema of a data warehouse for sales and shipping.

1b) Describe 3-tier data warehouse architecture with a neat diagram 8M



Data warehouses normally adopt three-tier architecture: 1. The bottom tiers is a warehouse database server that is almost always a relational database system. Data from operational databases and from external sources are extracted using application program interfaces known as **gateways.**

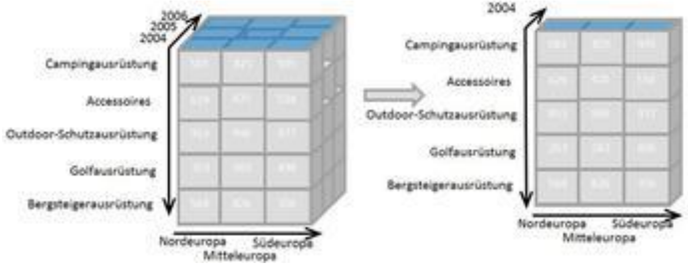
A gateway is supported by the underlying DBMS and allows client programs to execute code. 2. The middle tier is an OLAP server that is typically implemented using a relational OLAP (ROLAP) model. 3. The top tier is a client, which contains query and reporting tools, analysis tools and/or data mining tools. From the architecture point of view there are three data warehouse models: the enterprise warehouse, the data mart, and the virtual warehouse.

2a) Explain data cube operations with example for each operation. 8M

OLAP data is typically stored in a star schema or snowflake schema in a relational data warehouse or in a special-purpose data management system. Measures are derived from the records in the fact table and dimensions are derived from the dimension tables.

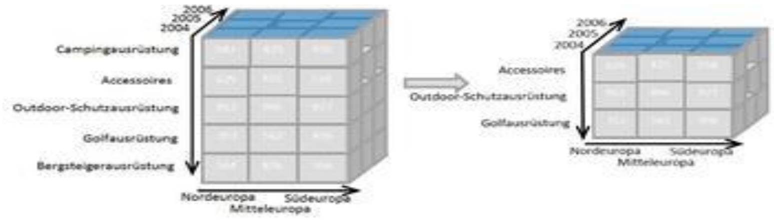
- Hierarchy
- The elements of a dimension can be organized as a hierarchy,[4] a set of parent-child relationships, typically where a parent member summarizes its children. Parent elements can further be aggregated as the children of another parent.[5]
- For example May 2005's parent is Second Quarter 2005 which is in turn the child of Year 2005. Similarly cities are the children of regions; products roll into product groups and individual expense items into types of expenditure.
- Operations
- Conceiving data as a cube with hierarchical dimensions leads to conceptually straightforward operations to facilitate analysis. Aligning the data content with a familiar visualization enhances analyst learning and productivity.[5] The user-initiated process of navigating by calling for page displays interactively, through the specification of slices via rotations and drill down/up is sometimes called "slice and dice".

Common operations include slice and dice, drill down, roll up, and pivot.

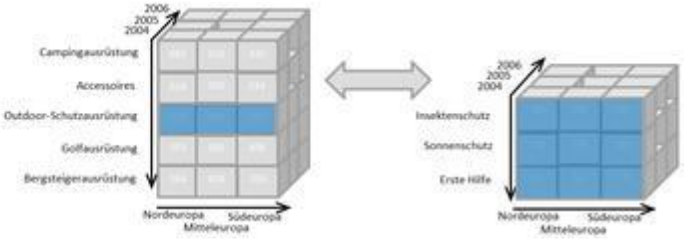


OLAP slicing

- Slice is the act of picking a rectangular subset of a cube by choosing a single value for one of its dimensions, creating a new cube with one fewer dimension.[5] The picture shows a slicing operation: The sales figures of all sales regions and all product categories of the company in the year 2004 are "sliced" out of the data cube.



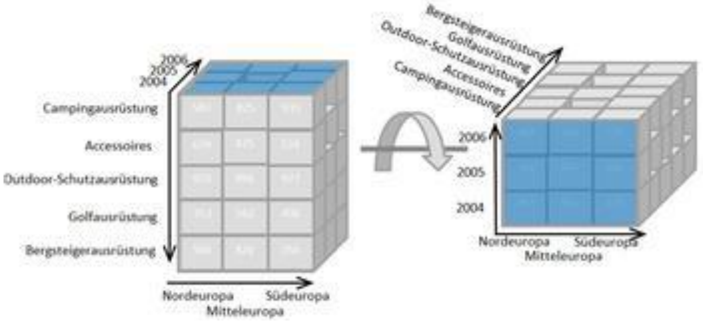
- OLAP dicing
- Dice: The dice operation produces a sub cube by allowing the analyst to pick specific values of multiple dimensions.[6] The picture shows a dicing operation: The new cube shows the sales figures of a limited number of product categories, the time and region dimensions cover the same range as before.



- OLAP Drill-up and drill-down

Drill Down/Up allows the user to navigate among levels of data ranging from the most summarized (up) to the most detailed (down).[5] The picture shows a drill-down operation: The analyst moves from the summary category "Outdoor-Schutzausrüstung" to see the sales figures for the individual products.

Roll-up: A roll-up involves summarizing the data along a dimension. The summarization rule might be computing totals along a hierarchy or applying a set of formulas such as "profit = sales - expenses".[5]



- OLAP pivoting
- Pivot allows an analyst to rotate the cube in space to see its various faces. For example, cities could be arranged vertically and products horizontally while viewing data for a particular quarter. Pivoting could replace products with time periods to see data across time for a single product.
- The picture shows a pivoting operation: The whole cube is rotational.

2b) What is datawarehouse? Compare OLAP and OLTP systems. 8M

Ans. According to W. H. Inmon: *A data warehouse is a subject-oriented, integrated, time-variant, and non-volatile collection of data in support of management's decision making process.*

Subject-oriented

- A DW is organized around major subjects, such as student, degree, country.
- Focusing on the modeling and analysis of data for decision makers, not on daily operations.
- A DW provides a simple and concise view around particular subject issues by excluding data that are not useful in the decision support process.

Integrated

- A DW may be constructed by integrating information from multiple data sources e.g. multiple OLTP databases.
- Data cleaning and data integration techniques are applied to ensure consistency in naming conventions, encoding structures, attribute measures, etc. among different data sources.

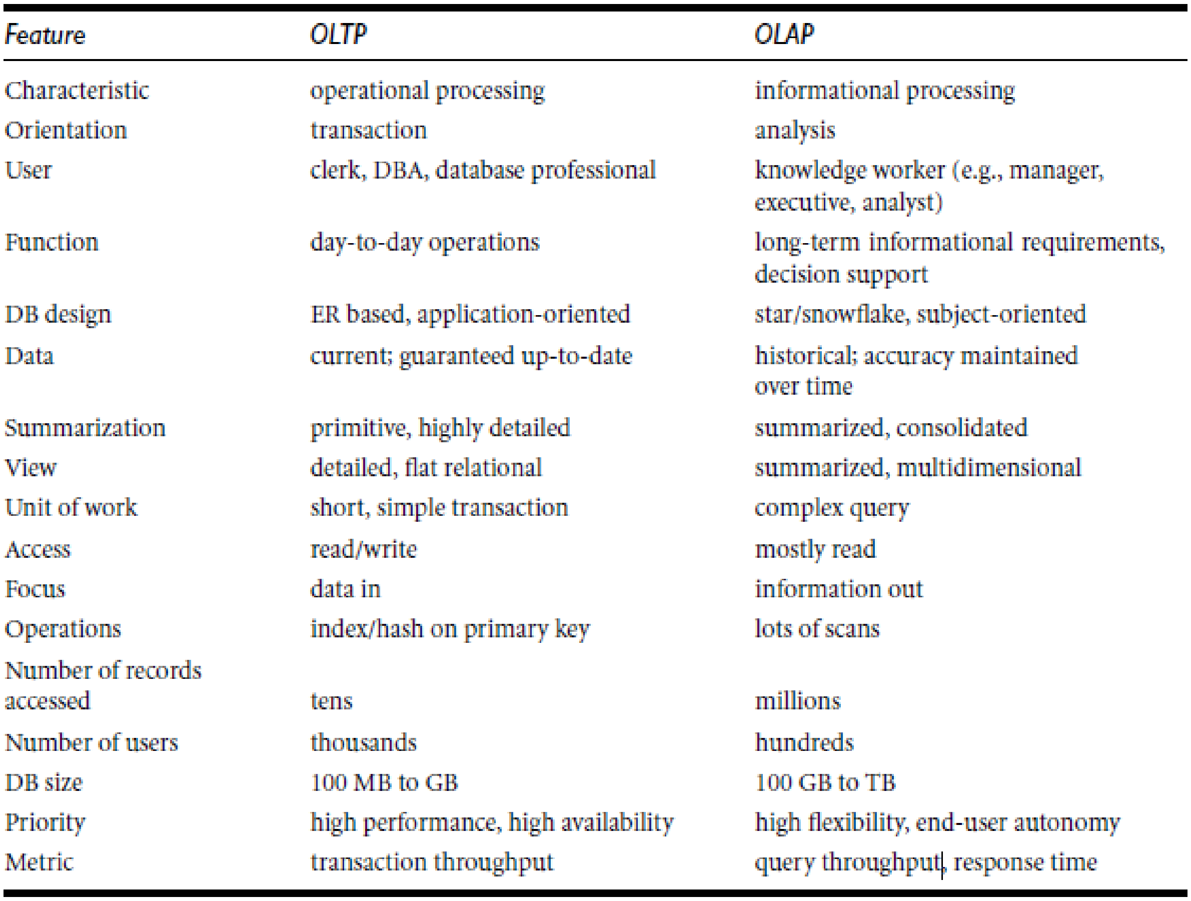
Time Variant

- A DW usually has long time horizon, significantly longer than that of operational systems.
- Operational database: current value data.
- DW data: provide information from a historical perspective (e.g. past 5-10 years)
- Every key structure in the DW contains an element of time, explicitly or implicitly
- Operational data may or may not contain time element.

Non-volatile

- A physically separate store of data transformed from the operational environment.
- No update of data
- Does not require transaction processing, recovery, and concurrency control mechanisms
- Requires only two operations in data accessing: initial loading of data and access of data.

Table 3.1 Comparison between OLTP and OLAP systems.



Feature	*OLTP*	*OLAP*
Characteristic	operational processing	informational processing
Orientation	transaction	analysis
User	clerk, DBA, database professional	knowledge worker (e.g., manager, executive, analyst)
Function	day-to-day operations	long-term informational requirements, decision support
DB design	ER based, application-oriented	star/snowflake, subject-oriented
Data	current; guaranteed up-to-date	historical; accuracy maintained over time
Summarization	primitive, highly detailed	summarized, consolidated
View	detailed, flat relational	summarized, multidimensional
Unit of work	short, simple transaction	complex query
Access	read/write	mostly read
Focus	data in	information out
Operations	index/hash on primary key	lots of scans
Number of records accessed	tens	millions
Number of users	thousands	hundreds
DB size	100 MB to GB	100 GB to TB
Priority	high performance, high availability	high flexibility, end-user autonomy
Metric	transaction throughput	query throughput, response time

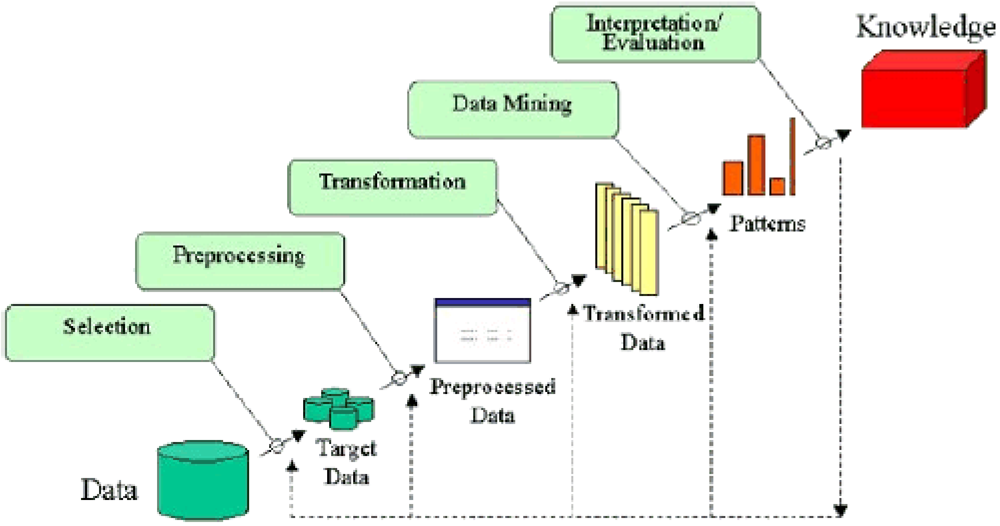
3a) What is data mining? Explain KDD process in data mining with neat diagram. 8M

Data mining refers to extracting or mining" knowledge from large amounts of data. There are many other terms related to data mining, such as knowledge mining, knowledge extraction, data/pattern analysis, data archaeology, and data dredging. Many people treat data mining as a synonym for another popularly used term, Knowledge Discovery in Databases", or KDD

Data mining is the process of discovering interesting knowledge from large amounts of data stored either in databases, data warehouses, or other information repositories.

The term *Knowledge Discovery in Databases*, refers to the broad process of finding knowledge in data, and emphasizes the "high-level" application of particular data mining methods. It is of interest to researchers in machine learning, pattern recognition, databases, statistics, artificial intelligence, knowledge acquisition for expert systems, and data visualization.
The goal of the KDD process is to extract knowledge from data in the context of large databases.
It does this by using data mining methods (algorithms) to extract (identify) what is deemed knowledge, according to the specifications of measures and thresholds, using a database along with any required preprocessing, sub sampling and transformations of that database.

Steps of KDD process:



- Data cleaning: to remove noise or irrelevant data
- Data integration: where multiple data sources may be combined
- Data selection: where data relevant to the analysis task are retrieved from the database
- Data transformation: where data are transformed or consolidated into forms appropriate for mining by performing summary or aggregation operations
- Data mining : An essential process where intelligent methods are applied in order to extract data patterns.
- Pattern evaluation to identify the truly interesting patterns representing knowledge based on some interestingness measures.
- Knowledge presentation: where visualization and knowledge representation techniques are used to present the mined knowledge to the user.

3b) Briefly explain motivating challenges in the field of data mining. 4M

Ans.

Motivational challenges in the field of data mining:

1.Scalability: If data mining algorithms are to handle these massive data sets, then they must be scalable.

2. High Dimensionality: For some data analysis algorithms, the computational complexity increases rapidly as the dimensionality increases.
3. Heterogeneous and Complex Data: Dealing with data with not the same type.
4. Data Ownership and Distribution: Data is geographically distributed among resources belonging to multiple entities.
5. Non-traditional Analysis: The traditional statistical approach is based on a hypothesize-and test paradigm.

Current data analysis tasks often require the generation and evaluation of thousands of hypotheses, and consequently, the development of some data mining techniques has been motivated by the desire to automate the process of hypothesis generation and evaluation.

3c) Briefly discuss various applications of data mining. 4M

Financial Data Analysis

The financial data in banking and financial industry is generally reliable and of high quality which facilitates systematic data analysis and data mining.

Design and construction of data warehouses for multidimensional data analysis and data mining.

Detection of money laundering and other financial crimes.

Retail Industry

Data mining collects large amount of data from on sales, customer purchasing history, goods transportation, consumption and services. It is natural that the quantity of data collected will continue to expand rapidly because of the increasing ease, availability and popularity of the web.

Ex: Product recommendation and cross-referencing of items.

Telecommunication Industry

Data mining in telecommunication industry helps in identifying the telecommunication patterns, catch fraudulent activities, make better use of resource, and improve quality of service

Examples for which data mining improves telecommunication services −

Multidimensional Analysis of Telecommunication data.

Fraudulent pattern analysis.

Identification of unusual patterns.

Biological Data Analysis

In the field of biology such as genomics, proteomics, functional Genomics and biomedical research. Biological data mining is a very important part of Bioinformatics. Following are the aspects in which data mining contributes for biological data analysis −

Semantic integration of heterogeneous, distributed genomic and proteomic databases.

Alignment, indexing, similarity search and comparative analysis multiple nucleotide sequences.

Discovery of structural patterns and analysis of genetic networks and protein pathways.

Association and path analysis.

Visualization tools in genetic data analysis.

Following are the applications of data mining in the field of Scientific Applications −

Data Warehouses and data preprocessing.

Graph-based mining.

Visualization and domain specific knowledge.

Intrusion Detection

Intrusion refers to any kind of action that threatens integrity, confidentiality, or the availability of network resources. In this world of connectivity, security has become the major issue. With increased usage of internet and availability of the tools and tricks for intruding and attacking network prompted intrusion detection to become a critical component of network administration.

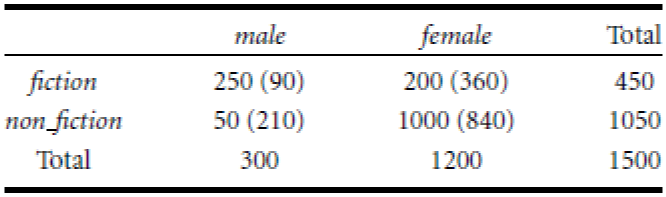
Here is the list of areas in which data mining technology may be applied for intrusion detection −

Development of data mining algorithm for intrusion detection.

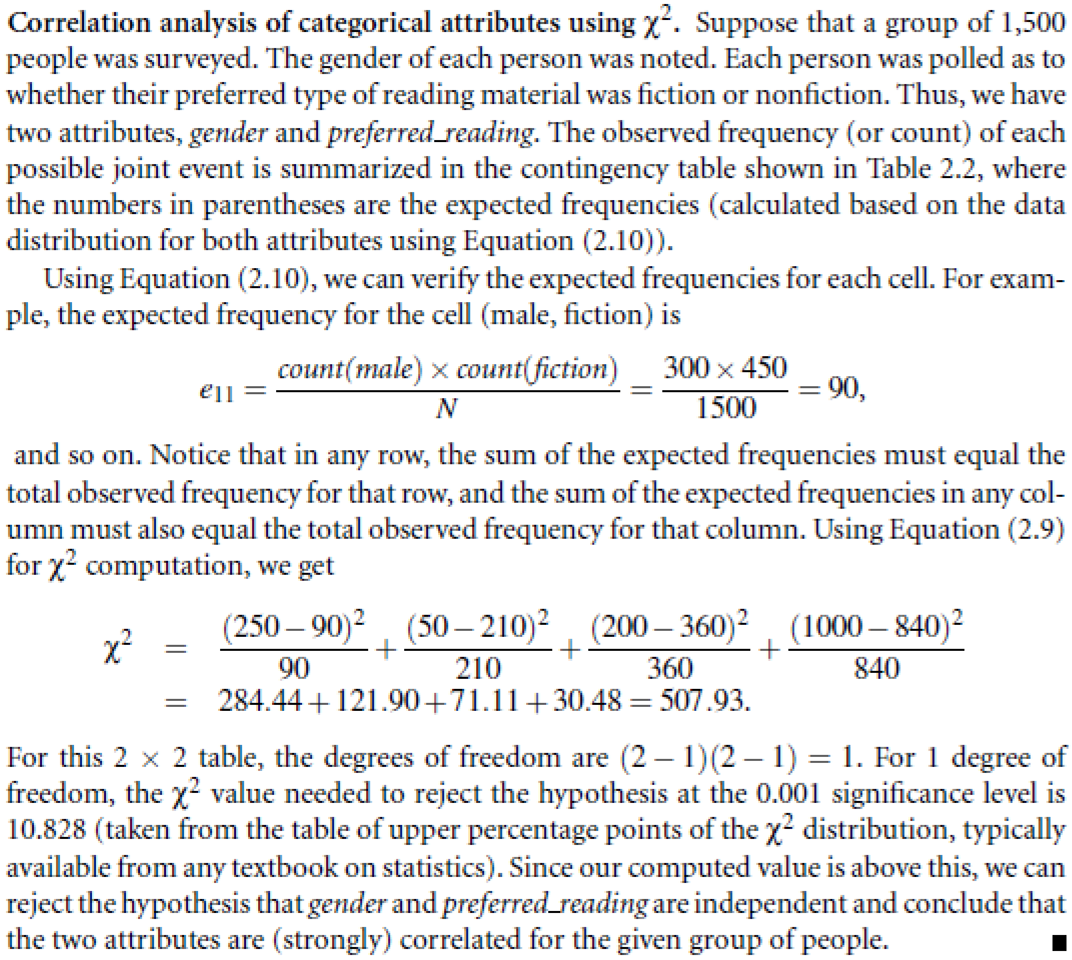
Association and correlation analysis, aggregation to help select and build discriminating attributes.

4a) **For the below given 2*2 contingency table with two attributes “gender” and “preferred reading” conduct correlation analysis between the given attributes using Chi-square test.** Note: Expected frequencies are given inside parentheses.

Gender



	male	*female*	Total
fiction	250 (90)	200 (360)	450
non_fiction	50 (210)	1000 (840)	1050
Total	300	1200	1500



Correlation analysis of categorical attributes using χ^2. Suppose that a group of 1,500 people was surveyed. The gender of each person was noted. Each person was polled as to whether their preferred type of reading material was fiction or nonfiction. Thus, we have two attributes, *gender* and *preferred_reading*. The observed frequency (or count) of each possible joint event is summarized in the contingency table shown in Table 2.2, where the numbers in parentheses are the expected frequencies (calculated based on the data distribution for both attributes using Equation (2.10)).

Using Equation (2.10), we can verify the expected frequencies for each cell. For example, the expected frequency for the cell (male, fiction) is

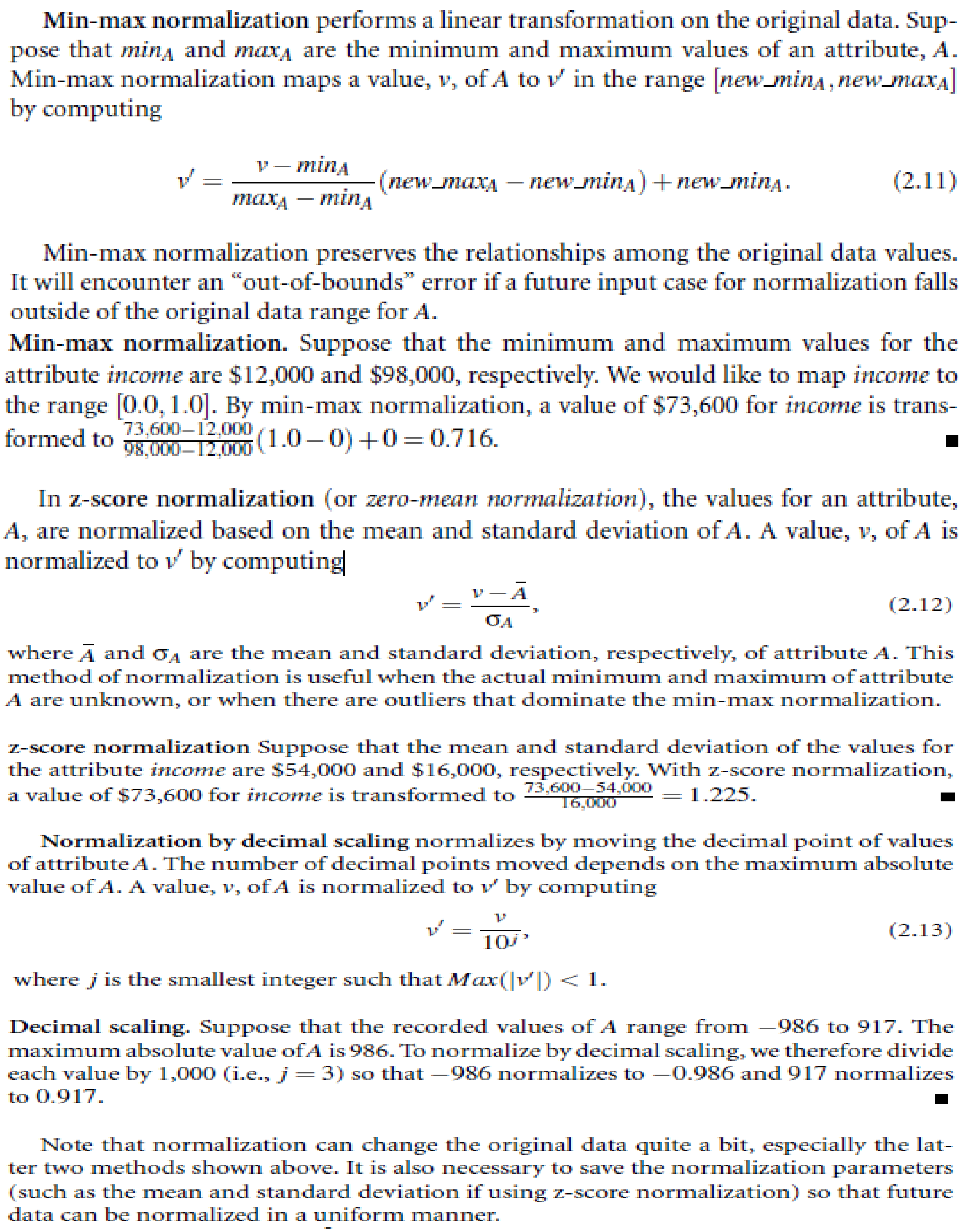
$$e_{11} = \frac{count(male) \times count(fiction)}{N} = \frac{300 \times 450}{1500} = 90,$$

and so on. Notice that in any row, the sum of the expected frequencies must equal the total observed frequency for that row, and the sum of the expected frequencies in any column must also equal the total observed frequency for that column. Using Equation (2.9) for χ^2 computation, we get

$$\begin{aligned} \chi^2 &= \frac{(250-90)^2}{90} + \frac{(50-210)^2}{210} + \frac{(200-360)^2}{360} + \frac{(1000-840)^2}{840} \\ &= 284.44 + 121.90 + 71.11 + 30.48 = 507.93. \end{aligned}$$

For this 2 × 2 table, the degrees of freedom are $(2-1)(2-1) = 1$. For 1 degree of freedom, the χ^2 value needed to reject the hypothesis at the 0.001 significance level is 10.828 (taken from the table of upper percentage points of the χ^2 distribution, typically available from any textbook on statistics). Since our computed value is above this, we can reject the hypothesis that *gender* and *preferred_reading* are independent and conclude that the two attributes are (strongly) correlated for the given group of people. ■

4b) Briefly explain Min-Max, Z-Score and Normalization by decimal scaling with a suitable example. 8M



Min-max normalization performs a linear transformation on the original data. Suppose that min_A and max_A are the minimum and maximum values of an attribute, A. Min-max normalization maps a value, v, of A to v' in the range $[new\_min_A, new\_max_A]$ by computing

$$v' = \frac{v - min_A}{max_A - min_A}(new\_max_A - new\_min_A) + new\_min_A. \quad (2.11)$$

Min-max normalization preserves the relationships among the original data values. It will encounter an "out-of-bounds" error if a future input case for normalization falls outside of the original data range for A.

Min-max normalization. Suppose that the minimum and maximum values for the attribute *income* are \$12,000 and \$98,000, respectively. We would like to map *income* to the range $[0.0, 1.0]$. By min-max normalization, a value of \$73,600 for *income* is transformed to $\frac{73{,}600-12{,}000}{98{,}000-12{,}000}(1.0-0)+0 = 0.716$. ■

In **z-score normalization** (or *zero-mean normalization*), the values for an attribute, A, are normalized based on the mean and standard deviation of A. A value, v, of A is normalized to v' by computing

$$v' = \frac{v - \bar{A}}{\sigma_A}, \quad (2.12)$$

where $\bar{A}$ and σ_A are the mean and standard deviation, respectively, of attribute A. This method of normalization is useful when the actual minimum and maximum of attribute A are unknown, or when there are outliers that dominate the min-max normalization.

z-score normalization Suppose that the mean and standard deviation of the values for the attribute *income* are \$54,000 and \$16,000, respectively. With z-score normalization, a value of \$73,600 for *income* is transformed to $\frac{73{,}600-54{,}000}{16{,}000} = 1.225$. ■

Normalization by decimal scaling normalizes by moving the decimal point of values of attribute A. The number of decimal points moved depends on the maximum absolute value of A. A value, v, of A is normalized to v' by computing

$$v' = \frac{v}{10^j}, \quad (2.13)$$

where j is the smallest integer such that $Max(|v'|) < 1$.

Decimal scaling. Suppose that the recorded values of A range from -986 to 917. The maximum absolute value of A is 986. To normalize by decimal scaling, we therefore divide each value by 1,000 (i.e., $j = 3$) so that -986 normalizes to -0.986 and 917 normalizes to 0.917. ■

Note that normalization can change the original data quite a bit, especially the latter two methods shown above. It is also necessary to save the normalization parameters (such as the mean and standard deviation if using z-score normalization) so that future data can be normalized in a uniform manner.

5a) Define Apriori principle, briefly discuss. Apriori algorithm for Frequent Item set Generation 8M

Ans.

Apriori principle:

– If an item set is frequent, then all of its subsets must also be frequent

Apriori principle holds due to the following property of the support measure:

– Support of an item set never exceeds the support of its subsets

– This is known as the anti-monotone property of support

Apriori algorithm Method:

Let k=1

Generate frequent item sets of length 1

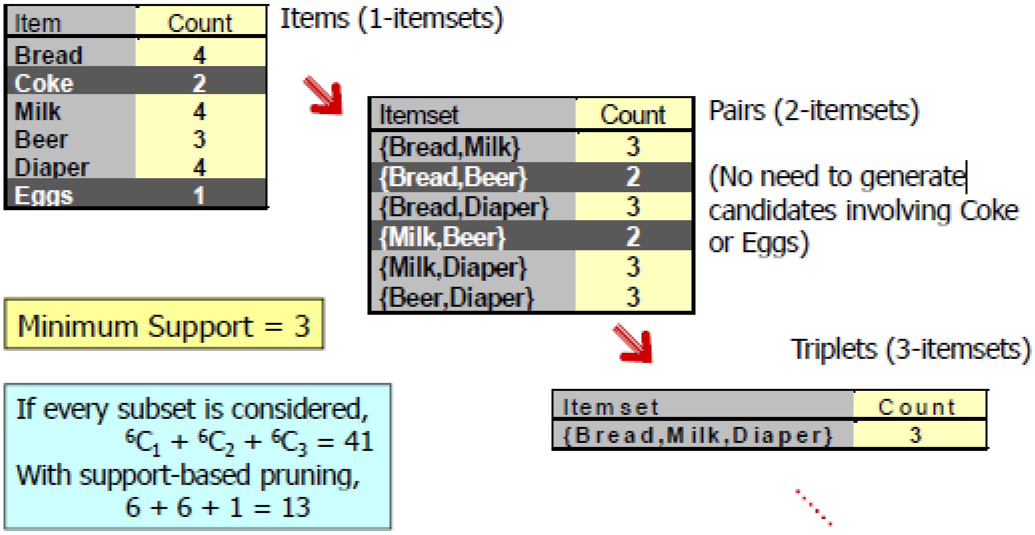
Repeat until no new frequent item sets are identified

□ Generate length (k+1) candidate item sets from length k frequent item sets

□ Prune candidate item sets containing subsets of length k that are infrequent

□ Count the support of each candidate by scanning the DB

□ Eliminate candidates that are infrequent, leaving only those that are frequent



Items (1-itemsets)

Item	Count
Bread	4
Coke	2
Milk	4
Beer	3
Diaper	4
Eggs	1

Pairs (2-itemsets)

Itemset	Count
{Bread,Milk}	3
{Bread,Beer}	2
{Bread,Diaper}	3
{Milk,Beer}	2
{Milk,Diaper}	3
{Beer,Diaper}	3

(No need to generate candidates involving Coke or Eggs)

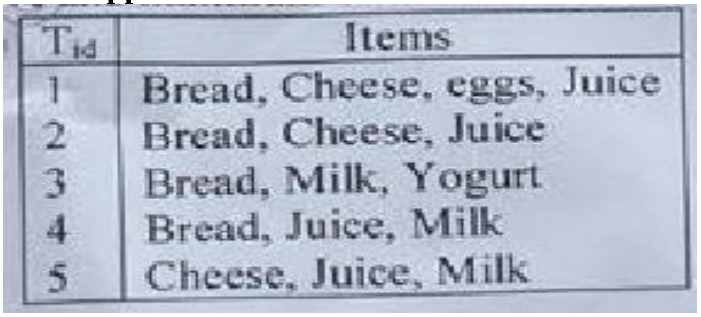
Minimum Support = 3

Triplets (3-itemsets)

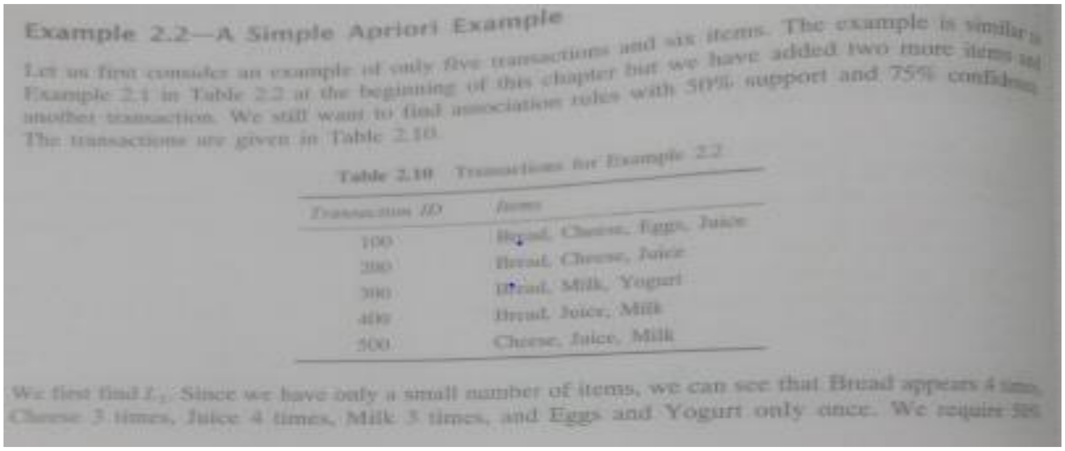
Itemset	Count
{Bread,Milk,Diaper}	3

If every subset is considered,
$^6C_1 + {}^6C_2 + {}^6C_3 = 41$
With support-based pruning,
6 + 6 + 1 = 13

5b) For a given transaction data, generate frequent item set and identify valid association rules with minimum support as 60% and minimum confidence as 75%. 8M



T_{id}	Items
1	Bread, Cheese, eggs, Juice
2	Bread, Cheese, Juice
3	Bread, Milk, Yogurt
4	Bread, Juice, Milk
5	Cheese, Juice, Milk

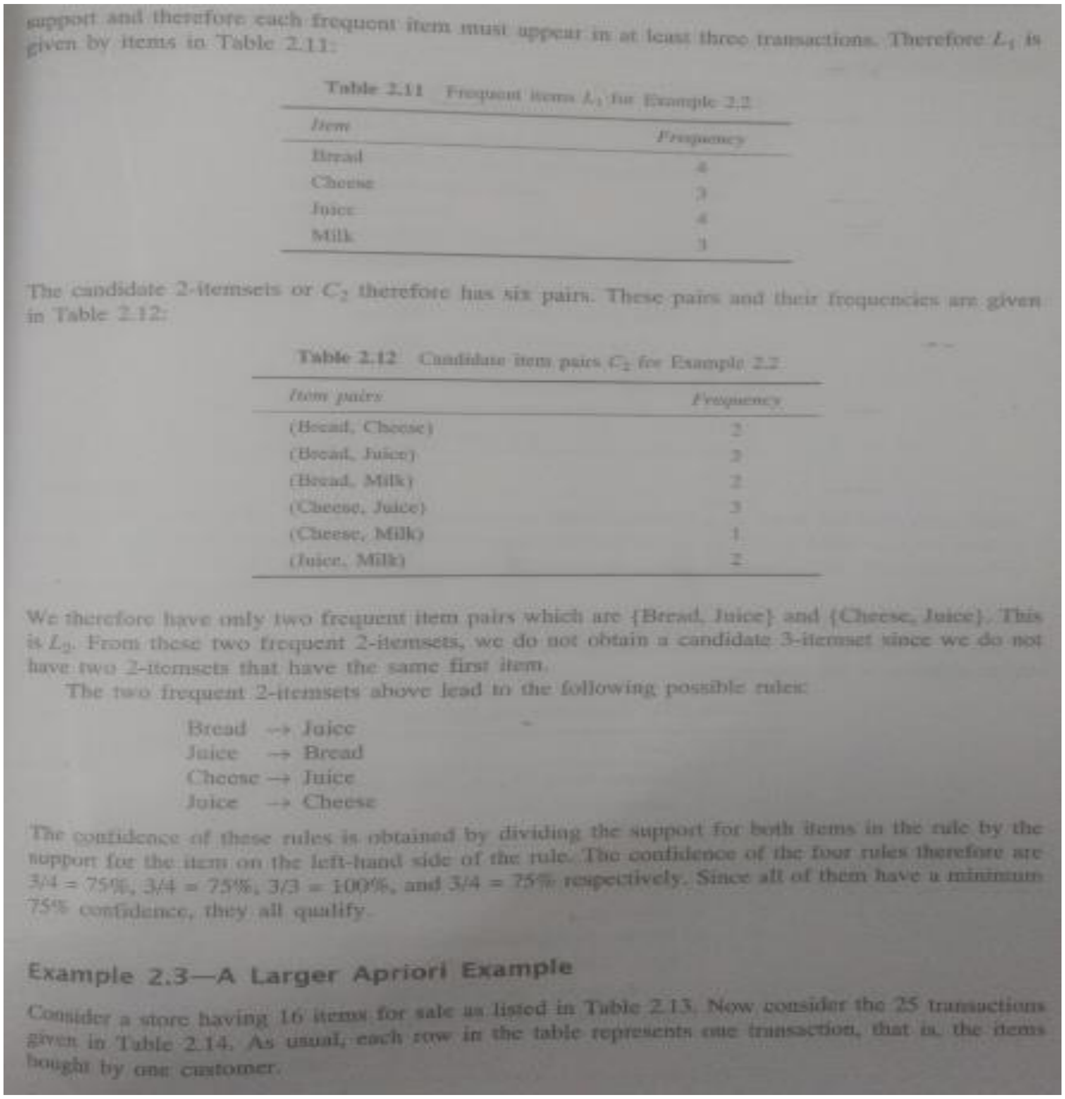


Example 2.2—A Simple Apriori Example

Let us first consider an example of only five transactions and six items. The example is similar to Example 2.1 in Table 2.2 at the beginning of this chapter but we have added two more items and another transaction. We still want to find association rules with 50% support and 75% confidence. The transactions are given in Table 2.10.

Table 2.10 Transactions for Example 2.2

Transaction ID	*Items*
100	Bread, Cheese, Eggs, Juice
200	Bread, Cheese, Juice
300	Bread, Milk, Yogurt
400	Bread, Juice, Milk
500	Cheese, Juice, Milk



We first find L_1. Since we have only a small number of items, we can see that Bread appears 4 times, Cheese 3 times, Juice 4 times, Milk 3 times, and Eggs and Yogurt only once. We require 50% support and therefore each frequent item must appear in at least three transactions. Therefore L_1 is given by items in Table 2.11:

Table 2.11 Frequent items L_1 for Example 2.2

Item	*Frequency*
Bread	4
Cheese	3
Juice	4
Milk	3

The candidate 2-itemsets or C_2 therefore has six pairs. These pairs and their frequencies are given in Table 2.12:

Table 2.12 Candidate item pairs C_2 for Example 2.2

Item pairs	*Frequency*
(Bread, Cheese)	2
(Bread, Juice)	3
(Bread, Milk)	2
(Cheese, Juice)	3
(Cheese, Milk)	1
(Juice, Milk)	2

We therefore have only two frequent item pairs which are {Bread, Juice} and {Cheese, Juice}. This is L_2. From these two frequent 2-itemsets, we do not obtain a candidate 3-itemset since we do not have two 2-itemsets that have the same first item.

The two frequent 2-itemsets above lead to the following possible rules:

Bread → Juice
Juice → Bread
Cheese → Juice
Juice → Cheese

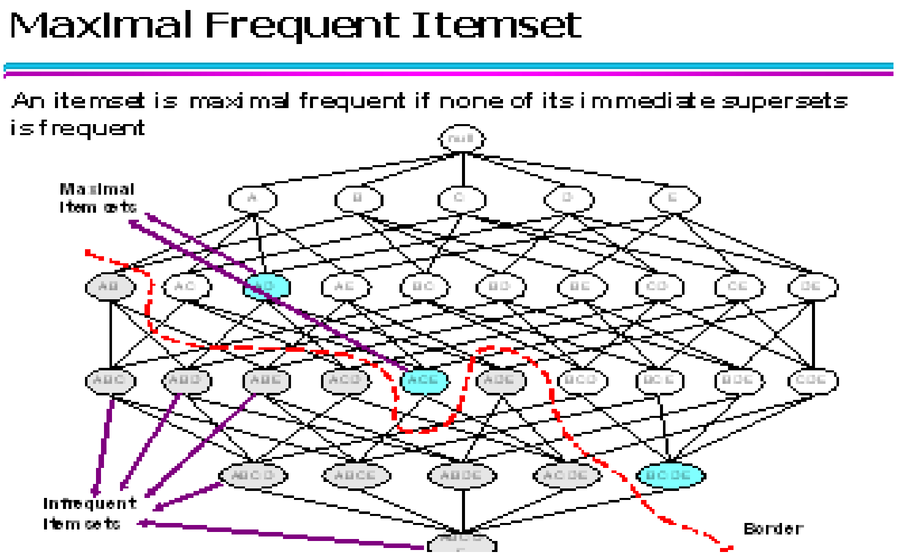
The confidence of these rules is obtained by dividing the support for both items in the rule by the support for the item on the left-hand side of the rule. The confidence of the four rules therefore are 3/4 = 75%, 3/4 = 75%, 3/3 = 100%, and 3/4 = 75% respectively. Since all of them have a minimum 75% confidence, they all qualify.

Example 2.3—A Larger Apriori Example

Consider a store having 16 items for sale as listed in Table 2.13. Now consider the 25 transactions given in Table 2.14. As usual, each row in the table represents one transaction, that is, the items bought by one customer.

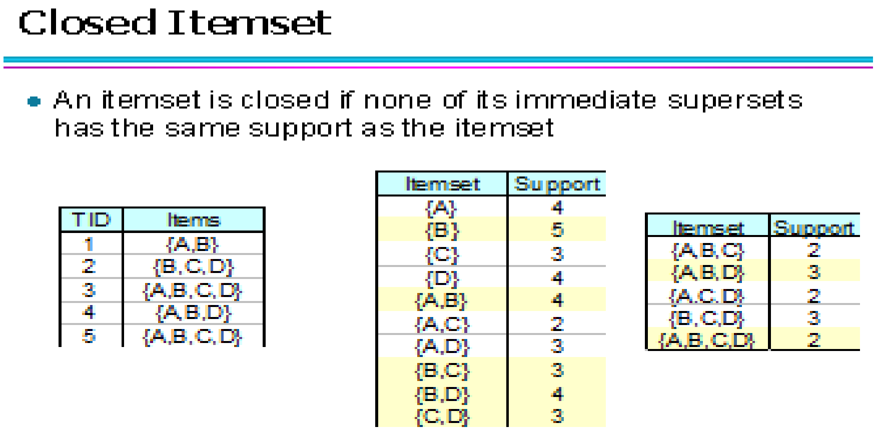
6 a. Explain Maximal, frequency Item set and closed Frequent Item set techniques for compact representation, Using an example for each.

Ans:



Maximal Frequent Itemset

An itemset is maximal frequent if none of its immediate supersets is frequent



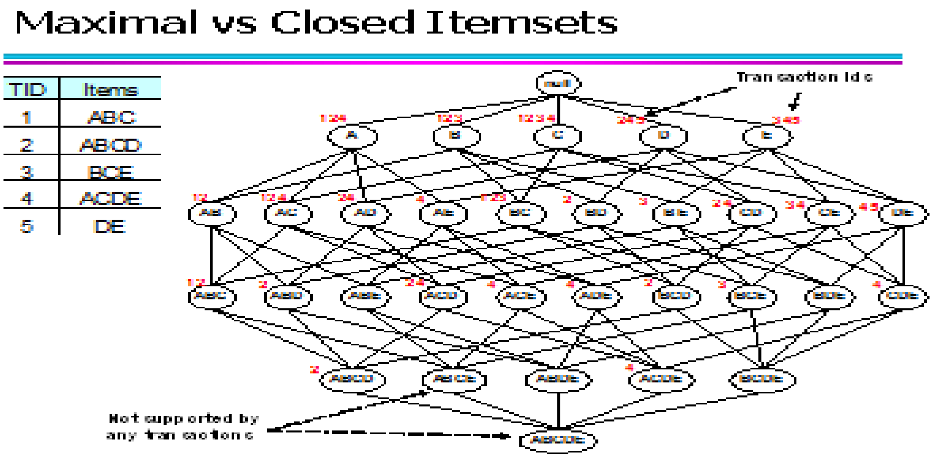
Closed Itemset

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

TID	Items
1	{A,B}
2	{B,C,D}
3	{A,B,C,D}
4	{A,B,D}
5	{A,B,C,D}

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

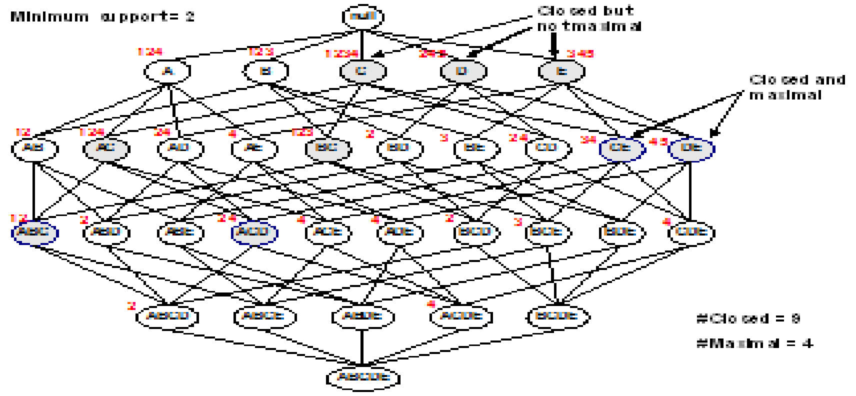
Itemset	Support
{A,B,C}	2
{A,B,D}	3
{A,C,D}	2
{B,C,D}	3
{A,B,C,D}	2



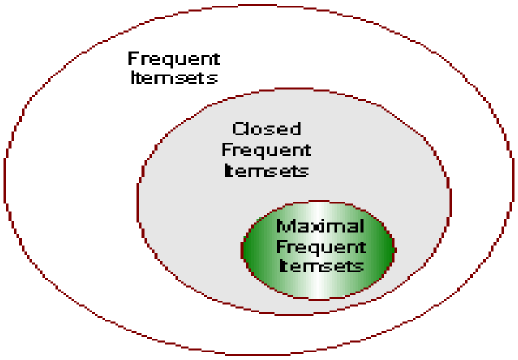
Maximal vs Closed Itemsets

TID	Items
1	ABC
2	ABCD
3	BCE
4	ACDE
5	DE

Maximal vs Closed Frequent Itemsets



Maximal vs Closed Itemsets



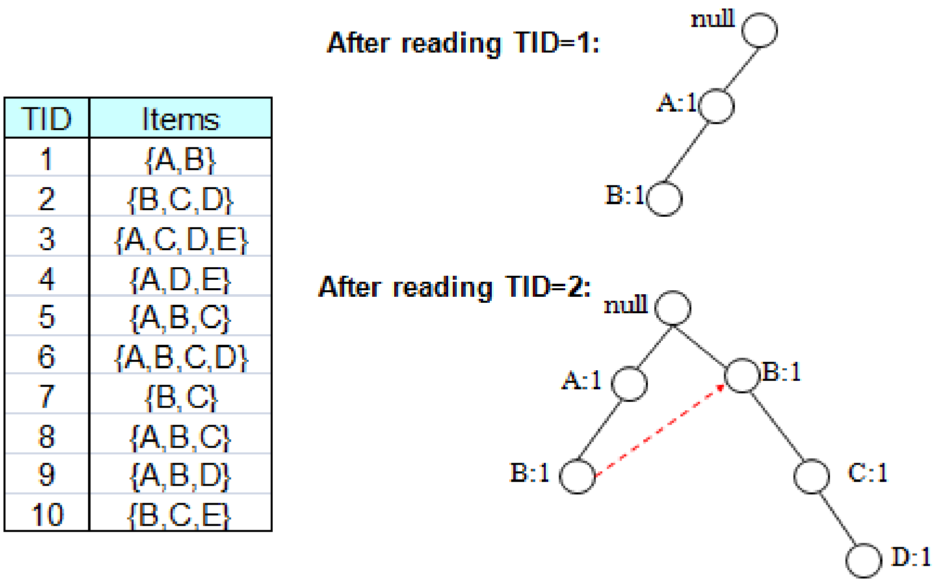
b. Explain and construct FP Tree for given Transaction data:
Data Set:

TID	Items
1	{a,b}
2	{b,c,d}
3	{a,c,d,e}
4	{a,d,e}
5	{a,b,c}
6	{a,b,c,d}
7	{a}
8	{a,b,c}
9	{a,b,c}
10	{b,c,e}

Ans: **FP-growth Algorithm**

-Use a compressed representation of the database using an FP-tree

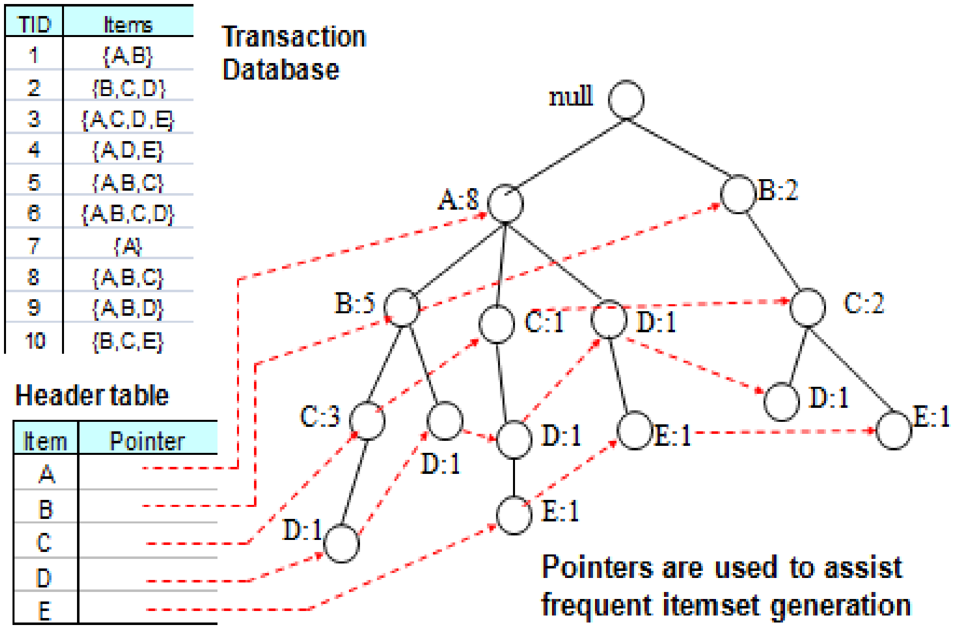
-Once an FP-tree has been constructed, it uses a recursive divide-and-conquer approach to mine the frequent item sets .



TID	Items
1	{A,B}
2	{B,C,D}
3	{A,C,D,E}
4	{A,D,E}
5	{A,B,C}
6	{A,B,C,D}
7	{B,C}
8	{A,B,C}
9	{A,B,D}
10	{B,C,E}

After reading TID=1:

After reading TID=2:



TID	Items
1	{A,B}
2	{B,C,D}
3	{A,C,D,E}
4	{A,D,E}
5	{A,B,C}
6	{A,B,C,D}
7	{A}
8	{A,B,C}
9	{A,B,D}
10	{B,C,E}

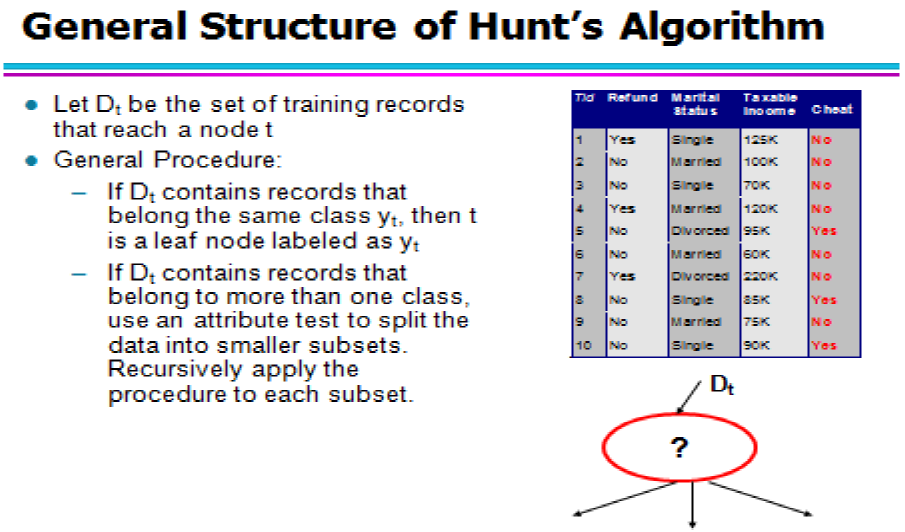
Transaction Database

Header table

Item	Pointer
A	
B	
C	
D	
E	

Pointers are used to assist frequent itemset generation

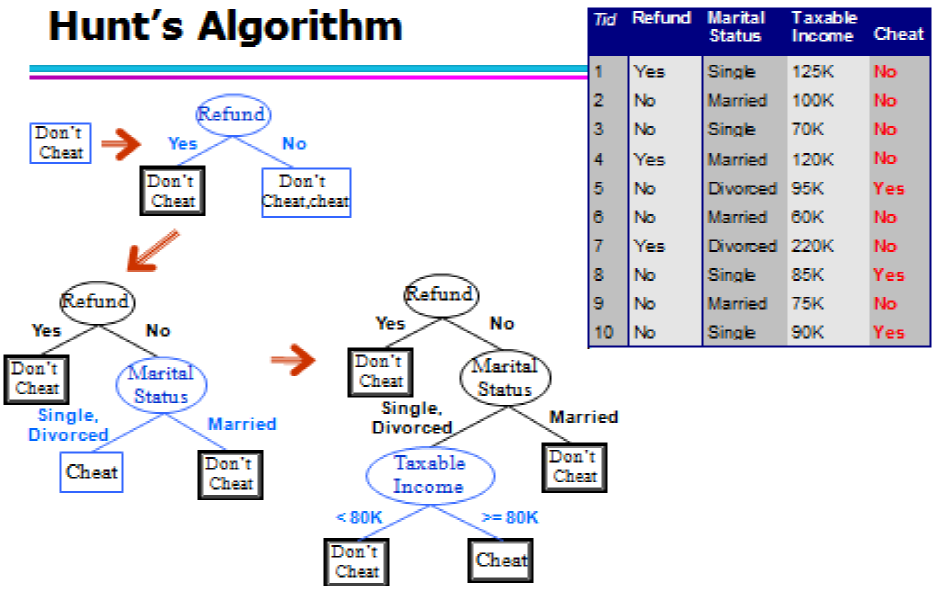
7 a. Write and explain Hunt's algorithm with a suitable example.
Ans:



General Structure of Hunt's Algorithm

- Let D_t be the set of training records that reach a node t
- General Procedure:
  - If D_t contains records that belong the same class y_t, then t is a leaf node labeled as y_t
  - If D_t contains records that belong to more than one class, use an attribute test to split the data into smaller subsets. Recursively apply the procedure to each subset.

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



Hunt's Algorithm

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

b. Explain rule based classifier technique with an example.

Ans: **Rule-Based Classifier**

Classify records by using a collection of "if…then…" rules

Rule: $(Condition) \rightarrow y$

Where *Condition* is a conjunction of attributes and *y* is the class label

LHS: is called rule antecedent or precondition and *RHS* is called rule consequent

Examples of classification rules:

- (Blood Type=Warm) $\wedge$ (Lay Eggs=Yes) $\rightarrow$ Birds
- (Taxable Income < 50K) $\wedge$ (Refund=Yes) $\rightarrow$ Evade=No

Rule-based Classifier (Example)

Name	Blood Type	Give Birth	Can Fly	Live in Water	Class
human	warm	yes	no	no	mammals
python	cold	no	no	no	reptiles
salmon	cold	no	no	yes	fishes
whale	warm	yes	no	yes	mammals
frog	cold	no	no	sometimes	amphibians
komodo	cold	no	no	no	reptiles
bat	warm	yes	yes	no	mammals
pigeon	warm	no	yes	no	birds
cat	warm	yes	no	no	mammals
leopard shark	cold	yes	no	yes	fishes
turtle	cold	no	no	sometimes	reptiles
penguin	warm	no	no	sometimes	birds
porcupine	warm	yes	no	no	mammals
eel	cold	no	no	yes	fishes
salamander	cold	no	no	sometimes	amphibians
gila monster	cold	no	no	no	reptiles
platypus	warm	no	no	no	mammals
owl	warm	no	yes	no	birds
dolphin	warm	yes	no	yes	mammals
eagle	warm	no	yes	no	birds

R1: (Give Birth = no) ∧ (Can Fly = yes) → Birds

R2: (Give Birth = no) ∧ (Live in Water = yes) → Fishes

R3: (Give Birth = yes) ∧ (Blood Type = warm) → Mammals

R4: (Give Birth = no) ∧ (Can Fly = no) → Reptiles

R5: (Live in Water = sometimes) → Amphibians

How does Rule-based Classifier Work?

R1: (Give Birth = no) ∧ (Can Fly = yes) → Birds

R2: (Give Birth = no) ∧ (Live in Water = yes) → Fishes

R3: (Give Birth = yes) ∧ (Blood Type = warm) → Mammals

R4: (Give Birth = no) ∧ (Can Fly = no) → Reptiles

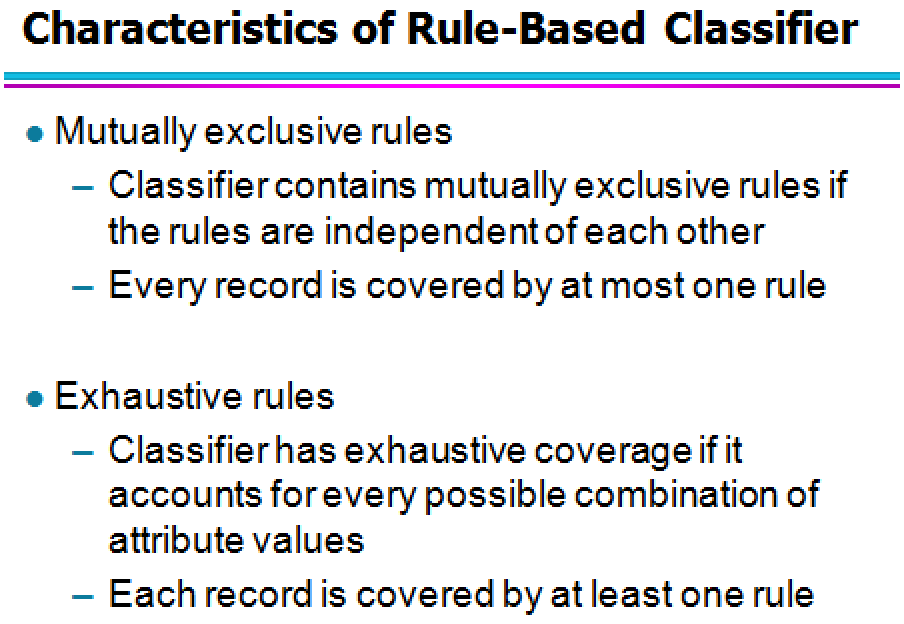
R5: (Live in Water = sometimes) → Amphibians

Name	Blood Type	Give Birth	Can Fly	Live in Water	Class
lemur	warm	yes	no	no	?
turtle	cold	no	no	sometimes	?
dogfish shark	cold	yes	no	yes	?

A lemur triggers rule R3, so it is classified as a mammal

A turtle triggers both R4 and R5

A dogfish shark triggers none of the rules

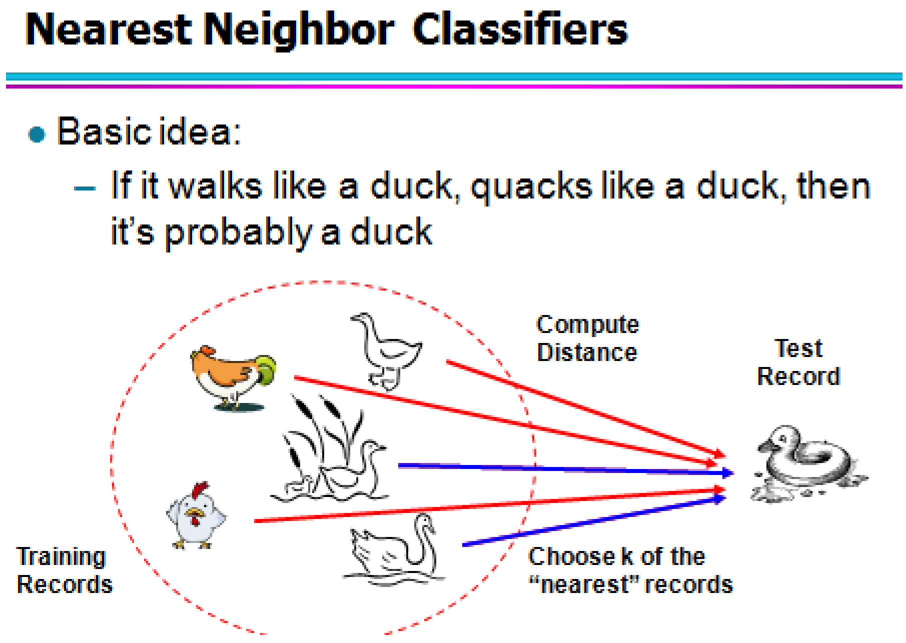


Characteristics of Rule-Based Classifier

- Mutually exclusive rules
  - Classifier contains mutually exclusive rules if the rules are independent of each other
  - Every record is covered by at most one rule
- Exhaustive rules
  - Classifier has exhaustive coverage if it accounts for every possible combination of attribute values
  - Each record is covered by at least one rule

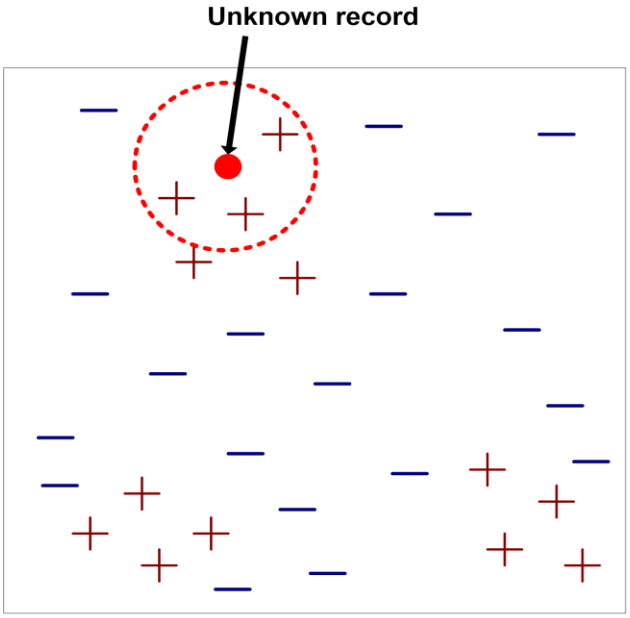
8 a. Write and explain K-nearest neighbor classification algorithm.

Ans:



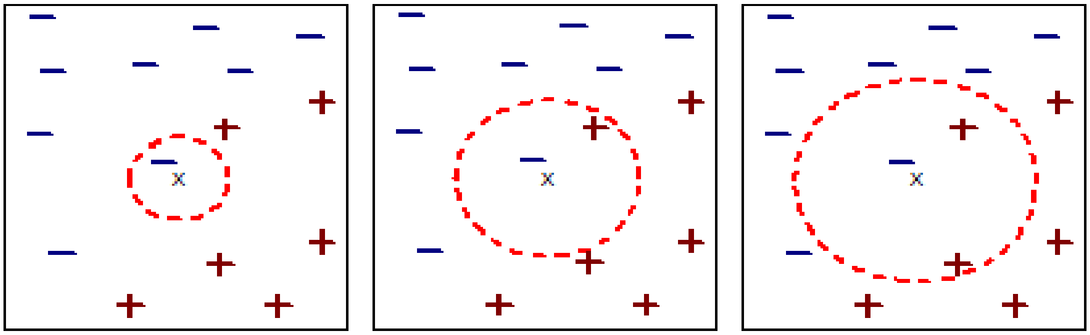
Nearest Neighbor Classifiers

- Basic idea:
  - If it walks like a duck, quacks like a duck, then it's probably a duck



- Requires three things
  - The set of stored records
  - Distance Metric to compute distance between records
  - The value of *k*, the number of nearest neighbors to retrieve
- To classify an unknown record:
  - Compute distance to other training records
  - Identify *k* nearest neighbors
  - Use class labels of nearest neighbors to determine the class label of unknown record (e.g., by taking majority vote)

Definition of Nearest Neighbor



(a) 1-nearest neighbor (b) 2-nearest neighbor (c) 3-nearest neighbor

K-nearest neighbors of a record x are data points that have the k smallest distance to x

Nearest Neighbor Classification

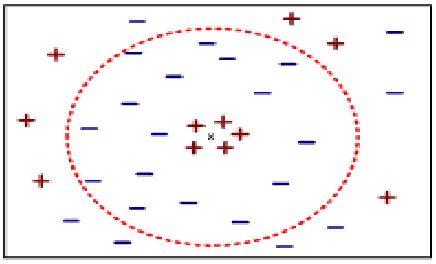
- Compute distance between two points:
  - Euclidean distance

$$d(p,q) = \sqrt{\sum_i (p_i - q_i)^2}$$

- Determine the class from nearest neighbor list
  - take the majority vote of class labels among the k-nearest neighbors
  - Weigh the vote according to distance
    - weight factor, $w = 1/d^2$

Nearest Neighbor Classification…

- Choosing the value of k:
  - If k is too small, sensitive to noise points
  - If k is too large, neighborhood may include points from other classes



Nearest Neighbor Classification…

- Scaling issues
  - Attributes may have to be scaled to prevent distance measures from being dominated by one of the attributes
  - Example:
    - height of a person may vary from 1.5m to 1.8m
    - weight of a person may vary from 90lb to 300lb
    - income of a person may vary from \$10K to \$1M

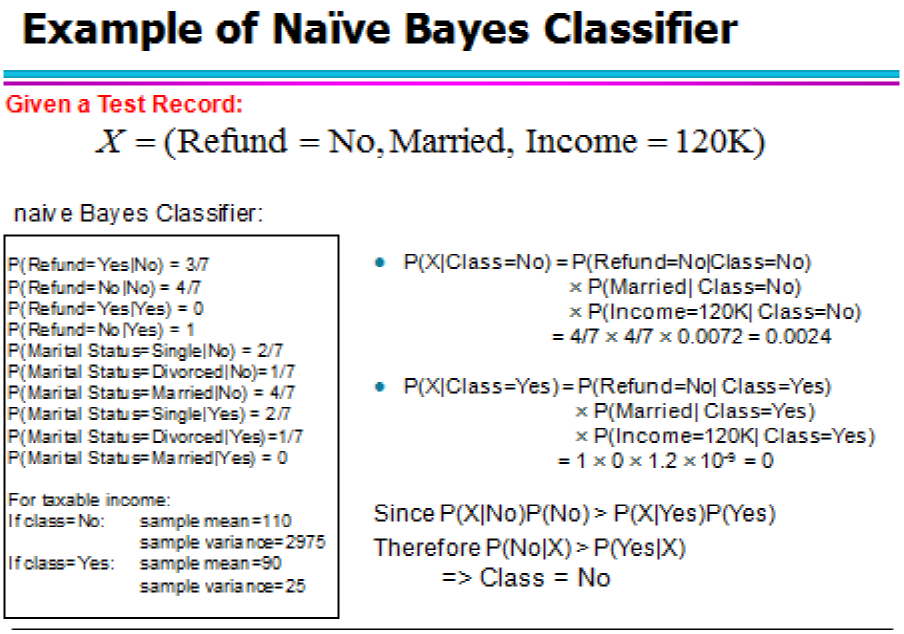
b. Write a note on Naïve Bayes Classifier.

Ans:

- Robust to isolated noise points
- Handle missing values by ignoring the instance during probability estimate calculations
- Robust to irrelevant attributes
- Independence assumption may not hold for some attributes

Naïve Bayes classifier assume independence among attributes A_i when class is given:

- $P(A_1, A_2, \ldots, A_n | C) = P(A_1| C_j) P(A_2| C_j) \ldots P(A_n| C_j)$
- Can estimate $P(A_i| C_j)$ for all A_i and C_j.
- New point is classified to C_j if $P(C_j) \Pi P(A_i| C_j)$ is maximal.



Example of Naïve Bayes Classifier

Given a Test Record:

$$X = (\text{Refund} = \text{No}, \text{Married}, \text{Income} = 120\text{K})$$

naive Bayes Classifier:

P(Refund=Yes|No) = 3/7
P(Refund=No|No) = 4/7
P(Refund=Yes|Yes) = 0
P(Refund=No|Yes) = 1
P(Marital Status=Single|No) = 2/7
P(Marital Status=Divorced|No)=1/7
P(Marital Status=Married|No) = 4/7
P(Marital Status=Single|Yes) = 2/7
P(Marital Status=Divorced|Yes)=1/7
P(Marital Status=Married|Yes) = 0

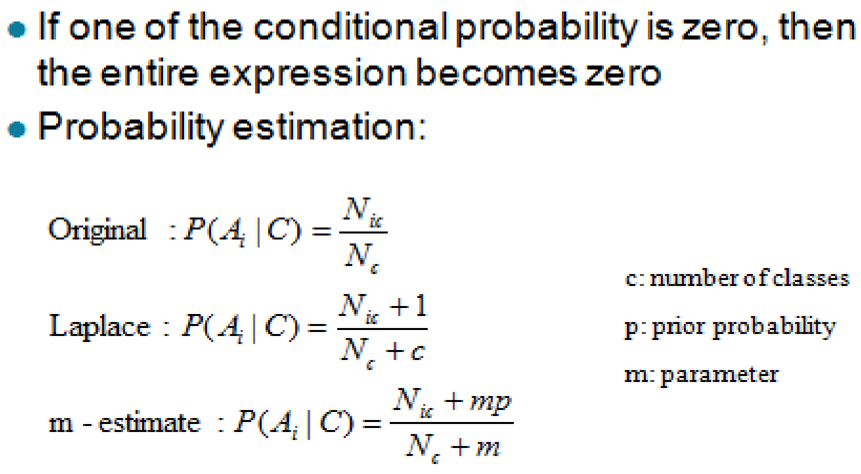
For taxable income:
If class=No: sample mean=110
sample variance=2975
If class=Yes: sample mean=90
sample variance=25

- P(X|Class=No) = P(Refund=No|Class=No)
  × P(Married| Class=No)
  × P(Income=120K| Class=No)
  = 4/7 × 4/7 × 0.0072 = 0.0024
- P(X|Class=Yes) = P(Refund=No| Class=Yes)
  × P(Married| Class=Yes)
  × P(Income=120K| Class=Yes)
  = 1 × 0 × 1.2 × 10^{-9} = 0

Since P(X|No)P(No) > P(X|Yes)P(Yes)

Therefore P(No|X) > P(Yes|X)

=> Class = No



- If one of the conditional probability is zero, then the entire expression becomes zero
- Probability estimation:

$$\text{Original} : P(A_i | C) = \frac{N_{ic}}{N_c}$$

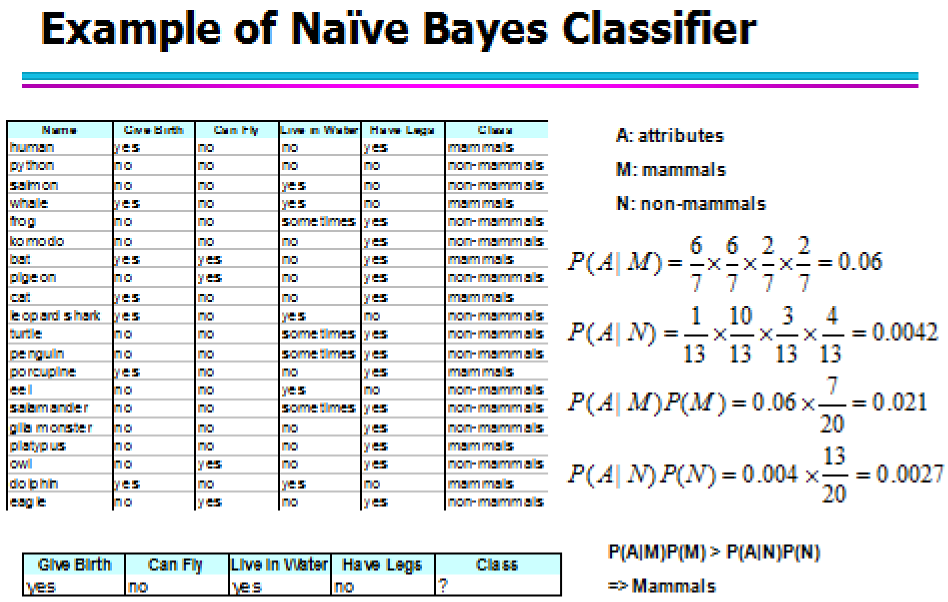
$$\text{Laplace} : P(A_i | C) = \frac{N_{ic} + 1}{N_c + c}$$

$$\text{m - estimate} : P(A_i | C) = \frac{N_{ic} + mp}{N_c + m}$$

c: number of classes

p: prior probability

m: parameter



Example of Naïve Bayes Classifier

Name	Give Birth	Can Fly	Live in Water	Have Legs	Class
human	yes	no	no	yes	mammals
python	no	no	no	no	non-mammals
salmon	no	no	yes	no	non-mammals
whale	yes	no	yes	no	mammals
frog	no	no	sometimes	yes	non-mammals
komodo	no	no	no	yes	non-mammals
bat	yes	yes	no	yes	mammals
pigeon	no	yes	no	yes	non-mammals
cat	yes	no	no	yes	mammals
leopard shark	yes	no	yes	no	non-mammals
turtle	no	no	sometimes	yes	non-mammals
penguin	no	no	sometimes	yes	non-mammals
porcupine	yes	no	no	yes	mammals
eel	no	no	yes	no	non-mammals
salamander	no	no	sometimes	yes	non-mammals
gila monster	no	no	no	yes	non-mammals
platypus	no	no	no	yes	mammals
owl	no	yes	no	yes	non-mammals
dolphin	yes	no	yes	no	mammals
eagle	no	yes	no	yes	non-mammals

A: attributes

M: mammals

N: non-mammals

$$P(A|M) = \frac{6}{7} \times \frac{6}{7} \times \frac{2}{7} \times \frac{2}{7} = 0.06$$

$$P(A|N) = \frac{1}{13} \times \frac{10}{13} \times \frac{3}{13} \times \frac{4}{13} = 0.0042$$

$$P(A|M)P(M) = 0.06 \times \frac{7}{20} = 0.021$$

$$P(A|N)P(N) = 0.004 \times \frac{13}{20} = 0.0027$$

Give Birth	Can Fly	Live in Water	Have Legs	Class
yes	no	yes	no	?

P(A|M)P(M) > P(A|N)P(N)

=> Mammals

c. List out and explain any four Evaluation criteria for classification methods.

Ans: Evaluation Criteria for classification methods:

- Speed
- Robustness
- Scalability
- Interpretability
- Goodness of the model
- Flexibility
- Time complexity

Speed includes

- Time or computation cost of constructing a model
- Time required to learn to use the model.
- Aim- to minimize both times.

Robustness

- Method be able to produce good results in spite of some errors and missing values in datasets.

Scalability

- Method continues to work efficiently for large disk-resident databases as well.

Interpretability

- End-user be able to understand and gain insight from the results produced by the classification method.

Goodness of the model

- It needs to fit the problem that is being solved.

9 a. Mention and explain the desired features of cluster analysis.

Ans: 1. (For large data sets) Scalability: Cluster Analysis method be able to deal with small as well as large problems gracefully.

2. (For large data sets) Only one scan of the dataset: Cluster analysis method should not require more than one scan of the disk-resident data.

3. (For large data sets) Ability to stop and resume: when the data set is very large, cluster analysis may take more time, it is desirable that the task be able to stopped and then resumed when convenient.

4. Minimal input parameters: the user not be expected to have domain knowledge of the data and not be expected to possess insight into clusters that might exist in the data.

5. Robustness: Cluster analysis method be able to deal with noise, outliers and missing values gracefully.

6. Ability to discover different cluster shapes: Cluster analysis method be able to discover cluster shapes other than spherical.

7. Different data types: Cluster analysis methods be able to deal with not only numerical data but also Boolean and categorical data.

8. Result independent of data input order: Cluster analysis method should not be sensitive to data input order. Whatever the order, the result of the same data should be same.

b. Write a note on:

(i) Manhattan distance

(ii) Euclidean distance.

Ans: (i) Manhattan distance: This is also most commonly used metrics also called L_1 norm. Largest valued attribute can dominate the distance

$$dist = (\sum_{k=1}^{n} | p_k - q_k |)$$

Euclidean Distance: This is the most commonly used distance metric also called L_2 norm. Largest valued attribute may dominate the distance.

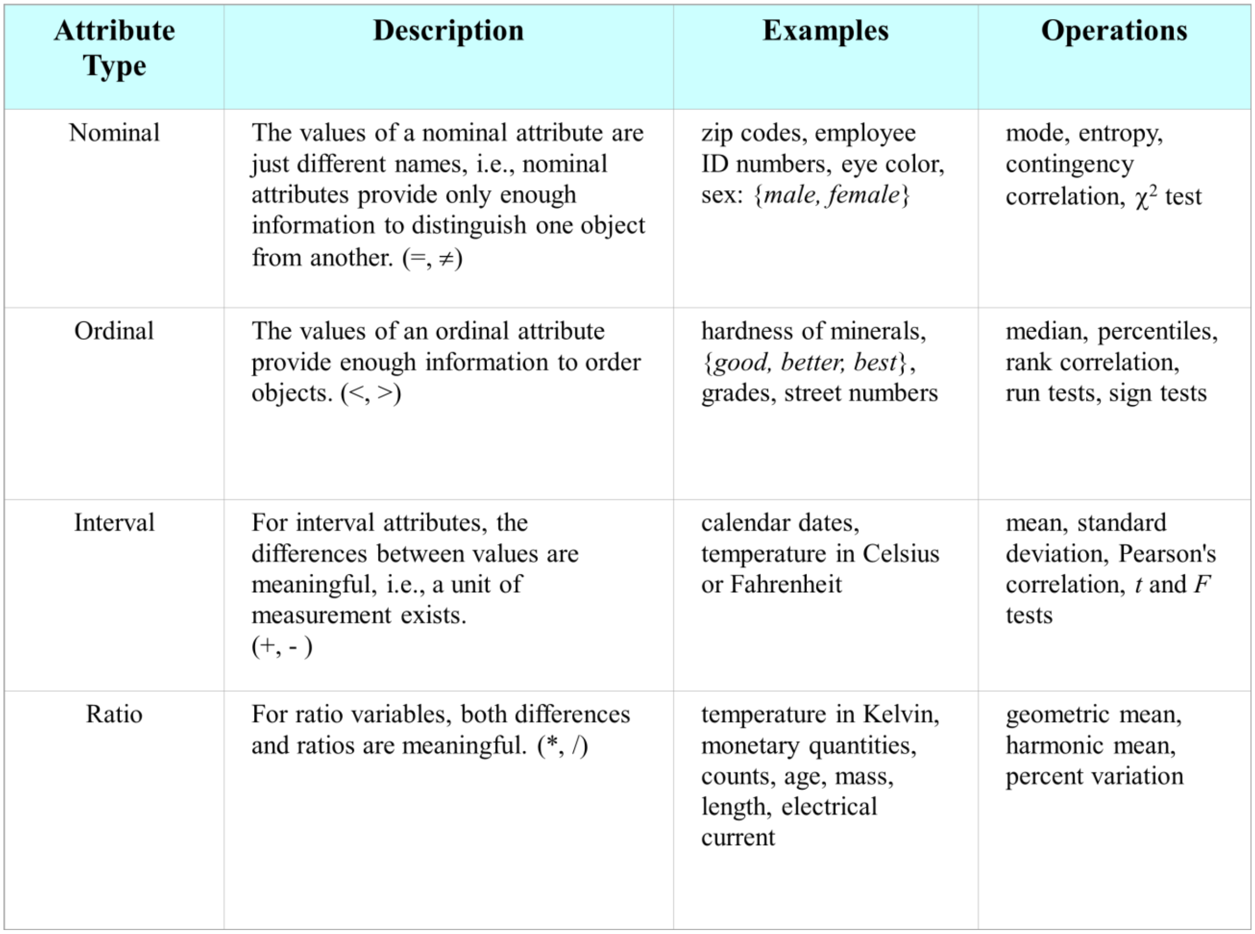
$$dist = (\sum_{k=1}^{n} | p_k - q_k |^{2})^{\frac{1}{2}}$$

c. Briefly explain different types of data used for data mining.

Ans: Types of data:

- l There are different types of attributes
  - Nominal
    - Examples: ID numbers, eye color, zip codes
  - Ordinal
    - Examples: rankings (e.g., taste of potato chips on a scale from 1-10), grades, height in {tall, medium, short}
  - Interval
    - Examples: calendar dates, temperatures in Celsius or Fahrenheit.
  - Ratio
    - Examples: temperature in Kelvin, length, time, counts

- l The type of an attribute depends on which of the following properties it possesses:
  - Distinctness: = ≠
  - Order: < >
  - Addition: + -
  - Multiplication: * /
  - Nominal attribute: distinctness
  - Ordinal attribute: distinctness & order
  - Interval attribute: distinctness, order & addition
  - Ratio attribute: all 4 properties



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

Types of data sets:

Record

- Data Matrix
- Document Data
- Transaction Data

Graph

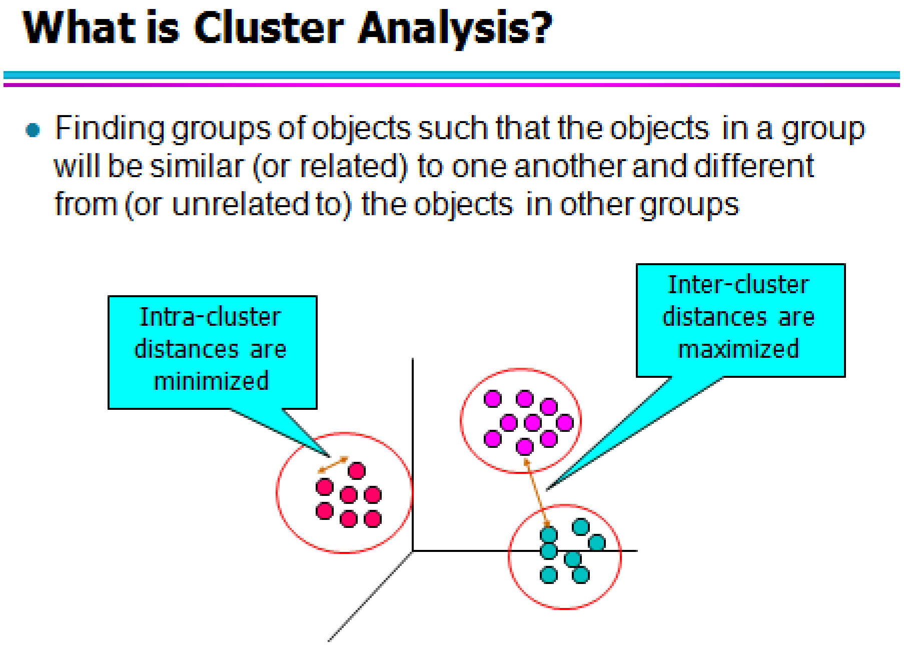
- World Wide Web
- Molecular Structures

Ordered

- Spatial Data
- Temporal Data
- Sequential Data

Genetic Sequence Data

10 a. What is cluster analysis? Briefly explain different types of cluster analysis methods.
Ans:

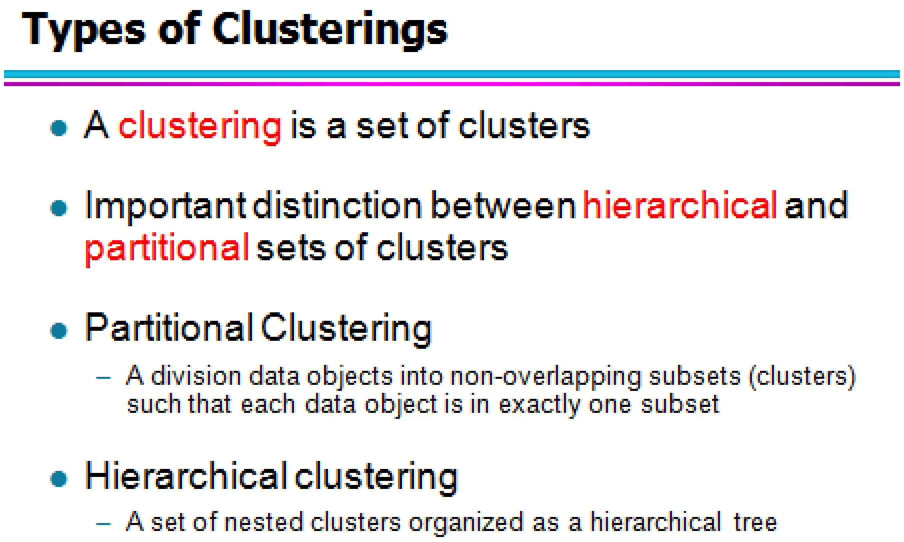


What is Cluster Analysis?

- Finding groups of objects such that the objects in a group will be similar (or related) to one another and different from (or unrelated to) the objects in other groups

Intra-cluster distances are minimized

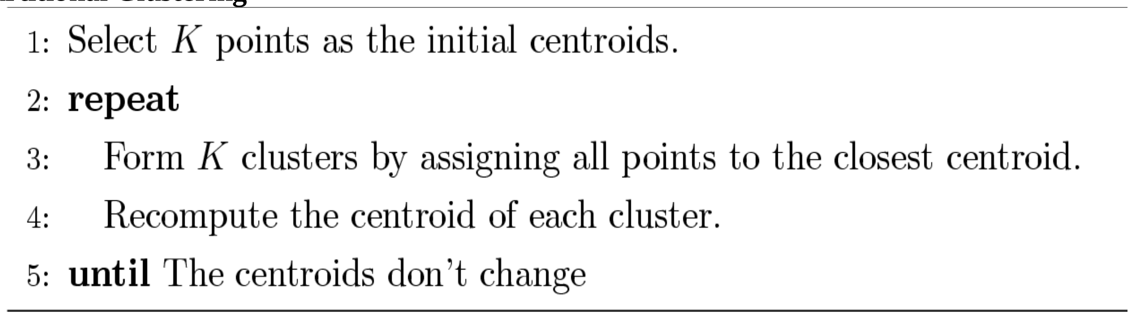
Inter-cluster distances are maximized



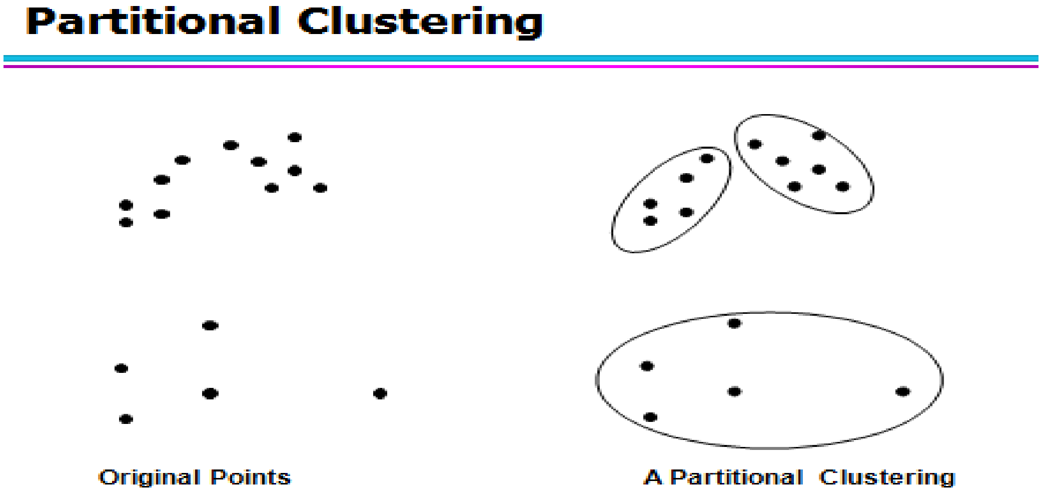
Types of Clusterings

- A clustering is a set of clusters
- Important distinction between hierarchical and partitional sets of clusters
- Partitional Clustering
  - A division data objects into non-overlapping subsets (clusters) such that each data object is in exactly one subset
- Hierarchical clustering
  - A set of nested clusters organized as a hierarchical tree

Partitional Clustering



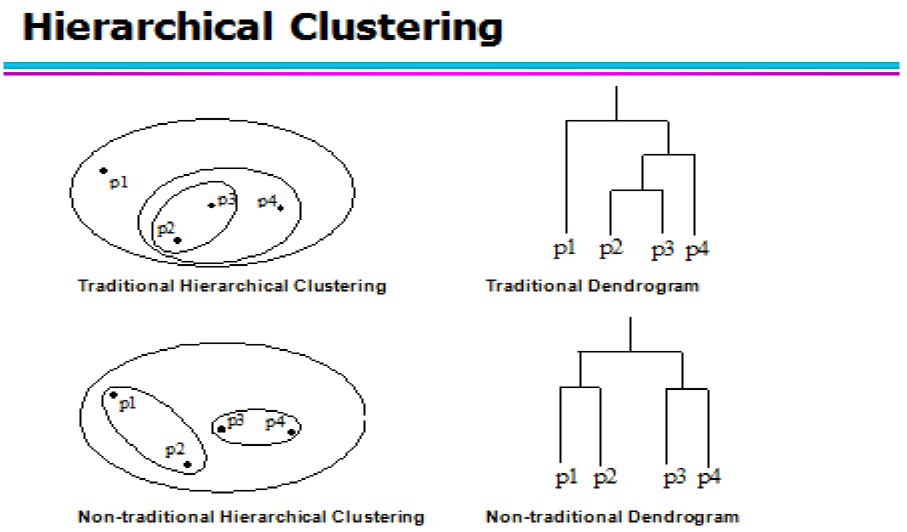
```
Select $K$ points as the initial centroids.
repeat
    Form $K$ clusters by assigning all points to the closest centroid.
    Recompute the centroid of each cluster.
until The centroids don't change
```



Partitional Clustering

Two main types of hierarchical clustering

- Agglomerative:
    - Start with the points as individual clusters
    - At each step, merge the closest pair of clusters until only one cluster (or k clusters) left
- Divisive:
    - Start with one, all-inclusive cluster
    - At each step, split a cluster until each cluster contains a point (or there are k clusters)
    - Traditional hierarchical algorithms use a similarity or distance matrix
- Merge or split one cluster at a time



Hierarchical Clustering

Traditional Hierarchical Clustering

Traditional Dendrogram

Non-traditional Hierarchical Clustering

Non-traditional Dendrogram

Density- based methods: In this method based on density clustering will be done.
Grid based methods: in this class of methods, the object space rather than the data is divided into a grid.

Model Based Methods: a model is assumed, perhaps based on probability distribution.

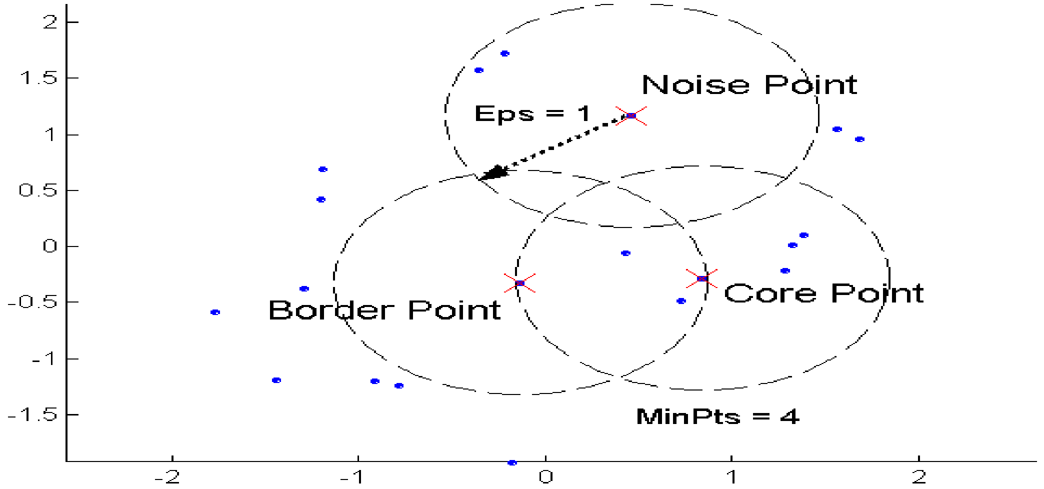
b. Briefly explain DBSCAN algorithm in density based clustering.
Ans: DBSCAN is a density-based algorithm.

Density = number of points within a specified radius (Eps)

- A point is a core point if it has more than a specified number of points (MinPts) within Eps .These are points that are at the interior of a cluster

- A border point has fewer than MinPts within Eps, but is in the neighborhood of a core point

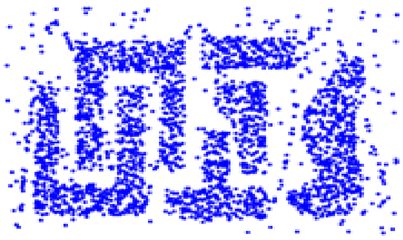
- A noise point is any point that is not a core point or a border point.



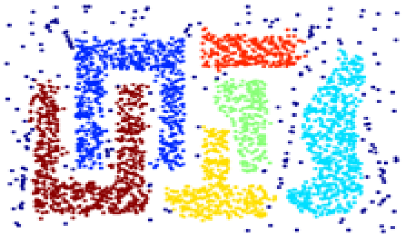
DBSCAN Algorithm

- Label all points as core , border, or noise points.
- Eliminate noise points
- Put an edge between all core points that are within Eps of each other.
- Make each group of connected core points into a separate cluster.
- Assign each border point to one of the clusters of its associated core points.

When DBSCAN Works Well



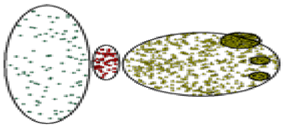
Original Points



Clusters

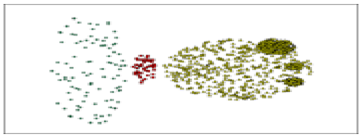
- Resistant to Noise
- Can handle clusters of different shapes and sizes

When DBSCAN Does NOT Work Well

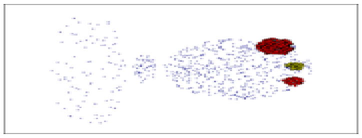


Original Points

- Varying densities
- High-dimensional data



(MinPts=4, Eps=9.75).



(MinPts=4, Eps=9.92)