# MOTHER TERESA WOMEN'S UNIVERSITY

## DIRECTORATE OF DISTANCE EDUCATION

## KODAIKANAL – 624 102

# M.Sc. COMPUTER SCIENCE

## (P.G.)

# MATERIAL FOR DATA MINING

## SUBJECT CODE

**Compiled by Dr. K.Kavitha**

# DATA MINING

**UNIT - I**

**Introduction**– What is Data Mining - Data mining definition – KDD vs Data Mining – DBMS vs Data Mining – DM Techniques – DM Application – Data Warehousing: Introduction – What is a Data Warehouse- Definition – Multidimensional data model – Data Warehousing Architecture – Data Warehouse Back end Process.

**UNIT – II**

**Association Rules** : Introduction – What is an association rule – Methods to discover association rule – A Priori Algorithm – Partition Algorithm – Pincer – Search Algorithm – Dynamic Item set Counting Algorithm – Rapid Association Rule Mining – Incremental Algorithm – Generalized Association Rule.

**UNIT – III**

**Clustering Techniques :** Introduction – Clustering Paradigm – Partitioning Algorithms- K –Medoid Algorithm – CLARANS – Hierarchical Clustering – DB Scan Categorical Clustering Algorithm - ROCK.

**UNIT – IV**

**Decision Trees :** Introduction – Decision Tree Principle – Best Split – Splitting Indicies – ID3 – C4.5 – Rough set theory : Introduction – Definition – Example Reduct – Types of Reduct – Rule Extraction.

**UNIT – V**

**Web Mining:** Introduction – Web Mining - Web Content Mining – Web Structure Mining – Web Usage Mining – Text Mining – Temporal and Spatial Data Mining : Introduction Temporal Association Rule – GSP Algorithm – Spatial Mining Task – Spatial Clustering.

**TEXT BOOK**

1.Arun K. Pujari, “Data Mining Techniques”, Universities Press(India) Limited, 2001.

**REFERENCE BOOK**

1.Pang – Ning Tan, Michael Steinbach, Vipin Kumar, Introduction to Data Mining, Pearson 2008.

2. JiaweiHan, MichelineKamber, Jianpei, “Data Mining Concepts and Techniques:, Morgan Kaufmann Publishers an Imprint of Elsevier, 2012.

# DATA MINING

# UNIT -I

## Introduction

**Data mining** is the process of discovering patterns in large data sets involving methods at the intersection of machine learning, statistics, and database systems. Data mining is an interdisciplinary subfield of computer science and statistics with an overall goal to extract information (with intelligent methods) from a data set and transform the information into a comprehensible structure for further use. Data mining is the analysis step of the "knowledge discovery in databases" process or KDD.

## What is Data Mining

Data mining is the process of analyzing hidden patterns of data according to different perspectives for categorization into useful information, which is collected and assembled in common areas, such as data warehouses, for efficient analysis, data mining algorithms, facilitating business decision making and other information requirements to ultimately cut costs and increase revenue.Data mining is also known as data discovery and knowledge discovery.

Data mining refers to extracting or mining knowledge from large amountsof data. The term is actually a misnomer. Thus, data miningshould have been more appropriately named as knowledge mining which emphasis on mining from large amounts of data. It is the computational process of discovering patterns in large data sets involving methods at the intersection of artificial intelligence, machine learning, statistics, and database systems. The overall goal of the data mining process is to extract information from a data set and transform it into an understandable structure for further use. The key properties of data mining are Automatic discovery of patterns Prediction of likely outcomes Creation of actionable information Focus on large datasets and databases

**Data mining definition**

Data mining is the process of sorting through large data sets to identify patterns and establish relationships to solve problems through data analysis. Data mining tools allow enterprises to predict future trends.

**KDD vs Data Mining**

KDD refers to the overall process of discovering useful knowledge from data. It involves the evaluation and possibly interpretation of the patterns to make the decision of what qualifies as knowledge. It also includes the choice of encoding schemes, preprocessing, sampling, and projections of the data prior to the data mining step. Data mining refers to the application of algorithms for extracting patterns from data without the additional steps of the KDD process.

**DBMS vs Data Mining**

DBMS is a full-fledged system for housing and managing a set of digital databases. However Data Mining is a technique or a concept in computer science, which deals with extracting useful and previously unknown information from raw data. Most of the times, these raw data are stored in very large databases. Therefore Data miners use the existing functionalities of DBMS to handle, manage and even preprocess raw data before and during the Data mining process. However, a DBMS system alone cannot be used to analyze data. But, some DBMS at present have inbuilt data analyzing tools or capabilities.

## DM Techniques

**1.Classification:**

This analysis is used to retrieve important and relevant information about data, and metadata. This data mining method helps to classify data in different classes.

**2. Clustering:**

Clustering analysis is a data mining technique to identify data that are like each other. This process helps to understand the differences and similarities between the data.

**3. Regression:**

Regression analysis is the data mining method of identifying and analyzing the relationship between variables. It is used to identify the likelihood of a specific variable, given the presence of other variables.

**4. Association Rules:**

This data mining technique helps to find the association between two or more Items. It discovers a hidden pattern in the data set.

**5. Outer detection:**

This type of data mining technique refers to observation of data items in the dataset which do not match an expected pattern or expected behavior. This technique can be used in a variety of domains, such as intrusion, detection, fraud or fault detection, etc. Outer detection is also called Outlier Analysis or Outlier mining.

**6. Sequential Patterns:**

This data mining technique helps to discover or identify similar patterns or trends in transaction data for certain period.

**7. Prediction:**

Prediction has used a combination of the other data mining techniques like trends, sequential patterns, clustering, classification, etc. It analyzes past events or instances in a right sequence for predicting a future event.

**Data Mining Application**

**Communications -** Data mining techniques are used in communication sector to predict customer behavior to offer highly targetted and relevant campaigns.

**Insurance -** Data mining helps insurance companies to price their products profitable and promote new offers to their new or existing customers.

**Education -** Data mining benefits educators to access student data, predict achievement levels and find students or groups of students which need extra attention. For example, students who are weak in maths subject.

**Manufacturing -** With the help of Data Mining Manufacturers can predict wear and tear of production assets. They can anticipate maintenance which helps them reduce them to minimize downtime.

**Banking -** Data mining helps finance sector to get a view of market risks and manage regulatory compliance. It helps banks to identify probable defaulters to decide whether to issue credit cards, loans, etc.

**Retail -** Data Mining techniques help retail malls and grocery stores identify and arrange most sellable items in the most attentive positions. It helps store owners to comes up with the offer which encourages customers to increase their spending.

**Service Providers -** Service providers like mobile phone and utility industries use Data Mining to predict the reasons when a customer leaves their company. They analyze billing details, customer service interactions, complaints made to the company to assign each customer a probability score and offers incentives.

**E-Commerce -** E-commerce websites use Data Mining to offer cross-sells and up-sells through their websites. One of the most famous names is Amazon, who use Data mining techniques to get more customers into their eCommerce store.

**Super Markets -** Data Mining allows supermarket's develope rules to predict if their shoppers were likely to be expecting. By evaluating their buying pattern, they could find woman customers who are most likely pregnant. They can start targeting products like baby powder, baby shop, diapers and so on.

**Crime Investigation -** Data Mining helps crime investigation agencies to deploy police workforce (where is a crime most likely to happen and when?), who to search at a border crossing etc.

Bioinformatics - Data Mining helps to mine biological data from massive datasets gathered in biology and medicine.

**Data Warehousing:**

**Introduction**

Data warehousing is the process of constructing and using a data warehouse. A data warehouse is constructed by integrating data from multiple heterogeneous sources that support analytical reporting, structured and/or ad hoc queries, and decision making. Data warehousing involves data cleaning, data integration, and data consolidations.

**What is a Data Warehouse**

A data warehousing is defined as a technique for collecting and managing data from varied sources to provide meaningful business insights. It is a blend of technologies and components which aids the strategic use of data.

It is electronic storage of a large amount of information by a business which is designed for query and analysis instead of transaction processing. It is a process of transforming data into information and making it available to users in a timely manner to make a difference.

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 data warehouse can be used to analyze a particular subject area.

For example, "sales" can be a particular subject. Integrated: A data warehouse integrates data from multiple data sources. For example, source A and source B may have different ways of identifying a product, but in a data warehouse, there will be only a single way of identifying a product. Time-Variant: Historical data is kept in a data

warehouse. For example, one can retrieve data from 3 months, 6 months, 12 months, or even older data from a data warehouse. This contrasts with a transactions system, where often only the most recent data is kept. For example, a transaction system may hold the most recent address of a customer, where a data warehouse can hold all addresses associated with a customer. Non-volatile: Once data is in the data warehouse, it will not change. So, historical data in a data warehouse should never be altered

**Definition**

Data warehousing is the process of constructing and using a data warehouse. A data warehouse is constructed by integrating data from multiple heterogeneous sources that support analytical reporting, structured and/or ad hoc queries, and decision making. Data warehousing involves data cleaning, data integration, and data consolidations.

**Multidimensional data model**

The multidimensional data model is an integral part of On-Line Analytical Processing or OLAP. Because OLAP is on-line, it must provide answers quickly; analysts pose iterative queries during interactive sessions, not in batch jobs that run overnight. And because OLAP is also analytic, the queries are complex. The multidimensional data model is designed to solve complex queries in real-time.

The multidimensional data model is important because it enforces simplicity. As Ralph Kimball states in his landmark book, *The Data Warehouse Toolkit*:"The central attraction of the dimensional model of a business is its simplicity.... that simplicity is the fundamental key that allows users to understand databases, and allows software to navigate databases efficiently."

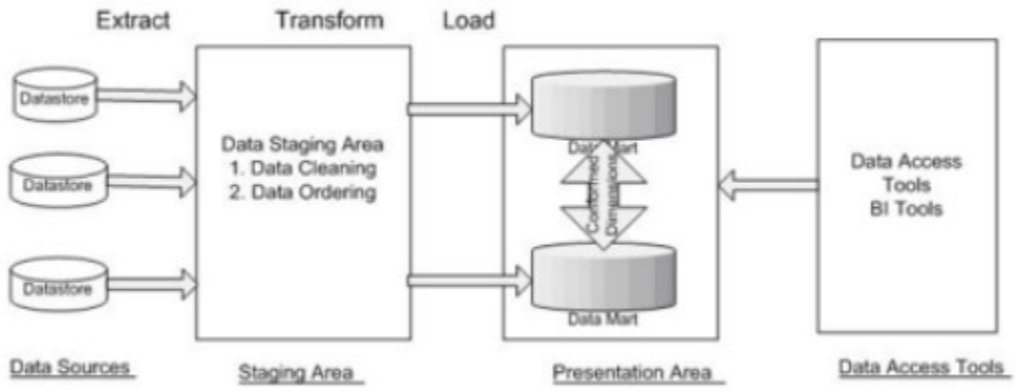
The multidimensional data model is composed of logical cubes, measures, dimensions, hierarchies, levels, and attributes. The simplicity of the model is inherent because it defines objects that represent real-world business entities. Analysts know which business measures they are interested in examining, which dimensions and attributes

make the data meaningful, and how the dimensions of their business are organized into levels and hierarchies.

**Data Warehousing Architecture**

There are two main components to building a data warehouse- an interface design from operational systems and the individual data warehouse design. Thus, the construction of DWH depends on the business requirements, where one development stage depends on the results of previously developed phase.

The structure of a DWH can be understood better through its layered model, which lists the main components of the data warehousing architecture.Below is the typical architecture of data warehouse consisting of different important components.




The movement of data from source to staging area and then finally to conformed data marts through ETL (Extract, Transform and Load) technology.

The first layer is the **Data Source layer**, which refers to various data stores in multiple formats like relational database, Excel file and others. These stores can consists of different types of data – Operational data including business data like Sales, Customer, Finance, Product and others, web server logs, Internet research data and data relating to third party like census, survey.

The next step is **Extract**, where the data from data sources is extracted and put into the warehouse staging area. The extracted data is minimally cleaned with no major transformations.

Then comes the **Staging area**, which is divided into two stages – data cleaning and data ordering. As the name suggests, this layer takes care of data processing methods, i.e. cleaning (removing data redundancy, filtering bad data) and ordering (allowing proper integration) of data. Overall, this stage allows application of business intelligent logic to transform transactional data into analytical data. It is indeed the most time consuming phase in the whole DWH architecture and is the chief process between data source and presentation layer of DWH.

Finally, we have the **Data Presentation** layer, which is the target data warehouse – the place where the successfully cleaned, integrated, transformed and ordered data is stored in a multi-dimensional environment. Now, the data is available for analysis and query purposes. The information is also available to end-users in the form of data marts.This data can then be accessed by various like Tableau, Business Objects, and presented in multiple formats like tables, graphs, reports and others.It is important to note that the data warehouse supports and holds both persistent (stored for longer time) and transient/temporary data. The major purpose of a data warehouse is the attainment of cleansed, integrated and properly aligned data so that it is easy to analyze and present to clients and customers in several businesses.

**Data Warehouse Back end Process.**

**Data extraction:**

- get data from multiple, heterogeneous, and external sources

**Data cleaning:**

- detect errors in the data and rectify them when possible

**Data transformation:**

- convert data from legacy or host format to warehouse format

**Load:**

- sort, summarize, consolidate, compute views, check integrity, and build indicies and partitions

**Refresh**

- propagate the updates from the data sources to the warehouse

# UNIT – II

# ASSOCIATION RULES

**Association Rules** :

**Introduction:**

Association rule learning is a rule-based machine learning method for discovering interesting relations between variables in large databases. It is intended to identify strong rules discovered in databases using some measures of interestingness.

**What is an association rule:**

Association rules are if-then statements that help to show the probability of relationships between data items within large data sets in various types of databases. Association rule mining has a number of applications and is widely used to help discover sales correlations in transactional data or in medical data sets.

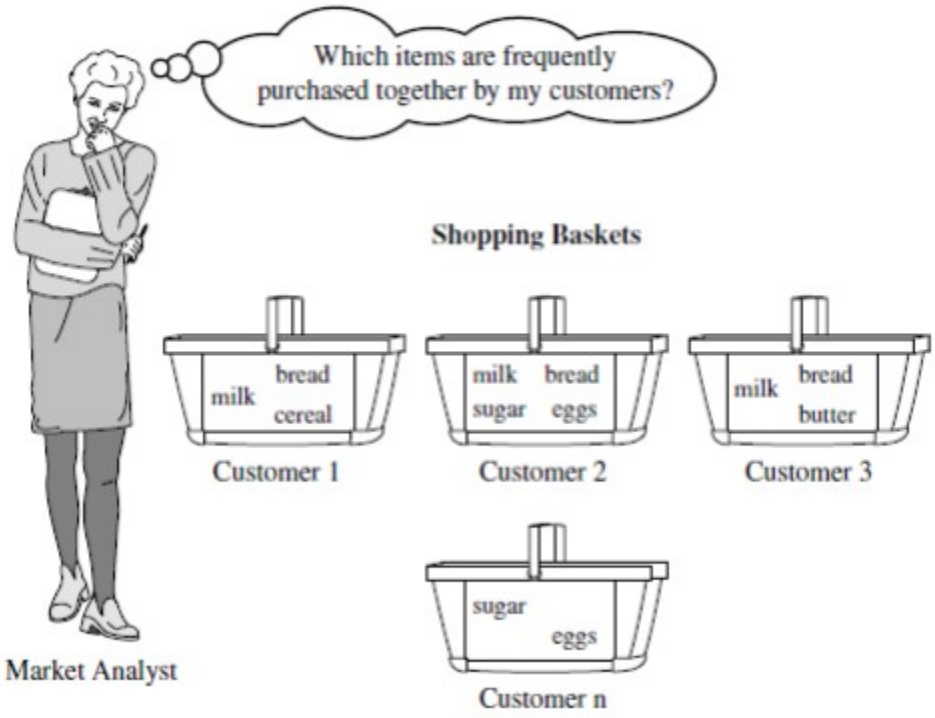
Association rule mining, at a basic level, involves the use of machine learning models to analyze data for patterns, or co-occurrence, in a database. It identifies frequent if-then associations, which are called *association rules*.An association rule has two parts: an antecedent (if) and a consequent (then). An antecedent is an item found within the data. A consequent is an item found in combination with the antecedent.

Association rules are created by searching data for frequent if-then patterns and using thecriteria *support* and *confidence* toidentify the most importantrelationships. Support isan indication of how frequently the items appear in the data. Confidence indicates the number of times the if-then statements are found true. A third metric, called *lift*, can be used to compare confidence with expected confidence.

Association rules are calculated from *itemsets*, which are made up of two or more items. If rules are built from analyzing all the possible itemsets, there could be so many rules that the rules hold little meaning. With that, association rules are typically created from rules well-represented in data.

**Market basket analysis:**

This process analyzes customer buying habits by finding associations between the different items thatcustomers place in their shopping baskets. The discovery of such associationscan help retailers develop marketing strategies by gaining insight into which itemsare frequently purchased together by customers. For instance, if customers are buyingmilk, how likely are they to also buy bread (and what kind of bread) on the same trip to the supermarket. Such information can lead to increased sales by helping retailersdoselective marketing and plan their shelf space.




**Example:**

If customers who purchase computers also tend to buy antivirussoftware at the same time, then placing the hardware display close to the software displaymay help increase the sales of both items. In an alternative strategy, placing hardware andsoftware at opposite ends of the store may entice customers who purchase such items topick up other items along the way. For instance, after deciding on an expensive computer,a customer may observe security systems for sale while heading toward the software

displayto purchase antivirus software and may decide to purchase a home security systemas well. Market basket analysis can also help retailers plan which items to put on saleat reduced prices. If customers tend to purchase computers and printers together, thenhaving a sale on printers may encourage the sale of printers as well as computers

## Methods to discover association rule

Association rule learning is a rule-based machine learning method for discovering interesting relations between variables in large databases. It is intended to identify strong rules discovered in databases using some measures of interestingness.

## Apriori Algorithm

For finding frequent itemsets in a dataset for boolean association rule. Name of the algorithm is Apriori because it uses prior knowledge of frequent itemset properties. We apply an iterative approach or level-wise search where k-frequent itemsets are used to find k+1 itemsets.To improve the efficiency of level-wise generation of frequent itemsets, an important property is used called Apriori property which helps by reducing the search space.

**Efficient Frequent Itemset Mining Methods:**

**Finding Frequent Itemsets Using Candidate Generation:**

**The Apriori Algorithm**

Apriori is a seminal algorithm proposed by R. Agrawal and R. Srikant in 1994 for mining frequent itemsets for Boolean association rules. The name of the algorithm is based on the fact that the algorithm uses prior knowledge of frequent itemset properties. Apriori employs an iterative approach known as a level-wise search, where k-itemsets are used to explore (k+1)-itemsets. First, the set of frequent 1-itemsets is found by scanning the database to accumulate the count for each item, and collecting those items that satisfy minimum support. The resulting set is denoted L1.Next, L1 is used to find L2, the set of frequent 2-itemsets, which is used to find L3, and so on, until no more frequent

k-itemsets can be found. The finding of each Lkrequires one full scan of the database. A two-step process is followed in Aprioriconsisting of joinand prune action.

**Apriori Property**

All non-empty subset of frequent itemset must be frequent. The key concept of Apriori algorithm is its anti-monotonicity of support measure. Apriori assumes thatBefore we start understanding the algorithm, go through some definitions which are explained in my previous post.

Consider the following dataset and we will find frequent itemsets and generate association rules for them.

| TID | items |
|---|---|
| T1 | I1, I2 , I5 |
| T2 | I2,I4 |
| T3 | I2,I3 |
| T4 | I1,I2,I4 |
| T5 | I1,I3 |
| T6 | I2,I3 |
| T7 | I1,I3 |
| T8 | I1,I2,I3,I5 |
| T9 | I1,I2,I3 |

minimum support count is 2

minimum confidence is 60%

**Step-1**: K=1

(I) Create a table containing support count of each item present in dataset –

Called C1(candidate set)

| Itemset | sup_count |
|---|---|
| I1 | 6 |
| I2 | 7 |
| I3 | 6 |
| I4 | 2 |
| I5 | 2 |

(II) compare candidate set item's support count with minimum support count(here min_support=2 if support_count of candidate set items is less than min_support then remove those items). This gives us itemset L1.

| Itemset | sup_count |
|---|---|
| I1 | 6 |
| I2 | 7 |
| I3 | 6 |
| I4 | 2 |
| I5 | 2 |

**Step-2:** K=2

- Generate candidate set C2 using L1 (this is called join step). Condition of joining Lk-1 and Lk-1 is that it should have (K-2) elements in common.
- Check all subsets of an itemset are frequent or not and if not frequent remove that itemset.(Example subset of{I1, I2} are {I1}, {I2} they are frequent.Check for each itemset)
- Now find support count of these itemsets by searching in dataset.

| Itemset | sup_count |
|---|---|
| I1,I2 | 4 |
| I1,I3 | 4 |
| I1,I4 | 1 |
| I1,I5 | 2 |
| I2,I3 | 4 |
| I2,I4 | 2 |
| I2,I5 | 2 |
| I3,I4 | 0 |
| I3,I5 | 1 |
| I4,I5 | 0 |

(II) compare candidate (C2) support count with minimum support count(here min_support=2 if support_count of candidate set item is less than min_support then remove those items) this gives us itemset L2.

| Itemset | sup_count |
|---|---|
| I1,I2 | 4 |
| I1,I3 | 4 |
| I1,I5 | 2 |
| I2,I3 | 4 |
| I2,I4 | 2 |
| I2,I5 | 2 |
| I2,I5 | 2 |

**Step-3:**

- Generate candidate set C3 using L2 (join step). Condition of joining Lk-1 and

  Lk-1 is that it should have (K-2) elements in common. So here, for L2, first element should match.So itemset generated by joining L2 is {I1, I2, I3}{I1, I2, I5}{I1, I3, i5} {I2, I3, I4}{I2, I4, I5}{I2, I3, I5}
- Check if all subsets of these itemsets are frequent or not and if not, then remove that itemset.(Here subset of {I1, I2, I3} are {I1, I2},{I2, I3},{I1, I3} which are frequent. For {I2, I3, I4}, subset {I3, I4} is not frequent so remove it. Similarly check for every itemset)
- find support count of these remaining itemset by searching in dataset.

| Itemset | sup_count |
|---|---|
| I1,I2,I3 | 2 |
| I1,I2,I5 | 2 |

(II) Compare candidate (C3) support count with minimum support count(here min_support=2 if support_count of candidate set item is less than min_support then remove those items) this gives us itemset L3.

| Itemset | sup_count |
|---|---|
| I1,I2,I3 | 2 |
| I1,I2,I5 | 2 |

**Step-4**:

- Generate candidate set C4 using L3 (join step). Condition of joining Lk-1 and

Lk-1 (K=4) is that, they should have (K-2) elements in common. So here, for L3, first 2 elements (items) should match.

- Check all subsets of these itemsets are frequent or not (Here itemset formed by joining L3 is {I1, I2, I3, I5} so its subset contains {I1, I3, I5}, which is not frequent). So no itemset in C4
- We stop here because no frequent itemsets are found further

Thus, we have discovered all the frequent item-sets. Now generation of strong association rule comes into picture. For that we need to calculate confidence of each rule.

Confidence-

A confidence of 60% means that 60% of the customers, who purchased milk and bread also bought butter.

Confidence(A->B)=Support_count(A∪B)/Support_count(A)

So here, by taking an example of any frequent itemset, we will show the rule generation.

Itemset {I1, I2, I3} //from L3

SO rules can be

[I1^I2]=>[I3] //confidence = sup(I1^I2^I3)/sup(I1^I2) = 2/4*100=50%

[I1^I3]=>[I2] //confidence = sup(I1^I2^I3)/sup(I1^I3) = 2/4*100=50%

[I2^I3]=>[I1] //confidence = sup(I1^I2^I3)/sup(I2^I3) = 2/4*100=50%

[I1]=>[I2^I3] //confidence = sup(I1^I2^I3)/sup(I1) = 2/6*100=33%

[I2]=>[I1^I3] //confidence = sup(I1^I2^I3)/sup(I2) = 2/7*100=28%

[I3]=>[I1^I2] //confidence = sup(I1^I2^I3)/sup(I3) = 2/6*100=33%

So if minimum confidence is 50%, then first 3 rules can be considered as strong association rules.

**Partition Algorithm**

The simplest and most fundamental version of cluster analysis is partitioning, which organizes the objects of a set into several exclusive groups or clusters. To keep the problem specification concise, we can assume that the number of clusters is given as background knowledge. This parameter is the starting point for partitioning methods. It attempts to find the frequent item sets in a bottom – up manner but, at the same time, it

maintains a list of maximal frequent item sets. While making a database pass, it also counts the support of these candidate maximal frequent item sets to see if any one of these is actually frequent. In that event, it can conclude that all the subsets of these frequent sets are going to be frequent and, hence, they are not verified for the support count in the next pass. If we are lucky, we may discover a very large maximal frequent item set very early in the algorithm. If this set subsumes all the candidate sets of level k, then we need not proceed further and thus we save many database passes. Clearly, the pincer – search has an advantage over a priori algorithm when the largest frequent item set is long.

Formally, given a data set, *D*, of *n* objects, and *k*, the number of clusters to form, a **partitioning algorithm** organizes the objects into *k* partitions ($k \leq n$), where each partition represents a cluster. The clusters are formed to optimize an objective partitioning criterion, such as a dissimilarity function based on distance, so that the objects within a cluster are “similar” to one another.

Apriori algorithm operates in a bottom – up, breadth – first search method. The computation starts from the smallest set of frequent item sets and moves upward till it reaches the largest frequent item set the number of database passes is equal to the largest size of the frequent item set. When any one of the frequent item sets becomes longer, the algorithm has to go through many iterations and, as a result, the performance decreases.

**Pincer Search Algorithm**

An Efficient Algorithm for Discovering the Maximum Frequent Set. Abstract - Discovering frequent itemsets is a key problem in important data mining applications, such as the discovery of association rules, strong rules, episodes, and minimal keys.

**Algorithm**

$L_0$ := null; $k$ := 1; $C_1$ := {{ $i$} | $i$ belong to $I_0$}

MFCS := $I_0$; MFS := null;

while$C_k$ != null
read database and count supports for $C_k$ and MFCS
remove frequent itemsets from MFCS and add them to MFS
determine frequent set $L_k$ and infrequent set $S_k$
use$S_k$ to update MFCS
generate new candidate set $C_{k+1}$(join, recover, and prune)
$k$:= $k$ +1
return MFS

**Dynamic Item set Counting Algorithm**

This algorithm was proposed by Bin et al. in 1997. The rationale behind DIC is that it works like a train running over the data, with stops at intervals M between transactions. When the train reaches the end of the transaction file, it has made one pass over the data, and it starts all over again from the beginning for the next pass. The passengers on the train are itemsets. When an itemset is on the train, we count its occurrence in the transactions that are read. When an a priori algorithm is considered in this metaphor, all itemsets get on at the start of a pass and get off at the end. The 1 – itemsets take the first pass, the 2 – itemsets take the second pass, and so on. In DIC, there is the added flexibility of allowing itemsets to get on at any stop as long as they get off at the same stop the next time the train goes around. Therefore, the itemset has seen all the transactions in the file.

Consider for example, if we are mining 40,000 transactions and M = 10,000, we will count all the 1 – itemsets in the first 40,000 transactions we read. However, we will begin counting 2 – itemsets after the first 10,000 transactions have been read. We will begin counting 3 itemsets after 20,000 transactions. For now, let us assume that there are no 4 – itemsets we need to count. Once we get to the end of the file, we will top counting the 1 – itemsets and go back to the start of the file to count the 2 – and 3 – itemsets. After the first 10,000 transactions, we will finish counting the 2 – itemsets and after 20,000 transaction, we will finish counting the 3 – itemsets. In total, we have made 1.5 passes over the data instead of the 3 passes a level – wise algorithm would make.

Initially, we identify certain „stops" in the database. It is assumed that we read the records sequentially as we do in other algorithms, but pause to carryout certain computations at the „stop" points. For notational convenience, we assign numbers to each stop sequentially.

We then define four different structures:

- ♣ Dashed Box
- ♣ Dashed Circle
- ♣ Solid Box
- ♣ Solid Circle

Each of these structures maintains a list of itemsets. Itemsets in the „dashed" category of structures have a counter and the stop number with them. The counter is to keep track of the support value of the corresponding itemset. To stop umber is to keep track whether an itemset has completed one full pass over a database.

The itemsets in the "solid" category structures are not subjected to any counting. The itemset in the solid box is the confirmed set of frequent sets. The itemsets in the solid circle are the confirmed set of infrequent sets.

The algorithm counts the support values of the itemsets in the dashed structure as it moves along from one stop point to another.

During the execution of the algorithm, at any stop point, the following events take place:

- ♣ Certain itemsets in the dashed circle move into the dashed box. These are the itemsets whose support – counts reach value during this iteration (reading records between two consecutive stops).
- ♣ Certain itemsets enter afresh into the system and get into the dashed circle. These are essentially the supersets of the itemsets that move from the dashed circle to the dashed box.

- ♣ The itemsets that have completed one full pass, move from the dashed structure to solid structure. That is, if the itemset is in a dashed circle while completing a full pass, it moves to the solid circle. If it is in the dashed box, then it moves into solid box after completing a full pass.

We shall formally describe the DIC algorithm now.

**DIC Algorithm**

Initially,
Solid box contains the empty itemset;
Solid circle is empty;
Dashed box is empty;
Dashed circle contains all 1 – itemsets with the respective stop – number as 0;
Current stop – number := 0;
**do** until the dashed circle is empty read the database till the next stop point and increase the counters of the itemsets in the dashed box and in the dashed circle as we go along, record by record, to reach the next stop.
increase the current – stop – number by 1;
**for** each itemset in the dashed circle
**if** count of the itemset is greater than
**then** move the itemset to the dashed box
generate a new itemset to be put into the dashed circle
with counter value = 0 and stop number = current stop number.
**else**
**if** its stop number is equal to the current stop number
**then** move this itemset to solid circle.
**for** each itemset in the dashed box
**if** its stop – number is equal to the current stop umber
**then** move this itemset to the solid box
**end**
**return** the itemsets in solid box

**Rapid Association Rule Mining**

Rapid Association Rule Mining. Association rule mining is a well-researched area where many algorithms have been proposed to improve the speed of mining.... It achieves significant speed-ups because the main bottleneck in association rule mining using the Apriori property is the generation of candidate 2-itemsets.

- Propose an innovative algorithm to push the speed-up barrier
- RARM uses a tree structure─Support-OrderedTrieItemset (SOTrieIT)
    - Hold pre-processed transactional data quickly discover large 1-itemsets and 2-itemsets without scanning the database and without candidate 2-itemsets generation

## Incremental Algorithm

The mining of association rules on transactional database is usually an offline process since it is costly to find the association rules in large databases. With usual market-basket applications, new transactions are generated and old transactions may be obsolete as time advances. As a result, incremental updating techniques should be developed for maintenance of the discovered association rules to avoid redoing mining on the whole updated database. A database may allow frequent or occasional updates and such updates may not only invalidate existing association rules but also activate new rules. Thus it is nontrivial to maintain such discovered rules in large databases.

Considering an original database and newly inserted transactions, the following four cases may arise:

Case 1: An itemset is large in the original database and in the newly inserted transactions.

Case 2: An itemset is large in the original database, but is not large in the newly inserted transactions.

Case 3: An itemset is not large in the original database, but is large in the newly inserted transactions.

Case 4: An itemset is not large in the original database and in the newly inserted transactions.

Since itemsets in Case 1 are large in both the original database and the new transactions, they will still be large after the weighted average of the counts. Similarly, itemsets in Case 4 will still be small after the new transactions are inserted. Thus Cases 1 and 4 will not affect the final association rules. Case 2 may remove existing association rules, and Case 3 may add new association rules.

A good rule-maintenance algorithm should thus accomplish the following: 1. Evaluate large itemsets in the original database and determine whether they are still large in the updated database; 2. Find out whether any small itemsets in the original database may become large in the updated database; 3. Seek itemsets that appear only in the newly inserted transactions and determine whether they are large in the updated database.

## Generalized Association Rule.

- Users are interested in generating rules that span different levels of the taxonomy.
- Rules of lower levels may not have minimum support
- Taxonomy can be used to prune uninteresting or redundant rules
- Multipletaxonomies may be present.
  **For example:** category, price(cheap, expensive), “items-on-sale”. etc.
- Multiple taxonomies may be modeled as a forest, or a DAG.

**Notations:**

- $I = \{i_1, i_2, \ldots, i_m\}$- items.
- T- transaction, set of items $T \subseteq I$
  (we expect the items in T to be leaves in T .)
- D – set of transactions
- T *supports* item x, if x is in T or x is an ancestor of some item in T.
- T *supports* $X \subseteq I$ if it supports every item in X.

- A *generalized association rule*: X➔ Y
  if X⊂I , Y⊂I , X∩Y = ⊘ , and no item in Y is an ancestor of any item in X.
- The rule X➔Y has *confidence* c in D if c% of transactions in D that support X also support Y.
- The rule X➔Y has *support* s in D if s% of transactions in D supports X∪Y.

To find all generalized association rules that have support and confidence greater than the user-specified minimum support (called *minsup*) and minimum confidence (called *minconf*) respectively.

# UNIT – III

# CLUSTERING TECHNIQUES

**Clustering** is the process of making a group of abstract objects into classes of similar objects. A **cluster** of **data** objects can be treated as one group. Whiledoing **cluster** analysis, we first partition the set of **data** into groups based on **data** similarity and then assign the labels to the groups.

**Introduction**

The method of identifying similar groups of data in a dataset is called clustering. It is one of the most popular techniques in data science. Entities in each group are comparatively more similar to entities of that group than those of the other groups.

Clustering is the task of dividing the population or data points into a number of groups such that data points in the same groups are more similar to other data points in the same group than those in other groups. In simple words, the aim is to segregate groups with similar traits and assign them into clusters.

Suppose, you are the head of a rental store and wish to understand preferences of your costumers to scale up your business. Is it possible for you to look at details of each costumer and devise a unique business strategy for each one of them? Definitely not. But, what you can do is to cluster all of your costumers into say 10 groups based on their purchasing habits and use a separate strategy for costumers in each of these 10 groups.

**Clustering Paradigms**

The task of clustering is subjective, the means that can be used for achieving this goal are plenty. Every methodology follows a different set of rules for defining the '*similarity'* among data points.

- **Connectivity models:** As the name suggests, these models are based on the notion that the data points closer in data space exhibit more similarity to each other than the data points lying farther away. These models can follow two approaches. In the first approach, they start with classifying all data points into separate clusters & then aggregating them as the distance decreases. In the second approach, all data points are classified as a single cluster and then partitioned as the distance increases. Also, the choice of distance function is subjective. These models are very easy to interpret but lacks scalability for handling big datasets. Examples of these models are hierarchical clustering algorithm and its variants.
- **Centroid models:** These are iterative clustering algorithms in which the notion of similarity is derived by the closeness of a data point to the centroid of the clusters. K-Means clustering algorithm is a popular algorithm that falls into this category. In these models, the no. of clusters required at the end have to be mentioned beforehand, which makes it important to have prior knowledge of the dataset. These models run iteratively to find the local optima.
- **Distribution models:** These clustering models are based on the notion of how probable is it that all data points in the cluster belong to the same distribution (For example: Normal, Gaussian). These models often suffer from overfitting. A popular example of these models is Expectation-maximization algorithm which uses multivariate normal distributions.
- **Density Models:** These models search the data space for areas of varied density of data points in the data space. It isolates various different density regions and assign the data points within these regions in the same cluster. Popular examples of density models are DBSCAN and OPTICS.

## Partitioning Algorithms

The simplest and most fundamental version of cluster analysis is partitioning, which organizes the objects of a set into several exclusive groups or clusters. To keep the problem specification concise, we can assume that the number of clusters is given as background knowledge. This parameter is the starting point for partitioning methods.

Formally, given a data set D of n objects, and k, the number of clusters to form, a partitioning algorithm organizes the objects into k partitions (k ≤ n), where each partition represents a cluster.

Suppose we are given a database of 'n' objects and the partitioning method constructs 'k' partition of data. Each partition will represent a cluster and k ≤ n. It means that it will classify the data into k groups, which satisfy the following requirements −

Each group contains at least one object.

Each object must belong to exactly one group.

The clusters are formed to optimize an objective partitioning criterion, such as a dissimilarity function based on distance, so that the objects within a cluster are "similar" to one another and "dissimilar" to objects in other clusters in terms of the data set attributes.

## K –Medoid Algorithm

K-Medoids (also called as Partitioning AroundMedoid) algorithm was proposed in 1987 by Kaufman and Rousseeuw. A medoid can be defined as the point in the cluster, whose dissimilarities with all the other points in the cluster is minimum.k-means is sensitive to outliers because an object far away from the majority of the data, when assigned to a cluster, can distract the mean value of the cluster dramatically. This inadvertently affects the assignment of other objects to clusters.

The dissimilarity of the medoid(Ci) and object(Pi) is calculated by using E = |Pi - Ci|

The cost in K-Medoids algorithm is given as –

$$c = \sum_{Ci} \sum_{Pi \in Ci} |Pi - Ci|$$

***Algorithm***

*1. Initialize: select k random points out of the n data points as the medoids.*

*2. Associate each data point to the closed medoid by using any common distance metric methods.*

*3.While the cost decreases:*

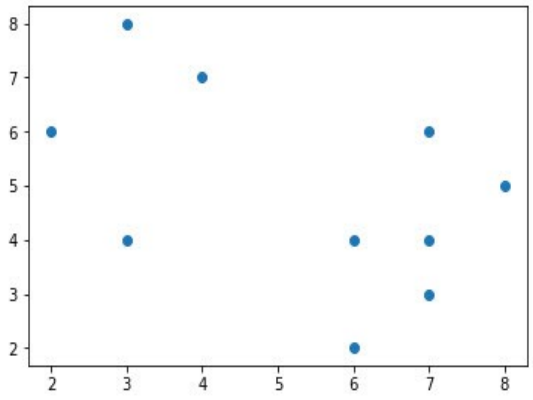
*For each medoid m, for each data o point which is not a medoid:*

1. *Swap m and o, associate each data point to the closest medoid, recompute the cost.*
2. *If the total cost is more than that in the previous step, undo the swap.*

Let's consider the following example:

| | X | Y |
|---|---|---|
| 0 | 7 | 6 |
| 1 | 2 | 6 |
| 2 | 3 | 8 |
| 3 | 8 | 5 |
| 4 | 7 | 4 |
| 5 | 4 | 7 |
| 6 | 6 | 2 |
| 7 | 7 | 3 |
| 8 | 6 | 4 |
| 9 | 3 | 4 |

If a graph is drawn using the above data points, we obtain the following:



**Step #1:** k = 2

Let the randomly selected 2 medoids be C1 -(3, 4) and C2 -(7, 4).

**Step #2:** Calculating cost.

The dissimilarity of each non-medoid point with the medoids is calculated and tabulated:

| | X | Y | Dissimilarity From C1 | Dissimilarity From C2 |
|---|---|---|---|---|
| 0 | 7 | 6 | 6 | 2 |
| 1 | 2 | 6 | 3 | 7 |
| 2 | 3 | 8 | 4 | 8 |
| 3 | 8 | 5 | 6 | 2 |
| 4 | 7 | 4 | 4 | 0 |
| 5 | 4 | 7 | 4 | 6 |
| 6 | 6 | 2 | 5 | 3 |
| 7 | 7 | 3 | 5 | 1 |
| 8 | 6 | 4 | 3 | 1 |
| 9 | 3 | 4 | 0 | 4 |

Each point is assigned to the cluster of that medoid whose dissimilarity is less.

The points 1, 2, 5 go to cluster C1 and 0, 3, 6, 7, 8 go to cluster C2.

The cost C = (3 + 4 + 4) + (3 + 1 + 1 + 2 + 2)

**C = 20**

**Step #3:** Now randomly select one non-medoid point and recalculate the cost.

Let the randomly selected point be (7, 3). The dissimilarity of each non-medoid point with the medoids – C1 (3, 4) and C2 (7, 3) is calculated and tabulated.

| | X | Y | Dissimilarity From C1 | Dissimilarity From C2 |
|---|---|---|---|---|
| 0 | 7 | 6 | 6 | 3 |
| 1 | 2 | 6 | 3 | 8 |
| 2 | 3 | 8 | 4 | 9 |
| 3 | 8 | 5 | 6 | 3 |
| 4 | 7 | 4 | 4 | 1 |
| 5 | 4 | 7 | 4 | 7 |
| 6 | 6 | 2 | 5 | 2 |
| 7 | 7 | 3 | - | - |
| 8 | 6 | 4 | 3 | 2 |
| 9 | 3 | 4 | - | - |

Each point is assigned to that cluster whose dissimilarity is less. So, the points 1, 2, 5 go to cluster C1 and 0, 3, 6, 7, 8 go to cluster C2.

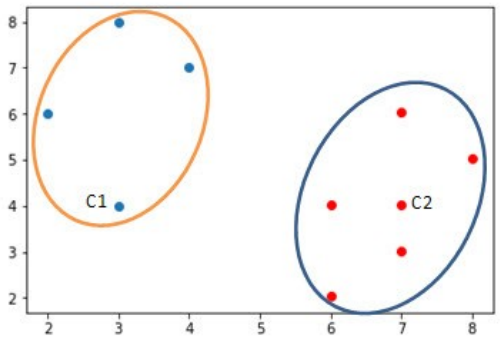
The cost C = (3 + 4 + 4) + (2 + 2 + 1 + 3 + 3)

**C = 22**

Swap Cost = Present Cost – Previous Cost

= 22 – 20 **= 2 >0**

As the swap cost is not less than zero, we undo the swap. Hence (3, 4) and (7, 4) are the final medoids. The clustering would be in the following way




**Advantages:**

1. It is simple to understand and easy to implement.
2. K-Medoid Algorithm is fast and converges in a fixed number of steps.
3. PAM is less sensitive to outliers than other partitioning algorithms.

**Disadvantages:**

1. The main disadvantage of K-Medoid algorithms is that it is not suitable for clustering non-spherical (arbitrary shaped) groups of objects. This is because it relies on minimizing the distances between the non-medoid objects and the medoid (the cluster center) – briefly, it uses compactness as clustering criteria instead of connectivity.
2. It may obtain different results for different runs on the same dataset because the first k medoids are chosen randomly.

**CLARANS**

Clustering Large Applications based upon RANdomized Search. CLARANS is an efficient medoid-based clustering algorithm. The k-medoids algorithm is an adaptation of the k-means algorithm. Rather than calculate the mean of the items in each cluster, a representative item, or medoid, is chosen for each cluster at each iteration. In CLARANS, the process of finding k medoids from n objects is viewed abstractly as searching through a certain graph. In the graph, a node is represented by a set of k objects as selected medoids. Two nodes are neighbors if their sets differ by only one

object. In each iteration, CLARANS considers a set of randomly chosen neighbor nodes as candidate of new medoids. We will move to the neighbor node if the neighbor is a better choice for medoids. Otherwise, a local optima is discovered. The entire process is repeated multiple time to find better.

CLARANS has two parameters: the maximum number of neighbors examined (maxNeighbor) and the number of local minima obtained (numLocal). The higher the value of maxNeighbor, the closer is CLARANS to PAM, and the longer is each search of a local minima. But the quality of such a local minima is higher and fewer local minima needs to be obtained.

To deal with larger data sets, CLARA (Clustering LARge Applications), a sampling-based method, use a random sample from the data set as the candidates of medoids instead of taking the whole set of data into consideration. The algorithm PAM is applied to compute the best medoids from the sample. If the sample is good enough, it should closely represent the original data set. In many cases, a large enough sample by an equal probability of selection design (that is, each object has the same probability to be chosen into the sample) works well. The representative objects (medoids) chosen will likely be similar to those that would have been chosen from the whole data set. CLARA tries multiple random samples and returns the best clustering as the output. The complexity computing the medoids on a random sample is O(ks2 +k(n−k)), where s is the size of the sample, k is the number of clusters, and n is the total number of objects. CLARA can deal with larger data sets than PAM. The effectiveness of CLARA depends on the sample size.

CLARA cannot find a good clustering if any of the best sampled medoids is far from the best k medoids. If an object is one of the best k medoids but is not selected during sampling, CLARA will never find the best clustering. “How might we improve the quality and scalability of CLARA?” Recall that when searching for better medoids, PAM examines every object in the data set against every current medoid, whereas CLARA

confines the candidate medoids to only a random sample of the data set. A randomized algorithm called CLARANS (Clustering Large Applications based upon RANdomized Search) presents a tradeoff between the cost and the effectiveness of using samples to obtain clustering.

**Algorithm:**

k-medoids. PAM, a k-medoids algorithm for partitioning based on medoid or central objects.

**Input:** • k: the number of clusters,

• D: a data set containing n objects.

**Output:** A set of k clusters.

**Method:**

(1) arbitrarily choose k objects in D as the initial representative objects or seeds;

(2) repeat

(3) assign each remaining object to the cluster with the nearest representative object;

(4) randomly select a nonrepresentative object, orandom;

(5) compute the total cost, S, of swapping representative object, oj , with orandom;

(6) if S < 0 then swap oj with orandom to form the new set of k representative objects;

(7) until no change;

**Figure :** PAM, a k-medoids partitioning algorithm.

First, it randomly selects k objects in the data set as the currentmedoids. It then randomly selects a current medoid x and an object y that is not one of the current medoids. Can replacing x by y improve the absolute-error criterion? If yes, the replacement is made. CLARANS conducts such randomized search l times. The set of the current medoids after the l steps is considered as a local optimum. CLARANS repeats the above randomized process m times and returns the best local optimal as the final result.

**Hierarchical Clustering:**

This method creates a hierarchical decomposition of the given set of data objects. We can classify hierarchical methods on the basis of how the hierarchical decomposition is formed. There are two approaches here −

- Agglomerative Approach
- Divisive Approach

**Agglomerative Approach**

This approach is also known as the bottom-up approach. In this, we start with each object forming a separate group. It keeps on merging the objects or groups that are close to one another. It keep on doing so until all of the groups are merged into one or until the termination condition holds.

**Algorithm:**

given a dataset ($d_1$,$d_2$,$d_3$……dn) of size N

#compute the distance matrix

For i=1 to N:

    #  as the distance matrix is symmetric about

    # the primary diagonal so we compute only lowe

    # part of the primary diagonal

For j=1 to i:

    dis_mat[i][j] = distance[$d_i$,$d_j$]

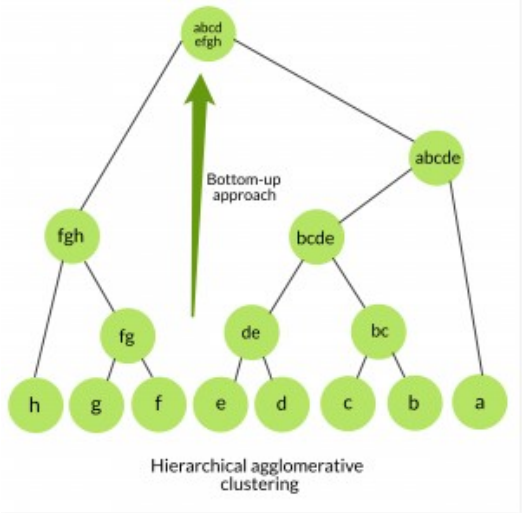
    each data point is a singleton cluster

repeat

    merge the two cluster having minimum distance

    update the distance matrix

until only a single cluster remains

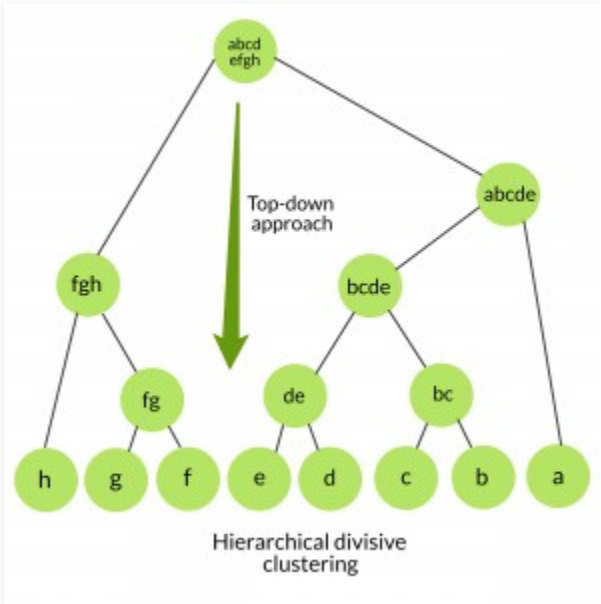



**Divisive Approach**

This approach is also known as the top-down approach. In this, we start with all of the objects in the same cluster. In the continuous iteration, a cluster is split up into smaller clusters. It is down until each object in one cluster or the termination condition holds. This method is rigid, i.e., once a merging or splitting is done, it can never be undone.

**Algorithm**

given a dataset($d_1$,$d_2$,$d_3$……dn) of size N
at the top we have all data in one cluster
the cluster is split using a flat clustering methods eg., K-Means etc
repeat
choose the best cluster among all the clusters to split
split that cluster by the flat clustering algorithm
until each data is in its own singleton cluster




**Approaches to Improve Quality of Hierarchical Clustering**

Here are the two approaches that are used to improve the quality of hierarchical clustering Perform careful analysis of object linkages at each hierarchical partitioning.

- Integrate hierarchical agglomeration by first using a hierarchical agglomerative algorithm to group objects into micro-clusters, and then performing macro-clustering on the micro-clusters.

**Hierarchical Agglomerative vs Divisive clustering**

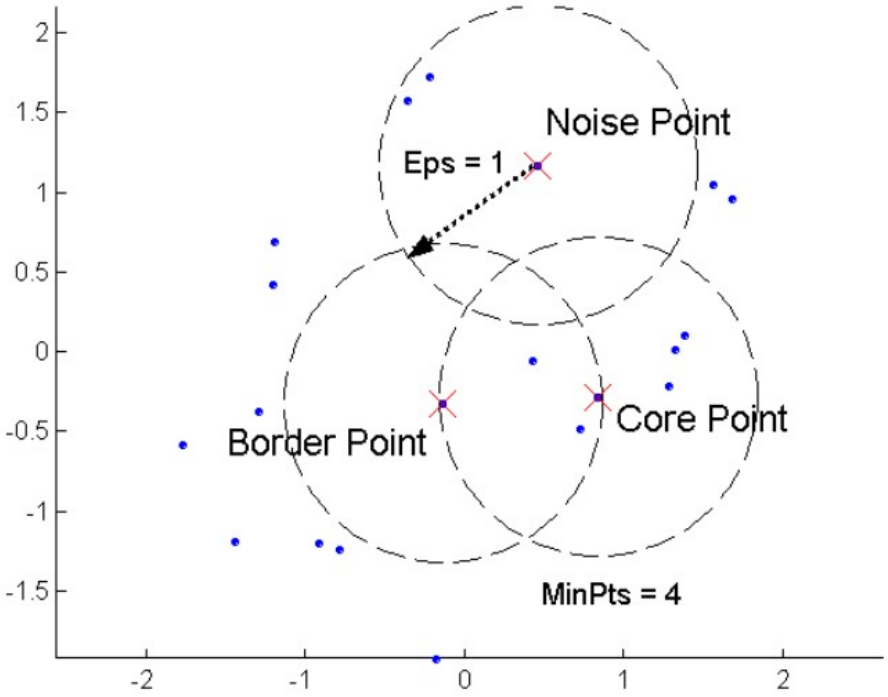
- Divisive clustering is more complex as compared to agglomerative clustering, as in case of divisive clustering we need a flat clustering method as “subroutine” to split each cluster until we have each data having its own singleton cluster.
- Divisive clustering is more efficient if we do not generate a complete hierarchy all the way down to individual data leaves. Time complexity of a naive agglomerative clustering is O(n3) because we exhaustively scan the N x N matrix dist_mat for the lowest distance in each of N-1 iterations. Using priority queue data structure we can reduce this complexity to O(n2logn). By using some more optimizations it can be brought down to O(n2). Whereas for divisive clustering given a fixed number of top levels, using an efficient flat algorithm like K-Means, divisive algorithms are linear in the number of patterns and clusters.

- Divisive algorithm is also more accurate. Agglomerative clustering makes decisions by considering the local patterns or neighbor points without initially taking into account the global distribution of data. These early decisions cannot be undone. whereas divisive clustering takes into consideration the global distribution of data when making top-level partitioning decisions.

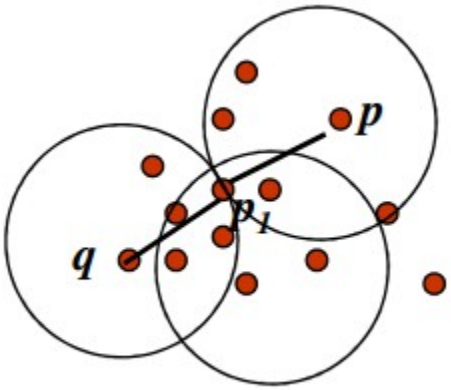
**DB Scan Categorical Clustering Algorithm**

- DBSCAN is a Density-Based Clustering algorithm .
- Reminder: In density based clustering we partition points into dense regions separated by not-so-dense regions.
- DBSCAN: Density at point p: number of points within a circle of radius Eps Dense Region: A circle of radius Eps that contains at least MinPts points
- Characterization of points

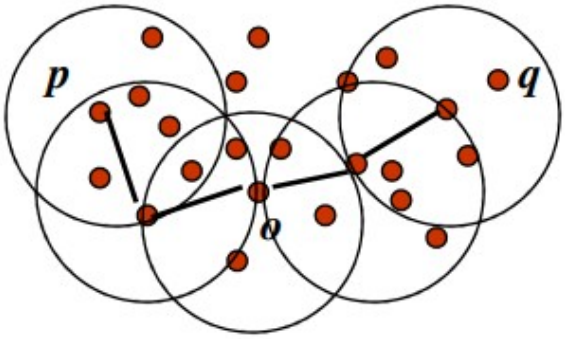
  A point is a core point if it has more than a specified number of points (MinPts) within Eps ▪ These points belong in a dense region and are at the interior of a cluster
  - A border point has fewer than MinPts within Eps, but is in the neighborhood of a core point.
  - A noise point is any point that is not a core point or a border point.




- Density edge We place an edge between two core points q and p if they are within distance Eps.




- Density-connected A point p is density-connected to a point q if there is a path of edges from p to q




**DBSCAN Algorithm**

Label points as core, border and noise.

Eliminate noise points

For every core point p that has not been assigned to a cluster

Create a new cluster with the point p and all the points that are density-connected to p.

Assign border points to the cluster of the closest core point.

**ROCK**

ROCK (RObust Clustering using linKs) clustering algorithm which belongs to the class of agglomerative hierarchical clustering algorithms.

Clustering algorithm for data with categorical and Boolean attributes

– A pair of points is defined to be neighbors if their similarity is greater than some threshold

– Use a hierarchical clustering scheme to cluster the data.

1. Obtain a sample of points from the data set

2. Compute the link value for each set of points, i.e., transform the original similarities (computed by Jaccard coefficient) into similarities that reflect the number of shared neighbors between points

3. Perform an agglomerative hierarchical clustering on the data using the “number of shared neighbors” as similarity measure and maximizing “the shared neighbors” objective function

4. Assign the remaining points to the clusters that have been found

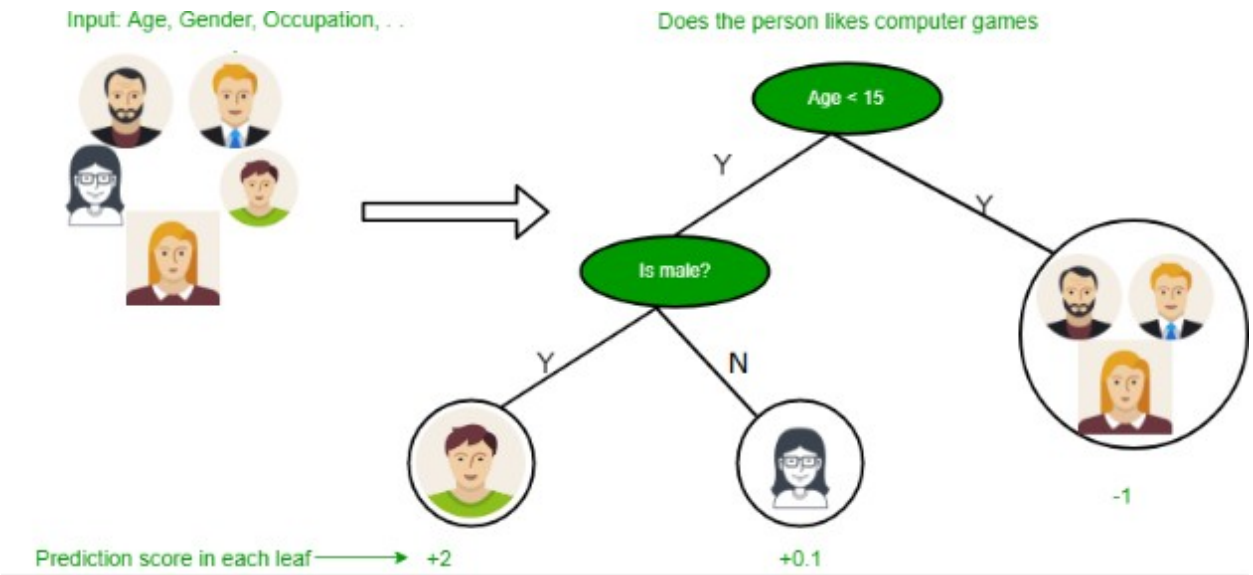
# UNIT – IV

# DECISION TREES

Decision tree is the most powerful and popular tool for classification and prediction. A Decision tree is a flowchart like tree structure, where each internal node denotes a test on an attribute, each branch represents an outcome of the test, and each leaf node (terminal node) holds a class label.

**Introduction**

- Decision tree algorithm falls under the category of supervised learning. They can be used to solve both regression and classification problems.
- Decision tree uses the tree representation to solve the problem in which each leaf node corresponds to a class label and attributes are represented on the internal node of the tree.
- We can represent any boolean function on discrete attributes using the decision tree.




**Decision Tree Construction Principle**

**Best Split**

Greedy strategy

– Split the records based on an attribute test that optimizes certain criterion.

Issues

– Determine how to split the records

• How to specify the attribute test condition?

• How to determine the best split?

– Determine when to stop splitting

Greedy approach:

– Nodes with homogeneous class distribution are preferred

• Need a measure of node impurity:


How to specify the attribute test condition?

● Depends on attribute types

– Nominal

– Ordinal

– Continuous

● Depends on number of ways to split

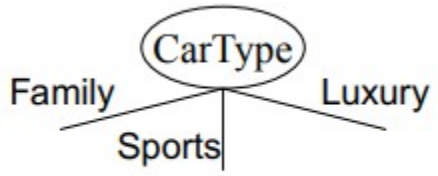
– 2-way split

– Multi-way split

- Splitting Based on Nominal Attributes

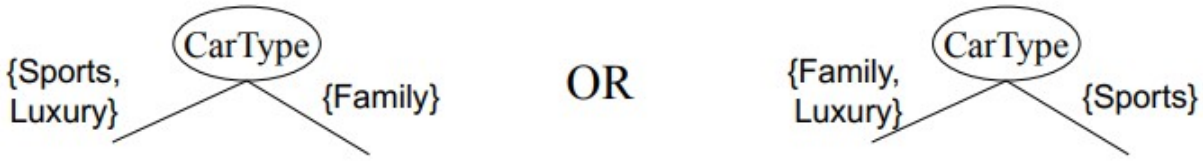
● Multi-way split:

- Use as many partitions as distinct values.




● Binary split:

Divides values into two subsets. Need to find optimal partitioning.




**Splitting Indicies**

- Work out entropy based on distribution of classes.
- Trying splitting on each attribute.
- Work out expected information gain for each attribute.
- Choose best attribute.

**ID3**

The ID3 follows the Occam's razor principle.

• Attempts to create the smallest possible decision tree.

The Process

• Take all unused attributes and calculates their entropies.

• Chooses attribute that has the lowest entropy is minimum or when information gain is maximum

• Makes a node containing that attribute

The Algorithm

• Create a root node for the tree • If all examples are positive, Return the single-node tree Root, with label = +.

• If all examples are negative, Return the single-node tree Root, with label = -.

• If number of predicting attributes is empty, then Return the single node tree Root, with label = most common value of the target attribute in the examples.

• Else – A = The Attribute that best classifies examples.

– Decision Tree attribute for Root = A.

– For each possible value, vi, of A,

• Add a new tree branch below Root, corresponding to the test A = vi.

• Let Examples(vi), be the subset of examples that have the value vi for A

• If Examples(vi) is empty

– Then below this new branch add a leaf node with label = most common target value in the examples

• Else below this new branch add the subtree ID3 (Examples(vi), Target_Attribute, Attributes – {A})

• End

• Return Root E

**C4.5**

C4.5 is a software extension of the basic ID3 algorithm designed by Quinlan to address the following issues not dealt with by ID3:

- Avoiding overfitting the data
- Determining how deeply to grow a decision tree.
- Reduced error pruning.
- Rule post-pruning.
- Handling continuous attributes.
  e.g., temperature
- Choosing an appropriate attribute selection measure.
- Handling training data with missing attribute values.
- Handling attributes with differing costs.
- Improving computational efficiency.

**Rough set theory:**

**Introduction**

The main goal of the rough set analysis is induction of (learning) approximations of concepts. „ Rough sets constitutes a sound basis for KDD. It offers mathematical tools to discover patterns hidden in data. „ It can be used for feature selection, feature extraction, data reduction, decision rule generation, and pattern extraction (templates, association rules) etc. „ identifies partial or total dependencies in data, eliminates redundant data, gives approach to null values, missing data, dynamic data and others.

Rough sets and fuzzy sets are complementary generalizations of classical sets. The approximation spaces of rough set theory are sets with multiple memberships, while fuzzy sets are concerned with partial memberships. The rapid development of these two approaches provides a basis for “soft computing, ” initiated by Lotfi A. Zadeh. Soft Computing includes along with rough sets, at least fuzzy logic, neural networks, probabilistic reasoning, belief networks, machine learning, evolutionary computing, and chaos theory.

**Basic problems in data analysis solved by Rough Set:**

- Rough sets and fuzzy sets are complementary generalizations of classical sets. The approximation spaces of rough set theory are sets with multiple memberships, while fuzzy sets are concerned with partial memberships. The rapid development of these two approaches provides a basis for “soft computing, ” initiated by Lotfi A. Zadeh. Soft Computing includes along with rough sets, at least fuzzy logic, neural networks, probabilistic reasoning, belief networks, machine learning, evolutionary computing, and chaos theory.
- Basic problems in data analysis solved by Rough Set:
- Characterization of a set of objects in terms of attribute values.
- Finding dependency between the attributes.
- Reduction of superfluous attributes.
- Finding the most significant attributes.
- Decision rule generation.

**Goals of Rough Set Theory –**

- The main goal of the rough set analysis is the induction of (learning) approximations of concepts. Rough sets constitute a sound basis for KDD. It offers mathematical tools to discover patterns hidden in data.
- It can be used for feature selection, feature extraction, data reduction, decision rule generation, and pattern extraction (templates, association rules) etc.
- Identifies partial or total dependencies in data, eliminates redundant data, gives approach to null values, missing data, dynamic data and others.

**Definition**

- A subset defined through its lower and upper approximations is called a Rough Set. When the boundary region is a non-empty set that is B(X ) ≠ B(X ) then the set is called a Rough Set

  Rough sets approach shows many advantages. The most important ones are :

  • Synthesis of efficient algorithms for finding hidden patterns in data;

  • Identification of relationships that would not be found using statistical methods;

  •Representation and processing of both qualitative and quantitative parameters and mixing of user-defined and measured data;

  • Reduction of data to a minimal representation (data reduction);

  • Evaluation of the significance of data;

  • Synthesis of classification or decision rules from data;

  • Legibility and straightforward interpretation of synthesized models;

  • Generates sets of decision rules from data;

  • It is easy to understand;

  • Offers straightforward interpretation of obtained results;

## An Example of Indiscernibility

| | Age | LEMS | Walk |
|---|---|---|---|
| x 1 | 16-30 | 50 | yes |
| x2 | 16-30 | 0 | no |
| x3 | 31-45 | 1-25 | no |
| x4 | 31-45 | 1-25 | yes |
| x5 | 46-60 | 26-49 | no |
| x6 | 16-30 | 26-49 | yes |
| x7 | 46-60 | 26-49 | no |

- The non-empty subsets of the condition attributes are {*Age*}, {*LEMS*}, and {*Age, LEMS*}.
- *IND*({*Age*}) = {{*x1,x2,x6*}, {*x3,x4*}, {*x5,x7*}}
- *IND*({*LEMS*}) = {{*x1*}, {*x2*}, {*x3,x4*}, {*x5,x6,x7*}}
- *IND*({*Age,LEMS*}) = {{*x1*}, {*x2*}, {*x3,x4*}, {*x5,x7*}, {*x6*}}.

**Reduct**

- A system T = (U, A, C, D) is independent if all c in C are indispensable. A set of features R $\subseteq$ C is called the reduct of C if T'= (U, A, R, D) is independent and POSR (D) = POSC (D). Furthermore, there is no T $\subset$ R such that POST (D) = POSC (D) A Reduct is a minimal set of features that preserves the indiscernibility relation produced by a partition of C. There could be several subsets of attributes like R. Similar or indiscernible objects may be represented several times on an information table, some of the attributes maybe superfluous or irrelevant, and they could be removed without loss of classification performance

**Types of data reduction :**

There are three types of data reduction techniques: feature reduction, case reduction and value reduction.

- Feature reduction reduces the number of features (columns) in the data set through selection of the most relevant features or combination of two or more features into a single feature.
- Case reduction reduces the number of cases in a data set (rows) which is usually achieved through specialized sampling methods or sampling strategies.
- Value reduction means reducing the number of different values a feature can take through grouping of values into a single category.
- Possible feature reduction techniques are techniques such as principle components, heuristic feature selection with wrapper method and feature selection with decision trees.
- Examples for case reduction techniques are incremental samples, average samples, increasing the sampling period and strategic sampling of key events.
- For value reduction prominent techniques are rounding, using k-means clustering and discretization using entropy minimization. Not many

forecasting methods can be applied in situations of high uncertainty and high volatility of demand.

**Rule Extraction**

The category representations discussed above are all extensional in nature; that is, a category or complex class is simply the sum of all its members. To represent a category is, then, just to be able to list or identify all the objects belonging to that category. However, extensional category representations have very limited practical use, because they provide no insight for deciding whether novel (never-before-seen) objects are members of the category.

A representation of the category based on a set of rules that describe the scope of the category. The choice of such rules is not unique, and therein lies the issue of inductive bias.

# UNIT – V

# WEB MINING

Web mining is the application of data mining techniques to discover patterns from the World Wide Web. As the name proposes, this is information gathered by mining the web. It makes utilization of automated apparatuses to reveal and extricate data from servers and web2 reports, and it permits organizations to get to both organized and unstructured information from browser activities, server logs, website and link structure, page content and different sources.

The goal of Web structure mining is to generate structural summary about the Web site and Web page. Technically, Web content mining mainly focuses on the structure of inner-document, while Web structure mining tries to discover the link structure of the hyperlinks at the inter-document level. Based on the topology of the hyperlinks, Web structure mining will categorize the Web pages and generate the information, such as the similarity and relationship between different Web sites.

Web structure mining can also have another direction -- discovering the structure of Web document itself. This type of structure mining can be used to reveal the structure (schema) of Web pages, this would be good for navigation purpose and make it possible to compare/integrate Web page schemes. This type of structure mining will facilitate introducing database techniques for accessing information in Web pages by providing a reference schema.

**Introduction**

Web mining is an application of data mining techniques to find information patterns from the web data.Web mining helps to improve the power of web search engine by identifying the web pages and classifying the web documents. Web mining is very useful to e-commerce websites and e-services.

**Web Mining**

Web Mining is the process of Data Mining techniques to automatically discover and extract information from Web documents and services. The main purpose of web mining is discovering useful information from the World-Wide Web and its usage patterns.

**Applications of Web Mining:**

1. Web mining helps to improve the power of web search engine by classifying the web documents and identifying the web pages.
2. It is used for Web Searching e.g., Google, Yahoo etc and Vertical Searching e.g., FatLens, Become etc.
3. Web mining is used to predict user behavior.
4. Web mining is very useful of a particular Website and e-service e.g., landing page optimization.

**Web Content Mining**

- Web content mining can be used for mining of useful data, information and knowledge from web page content.
- Web structure mining helps to find useful knowledge or information pattern from the structure of hyperlinks.
- Due to heterogeneity and absence of structure in web data, automated discovery of new knowledge pattern can be challenging to some extent.
- Web content mining performs scanning and mining of the text, images and groups of web pages according to the content of the input (query), by displaying the list in search engines.

**For example:**

If an user wants to search for a particular book, then search engine provides the list of suggestions.

**Web Structure Mining –**

Web structure mining is the application of discovering structure information from the web. The structure of the web graph consists of web pages as nodes, and hyperlinks as edges connecting related pages. Structure mining basically shows the structured summary of a particular website. It identifies relationship between web pages linked by information or direct link connection. To determine the connection between two commercial websites, Web structure mining can be very useful.

**Example:** Web structure mining can be very useful to companies to determine the connection between two commercial websites.

**Web Usage Mining**

- Web usage mining is used for mining the web log records (access information of web pages) and helps to discover the user access patterns of web pages.
- Web server registers a web log entry for every web page.
- Analysis of similarities in web log records can be useful to identify the potential customers for e-commerce companies.

**Some of the techniques to discover and analyze the web usage pattern** are:

**i) Session and visitor analysis**

The analysis of preprocessed data can be performed in session analysis ,which includes the record of visitors, days, sessions etc. This information can be used to analyze the behavior of visitors.

Report is generated after this analysis, which contains the details of frequently visited web pages, common entry and exit.

**ii) OLAP (Online Analytical Processing)**

OLAP performs Multidimensional analysis of complex data.OLAP can be performed on different parts of log related data in a certain interval of time.The OLAP tool can be used to derive the important business intelligence metrics.

**Text Mining**

Text mining, also referred to as text data mining, roughly equivalent to text analytics, is the process of deriving high-quality information from text. High-quality information is typically derived through the devising of patterns and trends through means such as statistical pattern learning.

**For example**,

text categorization, text clustering,concept/entity extraction,sentiment analysis,
document summarization, production of granular taxonomies, entity relation modelling.

**Temporal and Spatial Data Mining:**

**Introduction**

Spatial and temporal aspects form a major portion of the vast amount of data generated by mobile devices, GIS systems, computer vision applications and many other processes. Users working with spatio-Temporal data are interested in the properties of the data which makes the interpretation of data easy and intuitive.

A temporal database is a database with built-in time aspects that is a temporal data model and a temporal version of structured query language. More specifically the temporal aspects usually include valid time and transaction time. These attributes go together to form bi-temporal data.

1. Valid time denotes the time period during which a fact is true with respect to the real world.
2. Transaction time is the time period during which a fact is stored in the database.

**Temporal Association Rule**

A temporal association rule expresses that a set of items tends to appear along with another set of items in the same transactions, in a specific time frame. ... A temporal association rule has three factors associated with it: support, temporal support, both already defined, and confidence, that will be defined next.

**GSP Algorithm**

GSP algorithm (Generalized Sequential Pattern algorithm) is an algorithm used for sequence mining. The algorithms for solving sequence mining problems are mostly based on the apriori (level-wise) algorithm. One way to use the level-wise paradigm is to first discover all the frequent items in a level-wise fashion. It simply means counting the occurrences of all singleton elements in the database. Then, the transactions are filtered by removing the non-frequent items. At the end of this step, each transaction consists of only the frequent elements it originally contained. This modified database becomes an input to the GSP algorithm. This process requires one pass over the whole database.

GSP algorithm makes multiple database passes. In the first pass, all single items (1-sequences) are counted. From the frequent items, a set of candidate 2-sequences are formed, and another pass is made to identify their frequency. The frequent 2-sequences are used to generate the candidate 3-sequences, and this process is repeated until no more frequent sequences are found. There are two main steps in the algorithm.

Candidate Generation. Given the set of frequent (k-1)-frequent sequences Fk-1, the candidates for the next pass are generated by joining F(k-1) with itself. A pruning phase eliminates any sequence, at least one of whose subsequences is not frequent.

Support Counting. Normally, a hash tree–based search is employed for efficient support counting. Finally non-maximal frequent sequences are removed.

**Algorithm**

F1 = the set of frequent 1- sequence
K=2,
Do while Fk-1 !=Null;
Generate candidate sets Ck (set of candidate k-sequences);
For all input sequences s in the database D
Do

Incerment count of all a in Ck if s supports a

End do

Fk = { a€ Ck  such that its frequency exceeds the threshold)

K=k+1

End do

Result = set of all frequent sequences is the union of all Fk's

The above algorithm looks like the Apriori algorithm. One main difference is however the generation of candidate sets. Let us assume that:

$A \rightarrow B$ and $A \rightarrow C$

are two frequent 2-sequences. The items involved in these sequences are (A, B) and (A,C) respectively. The candidate generation in a usual Apriori style would give (A, B, C) as a 3-itemset, but in the present context we get the following 3-sequences as a result of joining the above 2- sequences

$A \rightarrow B \rightarrow C$, $A \rightarrow C \rightarrow B$ and $A \rightarrow BC$

The candidate–generation phase takes this into account. The GSP algorithm discovers frequent sequences, allowing for time constraints such as maximum gap and minimum gap among the sequence elements. Moreover, it supports the notion of a sliding window, i.e., of a time interval within which items are observed as belonging to the same event, even if they originate from different events.

**Spatial Mining Task**

Spatial data mining is the process of discovering interesting, useful, non-trivial patterns from large spatial datasets – E.g. Determining hotspots: unusual location.

• Spatial data mining refers to the extraction of knowledge, spatial relationships, or other interesting patterns not explicitly stored in spatial databases.

• Such mining demands an integration of data mining with spatial database technologies. It can be used for understanding spatial data, discovering spatial relationships and relationships between spatial and non-spatial data, constructing spatial knowledge bases, reorganizing spatial databases, and optimizing spatial queries.

• It is expected to have wide applications in geographic information systems, geo marketing, remote sensing, image database exploration, medical imaging, navigation, traffic control, environmental studies, and many other areas where spatial data are used.

• A crucial challenge to spatial data mining is the exploration of efficient spatial data mining techniques due to the huge amount of spatial data and the complexity of spatial data types and spatial access methods.

• Spatial data mining will further develop spatial statistical analysis methods and extend them for huge amounts of spatial data, with more emphasis on efficiency, scalability, cooperation with database and data warehouse systems, improved user interaction, and the discovery of new types of knowledge.

**Spatial Clustering.**

The first type of spatial analysis we will discuss is cluster analysis. A cluster can be defined as a geographically bounded group of occurrences of sufficient size and concentration that is unlikely to have occurred by chance.

There are many different techniques/algorithms that can be used to find clusters in disease datasets. Clusters can be found using 1) point data, showing every disease case as a point, or by using 2) aerial (polygon) data showing the disease expressed as a population rate.

**Measuring clusters**

Point data clustering is measured by calculating the average distances between points. When this average point distance is less than what can be expected for a random distribution, the point dataset displays clustering.

When using data expressing the rate of disease per area (for example the municipality or district), clustering can be measured by the Moran’s scatterplot.

**What causes disease to cluster in space**

We can distinguish between two different processes that may explain disease clustering in space.

First, there are variations in the external environment, i.e. the so-called exogenous factors. For example, diseases can cluster because people cluster, or respiratory syndromes might cluster in space because the air pollution, which causes these syndromes, is clustered in space.

Second, clustering may also be explained by endogenous factors through interdependence between points or areas themselves. In this case, for example, we mean that diseases may cluster because people may catch this disease from other people who have the disease. This is an intrinsic property of the disease itself, hence this is called an endogenous process.