**Unit 5: Cluster Analysis** **LH 7**

- **Basics and Algorithms**
- **K-means Clustering**
- **Hierarchical Clustering**
- **DBSCAN Clustering**

**Basics and Algorithms**

- What is Clustering?
  - Clustering is the process of making a group of abstract objects into classes of similar objects.
  - It groups the unlabeled dataset.

**Note**

- A cluster of data objects can be treated as one group.
- While doing cluster analysis, we first partition the set of data into groups based on data similarity and then assign the labels to the groups.
- The main advantage of clustering over classification is that, it is adaptable to changes and helps single out useful features that distinguish different groups.

## Application Of Cluster Analysis

- Clustering analysis is broadly used in many applications such as market research, pattern recognition, data analysis, and image processing.
- Clustering can also help marketers discover distinct groups in their customer base. And they can characterize their customer groups based on the purchasing patterns.
- In the field of biology, it can be used to derive plant and animal taxonomies, categorize genes with similar functionalities and gain insight into structures inherent to populations.
- Clustering also helps in identification of areas of similar land use in an earth observation database. It also helps in the identification of groups of houses in a city according to house type, value, and geographic location.
- Clustering also helps in classifying documents on the web for information discovery.
- Clustering is also used in outlier detection applications such as detection of credit card fraud.

**Examples of Clustering Applications**

- Marketing: Help marketers discover distinct groups in their customer bases, and then use this knowledge to develop targeted marketing programs.

- Land use: Identification of areas of similar land use in an earth observation database.

- Insurance: Identifying groups of motor insurance policy holders with a high average claim cost.

- City-planning: Identifying groups of houses according to their house type, value, and geographical location.

- Earth-quake studies: Observed earth quake epicenters should be clustered along continent faults.

- Media: Youtube, Netflix etc recommend the video being based on previous search

Types of clustering algorithm.

- In general, the major clustering methods can be classified into the following categories.

**Partitioning (Centroid Based Method) methods**

- Given a database of n objects or data tuples, a partitioning method constructs k partitions of the data, where each partition represents a cluster and $k \leq n$. Given the number of partitions (k) to construct, a partitioning method creates an initial partitioning. It then uses **an iterative relocation technique** that attempts to improve the partitioning by moving objects from one group to another. The algorithms that use such methods are k-means algorithm etc.

**Hierarchical Methods:**

- A **Hierarchical clustering** method works via grouping data into a tree of clusters. Hierarchical clustering begins by treating every data point as a separate cluster. There are two methods in hierarchical method.
  - **Agglomerative**
  - **Divisive**

**Density Based Algorithm:**

- Density-Based Clustering refers to unsupervised machine learning methods that identify distinctive clusters in the data, based on the idea that a cluster/group in a data space is a contiguous region of high point density. DBSCAN (Density-Based Spatial Clustering of Applications with Noise) is the most well-known density-based clustering algorithm, separated from other clusters by sparse regions.

**Grid-based methods:**

- It uses a multi resolution grid data structure. It divide the object into finite no of cells that form a grid like structure cell.

**Model Based Methods:**

- Model-based clustering is **a statistical approach to data clustering**. The observed (multivariate) data is considered to have been created from a finite combination of component models. Each component model is a probability distribution, generally a parametric multivariate distribution

| Method | General Characteristics |
|---|---|
| Partitioning methods | – Find mutually exclusive clusters of spherical shape<br>– Distance-based<br>– May use mean or medoid (etc.) to represent cluster center<br>– Effective for small- to medium-size data sets |
| Hierarchical methods | – Clustering is a hierarchical decomposition (i.e., multiple levels)<br>– Cannot correct erroneous merges or splits<br>– May incorporate other techniques like microclustering or consider object "linkages" |
| Density-based methods | – Can find arbitrarily shaped clusters<br>– Clusters are dense regions of objects in space that are separated by low-density regions<br>– Cluster density: Each point must have a minimum number of points within its "neighborhood"<br>– May filter out outliers |
| Grid-based methods | – Use a multiresolution grid data structure<br>– Fast processing time (typically independent of the number of data objects, yet dependent on grid size) |

**K-Means Clustering:**

- K-Means Clustering is **an unsupervised learning algorithm** that is used to solve the clustering problems in machine learning. It is an Unsupervised Learning algorithm, which groups the unlabeled dataset into different clusters.
- Here K defines the number of pre-defined clusters that need to be created in the process, as if K=2, there will be two clusters, and for K=3, there will be three clusters, and so on.
- It is a centroid-based algorithm, where each cluster is associated with a centroid.
- The main aim of this algorithm is to minimize the sum of distances between the data point and their corresponding clusters.
- The algorithm takes the unlabeled dataset as input, divides the dataset into k-number of clusters, and repeats the process until it does not find the best clusters.

The k-means algorithm proceeds as follows.

- First, it randomly selects k of the objects, each of which initially represents a cluster mean or center.
- For each of the remaining objects, an object is assigned to the cluster to which it is the most similar, based on the distance between the object and the cluster mean.
- It then computes the new mean for each cluster.
- This process iterates until the criterion function converges

**Algorithm:**

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 from D as the initial cluster centers; 2
2. Repeat
   2.1 Form k clusters by assigning each point to its closest centroid.
   2.2 Recompute the centroid of each cluster.
3. Until Centroids do not change.

Initial Centroids:
A1: (2, 10)
B1: (5, 8)
C1: (1, 2)

New Centroids:
A1: (2, 10)
B1: (6, 6)
C1: (1.5, 3.5)

| Data Points | | | Distance to | | | | | | Cluster | New Cluster |
|---|---|---|---|---|---|---|---|---|---|---|
| | | | 2 | 10 | 5 | 8 | 1 | 2 | | |
| A1 | 2 | 10 | 0.00 | | 3.61 | | 8.06 | | 1 | |
| A2 | 2 | 5 | 5.00 | | 4.24 | | 3.16 | | 3 | |
| A3 | 8 | 4 | 8.49 | | 5.00 | | 7.28 | | 2 | |
| B1 | 5 | 8 | 3.61 | | 0.00 | | 7.21 | | 2 | |
| B2 | 7 | 5 | 7.07 | | 3.61 | | 6.71 | | 2 | |
| B3 | 6 | 4 | 7.21 | | 4.12 | | 5.39 | | 2 | |
| C1 | 1 | 2 | 8.06 | | 7.21 | | 0.00 | | 3 | |
| C2 | 4 | 9 | 2.24 | | 1.41 | | 7.62 | | 2 | |

$$d(p_1, p_2) = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$$

Step 1 :Let us consider Three centroid Point for the given Data points
A1 : (2,10)
B1: (5,8)
C1: (1,2)

Step2 : Find the Distance between every point with consider Centroid

Step3 : Categorize Data Point into cluster on the basis of nearest distance

Current Centroids:
A1: (2, 10)
B1: (6, 6)
C1: (1.5, 3.5)

| Data Points | | | Distance to | | | | | | Cluster | New Cluster |
|---|---|---|---|---|---|---|---|---|---|---|
| | | | 2 | 10 | 6 | 6 | 1.5 | 1.5 | | |
| A1 | 2 | 10 | 0.00 | | 5.66 | | 6.52 | | 1 | 1 |
| A2 | 2 | 5 | 5.00 | | 4.12 | | 1.58 | | 3 | 3 |
| A3 | 8 | 4 | 8.49 | | 2.83 | | 6.52 | | 2 | 2 |
| B1 | 5 | 8 | 3.61 | | 2.24 | | 5.70 | | 2 | 2 |
| B2 | 7 | 5 | 7.07 | | 1.41 | | 5.70 | | 2 | 2 |
| B3 | 6 | 4 | 7.21 | | 2.00 | | 4.53 | | 2 | 2 |
| C1 | 1 | 2 | 8.06 | | 6.40 | | 1.58 | | 3 | 3 |
| C2 | 4 | 9 | 2.24 | | 3.61 | | 6.04 | | 2 | 1 |

Step 4: Calculate new cluster by taking mean of Data point in of respective cluster

Current Centroids:
A1: (3, 9.5)
B1: (6.5, 5.25)
C1: (1.5, 3.5)

New Centroids:
A1: (3.67, 9)
B1: (7, 4.33)
C1: (1.5, 3.5)

| Data Points | | | Distance to | | | | | | Cluster | New Cluster |
|---|---|---|---|---|---|---|---|---|---|---|
| | | | 3 | 9.5 | 6.5 | 5.25 | 1.5 | 3.5 | | |
| A1 | 2 | 10 | 1.12 | | 6.54 | | 6.52 | | 1 | 1 |
| A2 | 2 | 5 | 4.61 | | 4.51 | | 1.58 | | 3 | 3 |
| A3 | 8 | 4 | 7.43 | | 1.95 | | 6.52 | | 2 | 2 |
| B1 | 5 | 8 | 2.50 | | 3.13 | | 5.70 | | 2 | 1 |
| B2 | 7 | 5 | 6.02 | | 0.56 | | 5.70 | | 2 | 2 |
| B3 | 6 | 4 | 6.26 | | 1.35 | | 4.53 | | 2 | 2 |
| C1 | 1 | 2 | 7.76 | | 6.39 | | 1.58 | | 3 | 3 |
| C2 | 4 | 9 | 1.12 | | 4.51 | | 6.04 | | 1 | 1 |

$$d(p_1, p_2) = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$$

Current Centroids:
A1: (3.67, 9)
B1: (7, 4.33)
C1: (1.5, 3.5)

| Data Points | | | Distance to 3.67, 9 | Distance to 7, 4.33 | Distance to 1.5, 3.5 | Cluster | New Cluster |
|---|---|---|---|---|---|---|---|
| A1 | 2 | 10 | 1.94 | 7.56 | 6.52 | 1 | 1 |
| A2 | 2 | 5 | 4.33 | 5.04 | 1.58 | 3 | 3 |
| A3 | 8 | 4 | 6.62 | 1.05 | 6.52 | 2 | 2 |
| B1 | 5 | 8 | 1.67 | 4.18 | 5.70 | 1 | 1 |
| B2 | 7 | 5 | 5.21 | 0.67 | 5.70 | 2 | 2 |
| B3 | 6 | 4 | 5.52 | 1.05 | 4.53 | 2 | 2 |
| C1 | 1 | 2 | 7.49 | 6.44 | 1.58 | 3 | 3 |
| C2 | 4 | 9 | 0.33 | 5.55 | 6.04 | 1 | 1 |

$$d(p_1, p_2) = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$$

**Agglomerative Method can be carry in different linkage method.**

- Single linkage, also known as nearest neighbor linkage, determines the distance between two clusters as the shortest distance between any two points in the two clusters. In other words, the distance between two clusters is defined by the distance between their closest points. This method tends to produce long, chain-like clusters that are sensitive to outliers and noise in the data.

- Complete linkage, also known as farthest neighbor linkage, determines the distance between two clusters as the longest distance between any two points in the two clusters. In other words, the distance between two clusters is defined by the distance between their farthest points. This method tends to produce compact, spherical clusters that are less sensitive to outliers and noise in the data.

**Hierarchical Clustering :**

**Agglomerative Method: (Single Linkage)**

Step1 : Find the Distance between Every point with respect to eachother.

| Sample No. | X | Y |
|---|---|---|
| P1 | 0.40 | 0.53 |
| P2 | 0.22 | 0.38 |
| P3 | 0.35 | 0.32 |
| P4 | 0.26 | 0.19 |
| P5 | 0.08 | 0.41 |
| P6 | 0.45 | 0.30 |

$$\begin{pmatrix} & P1 & P2 & P3 & P4 & P5 & P6 \\ P1 & 0 & & & & & \\ P2 & 0.23 & 0 & & & & \\ P3 & 0.22 & 0.14 & 0 & & & \\ P4 & 0.37 & 0.19 & 0.13 & 0 & & \\ P5 & 0.34 & 0.14 & 0.28 & 0.23 & 0 & \\ P6 & 0.24 & 0.24 & 0.10 & 0.22 & 0.39 & 0 \end{pmatrix}$$

**Step 2 : Merging two closet number i.e 0.10 of P3 and P6 .**

| | P1 | P2 | P3 | P4 | P5 | P6 |
|---|---|---|---|---|---|---|
| P1 | 0 | | | | | |
| P2 | 0.23 | 0 | | | | |
| P3 | 0.22 | 0.14 | 0 | | | |
| P4 | 0.37 | 0.19 | 0.13 | 0 | | |
| P5 | 0.34 | 0.14 | 0.28 | 0.23 | 0 | |
| P6 | 0.24 | 0.24 | 0.10 | 0.22 | 0.39 | 0 |

| | P1 | P2 | P3, P6 | P4 | P5 |
|---|---|---|---|---|---|
| P1 | 0 | | | | |
| P2 | 0.23 | 0 | | | |
| P3, P6 | 0.22 | 0.14 | 0 | | |
| P4 | 0.37 | 0.19 | 0.13 | 0 | |
| P5 | 0.34 | 0.14 | 0.28 | 0.23 | 0 |

**Find the distance between {(p3, p6),p1}**
**Dmin {(p1, p3), (p1,p6)}**
**Dmin {0.22, 0.24}**
**0.22**

**Find the distance between {(p3, p6),p2}**
**Dmin {(p2, p3), (p2,p6)}**
**Dmin {0.14, 0.24}**
**0.14**

**Step3 : Repeat Step 1 and Step 2. Until We find cluster of every datapoint.**
**Here P3, P6 and P4 has closet number i.e 0.13**

**Now we will update the Distance Matrix:**

| | P1 | P2 | P3, P6 | P4 | P5 |
|---|---|---|---|---|---|
| P1 | 0 | | | | |
| P2 | 0.23 | 0 | | | |
| P3, P6 | 0.22 | 0.14 | 0 | | |
| P4 | 0.37 | 0.19 | 0.13 | 0 | |
| P5 | 0.34 | 0.14 | 0.28 | 0.23 | 0 |

| | P1 | P2 | P3, P6, P4 | P5 |
|---|---|---|---|---|
| P1 | 0 | | | |
| P2 | 0.23 | 0 | | |
| P3, P6, P4 | 0.22 | 0.14 | 0 | |
| P5 | 0.34 | 0.14 | 0.28 | 0 |

**{(P3, P6), P4}**

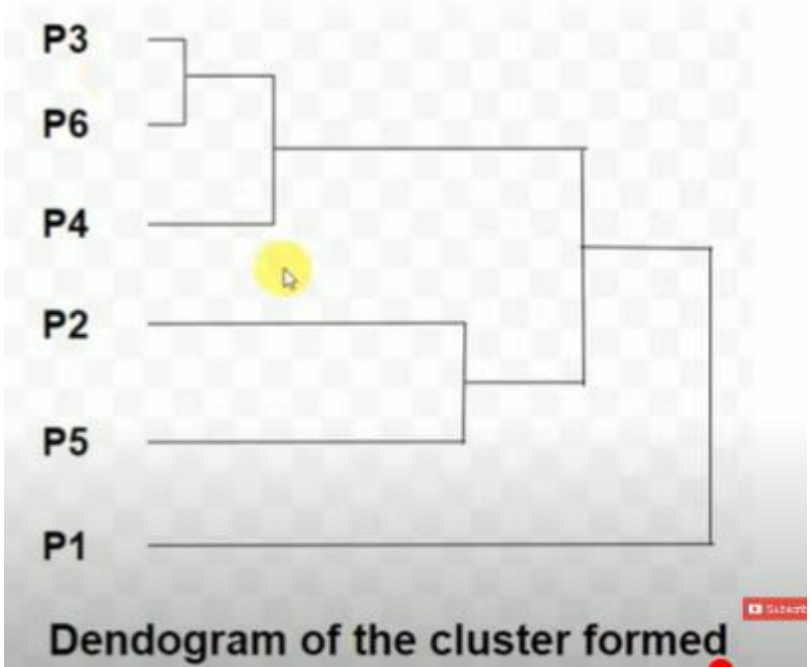
**Step3: Since {p2, (p3,p6,p4)} distance and {p2, p5} distance seems to be same in two digit point. So in this case go beyond two digit. To see the difference.**

$$\begin{pmatrix} & P1 & P2, P5 & P3, P6, P4 \\ P1 & 0 & & \\ P2, P5 & 0.23 & 0 & \\ P3, P6, P4 & 0.22 & 0.14 & 0 \end{pmatrix}$$

$$\begin{pmatrix} & P1 & P2, P5, P3, P6, P4 \\ P1 & 0 & \\ P2, P5, P3, P6, P4 & 0.22 & 0 \end{pmatrix}$$

[{(P3, P6), P4}, (P2, P5)]

[{(P3, P6), P4}, (P2, P5)] , P1



**Hierarchical Clustering :**
**Agglomerative Method: (Complete Linkage)**

- https://www.youtube.com/watch?v=d1qAwe8hthM&t=167s

$$Euclidean\ distance = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$$

$$Euclidean\ distance = \sqrt{(x_2 - x_1)^2}$$

- In order to use the agglomerative algorithm,
- we need to calculate the distance matrix.
- One-dimensional data set {1, 5, 8, 10, 2}

| | 1 | 5 | 8 | 10 | 2 |
|---|---|---|---|---|---|
| 1 | 0 | 4 | 7 | 9 | 1 |
| 5 | 4 | 0 | 3 | 5 | 3 |
| 8 | 7 | 3 | 0 | 2 | 6 |
| 10 | 9 | 5 | 2 | 0 | 8 |
| 2 | 1 | 3 | 6 | 8 | 0 |

**Step1 : Find the Distance between every point**

- Replace Data with column and Row Number:

| | 1 | 5 | 8 | 10 | 2 |
|---|---|---|---|---|---|
| 1 | 0 | 4 | 7 | 9 | 1 |
| 5 | 4 | 0 | 3 | 5 | 3 |
| 8 | 7 | 3 | 0 | 2 | 6 |
| 10 | 9 | 5 | 2 | 0 | 8 |
| 2 | 1 | 3 | 6 | 8 | 0 |

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

**Step2 : Group point (number) which have minimum distance i.e between 1 and 5**

**Step3 : d(2, {1,5}) = max {d(2,1) and d(2,5)} = max {4,3} = 4**
**Follow same method for all point: i.e d(3, {1,5}) , d(4, {1,5})**

| | 1,5 | 2 | 3 | 4 |
|---|---|---|---|---|
| 1,5 | 0 | 4 | 7 | 9 |
| 2 | 4 | 0 | 3 | 5 |
| 3 | 7 | 3 | 0 | 2 |
| 4 | 9 | 5 | 2 | 0 |

**From This matrix we can see that the distance between 3,4 is smallest i.e 2.**

**Hence They Merge Together i.e {3,4}**

D({1,5}, {3,4}) = max {d({1,5}, 3), d({1,5}, 4}) = max (7,9) = 9

D(2, {3,4}) = max {d(2, 3), d(2, 4}) = max (3,5) = 5

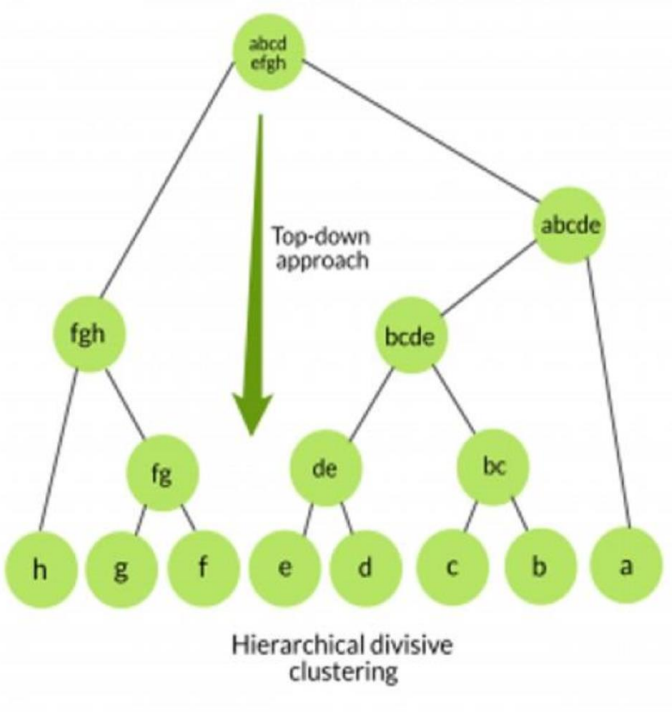
- Following the same procedure, we merge point 2 with the cluster {1, 5} to form {1, 2, 5} and update the distance matrix as follows:

| | [1,5],2 | [3,4] |
|---|---|---|
| [1,5],2 | 0 | 9 |
| [3,4] | 9 | 0 |

| | 1,5 | 2 | 3,4 |
|---|---|---|---|
| 1,5 | 0 | 4 | 9 |
| 2 | 4 | 0 | 5 |
| 3,4 | 9 | 5 | 0 |

# ii) Divisive Hierarchical Clustering

- Unsupervised machine learning algorithm
- Also known as a top-down approach.
- Divisive hierarchical clustering is exactly the opposite of Agglomerative Hierarchical clustering.
- In Divisive Hierarchical clustering, all the data points are considered an individual cluster, and in every iteration, the data points that are not similar are separated from the cluster.
- The separated data points are treated as an individual cluster.
- Finally, we are left with N clusters.
- This hierarchy of clusters is represented in the form of the dendrogram.



Hierarchical divisive clustering

Reference: https://www.javatpoint.com/hierarchical-clustering-in-machine-learning

- In this data objects are grouped in a top down manner.
- Initially all objects are in one cluster.
- Then the cluster is subdivided into smaller and smaller pieces, until each object forms a cluster on its own or until it satisfies certain termination conditions as the desired number of clusters is obtained.

# Divisive Algorithm: Simple approach based on the MST

1. Compute a minimum spanning tree (MST) for the given adjacency matrix.
2. Repeat
   - Create a new cluster by breaking the link corresponding to the largest distance.
3. Until only single cluster remains.

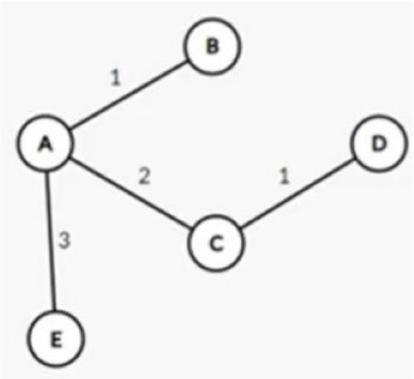
# Solved Example 1:

- Consider the following matrix of distance between five points A, B, C, D and E. Apply divisive hierarchical clustering to build hierarchical clustering dendrogram.

| | A | B | C | D | E |
|---|---|---|---|---|---|
| A | 0 | 1 | 2 | 2 | 3 |
| B | 1 | 0 | 2 | 4 | 3 |
| C | 2 | 2 | 0 | 1 | 5 |
| D | 2 | 4 | 1 | 0 | 3 |
| E | 3 | 3 | 5 | 3 | 0 |

| | A | B | C | D | E |
|---|---|---|---|---|---|
| A | 0 | 1 | 2 | 2 | 3 |
| B | 1 | 0 | 2 | 4 | 3 |
| C | 2 | 2 | 0 | 1 | 5 |
| D | 2 | 4 | 1 | 0 | 3 |
| E | 3 | 3 | 5 | 3 | 0 |

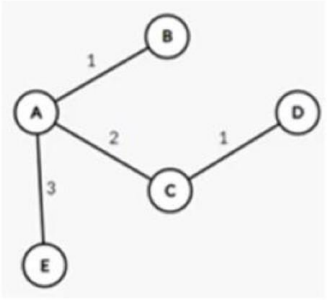
From above adjacency matrix,
create MST by Prim's or Kruskal's algorithm



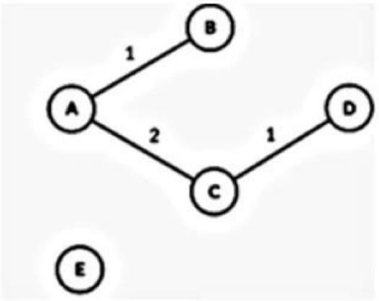

| Edge | Cost |
|---|---|
| A-B | 1 |
| C-D | 1 |
| A-C | 2 |
| A-D | 2 |
| B-C | 2 |
| A-E | 3 |
| B-E | 3 |
| D-E | 3 |
| B-D | 4 |
| C-E | 5 |

| Edge | Cost | |
|---|---|---|
| A-B | 1 | |
| C-D | 1 | |
| A-C | 2 | |
| A-D | 2 | × |
| B-C | 2 | × |
| A-E | 3 | |
| B-E | 3 | × |
| D-E | 3 | × |
| B-D | 4 | × |
| C-E | 5 | × |

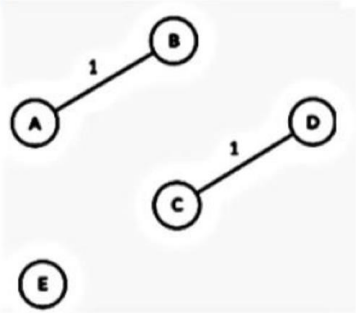
- Here, we take the edges marked with red and they cover all the vertices.
- We omit the edges marked with x because they form loop



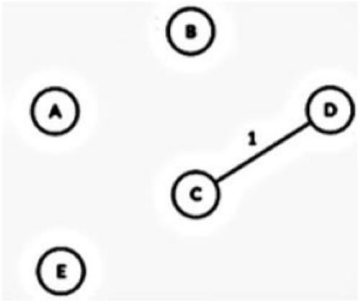

1. Largest edge is between A and E. Cutting this edge results into two clusters {E} and {A,B,C,D}



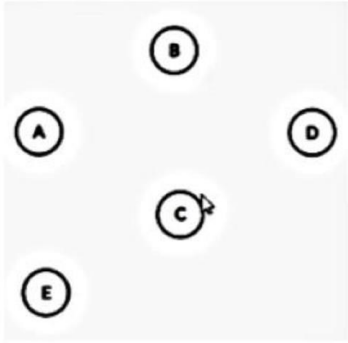

2. Next, remove the edge between A and C.
This split creates three clusters {A,B} , {C,D} and {E}



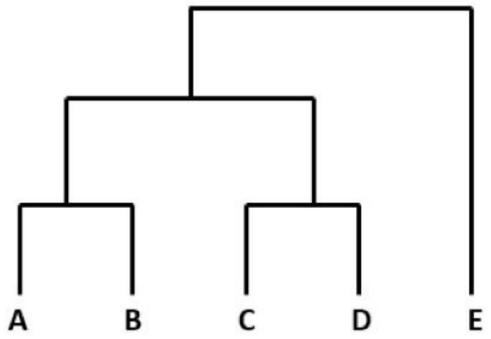

3. Next break A and B
This split creates three clusters {A} , {B} , {C,D} and {E}




4. Next break C and D
This split creates three clusters {A} , {B} , {C}, {D} and {E}




**Required dendrogram is:**



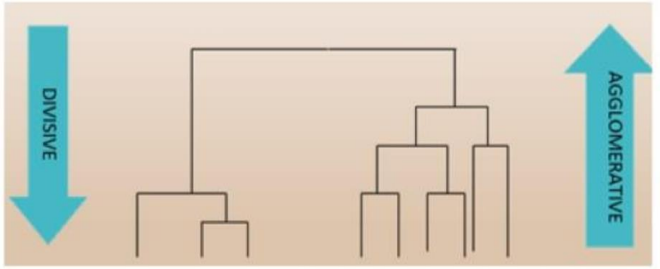
# Reference videos:

- https://www.youtube.com/watch?v=vQEXvV5W7s0
- https://www.youtube.com/watch?v=4GbhWbMJLMY

# Agglomerative Vs Divisive Clustering

### Agglomerative Hierarchical Clustering

- Bottom-up strategy
- Each cluster starts with only one object
- Clusters are merged into larger and larger clusters until:
  - → All the objects are in a single cluster
  - → Certain termination conditions are satisfied




### Divisive Hierarchical Clustering

- Top-down strategy
- Start with all objects in one cluster
- Clusters are subdivided into smaller and smaller clusters until:
  - → Each object forms a cluster on its own
  - → Certain termination conditions are satisfied

# Advantages and Disadvantages of Hierarchical clustering

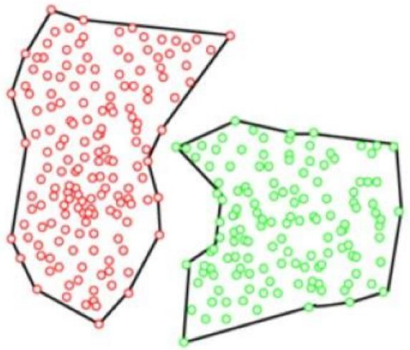
- **Advantages**
  1. No need for information about how many numbers of clusters are required
  2. Easy to use and implement
  3. Dendrogram provides clear visualization.

- **Disadvantages**
  1. We can not take a step back in this algorithm.
  2. Time complexity is higher
  3. Not suitable for larger dataset due to high time and space complexity.

# 5.4 DBSCAN Clustering

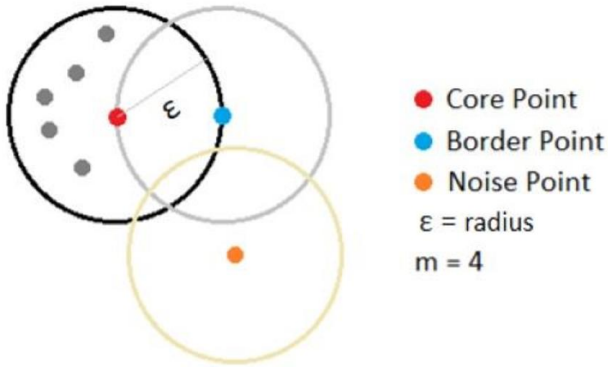
# 5.4 DBSCAN Clustering

- K-Means and Hierarchical Clustering **both fail in creating clusters of arbitrary shapes.**
- They are **not able to form clusters based on varying densities.**
- That's why **we need DBSCAN clustering.**
- DBSCAN, help us identify arbitrary shaped clusters.



# 5.4 DBSCAN Clustering

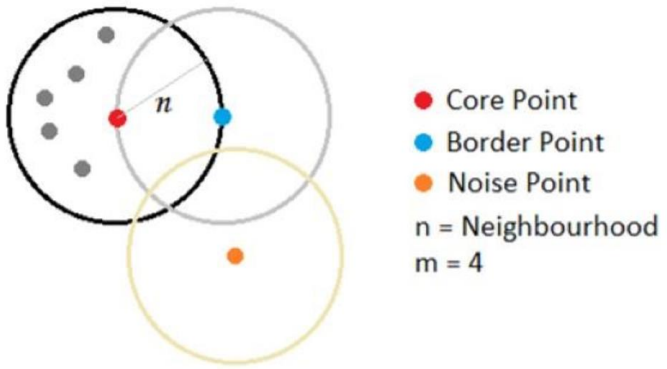
- **DBSCAN** stands for **D**ensity-**B**ased **S**patial **C**lustering of **A**pplications with **N**oise.
- DBSCAN requires only two parameters: ***epsilon*** and ***minPoints*.**
  - ***Epsilon*** is the radius of the circle to be created around each data point to check the density and
  - ***minPoints*** is the minimum number of data points required inside that circle for that data point to be classified as a **Core** point.
- In higher dimensions the circle becomes hypersphere,
  - ***epsilon*** becomes the radius of that hypersphere, and
  - ***minPoints*** is the minimum number of data points required inside that hypersphere.




DBSCAN CLUSTERING

# Core point, Border point and Noise

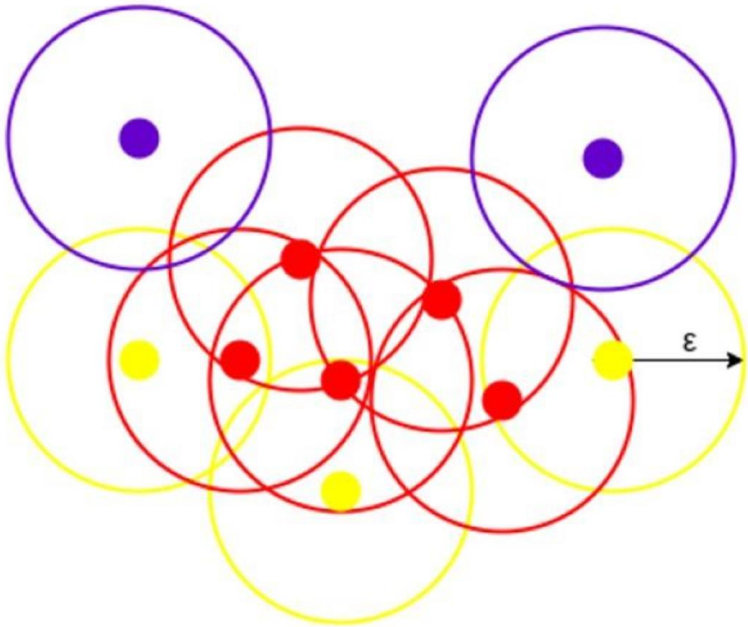
- DBSCAN creates a circle of *epsilon* radius around every data point and classifies them into **Core** point, **Border** point, and **Noise**.
  - **Core Point(x):** Data point that has at least *minPoints (n)* within *epsilon (ε)* distance.
  - **Border Point(y):** Data point that has at least one core point within *epsilon (ε)* distance and lower than *minPoints (n)* within *epsilon (ε)* distance from it.
  - **Noise Point(z):** Data point that has no core points within *epsilon (ε)* distance.




DBSCAN CLUSTERING

# Core point, Border point and Noise

- The above figure shows us a cluster created by DBCAN with *minPoints = 3*. Here, we draw a circle of equal radius *epsilon* around every data point. These two parameters help in creating spatial clusters.
- All the data points with at least 3 points in the circle including itself are considered as **Core** points represented by **red color.**
- All the data points with less than 3 but greater than 1 point in the circle including itself are considered as **Border** points. They have at least a core point within it. They are represented by **yellow** color.
- Finally, data points with no point other than itself present inside the circle are considered as **Noise** represented by the **purple** color.



# Algorithm:

- Step 1: Label Core point and Noise point
  - Select a random starting point, say O
  - Identify neighborhood of this point O using the radius $\epsilon$
  - Count the number of points, say k, in this neighbourhood including point O
  - If k >= Minpts then mark O as a core point
  - Else it will be marked as noise point
  - Select a new unvisited point and repeat the above steps
- Step 2: Check if noise point can become boundary point
  - If noise point is directly density reachable (That is within the boundary of radius $\epsilon$ from the core point), mark it as boundary point and it will form the part of the cluster
  - A point which is neither core point nor boundary point is marked as noise

# Solved Example 1:

- Perform DBSCAN on the given problem with ε = 2 and minpoint = 2

| | x | y |
|---|---|---|
| A1 | 2 | 10 |
| A2 | 2 | 5 |
| A3 | 8 | 4 |
| A4 | 5 | 8 |
| A5 | 7 | 5 |
| A6 | 6 | 4 |
| A7 | 1 | 2 |
| A8 | 4 | 9 |

Reference Video:
https://www.youtube.com/watch?v=3l1vpcRMGcc

## Step 1: Calculation of Euclidian distance

| Euclidean Distance | A1 | A2 | A3 | A4 | A5 | A6 | A7 | A8 |
|---|---|---|---|---|---|---|---|---|
| A1 | 0 ✓ | 5 | 8.49 | 3.61 | 7.07 | 7.21 | 8.06 | 2.24 |
| A2 | 5 | 0 ✓ | 6.08 | 4.24 | 5 | 4.12 | 3.16 | 4.47 |
| A3 | 8.49 | 6.08 | 0 ✓ | 5 | 1.41 ✓ | 2 ✓ | 7.28 | 6.4 |
| A4 | 3.61 | 4.24 | 5 | 0 ✓ | 3.61 | 4.12 | 7.21 | 1.41 ✓ |
| A5 | 7.07 | 5 | 1.41 ✓ | 3.61 | 0 ✓ | 1.41 ✓ | 6.71 | 5 |
| A6 | 7.21 | 4.12 | 2 ✓ | 4.12 | 1.41 ✓ | 0 ✓ | 5.39 | 5.39 |
| A7 | 8.06 | 3.16 | 7.28 | 7.21 | 6.71 | 5.39 | 0 ✓ | 7.62 |
| A8 | 2.24 | 4.47 | 6.4 | 1.41 ✓ | 5 | 5.39 | 7.62 | 0 ✓ |

| | A1 | A2 | A3,A5,A6 | A4,A8 | A3,A5,A6 | A3,A5,A6 | A7 | A4,A8 |
|---|---|---|---|---|---|---|---|---|

## Step 2: Count of points within ε = 2 and identify each points as core, border or noise point w.r.t. Minpts=2

| Points | No of points | Remarks |
|---|---|---|
| A1 | 1 (A1) | Noise |
| A2 | 1 (A2) | Noise |
| **A3** | 3 (A3, A5, A6) | **Core** |
| **A4** | 2 (A4, A8) | **Core** |
| **A5** | 3 (A3, A5, A6) | **Core** |
| **A6** | 3 (A3, A5, A6) | **Core** |
| A7 | 1 (A7) | Noise |
| **A8** | 2 (A4, A8) | **Core** |

**Here:**

**A1, A2 andA7 are Noise (i.e. outlier)**

**Cluster 1: A3, A5 , A6**

**Cluster2: A4, A8**

# Solved Example 2:

- Perform DBSCAN on the given problem with ε = 3.5 and minpoint = 3

| | | |
|---|---|---|
| S1 | 5 | 7 |
| S2 | 8 | 4 |
| S3 | 3 | 3 |
| S4 | 4 | 4 |
| S5 | 3 | 7 |
| S6 | 6 | 7 |
| S7 | 6 | 1 |
| S8 | 5 | 5 |

Reference: https://www.youtube.com/watch?v=jISFQ0I5Gj4

**Step 1: Calculation of Euclidian distance**

| | S1 | S2 | S3 | S4 | S5 | S6 | S7 | S8 |
|---|---|---|---|---|---|---|---|---|
| S1 | 0 | | | | | | | |
| S2 | 4.24 | 0 | | | | | | |
| S3 | 4.47 | 5.1 | 0 | | | | | |
| S4 | 3.16 | 4 | 1.41 | 0 | | | | |
| S5 | 2 | 5.83 | 4 | 3.16 | 0 | | | |
| S6 | 1 | 3.61 | 5 | 3.61 | 3 | 0 | | |
| S7 | 6.08 | 3.61 | 3.61 | 3.61 | 6.71 | 6 | 0 | |
| S8 | 2 | 3.16 | 2.83 | 1.41 | 2.83 | 2.24 | 4.12 | 0 |

**Step 2: Count of points within ε = 3.5 and identify each points as core, border or noise point w.r.t. Minpts=3**

| Points | No of points | Remarks |
|---|---|---|
| **S1** | 5 (S1,S4,S5,S6,S8) | Core |
| S2 | 2 (S2,S8) | Border/Noise |
| **S3** | 3 (S3,S4,S8) | Core |
| **S4** | 5 (S4, S1,S5,S3,S8) | Core |
| **S5** | 5 (S5,S1,S4,S6,S8) | Core |
| **S6** | 4 (S6,S1,S5,S8) | Core |
| S7 | 1 (S7) | Border/Noise |
| S8 | 7 (S8,S1,S2,S3,S4,S5,S6) | Core |

| Points | No of points | Remarks |
|---|---|---|
| **S1** | 5 (S1,S4,S5,S6,S8) | Core |
| S2 | 2 (S2,**S8**) | Border/Noise |
| **S3** | 3 (S3,S4,S8) | Core |
| **S4** | 5 (S4, S1,S5,S3,S8) | Core |
| **S5** | 5 (S5,S1,S4,S6,S8) | Core |
| **S6** | 4 (S6,S1,S5,S8) | Core |
| S7 | 1 (S7) | Border/Noise |
| **S8** | 7 (S8,S1,**S2**,S3,S4,S5,S6) | Core |

**Step 3: Conversion of Noise to Border point**

If density reachable condition is satisfied, convert noise to border point.

Here: S2 is converted to Border point since it has a core point S8 which its neighbor

| Points | No of points | Remarks | |
|---|---|---|---|
| **S1** | 5 (S1,S4,S5,S6,S8) | Core | |
| S2 | 2 (S2, **S8**) | Border/Noise | **Border** |
| **S3** | 3 (S3,S4,S8) | Core | |
| **S4** | 5 (S4, S1,S5,S3,S8) | Core | |
| **S5** | 5 (S5,S1,S4,S6,S8) | Core | |
| **S6** | 4 (S6,S1,S5,S8) | Core | |
| S7 | 1 (S7) | Border/Noise | **Noise** |
| **S8** | 7 (S8,S1,**S2**,S3,S4,S5,S6) | Core | |

**Hence: Clusters are:**
**Cluster 1: {S1,S4,S5,S6,S8}**
**Cluster 2: {S3,S4,S8}**
**Cluster 3: {S1,S3,S4,S5,S8}**
**Cluster 4: {S1,S5,S6,S8}**
**Cluster 5: {S1, S2,S3,S4,S5,S6,S8}**
**Outlier: {S7}**

Other reference videos:

- https://www.youtube.com/watch?v=kG93_zbTzQY
- https://www.youtube.com/watch?v=S5OvKmWIdZA

# Advantages and Disadvantages

- **Advantages:**
  1. Handles irregularly shaped and sized clusters.
  2. Robust to outliers.
  3. Does not require the number of clusters to be specified.
  4. Relatively fast.

- **Disadvantages:**
  1. Difficult to incorporate categorical features.
  2. Struggles with clusters of similar density
  3. Struggles with high dimensional data.

# DBSCAN Vs K-means Clustering

| S. No. | K-means Clustering | DBSCAN |
|---|---|---|
| 1. | • Distance based clustering | • Density based clustering |
| 2. | • Every observation becomes a part of some cluster eventually | • Clearly separates outliers and clusters observations in high density areas |
| 3. | • Build clusters that have a shape of a hypersphere | • Build clusters that have an arbitrary shape or clusters within clusters. |
| 4. | • Sensitive to outliers | • Robust to outliers |
| 5. | • Require no. of clusters as input | • Doesn't require no. of clusters as in |

# Exercise:

# Exercise:

1. Define cluster and clustering.
2. Mention the different approaches of clustering.
3. What is partition clustering method?
4. Write the algorithm for K-means clustering.
5. Mention the major pitfall of K-means clustering compared to DBSCAN?
6. What is hierarchical clustering? Mention its type.
7. What is a dendrogram? What is its importance?
8. Differentiate between Agglomerative and Divisive clustering.
9. Differentiate between DBSCAN and K-means clustering.
10. What are border, core and noise points in DBSCAN mechanism.
11. What is Density reachable and Density connected points?
12. What are the roles of Minpts and epsilon in DBSCAN algorithm?

# Exercise:

13. Divide into three clusters using K-means: D={2, 3, 4, 10, 11, 12, 20, 25, 30}
14. Generate cluster from following dataset using K-means algorithm (take k=2 and consider up to 2 iterations)

T1: Bread, Jelly, Butter

T2: Bread, Butter

T3: Bread, Milk, Butter

T4: Coke, Bread

T5: Coke, Milk

15. Perform clustering using Agglomerative algorithm for the following:

    Points: A(1, 1), B(1.5, 1.5), C(5, 5), D(3, 4), E(4, 4), F(3, 3.5)

16. Find the core, border and noise points using DBSCAN algorithm for below dataset. Consider epsilon= 1.5 and minpts=4

| | | |
|---|---|---|
| S1 | 5 | 7 |
| S2 | 8 | 4 |
| S3 | 3 | 3 |
| S4 | 4 | 4 |
| S5 | 3 | 7 |
| S6 | 6 | 7 |
| S7 | 6 | 1 |
| S8 | 5 | 5 |

# Exam Questions:

# Exam Questions:

1. Differentiate agglomerative and divisive hierarchical clustering. [BIM 2022, Group A]
2. Mention any two limitations of K-means algorithm. [BIM 2021, Group A]
3. What are the limitations of K-means algorithm? [BIM 2018, Group A]
4. If epsilon =2 and Minpts= 2, what are core point, border point and outliner that DBSCAN would find from the data set A(3,10), B(2,3), C(3,4), D(6,7) and F(7,6). [BIM 2022, Group B]
5. What are the roles of Minpts and epsilon in DBSCAN algorithm? Explain. [BIM 2021, Group B]
6. Describe DBSCAN algorithm. [BIM 2018, Group B]