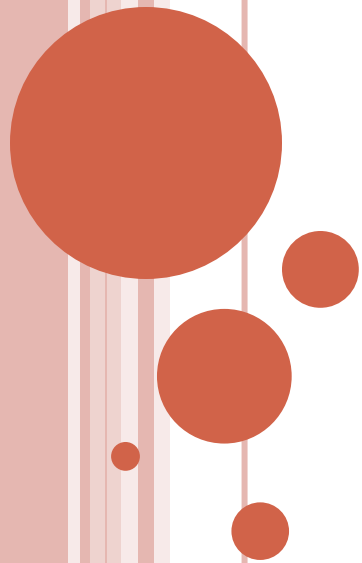


UNSUPERVISED LEARNING: CLUSTERING

By

Dr. Vandana Bhatia



CONTENTS

- Partitioning methods
- Hierarchical clustering
- Fuzzy clustering
- Density-based clustering
- Model-based clustering



CLUSTERING: INTRODUCTION

- ❑ 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.
- ❑ The aim is to segregate groups with similar traits and assign them into clusters
- ❑ Unsupervised Learning-
 - Requires Data, but no labels.
- ❑ Detect Patterns:
 - Group emails or search results
 - Customer shopping patterns
 - Regions of images



CLUSTERING EXAMPLE:

Image segmentation

Goal: Break up the image into meaningful or perceptually similar regions



CLUSTERING: TYPES

- **Partitioning methods:**

Its simply a division of the set of data objects into non-overlapping clusters such that each objects is in exactly one subset.

- **Hierarchical clustering:**

Also known as 'nesting clustering' as it also clusters to exist within bigger clusters to form a tree.

- **Fuzzy clustering:**

It is used to reflect the fact that an object can simultaneously belong to more than one group.

- **Density-based clustering:**

In this clustering model there will be a searching of data space for areas of varied density of data points in the data space .

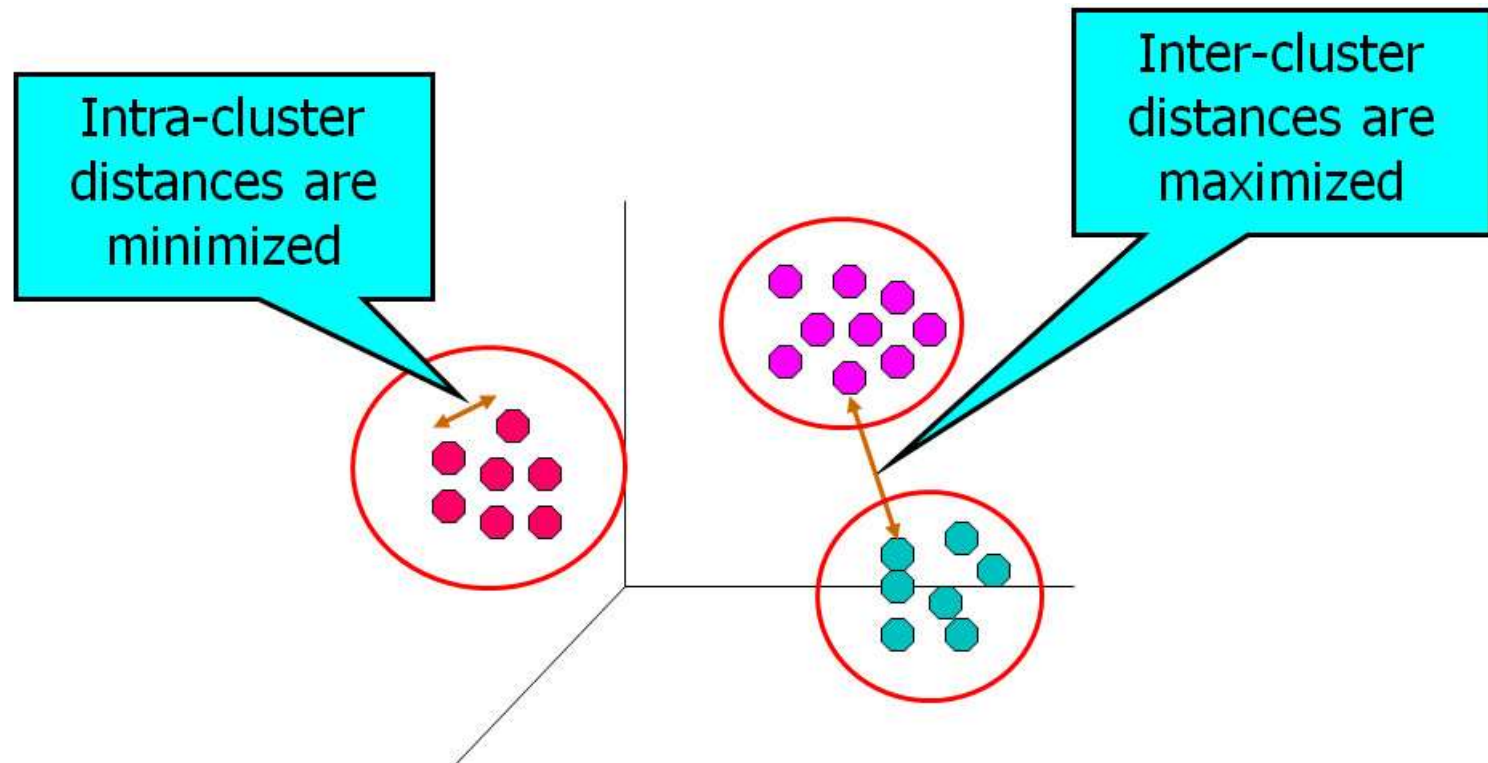
- **Model-based clustering:**

It provides a framework for incorporating our knowledge about a domain.



PARTITION BASED CLUSTERING

- Aim: To minimize Intra-cluster distance



K-MEANS CLUSTERING

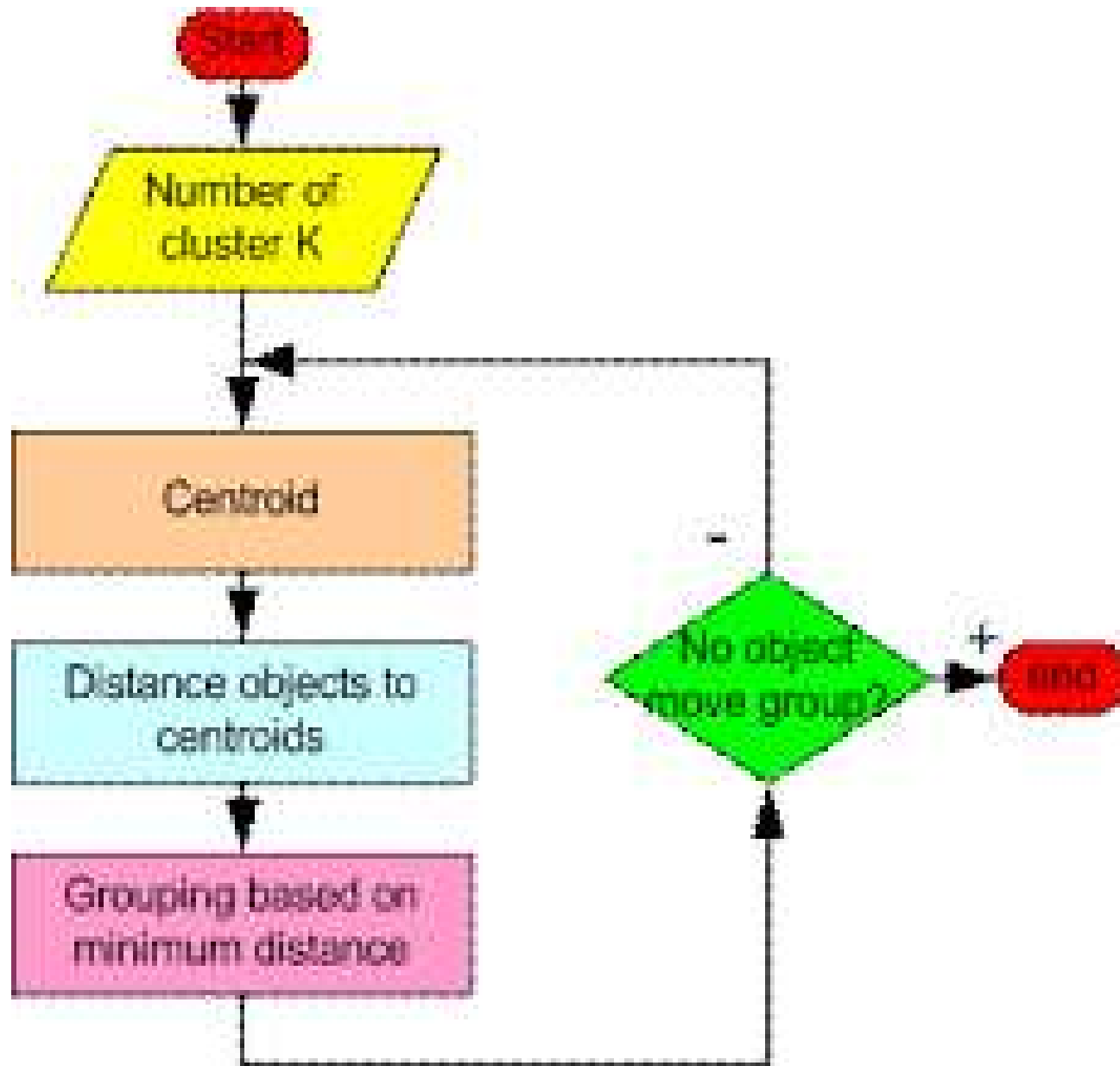
- ❑ An Iterative Clustering Algorithm
- ❑ Partition-based Clustering
- ❑ Each Cluster is associated with a centroid
- ❑ Each point is assigned to the cluster with the closest centroid
- ❑ Number of clusters, K , must be specified.

Algorithm 1 Basic K-means Algorithm.

- 1: Select K points as the initial centroids.
 - 2: **repeat**
 - 3: Form K clusters by assigning all points to the closest centroid.
 - 4: Recompute the centroid of each cluster.
 - 5: **until** The centroids don't change
-



K-MEANS CLUSTERING



K-MEANS CLUSTERING

Details of K-means

1. Initial centroids are often chosen randomly.
 - Clusters produced vary from one run to another
2. The centroid is (typically) the mean of the points in the cluster.
3. 'Closeness' is measured by Euclidean distance, cosine similarity, correlation, etc.
4. K-means will converge for common similarity measures mentioned above.
5. Most of the convergence happens in the first few iterations.
 - Often the stopping condition is changed to 'Until relatively few points change clusters'



K-MEANS CLUSTERING: EXAMPLE

- Given: No. of clusters -2

- Datapoints:

Sample No.	X	Y
1	185	72
2	170	56
3	168	60
4	179	68
5	182	72
6	188	77

- **Solution:** 1. Initial Centroids: K1- (185,72)
K2- (170,56)



K-MEANS CLUSTERING: EXAMPLE

2. Calculate Euclidean Distance of both the centroids with each of the datapoint

$$\text{Distance: } [(x, y), (a, b)] = \sqrt{(x - a)^2 + (y - b)^2}$$

➤ Point (185,72)

- Distance from K1: $(185,72) = \sqrt{(185 - 185)^2 + (72 - 72)^2} = 0$

- Distance from K2: $(170,56) = \sqrt{(170 - 185)^2 + (56 - 72)^2} = 21.93$

❖ *Allocated Cluster: K1*

➤ Point (170,56)

- Distance from K1: $(185,72) = \sqrt{(185 - 170)^2 + (72 - 56)^2} = 21.93$

- Distance from K2: $(170,56) = \sqrt{(170 - 170)^2 + (56 - 56)^2} = 0$

❖ *Allocated Cluster: K2*

➤ Point (168,60)

- Distance from K1: $(185,72) = \sqrt{(185 - 168)^2 + (72 - 60)^2} = 20.808$

- Distance from K2: $(170,56) = \sqrt{(170 - 168)^2 + (56 - 60)^2} = 4.472$

❖ *Allocated Cluster: K2*



K-MEANS CLUSTERING: EXAMPLE (CONT.)

❖ Update Centroid:

Cluster	X	Y
K1	185	72
K2	$(170+168)/2=169$	$(60+56)/2=58$

➤ Point (179,68)

- Distance from K1: $(185,72)=\sqrt{(185-179)^2+(72-68)^2}=7.211$
- Distance from K2: $(169,58)=\sqrt{(179-169)^2+(68-58)^2}=14.142$

❖ *Allocated Cluster: K1*

❖ Update Centroid:

Cluster	X	Y
K1	$(185+179)/2=182$	$(72+68)/2=70$
K2	169	58

➤ Point (182,72)

- Distance from K1: $(182,70)=\sqrt{(182-182)^2+(72-70)^2}=2$
- Distance from K2: $(169,58)=\sqrt{(169-182)^2+(58-72)^2}=19.10$

Allocated Cluster: K1

❖ Update Centroid:

Cluster	X	Y
K1	$(182+182)/2=182$	$(72+70)/2=71$
K2	169	58

K-MEANS CLUSTERING: EXAMPLE (CONT.)

➤ Point (188,77)

- Distance from K1: (182,71) = $\sqrt{(182 - 188)^2 + (77 - 71)^2} = 8.4852$
- Distance from K2: (169,58) = $\sqrt{(188 - 169)^2 + (77 - 58)^2} = 26.87$

❖ *Allocated Cluster: K1*

❖ Update Centroid:

Cluster	X	Y
K1	(182+178)/2=185	(71+77)/2=74
K2	169	58

3. Final Cluster Allocation

Sample No.	X	Y	Cluster
1	185	72	K1
2	170	56	K2
3	168	60	K2
4	179	68	K1
5	182	72	K1
6	188	77	K1




K-MEANS ADVANTAGES AND DISADVANTAGE

Advantages

- 1) Fast, robust and easier to understand
- 2) Relatively efficient: $O(tknd)$, where n is # objects, k is # clusters, d is # dimension of each object, and t is # iterations. Normally, $k, t, d \ll n$.
- 3) Gives best result when data set are distinct or well separated from each other.

Disadvantages

- 1) The learning algorithm requires apriori specification of the number of cluster centers.
 - 2) The use of Exclusive Assignment - If there are two highly overlapping data then k-means will not be able to resolve that there are two clusters.
 - 3) The learning algorithm is not invariant to non-linear transformations i.e. with different representation of data we get different results (data represented in form of cartesian co-ordinates and polar co-ordinates will give different results).
 - 4) Euclidean distance measures can unequally weight underlying factors.
 - 5) The learning algorithm provides the local optima of the squared error function.
 - 6) Randomly choosing of the cluster center cannot lead us to the fruitful result.
 - 7) Applicable only when mean is defined i.e. fails for categorical data.
 - 8) Unable to handle noisy data and outliers.
 - 9) Algorithm fails for non-linear data set.
- 

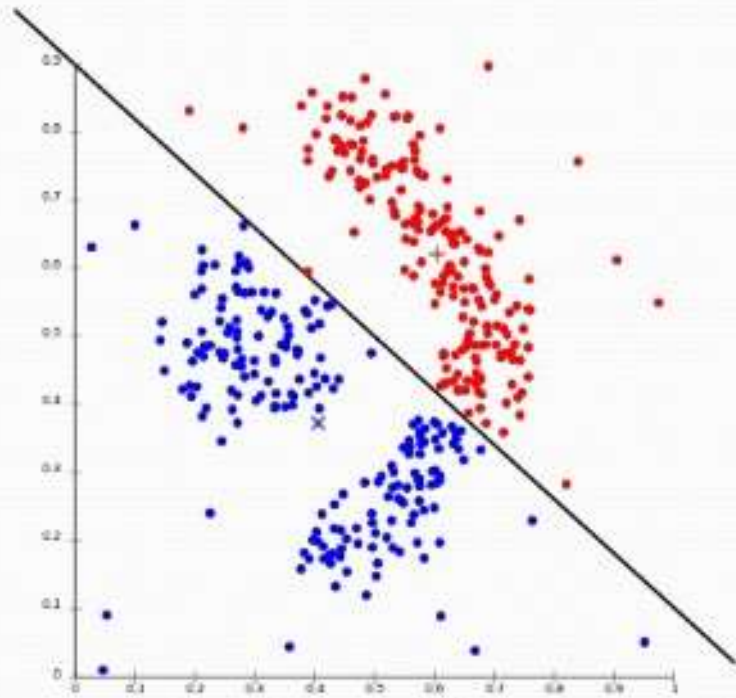
FUZZY CLUSTERING (SOFT CLUSTERING)

- Each data point can belong to more than one **cluster**.
- Fuzzy c-means (FCM) is a method of clustering which allows one piece of data to belong to two or more clusters.
- Works by assigning membership to each data-point corresponding to each cluster on the basis of distance between the cluster center and the data points.
- More the data is near to the cluster center, more is the membership towards that cluster.
- Summation of membership of each data point should be equal to 1.
- After each iteration, membership towards cluster and the cluster centers are updated



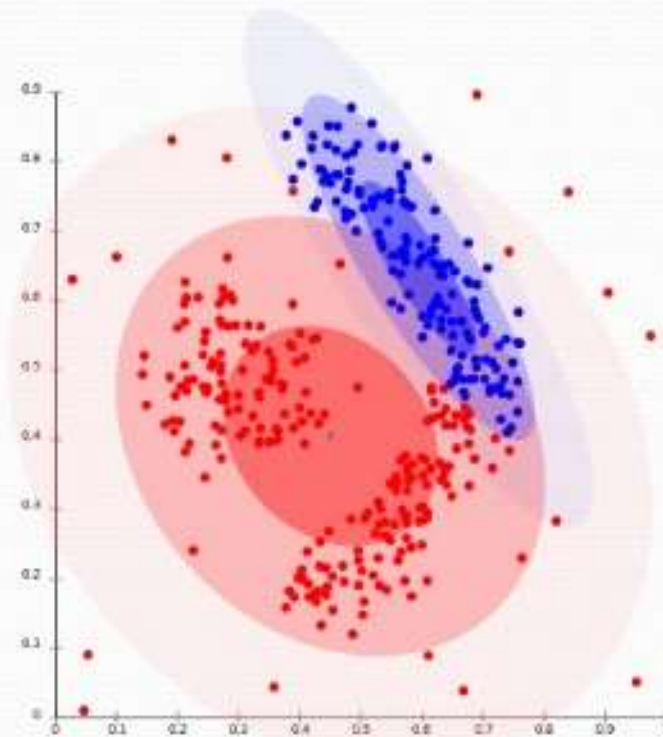
Clustering (Schemas)

- Hard Clustering (ex:k-means)



each data element belongs to exactly one cluster

- Soft Clustering (ex: EM, FCM)



elements can belong to more than one cluster, and associated with each element is a set of membership levels.

FUZZY C-MEAN ALGORITHM

Let $X = \{x_1, x_2, \dots, x_n\}$ be the set of data points and $V = \{v_1, v_2, \dots, v_c\}$ be the set of centers.

1) Randomly select ' c ' cluster centers.

2) Calculate the fuzzy membership ' μ_{ij} ' using:

$$\mu_{ij}^k = \frac{1}{\sum_{k=1}^c \left\{ \frac{x_i - c_j}{x_i - c_k} \right\}^{\frac{2}{m-1}}}$$

3) Compute the fuzzy centers ' v_j ' using:

$$c_j = \frac{\sum_{i=1}^n (\mu_{ij}^k)^m \cdot v_i}{\sum_{i=1}^n (\mu_{ij}^k)^m}$$

4) Repeat step 2) and 3) until the minimum ' J ' value is achieved or $\|U^{(k+1)} - U^{(k)}\| < \beta$.

where,

' k ' is the iteration step.

' β ' is the termination criterion between $[0, 1]$.

' $U = (\mu_{ij})_{n \times c}$ ' is the fuzzy membership matrix.

' J ' is the objective function.

$$J = \sum_{i=1}^N \sum_{j=1}^C (U_{ij})^m \|v_i - c_j\|^2$$


FCM ADVANTAGES AND DISADVANTAGE

➤ Advantages

1. Gives best result for overlapped data set and comparatively better than k-means algorithm.
2. Unlike k-means where data point must exclusively belong to one cluster center here data point is assigned membership to each cluster center as a result of which data point may belong to more than one cluster center.

Disadvantages

- 1) Apriori specification of the number of clusters.
- 2) With lower value of β we get the better result but at the expense of more number of iteration.
- 3) Euclidean distance measures can unequally weight underlying factors.

HIERARCHICAL CLUSTERING

- ❑ It is the hierarchical decomposition of the data based on group similarities.
- ❑ There are two top-level methods for finding these hierarchical clusters:

- **Agglomerative** clustering:

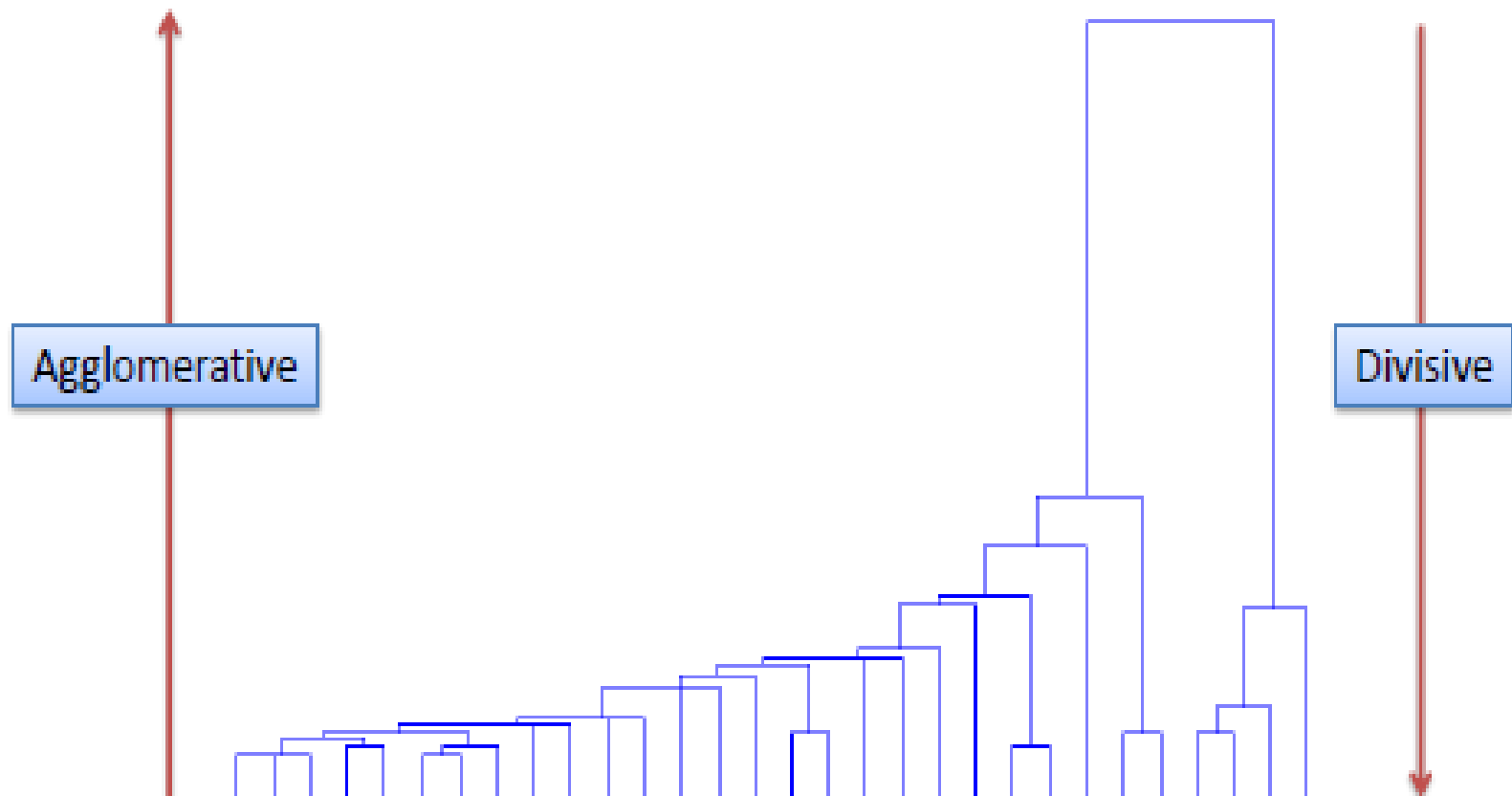
Uses a *bottom-up* approach, wherein each data point starts in its own cluster. These clusters are then joined greedily, by taking the two most similar clusters together and merging them.

- **Divisive** clustering

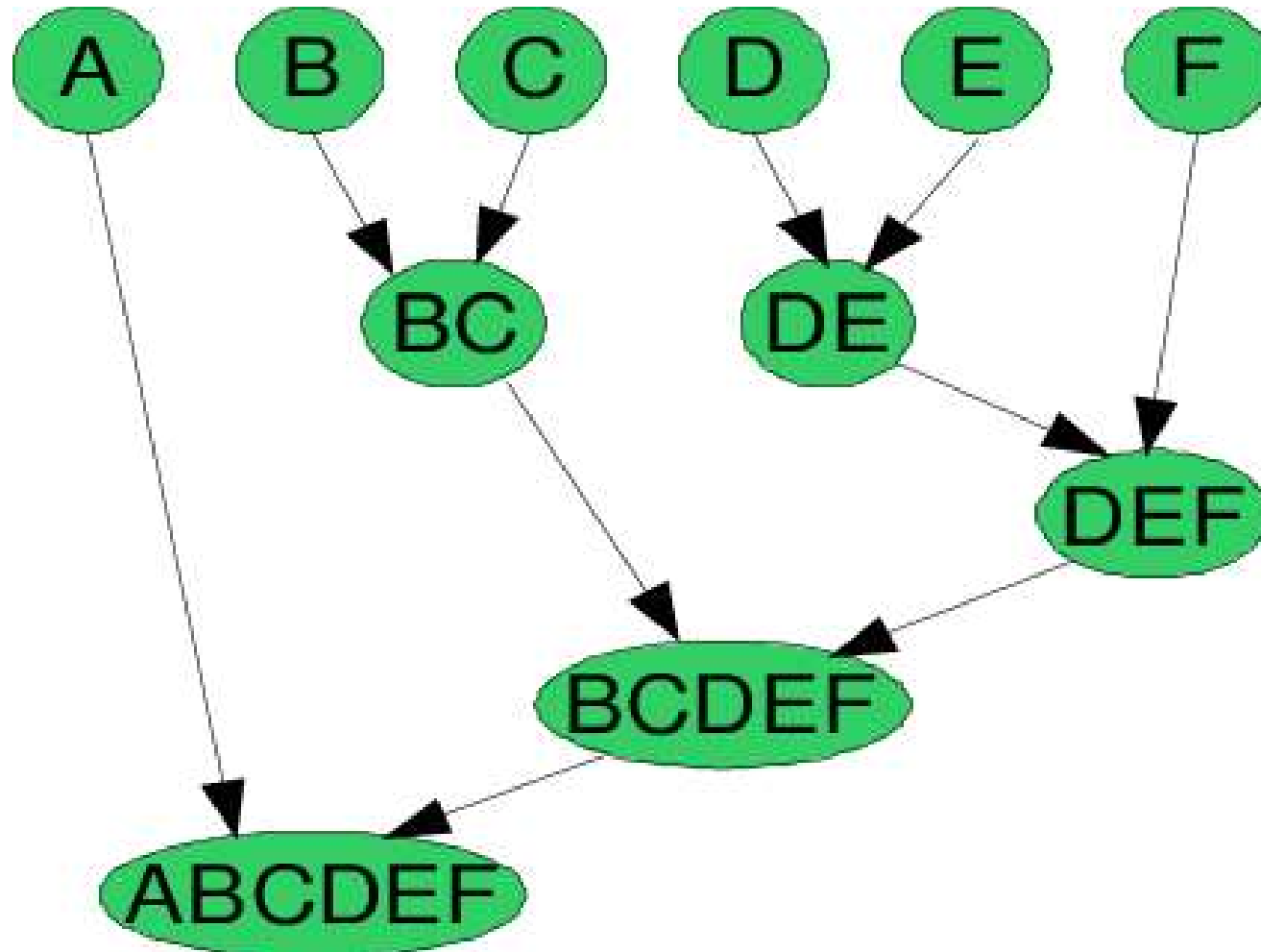
Uses a *top-down* approach, wherein all data points start in the same cluster. You can then use a parametric clustering algorithm like K-Means to divide the cluster into two clusters. For each cluster, you further divide it down to two clusters until you hit the desired number of clusters.

HIERARCHICAL CLUSTERING

Hierarchical Clustering



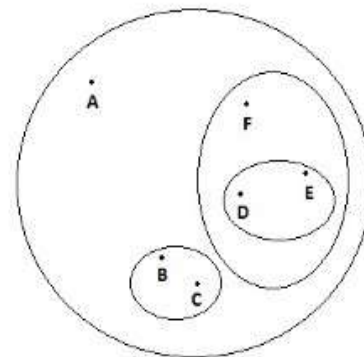
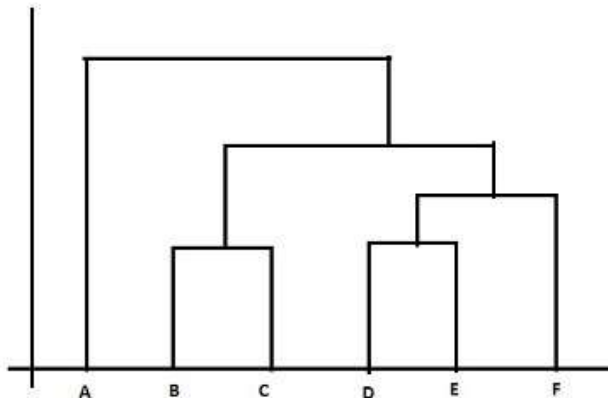
AGGLOMERATIVE HIERARCHICAL CLUSTERING



AGGLOMERATIVE HIERARCHICAL CLUSTERING

- ❑ The Hierarchical clustering technique can be visualized using a **Dendrogram**.
- A **Dendrogram** is a tree like diagram that records the sequences of merges or splits.
- A cluster at level i is the merger of its child cluster at level $i-1$

Dendrogram Representation:



AGGLOMERATIVE HIERARCHICAL CLUSTERING

- Initially each data point is considered as an individual clusters.
- At each iteration, the similar clusters merge with other clusters until one cluster or K clusters are formed.

Basic algorithm:

- 1) Compute the proximity matrix
- 2) Let each data point be a cluster
- 3) Repeat : Merge the two closest clusters and update the proximity matrix
- 4) Until only a single cluster remains



AGGLOMERATIVE HIERARCHICAL CLUSTERING

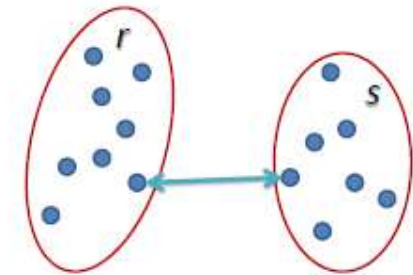
- ❑ Before any clustering is performed, it is required to determine the proximity matrix containing the distance between each point using a distance function.
- ❑ Then, the matrix is updated to display the distance between each cluster.
- ❑ The following three methods differ in how the distance between each cluster is measured.
 - Single Linkage (min is used)
 - Complete Linkage (max is used)
 - Average Linkage



AGGLOMERATIVE HIERARCHICAL CLUSTERING

❑ Single Linkage (min is used)

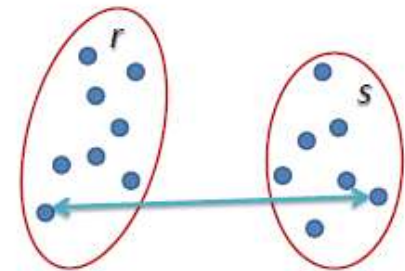
Distance between the closest members of two clusters



$$L(r, s) = \min(D(x_{ri}, x_{sj}))$$

❑ Complete Linkage (max is used)

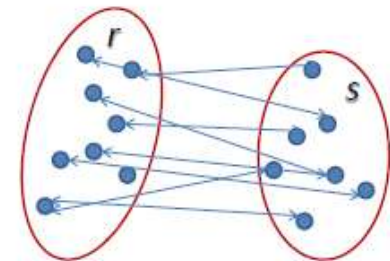
Distance between the members that are farthest apart in two clusters



$$L(r, s) = \max(D(x_{ri}, x_{sj}))$$

❑ Average Linkage

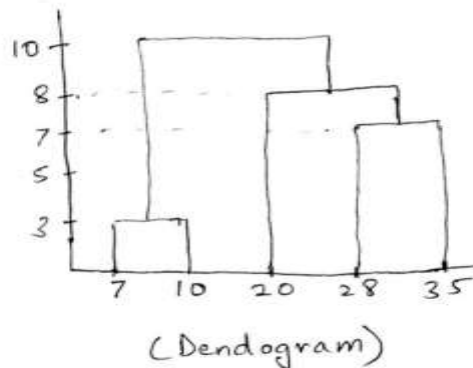
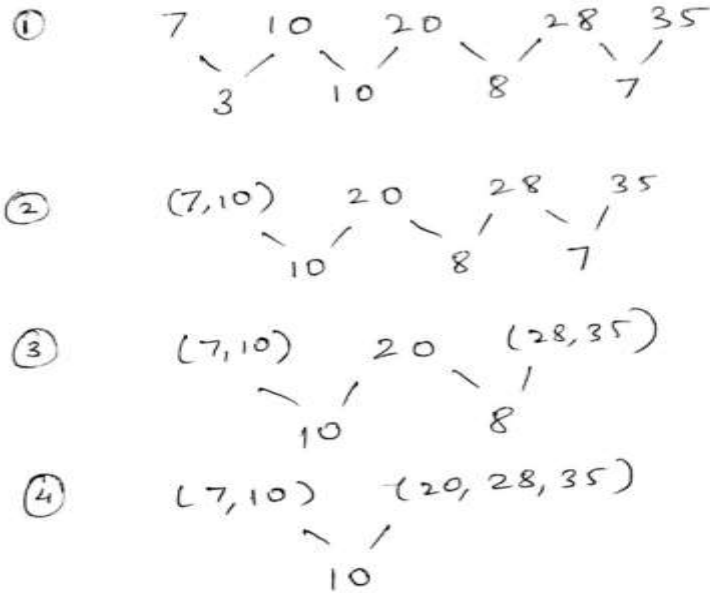
It involves the average distance between each point in one cluster to every point in the other cluster.



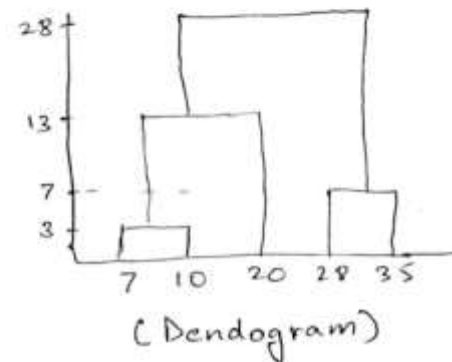
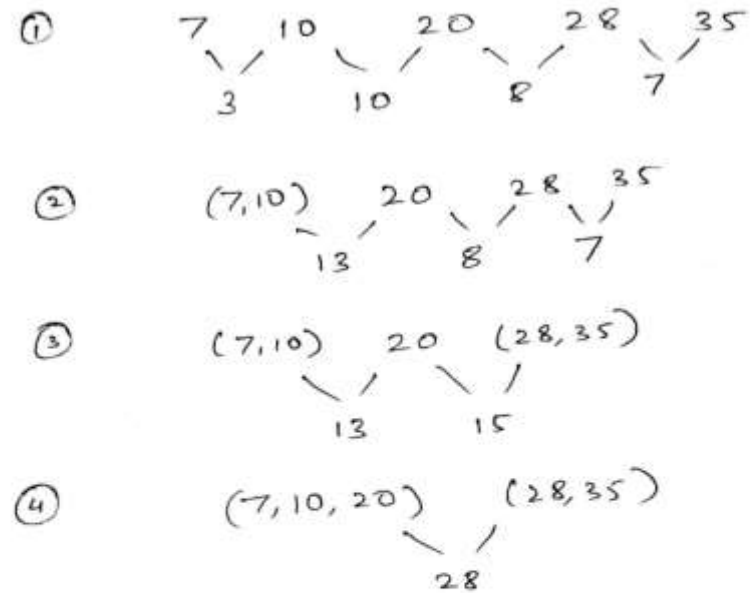
$$L(r, s) = \frac{1}{n_r n_s} \sum_{i=1}^{n_r} \sum_{j=1}^{n_s} D(x_{ri}, x_{sj})$$

AGGLOMERATIVE HIERARCHICAL CLUSTERING

Single Linkage



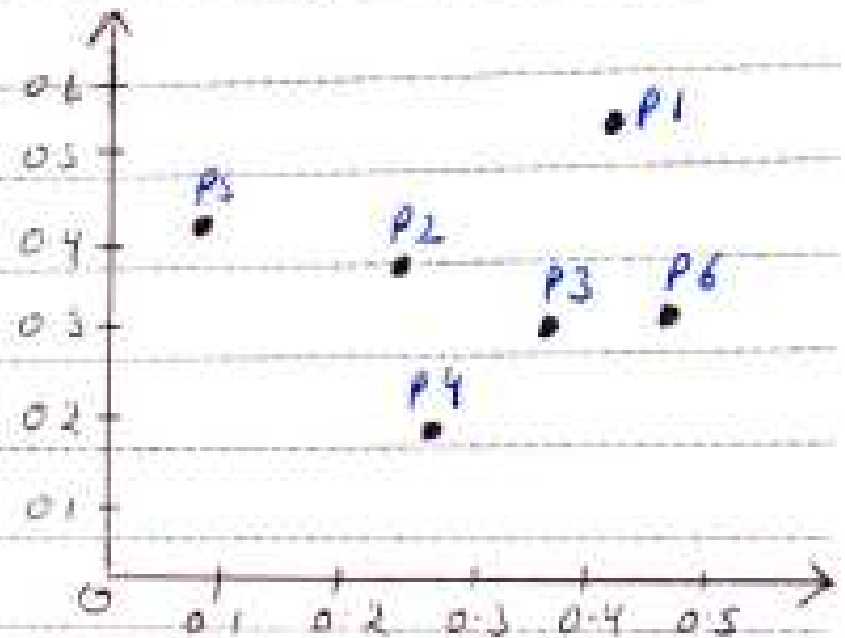
Complete Linkage



AGGLOMERATIVE HIERARCHICAL CLUSTERING: EXAMPLE

Q. Find the clusters using single link techniques. Use Euclidean distance, draw the dendrogram.

	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



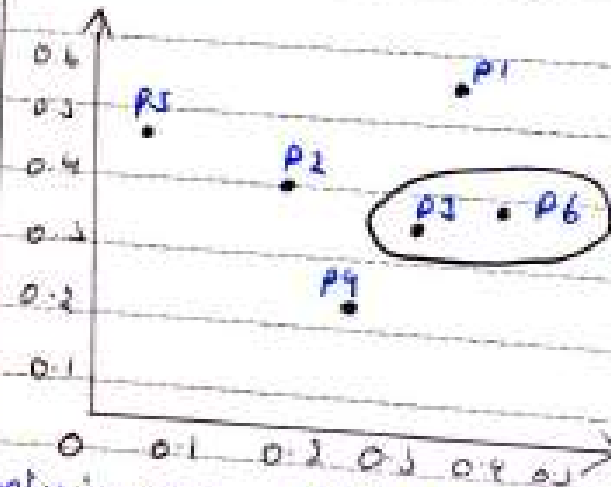
AGGLOMERATIVE HIERARCHICAL CLUSTERING: EXAMPLE

- ① Calculate Euclidian distance, create the distance matrix.
 $\text{Distance}[(x,y), (a,b)] = \sqrt{(x-a)^2 + (y-b)^2}$
 $\text{Distance}(P_1, P_2) = \sqrt{(0.40 - 0.22)^2 + (0.51 - 0.38)^2}$
 $= 0.23$

→ Find the distance for each pair (P_i, P_j)

- ② Form the distance Matrix

	P1	P2	P3	P4	P5	P6
P1	0					
P2	0.23	0				
P3	0.22	0.15	0			
P4	0.37	0.20	0.15	0		
P5	0.34	0.14	0.28	0.29	0	
P6	0.23	0.25	0.11	0.22	0.39	0



→ Find the pair with minimum distance from matrix

AGGLOMERATIVE HIERARCHICAL CLUSTERING: EXAMPLE

③ Update the distance:-



$$\text{Point } P_1 \rightarrow \min [\text{dist}((P_3, P_0), P_1)]$$

$$\rightarrow \min [\text{dist}((P_3, P_1), (P_0, P_1))]$$

$$\rightarrow \min [0.22, 0.23]$$

$$= 0.22$$

$$\text{Point } P_2 \rightarrow \min [\text{dist}((P_3, P_0), P_2)]$$

$$\rightarrow \min [(P_3, P_2), (P_0, P_2)]$$

$$\rightarrow \min [0.15, 0.25]$$

$$= 0.15$$

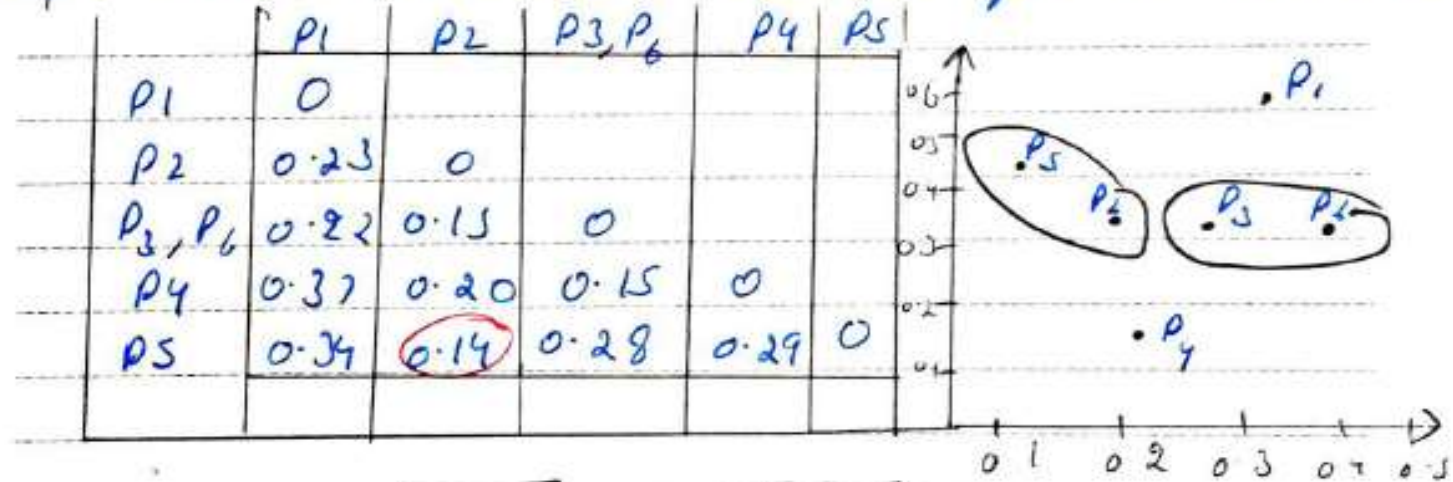
$$\text{Point } P_4 \rightarrow \min [\text{dist}((P_3, P_0), P_4)]$$

$$= \min [0.15, 0.22] = 0.15$$

$$\text{Point } P_5 \rightarrow \min [0.28, 0.39] = 0.28$$

AGGLOMERATIVE HIERARCHICAL

Updated Distance matrix for cluster (P_3, P_6)



Dendrogram \rightarrow $\begin{matrix} \text{---} & \text{---} \\ | & | \\ P_3 & P_6 \end{matrix}$ $\begin{matrix} \text{---} & \text{---} \\ | & | \\ P_2 & P_5 \end{matrix}$

Update the distance Matrix

$$\begin{aligned}
 \text{Point } P_1 &\rightarrow \min[\text{dist}[(P_2, P_5), P_1]] \\
 &\rightarrow \min[\text{dist}(P_2, P_1), (P_5, P_1)] \\
 &\rightarrow \min[0.23, 0.34] \\
 &= 0.23
 \end{aligned}$$

$$\begin{aligned}
 \text{Point } (P_2, P_5) &\rightarrow \min[\text{dist}[(P_2, P_5), (P_3, P_6)]] \\
 &= \min[\text{dist}(P_2, (P_3, P_6)), P_5(P_3, P_6)] \\
 &= \min[0.15, 0.20] = 0.15
 \end{aligned}$$

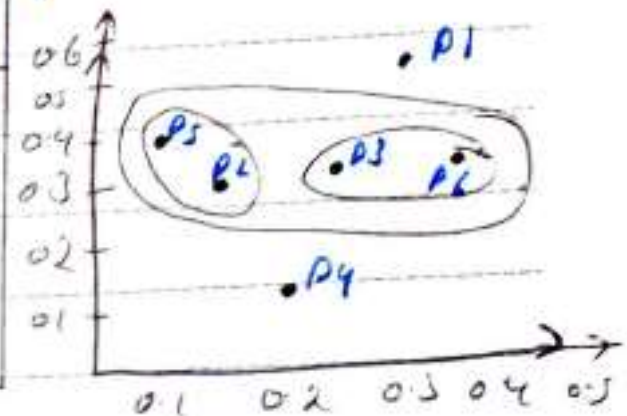
AGGLOMERATIVE HIERARCHICAL CLUSTERING: EXAMPLE

- Point $P_4 \rightarrow \min [\text{dist}(P_2, P_3), P_4]$
 $\Rightarrow \min [\text{dist}(P_2, P_4), (P_3, P_4)]$
 $= \min (0.20, 0.29)$
 $= 0.20$



→ Updated distance matrix for P_2, P_3 is:-

	P_1	P_2, P_3	P_3, P_6	P_4
P_1	0			
P_2, P_3	0.23	0		
P_3, P_6	0.22	0.15	0	
P_4	0.37	0.20	0.15	0



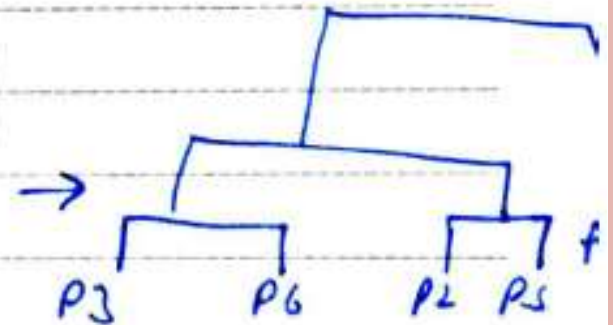
AGGLOMERATIVE HIERARCHICAL CLUSTERING: EXAMPLE

→ #update Distance Matrix:-


• Point $P_1 \rightarrow \min [\text{dist}((P_2, P_3), (P_3, P_6)), P_1]$
 $\rightarrow \min [(P_2, P_3), P_1], (P_3, P_6), P_1]$
 $\rightarrow \min [0.23, 0.22]$
 $= 0.22$

• Point $P_4 \rightarrow \min [\text{dist}((P_2, P_3), (P_3, P_6)), P_4]$
 $\rightarrow \min [(P_2, P_3), P_4], (P_3, P_6), P_4]$
 $= \min [0.20, 0.15]$
 $= 0.15$

	P_1	P_2, P_3, P_3, P_6	P_4
P_1	0		
P_2, P_3, P_3, P_6	0.22	0	
P_4	0.37	0.15	0



AGGLOMERATIVE HIERARCHICAL CLUSTERING: EXAMPLE

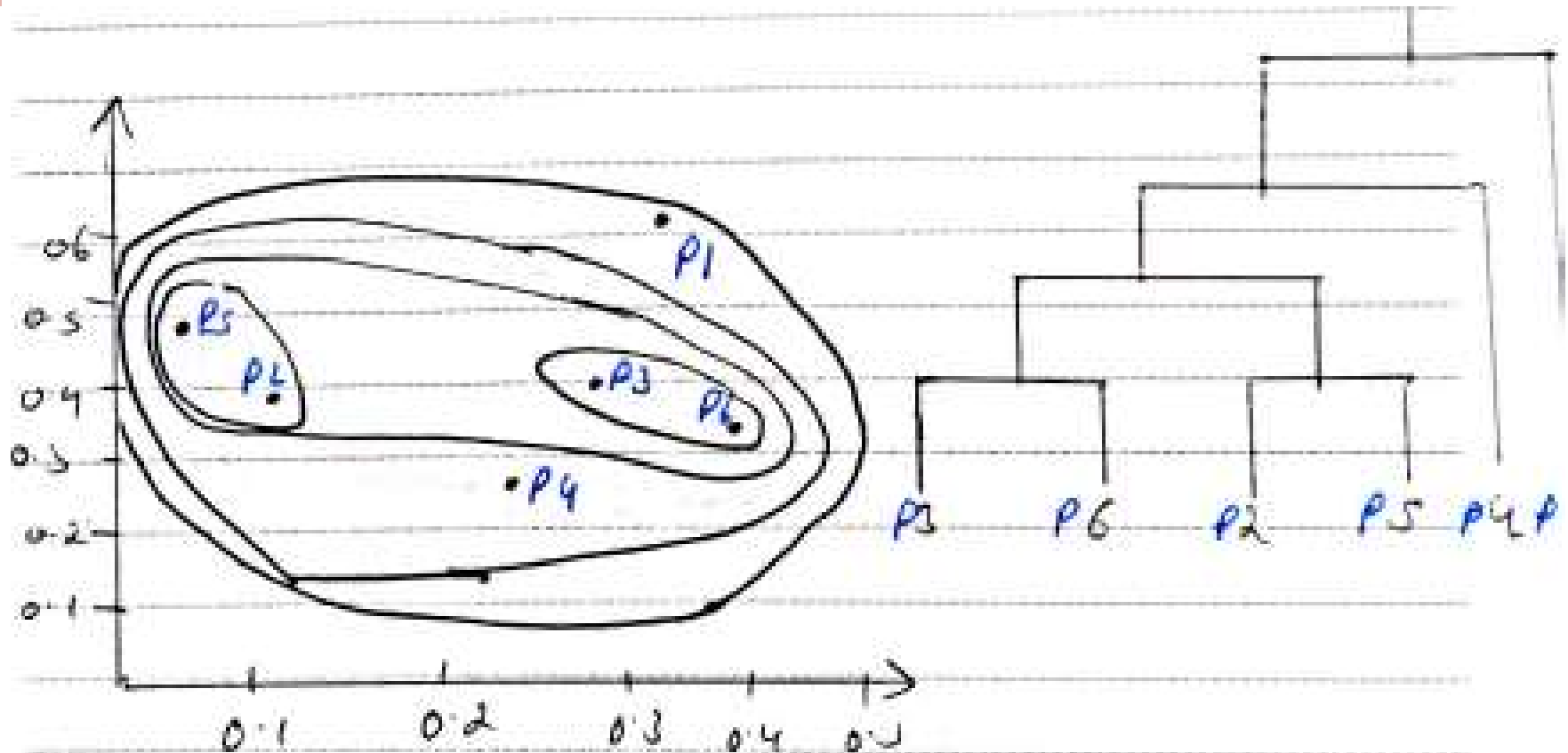
 Update distance matrix

$$\begin{aligned} \text{Point } P_1 &= \text{Min}[\text{dist}((P_2, P_3, P_4, P_5), P_1), (P_1, P_2)] \\ &= \text{Min}[\text{dist}((P_2, P_3, P_4, P_5), P_1), (P_1, P_2)] \\ &= \text{min}[0.22, 0.32] \\ &= 0.22 \end{aligned}$$

Final Matrix

	P_1	P_2, P_3, P_4, P_5
P_1	0	
P_2, P_3, P_4, P_5	0.22	0

AGGLOMERATIVE HIERARCHICAL CLUSTERING: EXAMPLE




HIERARCHICAL CLUSTERING

Advantages

- 1) No apriori information about the number of clusters required.
- 2) Easy to implement and gives best result in some cases.

Disadvantages

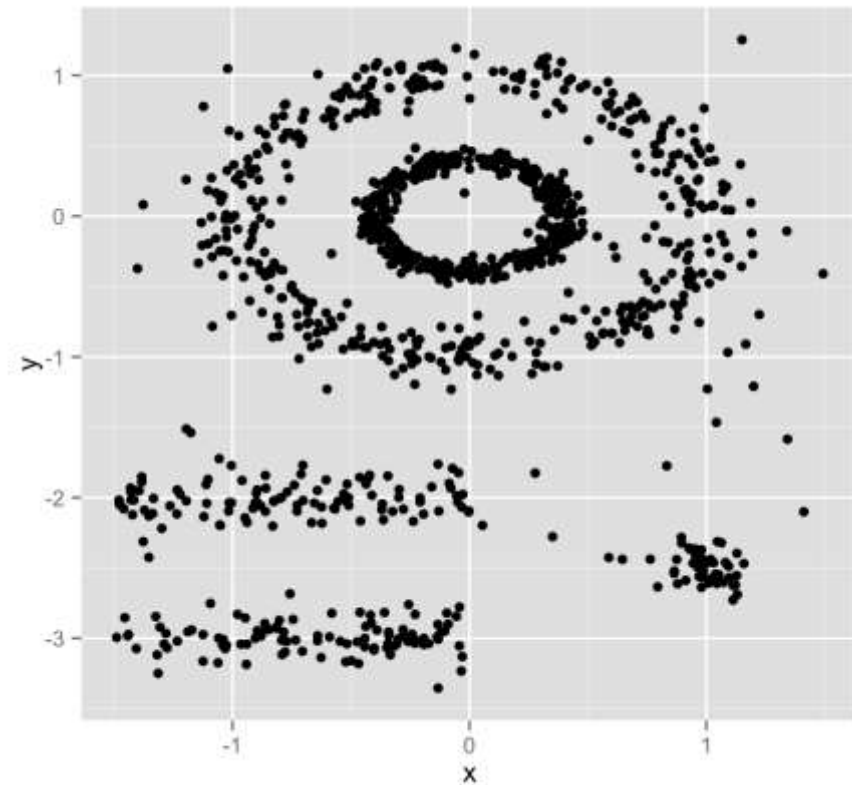
- 1) Algorithm can never undo what was done previously.
 - 2) Time complexity of at least $O(n^2 \log n)$ is required, where ' n ' is the number of data points.
 - 3) Based on the type of distance matrix chosen for merging different algorithms can suffer with one or more of the following:
 - i) Sensitivity to noise and outliers
 - ii) Breaking large clusters
 - iii) Difficulty handling different sized clusters and convex shapes
 - 4) No objective function is directly minimized
 - 5) Sometimes it is difficult to identify the correct number of clusters by the dendrogram.
- 

DENSITY BASED CLUSTERING

- **Partitioning methods (K-means, PAM clustering) and hierarchical clustering** are suitable for finding spherical-shaped **clusters** or convex clusters.
- In other words, they work well for compact and well separated clusters.
- Moreover, they are also severely affected by the presence of noise and outliers in the data.



K-MEANS VS DENSITY BASED CLUSTERING



- ❑ The figure shows a dataset containing nonconvex clusters and outliers/noises.
- ❑ The plot above contains 5 clusters and outliers, including:
 - 2 ovals clusters
 - 2 linear clusters
 - 1 compact cluster
- ❖ Given such data, k-means algorithm has difficulties for identifying these clusters with arbitrary shape.

DENSITY BASED CLUSTERING

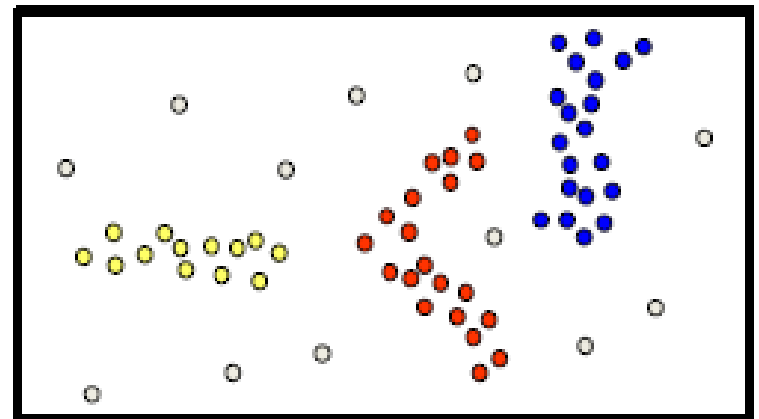
Basic idea

- Clusters are dense regions in the data space, separated by regions of lower object density
- A cluster is defined as a maximal set of density-connected points
- Discovers clusters of arbitrary shape
- ❖ *The goal is to identify dense regions, which can be measured by the number of objects close to a given point.*

Method

- DBSCAN

(Density-Based Spatial Clustering of Applications with Noise)



DBSCAN

❑ Two Parameters:

- **epsilon** (“eps”) : Radius of neighborhood around a point x
- **minimum points** (“MinPts”): Minimum number of neighbors within “eps” radius.

❑ **Direct density reachable:**

A point “A” is directly density reachable from another point “B” if:

- “A” is in the eps-neighborhood of “B” and
- “B” is a core point.

❑ **Density reachable:**

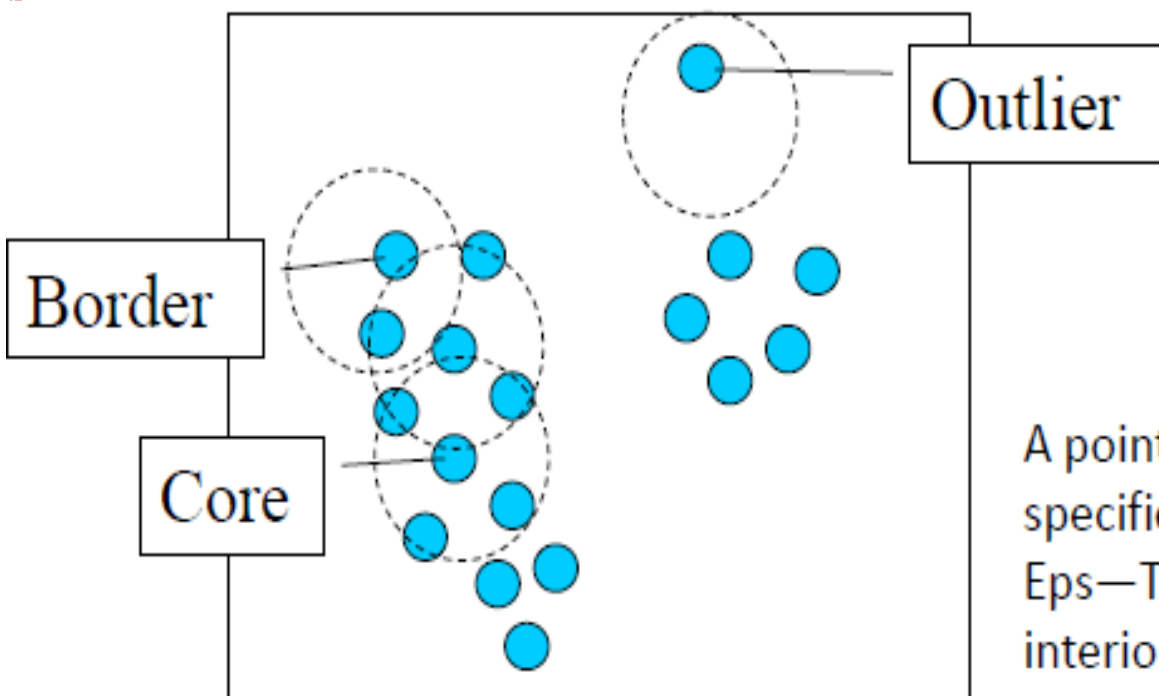
A point “A” is density reachable from “B” if there are a set of core points leading from “B” to “A”.

❑ **Density connected:**

Two points “A” and “B” are density connected if there are a core point “C”, such that both “A” and “B” are density reachable from “C”.



DBSCAN



Given ϵ and *MinPts*, categorize the objects into three exclusive groups.

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

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

A **noise point** is any point that is not a core point nor a border point.

$\epsilon = 1\text{unit}$, $\text{MinPts} = 5$

DBSCAN

1. For each point x_i , compute the distance between x_i and the other points. Finds all neighbor points within distance **eps** of the starting point x_i . Each point, with a neighbor count greater than or equal to **MinPts**, is marked as **core point** or **visited**.
2. For each **core point**, if it's not already assigned to a cluster, create a new cluster. Find recursively all its density connected points and assign them to the same cluster as the core point.
3. Iterate through the remaining unvisited points in the dataset.

Those points that do not belong to any cluster are treated as outliers or noise.



DBSCAN

Advantages

1. Does not require a-priori specification of number of clusters.
2. Able to identify noise data while clustering.
3. DBSCAN algorithm is able to find arbitrarily size and arbitrarily shaped clusters.

Disadvantages

- 1) DBSCAN algorithm fails in case of varying density clusters.
- 2) Fails in case of neck type of dataset.
- 3) Does not work well in case of high dimensional data.



A decorative graphic on the left side of the slide. It consists of several vertical stripes of varying widths and shades of red and pink. Overlaid on these stripes are several circles of different sizes, also in shades of red and pink, arranged in a cluster.

THANKS