Lecture 08 – Unsupervised Learning

Cluster Analysis: Basic Concepts and Methods

- Cluster Analysis: Basic Concepts
  1. Partitioning Methods
  2. Hierarchical Methods
  3. Density-Based Methods
- Summary

What is Cluster Analysis?

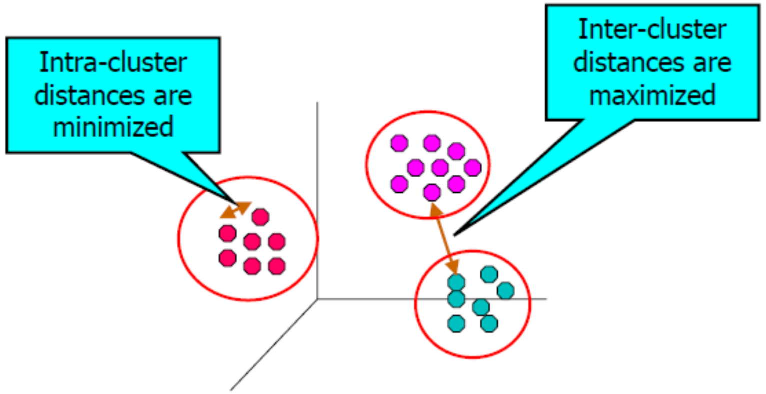
- Cluster: A collection of data objects
  - similar (or related) to one another within the same group
  - dissimilar (or unrelated) to the objects in other groups
- Cluster analysis (or *clustering*, *data segmentation, …*)
  - Finding **similarities between data** according to the characteristics found in the data and grouping similar data objects into clusters
- Unsupervised learning: no predefined classes (i.e., *learning by observations* vs. learning by examples: supervised)
- Typical applications
  - As a stand-alone tool to get insight into data distribution
  - As a preprocessing step for other algorithms

Clustering as a Preprocessing Tool (Utility)

- Summarization:
  - Preprocessing for regression, PCA, classification, and association analysis
- Compression:
  - Image processing
- Finding K-nearest Neighbors
  - Localizing search to one or a small number of clusters
- Outlier detection
  - Outliers are often viewed as those "far away" from any cluster

Quality: What Is Good Clustering?

- A good clustering method will produce high quality clusters
  - high <u>intra-class</u> similarity: ***cohesive*** متماسك within clusters
  - low <u>inter-class</u> similarity: ***distinctive*** تميز between clusters
- The quality of a clustering method depends on:
  - the **similarity measure** used by the method
  - its **implementation**, and
  - Its **ability to discover** some or all of the <u>hidden</u> patterns



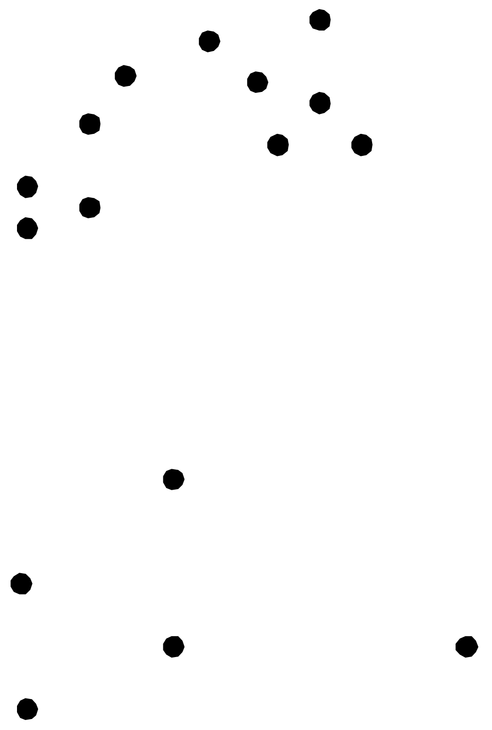
Un-Supervising Learning

- Cluster Analysis: Basic Concepts

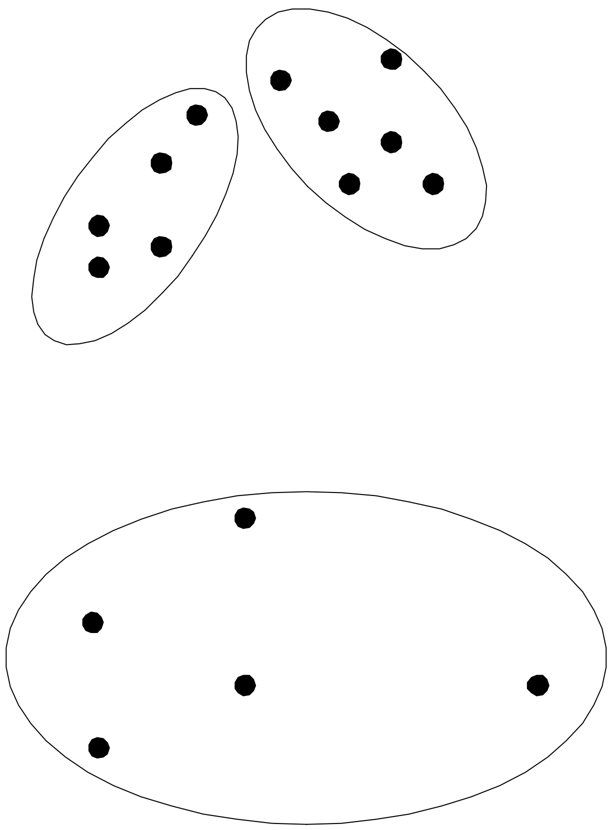
1. Partitioning Methods
2. Hierarchical Methods
3. Density-Based Methods

- Summary

Partitional Clustering



Original Points

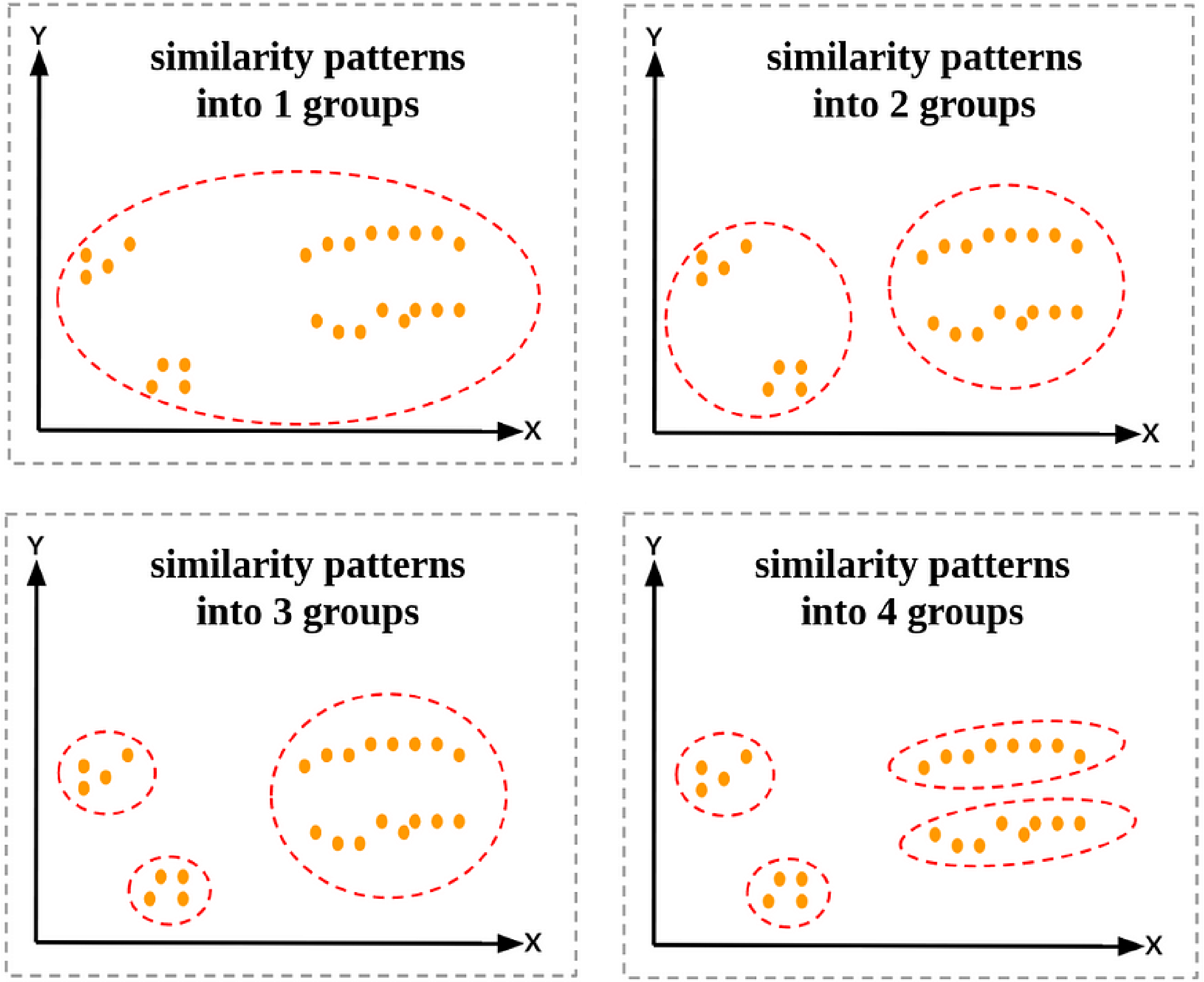


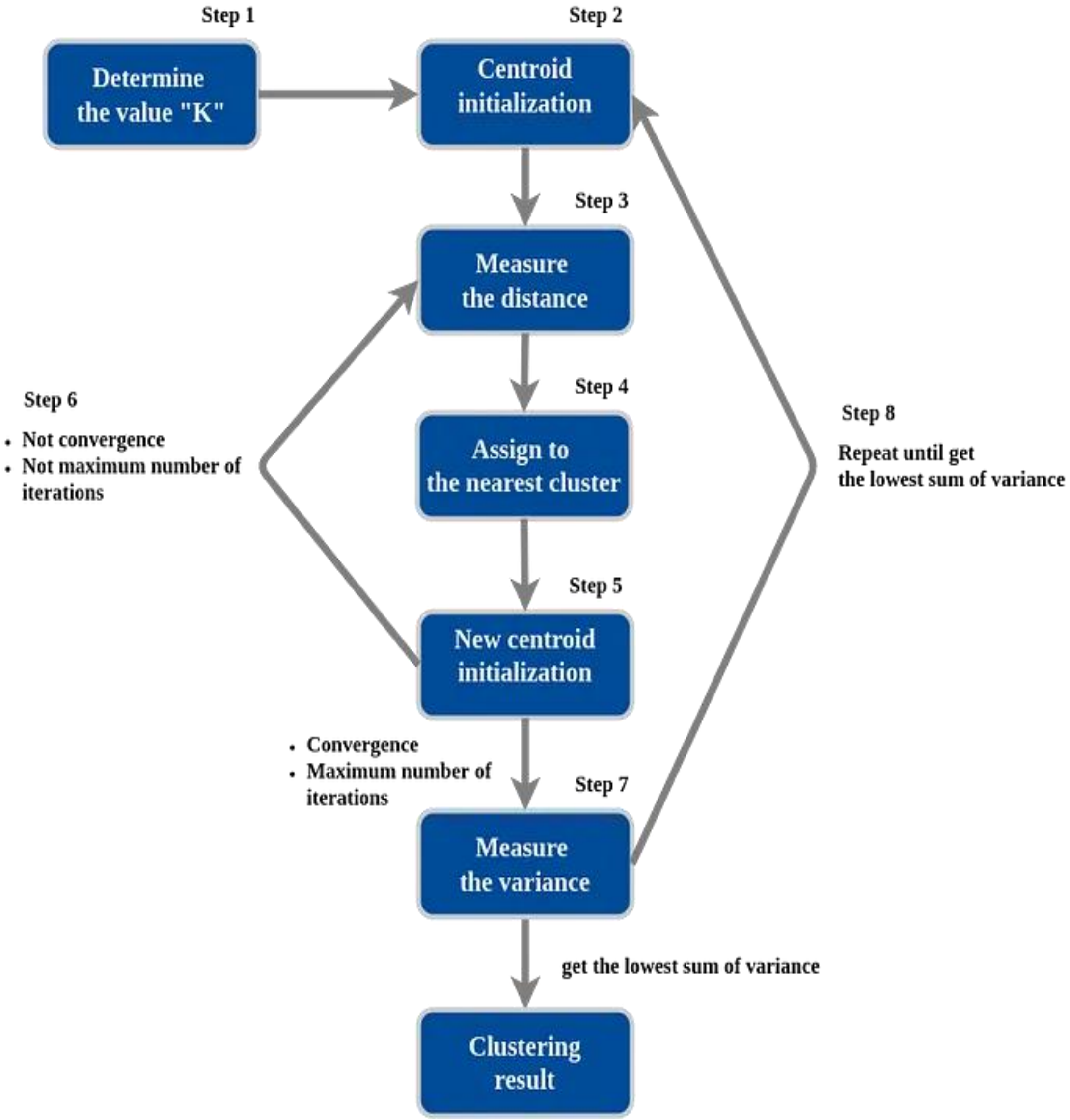
A Partitional Clustering

1-The K-Means Clustering Method

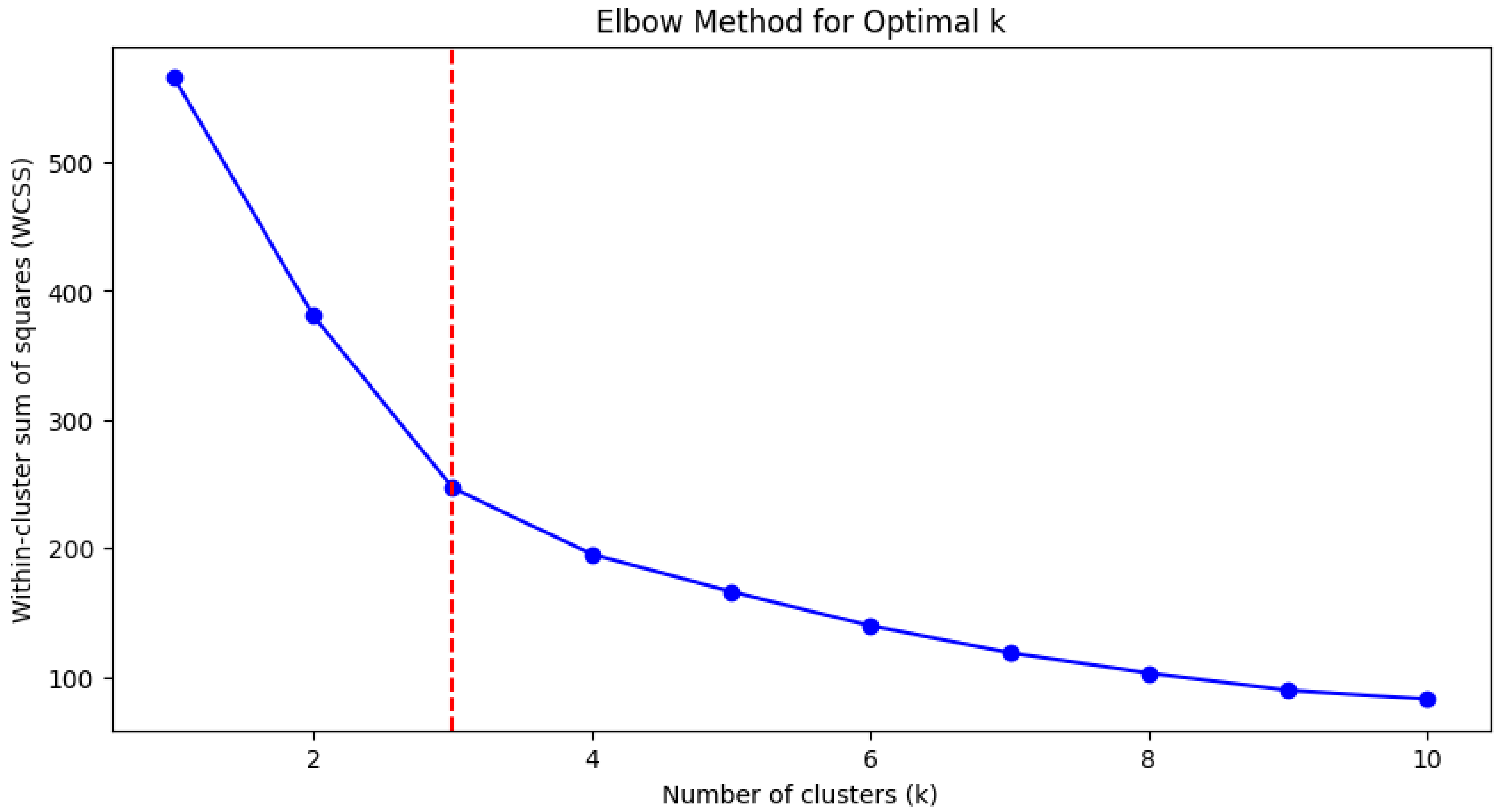
- Given *k*, the *k-means* algorithm is implemented in four steps:
  - **Partition objects** into *k* nonempty subsets
  - **Compute seed points** as the centroids of the clusters of the current partitioning (the centroid is the center, i.e., *mean point*, of the cluster)
  - **Assign each object** to the cluster with the nearest seed point
  - **Go back to Step 2**, stop when the assignment does not change

1-The K-Means Clustering Method … cont.



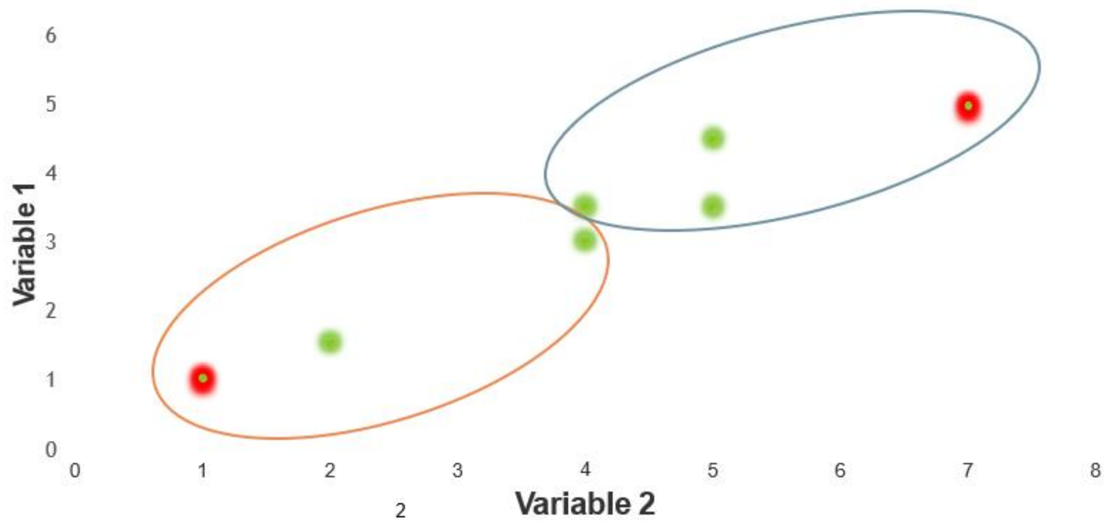


https://medium.com/data-folks-indonesia/step-by-step-to-understanding-k-means-clustering-and-implementation-with-sklearn-b55803f519d6


Elbow Method for Optimal k
Within-cluster sum of squares (WCSS)
500
400
300
200
100
2
4
6
8
10
Number of clusters (k)

A Simple example k-means (using K=2)

Individual	Variable 1	Variable 2
1	1	1
2	1.5	2
3	3	4
4	5	7
5	3.5	5
6	4.5	5
7	3.5	4.5



Step 1:

- **<u>Initialization</u>: Randomly we choose following two centroids (k=2) for two clusters. In this case the 2 centroid are: m1=(1.0,1.0) and m2=(5.0,7.0).**

	Individual	Mean Vector
Group 1	1	(1.0, 1.0)
Group 2	4	(5.0, 7.0)

Step 2:

	Centroid 1	Centroid 2
1	$\sqrt{(1-1)^2+(1-1)^2}$ = **0**	$\sqrt{(5-1)^2+(7-1)^2}$ = **7.21**
2	$\sqrt{(1-1.5)^2+(1-2)^2}$ = **1.12**	$\sqrt{(5-1.5)^2+(7-2)^2}$ = **6.10**
3	$\sqrt{(1-3)^2+(1-4)^2}$ = **3.61**	$\sqrt{(5-3)^2+(7-4)^2}$ = **3.61**
4	$\sqrt{(1-5)^2+(1-7)^2}$ = **7.21**	$\sqrt{(5-5)^2+(7-7)^2}$ = **0**
5	$\sqrt{(1-3.5)^2+(1-5)^2}$ = **4.72**	$\sqrt{(5-3.5)^2+(7-5)^2}$ = **2.5**
6	$\sqrt{(1-4.5)^2+(1-5)^2}$ = **5.31**	$\sqrt{(5-4.5)^2+(7-5)^2}$ = **2.06**
7	$\sqrt{(1-3.5)^2+(1-4.5)^2}$ = **4.30**	$\sqrt{(5-3.5)^2+(7-4.5)^2}$ = **2.92**

Step 2:

- Thus, we obtain two clusters containing: {1,2,3} and {4,5,6,7}.
- Their new centroids are:

$$\text{Group 1} = \left(\frac{1+1.5+3}{3}\right), \left(\frac{1+2+4}{3}\right) = (1.83\ ,\ 2.33)$$

$$\text{Group 2} = \left(\frac{5+3.5+4.5+3.5}{4}\right), \left(\frac{7+5+5+4.5}{4}\right) = (4.12\ ,\ 5.38)$$

Step 3:

	Centroid 1	Centroid 2
1	$\sqrt{(1.83 - 1)^2 + (2.33 - 1)^2} = 1.57$	$\sqrt{(4.12 - 1)^2 + (5.38 - 1)^2} = 5.38$
2	$\sqrt{(1.83 - 1.5)^2 + (2.33 - 2)^2} = 0.47$	$\sqrt{(4.12 - 1.5)^2 + (5.38 - 2)^2} = 4.29$
3	$\sqrt{(1.83 - 3)^2 + (2.33 - 4)^2} = 2.04$	$\sqrt{(4.12 - 3)^2 + (5.38 - 4)^2} = 1.78$
4	$\sqrt{(1.83 - 5)^2 + (2.33 - 7)^2} = 5.64$	$\sqrt{(4.12 - 5)^2 + (5.38 - 7)^2} = 1.84$
5	$\sqrt{(1.83 - 3.5)^2 + (2.33 - 5)^2} = 3.15$	$\sqrt{(4.12 - 3.5)^2 + (5.38 - 5)^2} = 0.73$
6	$\sqrt{(1.83 - 4.5)^2 + (2.33 - 5)^2} = 3.78$	$\sqrt{(4.12 - 4.5)^2 + (5.38 - 5)^2} = 0.54$
7	$\sqrt{(1.83 - 3.5)^2 + (2.33 - 4.5)^2} = 2.74$	$\sqrt{(4.12 - 3.5)^2 + (5.38 - 4.5)^2} = 1.08$

Therefore, the new clusters are:

{1,2} and {3,4,5,6,7}

Group 1 $=\left(\frac{1+1.5}{2}\right)$, $\left(\frac{1+2}{2}\right)$ $=(1.25 , 1.5)$

Group 2 $=\left(\frac{3+5+3.5+4.5+3.5}{5}\right)$, $\left(\frac{4+7+5+5+4.5}{5}\right)$ $=(3.9 , 5.1)$

Step 4:

	Centroid 1	Centroid 2
1	$\sqrt{(1.25 - 1)^2 + (1.5 - 1)^2} = \boxed{0.58}$	$\sqrt{(3.9 - 1)^2 + (5.1 - 1)^2} = 5.02$
2	$\sqrt{(1.25 - 1.5)^2 + (1.5 - 2)^2} = \boxed{0.56}$	$\sqrt{(3.9 - 1.5)^2 + (5.1 - 2)^2} = 3.92$
3	$\sqrt{(1.25 - 3)^2 + (1.5 - 4)^2} = 3.05$	$\sqrt{(3.9 - 3)^2 + (5.1 - 4)^2} = 1.42$
4	$\sqrt{(1.25 - 5)^2 + (1.5 - 7)^2} = 6.66$	$\sqrt{(3.9 - 5)^2 + (5.1 - 7)^2} = 2.20$
5	$\sqrt{(1.25 - 3.5)^2 + (1.5 - 5)^2} = 4.16$	$\sqrt{(3.9 - 3.5)^2 + (5.1 - 5)^2} = 0.41$
6	$\sqrt{(1.25 - 4.5)^2 + (1.5 - 5)^2} = 4.78$	$\sqrt{(3.9 - 4.5)^2 + (5.1 - 5)^2} = 0.61$
7	$\sqrt{(1.25 - 3.5)^2 + (1.5 - 4.5)^2} = 3.75$	$\sqrt{(3.9 - 3.5)^2 + (5.1 - 4.5)^2} = 0.72$

Therefore, there is no change in the cluster.

- Thus, the algorithm comes to a halt here and final result consist of 2 clusters {1,2} and {3,4,5,6,7}.

The K-Means Advantages

Simplicity: K-means is easy to implement and understand, making it a popular choice for clustering tasks, especially for large datasets.

Efficiency: It is computationally efficient and works well with large datasets.

Scalability: K-means can handle large datasets with ease, making it suitable for clustering in big data applications.

Interpretability قابلة للتفسير: The clusters produced by k-means are easy to interpret, as each data point is assigned to the cluster with the nearest mean.

Versatility تنوع فى الداتا : K-means can be applied to various types of data, including numerical and categorical data, making it a versatile clustering algorithm.

The K-Means Disadvantages

Sensitive to Initial Centroids: K-means clustering is sensitive to the initial selection of cluster centroids, which can lead to different results for each run.

Assumes Spherical Clusters مجموعات كروية : K-means assumes that clusters are spherical and have similar sizes, which may not always hold true for real-world datasets with irregularly shaped clusters or clusters of varying sizes.

Requires Predefined Number of Clusters: The number of clusters (k) needs to be specified in advance, which may not always be known and choosing an inappropriate value for k can result in suboptimal clustering.

Outlier Sensitivity: K-means is sensitive to outliers, as they can disproportionately affect the positions of cluster centroids and the overall clustering result.

Non-Robust to Noise: K-means may produce poor results when dealing with noisy data or datasets with overlapping clusters.

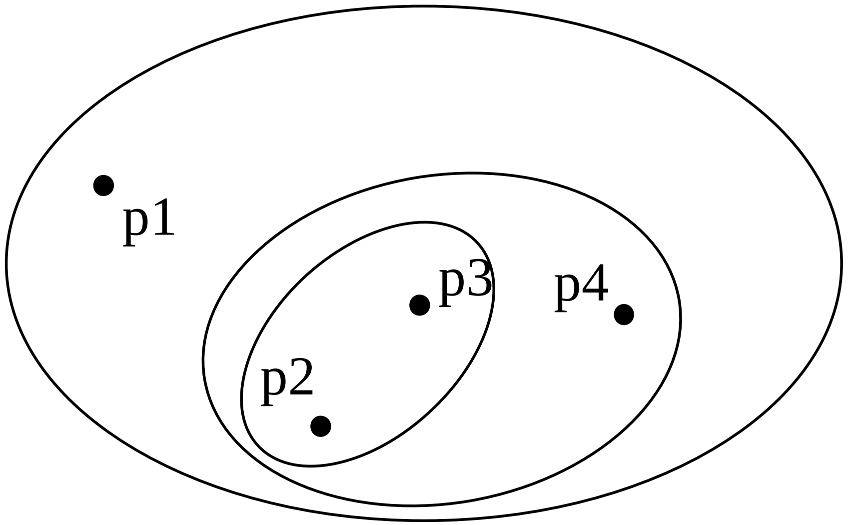
Un-Supervising Learning

- Cluster Analysis: Basic Concepts

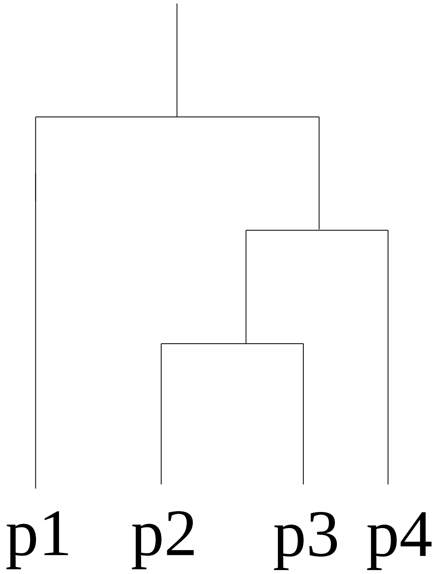
1. Partitioning Methods
2. Hierarchical Methods
3. Density-Based Methods

- Summary

Hierarchical Clustering



Traditional Hierarchical Clustering



Traditional Dendrogram

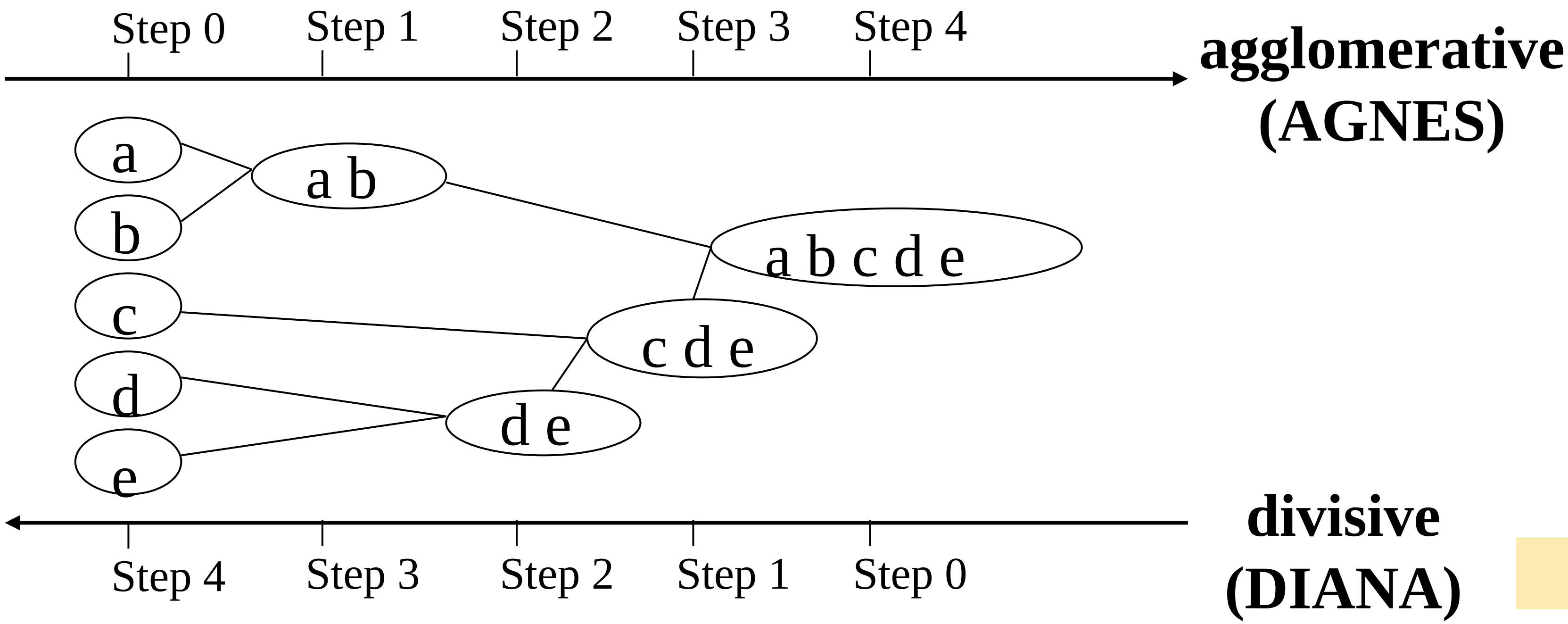
2- Hierarchical Clustering

- **Hierarchical clustering** is a method of cluster analysis that builds a hierarchy of clusters, use distance matrix as clustering criteria.
- This method *does not require the number of clusters k* as an input, but needs a termination condition

Main types of hierarchical clustering

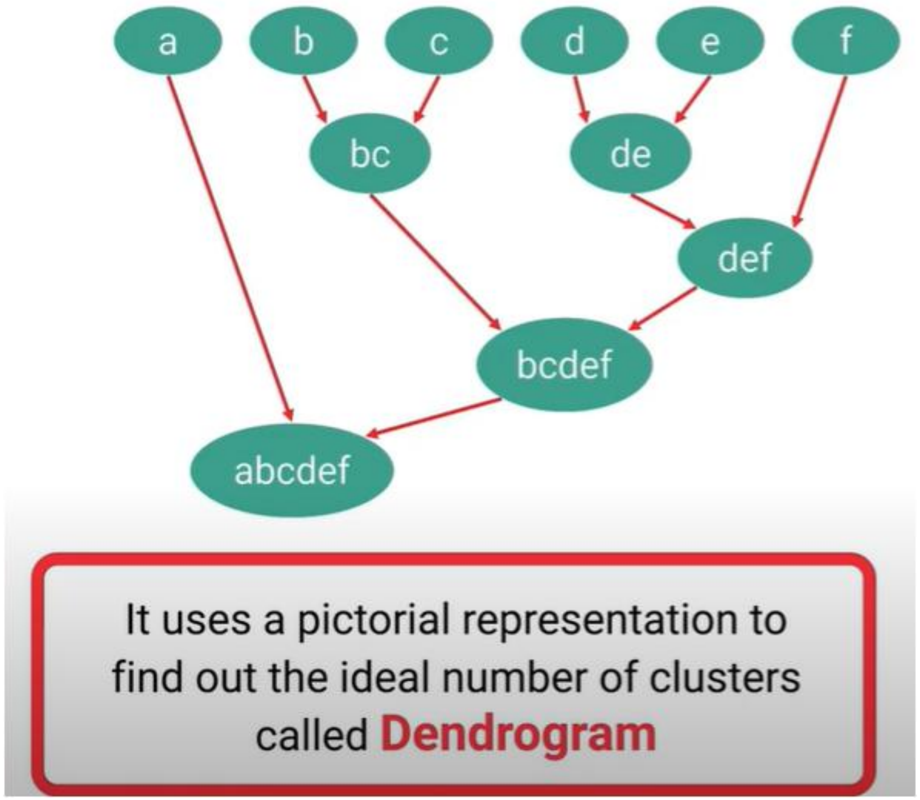
1. **Agglomerative Clustering**: This bottom-up approach begins with each data point as a separate cluster and then merges the closest pairs of clusters iteratively until only one cluster remains.

2. **Divisive Clustering**: This top-down approach starts with all data points in a single cluster and then recursively divides the dataset into smaller clusters until each cluster contains only one data point.



2-1 AGNES (Agglomerative Nesting)

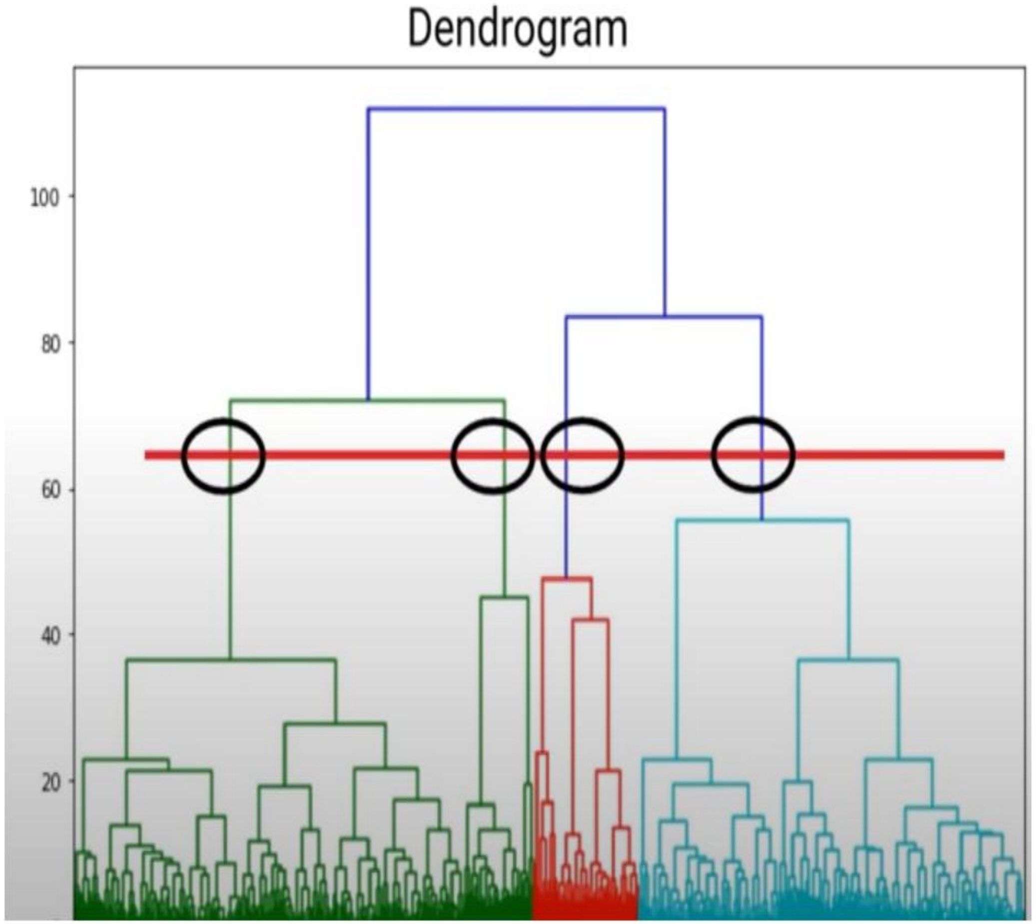
- Use the **single-link (smallest distance between one point and cluster)** method and the dissimilarity matrix
- Merge nodes that have the least dissimilarity الاقل اختلافا
- Go on in a Ascending fashion
- Eventually all nodes belong to the same cluster



https://www.youtube.com/watch?app=desktop&v=k1ZU51B-33k

Dendrogram: Shows How Clusters are Merged

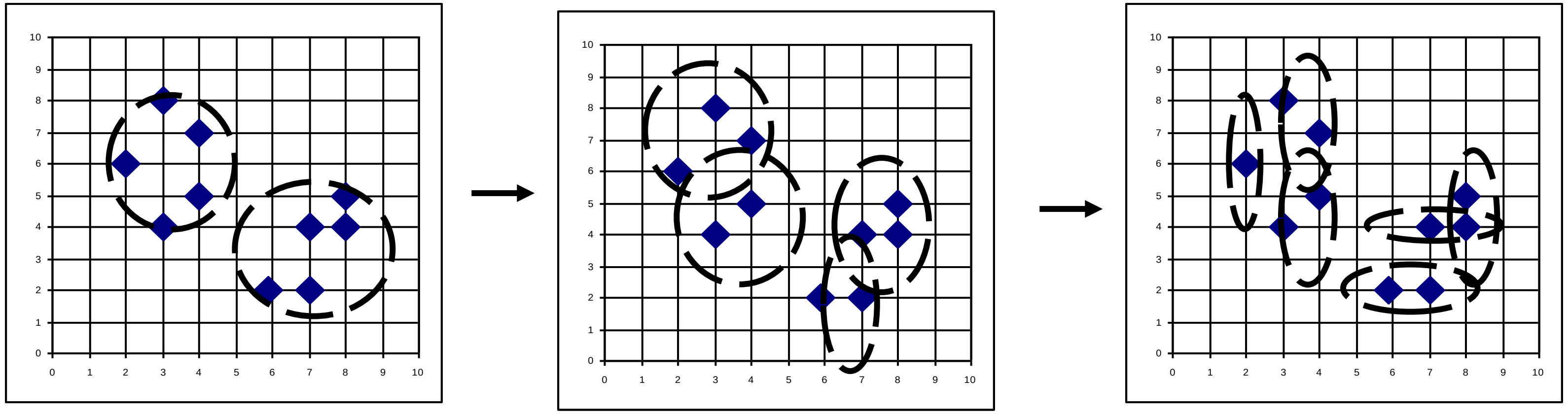
- Decompose data objects into a several levels of nested partitioning (tree of clusters), called a ***dendrogram***
- A ***clustering*** of the data objects is obtained by ***cutting*** the dendrogram at the desired level, then each ***connected*** component forms a cluster



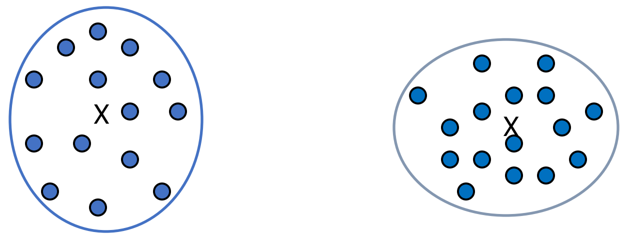
https://www.youtube.com/watch?app=desktop&v=k1ZU51B-33k

2-2 DIANA (Divisive Analysis)

- Introduced in Kaufmann and Rousseeuw (1990)
- Implemented in statistical analysis packages, e.g., Splus
- Inverse order of AGNES
- Eventually each node forms a cluster on its own



Distance between Clusters



- **Single link**: smallest distance between an element in one cluster and an element in the other, i.e., $dist(K_i, K_j) = min(t_{ip}, t_{jq})$
- **Complete link**: largest distance between an element in one cluster and an element in the other, i.e., $dist(K_i, K_j) = max(t_{ip}, t_{jq})$
- **Average**: avg distance between an element in one cluster and an element in the other, i.e., $dist(K_i, K_j) = avg(t_{ip}, t_{jq})$

- **Centroids** are the average of all points in the cluster and are not necessarily part of the data set.
- **medoids** are actual data points that represent the center of the cluster.

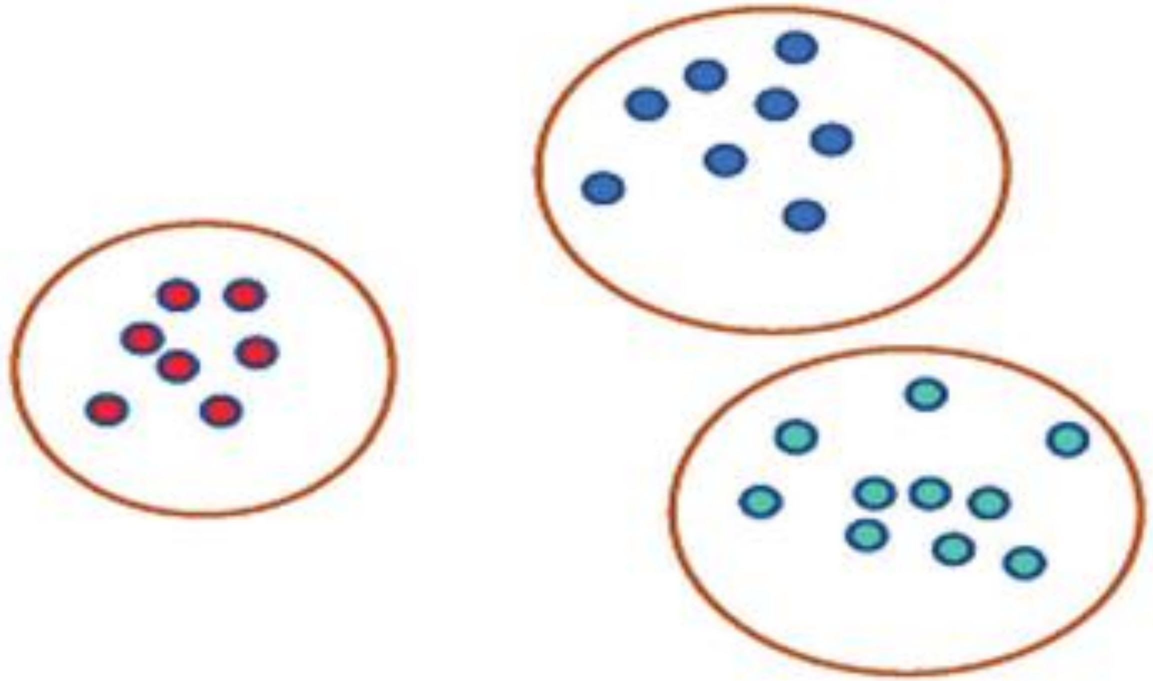
- Centroids are sensitive to outliers, while medoids are more robust to noise and outliers.
- Centroids are typically used in K-means clustering, whereas medoids are used in K-medoids or PAM clustering.

Un-Supervising Learning

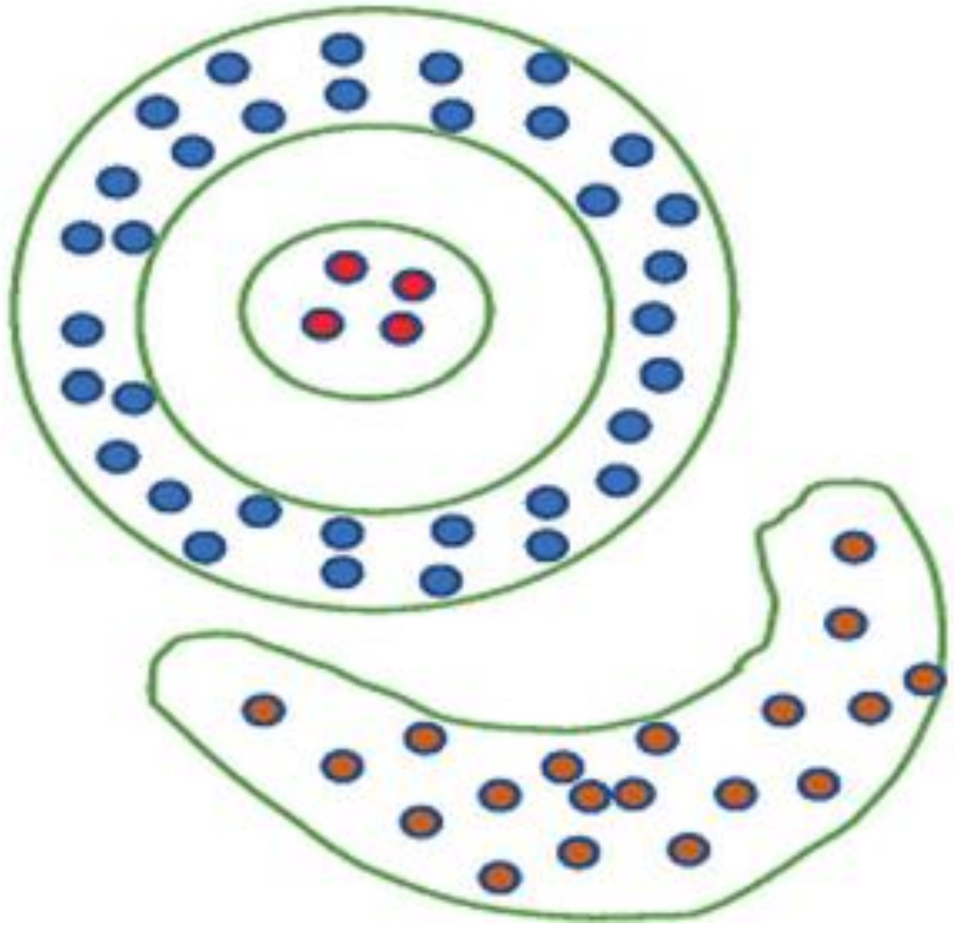
- Cluster Analysis: Basic Concepts

1. Partitioning Methods
2. Hierarchical Methods
3. Density-Based Methods

Spherical-shape clusters



Arbitrary-shape clusters



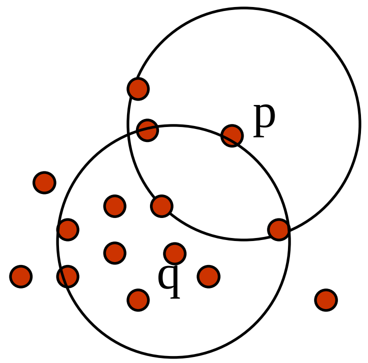
3-Density-Based Clustering Methods

- Density-based clustering methods are ***algorithms*** used in machine learning and data mining to group data points ***based on their density in the feature space***.
- One of the most popular density-based clustering algorithms is DBSCAN (Density-Based Spatial Clustering of Applications with Noise).

Density-Based Clustering: Basic Concepts

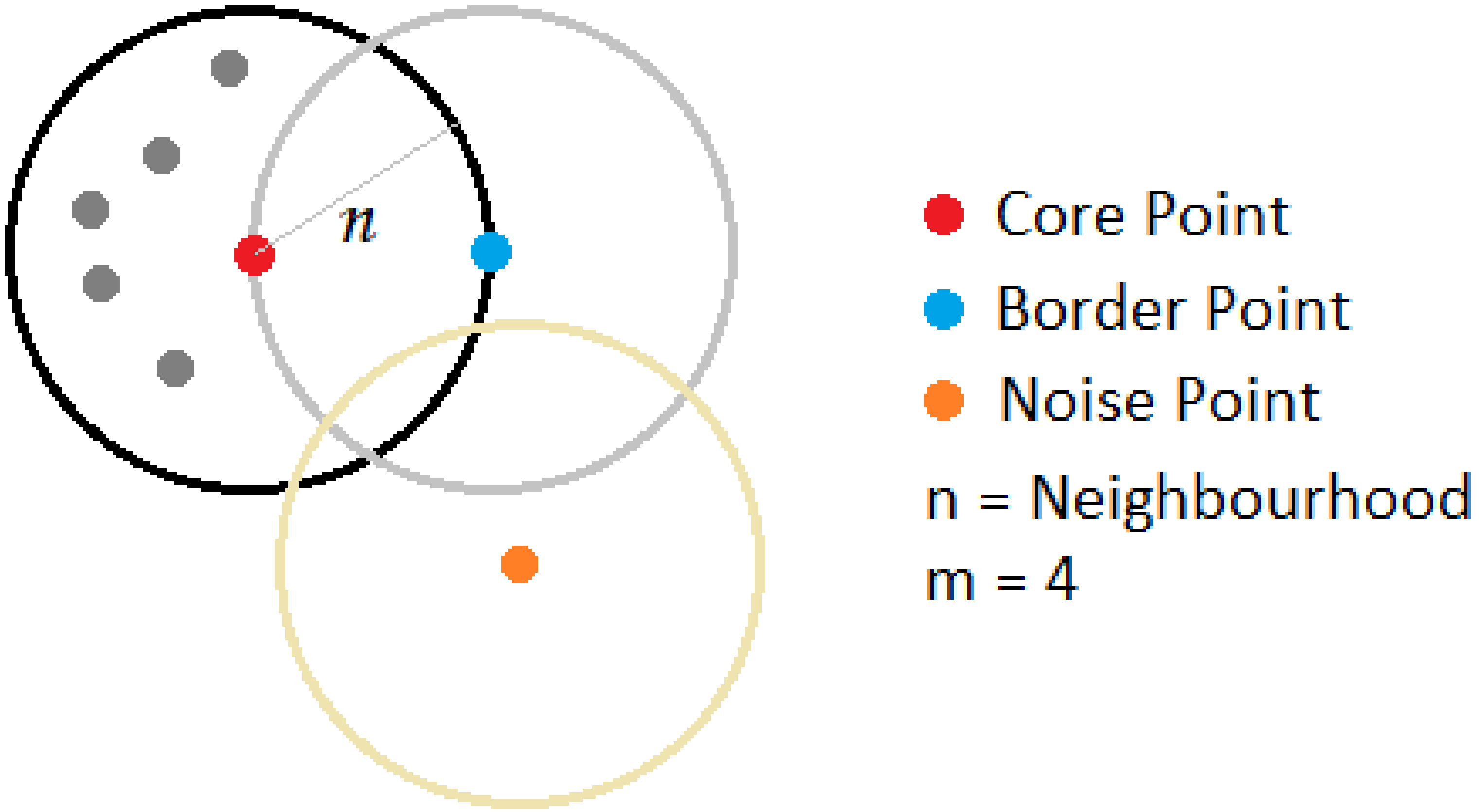
- **In DBSCAN, we define two parameters:**
    - **Eps(epsilon)**: Maximum radius of the neighborhood
    - **MinPts**: Minimum number of points in an Eps- neighborhood of that point
- **NEps(q)**: {p belongs to D | dist(p,q) ≤ Eps}
- **Directly density-reachable**: A point *p* is directly density-reachable from a point *q* w.r.t. *Eps*, *MinPts* if
    - *p* belongs to $N_{Eps}(q)$
    - core point condition:

$$|N_{Eps}(q)| \geq MinPts$$



MinPts = 5

Eps = 1 cm



DBSCAN CLUSTERING

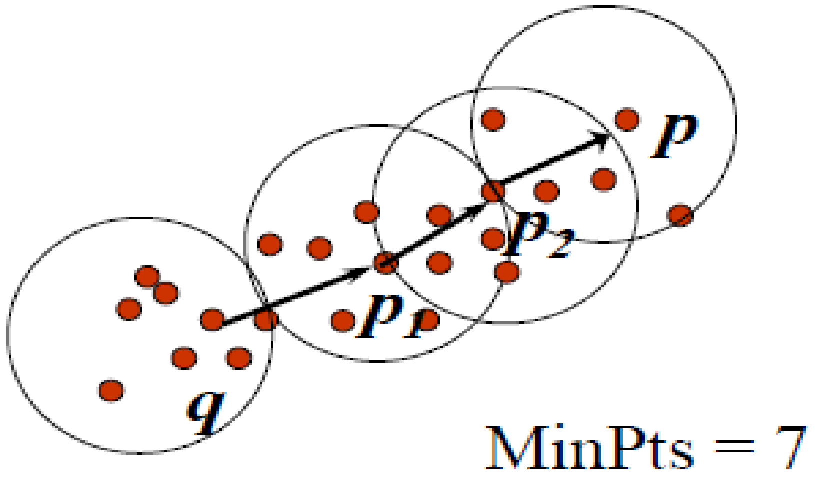
Abhijit Annaldas

Density-Based Clustering: Cluster Formation

- We start by randomly selecting a ***data point*** and examine its ε-neighborhood.
- If the ***number of points*** within the ***ε-neighborhood exceeds the MinPts threshold***, the point is labeled as a ***core point***.
- We then expand the cluster by iteratively examining the ***ε-neighborhood*** of each core point and adding its neighbors to the cluster.
- If a core point's neighborhood ***contains other core points***, their ***clusters are merged***.
- ***Border points***, which are within the ε-neighborhood of a core point but ***do not have enough neighbors*** to be considered core points themselves, are assigned to the same cluster as their core point.
- Points that are not core or border points are labeled as ***noise***.

Density-reachability

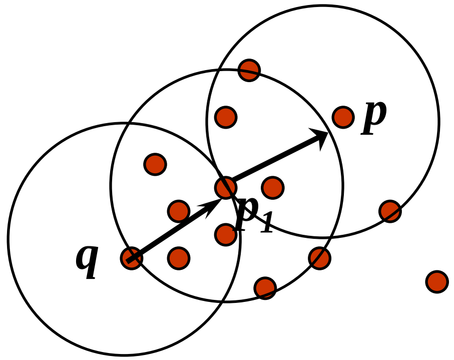
- Density-Reachable:
  - A point p is directly density-reachable from p2;
  - p2 is directly density-reachable from p1;
  - p1 is directly density-reachable from q;
  - p←p2←p1←q form a chain.



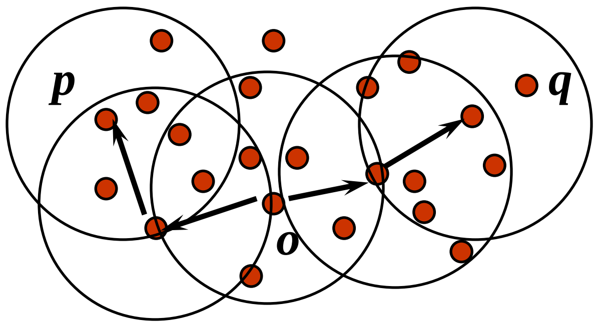
p is density-reachable from q

Density-Reachable and Density-Connected

- **Density-reachable**:
  - A point p is density-reachable from a point q w.r.t. *Eps*, *MinPts* if there is a chain of points $p_1, \ldots, p_n$, $p_1 = q$, $p_n = p$ such that p_{i+1} is directly density-reachable from p_i



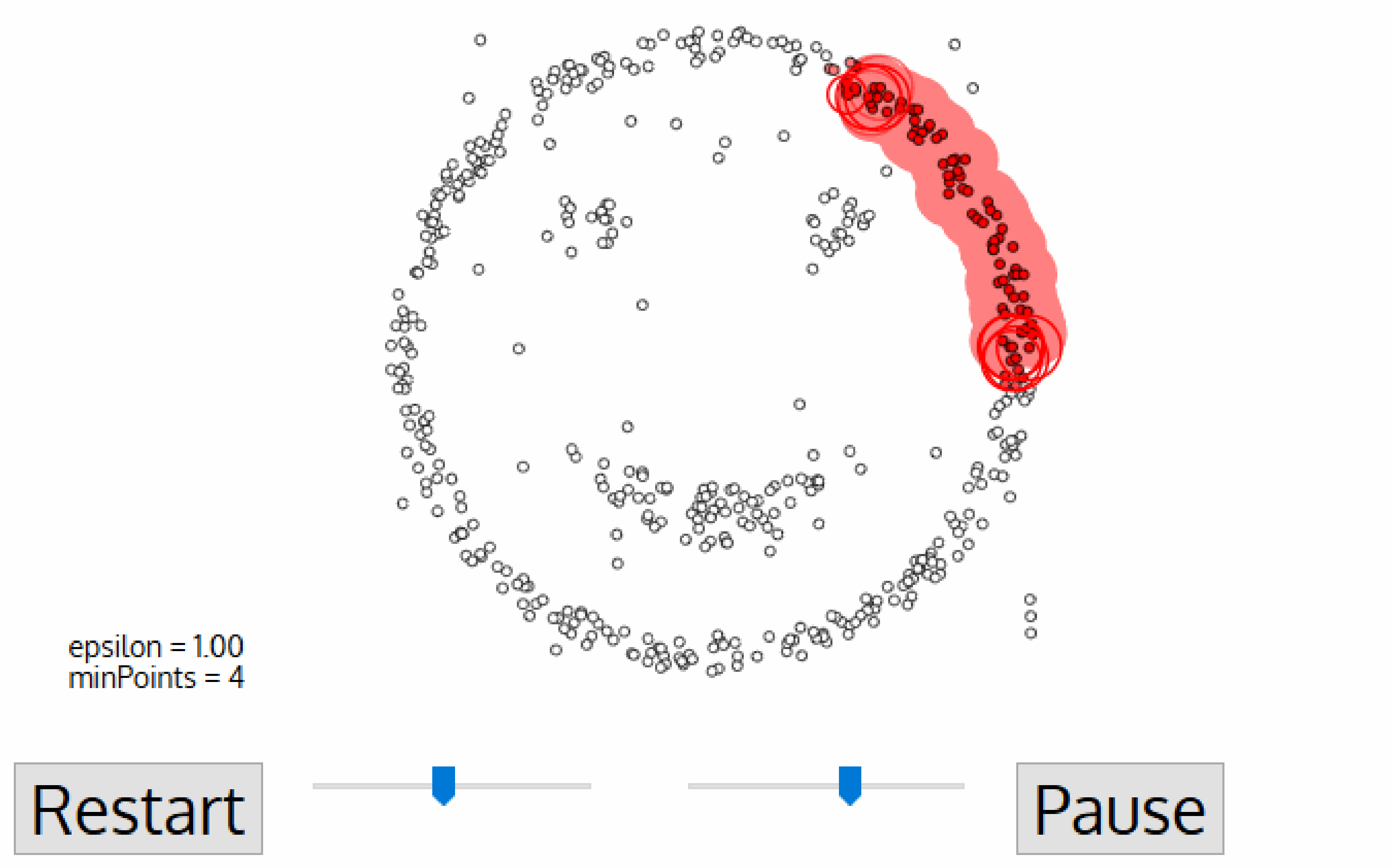
- **Density-connected**
  - A point p is density-connected to a point q w.r.t. *Eps*, *MinPts* if there is a point o such that both, p and q are density-reachable from o w.r.t. *Eps* and *MinPts*



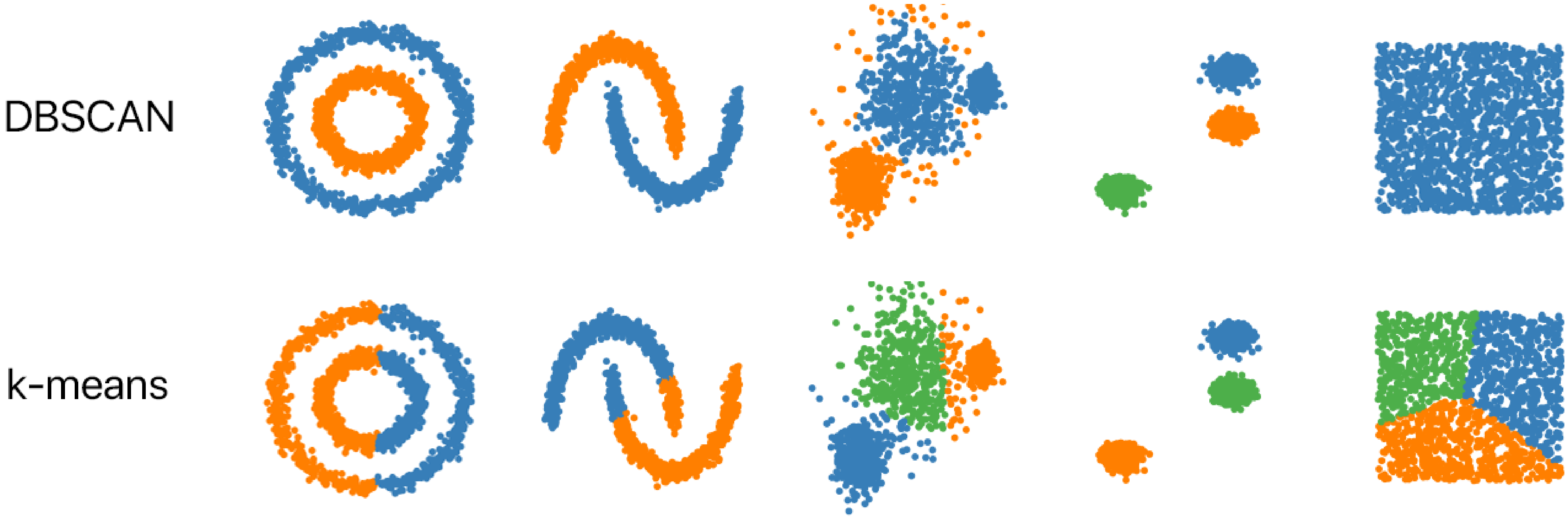
Example:

- Let's say ε = 0.1 (degrees) and MinPts = 4.
- We start by selecting a ***random data point***.
- We then examine its ***ε-neighborhood*** (a circle with radius 0.1 degrees) and count the number of points within this neighborhood.
- If there are at least 4 points within this neighborhood, the selected point is labeled as ***a core point***, and its neighbors are ***added to the same cluster***.
- We continue this process, expanding clusters and merging them until all points are assigned to clusters or labeled as noise.

DBSCAN Example



https://www.kdnuggets.com/2020/04/dbscan-clustering-algorithm-machine-learning.html


DBSCAN
k-means

Summary

- Cluster analysis groups objects based on their similarity and has wide applications
- Measure of similarity can be computed for various types of data
- Clustering algorithms can be categorized into partitioning methods, hierarchical methods, density-based methods, grid-based methods, and model-based methods
- K-means and K-medoids algorithms are popular partitioning-based clustering algorithms
- Birch and Chameleon are interesting hierarchical clustering algorithms, and there are also probabilistic hierarchical clustering algorithms
- DBSCAN, OPTICS, and DENCLU are interesting density-based algorithms

Silhouette score

- The silhouette value is a measure of how similar an object is to its own cluster (cohesion) compared to other clusters (separation).
- The silhouette ranges from −1 to +1, where a high value indicates that the object is well matched to its own cluster and poorly matched to neighboring clusters.

Example Dataset:

Imagine we have 6 data points that have been clustered into two clusters based on a clustering algorithm:

Cluster Assignments:

- Cluster 1: Points A, B, C
- Cluster 2: Points D, E, F

The coordinates of the points are:

Point	Coordinates	Cluster
A	(1, 1)	1
B	(2, 1)	1
C	(1, 2)	1
D	(10, 10)	2
E	(10, 11)	2
F	(11, 10)	2

Calculation of Silhouette Scores:

We will calculate the silhouette score for each point. To keep it simple, we will manually calculate the scores for Points A and D.

Point	Coordinates	Cluster
A	(1, 1)	1
B	(2, 1)	1
C	(1, 2)	1
D	(10, 10)	2
E	(10, 11)	2
F	(11, 10)	2

Calculation of Silhouette Scores:

We will calculate the silhouette score for each point. To keep it simple, we will manually calculate the scores for Points A and D.

Point A:

1. **Intra-cluster distance (a):** Calculate the average distance from Point A to Points B and C within Cluster 1.

   Distance(A, B) = `sqrt((1-2)^2 + (1-1)^2)` = 1

   Distance(A, C) = `sqrt((1-1)^2 + (1-2)^2)` = 1

   Average distance (a) = (Distance(A, B) + Distance(A, C)) / 2 = (1 + 1) / 2 = 1.

2. **Nearest-cluster distance (b):** Calculate the average distance from Point A to Points D, E, and F in Cluster 2.

   Distance(A, D) = `sqrt((1-10)^2 + (1-10)^2)` ≈ 12.73

   Distance(A, E) = `sqrt((1-10)^2 + (1-11)^2)` ≈ 13.45

   Distance(A, F) = `sqrt((1-11)^2 + (1-10)^2)` ≈ 13.45

   Average distance (b) = (Distance(A, D) + Distance(A, E) + Distance(A, F)) / 3 ≈ (12.73 + 13.45 + 13.45) / 3 ≈ 13.21.

3. **Silhouette score for Point A:**

   $s = \frac{b-a}{\max(a,b)} = \frac{13.21-1}{13.21} \approx 0.924$

Point B:

1. **Intra-cluster distance (a):** Same as Point A, since the cluster is symmetric.
   - Average distance (a) for Point B = 1.
2. **Nearest-cluster distance (b):** Same as Point A, since the cluster is symmetric.
   - Average distance (b) for Point B ≈ 13.21.
3. **Silhouette score for Point B:**
   - $s \approx \frac{13.21-1}{13.21} \approx 0.924.$

Point C:

- Repeat the same steps as for Point B (since all Points in Cluster 1 are equidistant from each other and from Cluster 2).
- **Silhouette score for Point C** ≈ 0.924.

Point E:

1. **Intra-cluster distance (a):** Same as Point D.
   - Average distance (a) for Point E = 1.
2. **Nearest-cluster distance (b):** Same as Point D.
   - Average distance (b) for Point E ≈ 13.21.
3. **Silhouette score for Point E:**
   - $s \approx \frac{13.21-1}{13.21} \approx 0.924.$

Point D:

1. **Intra-cluster distance (a):** Calculate the average distance from Point D to Points E and F within Cluster 2.

   Distance(D, E) = `sqrt((10-10)^2 + (10-11)^2)` = 1

   Distance(D, F) = `sqrt((10-11)^2 + (10-10)^2)` = 1

   Average distance (a) = (Distance(D, E) + Distance(D, F)) / 2 = (1 + 1) / 2 = 1.

2. **Nearest-cluster distance (b):** Calculate the average distance from Point D to Points A, B, and C in Cluster 1. (We already calculated this in reverse when computing for Point A, so we'll use the same value.)

   Average distance (b) = 13.21 (same as for Point A).

3. **Silhouette score for Point D:**

   $s = \frac{b-a}{\max(a,b)} = \frac{13.21-1}{13.21} \approx 0.924$

Point	Coordinates	Cluster
A	(1, 1)	1
B	(2, 1)	1
C	(1, 2)	1
D	(10, 10)	2
E	(10, 11)	2
F	(11, 10)	2

Point E:

1. **Intra-cluster distance (a):** Same as Point D.
   - Average distance (a) for Point E = 1.
2. **Nearest-cluster distance (b):** Same as Point D.
   - Average distance (b) for Point E ≈ 13.21.
3. **Silhouette score for Point E:**
   - $s \approx \frac{13.21-1}{13.21} \approx 0.924.$

Point F:

- Repeat the same steps as for Point E.
- **Silhouette score for Point F** ≈ 0.924.

Now, with the silhouette scores calculated for each point as approximat the average silhouette score for the entire dataset:

$$\text{Average Silhouette Score} = \frac{\sum_{i=1}^{n} s_i}{n}$$

where s_i is the silhouette score for each point and n is the total numbe

$$\text{Average Silhouette Score} = \frac{0.924 \times 6}{6} = 0.924$$

The average silhouette score for the clustering is 0.924, indicating a str are well matched to their own clusters and clearly separated from other that the clustering configuration is appropriate for the given dataset.