Data Science and Artificial Intelligence

Machine Learning

Unsupervised learning

Lecture No.3

By- SIDDHARTH SABHARWAL SIR

Recap of Previous Lecture

Topic Kmeans

Topic K medoid

Topic Aglomerative

Topic

Topic

Topics to be Covered

- Topic K medoid
- Topic Hierarchical
- Topic
- Topic
- Topic

WHEN YOU FOCUS ON THE GOOD THE GOOD GETS BETTER

Clustering

Example

The data points are :

- (2, 3)
- (3, 3)
- (6, 5)
- (8, 8)
- (3, 2)
- (5, 7)
- (9, 8)
- The centroids are initialized as
- Centroid 1: (2, 3)
- Centroid 2: (8, 8)
- Find the centroid after one iteration?

$C_1(8/3, 8/3)$

$C_2(7, 7)$

* K means clustering heavily effected by
Outlier and noise in data
we find mean of clusters → mean location was not Part of data
Only one change
in K medoid we find that point in cluster which has min Sum of E.D with all other points of cluster. → But medoid will be a point within data.

What is K medoid Algorithm

- K-medoids clustering is a partitioning method similar to K-means clustering, but it is more robust to noise and outliers. (How)
- The basic idea is to find representative objects (medoids) in the data set and form clusters around these medoids.

What is K medoid Algorithm

- Key Concepts
- Medoid: A medoid is the most centrally located data point in a cluster. Unlike the centroid in K-means, a medoid is an actual data point from the dataset.
- Cluster: A group of data points that are more similar to each other than to points in other clusters.

Steps K medoid Algorithm

- Steps in K-medoids Clustering K medoids → any K Random points from dataset.

1 Initialization: Select kk initial medoids randomly from the dataset.

2 Assignment: Assign each data point to the nearest medoid based on a chosen distance metric (e.g., Euclidean distance, Manhattan distance). → So we create K clusters.

3. Update Medoids: For each cluster, find the point that minimizes the sum of distances to all other points in the cluster. This point becomes the new medoid for that cluster. → highly Complex.

4. Repeat: Repeat the assignment and update steps until the medoids no longer change or the changes are very minimal.

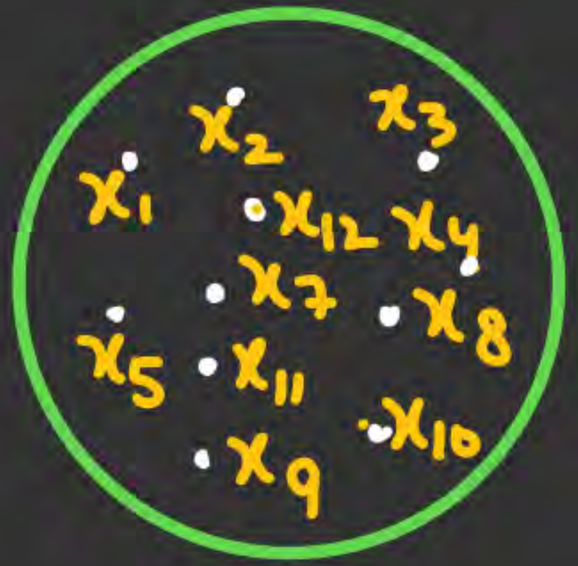
Step3 ⇒

Inside i^{th} cluster, with N_i No of points

$$\sum_{k=1}^{N_i} \|x_j - x_k\|^2$$

- we have to find this for $j = 1$ to N_i
- the point in cluster which has min value ⇒ medoid.

For any j^{th} point we find Sum of ED from all other points in cluster.



Example... (take K = 2)

Consider the following 2D points:

Initialise (3,4) (4,7)

Point	X	Y
A	2	6
B	3	4
C	3	8
D	4	7
E	6	2

	M_1	M_2
	$(3,4)$	$(4,7)$
$m_2 \leftarrow (2,6)$	$\sqrt{1^2+2^2}$	$\sqrt{2^2+1^2}$
$m_1 \leftarrow (3,4)$	$\sqrt{0}$	$\sqrt{1^2+3^2}$
$m_2 \leftarrow (3,8)$	$\sqrt{4^2}$	$\sqrt{1^2+1^2}$
$m_2 \leftarrow (4,7)$	$\sqrt{1+3^2}$	$\sqrt{0}$
$m_1 \leftarrow (6,2)$	$\sqrt{3^2+2^2}$	$\sqrt{2^2+5^2}$

$(3,4) \leftrightarrow (6,2)$

any one can be medoid

$(2,6)$, $(3,8)$, $(4,7)$ $\Rightarrow$ any one of $(3,8)$, $(4,7)$ can be medoid.

$(2,6) \Rightarrow \sqrt{1^2+2^2} + \sqrt{2^2+1^2}$

$\Rightarrow 2\sqrt{5}$

$(3,8) \Rightarrow \sqrt{1^2+2^2} + \sqrt{1^2+1^2}$

$(4,7) \Rightarrow \sqrt{2^2+1^2} + \sqrt{1^2+1^2}$

• Advantage ⇒ ?

How it is Robust ⇒

* bcoz outlier can effect the centroid but not medoid

Advantage and Disadvantage of K Medoid

- Advantages of K-medoids
- Robustness to Noise and Outliers: Medoids, being actual data points, are less influenced by extreme values compared to centroids.
- Flexibility: It can use various distance metrics and is not limited to Euclidean space.
- Disadvantages of K-medoids
- Computationally Intensive: Calculating the medoid for each cluster can be more time-consuming than computing centroids, especially for large datasets.
- Scalability: It may not scale well to very large datasets due to its iterative nature and the need to compute distances between all pairs of points in a cluster.

Clustering

Hierarchical Clustering

- Two Approach :
- 1. Bottom Up : Agglomerative ✓
- 2. Top Down : Divisive ✓

Here we create Trees...

⇒ we create trees

Hierarchical Clustering : Bottom Up Approach

- We have Leaf Nodes :

- We have Internal Nodes :

- Tree ends at the root node

Clustering

Hierarchical Clustering : Bottom Up Approach

So How we initialise and start this method ...

We don't need K ...

- Bottom up approach

N data points

In Bottom up approach

1. we start with N leaf nodes ⇒ N leaf clusters → this means only 1 Point in each cluster
2. Now we find distance b/w all clusters, Clusters with min distance shd be Combined, So we get N-1 Clusters.
3. Repeat Step 2 until we get a single cluster → Root node.

Hierarchical Clustering : Bottom Up Approach

Steps in Agglomerative Clustering

- **Initialization**:
  - Start with each data point as a single cluster. This results in N clusters for N data points.
- **Compute Distance Matrix**:
  - Calculate the distance matrix, which contains the distances between all pairs of data points.
- **Merge Clusters**:
  - Identify the pair of clusters that are closest to each other based on the chosen linkage criteria and merge them into a single cluster.
- **Update Distance Matrix**:
  - Update the distance matrix to reflect the distances between the new cluster and the remaining clusters.
- **Repeat**:
  - Repeat the merging process until all points are merged into a single cluster or until a stopping criterion is met (e.g., a desired number of clusters).
- **Construct Dendrogram**:
  - The merging process can be visualized as a dendrogram, a tree-like diagram that records the sequence of merges and the distances at which they occur.

Hierarchical Clustering : Bottom Up Approach

What is a active set…

At each step in this algorithm…

- Consider the following set of 6 one dimensional data points:
- 18, 22, 25, 42, 27, 43

	18	22	25	27	42	43
18	0	4	7	9	24	25
22	4	0	3	5	20	21
25	7	3	0	2	17	18
27	9	5	2	0	15	16
42	24	20	17	15	0	1
43	25	21	18	16	1	0

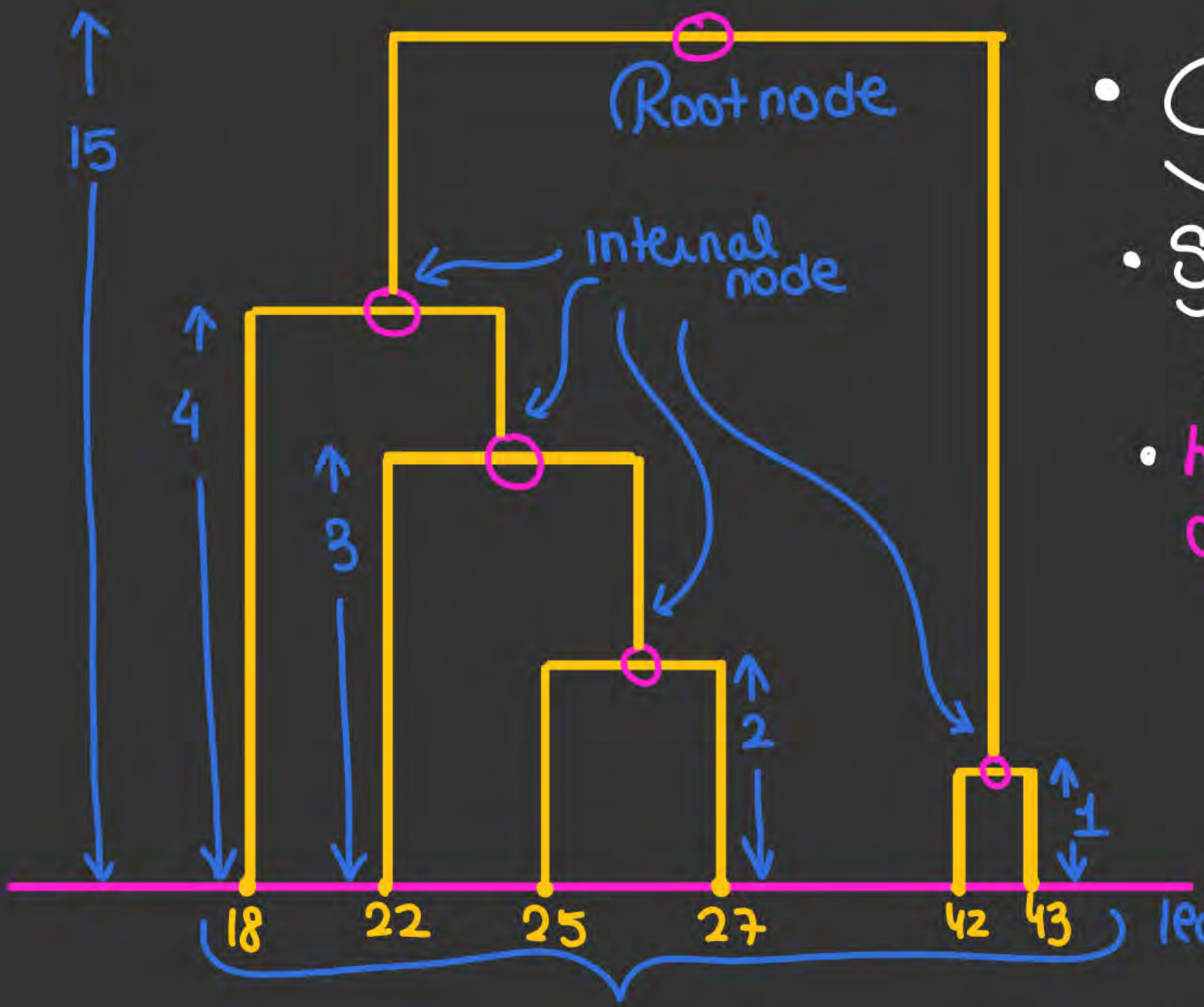
	18	22	25	27	(42,43)
18	0	4	7	9	24
22	4	0	3	5	20
25	7	3	0	2	17
27	9	5	2	0	15
(42,43)	24	20	17	15	0

	18	22	(25,27)	(42,43)
18	0	4	7	24
22	4	0	3	20
(25,27)	7	3	0	15
(42,43)	24	20	15	0

	18	(22,25,27)	(42,43)
18	0	4	24
22,25,27	4	0	15
42,43	24	15	0

	(18,22,25,27)	(42,43)
(18,22,25,27)	0	15
(42,43)	15	0

No need of K initialization



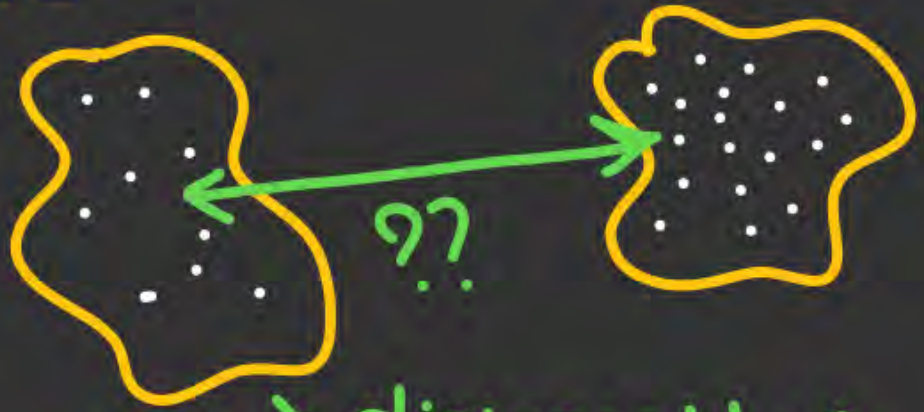
- dendogram
- Similarity metric → distance
- heights in dendogram $\propto$ dissimilarity

Hierarchical Clustering : Bottom Up Approach

- A dendrogram is a tree-like diagram that records the sequences of merges or splits in hierarchical clustering. It is a useful tool to visualize the arrangement of the clusters produced by hierarchical clustering. In a dendrogram:
  - Each leaf (or node) at the bottom represents an individual data point.
  - Branches that join together at a higher level represent clusters formed by combining two or more clusters at a previous stage.
  - The height of each branch indicates the distance or dissimilarity between the clusters or points that are being joined.
  - The y-axis represents the distance or similarity measure at which clusters are merged. The x-axis can represent the individual data points and clusters.

we know how to find distance b/w points

Eucledian
manhattan



→ distance b/w 2 Clusters
4 method.

Clustering

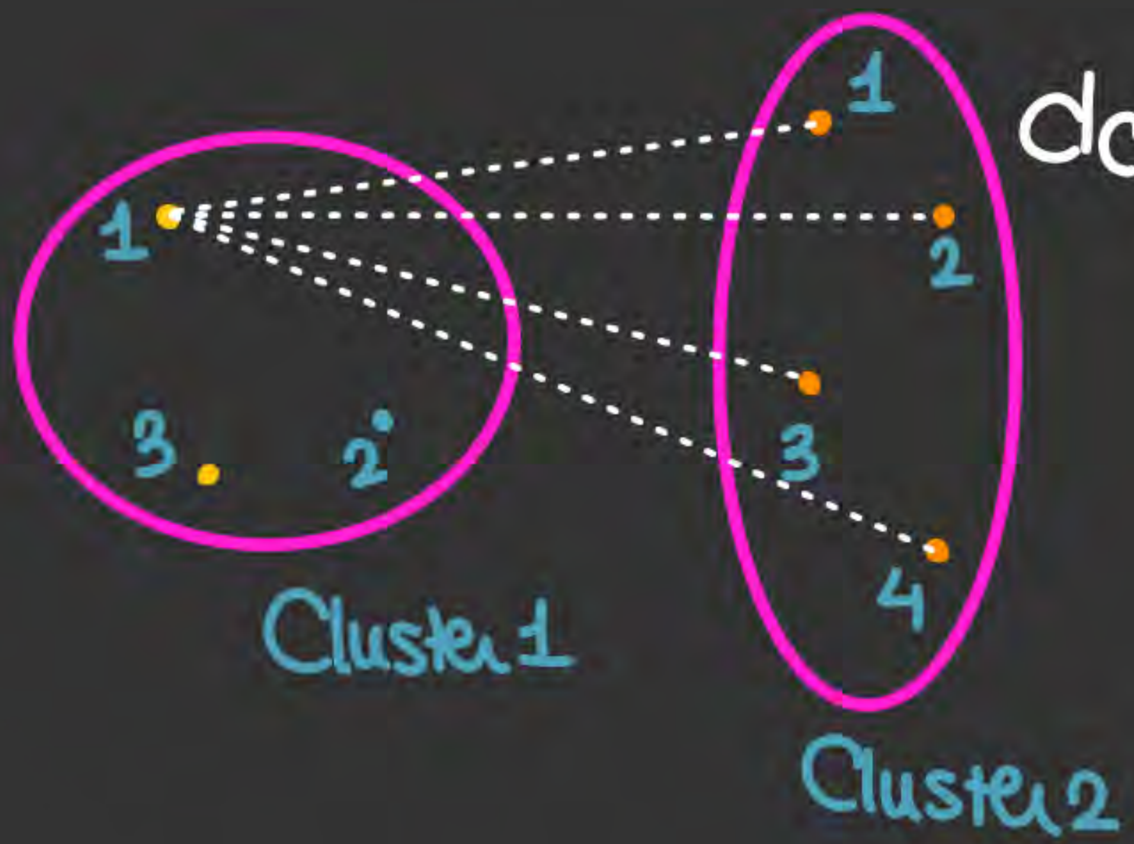
How we find the group linkage/ distance between groups ?

4 type of linkage

- Single Linkage
- Complete Linkage
- Average Linkage
- Centroid Linkage

- Single Linkage Clustering, the distance between two clusters is the minimum distance between members of the two clusters.
- Complete Linkage, the distance between two clusters is the maximum distance between members of the two clusters.
- Average Linkage, the distance between two clusters is the average of all distances between members of the two clusters.
- Centroid Linkage, the distance between two clusters is the distance between their centroids.

Single linkage



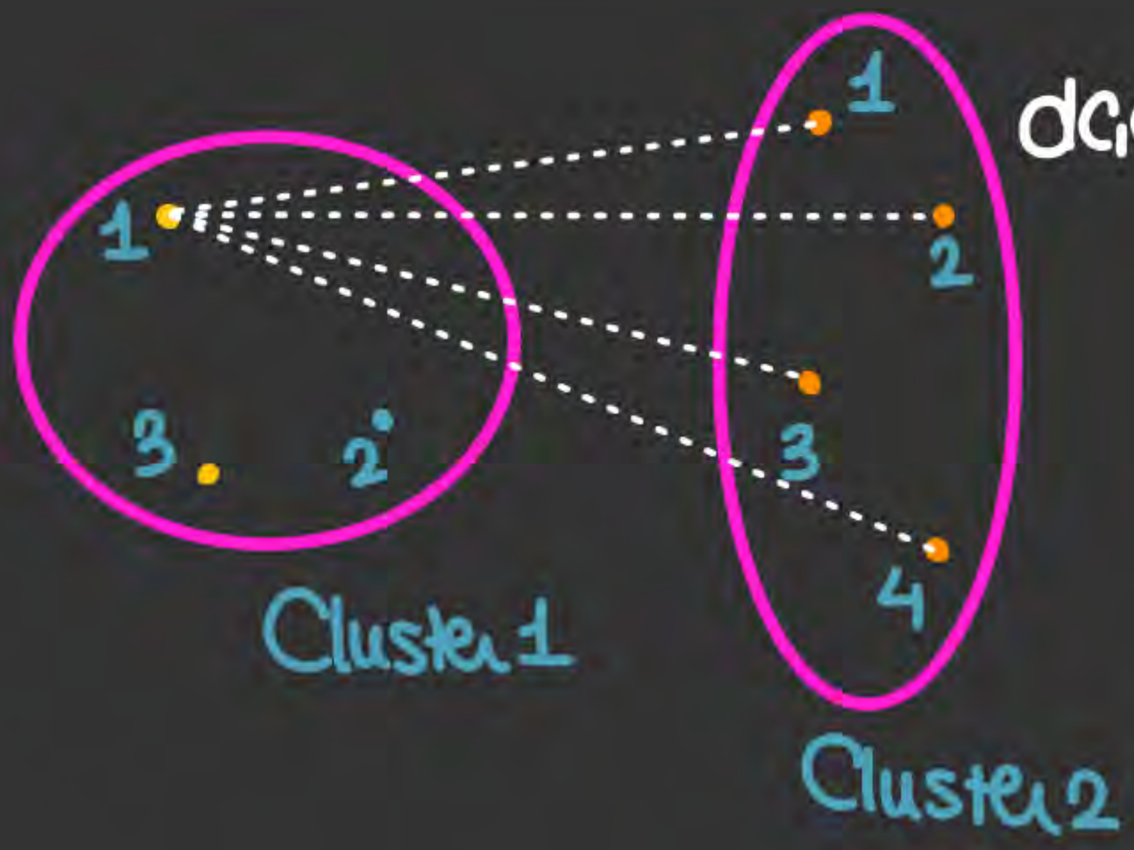
we measure distance of all points of cluster 1 to Cluster 2.

$$d_{c_1c_2} = \min \begin{bmatrix} d_{11} & d_{12} & d_{13} & d_{14} \\ d_{21} & d_{22} & d_{23} & d_{24} \\ d_{31} & d_{32} & d_{33} & d_{34} \end{bmatrix}$$

12 distances.

- So in case of Single linkage then distance b/w Clusters = min distance b/w points of Clusters

Complete linkage



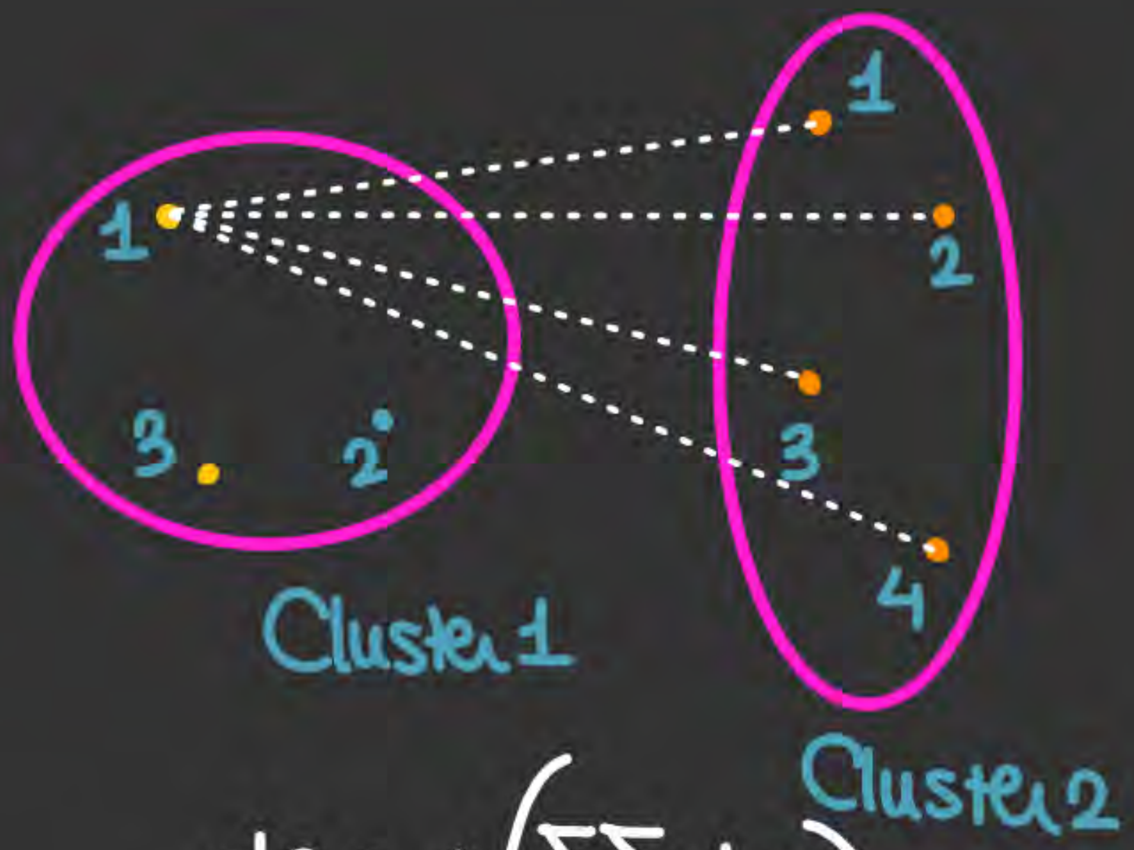
we measure distance of all points of cluster 1 to cluster 2.

$$d_{C_1C_2} = \max \overbrace{\begin{bmatrix} d_{11} & d_{12} & d_{13} & d_{14} \\ d_{21} & d_{22} & d_{23} & d_{24} \\ d_{31} & d_{32} & d_{33} & d_{34} \end{bmatrix}}$$

12 distances.

- So in case of Complete linkage then distance b/w Clusters = max. distance b/w points of Clusters

Avg linkage



$$d_{c_1 c_2} = \left(\frac{\sum\sum d_{ij}}{12} \right)$$

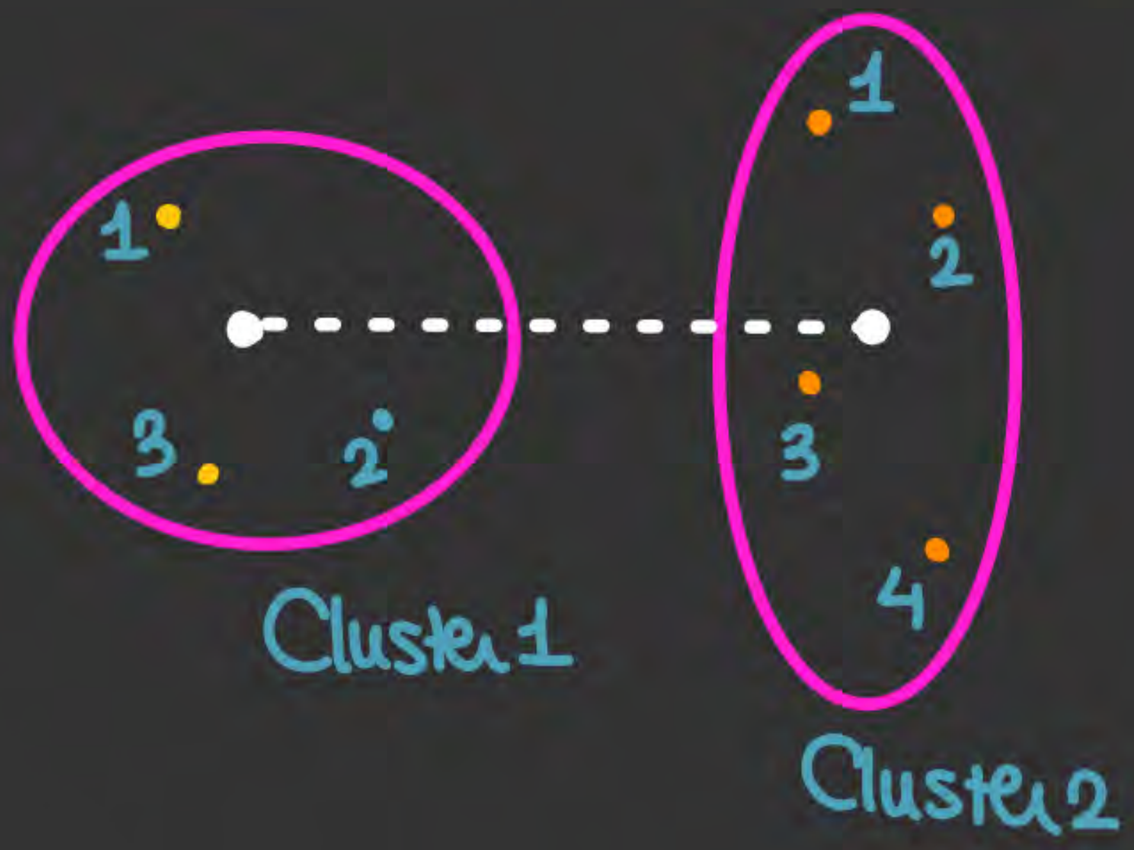
we measune distance of all points of cluster 1 to Cluste2.

$$\begin{bmatrix} d_{11} & d_{12} & d_{13} & d_{14} \\ d_{21} & d_{22} & d_{23} & d_{24} \\ d_{31} & d_{32} & d_{33} & d_{34} \end{bmatrix}$$

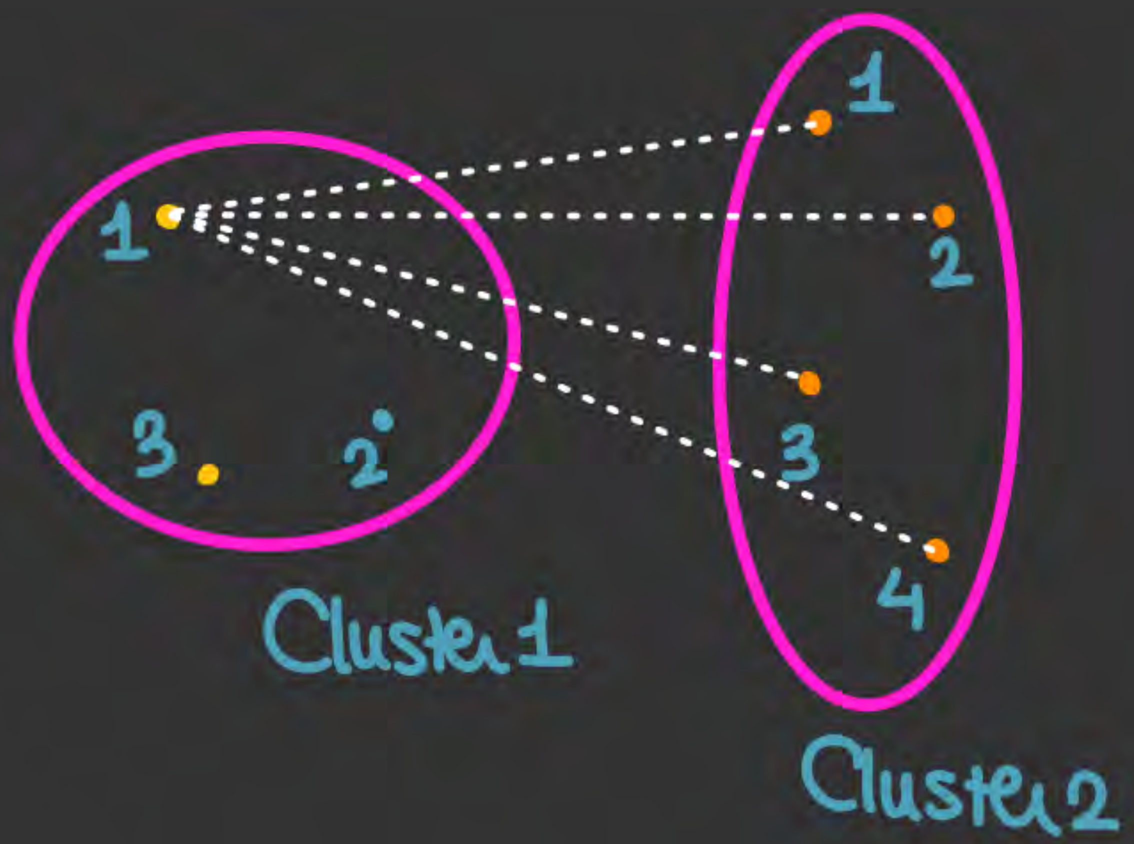
12 distances.

- So in case of avg. linkage then distance b/w Clusters = avg distance b/w points of Clusters

Centroid Linkage

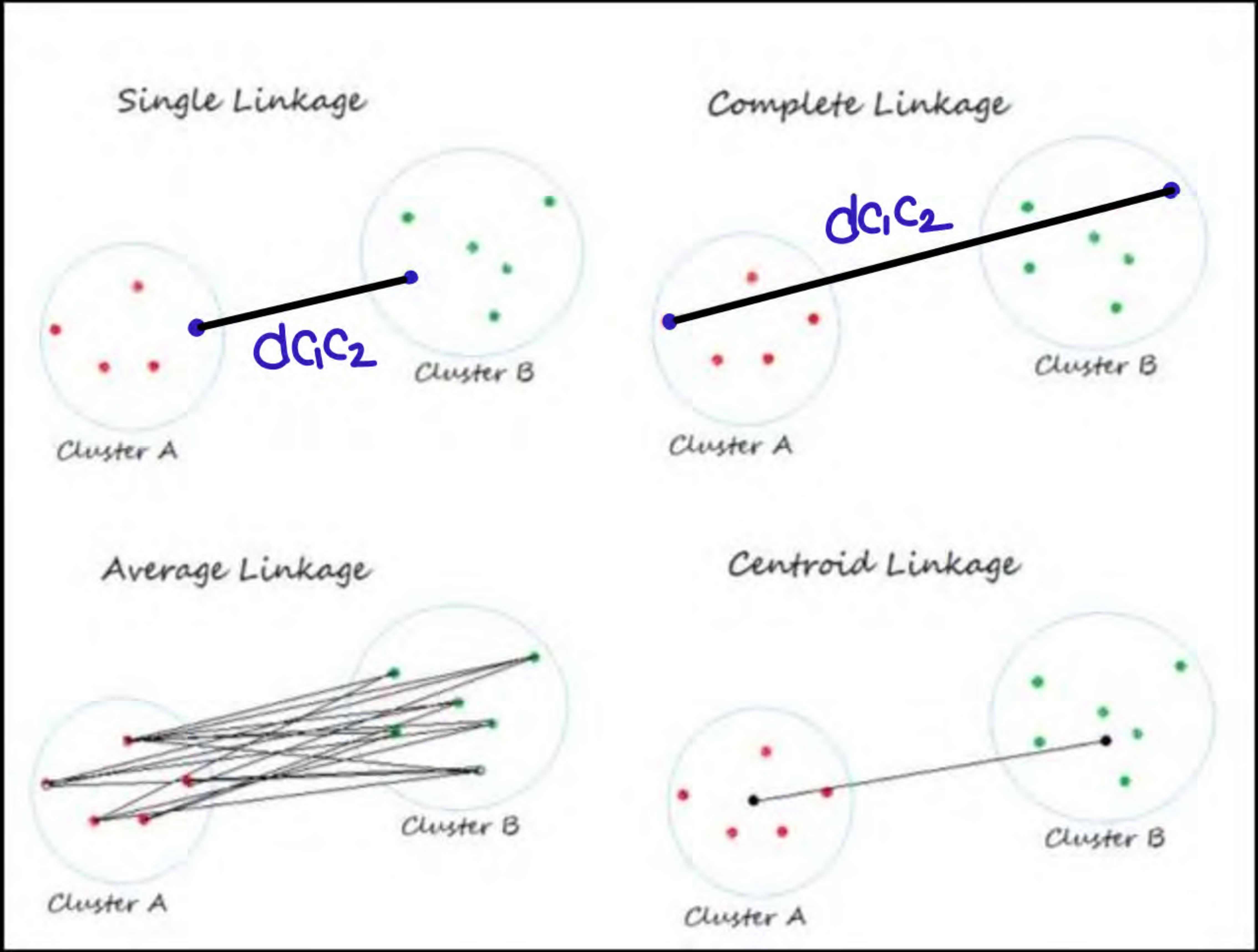


So $d_{C_1C_2} \Rightarrow$ distance b/w Centroids of two Clusters.



$$\begin{bmatrix} d_{11} & d_{12} & d_{13} & d_{14} \\ d_{21} & d_{22} & d_{23} & d_{24} \\ d_{31} & d_{32} & d_{33} & d_{34} \end{bmatrix}$$

12 distances.



Single Linkage
dc_1c_2
Cluster A
Cluster B
Complete Linkage
dc_1c_2
Cluster A
Cluster B
Average Linkage
Cluster A
Cluster B
Centroid Linkage
Cluster A
Cluster B

How we find the group linkage/ distance between groups ?

Problem in Single Linkage : Chaining...

- **Characteristics**
- **Tends to produce long, "loose" clusters** that may be less compact.
- **Sensitive to noise** and outliers.
- Can create chaining effects, where clusters are elongated.
- **Chain Effect**: Complete linkage can suffer from the chaining phenomenon, where clusters that are close together are merged, even if they should not be, resulting in elongated and less meaningful clusters.

Complete linkage

- While complete linkage has its advantages, such as producing compact clusters, it also has several potential issues:
- **Sensitivity to Outliers**: Since complete linkage uses the maximum distance between points, it is highly sensitive to outliers. A single outlier can significantly affect the distance calculation and, consequently, the clustering results.
- Cluster Shape: Complete linkage tends to produce clusters of roughly equal size and shape, which may not be appropriate for all datasets. If the data has clusters of varying shapes and sizes, complete linkage might not capture the true structure of the data.
- **Computational Complexity**: Hierarchical clustering, in general, has high computational complexity. For large datasets, the distance calculations in complete linkage can be particularly time-consuming.
- **Scalability**: As the dataset grows, the memory and computational requirements increase significantly, making complete linkage less suitable for large datasets.

- Single linkage can result in long stringy clusters and “chaining” while complete linkage tends to make highly compact clusters

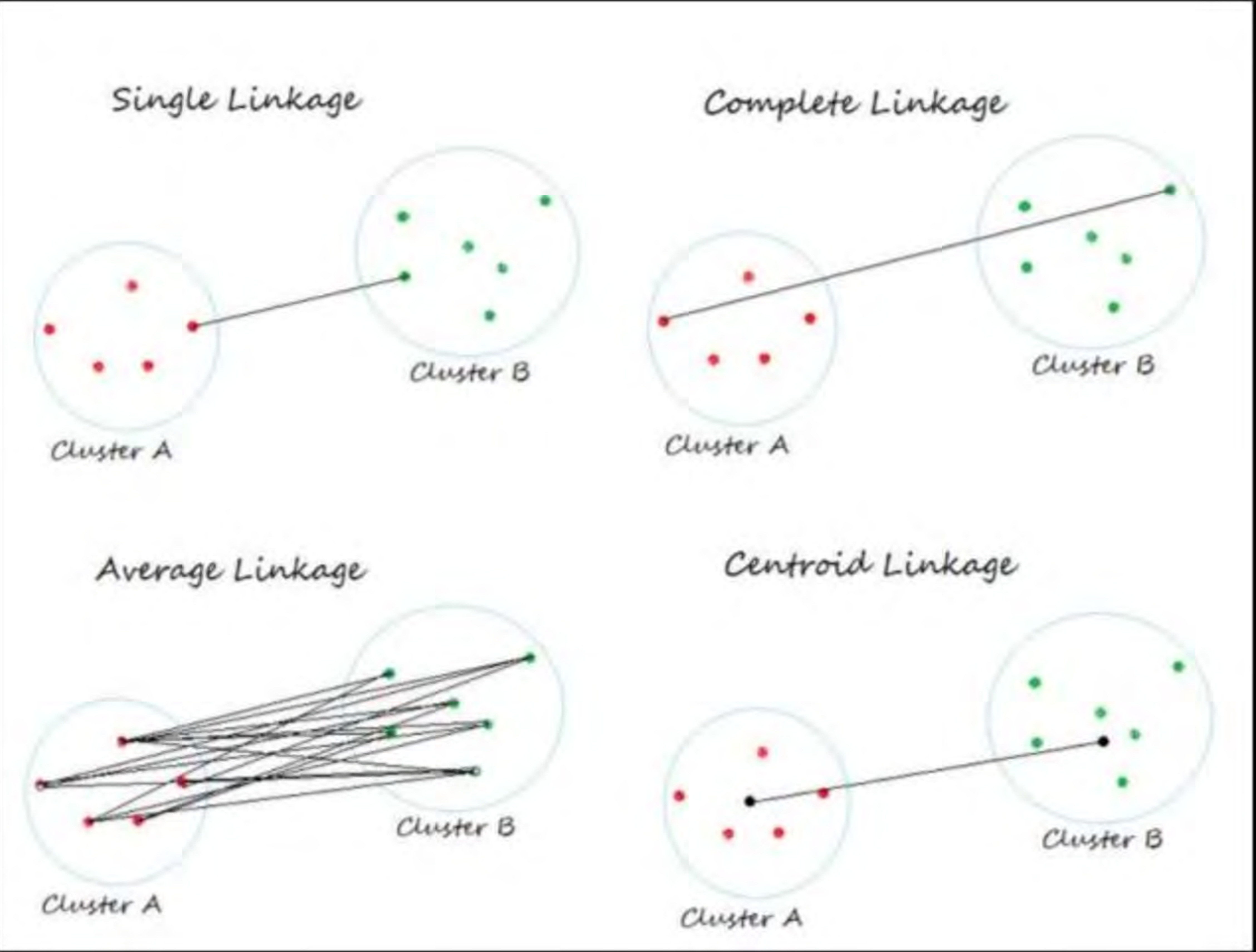
Clustering

How we find the group linkage/ distance between groups ?

Average Linkage

Solve the problem

Point	Coordinates (x, y)
A	(1, 2)
B	(2, 2)
C	(5, 5)
D	(6, 6)
E	(8, 8)


Single Linkage
Cluster A
Cluster B
Complete Linkage
Cluster A
Cluster B
Average Linkage
Cluster A
Cluster B
Centroid Linkage
Cluster A
Cluster B

- **Advantages of Agglomerative Clustering**
- **Versatility**: Can be used with various types of distance metrics and linkage criteria, making it adaptable to different types of data and clustering goals.
- **Hierarchy**: Produces a hierarchy of clusters, allowing the examination of data at different levels of granularity.
- **Intuitive Visualization**: The dendrogram provides a clear and interpretable visualization of the clustering process.
- **Disadvantages of Agglomerative Clustering**
- **Computational Complexity**: The algorithm can be computationally intensive, especially for large datasets, as it requires calculating and updating a distance matrix.
- **Sensitivity to Noise and Outliers**: Can be affected by noise and outliers, which may lead to less meaningful clusters.
- **Choice of Linkage and Distance Metric**: The results can vary significantly depending on the chosen linkage criteria and distance metric, which may require experimentation and domain knowledge to select appropriately.

Linkage Method	Description	Advantages	Disadvantages	Best Used For
Single Linkage	Minimum distance between points in the clusters	Tends to find long, chain-like clusters	Sensitive to noise and outliers, can produce chaining effect	Clusters with elongated shapes
Complete Linkage	Maximum distance between points in the clusters	Produces compact, spherical clusters	Sensitive to outliers, can create tightly packed clusters regardless of actual data structure	Clusters of similar size and shape, when compact clusters are desired
Average Linkage	Average distance between all points in the clusters	Balances between single and complete linkage	May not perform well if clusters are of different sizes or densities	Clusters with moderate structure, balance between compactness and separation
Centroid Linkage	Distance between centroids of the clusters	Takes into account the overall geometry of the cluster	Can produce clusters with centroids that are not part of the original data	Clusters where centroids are meaningful

How to find the best k

The distance in the dendogram show the dissimilarity between the clusters …

Divisive Clustering

The main idea behind this is ….

It is simply the iterative application of flat clustering

THANK - YOU