Data Science and Artificial Intelligence

Machine Learning

Unsupervised learning

Lecture No.2

By- SIDDHARTH SABHARWAL SIR

Recap of Previous Lecture

- Topic (Kmeans)
- Topic
- Topic
- Topic
- Topic

Turn on Slide map

Topics to be Covered

- Topic — Kmeans
- Topic — Kmedoid
- Topic — Hierarchical
- Topic
- Topic


update
"WE CANNOT BECOME WHAT WE WANT BY REMAINING WHAT WE ARE"
-Max Depree

REVISION

K-Means Clustering

→ So we decide K ⇒ No of clusters

→ Step1: Randomly initialise the K means $\mu_1 -- \mu_K$

Step2: • using these mean and training data create K Clusters,

• $\mu_i^{new} = \frac{\text{Sum of Points in cluster}}{\text{No of " " "}}$

• new K means 1st iteration end.

REVISION

K-Means Clustering

Step 3 use the K means obtained after 1st iteration and again use training data to create clusters

new clusters

$u_i^{new} \rightarrow$

new means, 2nd iteration ends.

- in any j^{th} iteration we use mean values obtained in pre-vious iteration and we get new K means

$$\sum_{i=1}^{K} \|u_i^{new} - u_i^{Previous}\|^2 < \delta$$

Stop.

- In Kmean clustering
  Our motive is to create
  Clusters such that distance distance of all points
  within cluster is min with the centroids of it's
  Cluster

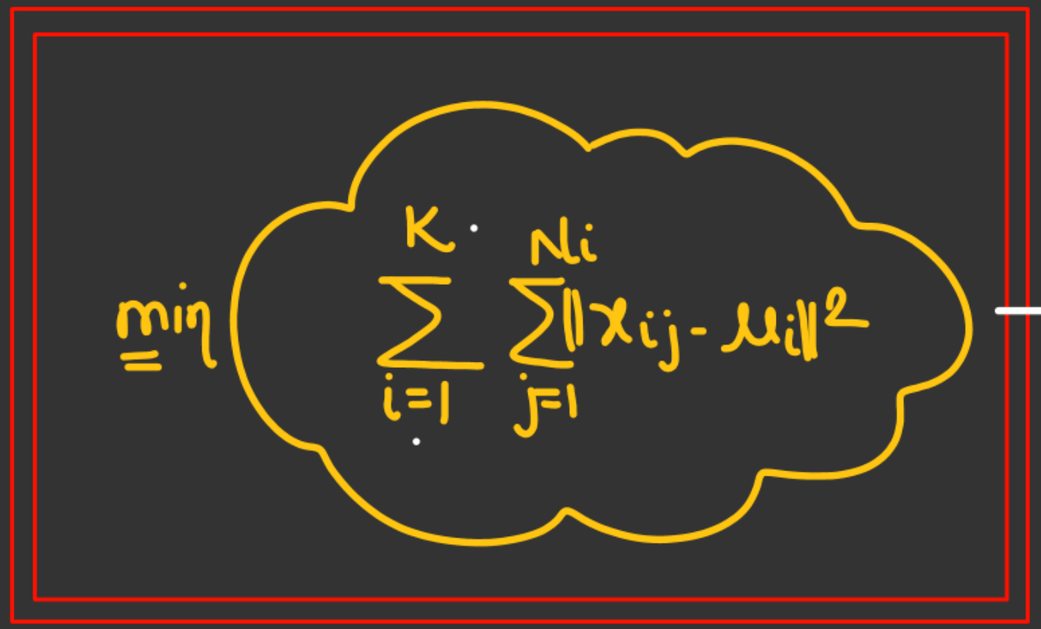
$\|x_i - \mu_i\|^2$ → eucledian distance of point from its cluster's centroid.

$$\sum_{j=1}^{N_i} \|x_{ij} - \mu_i\|^2$$

N_i : No of points in i^{th} cluster.

Where x_i are the points inside i^{th} cluster

So we add Eucledian distance of all points inside i^{th} cluster with its centroid



→ using this we will create the closely Packed clusters bcoz we are minimizing distance of point from Centroid.

Clustering

Lets see an example

Given a set of observations (x1, x2, ..., xn), where each observation is a D-dimensional real vector, k-means clustering aims to partition the n observations into k (≤ n) sets S = {S1, S2, ..., Sk} so as to minimize the within-cluster sum of squares (WCSS) (i.e. variance). This is sum of squared distance within a cluster

$$\arg\min_{\mathbf{S}} \sum_{i=1}^{k} \sum_{\mathbf{x} \in S_i} \|\mathbf{x} - \boldsymbol{\mu}_i\|^2$$

where $\boldsymbol{\mu}_i$ is the mean (also called centroid) of points in S_i, i.e.

$$\boldsymbol{\mu}_i = \frac{1}{|S_i|} \sum_{\mathbf{x} \in S_i} \mathbf{x},$$

K-Means Clustering

- The objective function in k-means is the WCSS (within cluster sum of squares).
- After each iteration, the WCSS decreases and so we have a nonnegative monotonically decreasing sequence.
- This guarantees that the k-means **always converges**, but not necessarily to the global optimum.

bcoz we start with random K-means.

Clustering

Objective of K Means clustering

- Grouping similar data points: K-means aims to identify patterns in your data by grouping data points that share similar characteristics together. This allows you to discover underlying structures within the data.
- ✔ Minimizing within-cluster distance: The algorithm strives to make sure data points within a cluster are as close as possible to each other, as measured by a distance metric (usually Euclidean distance). This ensures tight-knit clusters with high cohesiveness.
- ✔ Maximizing between-cluster distance: Conversely, k-means also tries to maximize the separation between clusters. Ideally, data points from different clusters should be far apart, making the clusters distinct from each other.

Clustering

Objective of K Means clustering

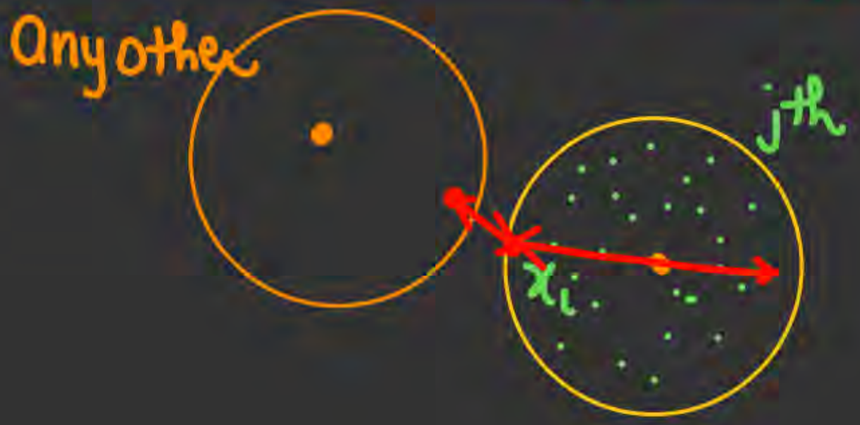
- Inter-cluster distance : we want distances between clusters to be as large as possible. Intra-cluster distance : we want distances between points within a cluster to be as small as possible.

→ Close Packed clusters bcoz we assume that Similar points are Close to each other.

• and in cluster we only keep Similar Points

In K mean clustering for any point x_i
within j^{th} cluster, the distance b/w x_i and all points
within j^{th} cluster < the " " " and any point of
any other cluster

(false)



Clustering

Advantages and Limitations

- Advantages:
- Simplicity: K-means is easy to understand and implement.
- Scalability: It can handle large datasets efficiently.
- Speed: The algorithm is computationally efficient, especially with K-means++ initialization.

Clustering

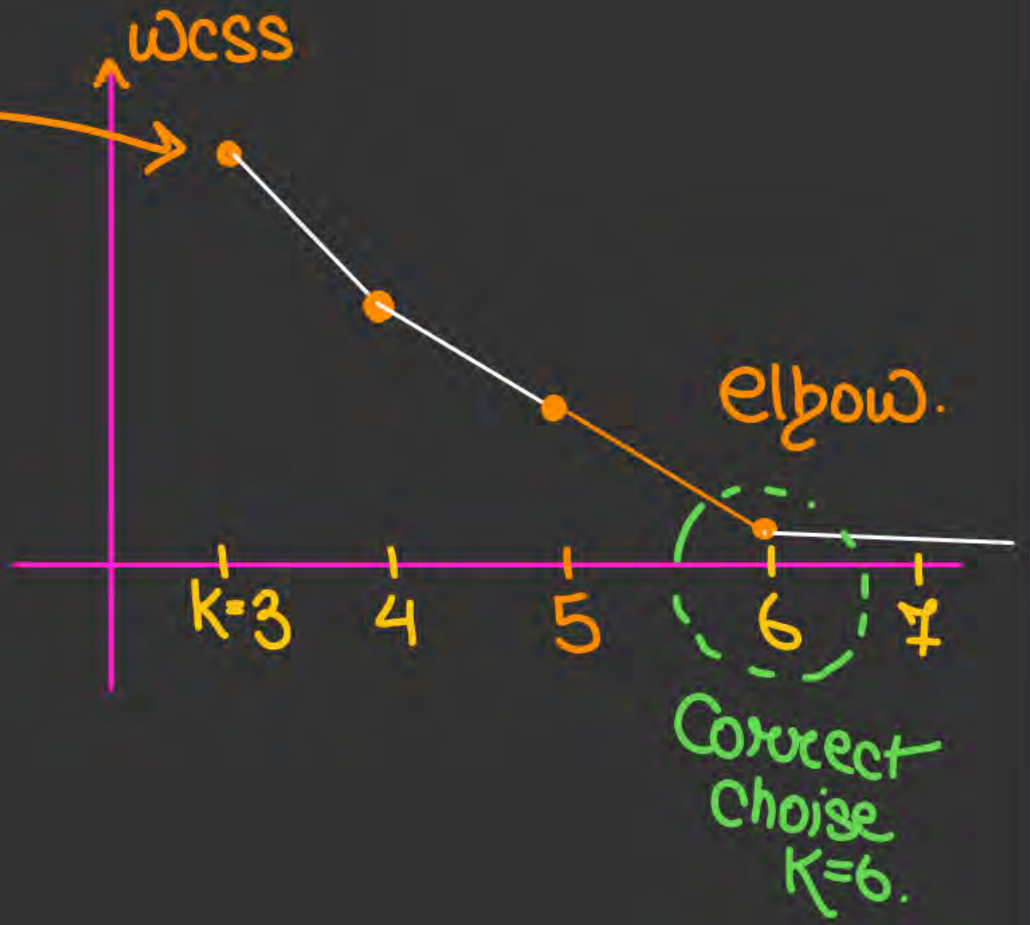
Advantages and Limitations

- Limitations:
- Predefined K: The number of clusters K must be specified in advance.
- Initial Centroid Sensitivity: The final clustering can depend on the initial placement of centroids, potentially leading to different results.
- Cluster Assumption: K-means assumes clusters are spherical and equally sized, which may not always be the case.
- Outliers: Sensitive to outliers and noise in the data.

↳ booz outlier shift Centroid.

we want

$$\min \left(\text{wcss} \right) \sum_{i=1}^{K} \sum_{j=1}^{N_i} \|x_{ij} - \mu_i\|^2$$



Clustering

How to find the Best K ?

- Choosing the right number of clusters is crucial for effective clustering. Several methods are used to determine the optimal K:
- Elbow Method: Plot the sum of squared errors (SSE) for different values of K. Look for an "elbow" point where the SSE reduction slows down significantly. This point often represents the optimal number of clusters.
- Silhouette Analysis: Calculate the silhouette score for different values of K. The silhouette score measures how similar each point is to its own cluster compared to other clusters. Choose the K with the highest average silhouette score.

Clustering

Silhouette Score

- Silhouette Score
- The Silhouette Score is a metric used to evaluate the quality of clustering results. It measures how similar an object is to its own cluster (cohesion) compared to other clusters (separation). The score ranges from -1 to 1, where:
  - +1 indicates that the object is well clustered.
  - 0 indicates that the object is on or very close to the decision boundary between two neighboring clusters.
  - -1 indicates that the object is misclassified and assigned to the wrong cluster.

Clustering

Silhouette Score

we want SS to be large.

(we want Clusters to be Far away b shd be large)

$$\text{Silhouette Score} = \frac{b-a}{\max(a,b)}$$

↳ Value -1 to 1.

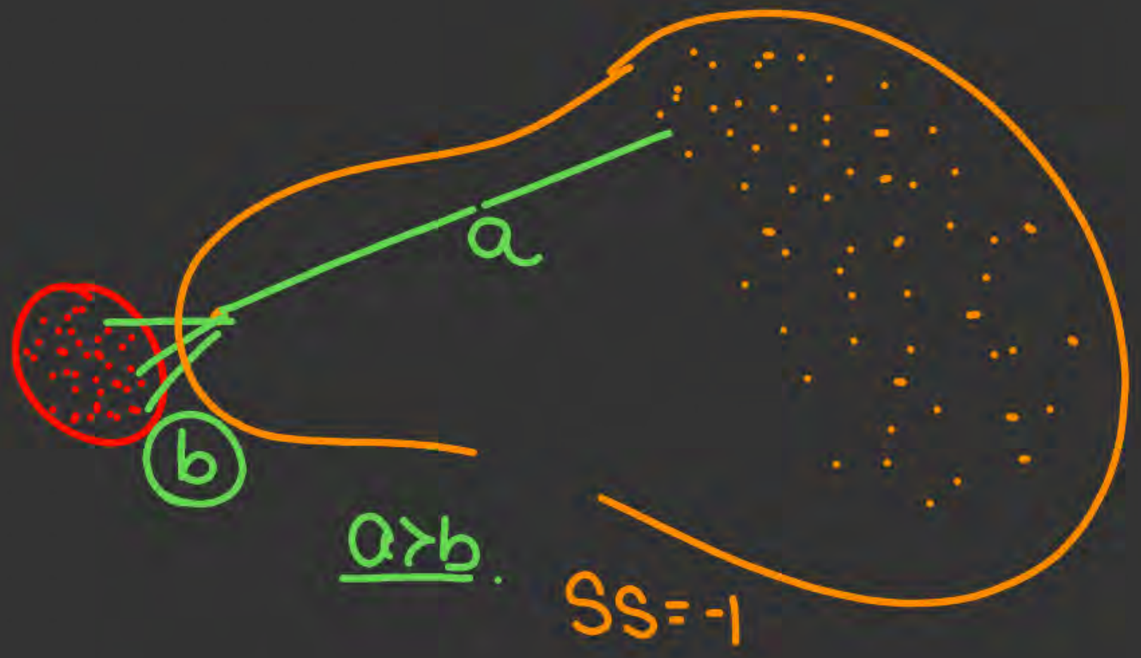
$$SS = \frac{b-a}{\max(a,b)}$$

- we need closely Packed cluster a shd be small

for best clustering SS ≈ 1.

Where:

- a is the mean distance between the sample and all other points in the same cluster.
- b is the mean distance between the sample and all points in the nearest cluster (the cluster that minimizes this mean distance). ✔.


a
b
a>b.
SS=-1


b=a
SS=0

Clustering

Silhouette Score

- **Steps to Compute Silhouette Score**
- **Calculate Cohesion (a)**: For each point i in a cluster A, calculate the average distance to all other points in the same cluster. This is the intra-cluster distance.
- **Calculate Separation (b)**: For each point i in a cluster A, calculate the average distance to all points in the nearest cluster B. This is the nearest-cluster distance.
- **Compute Silhouette Score for each point**: Use the formula given above.
- **Average Silhouette Score**: Calculate the average silhouette score for all points to get an overall assessment of clustering quality

Clustering

Silhouette Score

- Data Points and Cluster Assignments:
  - Cluster 1: [(2, 10), (2, 5), (5, 8)]
  - Cluster 2: [(8, 4), (7, 5)]

Like this we Calculate SS for all points

SS of whole data = $\frac{\text{Sum of SS of all points}}{\text{No of points}}$

⇒ SS for (2,10) ⇒ a = avg distance of (2,10) from within cluster points

$= \frac{\sqrt{5^2} + \sqrt{3^2+2^2}}{2} \Rightarrow 4.30$

b ⇒ avg distance of (2,10) from neighbour cluster

$\Rightarrow \frac{\sqrt{6^2+6^2} + \sqrt{5^2+5^2}}{2} \Rightarrow 7.77$

$SS = \frac{b-a}{\max(b,a)}$

$= .44$

- How to use SS to find K
  - Use a threshold SS > Threshold.
    - K = 3 Check SS
    - K = 4 " "
    - K = 5 " "
    - Select that K for which SS crosses threshold

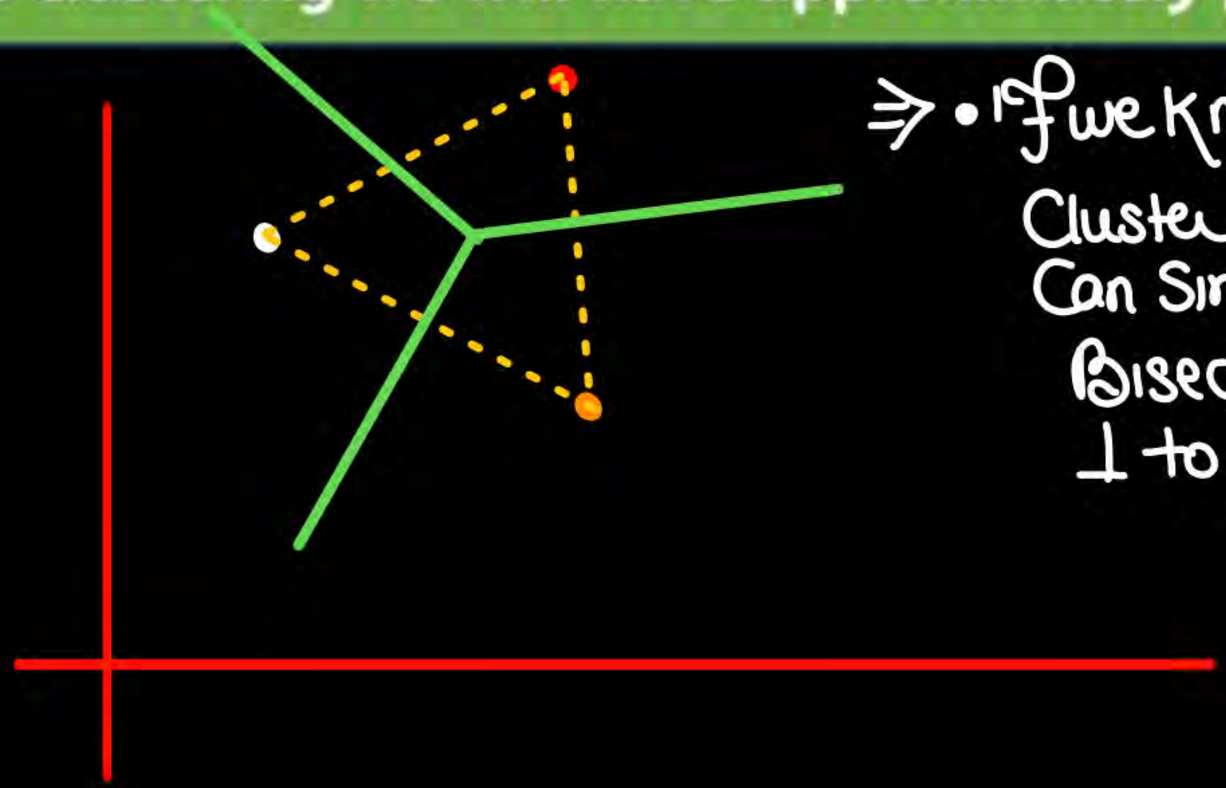
Clustering

Silhouette Score

- K Means clustering performs best data is well separated.
- When data points overlapped this clustering is not suitable.
- K Means is faster as compare to other clustering technique. It provides strong coupling between the data points.
- K Means cluster do not provide clear information regarding the quality of clusters.
- Different initial assignment of cluster centroid may lead to different clusters.
- Also, K Means algorithm is sensitive to noise. It may have stuck in local minima.

Clustering

In K means clustering we will have approximately perpendicular bisector



⇒ • If we know Centroid of Cluster then we Can Simply create the Bisector which are ⊥ to line joining the Centroids.

Clustering

Computational complexity

Q If N is No of data point
K is No of cluster
D " " " dimensions
t " " " iteration

@ each iteration we need to calculate distance of all points will all centroid
⇒ O(NKD)

Time Complexity

a) $O(N^2KDt)$
b) $O(NKD^2t)$
c) $O(NKDt)$
d) $O(NKDt^2)$

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?

Time Complexity of NN ⇒

a) O(No of Nodes)

b) O(No of weights)

c) O(No of weights x No of nodes)

d) O(No of weights^2 x No of nodes)

Clustering

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.
- The basic idea is to find representative objects (medoids) in the data set and form clusters around these medoids.

Clustering

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.

Clustering

Steps K medoid Algorithm

- Steps in K-medoids Clustering
- Initialization: Select kk initial medoids randomly from the dataset.
- Assignment: Assign each data point to the nearest medoid based on a chosen distance metric (e.g., Euclidean distance, Manhattan distance).
- 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.
- Repeat: Repeat the assignment and update steps until the medoids no longer change or the changes are very minimal.

Clustering

Example... (take K = 2)

Consider the following 2D points:

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

Clustering

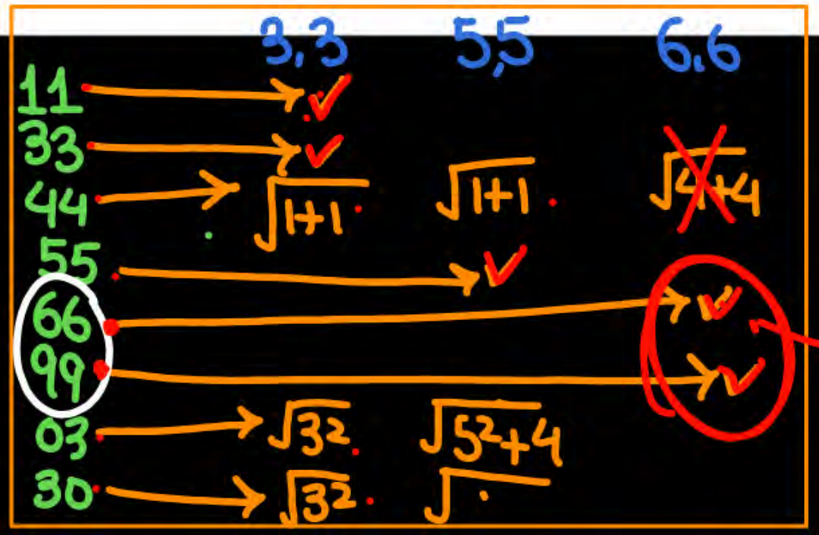
Advantage and Disadvantage of K Medoid

- 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

Examples

Given the dataset: (1, 1), (3, 3), (4, 4), (5, 5), (6, 6), (9, 9), (0, 3), (3, 0) and assuming the initial centroids for ($K = 3$ – means clustering) to be $C_1 = (3,3), C_2 = (5,5)$ and $C_3 = (6,6)$. One iteration of the Expectation Maximization Algorithm for K-means clustering, will update C_3 to (___, ___)



@end of 1st iteration
$C_3 \rightarrow (6,6)(9,9)$
$\rightarrow \mu \rightarrow 7.5, 7.5$

Clustering

Examples

Euclidean distance based k-means clustering algorithm was run on a dataset of 100 points with $k = 3$. If the points $\begin{bmatrix}1\\1\end{bmatrix}$ and $\begin{bmatrix}-1\\1\end{bmatrix}$ are both part of cluster 3, then which **ONE** of the following points is necessarily also part of cluster 3?

⇒ Cluster 3 ⇒ (1,1) , (-1,1)

- $\begin{bmatrix}0\\0\end{bmatrix}$ — (0,0): 1,1 → √2; -1,1 → √2
- $\begin{bmatrix}0\\2\end{bmatrix}$ — (0,2): 1,1 → √2; -1,1 → √2
- $\begin{bmatrix}2\\0\end{bmatrix}$ — (2,0): 1,1 → √2; -1,1 → $\sqrt{3^2+1}$
- $\begin{bmatrix}0\\1\end{bmatrix}$ — (0,1): (1,1) ⇒ 1; (-1,1) ⇒ 1

Clustering

Hierarchical Clustering

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

Here we create Trees...

Clustering

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

Steps in Agglomerative Clustering

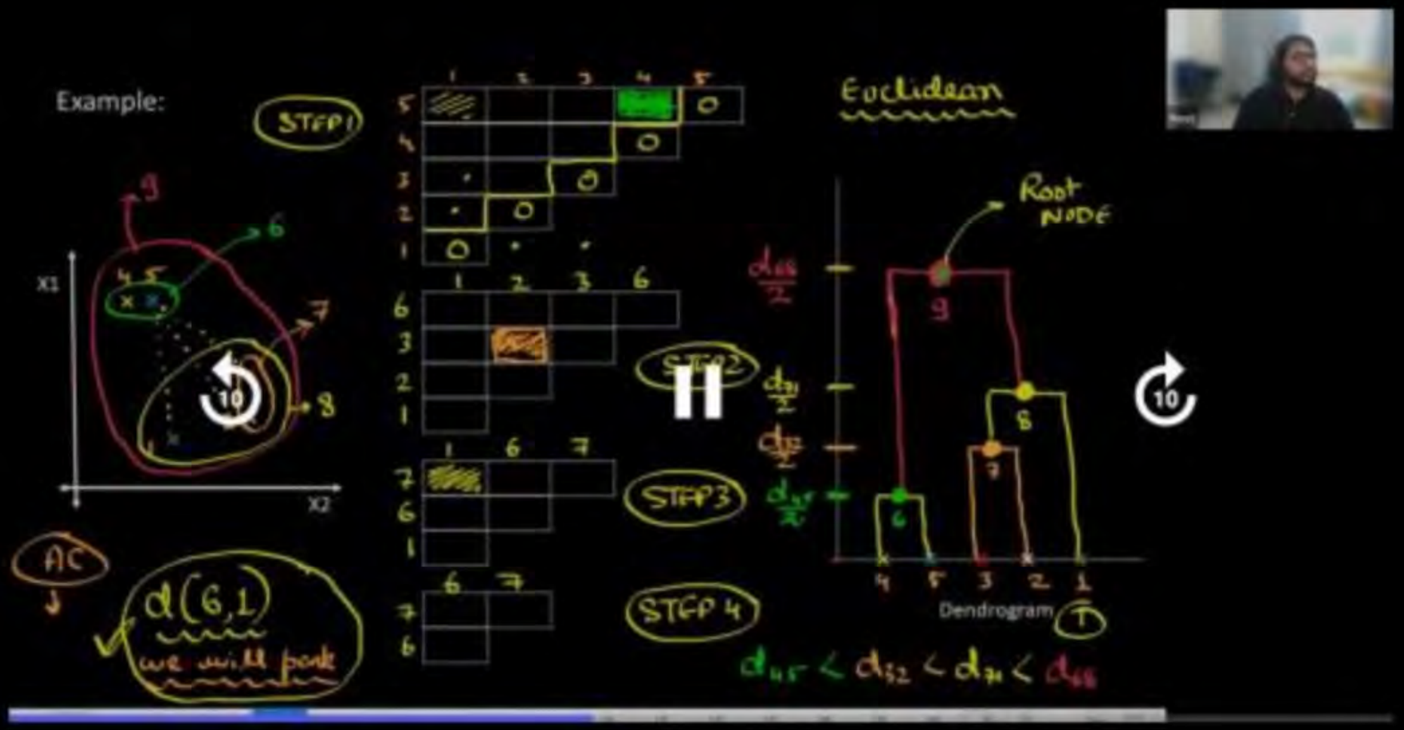
- **Initialization**:
  - Start with each data point as a single cluster. This results in NNN clusters for NNN data points.
- **Compute Distance Matrix**:
  - Calculate the distance matrix, which contains the distances between all pairs of data points.
- **Merge Clusters**:
  - Identify the pair of clusters 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.

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

Clustering

Hierarchical Clustering : Bottom Up Approach

Lets see an example…



Clustering

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.

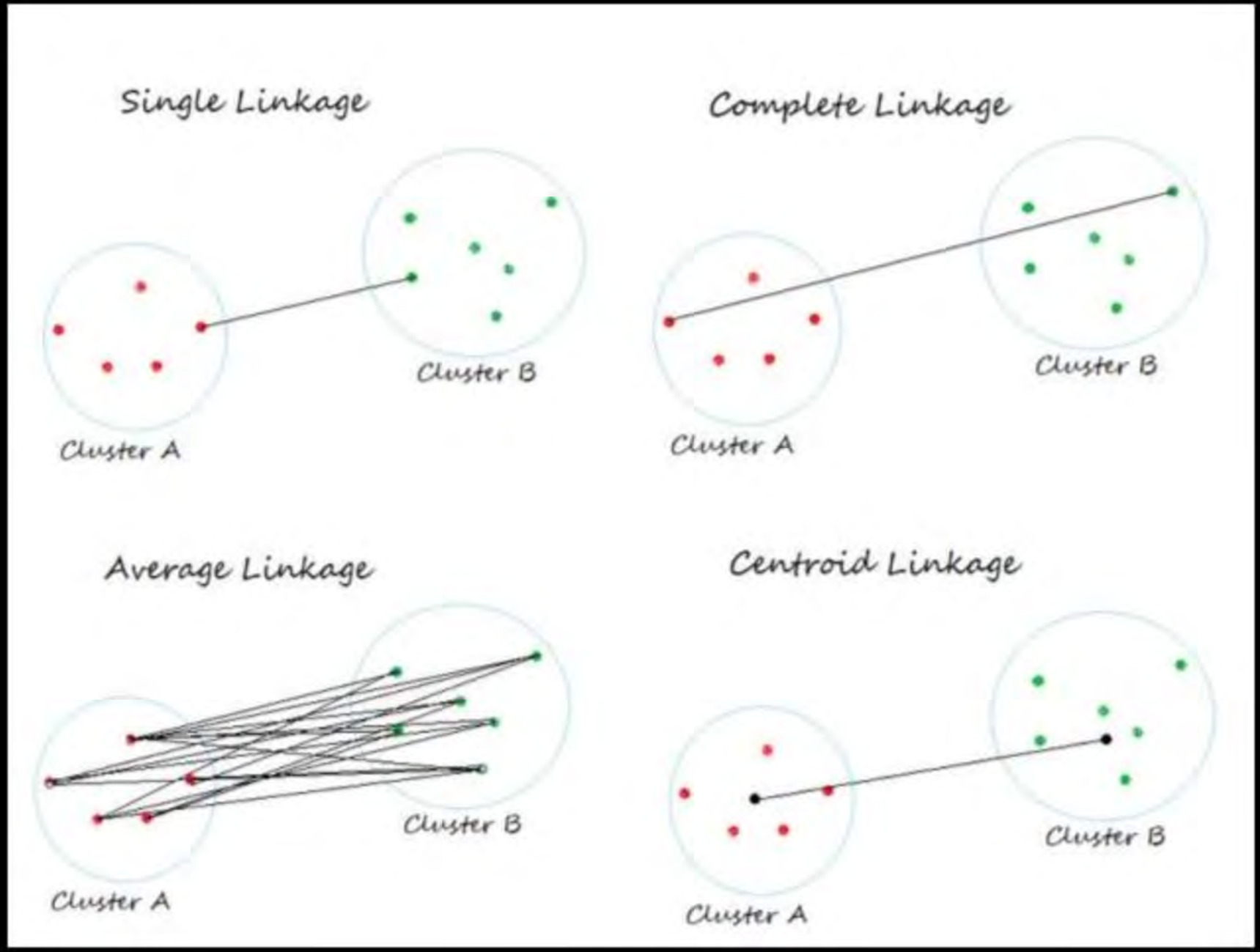
Clustering

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

Clustering

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.

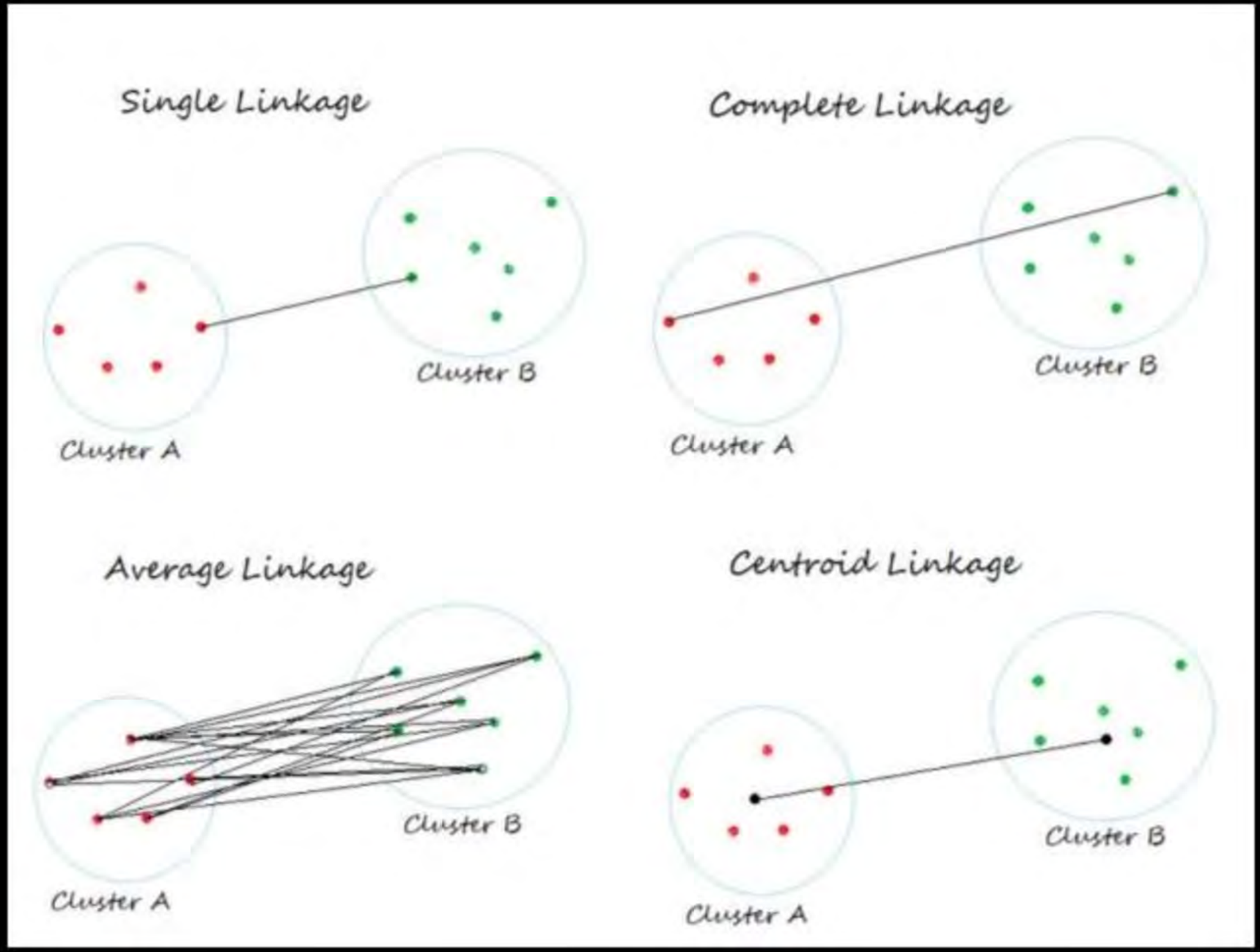
Clustering

How we find the group linkage/ distance between groups ?

- 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

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

Clustering

Divisive Clustering

The main idea behind this is ….

It is simply the iterative application of flat clustering

THANK - YOU