Dmv unit 4

# 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 for Data Understanding ar Applications

- Biology: taxonomy of living things: kingdom, phylum, class, order, family, genus and species
- Information retrieval: document clustering
- Land use: Identification of areas of similar land use in an earth observation database
- Marketing: Help marketers discover distinct groups in their customer bases, and then use this knowledge to develop targeted marketing programs
- 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
- Climate: understanding earth climate, find patterns of atmospheric and ocean
- Economic Science: market resarch

## Clustering as a Preprocessing Tool (Utility

- Summarization:
  - Preprocessing for regression, PCA, classification, and association analysis
- Compression:
  - Image processing: vector quantization
- 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 intra-class similarity: cohesive within clusters
  - low inter-class similarity: distinctive 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 hidden patterns

## Measure the Quality of Clustering

- Dissimilarity/Similarity metric
  - Similarity is expressed in terms of a distance function, typically metric: $d(i, j)$
  - The definitions of distance functions are usually rather different for interval-scaled, boolean, categorical, ordinal ratio, and vector variables
  - Weights should be associated with different variables based on applications and data semantics
- Quality of clustering:
  - There is usually a separate "quality" function that measures the "goodness" of a cluster.
  - It is hard to define "similar enough" or "good enough"
    - The answer is typically highly subjective

## Considerations for Cluster Analysis

- Partitioning criteria
  - Single level vs. hierarchical partitioning (often, multi-level hierarchical partitioning is desirable)
- Separation of clusters
  - Exclusive (e.g., one customer belongs to only one region) vs. non-exclusive (e.g., one document may belong to more than one class)
- Similarity measure
  - Distance-based (e.g., Euclidian, road network, vector) vs. connectivity-based (e.g., density or contiguity)
- Clustering space
  - Full space (often when low dimensional) vs. subspaces (often in high-dimensional clustering)

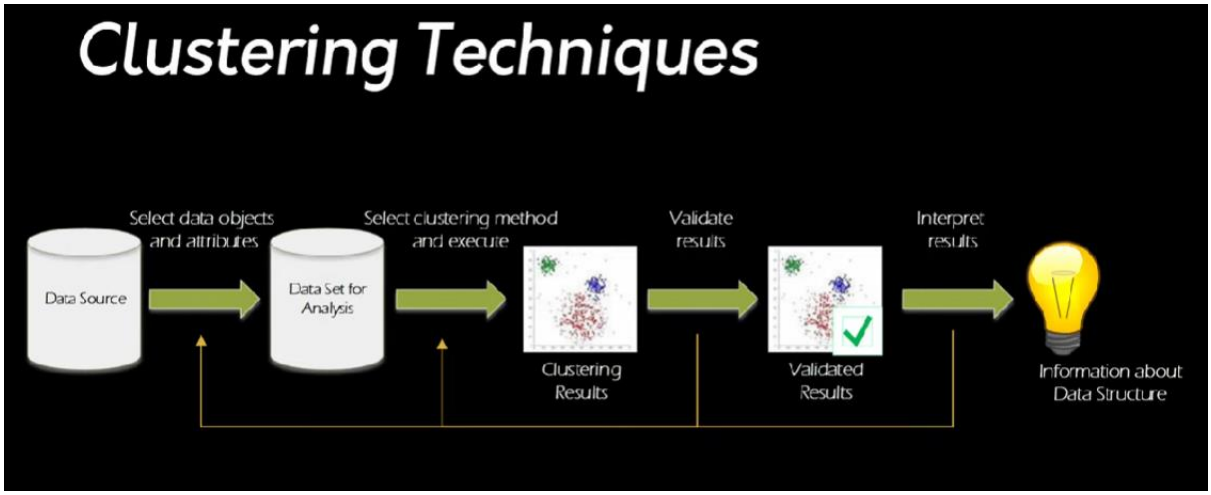
# Requirements and Challenges

- Scalability
  - Clustering all the data instead of only on samples
- Ability to deal with different types of attributes
  - Numerical, binary, categorical, ordinal, linked, and mixture of these
- Constraint-based clustering
  - User may give inputs on constraints
  - Use domain knowledge to determine input parameters
- Interpretability and usability
- Others
  - Discovery of clusters with arbitrary shape
  - Ability to deal with noisy data
  - Incremental clustering and insensitivity to input order
  - High dimensionality



# Major Clustering Approaches (I)

- Partitioning approach:
  - Construct various partitions and then evaluate them by some criterion, e.g., minimizing the sum of square errors
  - Typical methods: k-means, k-medoids, CLARANS
- Hierarchical approach:
  - Create a hierarchical decomposition of the set of data (or objects) using some criterion
  - Typical methods: Diana, Agnes, BIRCH, CAMELEON
- Density-based approach:
  - Based on connectivity and density functions
  - Typical methods: DBSACN, OPTICS, DenClue
- Grid-based approach:
  - based on a multiple-level granularity structure




## Clustering Methods:

The clustering methods can be classified into the following categories:



- Partitioning Method
- Hierarchical Method
- Density-based Method
- Grid-Based Method
- Model-Based Method
- Constraint-based Method

## There are 4 major categories of clustering algorithms:

1. Centroid-based Clustering
2. Density-based Clustering
3. Distribution-based Clustering
4. Hierarchical Clustering

# Major Clustering Approaches (II)

- Model-based:
  - A model is hypothesized for each of the clusters and tries to find the best fit of that model to each other
  - Typical methods: EM, SOM, COBWEB
- Frequent pattern-based:
  - Based on the analysis of frequent patterns
  - Typical methods: p-Cluster
- User-guided or constraint-based:
  - Clustering by considering user-specified or application-specific constraints
  - Typical methods: COD (obstacles), constrained clustering
- Link-based clustering:
  - Objects are often linked together in various ways
  - Massive links can be used to cluster objects: SimRank, LinkClus



# Partitioning Algorithms: Basic Concept

- Partitioning method: Partitioning a database ***D*** of ***n*** objects into a set of ***k*** clusters, such that the sum of squared distances is minimized (where $c_i$ is the centroid or medoid of cluster $C_i$)

$$E = \Sigma_{i=1}^{k} \Sigma_{p \in C_i} (p - c_i)^2$$

- Given *k*, find a partition of *k clusters* that optimizes the chosen partitioning criterion
  - Global optimal: exhaustively enumerate all partitions
  - Heuristic methods: *k-means* and *k-medoids* algorithms
  - *k-means* (MacQueen'67, Lloyd'57/'82): Each cluster is represented by the center of the cluster
  - *k-medoids* or PAM (Partition around medoids) (Kaufman & Rousseeuw'87): Each cluster is represented by one of the objects in the cluster

## Different Types of Centroid-based Clustering Algorithms

Centroid-based clustering has several variations, including:

1. K-Means Clustering - The most commonly used centroid-based clustering algorithm that minimizes the sum of the distances between the data points and their corresponding cluster centroids.
2. K-Medoids Clustering - A variation of k-means that uses medoids, actual data points, as the center of each cluster instead of centroids.
3. Fuzzy c-Means Clustering - A variation of k-means that allows data points to belong to more than one cluster, with varying degrees of membership.
4. Expectation Maximization (EM) Algorithm - A model-based clustering algorithm that uses a statistical model to define the relationships between the data points and clusters.

---

## Real-World Applications of Centroid-based Clustering for Partitioning Datasets

Centroid-based clustering has many real-world applications, including:

1. Image Segmentation - Dividing an image into multiple segments or regions based on color, texture, or other features using k-means or other centroid-based clustering algorithms.
2. Market Segmentation - Identifying smaller groups of consumers with similar needs or characteristics using k-means or other centroid-based clustering algorithms.
3. Customer Segmentation - Dividing a customer base into groups with similar characteristics using k-means or other centroid-based clustering algorithms.
4. Anomaly Detection - Identifying data points that are significantly different from the rest of the data using k-means or other centroid-based clustering algorithms.
5. Data Compression - Reducing the size of a dataset by replacing individual data points with their corresponding cluster centroids using k-means or other centroid-based clustering algorithms.

In the partitioning method when database(D) that contains multiple(N) objects then the partitioning method constructs user-specified(K)

partitions of the data in which each partition represents a cluster and a particular region. There are many algorithms that come under partitioning method some of the popular ones are K-Mean, PAM(K-Medoids), CLARA algorithm (Clustering Large Applications) etc. In this article, we will be seeing the working of K Mean algorithm in detail.

**K-Mean (A centroid based Technique):** The K means algorithm takes the input parameter K from the user and partitions the dataset containing N objects into K clusters so that resulting similarity among the data objects inside the group (intracluster) is high but the similarity of data objects with the data objects from outside the cluster is low (intercluster). The similarity of the cluster is determined with respect to the mean value of the cluster. It is a type of square error algorithm. At the start randomly k objects from the dataset are chosen in which each of the objects represents a cluster mean(centre). For the rest of the data objects, they are assigned to the nearest cluster based on their distance from the cluster mean. The new mean of each of the cluster is then calculated with the added data objects.

the cluster mean. The new mean of each of the cluster is then calculated with the added data objects. **Algorithm: K mean:**

```
Input:
K: The number of clusters in which the dataset has to be divided
D: A dataset containing N number of objects

Output:
A dataset of K clusters
```

**Method:**

1. Randomly assign K objects from the dataset(D) as cluster centres(C)
2. (Re) Assign each object to which object is most similar based upon mean values.
3. Update Cluster means, i.e., Recalculate the mean of each cluster with the updated values.
4. Repeat Step 2 until no change occurs.

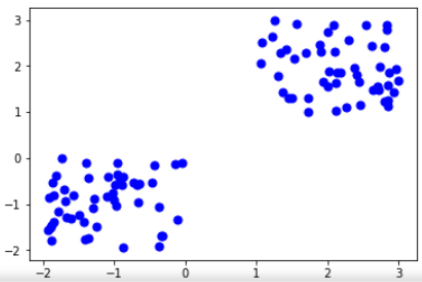
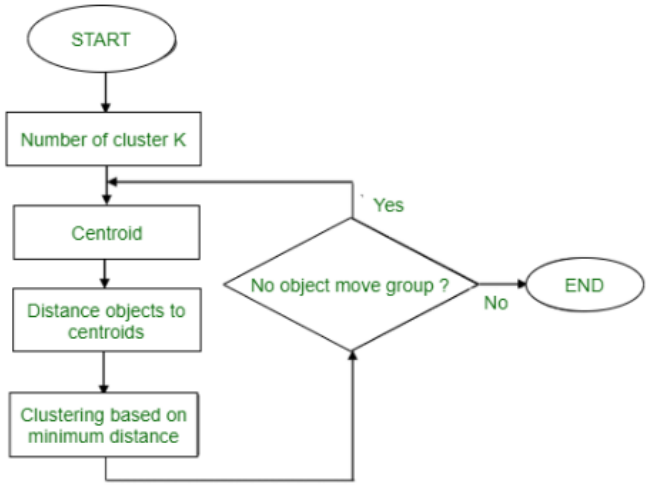


**Figure** – K-mean Clustering**Flowchart:**




# Example of k means clustering

```
Centroid(C1) = 16 [16]
Centroid(C2) = 22 [22]
```

**Note:** These two points are chosen randomly from the dataset. **Iteration-1:**

```
C1 = 16.33 [16, 16, 17]
C2 = 37.25 [20, 20, 21, 21, 22, 23, 29, 36, 41, 42, 43, 44, 45, 61, 62, 66]
```

**Iteration-2:**

```
C1 = 19.55 [16, 16, 17, 20, 20, 21, 21, 22, 23]
C2 = 46.90 [29, 36, 41, 42, 43, 44, 45, 61, 62, 66]
```

**Iteration-3:**

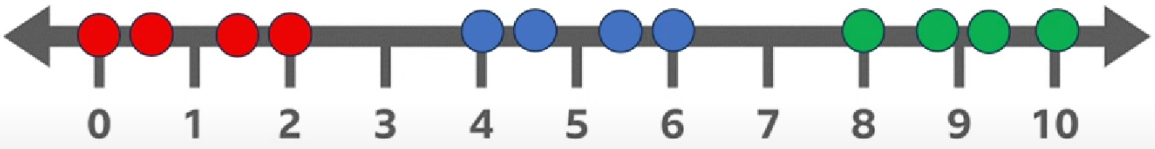
```
C1 = 20.50 [16, 16, 17, 20, 20, 21, 21, 22, 23, 29]
C2 = 48.89 [36, 41, 42, 43, 44, 45, 61, 62, 66]
```

**Iteration-4:**

```
C1 = 20.50 [16, 16, 17, 20, 20, 21, 21, 22, 23, 29]
C2 = 48.89 [36, 41, 42, 43, 44, 45, 61, 62, 66]
```

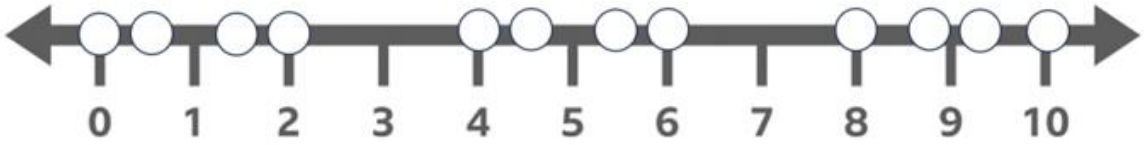
No change Between Iteration 3 and 4, so we stop. Therefore we get the clusters **(16-29)** and **(36-66)** as 2 clusters we get using K Mean Algorithm.

By looking at it we can divide it into these 3 groups

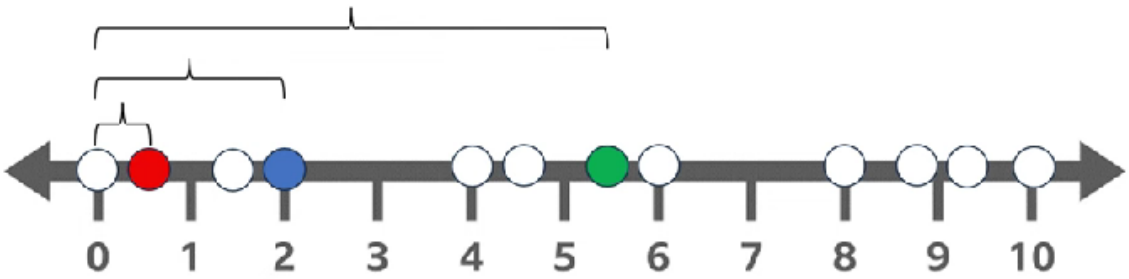



But computers can't see so what will it do?

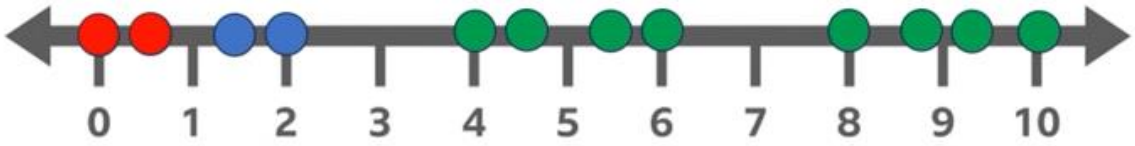
That's where we use K-Means Clustering



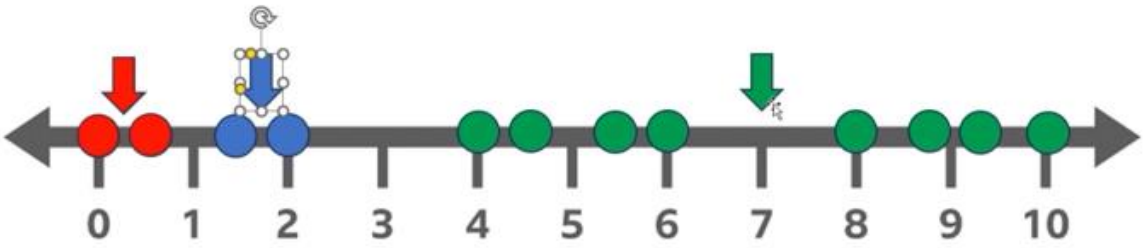

Step: 1 – Choose K value(number of clusters), say 3




**Step: 2 – Assign k(3) Centroids Randomly and calculate distance of all points from all 3 centroids**



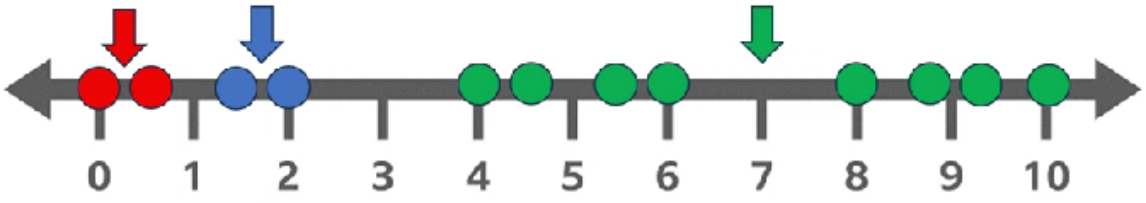

**Step: 3 – Assign data points to the centroids from which it has least distance**




**Step: 4 – Find new centroid for the 3 clusters by calculating mean of the values in the cluster**

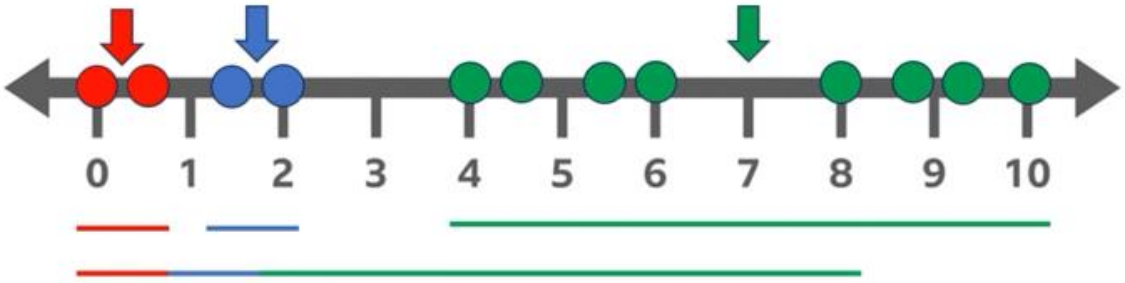
Phir ye mean value new centroids ho jaayenge

Aur use mean value calculate hogi




**Step: 5 – Repeat step 2 to 4 till centroid stops changing**

**Step: 6 – Calculate variance of the cluster**




then if the variance is high from the threshold value then it will again randomly assign the centroids

## K-Means Clustering Algorithm – Solved Example

- Use K Means clustering to cluster the following data into two groups.
- Data Points: { 2, 4, 10, 12, 3, 20, 30, 11, 25 }
- The distance function used is Euclidean distance.
- Initial cluster centroid are M1 = 4 and M2 = 11.

# K-Means Clustering Algorithm – Solved Example

Initial Centroids:
M1: 4
M2: 11

| Data Points | Distance to M1 | Distance to M2 | Cluster | New Cluster |
|---|---|---|---|---|
| 2 | 2 | 9 | C1 | |
| 4 | 0 | 7 | C1 | |
| 10 | 6 | 1 | C2 | |
| 12 | 8 | 1 | C2 | |
| 3 | 1 | 8 | C1 | |
| 20 | 16 | 9 | C2 | |
| 30 | 26 | 19 | C2 | |
| 11 | 7 | 0 | C2 | |
| 25 | 21 | 14 | C2 | |

$$d(x_2, x_1) = \sqrt{(x_2 - x_1)^2}$$

# K-Means Clustering Algorithm – Solved Example

Initial Centroids:
M1: 4
M2: 11

Therefore
C1= {2, 4, 3}
C2= {10, 12, 20, 30, 11, 25}

New Centroids:
M1: 3
M2: 18

| Data Points | Distance to M1 | Distance to M2 | Cluster | New Cluste |
|---|---|---|---|---|
| 2 | 2 | 9 | C1 | |
| 4 | 0 | 7 | C1 | |
| 10 | 6 | 1 | C2 | |
| 12 | 8 | 1 | C2 | |
| 3 | 1 | 8 | C1 | |
| 20 | 16 | 9 | C2 | |
| 30 | 26 | 19 | C2 | |
| 11 | 7 | 0 | C2 | |
| 25 | 21 | 14 | C2 | |

$$d(x_2, x_1) = \sqrt{(x_2 - x_1)^2}$$

## K-Means Clustering Algorithm – Solved Example

Current Centroids:
M1: 3
M2: 18

| Data Points | Distance to M1 | Distance to M2 | Cluster | New Cluster |
|---|---|---|---|---|
| 2 | 1 | 16 | C1 | C1 |
| 4 | 1 | 14 | C1 | C1 |
| 10 | 7 | 8 | C2 | C1 |
| 12 | 9 | 6 | C2 | C2 |
| 3 | 0 | 15 | C1 | C1 |
| 20 | 17 | 2 | C2 | C2 |
| 30 | 27 | 12 | C2 | C2 |
| 11 | 8 | 7 | C2 | C2 |
| 25 | 22 | 7 | C2 | C2 |

$$d(x_2, x_1) = \sqrt{(x_2 - x_1)^2}$$

If there is variation in new and old cluster the we calculate again the centroids

## K-Means Clustering Algorithm – Solved Example

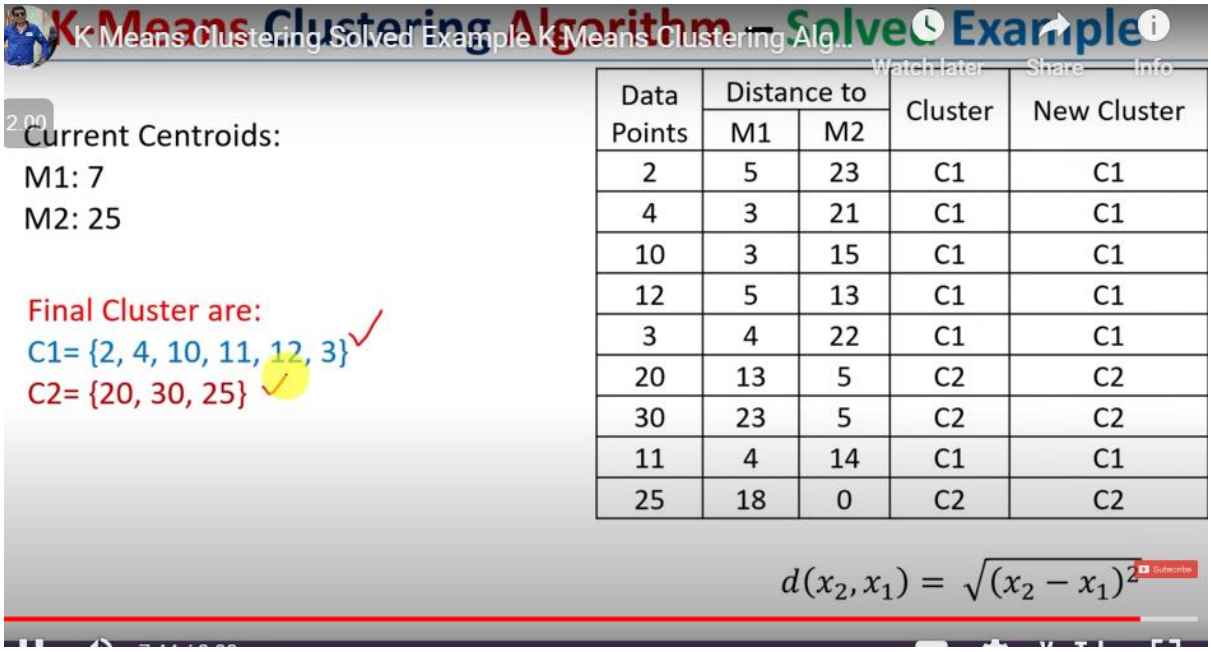
Current Centroids:
M1: 4.75
M2: 19.6

Therefore
C1= {2, 4, 10, 11, 12, 3}
C2= {20, 30, 25}

New Centroids:
M1: 7
M2: 25

| Data Points | Distance to M1 | Distance to M2 | Cluster | New Cluster |
|---|---|---|---|---|
| 2 | 2.75 | 17.6 | C1 | C1 |
| 4 | 0.75 | 15.6 | C1 | C1 |
| 10 | 5.25 | 9.6 | C1 | C1 |
| 12 | 7.25 | 7.6 | C2 | C1 |
| 3 | 1.75 | 16.6 | C1 | C1 |
| 20 | 15.25 | 0.4 | C2 | C2 |
| 30 | 25.25 | 10.4 | C2 | C2 |
| 11 | 6.25 | 8.6 | C2 | C1 |
| 25 | 20.25 | 5.4 | C2 | C2 |

$$d(x_2, x_1) = \sqrt{(x_2 - x_1)^2}$$



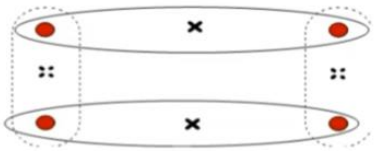

| Data Points | Distance to M1 | Distance to M2 | Cluster | New Cluster |
|---|---|---|---|---|
| 2 | 5 | 23 | C1 | C1 |
| 4 | 3 | 21 | C1 | C1 |
| 10 | 3 | 15 | C1 | C1 |
| 12 | 5 | 13 | C1 | C1 |
| 3 | 4 | 22 | C1 | C1 |
| 20 | 13 | 5 | C2 | C2 |
| 30 | 23 | 5 | C2 | C2 |
| 11 | 4 | 14 | C1 | C1 |
| 25 | 18 | 0 | C2 | C2 |

## Comments on the *K-Means* Method

- Strength: *Efficient*: $O(tkn)$, where $n$ is # objects, $k$ is # clusters, and $t$ is # iterations. Normally, $k, t << n$.
  - Comparing: PAM: $O(k(n-k)^2)$, CLARA: $O(ks^2 + k(n-k))$
- Comment: Often terminates at a *local optimal*.
- Weakness
  - Applicable only to objects in a continuous n-dimensional space
    - Using the k-modes method for categorical data
    - In comparison, k-medoids can be applied to a wide range of data
  - Need to specify $k$, the *number* of clusters, in advance (there are ways to automatically determine the best k (see Hastie et al., 2009)
  - Sensitive to noisy data and *outliers*

# Variations of the *K-Means* Method

- Most of the variants of the *k-means* which differ in
  - Selection of the initial *k* means
  - Dissimilarity calculations
  - Strategies to calculate cluster means

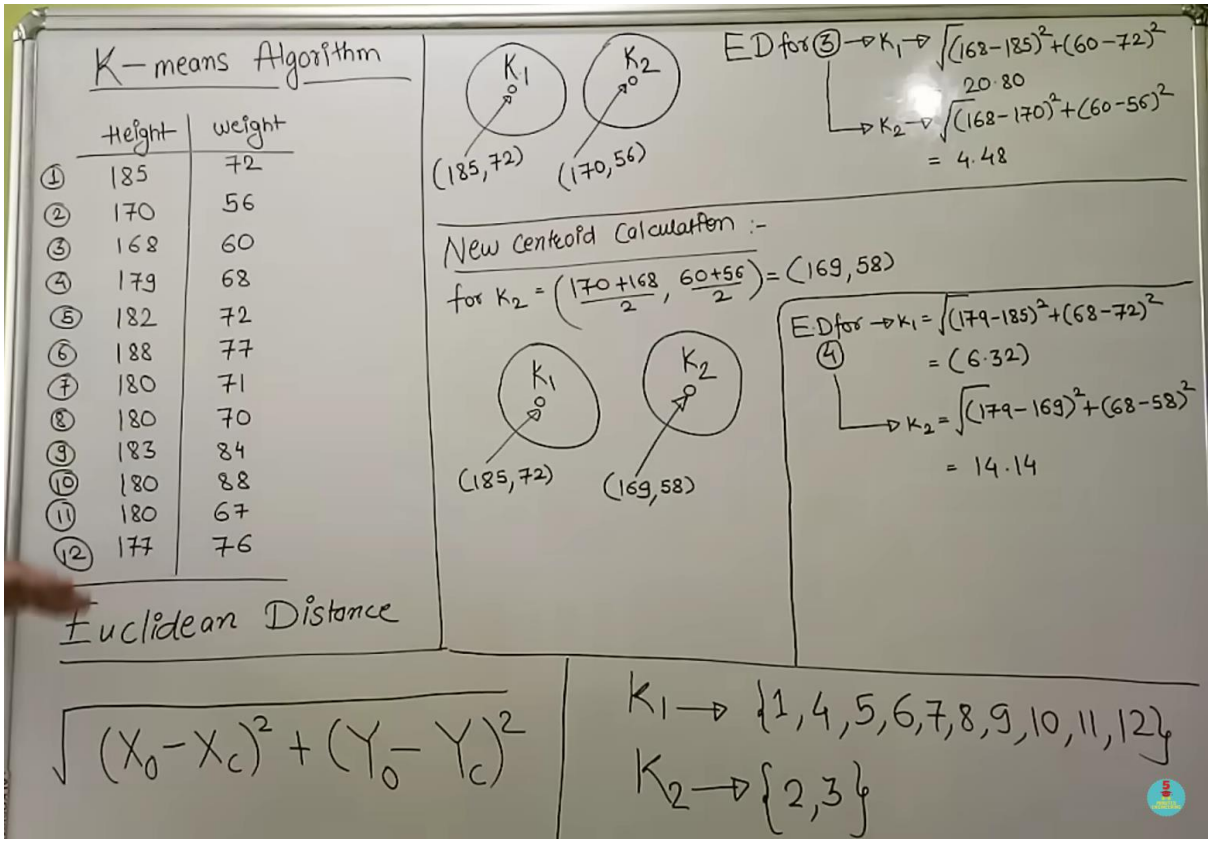


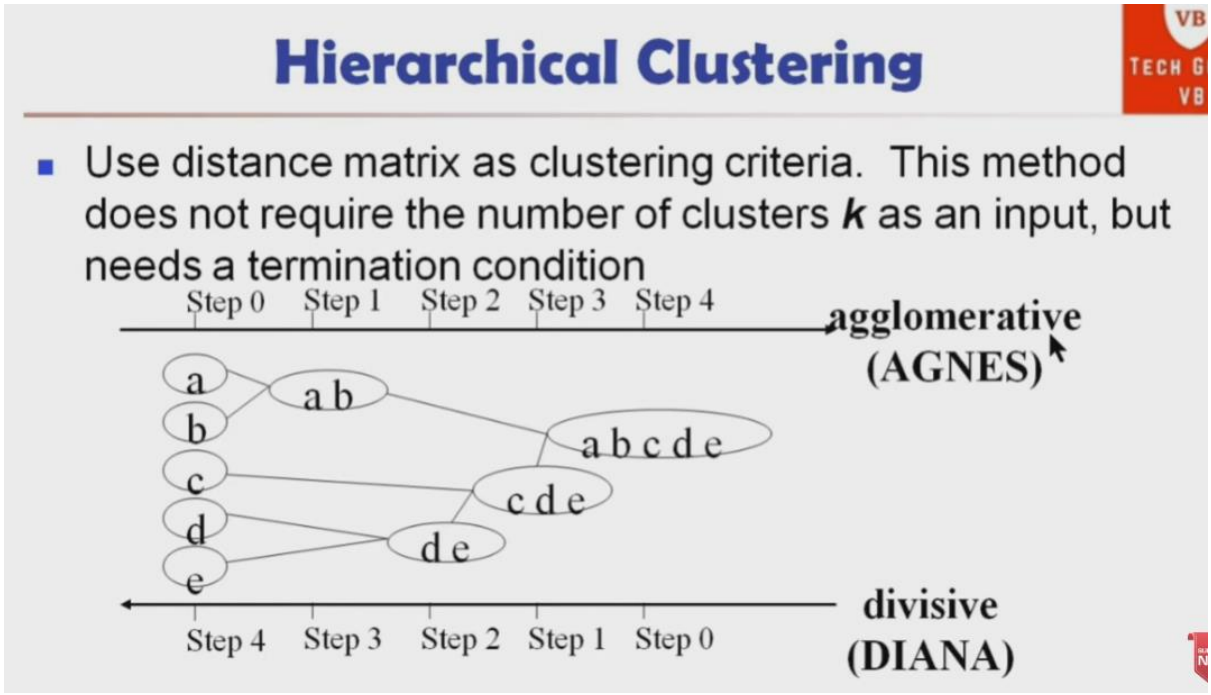
- Handling categorical data: *k-modes*
  - Replacing means of clusters with modes
  - Using new dissimilarity measures to deal with categorical objects
  - Using a frequency-based method to update modes of clusters
  - A mixture of categorical and numerical data: *k-prototype* method



# What Is the Problem of the K-Means Method?

- The k-means algorithm is sensitive to outliers !
  - Since an object with an extremely large value may substantially distort the distribution of the data
- K-Medoids: Instead of taking the **mean** value of the object in a cluster as a reference point, **medoids** can be used, which is the **most centrally located** object in a cluster

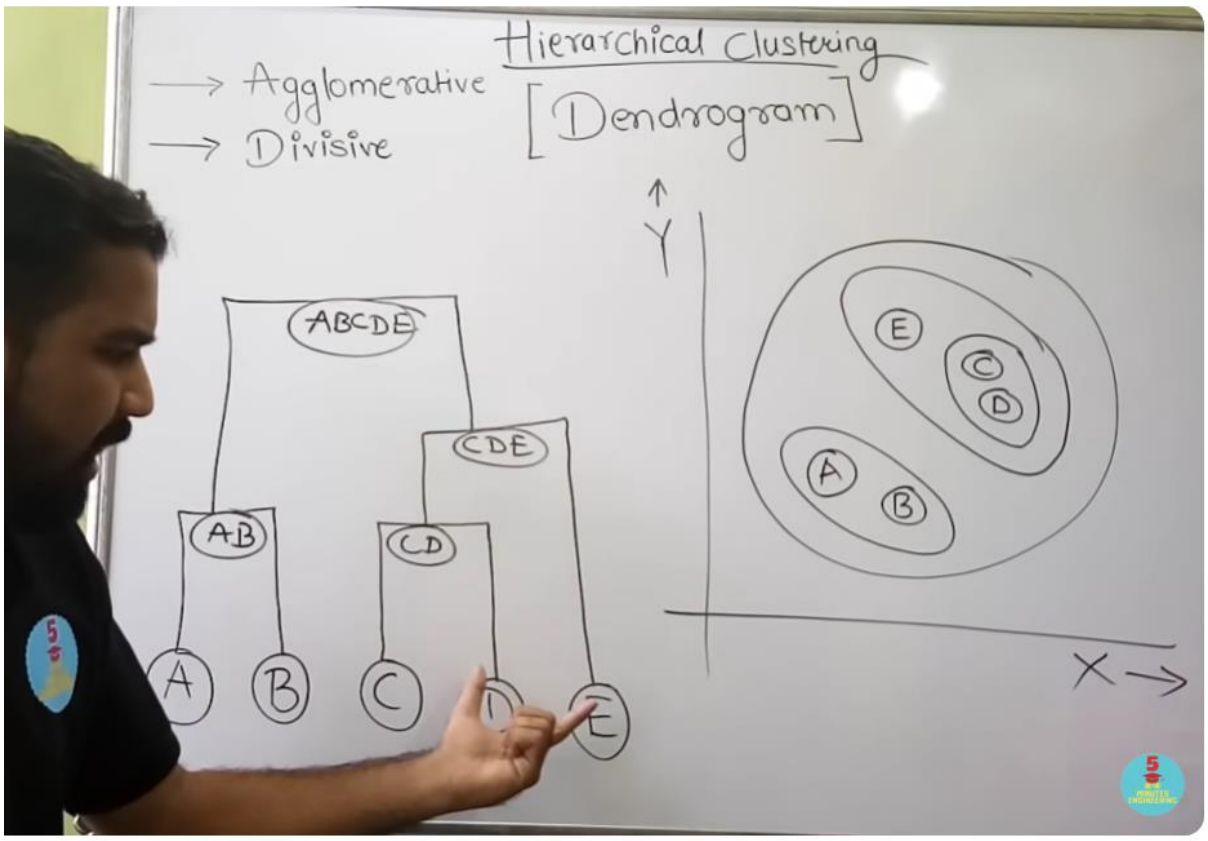


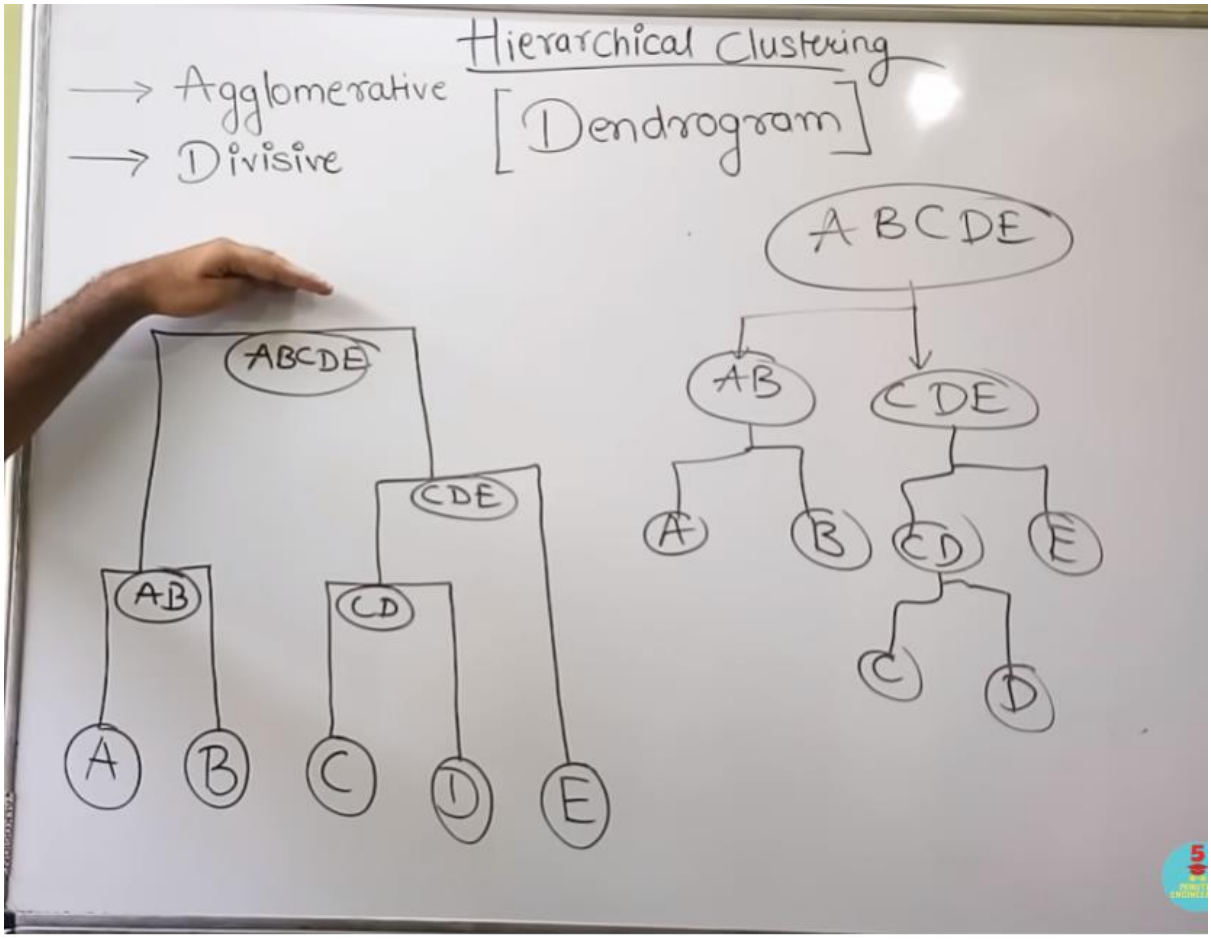

A **Hierarchical clustering** method works via grouping data into a tree of clusters. Hierarchical clustering begins by treating every data point as a separate cluster. Then, it repeatedly executes the subsequent steps:

1. Identify the 2 clusters which can be closest together, and
2. Merge the 2 maximum comparable clusters. We need to continue these steps until all the clusters are merged together.

In Hierarchical Clustering, the aim is to produce a hierarchical series of nested clusters. A diagram called **Dendrogram** (A Dendrogram is a tree-like diagram that statistics the sequences of merges or splits) graphically represents this hierarchy and is an inverted tree that describes the order in which factors are merged (bottom-up view) or clusters are broken up (top-down view).




Dendogram hai ye
agglomerative clustering – bottom-up approach

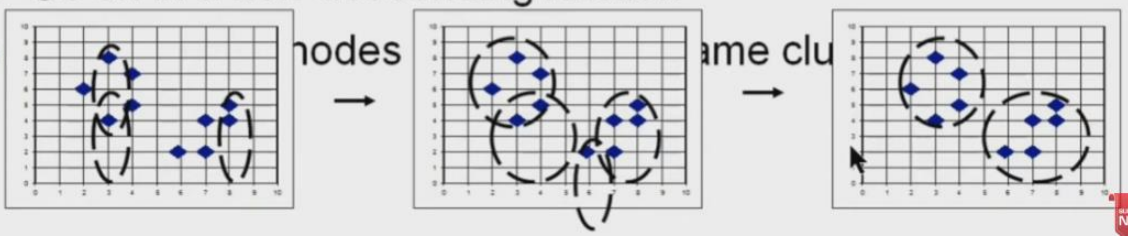

Hierarchical Clustering
→ Agglomerative
→ Divisive
[Dendrogram]
ABCDE
AB
CDE
A
B
C
D
E
ABCDE
AB
CDE
CD
A
B
C
D
E

Divisive

| S.No. | Parameters | Agglomerative Clustering | Divisive Clustering |
|---|---|---|---|
| 1. | **Category** | Bottom-up approach | Top-down approach |
| 2. | **Approach** | each data point starts in its own cluster, and the algorithm recursively merges the closest pairs of clusters until a single cluster containing all the data points is obtained. | all data points start in a single cluster, and the algorithm recursively splits the cluster into smaller sub-clusters until each data point is in its own cluster. |
| 3. | **Complexity level** | Agglomerative clustering is generally more computationally expensive, especially for large datasets as this approach requires the calculation of all pairwise distances between data points, which can be computationally expensive. | Comparatively less expensive as divisive clustering only requires the calculation of distances between sub-clusters, which can reduce the computational burden. |
| 4. | **Outliers** | Agglomerative clustering can handle outliers better than divisive clustering since outliers can be absorbed into larger clusters | divisive clustering may create sub-clusters around outliers, leading to suboptimal clustering results. |
| 5. | **Interpretability** | Agglomerative clustering tends to produce more interpretable results since the dendrogram shows the merging process of the clusters, and the user can choose the number of clusters based on the desired level of granularity. | divisive clustering can be more difficult to interpret since the dendrogram shows the splitting process of the clusters, and the user must choose a stopping criterion to determine the number of clusters. |
| 6. | **Implementation** | Scikit-learn provides multiple linkage methods for agglomerative clustering, such as “ward,” “complete,” “average,” and “single,” | divisive clustering is not currently implemented in Scikit-learn. |
| 7. | **Example** | Here are some of the applications in which Agglomerative Clustering is used :<br>Image segmentation, Customer segmentation, Social network analysis, Document clustering, Genetics, genomics, etc., and many more. | Here are some of the applications in which Divisive Clustering is used :<br>Market segmentation, Anomaly detection, Biological classification, Natural language processing, etc. |

# AGNES (Agglomerative Nesting)

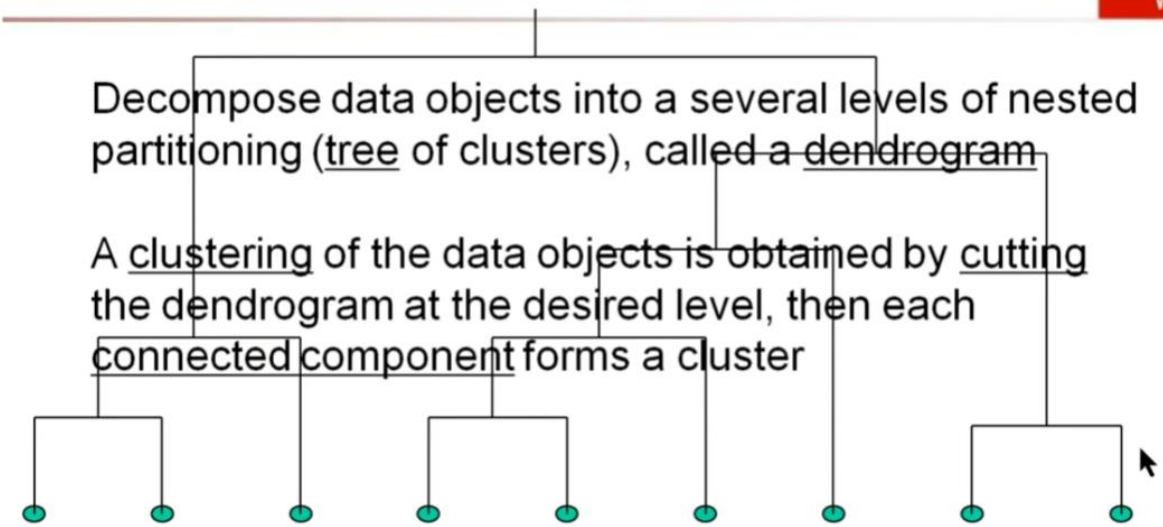
- Introduced in Kaufmann and Rousseeuw (1990)
- Implemented in statistical packages, e.g., Splus
- Use the **single-link** method and the dissimilarity matrix
- Merge nodes that have the least dissimilarity
- Go on in a non-descending fashion
- nodes → ame clu →



# *Dendrogram:* Shows How Clusters are Merg

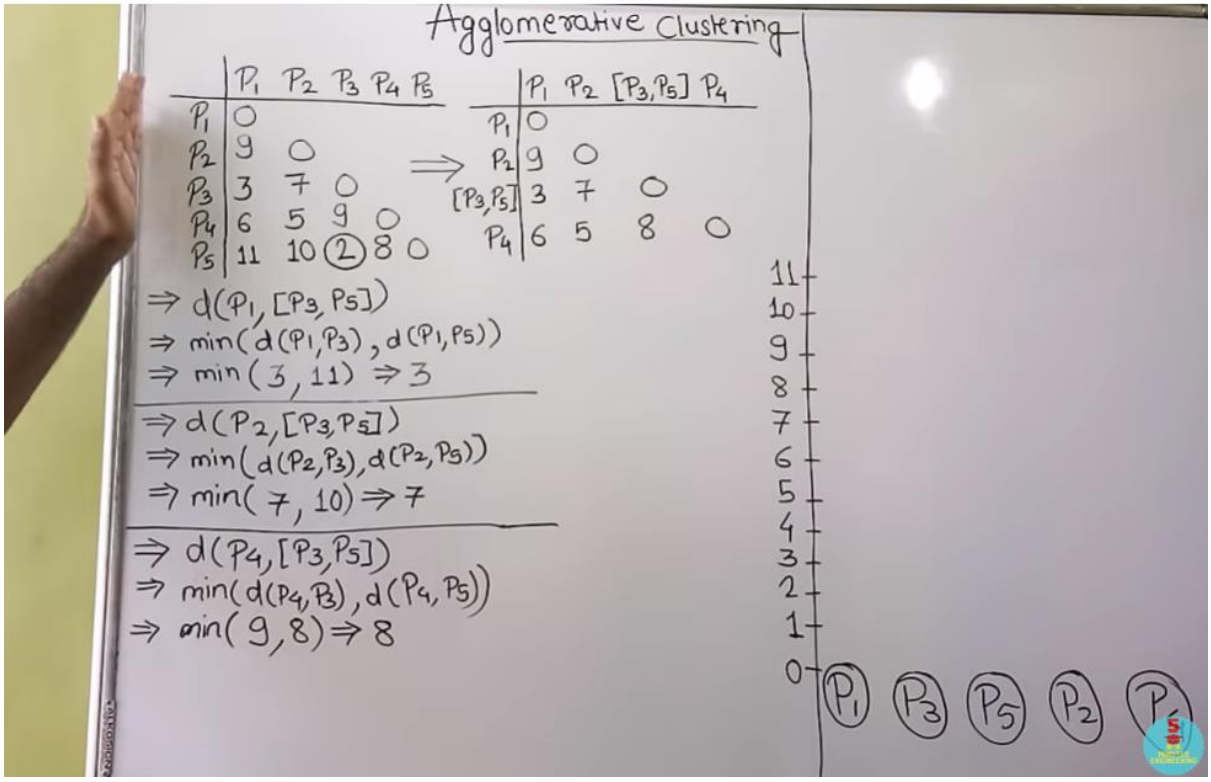
Decompose data objects into a several levels of nested partitioning (tree of clusters), called a dendrogram

A clustering of the data objects is obtained by cutting the dendrogram at the desired level, then each connected component forms a cluster



# DENDROGRAM

- A tree like structure which represents hierarchical technique.
  - ✓ Leaf- Individual.
  - ✓ Root – One cluster.
- A cluster at level 1, is the merger of its child cluster at level i + 1.

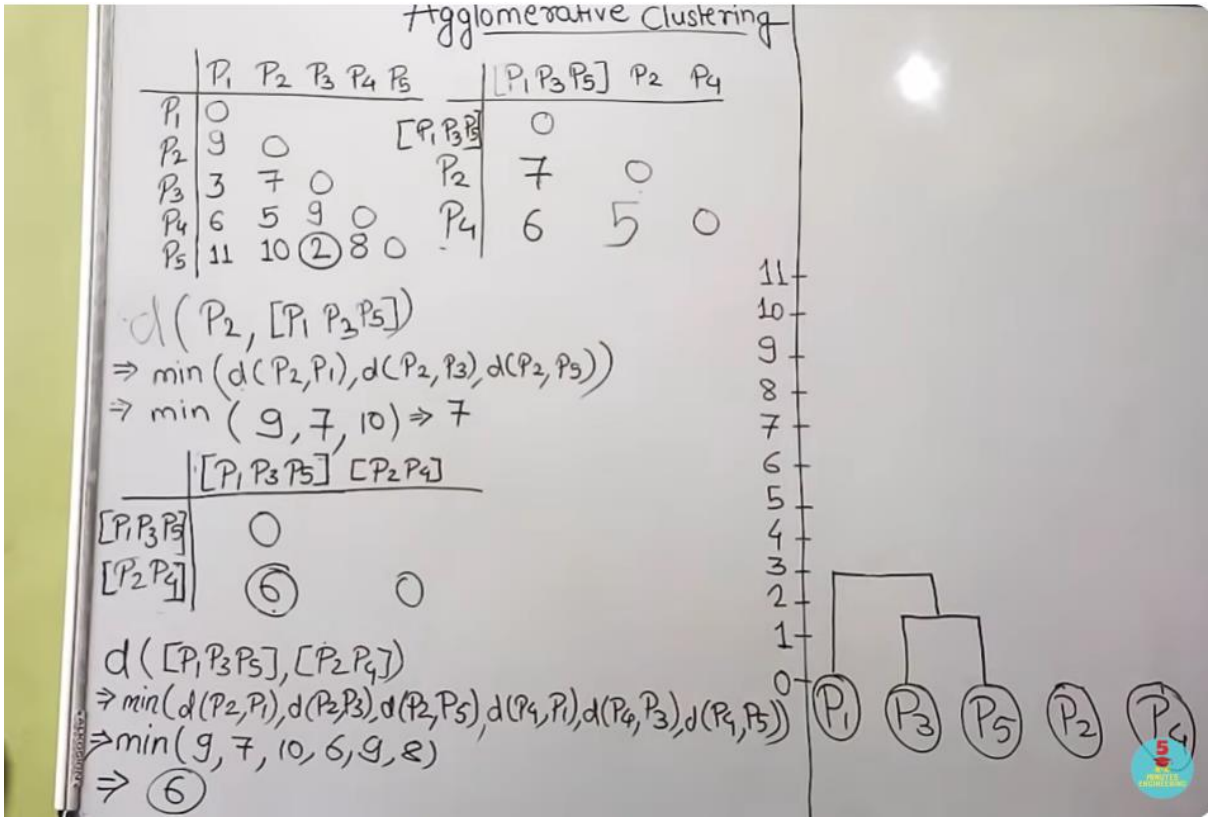


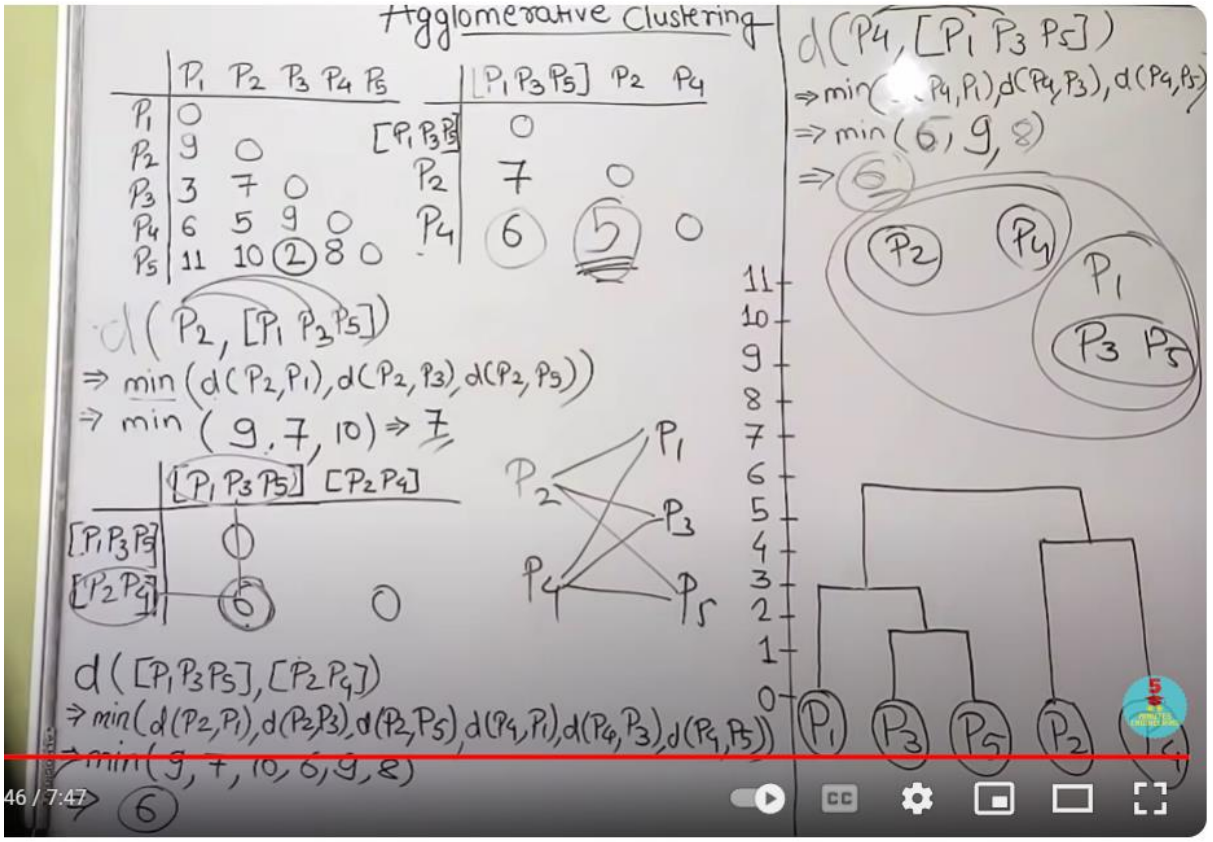

Distance matrix diya h

Sbse Chhota choose kiya aur usme clustering krdi

Dono m se jo min. hoga vo updated value hogi

(in single link technique )
agr complete link technique hoti to min ki jgh max hojaata

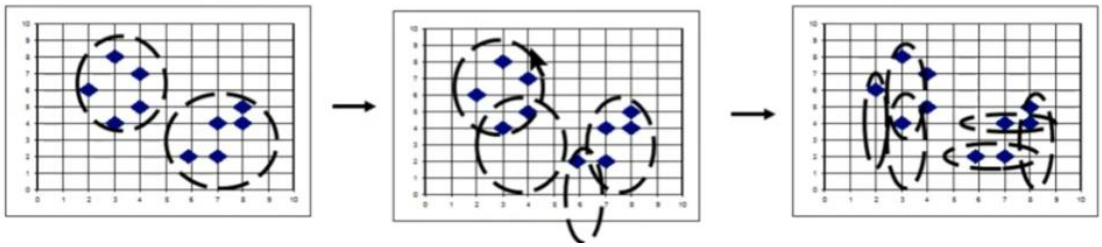



stering (Single Linkage) Part-2 Explained with Solved Example in Hindi

# 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

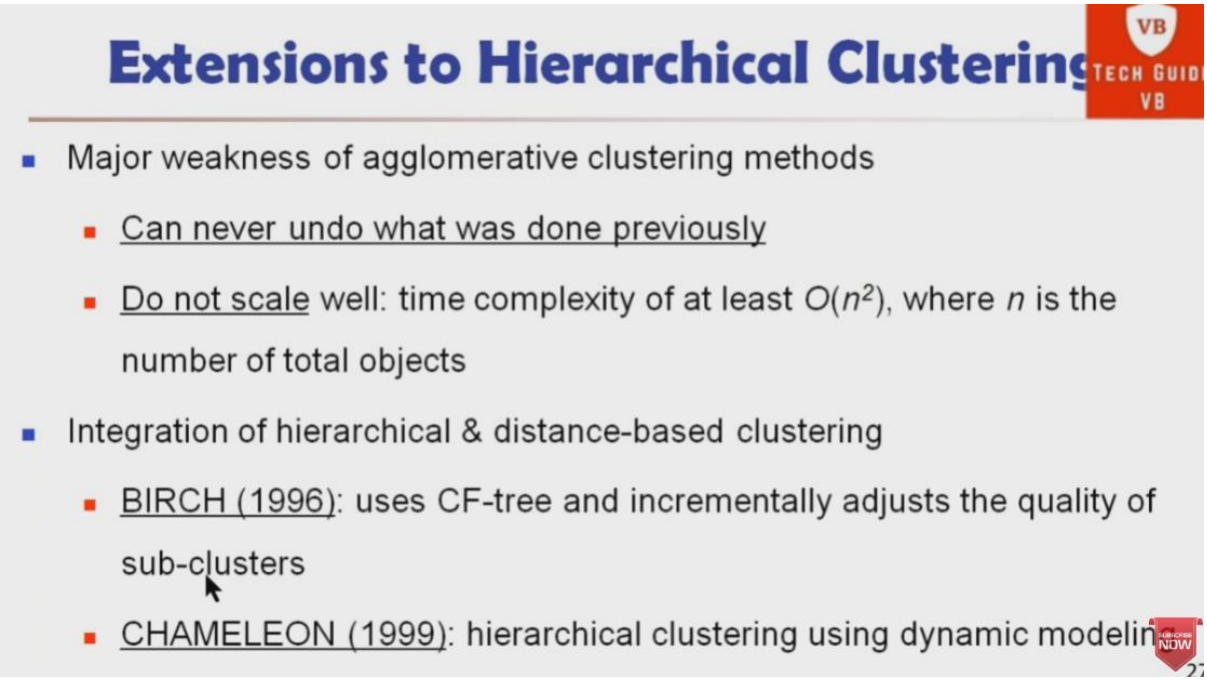
- Single link: smallest distance between an element in one cluster and an element in the other, i.e., $dist(K_i, K_j) = min(t_{ip}, t_{jq})$
- Complete link: largest distance between an element in one cluster and an element in the other, i.e., $dist(K_i, K_j) = max(t_{ip}, t_{jq})$
- Average: avg distance between an element in one cluster and an element in the other, i.e., $dist(K_i, K_j) = avg(t_{ip}, t_{jq})$
- Centroid: distance between the centroids of two clusters, i.e., $dist(K_i, K_j) = dist(C_i, C_j)$
- Medoid: distance between the medoids of two clusters, i.e., $dist(K_i, K_j) = dist(M_i, M_j)$
  - Medoid: a chosen, centrally located object in the cluster

**Hierarchical clustering has several advantages over other clustering methods**

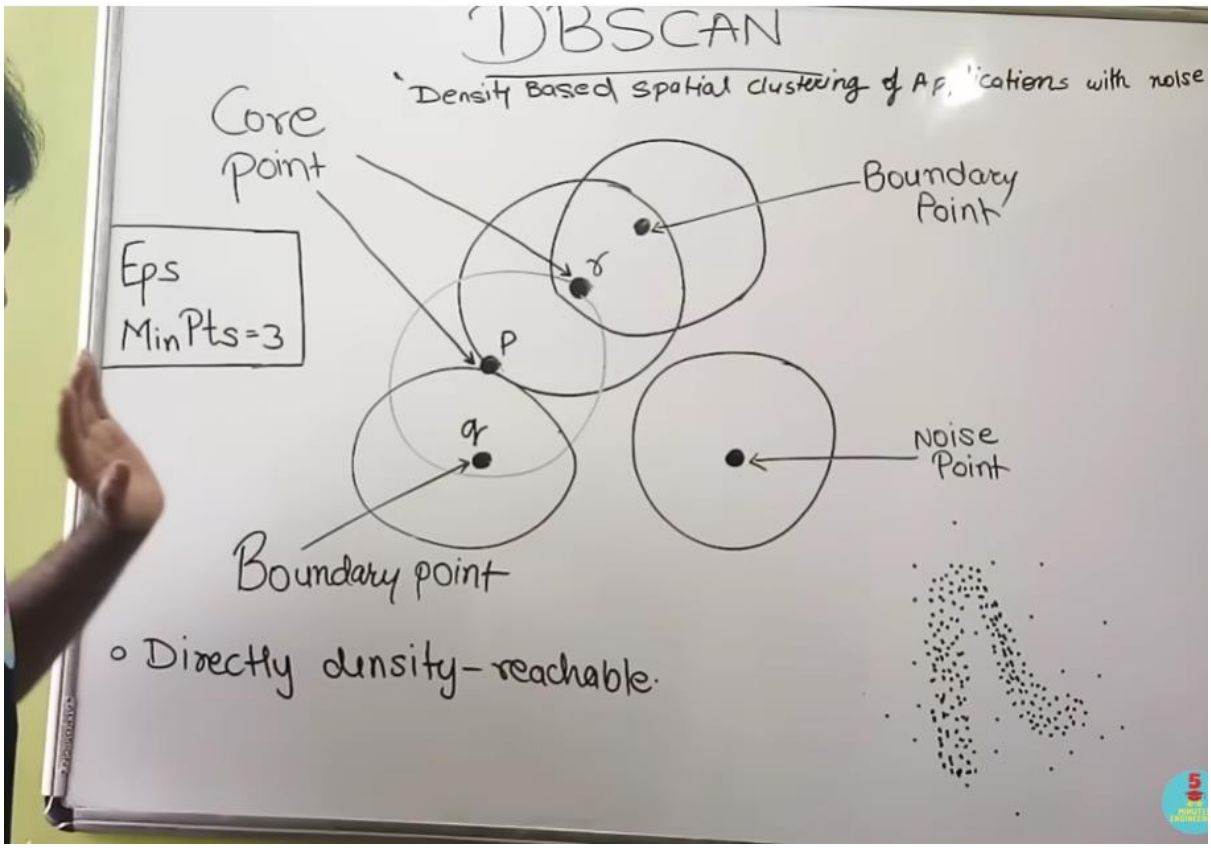
- The ability to handle non-convex clusters and clusters of different sizes and densities.
- The ability to handle missing data and noisy data.
- The ability to reveal the hierarchical structure of the data, which can be useful for understanding the relationships among the clusters.

**Drawbacks of Hierarchical Clustering**

- The need for a criterion to stop the clustering process and determine the final number of clusters.
- The computational cost and memory requirements of the method can be high, especially for large datasets.
- The results can be sensitive to the initial conditions, linkage criterion, and distance metric used.
  In summary, Hierarchical clustering is a method of data mining that groups similar data points into clusters by creating a hierarchical structure of the clusters.
- This method can handle different types of data and reveal the relationships among the clusters. However, it can have high computational cost and results can be sensitive to some conditions.



| k-means Clustering | Hierarchical Clustering |
| --- | --- |
| k-means, using a pre-specified number of clusters, the method assigns records to each cluster to find the mutually exclusive cluster of spherical shape based on distance. | Hierarchical methods can be either divisive or agglomerative. |
| K Means clustering needed advance knowledge of K i.e. no. of clusters one want to divide your data. | In hierarchical clustering one can stop at any number of clusters, one find appropriate by interpreting the dendrogram. |
| One can use median or mean as a cluster centre to represent each cluster. | Agglomerative methods begin with 'n' clusters and sequentially combine similar clusters until only one cluster is obtained. |
| Methods used are normally less computationally intensive and are suited with very large datasets. | Divisive methods work in the opposite direction, beginning with one cluster that includes all the records and Hierarchical methods are especially useful when the target is to arrange the clusters into a natural hierarchy. |
| In K Means clustering, since one start with random choice of clusters, the results produced by running the algorithm many times may differ. | In Hierarchical Clustering, results are reproducible in Hierarchical clustering |
| K- means clustering a simply a division of the set of data objects into non-overlapping subsets (clusters) such that each data object is in exactly one subset). | A hierarchical clustering is a set of nested clusters that are arranged as a tree. |
| K Means clustering is found to work well when the structure of the clusters is hyper spherical (like circle in 2D, sphere in 3D). | Hierarchical clustering don't work as well as, k means when the shape of the clusters is hyper spherical. |

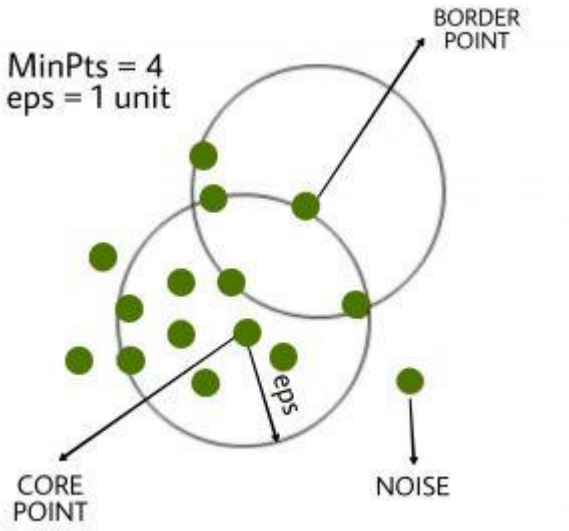



Epsilon radius hoga
aur uss circle m 3 min. points

Centre point coure point hoga

Agr core point k nazdeek hai to vo boundary point hai

Jo core bhi nahi hai aur boundary bhi nahi hai to vo hua outlier , mtb the noise point jise hum kisi bhi cluster m include nahi krte hai



**DBSCAN** stands for **Density-Based Spatial Clustering of Applications with Noise**.
It is a popular unsupervised learning method used for model construction and machine learning algorithms. It is a clustering method utilized for separating high-density clusters from low-density clusters. It divides the data points into many groups so that points lying in the same group will have the same properties. It was proposed by Martin Ester, Hans-Peter Kriegel, Jorg Sander, and Xiaowei Xu in 1996.
DBSCAN is designed for use with databases that can accelerate region queries. It can not cluster data sets with large differences in their densities.

**Characteristics**

- It identifies clusters of any shape in a data set, it means it can detect arbitrarily shaped clusters.
- It is based on intuitive notions of clusters and noise.
- It is very robust in detection of outliers in data set
- It requires only two points which are very insensitive to the order of occurrence of the points in data set

## 2.00 DBSCAN Clustering Algorithm Solved Example – 1

- Apply the DBSCAN algorithm to the given data points and
- Create the clusters with
- minPts = 4 and
- epsilon (ε) = 1.9.

Data Points:

| | |
|---|---|
| P1: (3, 7) | P2: (4, 6) |
| P3: (5, 5) | P4: (6, 4) |
| P5: (7, 3) | P6: (6, 2) |
| P7: (7, 2) | P8: (8, 4) |
| P9: (3, 3) | P10: (2, 6) |
| P11: (3, 5) | P12: (2, 4) |

## 2.00 DBSCAN Clustering Algorithm Solved Example – 1

- Use Eucladian distance and calculate the distance between each points.

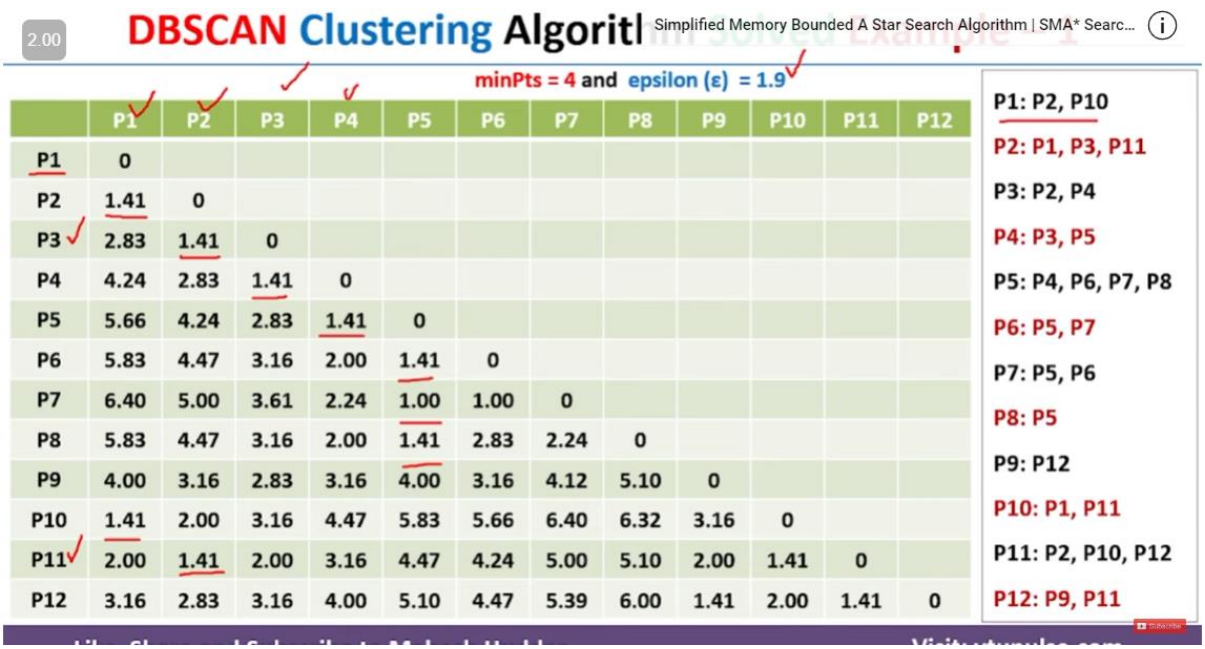
$$Distance(A(x_1, y_1), B(x_2, y_2)) = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$$

## 2.00 DBSCAN Clustering Algorithm Solved Example – 1

P1: (3, 7)
P2: (4, 6)
P3: (5, 5)
P4: (6, 4)
P5: (7, 3)
P6: (6, 2)
P7: (7, 2)
P8: (8, 4)
P9: (3, 3)
P10: (2, 6)
P11: (3, 5)
P12: (2, 4)

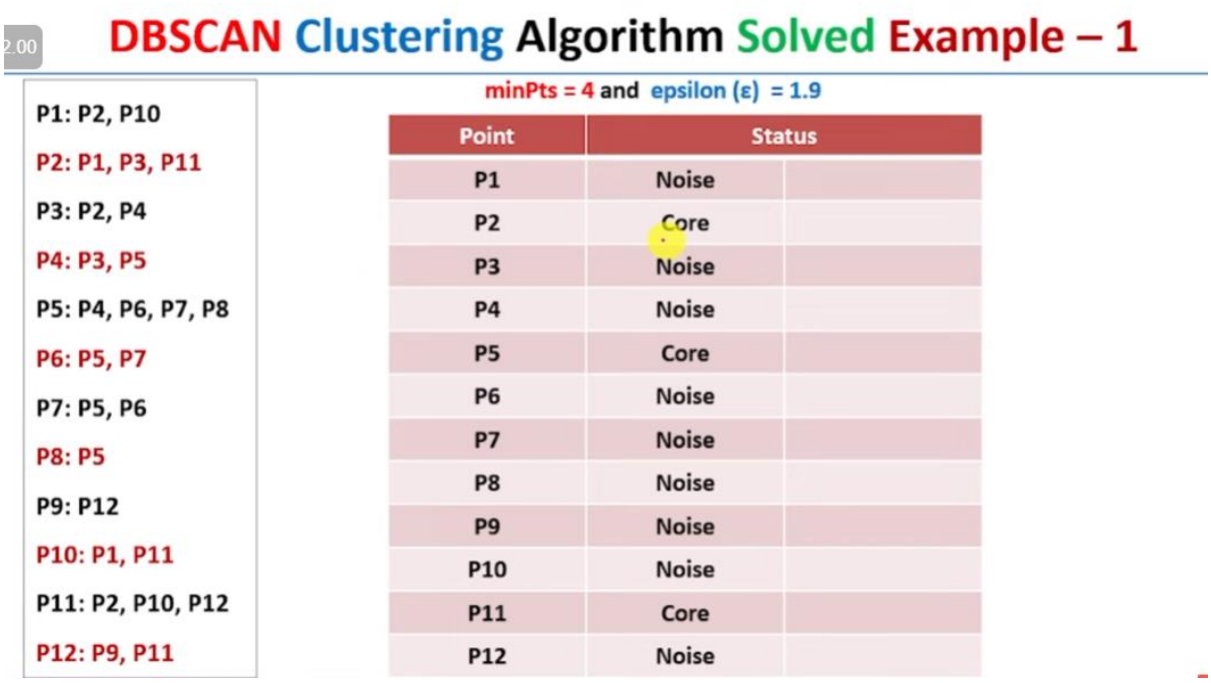
minPts = 4 and epsilon (ε) = 1.9

| | P1 | P2 | P3 | P4 | P5 | P6 | P7 | P8 | P9 | P10 | P11 | P12 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| P1 | 0 | | | | | | | | | | | |
| P2 | 1.41 | 0 | | | | | | | | | | |
| P3 | 2.83 | 1.41 | 0 | | | | | | | | | |
| P4 | 4.24 | 2.83 | 1.41 | 0 | | | | | | | | |
| P5 | 5.66 | 4.24 | 2.83 | 1.41 | 0 | | | | | | | |
| P6 | 5.83 | 4.47 | 3.16 | 2.00 | 1.41 | 0 | | | | | | |
| P7 | 6.40 | 5.00 | 3.61 | 2.24 | 1.00 | 1.00 | 0 | | | | | |
| P8 | 5.83 | 4.47 | 3.16 | 2.00 | 1.41 | 2.83 | 2.24 | 0 | | | | |
| P9 | 4.00 | 3.16 | 2.83 | 3.16 | 4.00 | 3.16 | 4.12 | 5.10 | 0 | | | |
| P10 | 1.41 | 2.00 | 3.16 | 4.47 | 5.83 | 5.66 | 6.40 | 6.32 | 3.16 | 0 | | |
| P11 | 2.00 | 1.41 | 2.00 | 3.16 | 4.47 | 4.24 | 5.00 | 5.10 | 2.00 | 1.41 | 0 | |
| P12 | 3.16 | 2.83 | 3.16 | 4.00 | 5.10 | 4.47 | 5.39 | 6.00 | 1.41 | 2.00 | 1.41 | 0 |




| | P1 | P2 | P3 | P4 | P5 | P6 | P7 | P8 | P9 | P10 | P11 | P12 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| P1 | 0 | | | | | | | | | | | |
| P2 | 1.41 | 0 | | | | | | | | | | |
| P3 | 2.83 | 1.41 | 0 | | | | | | | | | |
| P4 | 4.24 | 2.83 | 1.41 | 0 | | | | | | | | |
| P5 | 5.66 | 4.24 | 2.83 | 1.41 | 0 | | | | | | | |
| P6 | 5.83 | 4.47 | 3.16 | 2.00 | 1.41 | 0 | | | | | | |
| P7 | 6.40 | 5.00 | 3.61 | 2.24 | 1.00 | 1.00 | 0 | | | | | |
| P8 | 5.83 | 4.47 | 3.16 | 2.00 | 1.41 | 2.83 | 2.24 | 0 | | | | |
| P9 | 4.00 | 3.16 | 2.83 | 3.16 | 4.00 | 3.16 | 4.12 | 5.10 | 0 | | | |
| P10 | 1.41 | 2.00 | 3.16 | 4.47 | 5.83 | 5.66 | 6.40 | 6.32 | 3.16 | 0 | | |
| P11 | 2.00 | 1.41 | 2.00 | 3.16 | 4.47 | 4.24 | 5.00 | 5.10 | 2.00 | 1.41 | 0 | |
| P12 | 3.16 | 2.83 | 3.16 | 4.00 | 5.10 | 4.47 | 5.39 | 6.00 | 1.41 | 2.00 | 1.41 | 0 |

Nearest data points less than epsilon




| Point | Status |
|---|---|
| P1 | Noise |
| P2 | Core |
| P3 | Noise |
| P4 | Noise |
| P5 | Core |
| P6 | Noise |
| P7 | Noise |
| P8 | Noise |
| P9 | Noise |
| P10 | Noise |
| P11 | Core |
| P12 | Noise |

Min. points k acc. Noise aur core calc. kre

P1: P2, P10

P2: P1, P3, P11

P3: P2, P4

P4: P3, P5

P5: P4, P6, P7, P8

P6: P5, P7

P7: P5, P6

P8: P5

P9: P12

P10: P1, P11

P11: P2, P10, P12

P12: P9, P11

minPts = 4 and epsilon (ε) = 1.9

| Point | Status | |
|---|---|---|
| P1 | Noise | Border |
| P2 | Core | |
| P3 | Noise | Border |
| P4 | Noise | Border |
| P5 | Core | |
| P6 | Noise | Border |
| P7 | Noise | Border |
| P8 | Noise | Border |
| P9 | Noise | |
| P10 | Noise | Border |
| P11 | Core | |
| P12 | Noise | Border |

Jo kisi core point ka part hai vo hogye border point

2.00

## DBSCAN Clustering Algorithm Solved Example – 1

P1: P2, P10

P2: P1, P3, P11

P3: P2, P4

P4: P3, P5

P5: P4, P6, P7, P8

P6: P5, P7

P7: P5, P6

P8: P5

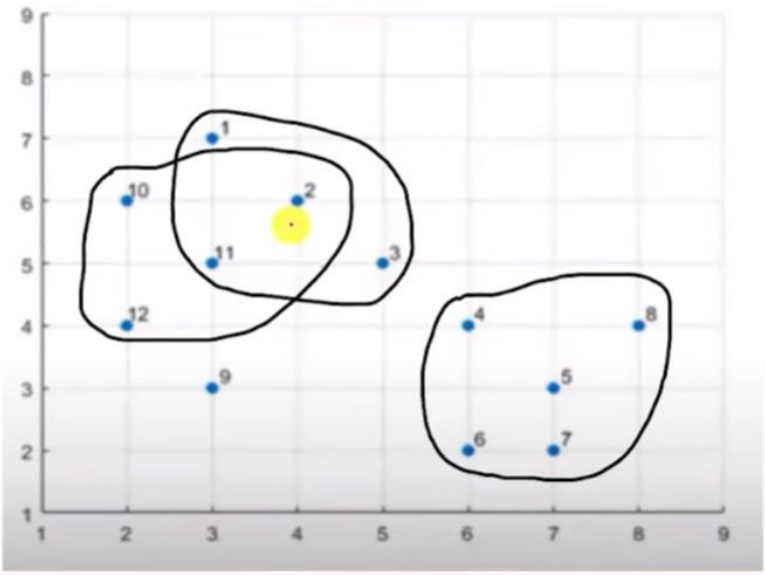
P9: P12

P10: P1, P11

P11: P2, P10, P12

P12: P9, P11

minPts = 4 and epsilon (ε) = 1.9



DBSCAN Algorithm

16 December 2023 15:14

Step1 → minPts (↳3) / epsilon ε

**Step 1** - Identify all points as either core point, border point or noise point

**Step 2** - For all of the unclustered core points

**Step 2a** - Create a new cluster

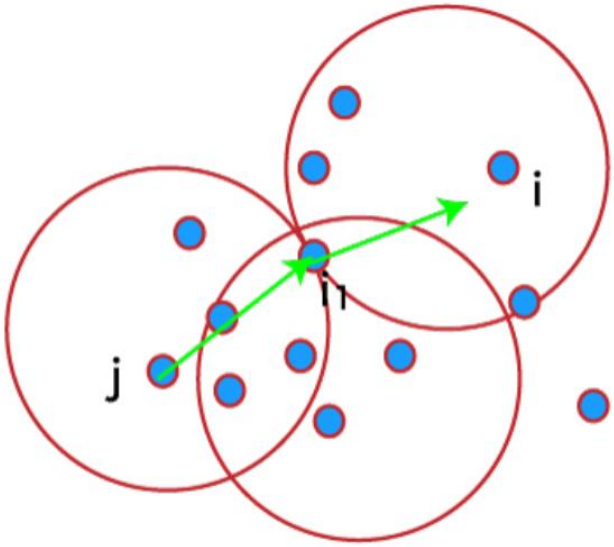
**Step 2b** - add all the points that are unclustered and density connected to the current point into this cluster

**Step 3** - For each unclustered border point assign it to the cluster of nearest core point

**Step 4** - Leave all the noise points as it is.

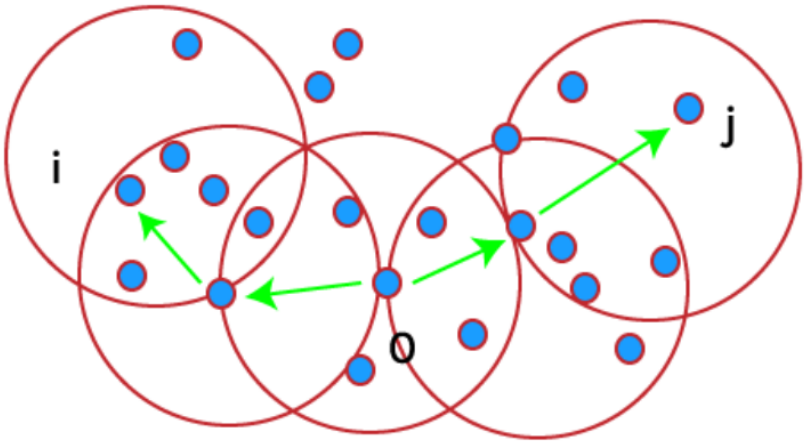
**Density reachable:**

A point denoted by i is a density reachable from a point j with respect to Eps, MinPts if there is a sequence chain of a point i1,...., in, i1 = j, pn = i such that $i_i$ + 1 is directly density reachable from $i_i$.



**Density connected:**

A point i refers to density connected to a point j with respect to Eps, MinPts if there is a point o such that both i and j are considered as density reachable from o with respect to Eps and MinPts.




**Advantages**

- Specification of number of clusters of data in the data set is not required.
- It can find any shape cluster even if the cluster is surrounded by any other cluster.
- It can easily find outliers in data set.
- It is not much sensitive to noise, it means it is noise tolerant.
- It is the second most used clustering method after K-means.

**Disadvantages**

- The quality of the result depends on the distance measure used in the regionQuery function.
- Border points may go in any cluster depending on the processing order so it is not completely deterministic.
- It can be expensive when cost of computation of nearest neighbor is high.
- It can be slow in execution for higher dimension.
- Adaptability of variation in local density is less.

| DBSCAN | K-Means |
|---|---|
| In DBSCAN we need not specify the number of clusters. | K-Means is very sensitive to the number of clusters so it need to specified |
| Clusters formed in DBSCAN can be of any arbitrary shape. | Clusters formed in K-Means are spherical or convex in shape |
| DBSCAN can work well with datasets having noise and outliers | K-Means does not work well with outliers data. Outliers can skew the clusters in K-Means to a very large extent. |
| In DBSCAN two parameters are required for training the Model | In K-Means only one parameter is required is for training the model |