UNIVERSITY OF ENGINEERING & MANAGEMENT, KOLKATA

Course Name : AI & ML

Clustering

Unsupervised method

Unsupervised learning

- Given a set of unlabeled data points / items
- Find patterns or structure in the data
- Clustering: automatically group the data points /
- items into groups of ‘similar’ or ‘related’ points

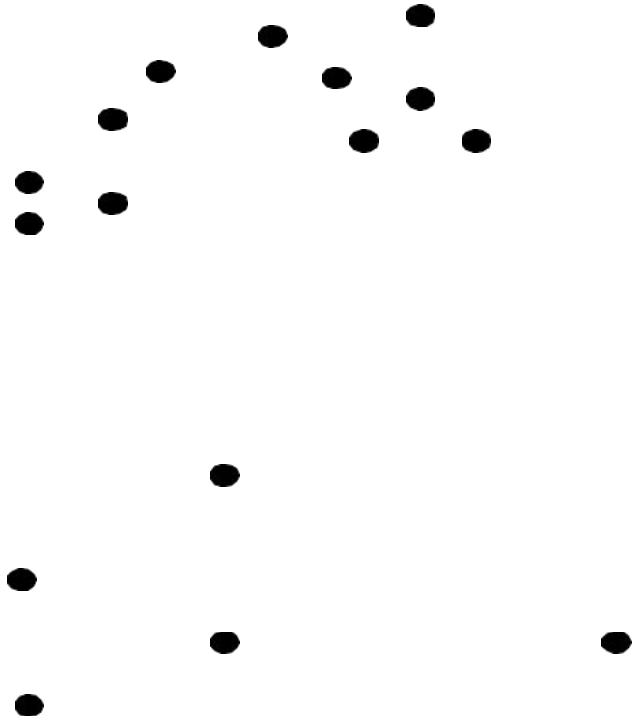
Motivations for Clustering

- Understanding the data better
  - Grouping Web search results into clusters, each of which
    - captures a particular aspect of the query
  - Segment the market or customers of a service
- As precursor for some other application
  - Summarization and data compression
  - Recommendation

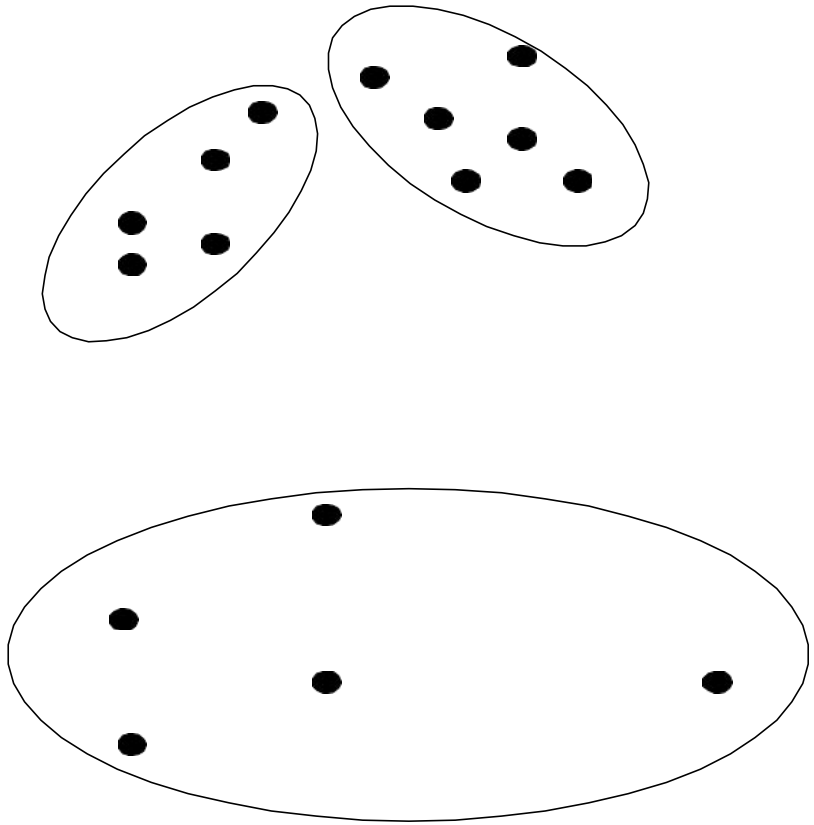
- Clustering for Data Understanding and Applications
- Biology: taxonomy of living things: kingdom, phylum, class, order, family, genus and species
- Information retrieval: document clustering
- Land use: Identification of areas of similar land use in an earth observation database
- Marketing: Help marketers 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 research

- Different types of clustering
- Partitional (Hard Clustering)
  - Divide set of items into non-overlapping subsets
  - Each item will be member of one subset
- Overlapping (Soft Clustering)
  - Divide set of items into potentially overlapping subsets
  - Each item can simultaneously belong to multiple subsets

Partitional Clustering (Hard)

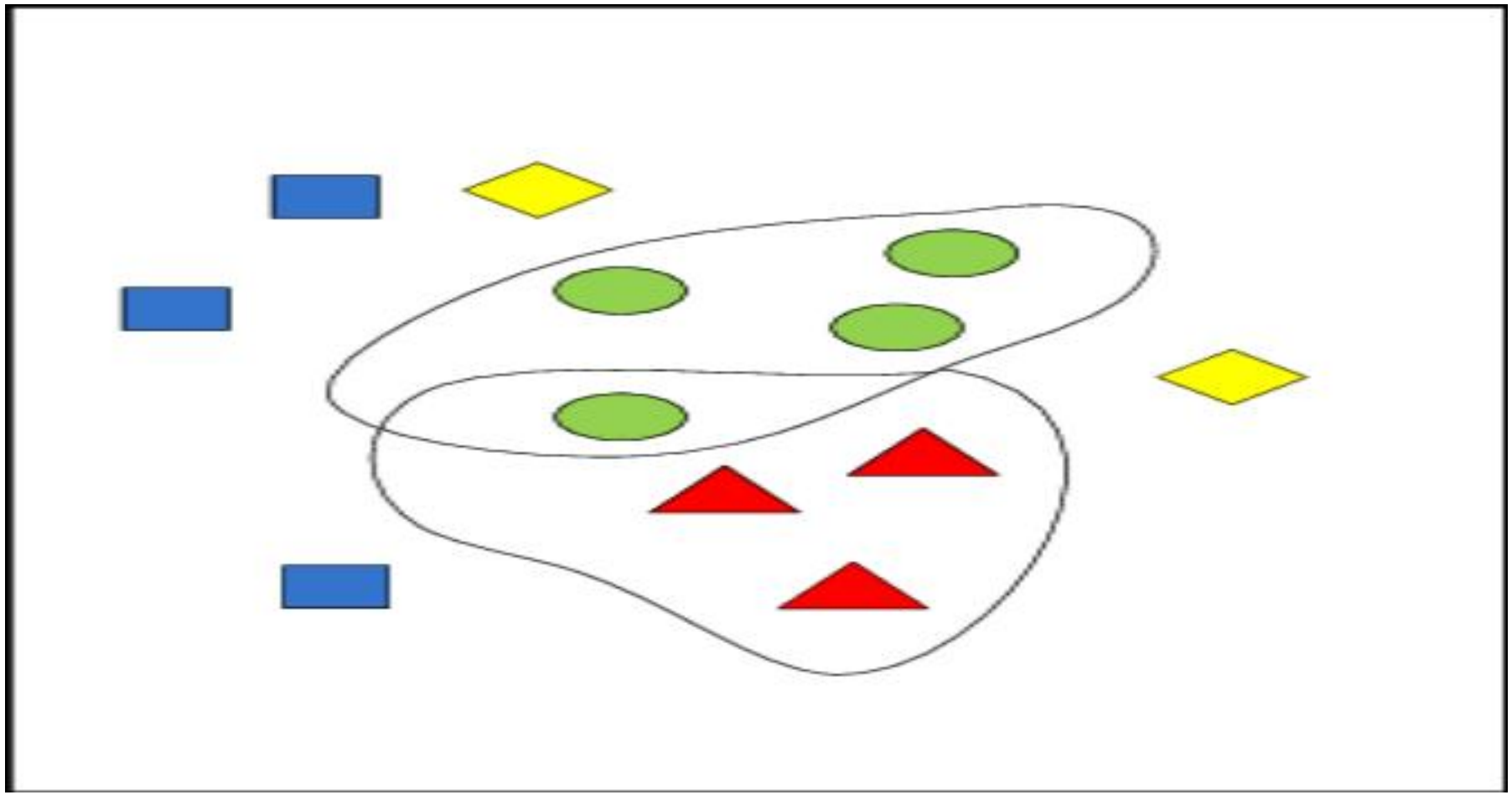


Original Points



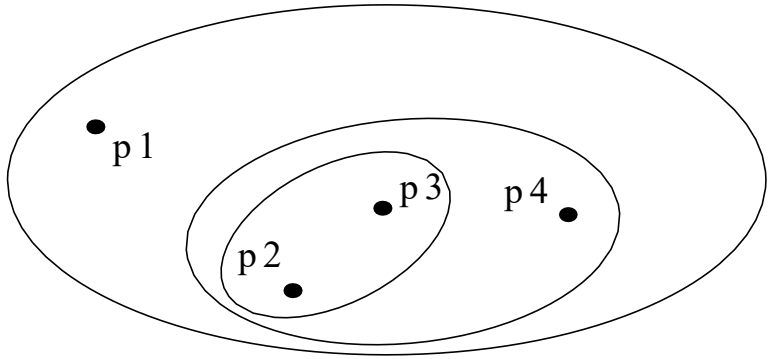
A Partitional Clustering

Overlapping (Soft Clustering)

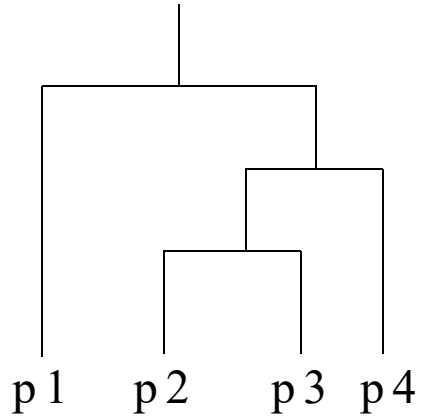


- Hierarchical
  - Set of nested clusters, where one larger cluster can contain smaller clusters
  - Organized as a tree (dendrogram): leaf nodes are singleton clusters containing individual items, each intermediate node is union of its children sub-clusters
  - A sequence of partitional clusterings – cut the dendrogram at a certain level to get a partitional clustering

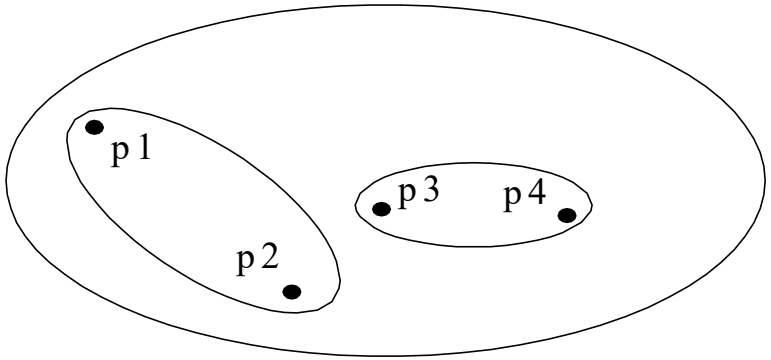
Hierarchical Clustering



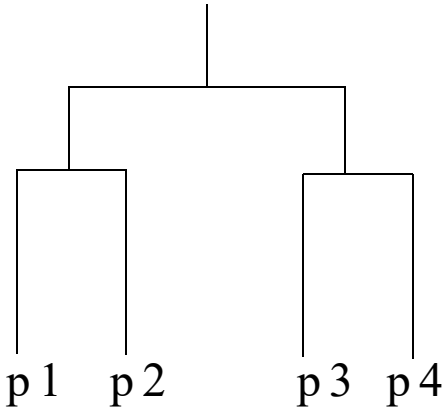
Traditional Hierarchical Clustering



Traditional Dendrogram



Non-traditional Hierarchical Clustering

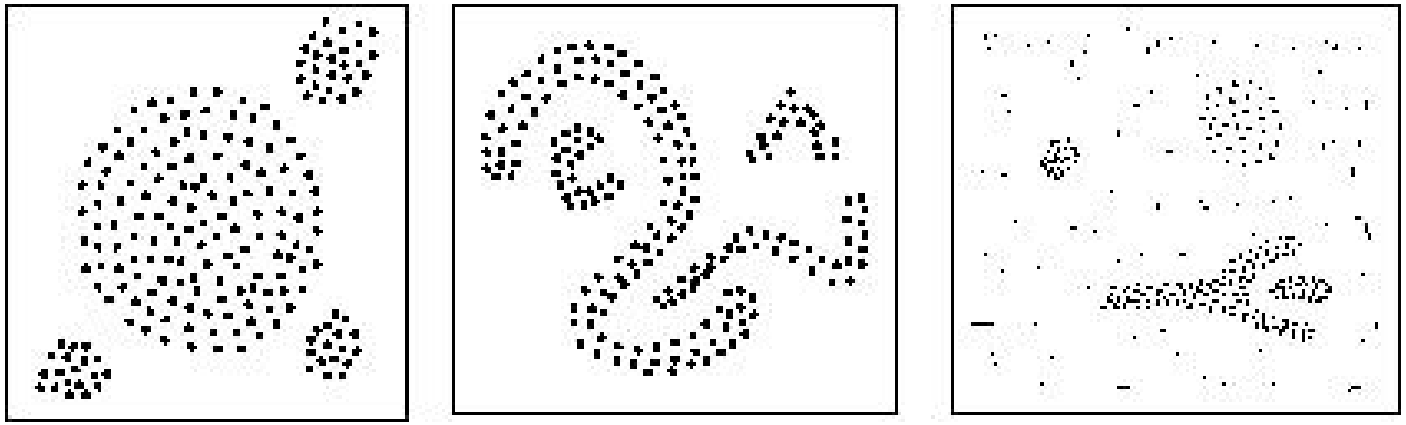


Non-traditional Dendrogram

- Density-based
  - Assumes items distributed in a space where ‘similar’ items
    - are placed close to each other (e.g., feature space)
  - A cluster is a dense region of items, that is surrounded by a region of low density
  - Example method: DBSCAN

Density based clustering methods

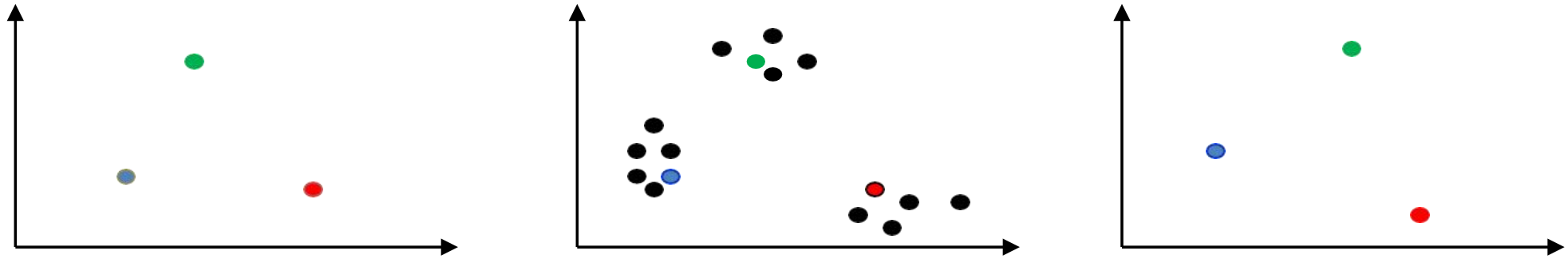
- Locates regions of high density, that are separated from one another by regions of low density



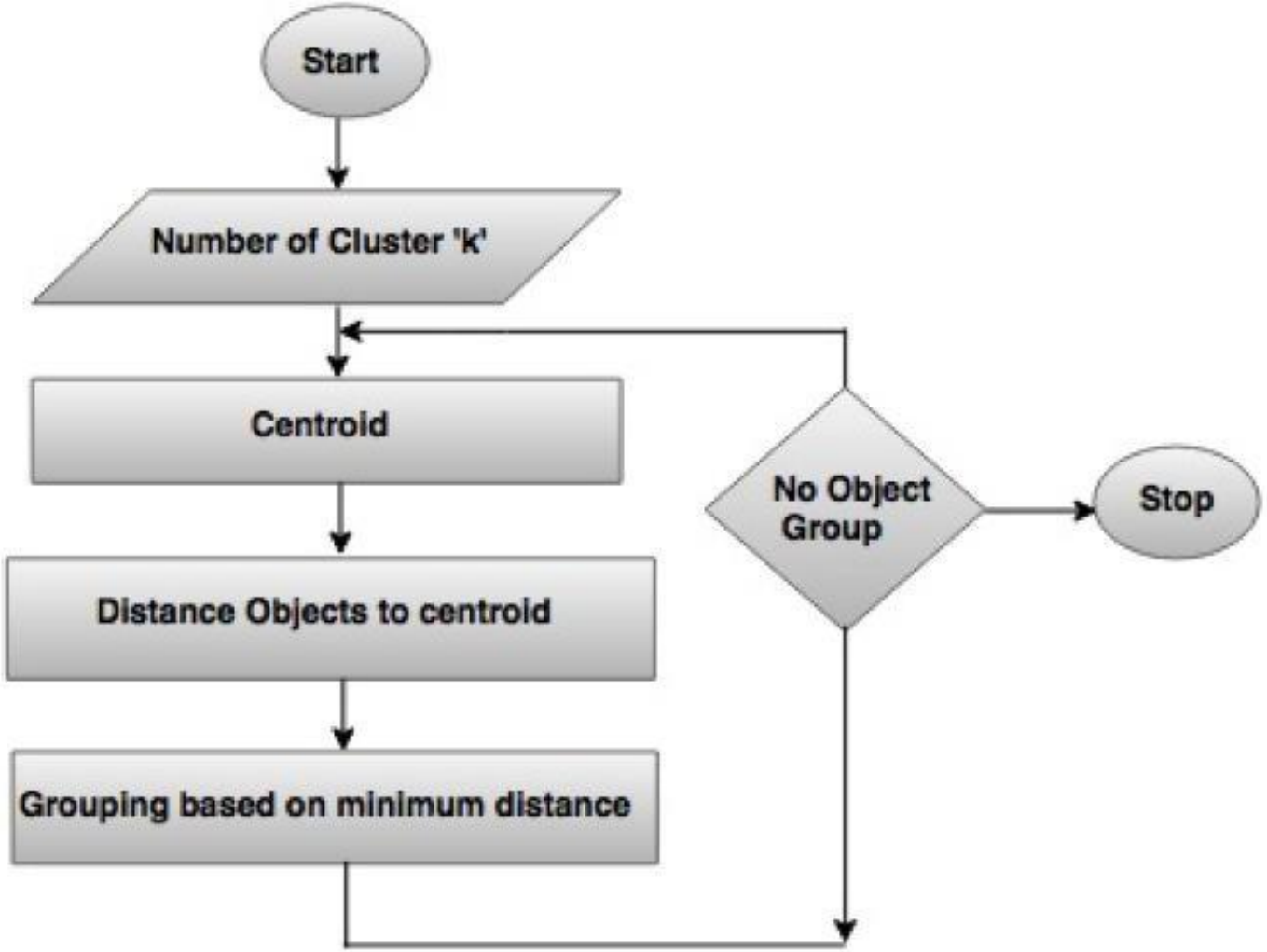
K-means algorithm

Given *k*

1. **Randomly choose *k* data points (seeds) to be the initial cluster centres**
2. **Assign each data point to the closest cluster centre**
3. **Re–compute the cluster centres using the current cluster memberships.**
4. **If a convergence criterion is not met, go to 2.**



K-means algorithm



For the below dataset find out the final clusters using K-Means algorithm.

Note: i) K=2; ii) Use ‘Euclidean distance’;

X	Y
1.0	1.0
1.5	2.0
3.0	4.0
5.0	7.0
3.5	5.0
4.5	5.0
3.5	4.5

Advantages

- Fast, robust easy to understand.
- Relatively efficient
- Gives best result when data set are distinct or well separated from each other

Disadvantages

- Requires apriori specification of the number of cluster centers.
- Hard assignment of data points to clusters
- Euclidean distance measures can unequally weight underlying factors.
- Applicable only when mean is defined i.e. fails for categorical data.
- Only local optima

- **K-Medoids** (also called Partitioning Around Medoid) algorithm was proposed in 1987 by Kaufman and Rousseeuw.
- A medoid can be defined as a point in the cluster, whose dissimilarities with all the other points in the cluster are minimum.
- The dissimilarity of the medoid(Ci) and object(Pi) is calculated by using $E = |Pi – Ci|$

Algorithm

Given the value of k and unlabelled data:

1. Choose k number of random points from the data and assign these k points to k number of clusters. These are the initial medoids.
2. For all the remaining data points, calculate the distance from each medoid and assign it to the cluster with the nearest medoid.
3. Calculate the total cost (Sum of all the distances from all the data points to the medoids)
4. Select a random point as the new medoid and swap it with the previous medoid. Repeat 2 and 3 steps.
5. If the total cost of the new medoid is less than that of the previous medoid, make the new medoid permanent and repeat step 4.
6. If the total cost of the new medoid is greater than the cost of the previous medoid, undo the swap and repeat step 4.
7. The Repetitions have to continue until no change is encountered with new medoids to classify data points.

Example to be solved using K mediod . K=2

	X	Y
0	8	7
1	3	7
2	4	9
3	9	6
4	8	5
5	5	8
6	7	3
7	8	4
8	7	5
9	4	5

Advantages of using K-Medoids:

- Deals with noise and outlier data effectively
- Easily implementable and simple to understand
- Faster compared to other partitioning algorithms

Disadvantages:

- Not suitable for Clustering arbitrarily shaped groups of data points.
- As the initial medoids are chosen randomly, the results might vary based on the choice in different runs.

Similarity of K – Means & K-mediods

Both methods are types of Partition Clustering.
Unsupervised iterative algorithms
Have to deal with unlabelled data
Both algorithms group n objects into k clusters based on similar traits where k is pre-defined.
Inputs: Unlabelled data and the value of k

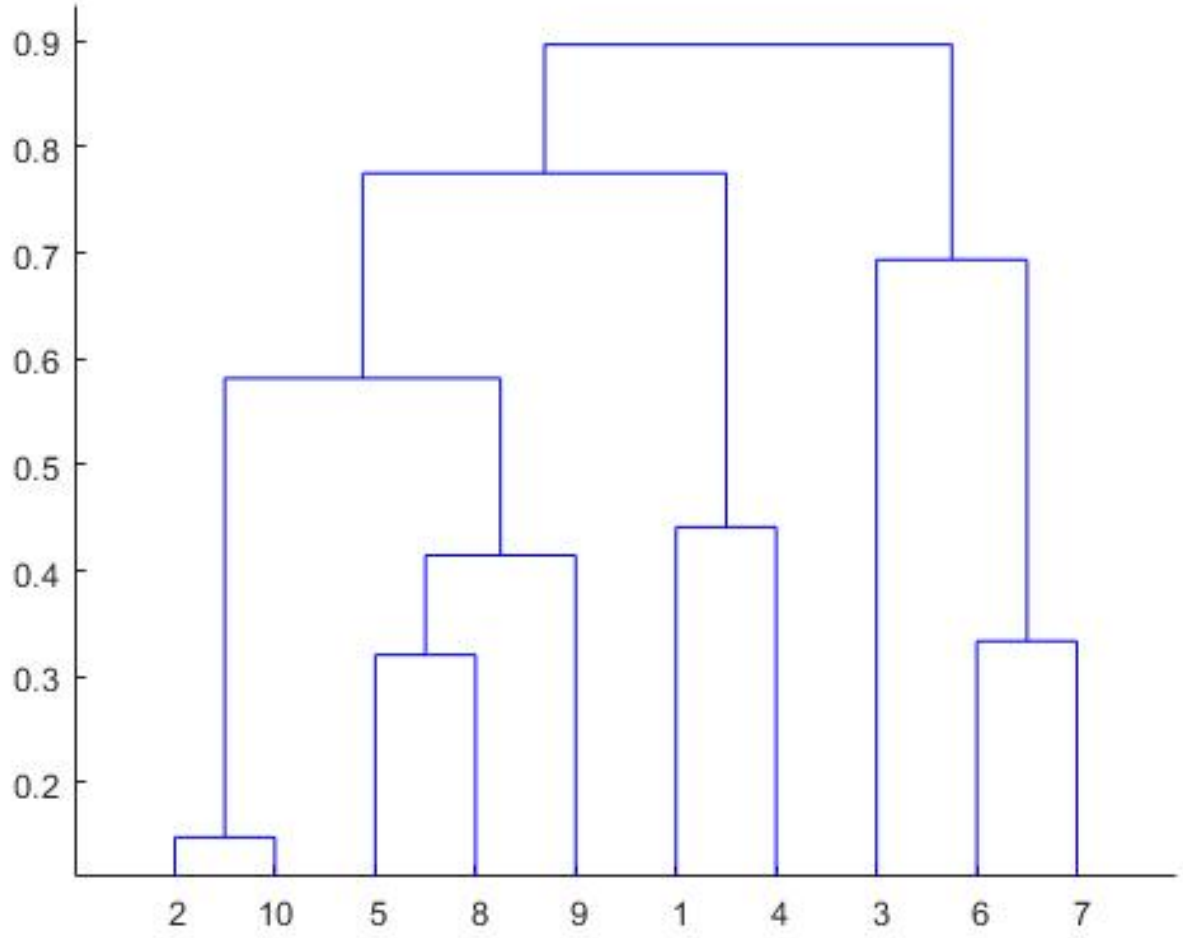
Differences

K-Means	K-Medoids
Metric of similarity: Euclidian Distance	**Metric of similarity:** Manhattan Distance
Clustering is done based on distance from **centroids.**	Clustering is done based on distance from **medoids.**
A centroid can be a data point or some other point in the cluster	A medoid is always a data point in the cluster.
Can't cope with outlier data	Can manage outlier data too
Sometimes, outlier sensitivity can turn out to be useful	Tendency to ignore meaningful clusters in outlier data

Hierarchical clustering

- Bottom-up or Agglomerative clustering
  - Start considering each data point as a singleton cluster
  - Successively merge clusters if similarity is sufficiently high
  - Until all points have been merged into a single cluster
- Top-down or Divisive clustering
  - Start with all data points in a single cluster
  - Iteratively split clusters into smaller sub-clusters if the
    - similarity between two sub-parts is low

Both Divisive and Agglomerative clustering can be represented as a Dendrogram



- Basic agglomerative hierarchical clustering algorithm
- Start with each item in a singleton cluster
- Compute the proximity/similarity matrix between clusters
- Repeat
  - Merge the closest/most similar two clusters
  - Update the proximity matrix to reflect proximity between the new cluster and the other clusters
- Until only one cluster remains

agglomerative hierarchical clustering Example

	X	Y
P1	0.40	0.53
P2	0.22	0.38
P3	0.35	0.32
P4	0.26	0.19
P5	0.08	0.41
P6	0.45	0.30

Basic agglomerative hierarchical clustering algorithm

Calculate Euclidean distance, create the distance matrix.

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

$$\text{)istance (P1,P2)} = \sqrt{(0.40-0.22)^2+(0.53-0.38)^2}$$

$$\text{).40,0.53), (0.22,0.38)} = \sqrt{(0.18)^2+(0.15)^2}$$

$$= \sqrt{0.0324 + 0.0225}$$

$$= \sqrt{0.0549}$$

$$= 0.23$$

agglomerative hierarchical clustering

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

agglomerative hierarchical clustering

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

agglomerative hierarchical clustering

To update the distance matrix MIN[dist(P3,P6),P1)]

MIN(dist(P3,P1), (P6,P1))

= min[(0.22,0.23)]

= 0.22

To update the distance matrix MIN[dist(P3,P6),P2)]

MIN(dist(P3,P2), (P6,P2))

= min[(0.15,0.25)]

= 0.15

agglomerative hierarchical clustering

To update the distance matrix MIN[dist(P3,P6),P4)]

MIN(dist(P3,P4), (P6,P4))

= min[(0.15,0.22)]

= 0.15

To update the distance matrix MIN[dist(P3,P6),P5)]

MIN(dist(P3,P5), (P6,P5))

= min[(0.28,0.39)]

= 0.28

agglomerative hierarchical clustering

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

agglomerative hierarchical clustering

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

agglomerative hierarchical clustering

To update the distance matrix MIN[dist(P2,P5),P4)]

MIN[dist(P2,P4), (P5,P4)]

= min[(0.20,0.29)]

= 0.20

agglomerative hierarchical clustering

To update the distance matrix MIN[dist(P2,P5),P1)]

MIN[dist(P2,P1), (P5,P1)]

= min[(0.23,0.34)]

= 0.23

To update the distance matrix MIN[dist(P2,P5),(P3,P6)]

MIN[dist(P2,(P3,P6)), (P5,(P3,P6))]

= min[(0.15,0.28)]

= 0.15

agglomerative hierarchical clustering

	P1	P2,P5	P3,P6	P4
P1	0			
P2,P5	0.23	0		
P3,P6	0.22	0.15	0	
P4	0.37	0.20	0.15	0

agglomerative hierarchical clustering

	P1	P2,P5	P3,P6	P4
P1	0			
P2,P5	0.23	0		
P3,P6	0.22	0.15	0	
P4	0.37	0.2υ	0.15	0

agglomerative hierarchical clustering

To update the distance matrix MIN[dist((P2,P5),(P3,P6)),P1]

MIN[dist((P2,P5),P1), ((P3,P6),P1)]

= min[(0.23,0.22)]

= 0.22

agglomerative hierarchical clustering

To update the distance matrix MIN[dist((P2,P5),(P3,P6)),P4]

MIN[dist((P2,P5),P4), ((P3,P6),P4)]

= min[(0.20,0.15)]

= 0.15

agglomerative hierarchical clustering

	P1	P2,P5,P3,P6	P4
P1	0		
P2,P5,P3,P6	0.22	0	
P4	0.37	0.15	0

agglomerative hierarchical clustering

To update the distance matrix MIN[dist(P2,P5,P3,P6),P4]

MIN[dist((P2,P5,P3,P6),P1), (P4,P1)]

= min[(0.22,0.37)]

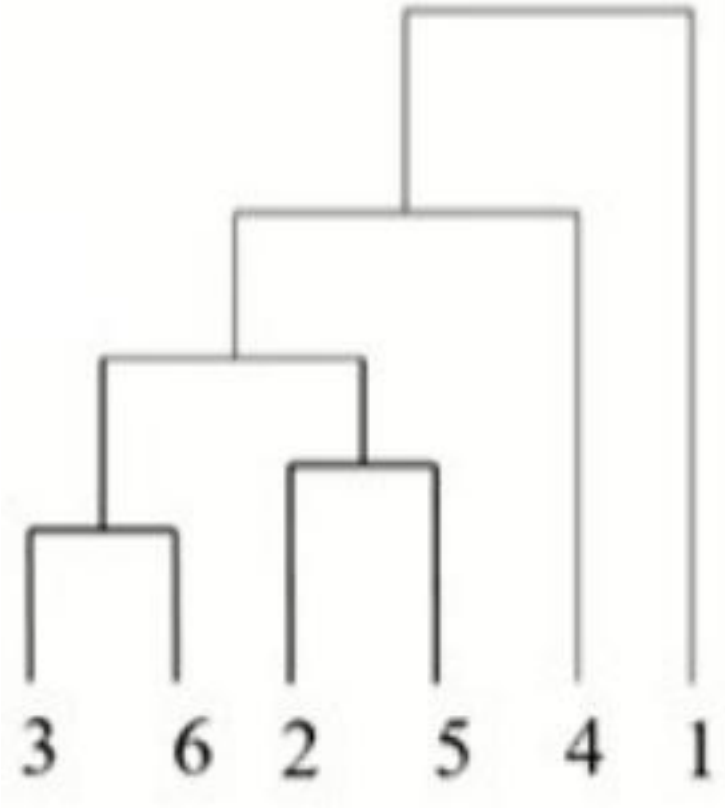
= 0.22

agglomerative hierarchical clustering

The updated distance matrix for cluster P2,P5,P3,P6,P4

	P1	P2,P5,P3,P6,P4
P1	0	
P2,P5,P3,P6,P4	0.22	0

agglomerative hierarchical clustering



Thank You