Subject: Machine Learning
Topic: Clustering

Prepared By
Prof. Sherin Mariam Jijo
IT Department

CVM UNIVERSITY

ADPATEL INSTITUTE OF TECHNOLOGY
ADIT
A D PATEL INSTITUTE OF TECHNOLOGY

Topic: Clustering

Contents:

- Introduction to clustering
- Types of clustering methods
- K-means
- Kmedoids
- Issues with clustering
- Applications of clustering

Unsupervised Machine Learning

- Unsupervised learning is the training of machine using information that is neither classified nor labeled and allowing the algorithm to act on that information without guidance.
- The task of machine is to group unsorted information according to similarities, patterns and differences without any prior training of data.
- Unsupervised learning algorithms are essentially complex algorithms, categorized as clustering.

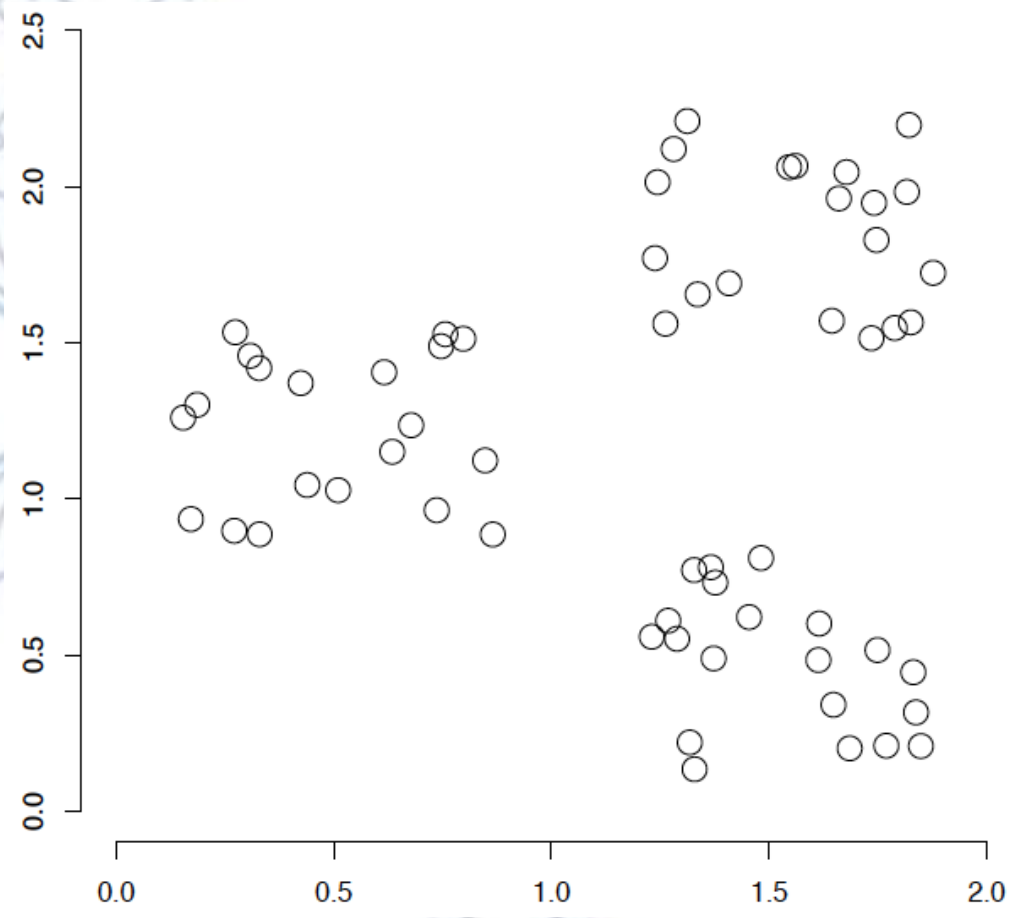
Clustering

- **Clustering:** the process of grouping a set of objects into classes of similar objects
  - Documents within a cluster should be similar.
  - Documents from different clusters should be dissimilar.

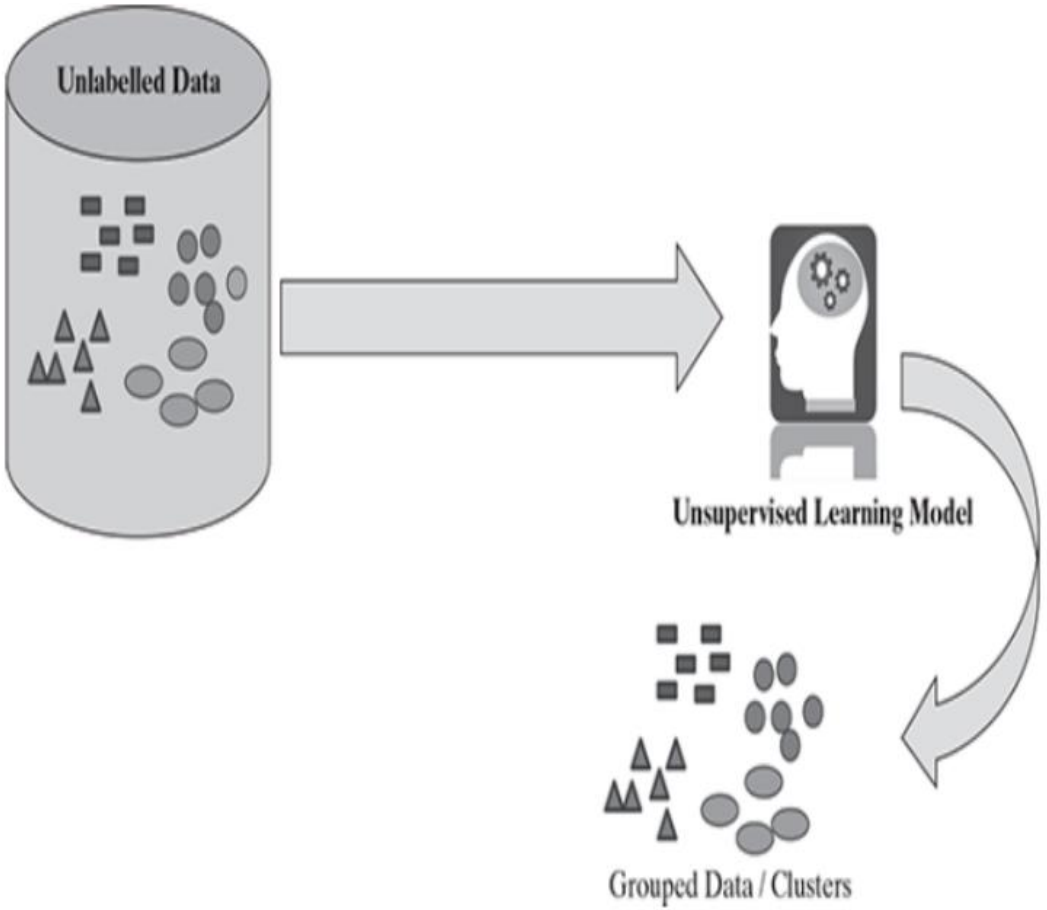
Clustering

- Data sets are divided into different groups in the cluster analysis, which is based on the similarity of the data.
- Clustering is a form of machine learning - the machine in this case is your computer, and learning refers to an algorithm that's repeated over and over until a certain set of predetermined conditions is met.
- Learning algorithms are generally run until the point that the final analysis results will not change, no matter how many additional times the algorithm is passed over the data.

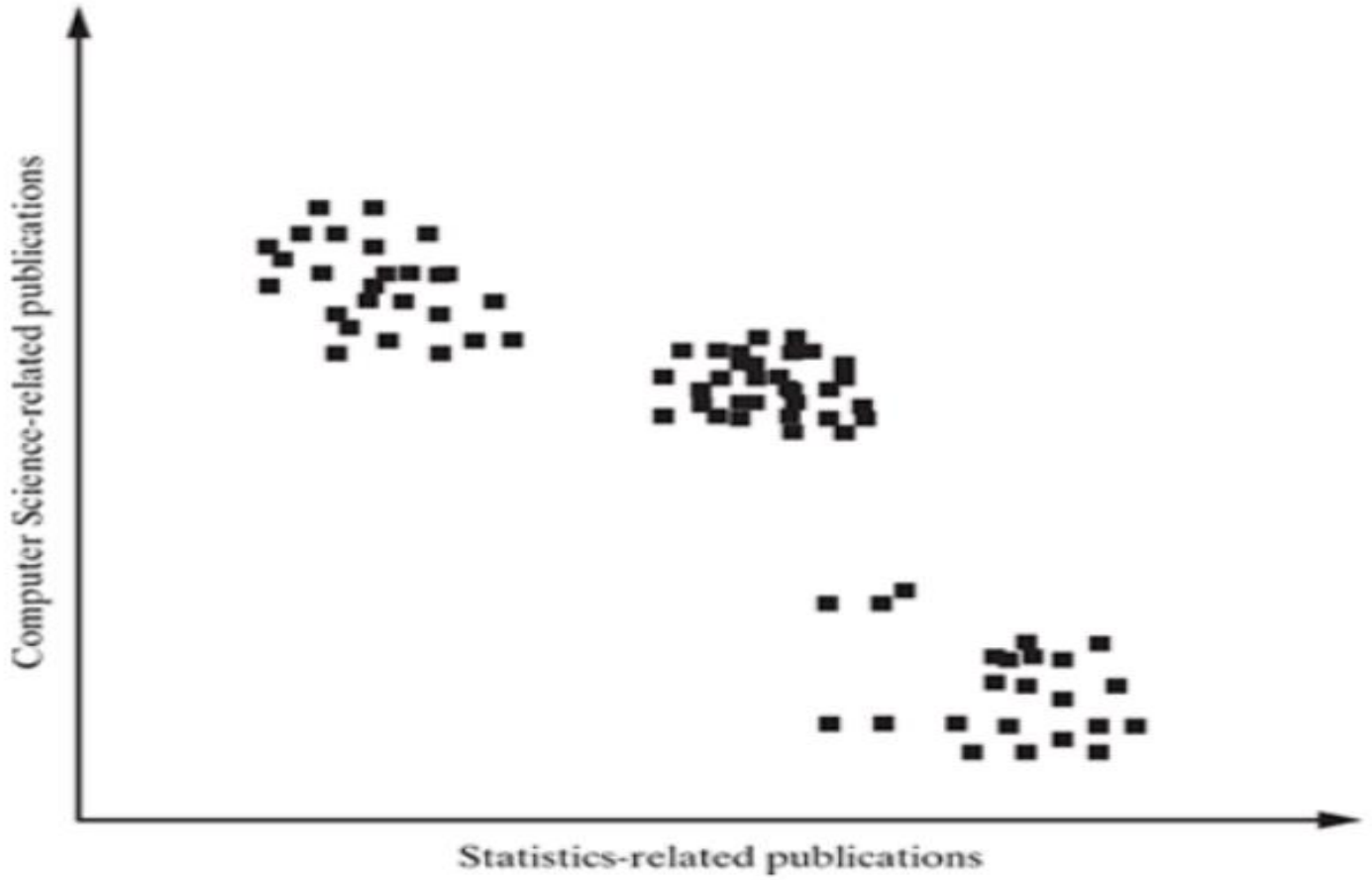
A data set with clear cluster structure



- How would you design an algorithm for finding the three clusters in this case?


Unlabelled Data
Unsupervised Learning Model
Grouped Data / Clusters

Data set for the conference attendees



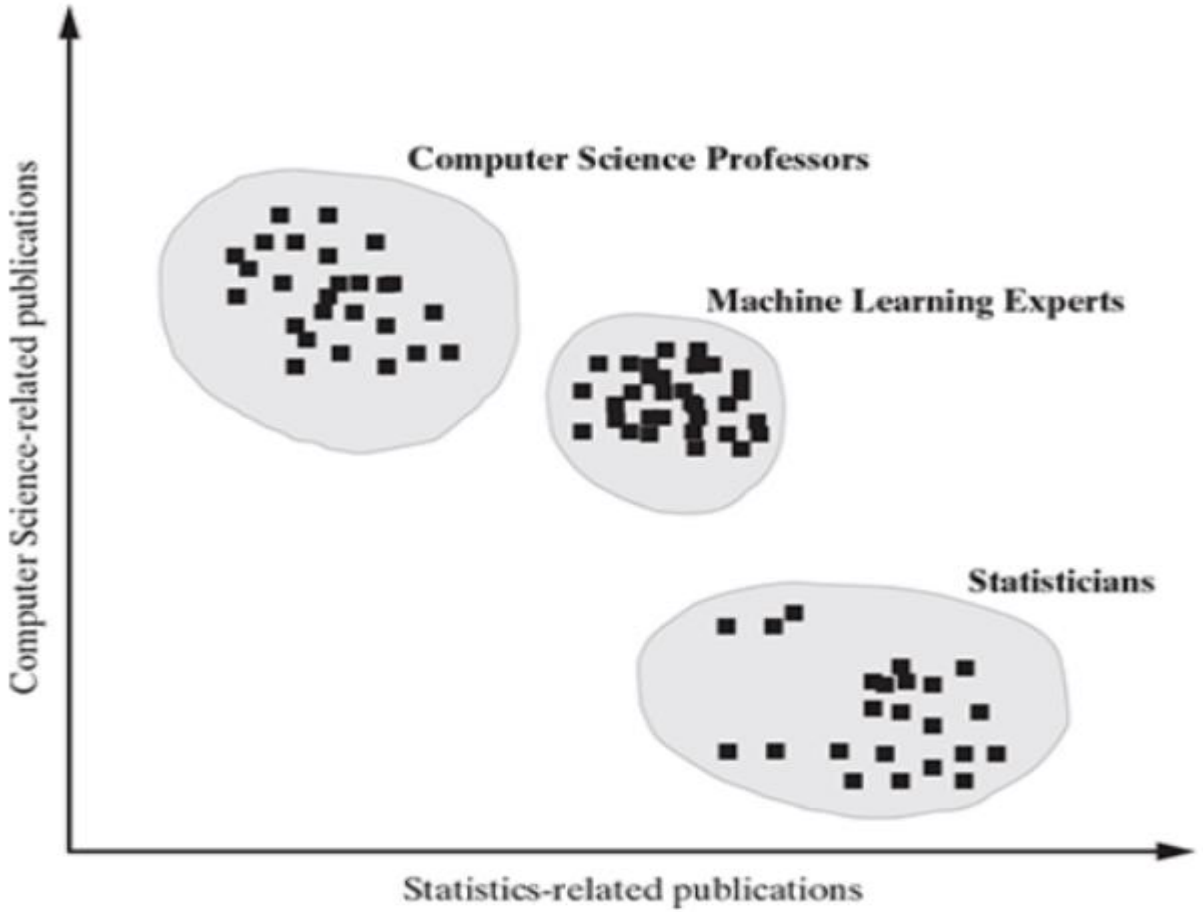


FIG. 9.3 Clusters for the conference attendees

- The primary driver of clustering knowledge is discovery rather than prediction.
- Clustering is defined as an unsupervised machine learning task that automatically divides the data into clusters or groups of similar items.
- The primary guideline of clustering task is that the data inside a cluster should be very similar to each other but very different from those outside the cluster.

- Through clustering, we are trying to label the objects with class labels.
- But clustering is somewhat different from the classification and numeric prediction discussed in supervised learning chapters.
- In each of these cases, the goal was to create a model that relates features to an outcome or to other features and the model identifies patterns within the data.
- In contrast, clustering creates new data.
- Unlabeled objects are given a cluster label which is inferred entirely from the relationship of attributes within the data.

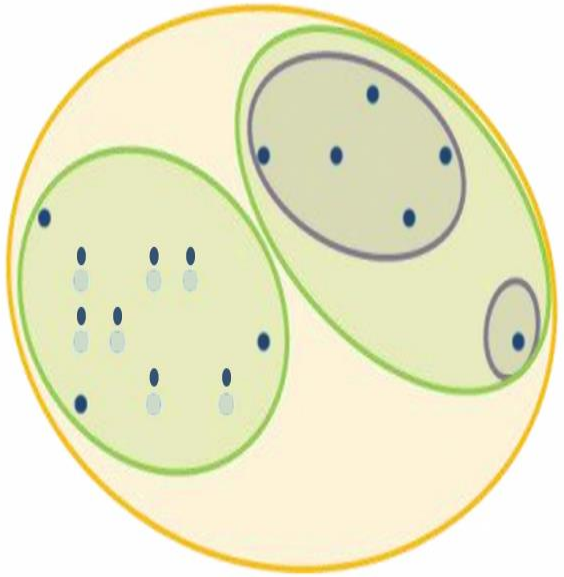
Hard vs. soft clustering

- Hard clustering: Each document belongs to exactly one cluster
  - More common and easier to do
- Soft clustering: A document can belong to more than one cluster.
  - Makes more sense for applications like creating browsable hierarchies
  - You may want to put a pair of sneakers in two clusters: (i) sports apparel and (ii) shoes
  - You can only do that with a soft clustering approach.

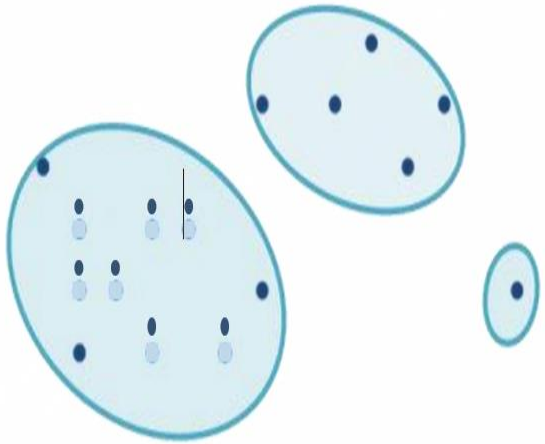
Clustering Techniques

- The major clustering techniques are
- Partitioning methods,
- Hierarchical methods, and
- Density-based methods.

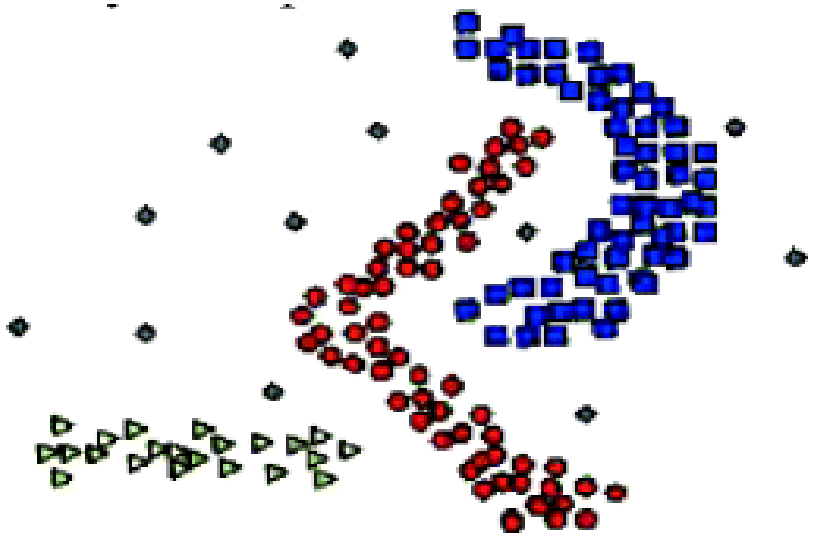
Hierarchical Clustering



Partitional Clustering



Density based Clustering

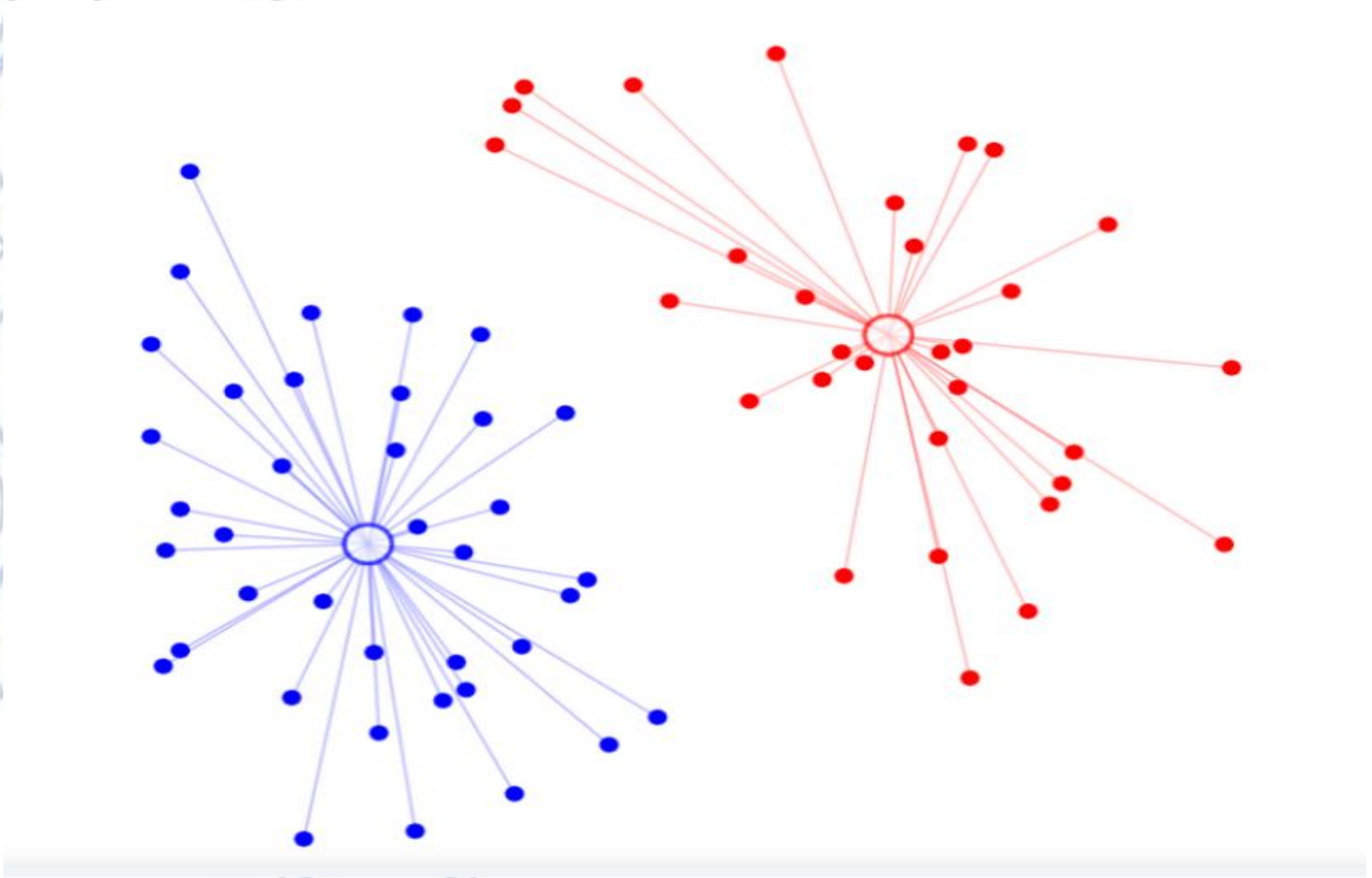


Method	Characteristics
Partitioning methods	• Uses mean or medoid (etc.) to represent cluster centre • Adopts distance-based approach to refine clusters • Finds mutually exclusive clusters of spherical or nearly spherical shape • Effective for data sets of small to medium size
Hierarchical methods	• Creates hierarchical or tree-like structure through decomposition or merger • Uses distance between the nearest or furthest points in neighbouring clusters as a guideline for refinement • Erroneous merges or splits cannot be corrected at subsequent levels
Density-based methods	• Useful for identifying arbitrarily shaped clusters • Guiding principle of cluster creation is the identification of dense regions of objects in space which are separated by low-density regions • May filter out outliers

Partitioning methods

- Partitioning method: Construct a partition of *n* documents into a set of *K* clusters
- Given: a set of documents and the number *K*
- Find: a partition of *K* clusters that optimizes the chosen partitioning criterion
  - Globally optimal.
  - Effective heuristic methods: *K*-means and *K*-medoids algorithms

- Two of the most important algorithms for partitioning based clustering are **k-means and k-medoid.**
- In the **k-means algorithm**, the centroid of the prototype is identified for clustering, which is normally the mean of a group of points.
- Similarly, the **k-medoid algorithm** identifies the medoid which is the most representative point for a group of points.
- We can also infer that in most cases, the centroid does not correspond to an actual data point, whereas medoid is always an actual data point.



K-means - A centroid-based technique

- The principle of the k-means algorithm is to assign each of the ‘n’ data points to one of the K clusters where ‘K’ is a user-defined parameter as the number of clusters desired.
- The objective is to maximize the homogeneity within the clusters and also to maximize the differences between the clusters.
- The homogeneity and differences are measured in terms of the distance between the objects or points in the data set.

Algorithm of K-means

Step 1: Select K points in the data space and mark them as initial centroids

loop

Step 2: Assign each point in the data space to the nearest centroid to form K clusters

Step 3: Measure the distance of each point in the cluster from the centroid

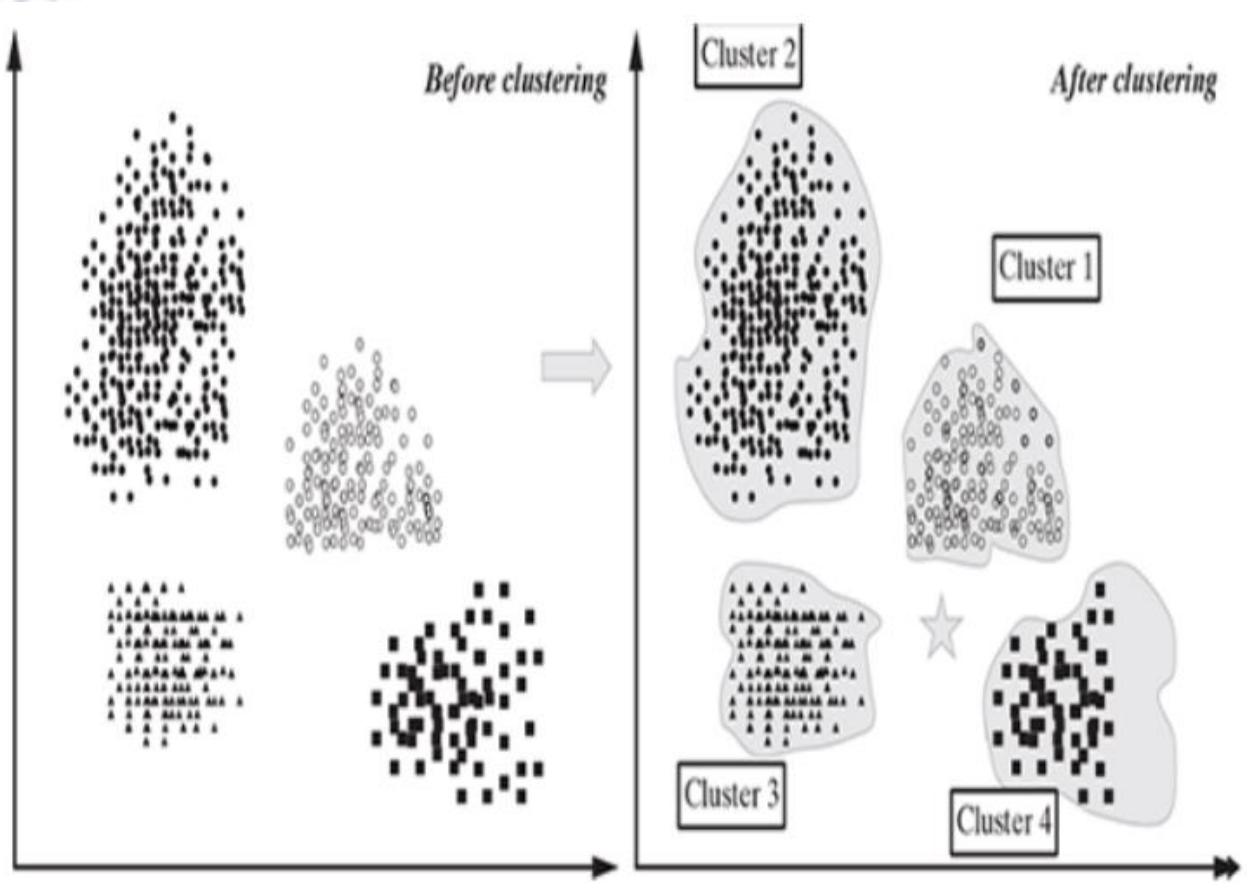
Step 4: Calculate the Sum of Squared Error (SSE) to measure the quality of the clusters.

Step 5: Identify the new centroid of each cluster on the basis of distance between points

Step 6: Repeat Steps 2 to 5 to refine until centroids do not change

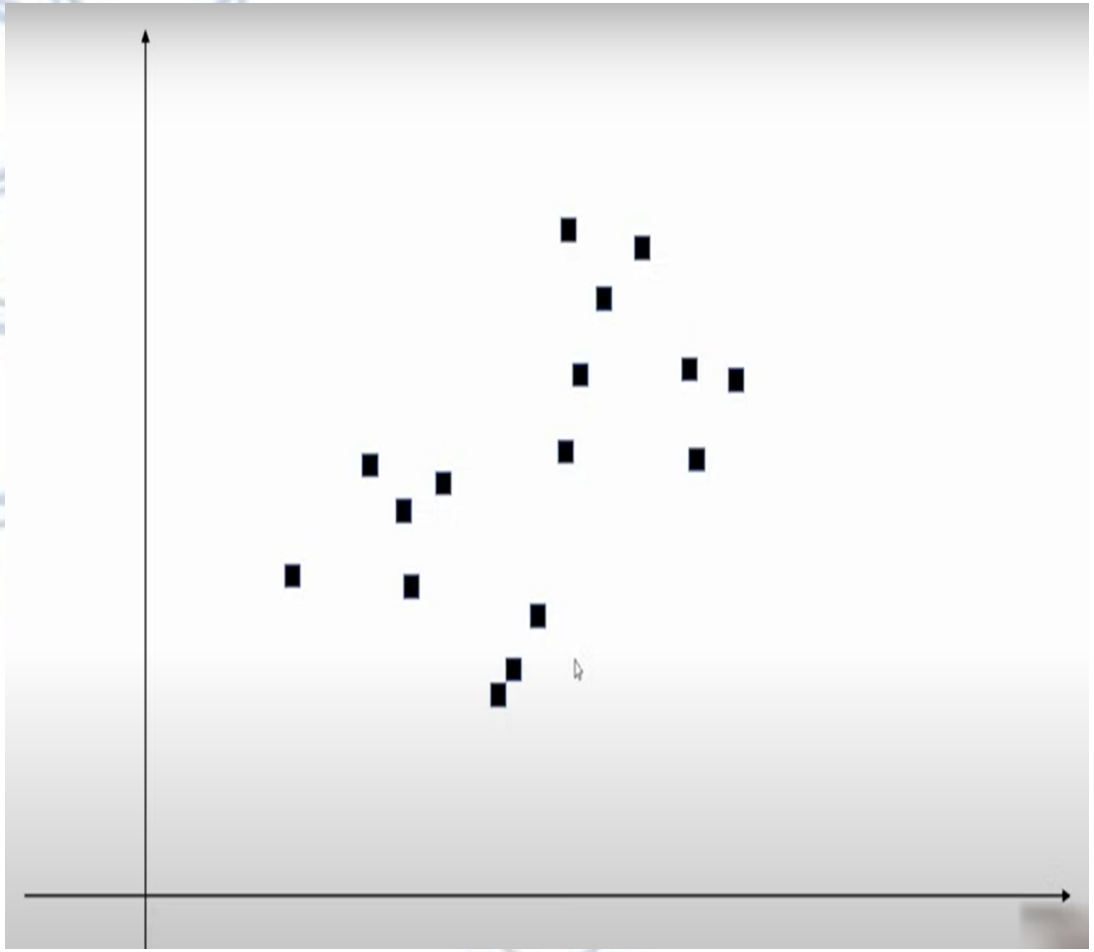
end loop

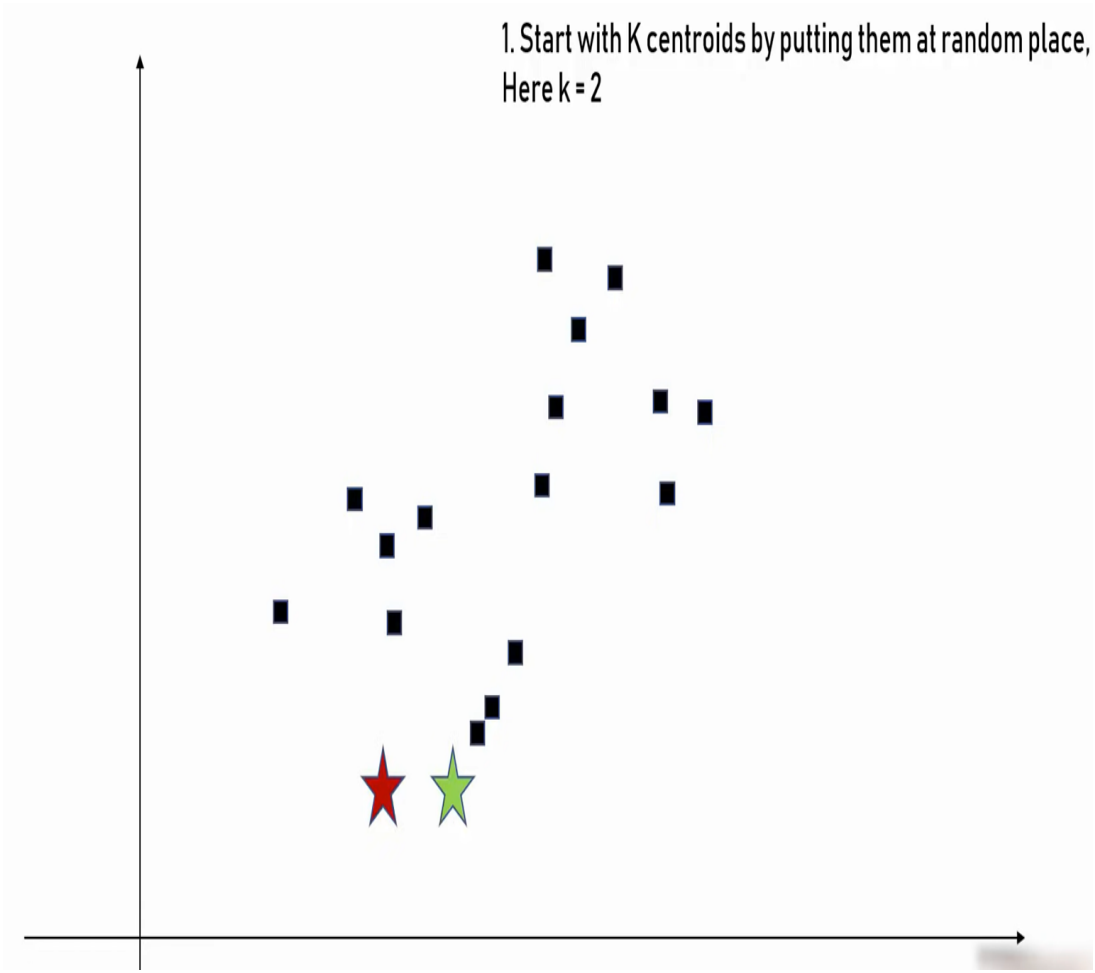
Example

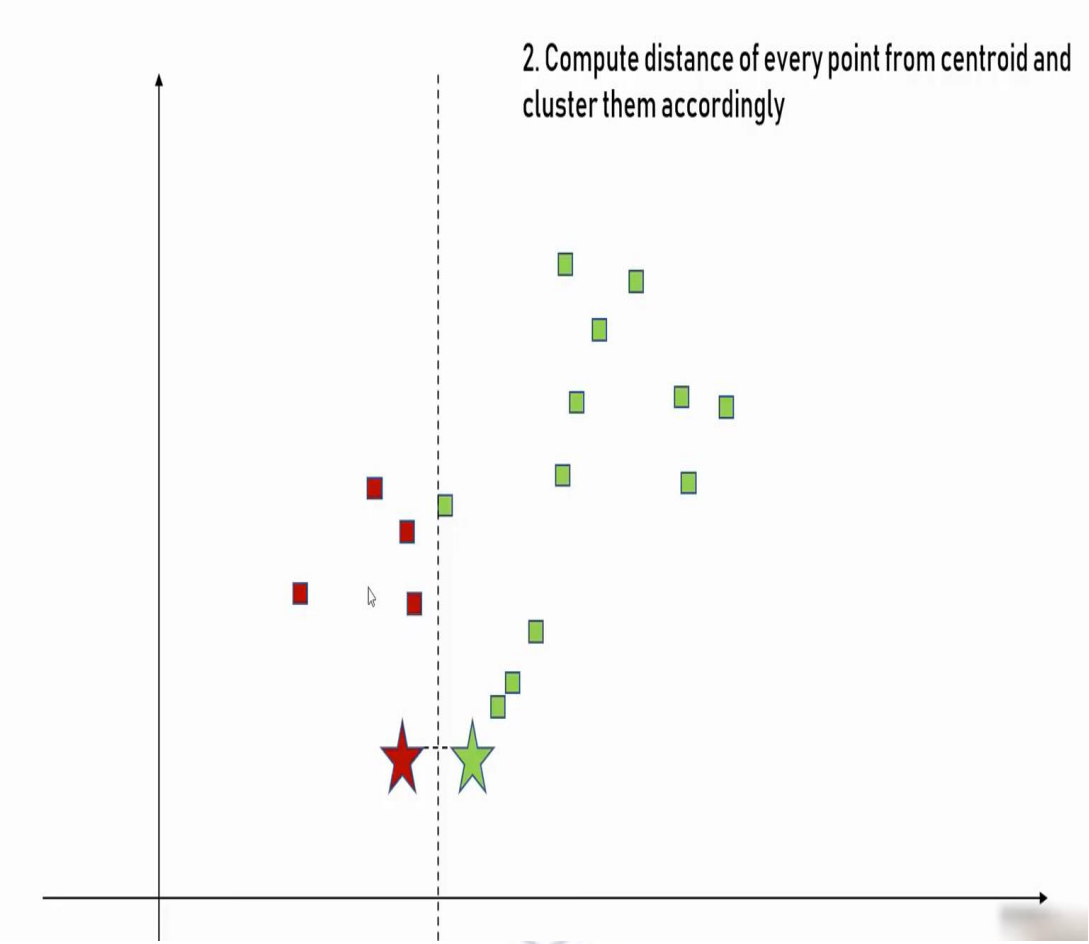

Before clustering
Cluster 2
After clustering
Cluster 1
Cluster 3
Cluster 4

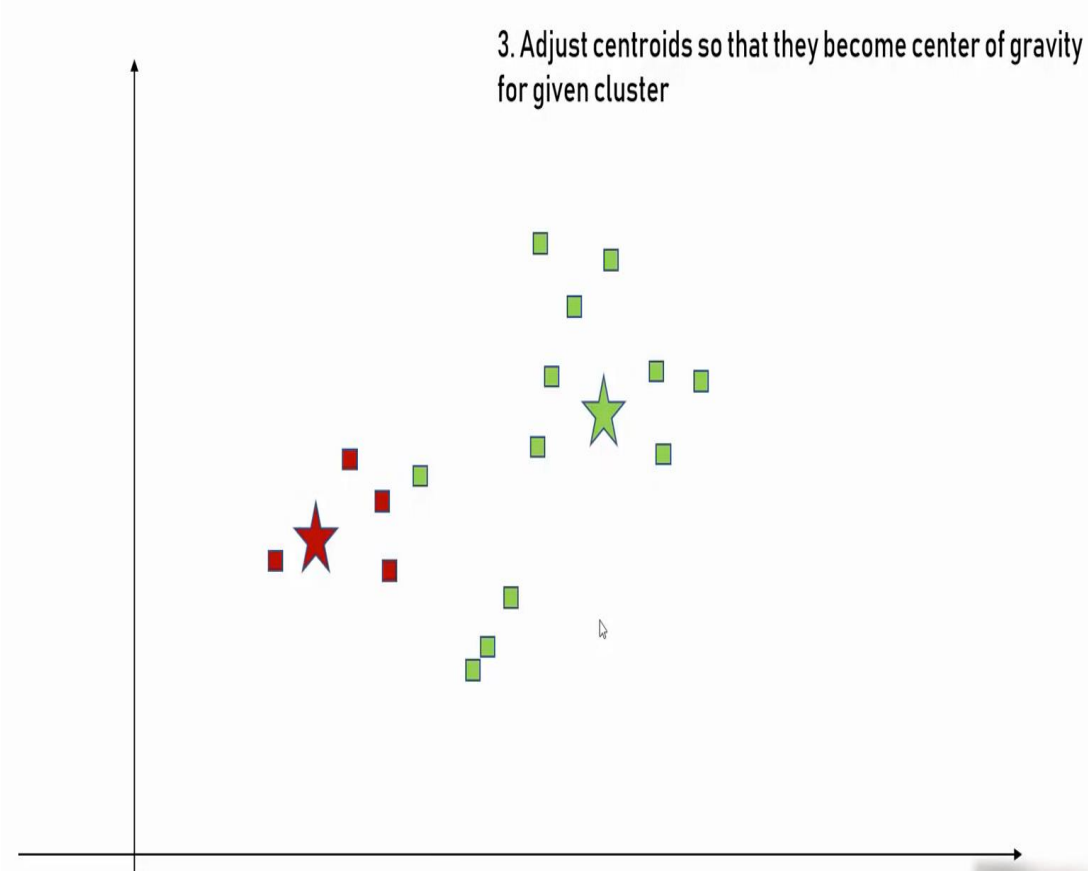
- Let us fix K = 4, implying that we want to create four clusters out of this data set.
- As the first step, we assign four random points from the data set as the centroids, as represented by the * signs, and we assign the data points to the nearest centroid to create four clusters.
- In the second step, on the basis of the distance of the points from the corresponding centroids, the centroids are updated and points are reassigned to the updated centroids.

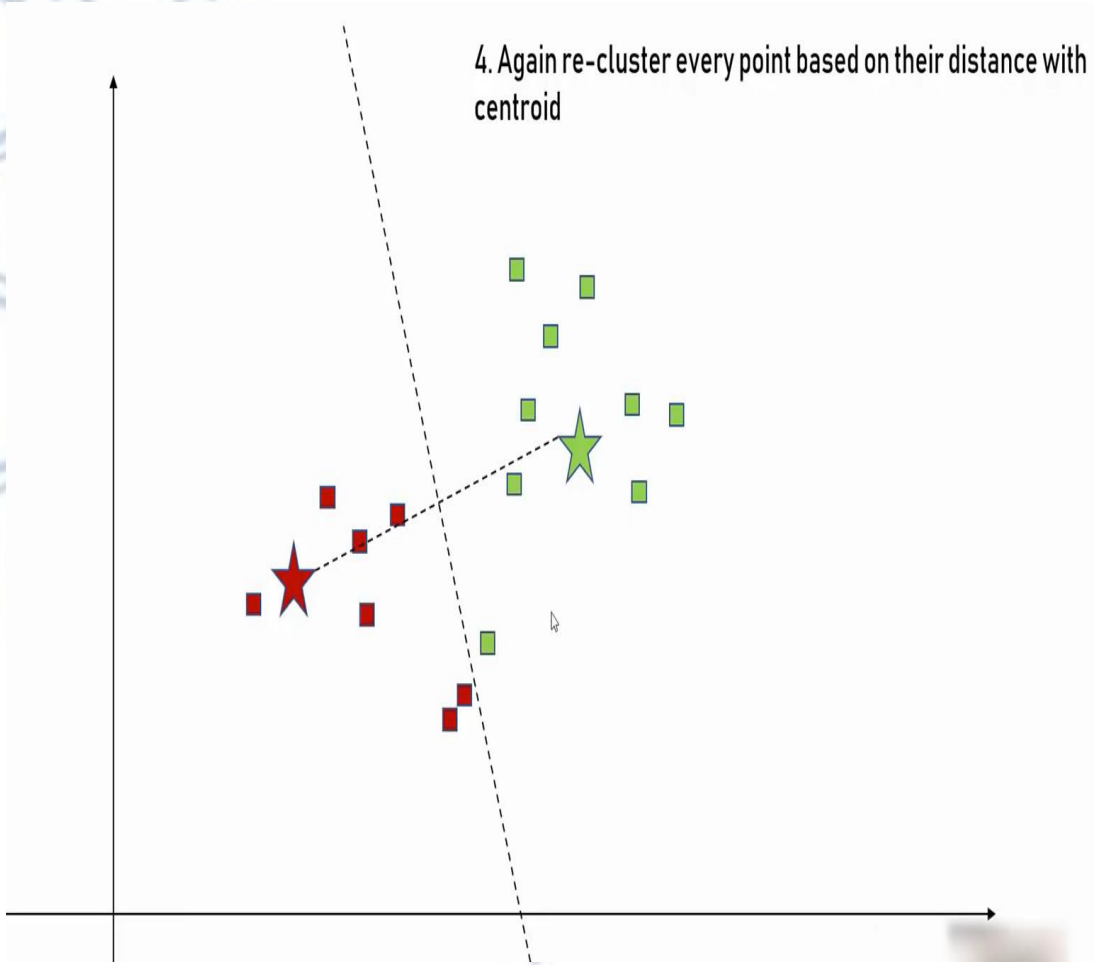
- After three iterations, we found that the centroids are not moving as there is no scope for refinement, and thus, the k-means algorithm will terminate.
- This provides us the most logical four groupings or cluster of the data sets where the homogeneity within the groups is highest and difference between the groups is maximum.

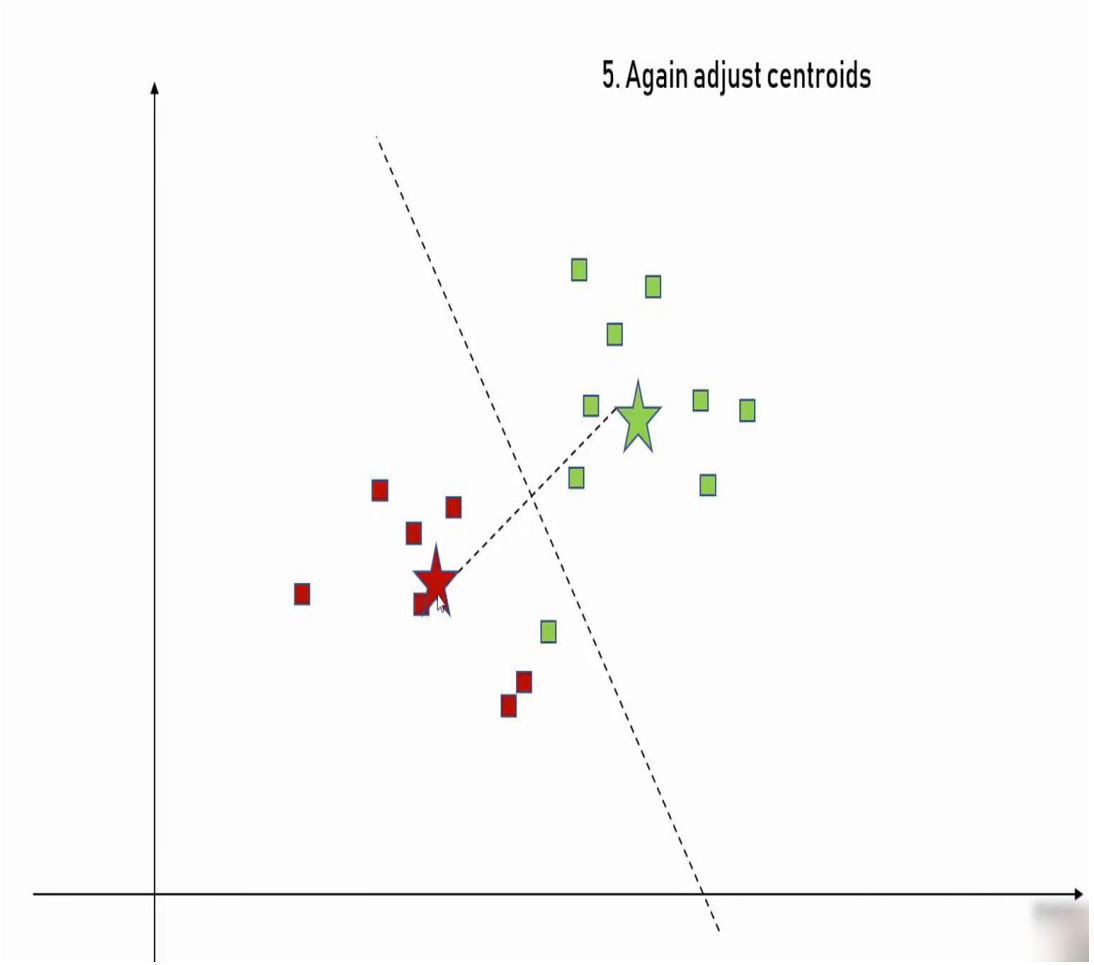


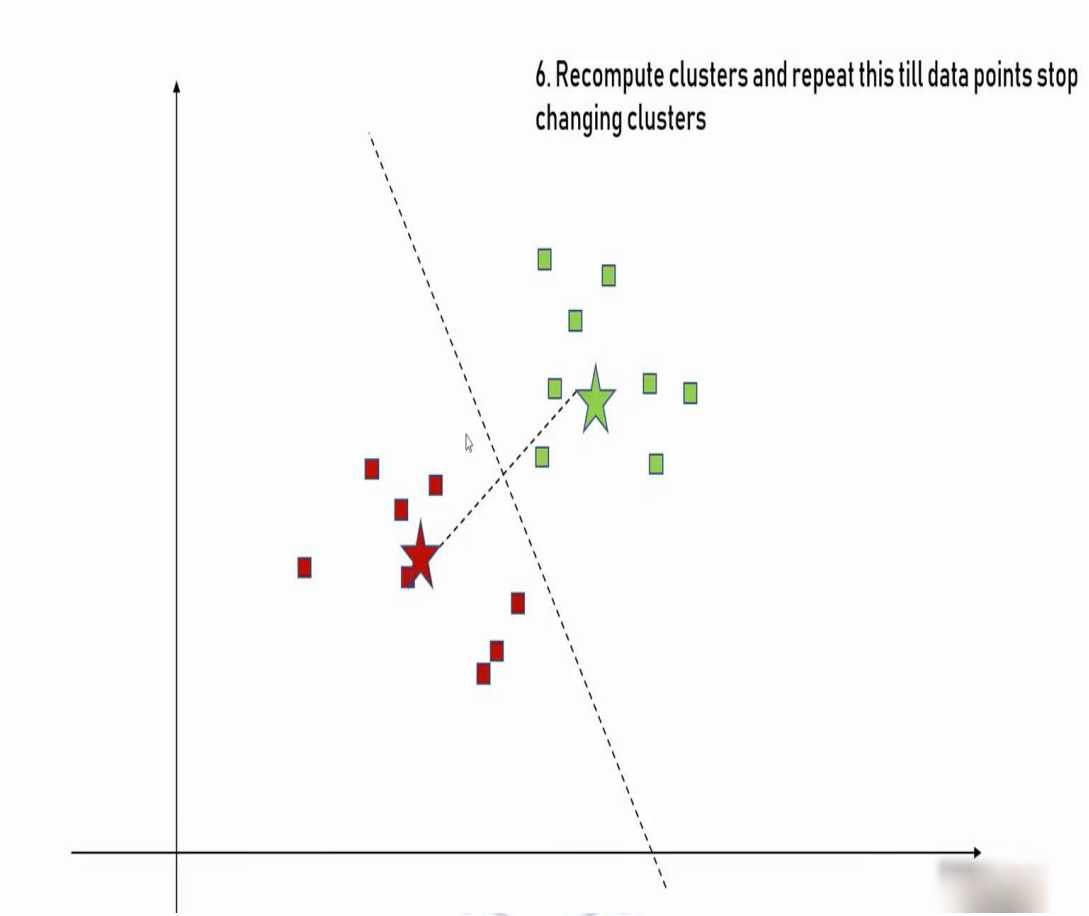

1. Start with K centroids by putting them at random place,
Here k = 2

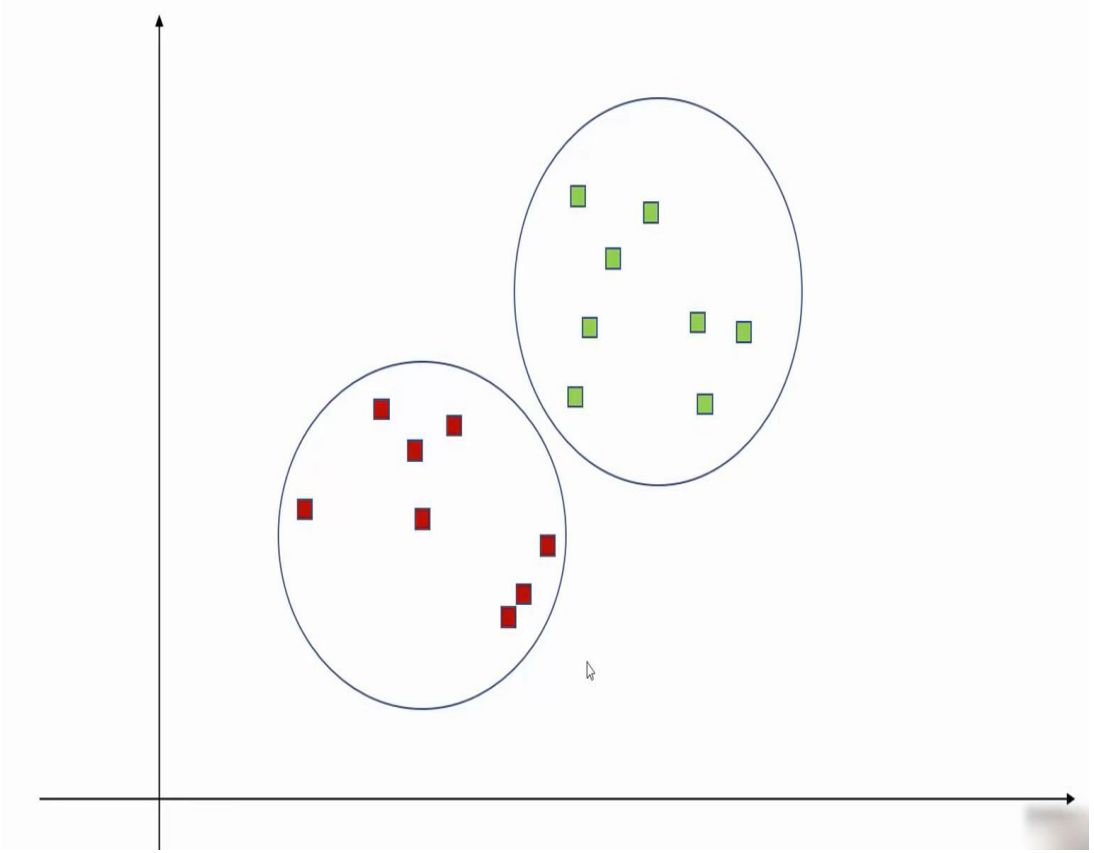

2. Compute distance of every point from centroid and cluster them accordingly


3. Adjust centroids so that they become center of gravity for given cluster


4. Again re-cluster every point based on their distance with centroid


5. Again adjust centroids


6. Recompute clusters and repeat this till data points stop changing clusters



- The k-means algorithm works by placing sample cluster centers on an n dimensional plot and then evaluating whether moving them in any single direction would result in a new center with higher density - with more observations closer to it.
- The centers are moved from regions of lower density to regions of higher density until all centers are within a region of local maximum density - a true center of the cluster, where each cluster gets a maximum number of points closest to its cluster center.

Strength & Weakness of K-means algm

Strengths	Weaknesses
• The principle used for identifying the clusters is very simple and involves very less complexity of statistical terms • The algorithm is very flexible and thus can be adjusted for most scenarios and complexities • The performance and efficiency are very high and comparable to those of any sophisticated algorithm in term of dividing the data into useful clusters	• The algorithm involves an element of random chance and thus may not find the optimal set of cluster in some cases • The starting point of guessing the number natural clusters within the data requires some experience of the user, so that the final outcome is efficient

Choosing appropriate number of clusters

- One of the most important success factors in arriving at correct clustering is to start with the correct number of cluster assumptions.
- Different numbers of starting cluster lead to completely different types of data split.
- It will always help if we have some prior knowledge about the number of clusters and we start our k-means algorithm with that prior knowledge.

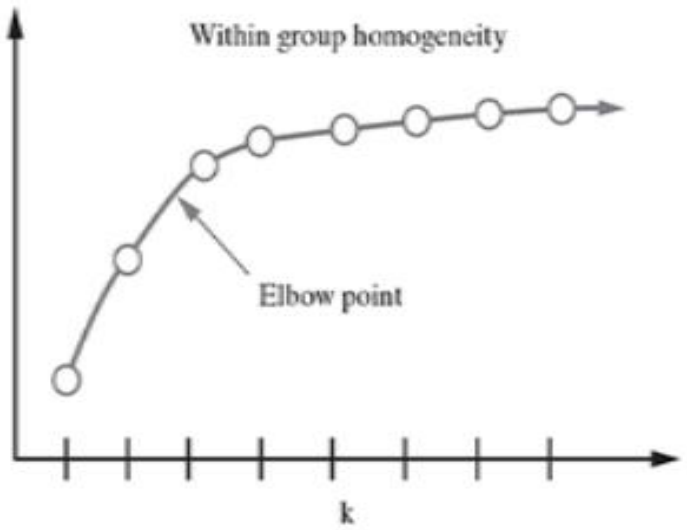
- For a small data set, sometimes a rule of thumb that is followed is:

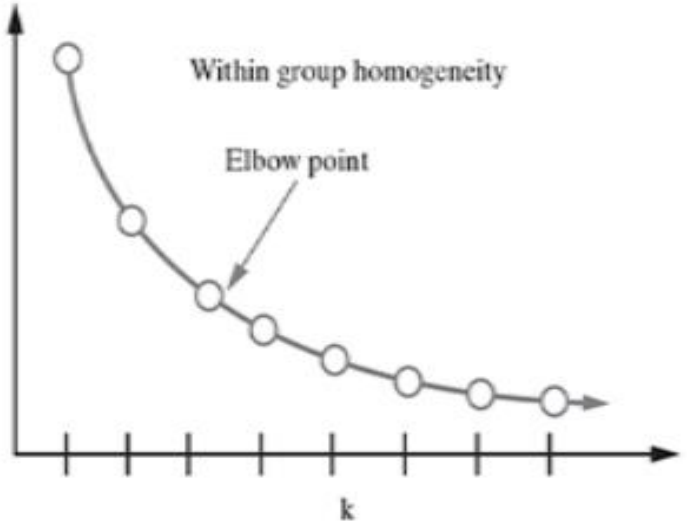
$$K = \sqrt{\frac{n}{2}}$$

- which means that K is set as the square root of n/2 for a data set of n examples.
- But unfortunately, this thumb rule does not work well for large data sets.
- There are several statistical methods to arrive at the suitable number of clusters.

Elbow method

- Tries to measure the homogeneity or heterogeneity within the cluster and for various values of ‘K’ and helps in arriving at the optimal ‘K’.
- The homogeneity will increase or heterogeneity will decrease with increasing ‘K’ as the number of data points inside each cluster reduces with this increase.
- But these iterations take significant computation effort, and after a certain point, the increase in homogeneity benefit is no longer in accordance with the investment required to achieve it, as is evident from the figure.
- This point is known as the elbow point, and the ‘K’ value at this point produces the optimal clustering performance.


Within group homogeneity
Elbow point
k


Within group homogeneity
Elbow point
k

Choosing the initial centroids

- One common practice is to choose random points in the data space on the basis of the number of cluster requirement and refine the points as we move into the iterations.
- But this often leads to higher squared error in the final clustering, thus resulting in sub-optimal clustering solution.
- The assumption for selecting random centroids is that multiple subsequent runs will minimize the SSE and identify the optimal clusters.
- But this is often not true on the basis of the spread of the data set and the number of clusters sought.

- One effective approach is to employ the hierarchical clustering technique on sample points from the data set and then arrive at sample K clusters.
- The centroids of these initial K clusters are used as the initial centroids.
- This approach is practical when the data set has small number of points and K is relatively small compared to the data points.
- There are procedures such as bisecting k-means and use of post-processing to fix initial clustering issues; these procedures can produce better quality initial centroids and thus better SSE for the final clusters.

Recomputing cluster centroids

- In the k-means algorithm, the iterative step is to recalculate the centroids of the data set after each iteration.
- The proximities of the data points from each other within a cluster is measured to minimize the distances.
- The distance of the data point from its nearest centroid can also be calculated to minimize the distances to arrive at the refined centroid.
- The Euclidean distance between two data points is measured as follows:

$$\text{dist}(x, y) = \sqrt{\sum_{1}^{n} (x_i - y_i)^2}$$

- The measure of quality of clustering uses the SSE technique. The formula used is as follows:

$$\mathrm{SSE} = \sum_{i=1}^{k} \sum_{x \in C_i} \mathrm{dist}\,(c_i, x)^2$$

- where dist() calculates the Euclidean distance between the centroid c of the cluster C and the data points x in the cluster.
- The summation of such distances over all the ‘K’ clusters gives the total sum of squared error.

- The lower the SSE for a clustering solution, the better is the representative position of the centroid.
- It is observed that the centroid that minimizes the SSE of the cluster is its mean.
- One limitation of the squared error method is that in the case of presence of outliers in the data set, the squared error can distort the mean value of the clusters.

- Because of the distance-based approach from the centroid to all points in the data set, the k-means method may not always converge to the global optimum and often terminates at a local optimum.
- The result of the clustering largely depends on the initial random selection of cluster centres.

- The complexity of the k-means algorithm is O (nKt), where ‘ n ’ is the total number of data points or objects in the data set, K is the number of clusters, and ‘ t ’ is the number of iterations.
- Normally, ’K ’ and ‘ t ’ are kept much smaller than ‘ n ’, and thus, the k-means method is relatively scalable and efficient in processing large data sets.

Issues of clustering

- Representation for clustering
  - Document representation
    - Vector space? Normalization?
      - Centroids aren’t length normalized
  - Need a notion of similarity/distance
- How many clusters?
  - Fixed a priori?
  - Completely data driven?
    - Avoid “trivial” clusters - too large or small
      - If a cluster's too large, then for navigation purposes you've wasted an extra user click without whittling down the set of documents much.

K-Medoids

- k-means algorithm is sensitive to outliers in the data set and inadvertently produces skewed clusters when the means of the data points are used as centroids.
- Let us take an example of eight data points, and for simplicity, we can consider them to be 1D data with values 1, 2, 3, 5, 9, 10, 11, and 25.
- Point 25 is the outlier, and it affects the cluster formation negatively when the mean of the points is considered as centroids.

- With K = 2, the initial clusters we arrived at are {1, 2,3, 6} and {9, 10, 11, 25}.
- The mean of the cluster $\{1, 2, 3, 6\} = \frac{12}{4} = 3,$
- and the mean of the cluster $\{9, 10, 12, 25\} = \frac{56}{4} = 14.$
- So, the SSE within the clusters is

$$(1-3)^2 + (2-3)^2 + (3-3)^2 + (6-3)^2 + (9-14)^2 + (10-14)^2 + (12-14)^2 + (25-14)^2 = 179$$

- If we compare this with the cluster {1, 2, 3, 6, 9} and{10, 11, 25},
- the mean of the cluster $\{1, 2, 3, 6, 9\} = \frac{21}{5} = 4.2,$
- and the mean of the cluster $\{10, 12, 25\} = \frac{47}{3} = 15.67.$

- So, the SSE within the clusters is

$$(1 - 4.2)^2 + (2 - 4.2)^2 + (3 - 4.2)^2 + (6 - 4.2)^2 + (9 - 4.2)^2$$
$$+ (10 - 15.67)^2 + (12 - 15.67)^2 + (25 - 15.67)^2 = 113.84$$

- Because the SSE of the second clustering is lower, k means tend to put point 9 in the same cluster with 1, 2, 3, and 6 though the point is logically nearer to points 10 and 11.
- This skewedness is introduced due to the outlier point 25, which shifts the mean away from the centre of the cluster.
- k-medoids provides a solution to this problem. Instead of considering the mean of the data points in the cluster, k-medoids considers k representative data points from the existing points in the data set as the centre of the clusters.
- It then assigns the data points according to their distance from these centres to form k clusters.

- The SSE is calculated as $$\text{SSE} = \sum_{i=1}^{k} \sum_{x \in C_i} \text{dist}\,(o_i, x)^2$$

- where oi is the representative point or object of cluster
- Ci.

- Thus, the k-medoids method groups n objects in k clusters by minimizing the SSE.
- Because of the use of medoids from the actual representative data points, k medoids is less influenced by the outliers in the data.
- One of the practical implementation of the k-medoids principle is the Partitioning Around Medoids (PAM).

- Step 1: Randomly choose k points in the data set as the initial representative points.

loop

- Step 2: Assign each of the remaining points to the cluster which has

the nearest representative point.

- Step 3: Randomly select a non-representative point r in each cluster.
- Step 4: Swap the representative point j with r and compute the new

SSE after swapping.

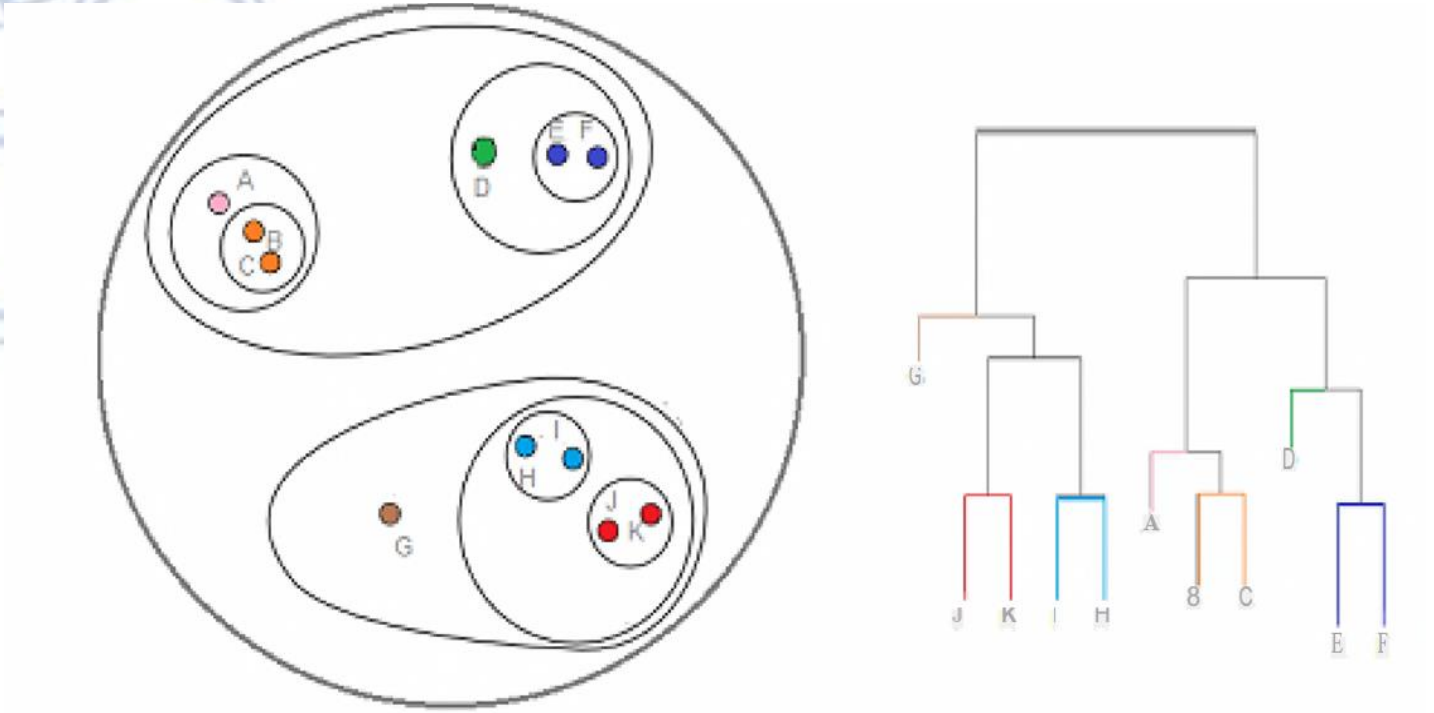
- Step 5: If SSEnew < SSEold, then swap j with r to form the new set of k representative objects;
- Step 6: Refine the k clusters on the basis of the nearest representative point. Logic continues until there is no change.
- end loop

- Though the k-medoids algorithm provides an effective way to eliminate the noise or outliers in the data set, which was the problem in the k-means algorithm, it is expensive in terms of calculations.
- The complexity of each iteration in the k-medoids algorithm is O(k(n - k)).
- For large value of ‘n’ and ‘k’, this calculation becomes much costlier than that of the k-means algorithm.

Hierarchical algorithms

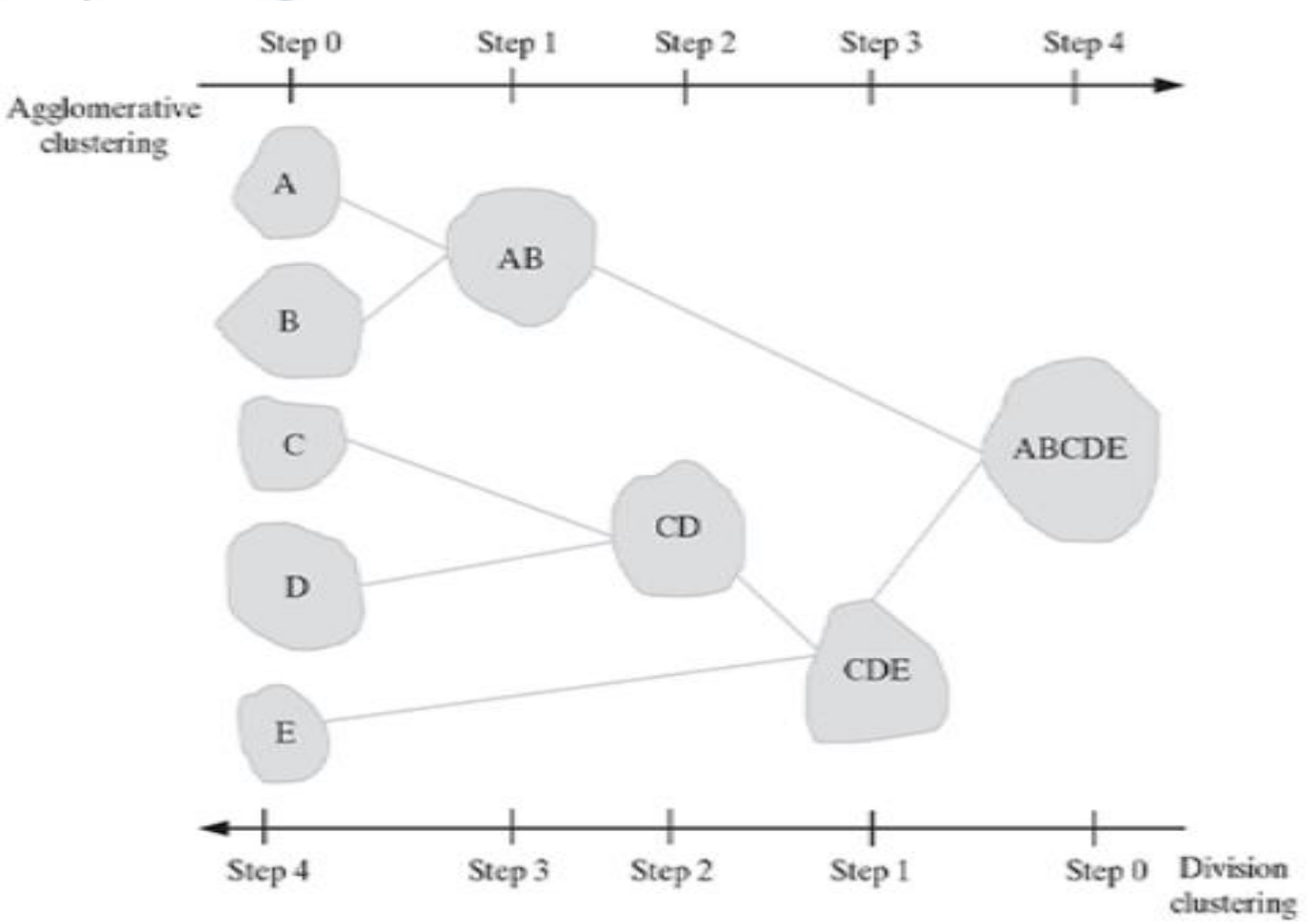
- The hierarchical clustering methods are used to group the data into hierarchy or tree-like structure.
- It predicts groupings within a dataset by calculating the distance and generating a link between each singular observation and its nearest neighbor.
- It then uses those distances to predict subgroups within a dataset.
- If carrying out a statistical study or analyzing biological or environmental data, hierarchical clustering might be your ideal machine learning solution.
- To visually inspect the results of your hierarchical clustering, generate a ***dendrogram*** **-** a visualization tool that depicts the similarities and branching between groups in a data cluster(Fig).

- Use several different algorithms to build a dendrogram, and the algorithm you choose dictates where and how branching occurs within the clusters.
-



- In hierarchical clustering, the distance between observations is measured in three different ways: Euclidean, Manhattan, or Cosine.
- Hierarchical clustering algorithms are more computationally expensive than k-means algorithms because with each iteration of hierarchical clustering, many observations must be compared to many other observations.
- Weakness: In comparison to k-means clustering, the hierarchical clustering algorithm is a slower, chunkier unsupervised clustering algorithm.
- However, the benefit, is that hierarchical clustering algorithms are not subject to errors caused by center convergence at areas of local minimum density (as exhibited with the k-means clustering algorithms).

- There are two main hierarchical clustering methods: **agglomerative clustering and divisive clustering.**
- Agglomerative clustering is a bottom-up technique which starts with individual objects as clusters and then iteratively merges them to form larger clusters.
- On the other hand, the divisive method starts with one cluster with all given objects and then splits it iteratively to form smaller clusters. See Figure on next slide.


Step 0
Step 1
Step 2
Step 3
Step 4
Agglomerative clustering
A
B
AB
C
D
CD
E
CDE
ABCDE
Step 4
Step 3
Step 2
Step 1
Step 0
Division clustering

Density based Methods

- Density-based spatial clustering of applications with noise (DBScan) is an unsupervised learning method that works by clustering core samples (dense areas of a dataset) while simultaneously demarking non-core samples (portions of the dataset that are comparatively sparse).

- When we used the partitioning and hierarchical clustering methods, the resulting clusters are spherical or nearly spherical in nature.
- In the case of the other shaped clusters such as S shaped or uneven shaped clusters, the above two types of method do not provide accurate results.
- The density based clustering approach provides a solution to identify clusters of arbitrary shapes.
- The principle is based on identifying the dense area and sparse area within the data set and then run the clustering algorithm.
- DBSCAN is one of the popular density-based algorithm which creates clusters by using connected regions with high density.

- DBSCAN is one of the density-based clustering approaches that provide a solution to identify clusters of arbitrary shapes.
- The principle is based on identifying the dense area and sparse area within the data set and then running the clustering algorithm.

Applications of clustering

- Text data mining
- Market segmentation
- Anomaly detection
- Data Mining
- Image processing and segmentation
- Identification of human errors during data entry
- Conducting accurate basket analysis, etc.
- Recommendation engines

Pblms

1) Apply k-means algorithm in given data for k=3. Use C1(2), C2(16), c3(38) as initial cluster centres.

- Data 2,4,6,3,31,12,15,16,38,35,14,21,23,25,30

Soln:

Calculating the distance between each data point and cluster centres, we get the following table.(Next slide)

Data Points	Distance from $C_1(2)$	Distance from $C_2(16)$	Distance from $C_3(3$
2	$(2-2)^2=0$	$(2-16)^2=196$	$(2-38)^2=1296$
4	$(4-2)^2=4$	$(4-16)^2=144$	$(4-38)^2=1156$
6	$(6-2)^2=16$	$(6-16)^2=100$	$(6-38)^2=1024$
3	$(3-2)^2=1$	$(3-16)^2=169$	$(3-38)^2=1225$
31	$(31-2)^2=841$	$(31-16)^2=225$	$(31-38)^2=49$
12	$(12-2)^2=100$	$(12-16)^2=16$	$(12-38)^2=676$
15	$(15-2)^2=169$	$(15-16)^2=1$	$(15-38)^2=529$
16	$(16-2)^2=196$	$(16-16)^2=0$	$(16-38)^2=484$
38	$(38-2)^2=1296$	$(38-16)^2=484$	$(38-38)^2=0$
35	$(35-2)^2=1089$	$(35-16)^2=361$	$(35-38)^2=9$
14	$(14-2)^2=144$	$(14-16)^2=4$	$(14-38)^2=576$
21	$(21-2)^2=361$	$(21-16)^2=25$	$(21-38)^2=289$
23	$(23-2)^2=441$	$(23-16)^2=49$	$(23-38)^2=225$
25	$(25-2)^2=529$	$(25-16)^2=81$	$(25-38)^2=169$
30	$(30-2)^2=784$	$(30-16)^2=196$	$(30-38)^2=64$

gning the data

By assigning the data points to the cluster center whose distance from it is minimum of all the cluster centers, we get the following table.

$C_1(2)$	$C_2(16)$	$C_3(38)$
m1 = 2	m2 = 16	m3 = 38
{2, 3, 4, 6}	{12, 14, 15, 16, 21, 23, 25}	{31, 35, 38}
New cluster centers		
m1 = 3.75	m2 = 18	m3 = 34.67

- Similarly, using the new cluster centers we can calculate the distance from it and allocate clusters based on minimum distance.
- It is found that there is no difference in the cluster formed and hence we stop this procedure.
- The final clustering result is given in the following table.

$C_1(3.75)$	$C_2(18)$	$C_3(34.67)$
m1 = 3.75	m2 = 18	m3 = 34.67
{2, 3, 4, 6}	{12, 14, 15, 16, 21, 23, 25}	{31, 35, 38}

2) Apply k-means clustering for the datasets given in table below. Tabulate all the assignments.

Sample No.	X	Y
1	185	72
2	170	56
3	168	60
4	179	68
5	182	72
6	188	77

Soln:

Sample No.	X	Y	Assignment
1	185	72	C1
2	170	56	C2

Centroid: $C1 = (185, 72)$ and $C2 = (170, 56)$

First Iteration:

Distance from $C1$ is Euclidean distance between $(185, 72)$ and $(168, 60) = 20.808$

Distance from $C2$ is Euclidean distance between $(170, 56)$ and $(168, 60) = 4.472$

Since $C2$ is closer to $(168, 60)$, the sample belongs to $C2$.

Sample No.	X	Y	Assignment
1	185	72	C1
2	170	56	C2
3	168	60	C2
4	179	68	
5	182	72	
6	188	77	

Similarly,

1. Distance from $C1$ for $(179, 68) = 7.21$
   Distance from $C2$ for $(179, 68) = 15$
   Since $C1$ is closer to $(179, 68)$, the sample belongs to $C1$.
2. Distance from $C1$ for $(182, 72) = 3$
   Distance from $C2$ for $(182, 72) = 20$
   Since $C1$ is closer to $(182, 72)$, the sample belongs to $C1$.
3. Distance from $C1$ for $(188, 77) = 5.83$
   Distance from $C2$ for $(188, 77) = 27.66$
   Since $C1$ is closer to $(188, 77)$, the sample belongs to $C1$.

Sample No.	*X*	*Y*	*Assignment*
1	185	72	$C1$
2	170	56	$C2$
3	168	60	$C2$
4	179	68	$C1$
5	182	72	$C1$
6	188	77	$C1$

ne new centroid for $C1$ is

$$\left(\frac{185+179+182+188}{4}, \frac{72+68+72+77}{4}\right) = (183.5, 73)$$

The new centroid for $C2$ is

$$\left(\frac{170+168}{2}, \frac{56+60}{2}\right) = (169, 58)$$

Second Iteration:

Distance from $C1$ is Euclidean distance between (183.5, 73) and (168, 60) = 20.2

Distance from $C2$ is Euclidean distance between (169, 58) and (168, 60) = 2.24

Since $C2$ is closer to (168, 60), the sample belongs to $C2$.

Similarly,

1. Distance from $C1$ for (179, 68) = 6.73
   Distance from $C2$ for (179, 68) = 14.14
   Since $C1$ is closer to (179, 68), the sample belongs to $C1$.
2. Distance from $C1$ for (182, 72) = 1.80
   Distance from $C2$ for (182, 72) = 19.10
   Since $C1$ is closer to (182, 72), the sample belongs to $C1$.
3. Distance from $C1$ for (188, 77) = 6.02
   Distance from $C2$ for (188, 77) = 26.87
   Since $C1$ is closer to (188, 77), the sample belongs to $C1$.

Sample No.	*X*	*Y*	*Assignment*
1	185	72	$C1$
2	170	56	$C2$
3	168	60	$C2$
4	179	68	$C1$
5	182	72	$C1$
6	188	77	$C1$

After the second iteration, the assignment has not changed and hence the algm is stopped and the points are clustered.

3) Apply k-medoid algorithm to cluster the following dataset of 6 objects into two clusters, that is k=2.

Solution:

Step 1: Two observations c1 = X2 = (3, 4) and c2 = X6 = (6, 4) are randomly selected as medoids (cluster centers).

Step 2: Manhattan distances are calculated to each center to associate each data object to its nearest medoid.

Data Object		*Distance To*	
Sample	*Point*	*c1 = (3, 4)*	*c2 = (6, 4)*
X1	(2, 6)	3	6
X2	(3, 4)	0	3
X3	(3, 8)	4	7
X4	(4, 2)	3	4
X5	(6, 2)	5	2
X6	(6, 4)	3	0
Cost		10	2

Step 3: We select one of the non-medoids O′. Let us assume O′ = (6, 2). So now the medoids are c1(3, 4) and O′(6, 2). If c1 and O′ are the new medoids. We calculate the total cost involved.

Data Object		*Distance To*	
Sample	*Point*	*c1 = (3, 4)*	*c2 = (6, 2)*
X1	(2, 6)	3	8
X2	(3, 4)	0	5
X3	(3, 8)	4	9
X4	(4, 2)	3	2
X5	(6, 2)	5	0
X6	(6, 4)	3	2
Cost		7	4

So cost of swapping medoid from c2 to O′ is 11. Since the cost is less, this is considered as a better cluster assignment. Here swapping is done as the cost is less.

Step 4: We select another non-medoid O′. Let us assume O′ = (4, 2). So now the medoids are c1(3, 4) and O′(4, 2). If c1 and O′ are new medoids, we calculate the total cost involved.

Data Object		Distance To	
Sample	*Point*	*c1 = (3, 4)*	*c2 = (4, 2)*
X1	(2, 6)	3	6
X2	(3, 4)	0	3
X3	(3, 8)	4	7
X4	(4, 2)	3	0
X5	(6, 2)	5	2
X6	(6, 4)	3	4
Cost		7	8

cost of swapping medoid from c2 to O′ is 15. Since the cost is more, this cluster assignment is n
nsidered and the swapping is not done.

Reference

- “Machine Learning”-Anuradha Srinivasaraghavan, Vincy Joseph
- “Machine Learning”- Saikat Dutt, Subramanian Chandramouli, Amit Kumar Das