Assignment No. 4

Title: K-means Clustering.

Problem Definition: We have given a collection of 8 points. P1=[0.1,0.6], P2=[0.15,0.71], P3=[0.08,0.9], P4=[0.16,0.85], P5=[0.2,0.3], P6=[0.25,0.5], P7=[0.24,0.1], p8=[0.3,0.2].

Perform the k-means clustering with initial centroids as m1=p1 =cluster#1=c1 and m2=p8=cluster#2=c2.

Answer the following.

1] Which cluster does p6 belongs to?

2] What is the population of cluster around m2?

3] What is updated value of m1 and m2?

Prerequisite: Basics of Python, Mining Algorithm, Concept of K-means Clustering.

Software Requirements: Anaconda with Python 3.7, Jupyter Notebook

Hardware Requirement: PIV, 2GB RAM, 500 GB HDD, Lenovo A13-4089Model.

Objectives: To learn How to Apply K-means Clustering for given data points.

Outcomes: After completion of this assignment we are able to implement code for the k-means clustering with initial centroids.

Theory:

What is Clustering?

Clustering is dividing data points into homogeneous classes or clusters:

Points in the same group are as similar as possible.

Points in different group are as dissimilar as possible.

When a collection of objects is given, we put objects into group based on similarity.

Applications of Clustering:

Clustering is used in almost all the fields.

Listed here are few more applications:

- Clustering helps marketers improve their customer base and work on the target areas. It helps group people (according to different criteria's such as willingness, purchasing power etc.) based on their similarity in many ways related to the product under consideration.
- Clustering helps in identification of groups of houses on the basis of their value, type and geographical locations.
- Clustering is used to study earth-quake. Based on the areas hit by an earthquake in a region, clustering can help analyze the next probable location where earthquake can occur.

Clustering Algorithms:

A Clustering Algorithm tries to analyze natural groups of data on the basis of some similarity. It locates the centroid of the group of data points. To carry out effective clustering, the algorithm evaluates the distance between each point from the centroid of the cluster.

The goal of clustering is to determine the intrinsic grouping in a set of unlabeled data.

Fig. 1

k-means clustering algorithm:

- k-means is one of the simplest unsupervised learning algorithms that solve the well-known clustering problem. The procedure follows a simple and easy way to classify a given data set through a certain number of clusters (assume k clusters) fixed apriori.
- The main idea is to define k centers, one for each cluster. These centers should be placed in a cunning way because of different location causes different result.
- So, the better choice is to place them as much as possible far away from each other.
- The next step is to take each point belonging to a given data set and associate it to the nearest center.
- When no point is pending, the first step is completed and an early group age is done. At this point we need to re-calculate k new centroids as barycenter of the clusters resulting from the previous step.
- After we have these k new centroids, a new binding has to be done between the same data set points and the nearest new center.
- A loop has been generated. As a result of this loop we may notice that the k centers change their location step by step until no more changes are done or in other words centers do not move any more.
- Finally, this algorithm aims at minimizing an objective function know as squared error function given by:

$$J(V) = \sum_{i=1}^{c} \sum_{j=1}^{c_i} \left(\left\| x_i - v_j \right\| \right)^2$$

where,

'$||x_i - v_j||$' is the Euclidean distance between x_i and v_j.

'c_i' is the number of data points in i^{th} cluster.

'c' is the number of cluster centres.

Algorithmic steps for k-means clustering

Let $X = \{x_1, x, x_3, \ldots\ldots, x_n\}$ be the set of data points and $V = \{v_1, v_2, \ldots\ldots, v_c\}$ be the set of centers.

1) Randomly select *'c'* cluster centers.

2) Calculate the distance between each data point and cluster centers.

3) Assign the data point to the cluster centre whose distance from the cluster centre is minimum of all the cluster centers..

4) Recalculate the new cluster centre using:

$$v_i = (1/c_i) \sum_{j=1}^{c_i} x_i$$

where, *'c_i'* represents the number of data points in i^{th} cluster.

5) Recalculate the distance between each data point and new obtained cluster centers.

6) If no data point was reassigned then stop, otherwise repeat from step 3).

Advantages

1) Fast, robust and easier to understand.

2) Relatively efficient: O(tknd), where n is # objects, k is # clusters, d is # dimension of each object, and t is # iterations. Normally, k, t, d << n.

3) Gives best result when data set are distinct or well separated from each other.

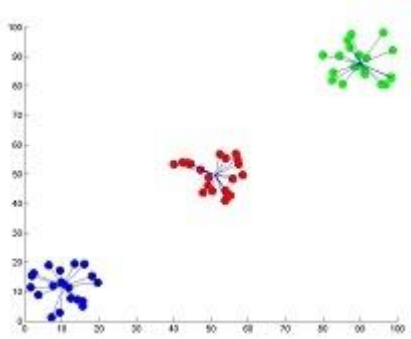


Fig I: Showing the result of k-means for *'N'* = 60 and *'c'* = 3

Note: For more detailed figure for k-means algorithm please refer to k-means figure subpage.

Disadvantages

1) The learning algorithm is not invariant to non-linear transformations i.e. with different representation of data we get different results (data represented in form of cartesian co-ordinates and polar co-ordinates will give different results).

2) Euclidean distance measures can unequally weight underlying factors.

3) The learning algorithm provides the local optima of the squared error function.

4) Randomly choosing of the cluster centre cannot lead us to the fruitful result. Pl. refer Fig.

5) Applicable only when mean is defined i.e. fails for categorical data.

6) Unable to handle noisy data and outliers.

7) Algorithm fails for non-linear data set.

Finding distances between points:

Let's calculate the distance between points P1 and P2:
P1 = [0.1,0.6] = [x1, y1]
P2 = [0.15,0.71] = [x2, y2]

Distance = sqrt $((x2-x1)^2 + (y2-y1)^2)$

= sqrt $((0.15-0.1)^2 + (0.71-0.6)^2)$

= sqrt (0.0146)

= 0.1208

Similarly, using this formula find the distance of every point with two centroids.

We'll get result as :

Points	Distance from centroid M1	Distance from centroid M2	Cluster centroid
P1	0	0.4472	M1
P2	0.1208	0.5316	M1
P3	0.3006	0.7338	M1
P4	0.2571	0.6649	M1
P5	0.3162	0.1414	M2
P6	0.1803	0.3041	M1
P7	0.5192	0.1166	M2
P8	0.4472	0	M2

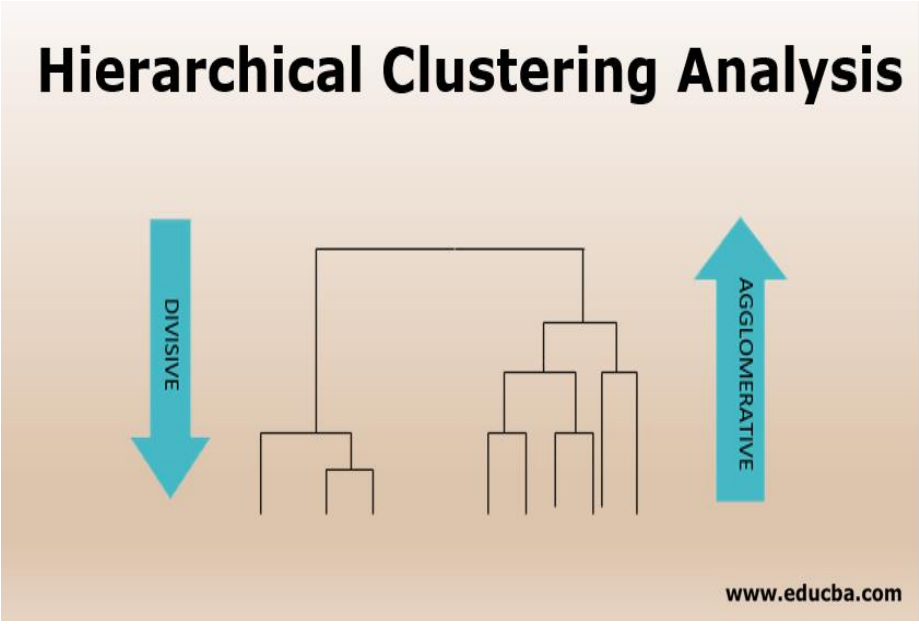
Hierarchical Clustering :

What is Hierarchical Clustering :

It is the hierarchical decomposition of the data based on group similarities. It is an algorithm that groups similar objects into groups called clusters. The endpoint is a set of clusters, where each cluster is distinct from each other cluster, and the objects within each cluster are broadly similar to each other.

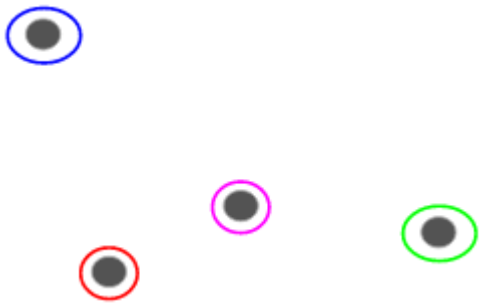
There are two top-level methods for finding these hierarchical clusters:

- **Agglomerative** clustering uses a bottom-up approach, wherein each data point starts in its own cluster. These clusters are then joined greedily, by taking the two most similar clusters together and merging them.
- **Divisive** clustering uses a top-down approach, wherein all data points start in the same cluster. You can then use a parametric clustering algorithm like K-Means to divide the cluster into two clusters. For each cluster, you further divide it down to two clusters until you hit the desired number of clusters. The main output of Hierarchical Clustering is a dendrogram, which shows the Hierarchical relationship between the clusters.

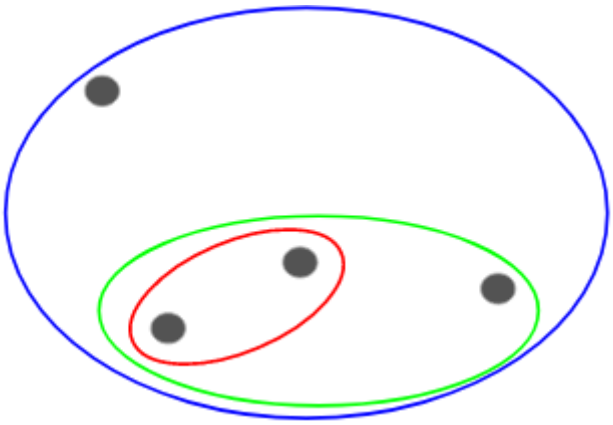


Agglomerative Hierarchical Clustering

We assign each point to an individual cluster in this technique. Suppose there are 4 data points. We will assign each of these points to a cluster and hence will have 4 clusters in the beginning:



Then, at each iteration, we merge the closest pair of clusters and repeat this step until only a single cluster is left:

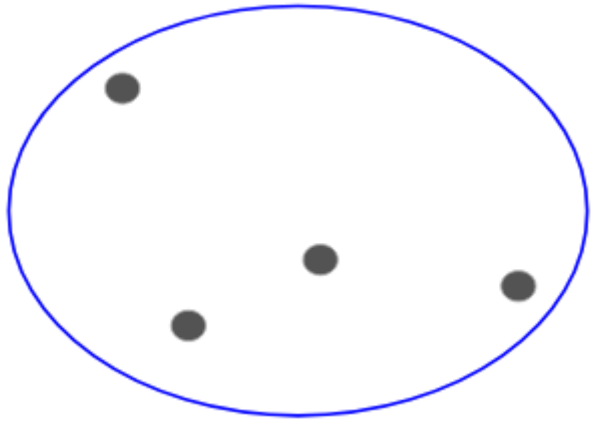


We are merging (or adding) the clusters at each step, right? Hence, this type of clustering is also known as **additive hierarchical clustering.**

Divisive Hierarchical Clustering

Divisive hierarchical clustering works in the opposite way. Instead of starting with n clusters (in case of n observations), we start with a single cluster and assign all the points to that cluster.

So, it doesn't matter if we have 10 or 1000 data points. All these points will belong to the same cluster at the beginning:



Now, at each iteration, we split the farthest point in the cluster and repeat this process until each cluster only contains a single point:

We are splitting (or dividing) the clusters at each step, hence the name divisive hierarchical clustering.

Agglomerative Clustering is widely used in the industry and that will be the focus in this article. Divisive hierarchical clustering will be a piece of cake once we have a handle on the agglomerative type.

How hierarchical clustering works

Hierarchical clustering starts by treating each observation as a separate cluster. Then, it repeatedly executes the following two steps:

(1) identify the two clusters that are closest together, and

(2) merge the two most similar clusters. This continues until all the clusters are merged together.

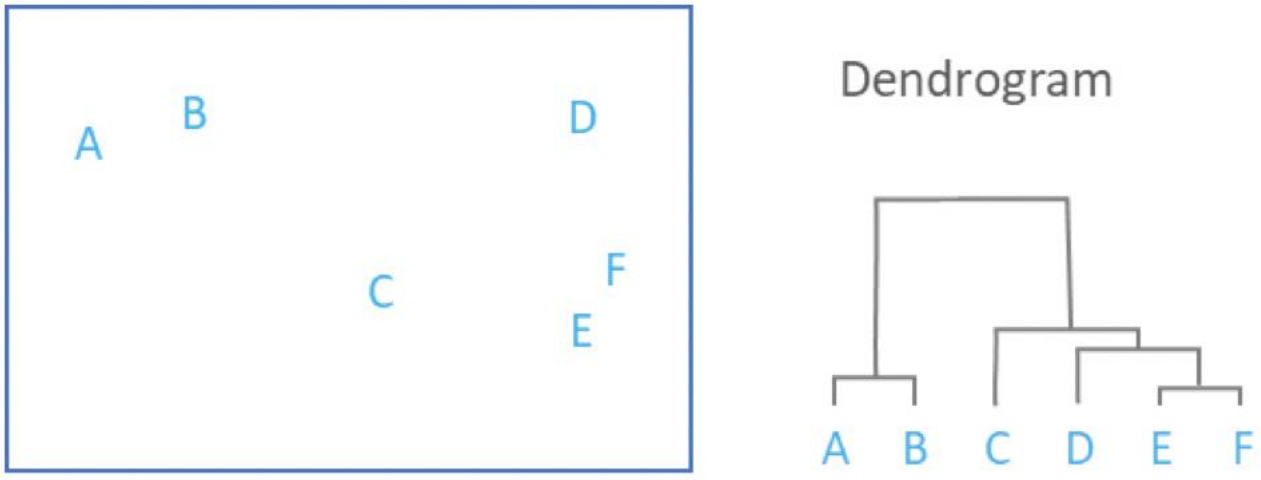
There are several ways to measure the distance between clusters in order to decide the rules for clustering, and they are often called Linkage Methods. Some of the common linkage methods are:

- **Complete-linkage**: the distance between two clusters is defined as the *longest* distance between two points in each cluster.
- **Single-linkage**: the distance between two clusters is defined as the *shortest* distance between two points in each cluster. This linkage may be used to detect high values in your dataset which may be outliers as they will be merged at the end.
- **Average-linkage**: the distance between two clusters is defined as the average distance between each point in one cluster to every point in the other cluster.
- **Centroid-linkage:** finds the centroid of cluster 1 and centroid of cluster 2, and then calculates the distance between the two before merging.

What is a Dendrogram?

A Dendrogram is a type of tree diagram showing hierarchical relationships between different sets of data.

Dendrogram example:



Conclusion: Hence in this assignment we have implemented clustering using two types i.e. k means and hierarchical clustering. In K means clustering assignment, we have formed two clusters and answered the questions asked. In hierarchical clustering we have used dendrography for visualizing.