Experiment No.: 10

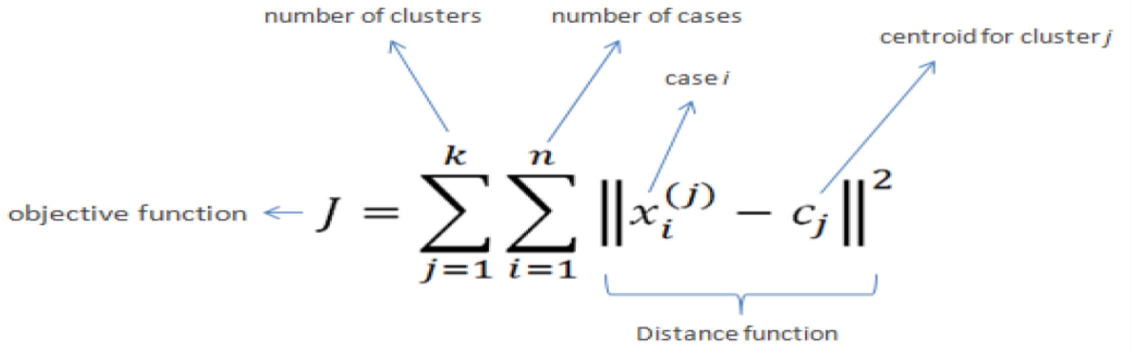
Title: Implementation of clustering using K-means.

Objectives: To learn K-means algorithm.

Theory:

K-means clustering is a method of vector quantization, originally from signal processing, that is popular for cluster analysis in data mining. K-means clustering aims to partition n observations into k clusters in which each observation belongs to the cluster with the nearest mean, serving as a prototype of the cluster.

- K-means clustering is a type of unsupervised learning, which is used when you have unlabeled data (i.e., data without defined categories or groups).
- The goal of this algorithm is to find groups in the data, with the number of groups represented by the variable K. The algorithm works iteratively to assign each data point to one of K groups based on the features that are provided.
- K-Means clustering intends to partition *n* objects into *k* clusters in which each object belongs to the cluster with the nearest mean.
- This method produces exactly *k* different clusters of greatest possible distinction.
- The best number of clusters *k* leading to the greatest separation (distance) is not known as a priori and must be computed from the data.
- The objective of K-Means clustering is to minimize the squared error function
- K-Means is relatively an efficient method.
- However, we need to specify the number of clusters, in advance and the final results are sensitive to initialization and often terminates at a local optimum.
- Unfortunately, there is no global theoretical method to find the optimal number of clusters.
- A practical approach is to compare the outcomes of multiple runs with different *k* and choose the best one based on a predefined criterion.
- In general, a large *k* probably decreases the error but increases the risk of overfitting.



$$J = \sum_{j=1}^{k} \sum_{i=1}^{n} \left\| x_i^{(j)} - c_j \right\|^2$$

Steps:

1. It starts with K as the input which is how many clusters you want to find. Place K centroids in random locations in your space.
2. Now, using the Euclidean distance between data points and centroids, assign each data point to the cluster which is close to it.

3. Recalculate the cluster centers as a mean of data points assigned to it.
4. Repeat 2 and 3 until no further changes occur.

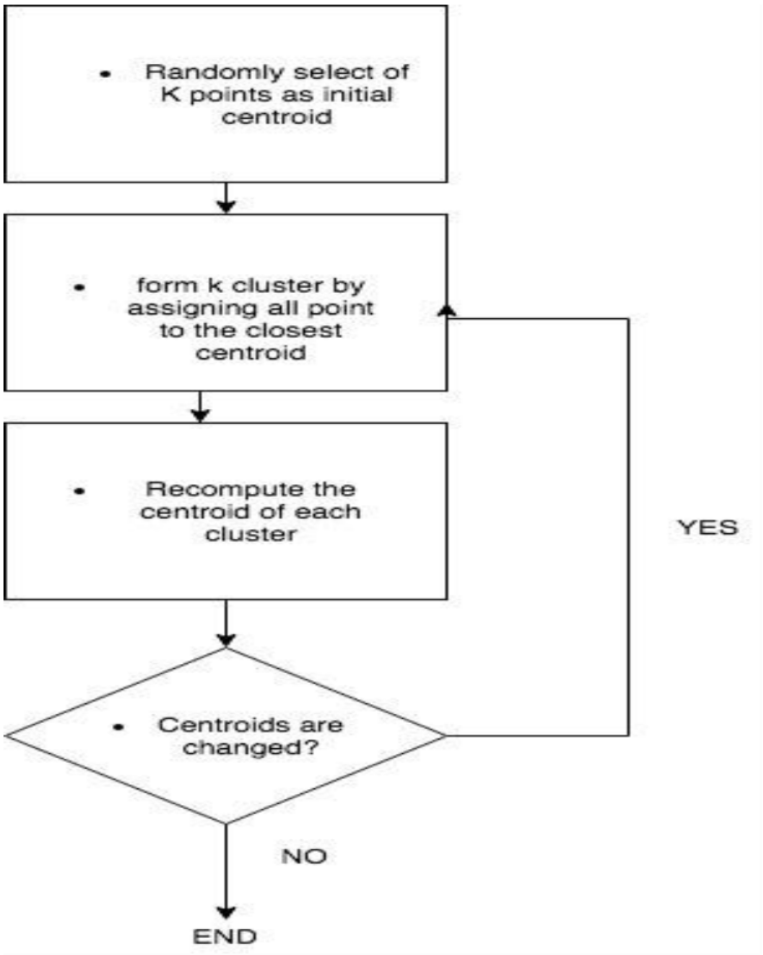

• Randomly select of K points as initial centroid
• form k cluster by assigning all point to the closest centroid
• Recompute the centroid of each cluster
• Centroids are changed?
YES
NO
END

Fig 1: K Means Clustering steps

Algorithm:

1. Suppose we want to group the visitors to a website using just their age (one-dimensional space) as follows *n* = 19 15,15,16,19,19,20,20,21,22,28,35,40,41,42,43,44,60,61,65
2. Consider initial cluster k=2.
3. Let C1=16 and C2=22
4. Calculate the Euclidean distance d1 and d2 using data points and centroids (C1 and C2).
5. Assign each data point to the respective cluster which is having minimum distance function value. For example, if first data point having d1<d2 then first data point is belonging to first cluster.
6. Calculate new centroid of each cluster by considering the mean of data points present in that cluster
7. Repeat step 4 to 6 till same centroids are generated.
8. Display the clusters