Final Year B. Tech. (CSE) – I : 2021-22
4CS462 : PE2 - Data Mining Lab
Assignment No. 7

Group id: DM21G12
Group members:
Abhishek More(2018BTECS00037)
Sushil Wagh(2018BTECS00031)

Title : Use / extend the data analysis tool (menu driven GUI) developed in Assignment No. 2 to perform the following task :

Objective/Aim :
1. Design and implement the following clustering algorithm:
    a) Hierarchical clustering - AGNES & DIANA. Plot Dendrogram.
    b) k-Means
    c) k-Medoids (PAM)
    d) DBSCAN
2. Tabulate the results with cluster validation accuracy
3. Use the following data sets from UCI machine learning repository :
    a) IRIS
    b) Breast Cancer
    c) For DBSCAN, use US Census Data (1990) Data Set

Introduction:
Theory of Hierarchical Clustering
There are two types of hierarchical clustering: Agglomerative and Divisive. In the former, data points are clustered using a bottom-up approach starting with individual data points, while in the latter top-down approach is followed where all the data points are treated as one big cluster and the clustering process involves dividing the one big cluster into several small clusters.

In this article we will focus on agglomerative clustering that involves the bottom-up approach.

Steps to Perform Hierarchical Clustering
Following are the steps involved in agglomerative clustering:

At the start, treat each data point as one cluster. Therefore, the number of clusters at the start will be K, while K is an integer representing the number of data points.
Form a cluster by joining the two closest data points resulting in K-1 clusters.
Form more clusters by joining the two closest clusters resulting in K-2 clusters.
Repeat the above three steps until one big cluster is formed.

Once single cluster is formed, dendrograms are used to divide into multiple clusters depending upon the problem. We will study the concept of dendrogram in detail in an upcoming section. There are different ways to find distance between the clusters. The distance itself can be Euclidean or Manhattan distance. Following are some of the options to measure distance between two clusters:

Measure the distance between the closes points of two clusters.
Measure the distance between the farthest points of two clusters.
Measure the distance between the centroids of two clusters.
Measure the distance between all possible combination of points between the two clusters and take the mean.

Hierarchical Agglomerative vs Divisive clustering –

Divisive clustering is more complex as compared to agglomerative clustering, as in the case of divisive clustering we need a flat clustering method as “subroutine” to split each cluster until we have each data having its own singleton cluster.
Divisive clustering is more efficient if we do not generate a complete hierarchy all the way down to individual data leaves. The time complexity of a naive agglomerative clustering is O(n3) because we exhaustively scan the N x N matrix dist_mat for the lowest distance in each of N-1 iterations. Using priority queue data structure we can reduce this complexity to O(n2logn). By using some more optimizations it can be brought down to O(n2). Whereas for divisive clustering given a fixed number of top levels, using an efficient flat algorithm like K-Means, divisive algorithms are linear in the number of patterns and clusters.
A divisive algorithm is also more accurate. Agglomerative clustering makes decisions by considering the local patterns or neighbor points without initially taking into account the global distribution of data. These early decisions cannot be undone. whereas divisive clustering takes into consideration the global distribution of data when making top-level partitioning decisions.

K-means clustering is one of the simplest and popular unsupervised machine learning algorithms.
Typically, unsupervised algorithms make inferences from datasets using only input vectors without referring to known, or labelled, outcomes.
AndreyBu, who has more than 5 years of machine learning experience and currently teaches people his skills, says that “the objective of K-means is simple: group similar data points together and discover underlying patterns. To achieve this objective, K-means looks for a fixed number (k) of clusters in a dataset.”
A cluster refers to a collection of data points aggregated together because of certain similarities.
You’ll define a target number k, which refers to the number of centroids you need in the dataset.
A centroid is the imaginary or real location representing the center of the cluster.
Every data point is allocated to each of the clusters through reducing the in-cluster sum of squares.
In other words, the K-means algorithm identifies k number of centroids, and then allocates every data point to the nearest cluster, while keeping the centroids as small as possible.

The ‘means’ in the K-means refers to averaging of the data; that is, finding the centroid.
How the K-means algorithm works
To process the learning data, the K-means algorithm in data mining starts with a first group of randomly selected centroids, which are used as the beginning points for every cluster, and then performs iterative (repetitive) calculations to optimize the positions of the centroids
It halts creating and optimizing clusters when either:
The centroids have stabilized — there is no change in their values because the clustering has been successful.
The defined number of iterations has been achieved.

K-Medoids Clustering:

A problem with the K-Means and K-Means++ clustering is that the final centroids are not interpretable or in other words, centroids are not the actual point but the mean of points present in that cluster. Here are the coordinates of 3-centroids that do not resemble real points from the dataset.

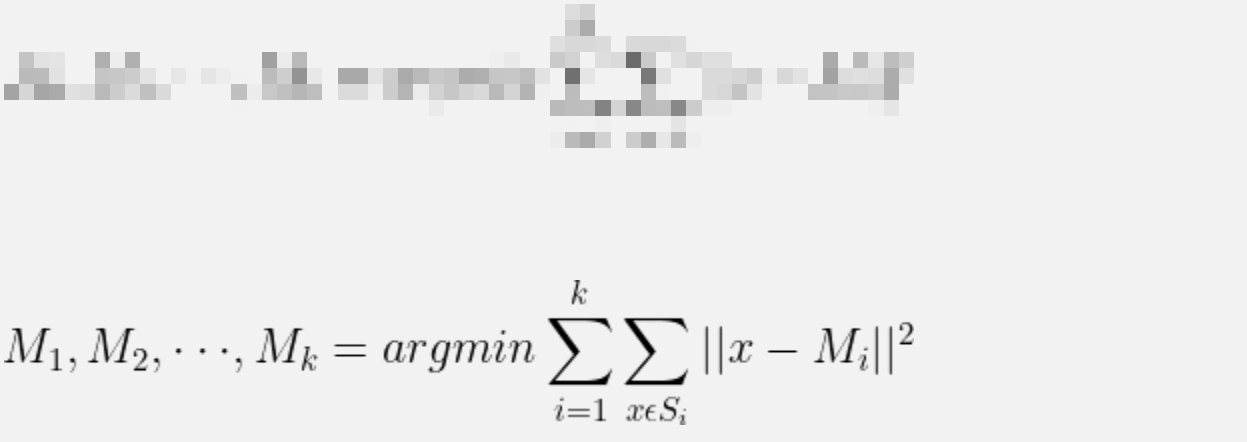
```
array([[ 5.88032652, -4.18480765],
       [-1.86315371, -8.87400825],
       [-0.86175784, -3.87190776]])
```

The idea of K-Medoids clustering is to make the final centroids as actual data-points. This result to make the centroids interpretable.

The algorithm of K-Medoids clustering is called Partitioning Around Medoids (PAM) which is almost the same as that of Lloyd's algorithm with a slight change in the update step.

Steps to follow for PAM algorithm:

- **Initialization:** Same as that of K-Means++
- **Assignment:** Same as that of K-Means
- **Update centroids:** In the case of K-Means we were computing mean of all points present in the cluster. But for the PAM algorithm updation of the centroid is different. If there are m-point in a cluster, swap the previous centroid with all other (m-1) points from the cluster and finalize the point as new centroid that have a minimum loss. Minimum loss is computed by below cost function:



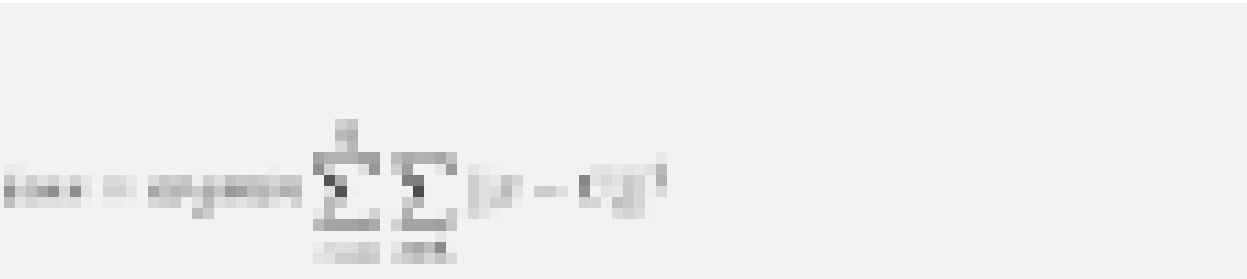
$$M_1, M_2, \cdots, M_k = argmin \sum_{i=1}^{k} \sum_{x \epsilon S_i} ||x - M_i||^2$$

- **Repeat:** Same as that of K-Means

How to pick the best value of K?

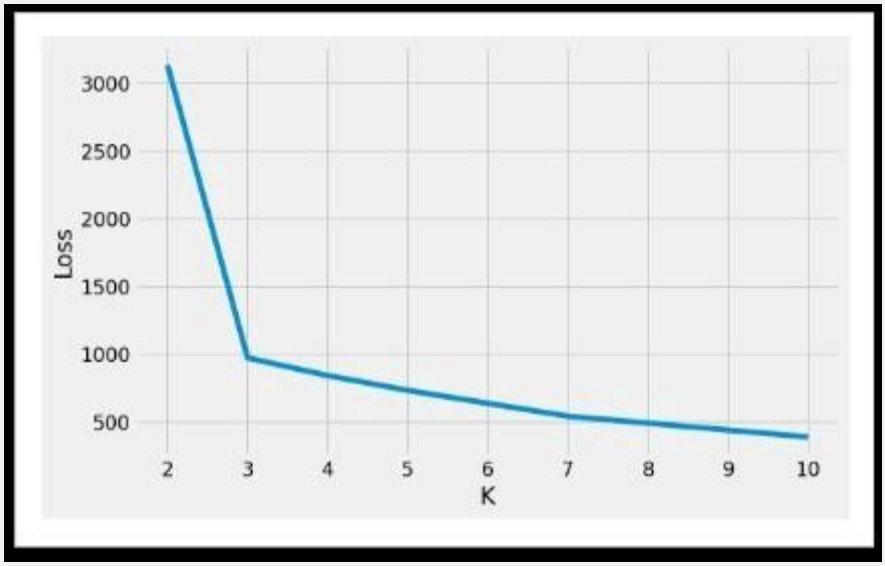
The best value of K can be computed using the *Elbow method*.

The cost function of K-Means, K-Means, and K-Medoids techniques is to minimize intercluster distance and maximize intracluster distance. This can be achieved by minimizing the loss function discussed above in the article:



$$loss = argmin \sum_{i=1}^{k} \sum_{x \epsilon S_i} ||x - C_i||^2$$

To determine the right ‘K’, draw a plot between loss vs k.



Plot for Loss vs K, (Image 10)

For the above plot, it is observed that with the increase in the value of ‘k’, loss decreases. To find the best value of ‘k’ as to draw k-clusters, we can pick k=3.

References:

- https://www.tensorflow.org/tutorials
- https://towardsdatascience.com/the-complete-tensorflow-tutorial-for-newbies-dc3acc1310f8
- https://towardsdatascience.com/beginners-guide-to-deep-learning-with-tensorflow-ca85969b2f2