**TITLE:** Implement a simple approach for k-means/ k-medoids clustering using C++.

**THEORY:**

**What is clustering?**

Clustering is the classification of objects into different groups, or more precisely, the partitioning of a data set into subsets (clusters), so that the data in each subset (ideally) share some common trait - often according to some defined distance measure.

**Types of clustering:**

1. **Hierarchical algorithms**: these find successive clusters using previously established clusters.

1. Agglomerative ("bottom-up"): Agglomerative algorithms begin with each element as a separate cluster and merge them into successively larger clusters.

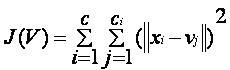
2. Divisive ("top-down"): Divisive algorithms begin with the whole set and proceed to divide it into successively smaller clusters.

**2. Partitional clustering:** Partitional algorithms determine all clusters at once. They include:

* + ***K*-means and derivatives**
  + Fuzzy *c*-means clustering

QT 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:

[](https://sites.google.com/site/dataclusteringalgorithms/k-means-clustering-algorithm/kmeans.JPG?attredirects=0)

where,  
                           *‘||xi - vj||’* is the Euclidean distance between *xi* and *vj.*

*‘ci’* is the number of data points in *ith* cluster.

*‘c’* is the number of cluster centers.

**Algorithmic steps for k-means clustering**

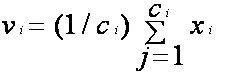
Let  X = {x1,x2,x3,……..,xn} be the set of data points and V = {v1,v2,…….,vc} be the set of centers.

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 center whose distance from the cluster center is minimum of all the cluster centers..

4) Recalculate the new cluster center using:



where, *‘ci’* represents the number of data points in *ith* 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).

**Time Complexity:** O (n)

**Applications of K-Mean Clustering**

* It is relatively *efficient and fast.* It computes result at **O(tkn),** where n is number of objects or points, k is number of clusters and t is number of iterations.
* k-means clustering can be applied to *machine learning or data mining*
* *Used on acoustic data in speech understanding to convert waveforms into one of k categories (known as Vector Quantization or Image Segmentation).*
* *Also used for choosing color palettes on old fashioned graphical display devices and Image Quantization.*



**ALGORITHM:**

1. Initialize the center of the clusters
2. Attribute the closest cluster to each data point
3. Set the position of each cluster to the mean of all data points belonging to that cluster
4. Repeat steps 2-3 until convergence

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*OUTPUT\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Enter 10 numbers:

1 2 3 4 5 50 51 52 53 54

Enter initial mean 1:

Enter initial mean 2:

Cluster 1 1 2 3 4 5

m1= 3

Cluster 2: 50 51 52 53 54

m2= 52

----

Clusters created

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

**TESTING**

**Positive Testing:**

* Elements enter must be integer, and 10 elements can be added, with 2 initial means in integer, output will be 2 clusters and their means
* If elements value exceeds the limit of, int size, then its value considered by program will be erroneous.

**Negative Testing:**

* If more than 10 elements are entered, the next two elements after 10th will be treated as means, and other extra elements won’t be considered by program.
* If character data type is entered, program will consider its ASCII value.
* If float or double is entered, program will crash.

**CONCLUSION**

Thus we have studied and implemented K-means Algorithm.