Improvised K-Means Algorithm

→ Introduction

- Clustering is a way that classifies the raw data reasonably and searches the hidden patterns that exists in dataset.

→ Shortcomings of K-means algorithm.

- By seeing normal K-means algorithm, we find that algorithm has to calculate the distance of each data object from every cluster centre in each iteration.
- Owing to which the time taken for its execution has increased manifold.

- Complexity of time has increased because certain data pts which remained in same cluster from start to end had to be checked for ~~each~~ its distance from every cluster centre after each iteration.

- Hence method although being quite effective was too tedious and time consuming.

- However by experiments we have found that it is not necessary for us to calculate distance each time.

→ Improvised K-Means Clustering Algorithm

- Standard K-means algorithm need to calculate the distance of each data of each cluster centroids every time the iteration is conducted, which takes up a lot of execution time especially for large capacity database
- Main idea of improvised algorithm is to use 2 data structures
  → 1 to store cluster label.
  → 2 to store distance of all the data objects to the nearest cluster during each iteration.
- After an iteration, we calculate distance of given data object from the cluster centroid of same cluster.
- If the new distance calculated is less than old distance then data object will be part of same cluster.
- Or else distance from all cluster centroid needs to be determined in order to calculate the cluster to which

given data object must be is part.

→ Algorithm

- Input:
The number of desired clusters k, and a database $D = \{d_1, d_2, \ldots, d_n\}$ containing n data objects.

- Output:
A set of k clusters.

1) Randomly select k objects from dataset D as initial cluster centres

2) Calculate the distance between each data object d_i $(1 <= i <= n)$ and all k cluster centres c_j as euclidean distance $d(d_i, c_j)$ and assign data objects d_i to nearest cluster

3) For each data object d_i, find the closest centre c_j and assign d_i to cluster centre j

4) Store the label of cluster in which data object d_i is and the distance of data object d to the nearest clusters

5) Recalculate cluster centroid for each cluster.

6) Repeat.

7) For each data object di

Compute it's distance to the centre of the present nearest cluster

a) If this distance is less than or equal to Dist[i], the data object stays in its initial cluster.

b) else.
For every cluster center, compute the distance d(di, cj) of each data object to all the centres, assign the data object di to the nearest centre. cj

8) Recalculate cluster centroids.

9) Iterate till variation of cluster centroid is minimum.

→ Advantages of Optimised clustering algorithm:-

∴ Time complexity of execution of code decreases manifold.

· It almost reduces to 't' times to normal k-means code execution.
t = total number of k-means iteration.