New Approach for K-mean and K-medoids Algorithm

Abhishek Patel  
Department of Information & Technology,  
Parul Institute of Engineering & Technology,  
Vadodara, Gujarat, India  

Purnima Singh  
Department of Computer Science & Engineering,  
Parul Institute of Engineering & Technology,  
Vadodara, Gujarat, India

Abstract: K-means and K-medoids clustering algorithms are widely used for many practical applications. Original k-mean and k-medoids algorithms select initial centroids and medoids randomly that affect the quality of the resulting clusters and sometimes it generates unstable and empty clusters which are meaningless. The original k-means and k-medoids algorithm is computationally expensive and requires time proportional to the product of the number of data items, number of clusters and the number of iterations. The new approach for the k mean algorithm eliminates the deficiency of exiting k mean. It first calculates the initial centroids k as per requirements of users and then gives better, effective and stable cluster. It also takes less execution time because it eliminates unnecessary distance computation by using previous iteration. The new approach for k-medoids selects initial k medoids systematically based on initial centroids. It generates stable clusters to improve accuracy.

Keywords: K-means; K-medoids; centroids; clusters;

1. INTRODUCTION

We Technology advances have made data collection easier and faster, resulting in large, more complex, datasets with many objects and dimensions. Important information is hidden in this data. Data Mining has become an intriguing and interesting topic for the information extraction from such data collection since the past decade. Furthermore there are so many subtopics related to it that research has become a fascination for data miners. Cluster analysis or primitive exploration with little or no prior knowledge, consists of research developed across a wide variety of communities. The goal of a clustering algorithm is to group similar data points in the same cluster while purring dissimilar data points in different clusters. It groups the data in such a way that inter-cluster similarity is maximized and intra-cluster similarity is minimized. Clustering is an unsupervised learning technique of machine learning whose purpose is to give the ability to machine to find some hidden structure within data.

K-means and K-medoids are widely used simplest partition based 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 a priori.

In this direction, we have tried to put our efforts in enhancing the partition base clustering algorithm to improve accuracy and generate better and stable cluster and also reduce its time complexity and to efficiently scale a large data set. For this I have used a concept of “initial centroids” which has proved to be a better option.

2. ANALYSIS OF EXISTING SYSTEM

K-means Clustering

A 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 a priori. The main idea is to define k centroids, one for each cluster. These centroids 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 centroid. When no point is pending, the first step is completed and an early group age is done. At this point, this method needs to re-calculate k new centroids as barycenters of the clusters resulting from the previous step. After these k new centroids, a new binding has to be done between the same data set points and the nearest new centroid. A loop has been generated. As a result of this loop we may notice that the k centroids change their location step by step until no more changes are done. In other words centroids do not move any more. Finally, this algorithm aims at minimizing an objective function, in this case a squared error function. The objective function

\[ J = \sum_{i=1}^{k} \sum_{j=1}^{n} ||x_i^{(j)} - c_j||^2 \]

where \( ||x_i^{(j)} - c_j|| \) is a chosen distance measure between a data point \( x_i^{(j)} \) and the cluster centre \( c_j \), is an indicator of the distance of the \( n \) data points from their respective cluster centres.

The algorithm is composed of the following steps:

1. Place K points into the space represented by the objects that are being clustered. These points represent initial group centroids.
2. Assign each object to the group that has the closest centroid.
3. When all objects have been assigned, recalculate the positions of the K centroids.
4. Repeat Steps 2 and 3 until the centroids no longer move. This produces a separation of the objects into groups from which the metric to be minimized can be calculated.

Time Complexity & Space Complexity:

Let \( n \) = number of objects  
\( K \) = number of clusters  
\( t \) = number of iteration  
The time complexity of k mean algorithm is \( O(nkt) \) and space complexity is \( O(n+k) \).
Consider a data set of ten objects as follows:

Cluster the following data set of ten objects into two clusters. Demonstration of PAM follows:

Partitioning Around Medoids (PAM) algorithm and is as follows:

K-medoids Algorithm

The k-medoids algorithm is a clustering algorithm related to the k-means algorithm and the medoidshift algorithm. Both the k-means and k-medoids algorithms are partitioned (breaking the dataset up into groups) and both attempt to minimize squared error, the distance between points labeled to be in a cluster and a point designated as the center of that cluster. In contrast to the k-means algorithm k-medoids chooses data points as centers.

K-medoid is a classical partitioning technique of clustering that clusters the data set of n objects into k clusters known a priori. A useful tool for determining k is the silhouette. It is that clusters the data set of n objects into k clusters known as.

A medoid can be defined as the object of a cluster, whose average dissimilarity to all the objects in the cluster is minimal i.e. it is a most centrally located point in the given data set.

The most common realization of k-medoid clustering is the Partitioning Around Medoids (PAM) algorithm and is as follows:

1. Initialize: randomly select k of the n data points as the medoids
2. Associate each data point to the closest medoid. ("closest" here is defined using any valid distance metric, most commonly Euclidean distance, Manhattan distance or Minkowski distance)
3. For each mediod m,
   For each non-mediod data point o Swap m and o and compute the total cost of the configuration.
4. Select the configuration with the lowest cost.
5. Repeat steps 2 to 5 until there is no change in the medoid.

Demonstration of PAM

Cluster the following data set of ten objects into two clusters i.e k = 2.

Consider a data set of ten objects as follows:

Table 1 – Data point for distribution of the data

<table>
<thead>
<tr>
<th>X_1</th>
<th>X_2</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>6</td>
</tr>
<tr>
<td>3</td>
<td>4</td>
</tr>
<tr>
<td>3</td>
<td>8</td>
</tr>
<tr>
<td>4</td>
<td>7</td>
</tr>
<tr>
<td>6</td>
<td>2</td>
</tr>
<tr>
<td>6</td>
<td>4</td>
</tr>
<tr>
<td>7</td>
<td>3</td>
</tr>
<tr>
<td>7</td>
<td>4</td>
</tr>
<tr>
<td>8</td>
<td>5</td>
</tr>
<tr>
<td>7</td>
<td>6</td>
</tr>
</tbody>
</table>

Then so the clusters become:

Cluster_1 = {(3,4),(2,6),(3,8),(4,7)}
Cluster_2 = {(7,4),(6,2),(6,4),(7,3),(8,5),(7,6)}

Since the points (2,6) (3,8) and (4,7) are close to c_1 hence they form one cluster whilst remaining points form another cluster. So the total cost involved is 20.

Where cost between any two points is found using Minkowski distance metric with r = 1.

Table 2 – Cost Calculation for distribution of the data

<table>
<thead>
<tr>
<th>c_1</th>
<th>Data objects (X_i)</th>
<th>Cost (distance)</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td>4, 2, 6</td>
<td>3</td>
</tr>
<tr>
<td>3</td>
<td>4, 3, 8</td>
<td>4</td>
</tr>
<tr>
<td>3</td>
<td>4, 4, 7</td>
<td>4</td>
</tr>
<tr>
<td>3</td>
<td>4, 6, 2</td>
<td>5</td>
</tr>
<tr>
<td>3</td>
<td>4, 6, 4</td>
<td>3</td>
</tr>
<tr>
<td>3</td>
<td>4, 7, 3</td>
<td>5</td>
</tr>
<tr>
<td>3</td>
<td>4, 8, 5</td>
<td>6</td>
</tr>
<tr>
<td>3</td>
<td>4, 7, 6</td>
<td>6</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>c_2</th>
<th>Data objects (X_i)</th>
<th>Cost (distance)</th>
</tr>
</thead>
<tbody>
<tr>
<td>7</td>
<td>4, 2, 6</td>
<td>7</td>
</tr>
<tr>
<td>7</td>
<td>4, 3, 8</td>
<td>8</td>
</tr>
<tr>
<td>7</td>
<td>4, 4, 7</td>
<td>6</td>
</tr>
<tr>
<td>7</td>
<td>4, 6, 2</td>
<td>3</td>
</tr>
<tr>
<td>7</td>
<td>4, 6, 4</td>
<td>1</td>
</tr>
<tr>
<td>7</td>
<td>4, 7, 3</td>
<td>1</td>
</tr>
<tr>
<td>7</td>
<td>4, 8, 5</td>
<td>2</td>
</tr>
<tr>
<td>7</td>
<td>4, 7, 6</td>
<td>2</td>
</tr>
</tbody>
</table>

Observation:

k-means algorithm is a popular clustering algorithm applied widely, but the standard algorithm which selects k objects randomly from population as initial centroids can not always give a good and stable clustering. Selecting centroids by our algorithm can lead to a better clustering.

www.ijcat.com
Step 2
Selection of nonmedoid O’ randomly.
Let us assume O’ = (7,3). So now the medoids are c1(3,4) and O’(7,3). If c1 and O’ are new medoids, calculate the total cost involved.
By using the formula in the step 1

\[
\begin{array}{cccc}
\text{Distance} & \text{Data objects (X)} & \text{Cost (distance)} \\
3 & 4 & 6 & 3 \\
3 & 4 & 3 & 8 & 4 \\
3 & 4 & 4 & 7 & 4 \\
3 & 4 & 6 & 2 & 5 \\
3 & 4 & 6 & 4 & 3 \\
3 & 4 & 7 & 4 & 4 \\
3 & 4 & 8 & 5 & 6 \\
3 & 4 & 7 & 6 & 6 \\
\end{array}
\]

\[
\begin{array}{cccc}
\text{Distance} & \text{Data objects (X)} & \text{Cost (distance)} \\
7 & 3 & 2 & 6 & 8 \\
7 & 3 & 3 & 8 & 9 \\
7 & 3 & 4 & 7 & 7 \\
7 & 3 & 6 & 2 & 2 \\
7 & 3 & 6 & 4 & 2 \\
7 & 3 & 7 & 4 & 1 \\
7 & 3 & 8 & 5 & 3 \\
7 & 3 & 7 & 6 & 3 \\
\end{array}
\]

Table 3: Cost Calculation for distribution of the data

Total cost = 3+4+4+2+2+1+3+3 = 20

So cost of swapping medoid from c2 to O’ is

\[
S = \text{current total cost} - \text{past total cost} = 22 - 20 = 2 > 0.
\]

So moving to O’ would be bad idea, so the previous choice was good and algorithm terminates here (i.e. there is no change in the medoids).
It may happen some data points may shift from one cluster to another cluster depending upon their closeness to medoid. For large values of n and k, such computations become very costly.

3. PROPOSED PARTITION METHOD ALGORITHM FOR NEW K-MEAN AND K-MEODIDS CLUSTERS

The proposed algorithm of classical Partition method is based on initial centroid, how to assign new data point to appropriate cluster and calculate new centroid. In Proposed algorithm of k-medoid how it initializes the cluster seeds means how it choose initial medoids. The proposed algorithm is an accurate and efficient because with this approach significant computation is reduce by eliminating all of the distance comparisons among points that do not fall within a common cluster.
Also, the size of all cluster are similar, it calculate the threshold value, which control the size of each cluster. Here we have to provide only number of clusters and data set as an input parameter.

3.1 Process in proposed algorithm.
The key idea behind the use of the calculation of initial centroid is greatly improve the accuracy and algorithm become good and more stable. Based on the initial cluster centroid and medoid is partition the data points into number of clusters. To calculate the initial centroid and medoid it requires only the input data points and number of clusters k.
In second stage it also stored the previous clusters centroid and Euclidean distance between two data points. When new centroid is calculate then a data point is assign to new cluster or previous cluster is decided by comparing present Euclidean distances between two points and previous Euclidean distance. It is greatly reduce the number of distance calculation required for clustering.

3.2 Proposed Algorithm for New k-means
Instead of initial centroids are selected randomly, for the stable cluster the initial centroids are determined systematically. It calculates the Euclidean distance between each data point and selects two data-points between which the distance is the shortest and form a data-point set which contains these two data-points, then we delete them from the population. Now find out nearest data point of this set and put it into new set. The numbers of elements in the set are decided by initial population and number of clusters systematically. These ways find the different sets of data points. Numbers of sets are depending on the value of k. Now calculate the mean value of each sets that become initial centroid of the proposed k mean algorithm.
After finding the initial centroids, it starts by forming the initial clusters based on the relative distance of each data-point from the initial centroids. These clusters are
subsequently fine-tuned by using a heuristic approach, thereby improving the efficiency.

**Input:**

\[ D = \{d_1, d_2, \ldots, d_n\} \] // set of n data items
\[ k \] // Number of desired clusters

**Output:**

A set of k clusters

**Steps:**

1. Compute the distance between each data point and all other data-points in the set D
2. Find the closest pair of data points from the set D and form a data-point set Am (1 ≤ p ≤ k) which contains these two data points, Delete these two data points from the set D
3. Find the data point in D that is closest to the data point set Ap, Add it to Ap and delete it from D
4. Repeat step 4 until the number of data points in Am reaches 0.75*(n/k)
5. If p<k, then p = p+1, find another pair of data points from D between which the distance is the shortest, form another data-point set Ap and delete them from D, Go to step 4
6. For each data-point set Am (1≤p≤k) find the arithmetic mean of the vectors of data points \( C_p(1≤p≤k) \) in Ap, these means will be the initial centroids
7. Compute the distance of each data point (1≤i≤n) to all the centroids \( C_j \) (1≤j≤k) as d(di, cj)
8. For each data-point di, find the closest centroid cj and assign di to cluster j
9. Set Clusterld[i] = j; // j; d of the closest cluster
10. Set Nearest_Dist[j] = d(di, cj)
11. For each cluster j (1≤j≤k), recalculate the centroids
12. Repeat
13. For each data-point di
   14.1 Compute its distance from the centroid of the present nearest cluster
   14.2 If this distance is less than or equal to the present nearest distance, the data-point stays in the cluster
   Else
   14.2.1 For every centroid cj (1≤j≤k) Compute the distance (di, cj);
   Endfor

For each cluster j (1≤j≤k), recalculate the centroids; Until the convergence Criteria is met.

**Complexity Of Algorithm**

Enhanced algorithm requires a time complexity of \( O(n^2) \) for finding the initial centroids, as the maximum time required here is for computing the distances between each data point and all other data-points in the set D.

In the original k-means algorithm, before the algorithm converges the centroids are calculated many times and the data points are assigned to their nearest centroids. Since complete redistribution of the data points takes place according to the new centroids, this takes \( O(n^k*l) \), where \( n \) is the number of data-points, \( k \) is the number of clusters and \( l \) is the number of iterations.

To obtain the initial clusters, it requires \( O(nk) \). Here, some data points remain in their cluster while the others move to other clusters depending on their relative distance from the new centroid and the old centroid. This requires \( O(1) \) if a data-point stays in its cluster and \( O(k) \) otherwise. As the algorithm converges, the number of data points moving away from their cluster decreases with each iteration. Assuming that half the data point’s move from their clusters, this requires \( O(nk/2) \). Hence the total cost of this phase of the algorithm is \( O(nk) \), not \( O(nk*l) \). Thus the overall time complexity of the improved algorithm becomes \( O(n^2) \), since \( k \) is much less than \( n \).

**3.3 Proposed Algorithm for new K-medoids**

Unfortunately, K-means clustering is sensitive to the outliers and a set of objects closest to a centroid may be empty, in which case centroids cannot be updated. For this reason, K-medoids clustering are sometimes used, where representative objects called medoids are considered instead of centroids. Because it uses the most centrally located object in a cluster, it is less sensitive to outliers compared with the K-means clustering. Among many algorithms for K-medoids clustering, Partitioning Around Medoids (PAM) proposed by Kaufman and Rousseeuw (1990) is known to be most powerful. However, PAM also has a drawback that it works inefficiently for large data sets due to its complexity. We are interested in developing a new K-medoids clustering method that should be fast and efficient.

**Algorithm for Select initial medoids**

**Input:**

\[ D = \{d_1, d_2, \ldots, d_n\} \] // set of n data items
\[ K = \text{number of desired cluster} \]

**Output:**

A set of k initial centroids \( C_m = \{C_1, C_2, \ldots, C_k\} \)

**Steps:**

1. Set p = 1
2. Compute the distance between each data point and all other data-points in the set D
3. Find the closest pair of data points from the set D and form a data-point set Ap (1 <= p <= k) which contains these two data-points. Delete these two data points from the set D.
4. Find the data point in D that is closest to the data point set Ap. Add it to Ap and delete it from D.
5. Repeat step 4 until the number of data points in Ap reaches 0.75*(n/k).
6. If p<k, then p = p+1, find another pair of data points from D between which the distance is the shortest, form another data-point set Ap and delete them from D. Go to step 4.
7. For each data-point set Ap (1<=p<=k) find the arithmetic mean \( C_p \) of the vectors of data points in Ap, these means will be the initial centroids.

**Algorithm for k-medoids**

**Input:**
(1) Database of O objects
(2) A set of k initial centroids \( C_m = \{C_1, C_2, \ldots, C_k\} \)

**Output:**
A set of k clusters

**Steps:**
1. Initialize initial medoid which is very close to centroids \( \{C_1, C_2, \ldots, C_k\} \) of the n data points.
2. Associate each data point to the closest medoid. (“closest” here is defined using any valid distance metric, most commonly Euclidean distance, Manhattan distance or Minkowski distance).
3. For each medoid m
   For each non-medoid data point o
   Swap m and o and compute the total cost of the configuration.
4. Select the configuration with the lowest cost.
5. Repeat steps 2 to 5 until there is no change in the medoid.

**Complexity of Algorithm**

Enhanced algorithm requires a time complexity of \( O(n^2) \) for finding the initial centroids, as the maximum time required here is for computing the distances between each data point and all other data-points in the set D. Complexity of remaining part of the algorithm is \( O(k(n-k)^2) \) because it is just like PAM algorithm. So over all complexity of algorithm is \( O(n^2) \), since k is much less than n.

4. CONCLUSION

The k-means algorithm is widely used for clustering large sets of data. But the standard algorithm do not always guarantee good results as the accuracy of the final clusters depend on the selection of initial centroids. Moreover, the computational complexity of the standard algorithm is objectionably high owing to the need to reassign the data points a number of times, during every iteration of the loop.

An enhanced k-means algorithm which combines a systematic method for finding initial centroids and an efficient way for assigning data points to clusters. This method ensures the entire process of clustering in \( O(n^2) \) time without sacrificing the accuracy of clusters. The previous improvements of the k-means algorithm compromise on either accuracy or efficiency.

The Proposed k medoid algorithm runs just like K-means clustering algorithm. The proposed algorithm is used systematic method of choosing the initial medoids. The performance of the algorithm may vary according to the method of selecting the initial medoids. It is more efficient than existing k medoid. Time complexity of clustering is \( O(n^2) \) time without sacrificing the accuracy of clusters.

5. FUTURE WORK

In new Approach of classical partition based clustering algorithm, the value of k (the number of desired clusters) is given as input parameter, regardless of distribution of the data points. It would be better to develop some statistical methods to compute the value of k, depending on the data distribution.

6. REFERENCES

[7] Zhexue Huang, “A Fast Clustering Algorithm to Cluster Very Large Categorical Data Sets in Data Mining”.