IOSR Journal of Mathematics (IOSR-JM)
e-ISSN: 2278-5728, p-ISSN:2319-765X. Volume 10, Issue 2 Ver. III (Mar-Apr. 2014), PP 49-53
www.iosrjournals.org

Modified Centroid Selection Method of K-Means Clustering

Rose Mawati[1),]I Made Sumertajaya[2)], Farit Mochamad Afendi[2)]

[1)]Student of Statistics Major, Bogor Agriculture University, Bogor, Indonesia
[2)]Lecturer in Department of Statistics, Bogor Agriculture University, Bogor, Indonesia

***Abstract**: Clustering is a process of classifying object into groups which have similarity. The result of clustering will show that objects in one cluster will be more homogeneous than others. There are two methods in classic clustering analysis i.e. hierarchical cluster method and non-hierarchical cluster method. Determination of the member of clusters which formed by them is done subjectively. K-means is one of the algorithms that solve the well known clustering problem. The algorithm classifies object to a predefined number of clusters, which is given by the user. The idea is to choose random cluster centers, one for each other. The centroid initialization plays an important role in determining the cluster assignment in effective ways. This paper presents results of the simulated data of different datasets using original k-means and other modified algorithms implemented using MATLAB R2010a. This results are calculated on some performance measures such as no. iterations, and accuracy.*

***Keywords:** Clustering; K-Means; Centroid Selection*

I. Introduction

Cluster analysis is one of the major data analysis methods which is widely used for many practical applications. Clustering is the process of partitioning a given set of objects into disjoint clusters [3]. The k-means algorithm is effective in producing clusters for many practical applications, but the computational complexity of the original k-means algorithm is very high, especially for large data sets. This result in different types of clusters depending on the random choice of initial centroids [1].

The data in this paper using simulated data. Simulated data was generated data multivariate distribution which useful to measure the performance of k-means method and its modified approach in classifying an object. simulated data was generated data numeric type which consisted of three clusters were clearly separated, and a population that consisted of three clusters each other in small, middle, and large numbers.

The results of methods showed that simulation data has the good ability to classify data.

This paper present the partition based clustering method. A partitioning method first creates an initial set of k partitions, where parameter k is the desired number of cluster as output. It is original and very fast, so in many practical applications this method is proved to be effective way that can produce good clustering results [2].

II. Methodology

K-Means Clustering Algorithm

k-means is one of the algorithms that solve the well known clustering problem. In 1967 MacQueen first proposed k-means clustering algorithm [5]. K-means algorithm is one of the popular partitioning algorithm. The idea is to classify the data into k clusters where k is the input parameter specified in advance through iterative relocation technique which converges to local minimum. K-means for describing an algorithm of his that assigns each item to the cluster having the nearest centroid. The algorithm is expressed as follows.

```
Algorithm 1: k-means Clustering Algorithm
Input:
D = {d1, d2,…,dn} // set of n data items.
K // number of desired clusters
Output:
A set of k clusters
Steps:
Arbitrarily choose k data items from D as initial centroids;
Repeat assign each item di to the cluster which has the closest centroid;
Calculate new mean for each cluster;
Until corvergence criteria is met.
```

Modified Approach K-Means

This paper proposes a systematic approach to determine the initial centroids so as to produce clusters with better accuracy. In the enhanced method in this paper, both the phases of the original k-means algorithm are modiified to improve the accuracy and efficiency. The enhanced method is outlined as Algorithm 2.

Algorithm 2: The enhanced method

```
Input:
        D = {d1, d2,......,dn}      // set of    n data items
k       // Number of desired clusters
Output:
        A set of    k clusters.
Steps:
Phase 1: determine the initial centroids of the clusters by using Algorithm 3.
Phase 2: assign each data point to the appropriate clusters by using Algorithm 4.
Phase 1: Determine the initial centroids of the clusters by
using Algorithm 3.
Phase 2:    Assign each data point to the appropriate clusters by
using Algorithm 4.
```

Algorithm 3: Finding the initial centroids

```
Input:
D = {d1, d2,......,dn}      // set of    n data items
k       // Number of desired clusters
Output:
A set of    k initial centroids .
Steps:
set m = 1;
            compute the distance between each data point and all other data points in the set D;
find the closest pair of data points from the set D and form a data point set Am (1<=m<=k) which contains these two data points, delete these two data points from the set D;
find the data points in D that is closest to the datapoint set Am. Add it to Am and delete it from D;
repeat step 4 until the number of data points in Am reaches 0.75*(n/k);
if m<k, then m = m+1, find another pair of datapoints from D between which the distance is the shortest, from another data points set Am and delete them from D, Go to step 4;
for each data points set Am (1<=m<=k) find the arithmetic mean of the vectors of data points in Am, these means will be the initial centroids.
```

Algorithm 4: Assigning data points to clusters

```
Input:
D = {d1, d2,…,dn} // set of n data points
C = {c1, c2,…,ck} // set of k centroids
Output:
A set of k clusters
Steps:
compute the distance of each data point di (1<=i<=n) to all the centroids cj (1<=j<=k) as d(di, cj);
For each data point di, find the closest centroids cj and assign di to cluster j.
Set ClusterId[i]=j; // j"Id of the closest cluster
Set Nearest_Dist[i]=d(di, cj);
For each clusters j (1<=j<=k), recalculate the centroids;
Repeat
For each data point di,
Compute its distance from the centroid of the present nearest cluster;
If this distance is less than or equal to the present nearest distance, the data point stays in the cluster;
Else
For every centroid cj (1<=j<=k) compute the distance d(di, cj);
Endfor;
Assign the data point di to the cluster with the nearest centroid cj
Set ClusterId[i]=j;
Set Nearest_Dist[i]=d(di, cj);
Endfor;
For each cluster j (1<=j<=k),    recalculate the centroid;
Until the convergence criteria is met.
```

The first step in Phase 2 is to determine the distance between each data-point and the initial centroids of all the clusters. The data-points are then assigned to the clusters having the closest centroids. This results in an initial grouping of the data-points. For each data-point, the cluster to which it is assigned (ClusterId) and its distance from the centroid of the nearest cluster (Nearest_Dist) are noted [4]. Inclusion of data-points in various clusters may lead to a change in the values of the cluster centroids. For each cluster, the centroids are recalculated by taking the mean of the values of its data-points. Up to this step, the procedure is almost similar to the original k-means algorithm except that the initial centroids are computed systematically.

The next stage is an iterative process which makes use of a heuristic method to improve the efficiency. During the iteration, the data-points may get redistributed to different clusters. The method involves keeping track of the distance between each data-point and the centroid of its present nearest cluster. At the beginning of the iteration, the distance of each data-point from the new centroid of its present nearest cluster is determined. If this distance is less than or equal to the previous nearest distance, that is an indication that the data point stays in that cluster itself and there is no need to compute its distance from other centroids. This results in the saving of time required to compute the distances to k-1 cluster centroids. On the other hand, if the new centroid of the present nearest cluster is more distant from the data-point than its previous centroid, there is a chance for the data-point getting included in another nearer cluster. In that case, it is required to determine the distance of the data-point from all the cluster centroids. Identify the new nearest cluster and record the new value of the nearest distance. The loop is repeated until no more data-points cross cluster boundaries, which indicates the convergence criterion. The heuristic method described above results in significant reduction in the number of computations and thus improves the efficiency.

III. Result Analysis

Implemented these 2 algorithms using MATLAB R2010a software and evaluated the result on simulation data with three different condition

Table 1. Combination simulation data

Distance	Size of data (n)	Variance	Correlation	Cases of simulation
Near			0	1
		Small	0,3	2
$\mu_1=\begin{pmatrix}1\\2\\5\end{pmatrix}$		$\sigma_k^2=1$	0,5	3
			0,8	4
			0	5
$\mu_2=\begin{pmatrix}2\\5\\1\end{pmatrix}$	900	Middle	0,3	6
		$\sigma_k^2=3$	0,5	7
			0,8	8
$\mu_3=\begin{pmatrix}5\\1\\2\end{pmatrix}$			0	9
		Big	0,3	10
		$\sigma_k^2=9$	0,5	11
			0,8	12
Middle			0	13
		Small	0,3	14
$\mu_1=\begin{pmatrix}1\\3\\7\end{pmatrix}$		$\sigma_k^2=1$	0,5	15
			0,8	16
			0	17
$\mu_2=\begin{pmatrix}3\\7\\1\end{pmatrix}$	900	Middle	0,3	18
		$\sigma_k^2=3$	0,5	19
			0,8	20
$\mu_3=\begin{pmatrix}7\\1\\3\end{pmatrix}$			0	21
		Big	0,3	22
		$\sigma_k^2=9$	0,5	23
			0,8	24
Far			0	25
		Small	0,3	26
$\mu_1=\begin{pmatrix}1\\4\\9\end{pmatrix}$		$\sigma_k^2=1$	0,5	27
			0,8	28
			0	29
$\mu_2=\begin{pmatrix}4\\9\\1\end{pmatrix}$	900	Middle	0,3	30
		$\sigma_k^2=3$	0,5	31
			0,8	32
$\mu_3=\begin{pmatrix}9\\1\\4\end{pmatrix}$			0	33
		Big	0,3	34
		$\sigma_k^2=9$	0,5	35
			0,8	36

Table 2. Comparison of the number of iterations on the distance between the center of clusters near

Cases of simulation	Iterastions	
	K-Means	Modified K-Means
1	5	1
2	22	1
3	2	1
4	3	1
5	11	10
6	9	2
7	5	4
8	2	1
9	16	12
10	25	11
11	16	10
12	7	1

Table 3. Comparison of the number of iterations on the distance between the center of clusters middle

Cases of simulation	Iterations	
	K-Means	Modified K-Means
13	7	1
14	18	1
15	4	1
16	3	1
17	18	3
18	5	2
19	3	1
20	2	1
21	13	10
22	17	10
23	7	3
24	5	1

Table 4. Comparison of the number of iterations on the distance between the center of clusters far

Cases of simulation	Iterations	
	K-Means	Modified K-Means
25	4	1
26	3	1
27	3	1
28	3	1
29	6	2
30	3	1
31	10	1
32	2	1
33	17	10
34	9	3
35	4	2
36	2	1

The number of iteration required for various techniques are compared. Table 1 represent the comparison of number of iterations requered for vaious techniques with different dataset. From the table, it can be observed that the proposed clustering results in lesser number of iteration when compared to k-means and modified k-means techniques.

Table 5. Comparison of clustering accuracy on the distance between the center of clusters near

Cases of simulation	Accuracy (%)	
	K-Mean s	Modified K-Means
1	100	100
2	100	100
3	100	100
4	100	100
5	79.78	80.78
6	93.56	93.56
7	98.44	98.44
8	100	100

9	55	56.67
10	69	66
11	79.78	79.89
12	99.89	99.89

Table 6. Comparison of clustering accuracy on the distance between the center of clusters middle

Cases of simulation	Accuracy (%)	
	K-Means	Modified K-Means
13	100	100
14	100	100
15	100	100
16	100	100
17	94	94
18	98.67	98.67
19	100	100
20	100	100
21	69.89	68.56
22	82.78	82.78
23	93.78	94
24	100	100

Table 7. Comparison of clustering accuracy on the distance between the center of clusters far

Cases of simulation	Accuracy (%)	
	K-Means	Modified K-Means
25	100	100
26	100	100
27	100	100
28	100	100
29	98	98
30	100	100
31	100	100
32	80	100
33	80	79.33
34	92.78	92.78
35	98	98
36	100	100

The accuracy for various techniques are compared. Table 1 represent the comparison of number of iterations requered for vaious techniques with different dataset. From the table, it can be observed that modified k-means technique has as same accuracy as k-means .

IV. 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. This paper presents 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 time without sacrificing the accuracy of clusters. Modified k-means more efficient than k-means, and has as same accuracy as k-means.

References

[1] K. A. Abdul Nazeer and M. P. Sebastian, "Improving the Accuracy and Efficiency of the k-means Clustering Algoritm," Proceedings of the World Congress on Engineering 2009, Vol. I, WCE 2009.

[2] M. P. S. Bhatia and D. Khurana, "Experimental study of Data clustering using k-Means and modified algorithms," International Journal of Data Mining and Knowledge Management Process (IJDKP). Vol. 3. No. 3. 2013.

[3] M. R. Anderberg. "Cluster Analysis for Application," Academic Press. New York, 1973.

[4] S. Sujatha and A. S. Sona, "New Fast K-Means Clustering Algorithm using Modified Centroid Selection Method," International Journal of Engineering Reseach and Technology (IJERT), Vol. 2, Issue. 2, 2013. ISSN: 2278-0181.

[5] R. A. Johnson and D. W. Wichern, "Applied Multivariate Statistical Analysis," 6th Edition, Pearson Education.,New Jersey, 2007.