Experimental study of Data clustering using k-Means and modified algorithms

Dr. M.P.S Bhatia[1] and Deepika Khurana[2]

[1]Department of Computer Engineering, Netaji Subhash Institute of Technology,
University of Delhi, New Delhi, India
bhatia.mps@gmail.com
[2]Department of Computer Engineering, Netaji Subhash Institute of Technology,
University of Delhi, New Delhi, India
deepika.khurana13@gmail.com

ABSTRACT

The k- Means clustering algorithm is an old algorithm that has been intensely researched owing to its ease and simplicity of implementation. Clustering algorithm has a broad attraction and usefulness in exploratory data analysis. This paper presents results of the experimental study of different approaches to k- Means clustering, thereby comparing results on different datasets using Original k-Means and other modified algorithms implemented using MATLAB R2009b. The results are calculated on some performance measures such as no. of iterations, no. of points misclassified, accuracy, Silhouette validity index and execution time.

KEYWORDS

Data Mining, Clustering Algorithm, k- Means, Silhouette Validity Index.

1. INTRODUCTION

Data Mining is defined as mining of knowledge from huge amount of data. Using Data mining we can predict the nature and behaviour of any kind of data. The past two decades has seen a dramatic increase in the amount of information being stored in the electronic format. This accumulation of data has taken place at an explosive rate. It was recognized that information is at the heart of the business operations and that decision makers could make the use of data stored to gain the valuable insight into the business. DBMS gave access to the data stored but this was only small part of what could be gained from the data. Analyzing data can further provide the knowledge about the business by going beyond the data explicitly stored to derive knowledge about the business.

Learning valuable information from the data made clustering techniques widely applied to the areas of artificial intelligence, customer – relationship management, data compression, data mining, image processing, machine learning, pattern recognition, market analysis, and fraud – detection and so on. Cluster Analysis of a data is an important task in Knowledge Discovery and Data Mining. Clustering is the process to group the data on the basis of similarities and dissimilarities among the data elements. Clustering is the process of finding the group of objects such that object in one group will be similar to one another and different from the objects in the other group. A good clustering method will produce high quality clusters with high intra cluster distance similarity and low inter cluster distance similarity. Similarity measure used is standard Euclidean distance but there can also be other distance measures such as Manhattan distance, Minkowski distance and many others. The quality of clustering depends on both the similarity measure used by the method and also by its ability to discover some or all of the hidden patterns.

DOI : 10.5121/ijdkp.2013.3302

The popular clustering approach can be partition based or hierarchy based, but both approaches have their own merits and demerits in terms of number of clusters, cluster size, separation between clusters , shape of clusters, etc… Some other approaches are also based on hybridization of different clustering techniques. Many Clustering algorithms use the center based cluster criterion. The center of a cluster is often a centroid, the average of all the points in a cluster.

This paper presents the partition based clustering method called as k- Means and its modified approaches with the experimental results. A partitioning method first creates an initial set of k partitions, where parameter k is the desired number of clusters as output. It then uses an iterative relocation technique that attempts to improve the partitioning of the data points. k-Means is a numerical, unsupervised iterative method. It is Original and very fast, so in many practical applications this method is proved to be very effective way that can produce good clustering results. But the computational complexity of original k- Means is very high, especially for large datasets. Moreover this algorithm results in different type of clusters depending upon the random choice of initial clusters.

We have implemented 3 algorithms and performance is measured on the basis of No. of iterations, No. of points misclassified, Accuracy, Silhouette validity index, and Time of execution.

The Silhouette index for each data element is simply the measure of how similar that data element is to other elements in its own cluster as compared to the elements in other clusters. It ranges from -1 to +1. For each data point value close to +1 indicates that it belongs to the cluster being assigned. A value close to zero indicates data point could also be assigned to another closest cluster. A value close to -1 indicates that data point is in wrong cluster or somewhere in between clusters.

In literature [1] there is an improved k-Means algorithm based on the improvement of the sensitivity of the initial centers. This algorithm partitions the whole data space into different segments and calculates the frequency of points in each segment. The segment which shows the maximum frequency will b considered for initial centroid depending upon the value of k.

In literature [2] another method of finding initial cluster centers is discussed. It first finds closest pair of data points and then on the basis of these points it forms the subset of dataset, and this process is repeated k times to find k small subsets, to find initial k centroids

The rest of the paper is organized as follows. The 2 section presents the related work. Section 3 details the k- Means algorithm. Section 4 and 5 details modified algorithms and their pseudo-code. Section 6 describes the experimental results and paper is concluded in last section.

The implementation work is done using MATLAB R2009b programming software. The number of clusters is input to the program. The closeness of data points in each cluster is displayed by Silhouette plot and the execution time is calculated in seconds.

2. RELATED WORK

The author in literature [3] uses Principal Component Analysis for dimension reduction and to find initial cluster centers.

In [4] first data set is pre-processed by transforming all data values to positive space and then data is sorted and divided into k equal sets and then middle value of each set is taken as initial center point.

In literature[5] a dynamic solution to k –Means is proposed that algorithm is designed with pre-processor using silhouette validity index that automatically determines the appropriate number of clusters, that increase the efficiency for clustering to a great extent.

In [6] a method is proposed to make algorithm independent of number of iterations that avoids computing distance of each data point to cluster centers repeatedly, saving running time and reducing computational complexity.

In the literature [7] dynamic means algorithm is proposed to improve the cluster quality and optimizing the number of clusters. The user has the flexibility either to fix the number of clusters or input the minimum number of clusters required. In the former case it works same as k-Means algorithm. In the latter case the algorithm computes the new cluster centers by incrementing the cluster count by one in every iteration until it satisfies the validity of cluster quality

In [8] the main purpose is to optimize the initial centroids for k-Means algorithm.Author proposed Hierarchical k-Means algorithm. It utilizes all the clustering results of k-Means in certain times, even though some of them reach the local optima. Then, transform the all centroids of clustering result by combining with Hierarchical algorithm in order to determine the initial centroids for k-Means. This algorithm is better used for the complex clustering cases with large data set and many dimensional attributes.

3. k- Means CLUSTERING

This section describes Original k-Means clustering algorithm. In 1967 MacQueen first proposed k-Means clustering algorithm. 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.

It consists of two separate phases : First phase is to determine k centers at random one for each cluster. Next phase is to determine distance between data points in Dataset and the cluster centers and assigning the data point to its nearest cluster.Euclidean distance is generally considered to determine the distance.When all the data points are included in some clusters an initial grouping is done. New centers are then calculated by taking the average of points in the clusters. This is done because of inclusion of new points may lead to change in cluster centers. This process of center updation is iterated until a sitaution where centers do not update anymore or criterion function becomes minimum.This signifies the convergence criteria. Typically square error criterion is used which is defined by equation 1:

$$E = \sum_{i=1}^{k} \sum_{p \in Ci} |p - m_i| \qquad (1)$$

Where p is the data point and m_i is the center for cluster C_i. E is th sum of squared error of all points in dataset. The distance of criterion function is the Euclidean distance which is used to calculate the distance between data point and cluster center. The Euclidean distance between two vectors x = $(x_1, x_2, x_3, x_4 \text{--------} x_n)$ and y= $(y_1, y_2, y_3, y_4 \text{----} y_n)$ can be calculated using equation 2:

$$d(x_i, y_i) = \sqrt{\sum_{i=1}^{n} (x_i - y_i)^2} \qquad (2)$$

Pseudo code for k-Means algorithm is as follows:

Input: Dataset of n data points d_i (i = 1 to N)
Desired number of clusters = k

Output: N data points clustered into k clusters.

Steps:

1. Randomly select k data objects from dataset D as the initial cluster centers.
2. Repeat
3. Calculate the distance between each data point d_i (i =1to N) and all k cluster centers C_j (j= 1 to k) and assign the data object d_i to the nearest cluster j.
4. For each cluster j, recalculate the cluster center.
5. Until no changing of cluster centers.

The Original k – Means algorithm also has some limitations:

1. Initial selection of number of clusters must be previously known. Because algorithm does not tell about correctness of the number of clusters, it may be possible to choose wrong value of k.
   For example your data might naturally have five clusters but you fed k= 3, then algorithm will return back 3 clusters. Those clusters will not be compact and well separated. If number of clusters chosen is small then there may be chance of putting dissimilar points in one cluster.
2. It can contain dead unit problem, that is if cluster center is wrongly chosen, it may never be updated and thus never represent a class.
3. Converges to local minima, i.e. for chosen value of k different initial centers are chosen randomly by the algorithm on different runs.

4. MODIFIED APPROACH I

The first approach discussed in [1] optimizes the Original k –Means algorithm by proposing a method on how to choose initial clusters. The author proposed a method that partitions the given input data space into k * k segments, where k is desired number of clusters. After portioning the data space, frequency of each segment is calculated and highest k frequency segments are chosen to represent initial clusters. If some parts are having same frequency, the adjacent segments with the same least frequency are merged until we get the k number of segments. Then initial centers are calculated by taking the mean of the data points in respective segments to get the initial k centers. By this process we will get the initial which are always same as compared to the Original k – Means algorithm which always select initial random centers for a given dataset.

Next, a threshold distance is calculated for each centroid is defined as distance between each cluster centroid and for each centroid take the half of the minimum distance from the remaining centroids . Threshold distance is denoted by $d_{c(i)}$ for the cluster C_i . It is given by equation 3 and 4 as:

$$d(C_i, C_j) = \{d(m_i, m_j): i, j \in [1, k] \;\&\; i \neq j\} \qquad (3)$$

where $d(C_{i,} C_j)$ is distance between centroid i and j.

$$\boldsymbol{d}_{c(i)=\frac{1}{2}}\ \boldsymbol{min}\{|\boldsymbol{C_i}, \boldsymbol{C_j}|, \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\} \qquad (4)$$

where $d_{c(i)}$ is half of the minimum distance of i th cluster from other remaining clusters.

To assign the data point to the cluster, take a point p in the dataset and calculate its distance from the centroid of cluster i and compare it with $\mathbf{d}_{\mathbf{c(i)}}$. If it is less than or equal to $\mathbf{d}_{\mathbf{c(i)}}$ then assign the data point p to the cluster i else calculate its distance from other centroids. This process is repeated until data point p is assigned to one of the cluster. If data point p is not assigned to any of the cluster then the centroid which shows minimum distance for the data point p becomes the centroid for that point. The centroid is then updated by calculating the mean of the data points in the cluster using equation 5:

$$M_j = 1/n_j \sum_{z_p \in C_i} z_p \tag{5}$$

where M_j is centroid of cluster j and n_j is the number of data points in cluster j.
This process is repeated until no more cluster centroids are updated.

Pseudo code for modified k-Means algorithm is as follows:

Input: Dataset of N data points D (i = 1 to N)
Desired number of clusters = k

Output: N data points clustered into k clusters.

Steps:

```
1.  Input the data set and value of k.
2.  If the value of k is 1 then Exit.
3.  Else
4.  /*divide the data point space into k*k, means k Vertically and k horizontally*/
5.  For each dimension{
6.      Calculate the minimum and maximum value of data points.
7.      Calculate range of group($R_g$) using equation ((min+max)/k)
8.      Divide the data point space in k group with width $R_g$
9.  }
10. Calculate the frequency of data points in each partitioned space.
11. Choose the k highest frequency group.
12. Calculate the mean of selected group. /* These will be the initial centroid of cluster.*/
13. Calculate the distance between each clusters using equation (3)
14. Take the minimum distance for each cluster and make it half using equation (4)
15. For each data points p= 1 to N {
16.     For each cluster j= 1 to k {
17.         Calculate d($Z_p$,$M_j$) using equation (2)
18.         If (d($Z_p$,$M_j$)) $\leq$ dc$_j$){
19.             Then $Z_p$ assign to cluster $C_j$ .
20.             Break
21.         }
22.         Else
23.             Continue;
24.     }
25.     If $Z_p$, does not belong to any cluster then
26.         $Z_p$, $\in$ min(d($Z_p$, , $M_i$)) where i$\in$ [1, k]
27. }
28. Check the termination condition of algorithm if Satisfied
29.     Exit.
30. Else
31.     Calculate the centroid of cluster using equation (5).
32.     Go to step 13.
```

5. MODIFIED APPROACH II

This paper proposes a systematic approach to determine the initial centroids so as to produce clusters with better accuracy. The paper also presents a different way of assigning clusters to center and improve overall efficiency and computational complexity.

First phase is to determine initial centroids, for this compute the distance between each data point and all other data points in the set D. Then find out the closest pair of data points and form a set A1 consisting of these two data points, and delete them from the data point set D. Then determine the data point which is closest to the set A1, add it to A1 and delete it from D. Repeat this procedure until the number of elements in the set A1 reaches a threshold. Then again form another data-point set A2. Repeat this till 'k' such sets of data points are obtained. Finally the initial centroids are obtained by averaging all the vectors in each data-point set. The Euclidean distance is used for determining the closeness of each data point to the cluster centroids

Next phase is to assign points to the clusters. Here the main idea is to set two simple data structures to retain the labels of cluster and the distance of all the data objects to the nearest cluster during the each iteration, that can be used in next iteration, we calculate the distance between the current data object and the new cluster center, if the computed distance is smaller than or equal to the distance to the old center, the data object stays in it's cluster that was assigned to in previous iteration. Therefore, there is no need to calculate the distance from this data object to the other k- 1clustering center, saving the calculative time to the k-1 cluster centers. Otherwise, we must calculate the distance from the current data object to all k cluster centers, and find the nearest cluster center and assign this point to the nearest cluster center. And then we separately record the label of nearest cluster center and the distance to its center. Because in each iteration some data points still remain in the original cluster, it means that some parts of the data points will not be calculated, saving a total time of calculating the distance, thereby enhancing the efficiency of the algorithm.

Pseudo code for modified k -Means algorithm

Input: Dataset D of N data points (i = 1 to N)
Desired number of clusters = k

Output: N data points clustered into k clusters.

Phase 1:

Steps:

1. Set m = 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 A_m (1<= m <= 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 A_m, Add it to A_m and delete it from D;
5. Repeat step 4 until the number of data points in A_m reaches 0.75*(N/k);
6. If m<k, then m = m+1, find another pair of data points from D between which the distance is the shortest, form another data-point set Am and delete them from D, Go to step 4;
7. For each data-point set Am (1<=m<=k) find the arithmetic mean of the vectors of data points in A_m, these means will be the initial centroids

Phase 2:

Steps:

```
1. Compute the distance of each data-point $d_i$ (1<=i<=N) to all the centroids $C_j$ (1<=j<=k) as $d(d_i, C_j)$;
2. For each data-point di, find the closest centroid $C_j$ and assign di to cluster j.
3. Set Cluster Id[i]=j; // j:Id of the closest cluster
4. Set Nearest _Dist[i]= $d(d_i, C_j)$;
5. For each cluster j (1<=j<=k), recalculate the centroids;
6. Repeat
7. For each data-point $d_i$,
      a. Compute its distance from the centroid of thepresent nearest cluster;
      b. If this distance is less than or equal to the present nearest distance, the data-point stays in the cluster;
      c. Else for every centroid cj (1<=j<=k) compute the distance $d(d_i, C_j)$;
      d. End for;

8. Assign the data-point di to the cluster with the nearest centroid $C_j$
9. Set ClusterId[i]=j;
10. Set Nearest_Dist[i] = $d(d_i, C_j)$;
11. End for (step(2));
12. For each cluster j (1<=j<=k), Recalculate the centroids until the convergence criteria is met.
```

6. RESULT ANALYSIS

We have implemented these 3 algorithms using MATLAB R2009b software and evaluated the results on 2 real datasets Fisher - iris and Wine from UCI machine repository.

Fisher-iris has 4 attributes, 150 instances, and 3 classes so we chose value of k=3.
Wine dataset has 14 attributes, 178 instances, and 3 classes and thus k=3.

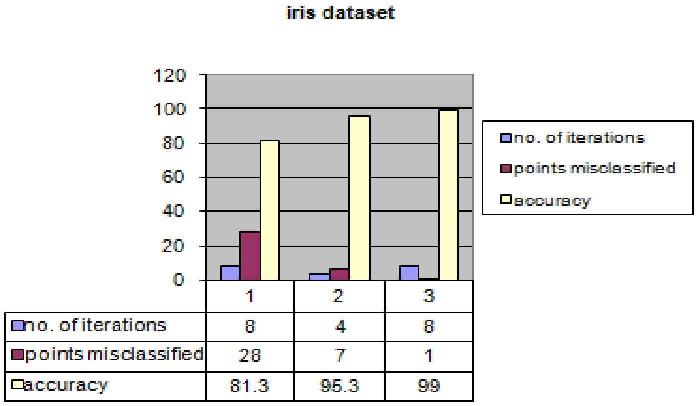
Before applying these 3 algorithms, datasets are pre-processed using correlation analysis to find the most correlated attributes and duplicates are removed from dataset and then clustering is performed on processed data.

We also did little additions to these algorithms for finding the performance parameters, all performance measuring parameters are also coded after applying algorithm. Experiments are also conducted on random datasets of 50,178 and 500 random 2-d data points.

Performance is evaluated on the basis of No. of iterations, No. of points misclassified, Accuracy, Silhouette validity index, and Time of execution.

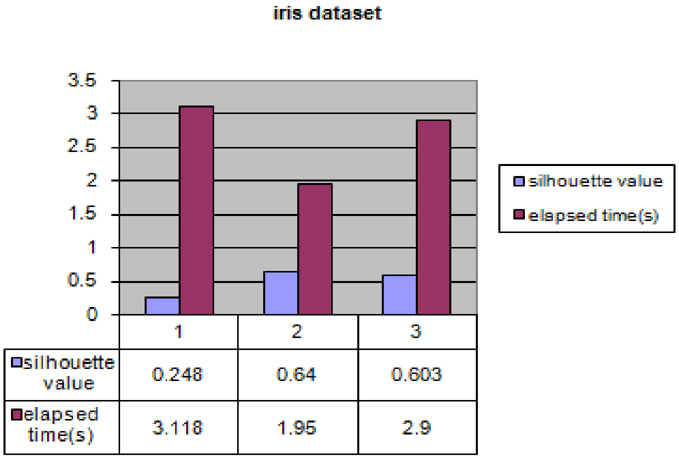
Results also show that Original k- Means algorithm shows different performance measures for same dataset and fixed value of k in different run because of its nature of randomly selecting initial centers, wherein both modified approaches this limitation is removed and same results are obtained in different runs.

Results shows comparison between all 3 algorithms: Original k-Means, Modified approach 1 and Modified approach 2. Figure 1 and 2 shows results for Fisher-iris dataset for all 3 algorithms 1, 2 and 3.Figure 3 and 4 shows results on Wine dataset. Figure 5 to 9 shows results for random dataset having points N= 500. Experiments are conducted for different values of k no. of clusters = 3,4,5,6 and 7 and all performance measures are calculated that evaluates all 3 algorithms.



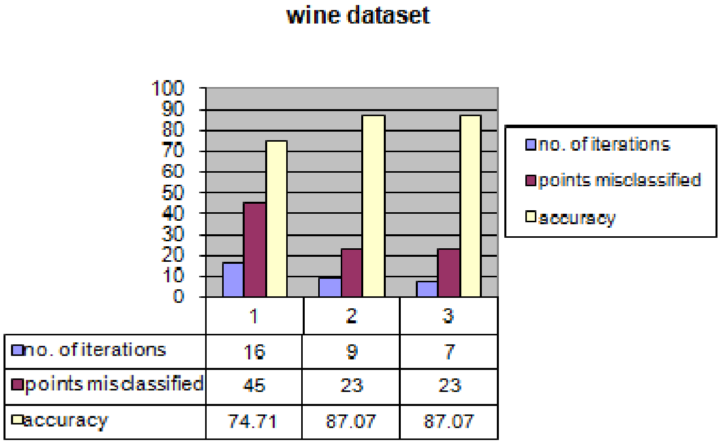
	1	2	3
no. of iterations	8	4	8
points misclassified	28	7	1
accuracy	81.3	95.3	99

Figure1. Comparison of no. of iterations, no. of points misclassified, accuracy for k =3



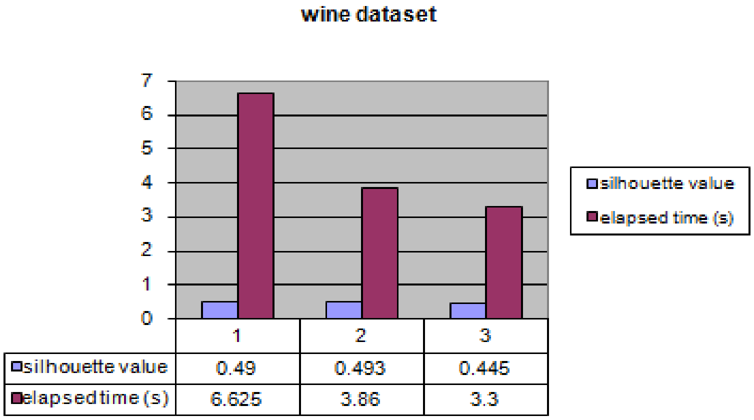
	1	2	3
silhouette value	0.248	0.64	0.603
elapsed time(s)	3.118	1.95	2.9

Figure2. Comparison of silhouette value, execution time for k=3



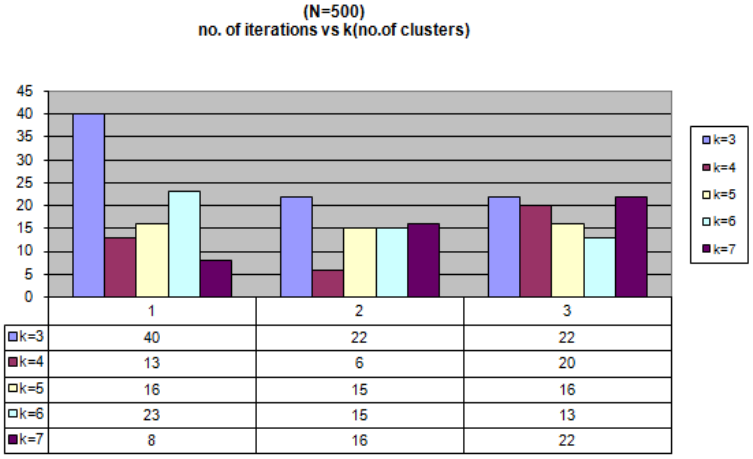
	1	2	3
no. of iterations	16	9	7
points misclassified	45	23	23
accuracy	74.71	87.07	87.07

Figure3. Comparison of no. of iterations, no. of points misclassified, accuracy for k=3



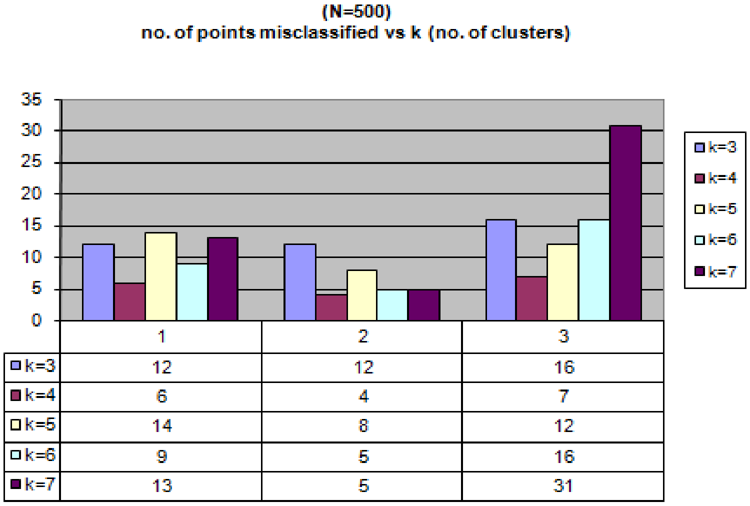
	1	2	3
silhouette value	0.49	0.493	0.445
elapsed time (s)	6.625	3.86	3.3

Figure4. Comparison of silhouette value, execution time for k=3



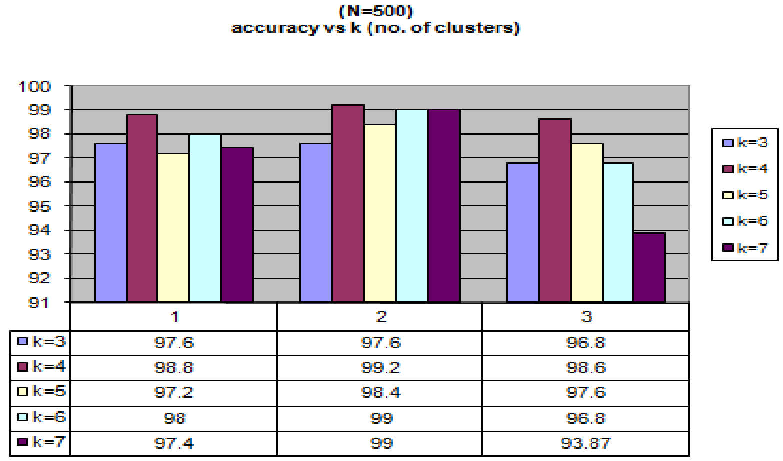
	1	2	3
k=3	40	22	22
k=4	13	6	20
k=5	16	15	16
k=6	23	15	13
k=7	8	16	22

Figure5. Comparing algorithms for no. of iterations vs. no. of clusters



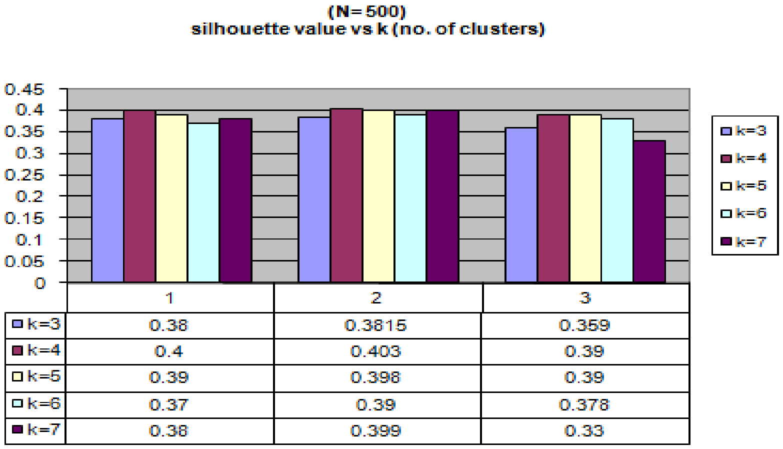
	1	2	3
k=3	12	12	16
k=4	6	4	7
k=5	14	8	12
k=6	9	5	16
k=7	13	5	31

Figure6. Comparing algorithms for no. of points misclassified vs. no. of clusters



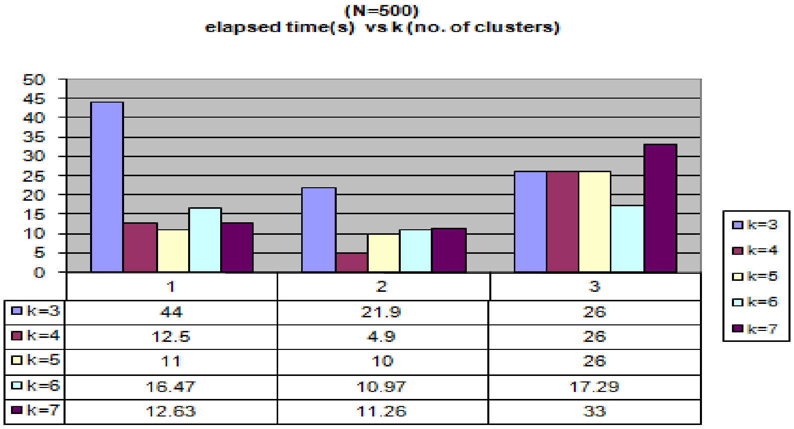
	1	2	3
k=3	97.6	97.6	96.8
k=4	98.8	99.2	98.6
k=5	97.2	98.4	97.6
k=6	98	99	96.8
k=7	97.4	99	93.87

Figure7. Comparing algorithms for accuracy vs. no. of clusters



	1	2	3
k=3	0.38	0.3815	0.359
k=4	0.4	0.403	0.39
k=5	0.39	0.398	0.39
k=6	0.37	0.39	0.378
k=7	0.38	0.399	0.33

Figure8. Comparing algorithms for silhouette index vs. no. of clusters



	1	2	3
k=3	44	21.9	26
k=4	12.5	4.9	26
k=5	11	10	26
k=6	16.47	10.97	17.29
k=7	12.63	11.26	33

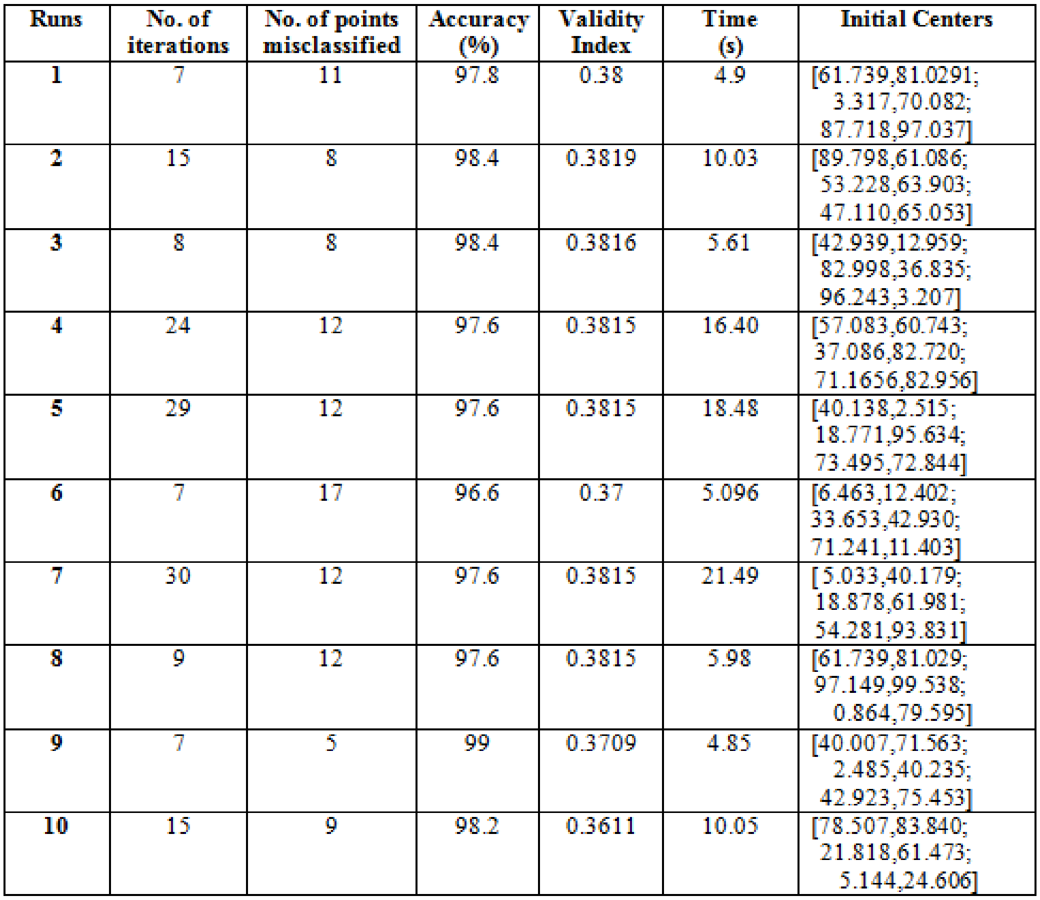
Figure 9. Comparing algorithms for execution time vs. no. of clusters

Result shows that modified approaches perform better than Original k-Means and also the modified approach 1 outperforms the approach 2 also. One more significant result is drawn when implementing the Original k-Means that on even fixing the number of clusters, Original k- Means algorithm provides different results in different runs i.e. “the initial center points are changing randomly even value of k is fixed, but this problem is not in other two modified approaches because it fixes the way initial centers are calculated.

Table 1 shows result for running Original k-Means for fixed value of k and how performance parameters are varying in each run. Here the value of k is fixed to 3 and a random data set of 178 points is used to find these parameters. It also shows that how different initial centers that are obtained during each run of algorithm. This shows that Original k -Means only attain local minima because running this algorithm at different runs will result in different initial centers every time.

From figure 8 it is shown that silhouette value for k = 4 is better thus desired number of clusters is 4 and also the value of silhouette index for modified approach 1 is better than other two algorithms. Figure 10 to 12 shows silhouette plots for 3 algorithms.

TABLE I- Performance measures for different runs of Original k –Means



Runs	No. of iterations	No. of points misclassified	Accuracy (%)	Validity Index	Time (s)	Initial Centers

1	7	11	97.8	0.38	4.9	[61.739,81.0291;
						3.317,70.082;
						87.718,97.037]

2	15	8	98.4	0.3819	10.03	[89.798,61.086;
						53.228,63.903;
						47.110,65.053]

3	8	8	98.4	0.3816	5.61	[42.939,12.959;
						82.998,36.835;
						96.243,3.207]

4	24	12	97.6	0.3815	16.40	[57.083,60.743;
						37.086,82.720;
						71.1656,82.956]

5	29	12	97.6	0.3815	18.48	[40.138,2.515;
						18.771,95.634;
						73.495,72.844]

6	7	17	96.6	0.37	5.096	[6.463,12.402;
						33.653,42.930;
						71.241,11.403]

7	30	12	97.6	0.3815	21.49	[5.033,40.179;
						18.878,61.981;
						54.281,93.831]

8	9	12	97.6	0.3815	5.98	[61.739,81.029;
						97.149,99.538;
						0.864,79.595]

9	7	5	99	0.3709	4.85	[40.007,71.563;
						2.485,40.235;
						42.923,75.453]

10	15	9	98.2	0.3611	10.05	[78.507,83.840;
						21.818,61.473;
						5.144,24.606]

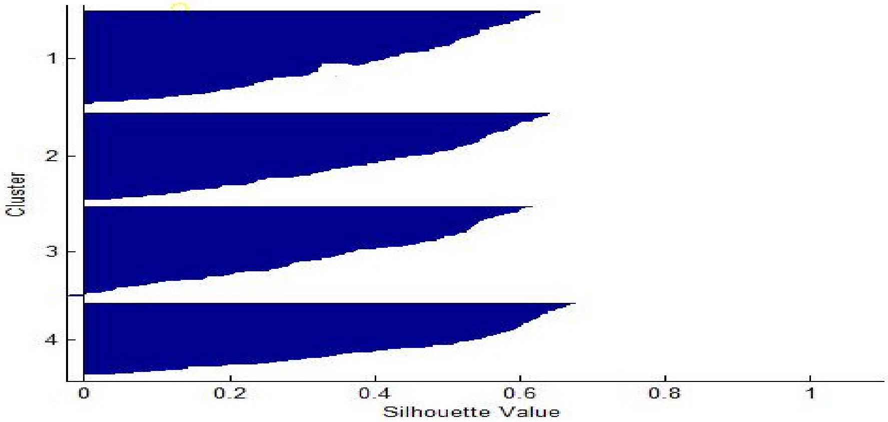


Figure10. Silhouette plot for Original k-means algorithm

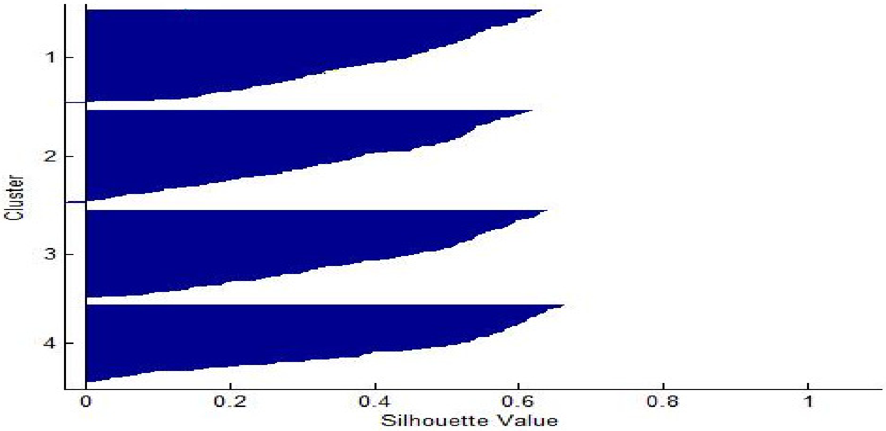


Figure11. Silhouette plot for Modified approach I

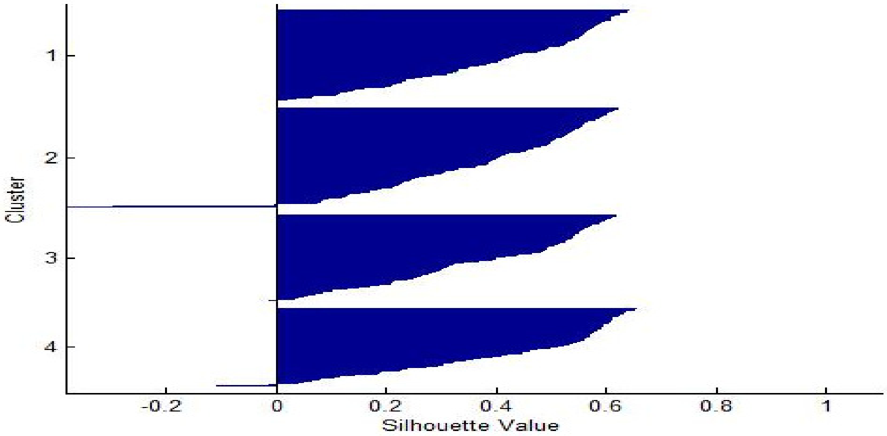


Figure12. Silhouette plot for Modified approach II

7. CONCLUSION

The algorithms are implemented using MATLAB and experiments are conducted and results are evaluated based on performance parameters. It is also shown that how Original k-Means always choose different initial centers at each run of the algorithm for fixed dataset and value of k but modifications to Original k-Means algorithm optimizes the way, the initial centers are selected for clustering. From our experiments on different datasets we also conclude that the modified approaches take less execution time for large datasets. We also used Silhouette validity index to compare which algorithm is producing compact and well separated clusters for given k. A future in this direction would be the application of modified algorithms on the complex datasets.

ACKNOWLEDGMENT

I am eternally grateful to my research supervisor Dr. M.P.S Bhatia for their invigorating support and valuable suggestions and guidance. I thank him for his supervision and patiently correcting me with my work. It is a great experience and I gained a lot here. Finally, we are thankful to the almighty God who had given us the power, good sense and confidence to complete my research analysis successfully. I also thank my parents and my friends who were a constant source of encouragement. I would also like to thanks Navneet Singh for his appreciation.

REFERENCES

[1] Ran Vijay Singh and M.P.S Bhatia , "Data Clustering with Modified K-means Algorithm", IEEE International Conference on Recent Trends in Information Technology, ICRTIT 2011, pp 717-721.

[2] D. Napoleon and P. Ganga lakshmi, "An Efficient K-Means Clustering Algorithm for Reducing Time Complexity using Uniform Distribution Data Points", IEEE 2010.

[3] Tajunisha and Saravanan, "Performance Analysis of k-means with different initialization methods for high dimensional data" International Journal of Artificial Intelligence & Applications (IJAIA), Vol.1, No.4, October 2010

[4] Neha Aggarwal and Kriti Aggarwal,"A Mid- point based k –mean Clustering Algorithm for Data Mining". International Journal on Computer Science and Engineering (IJCSE) 2012.

[5] B, Bariseé Barisi Baridam," More work on k-means Clustering algortithm: The Dimensionality Problem ". International Journal of Computer Applications (0975 – 8887)Volume 44– No.2, April 2012.

[6] Shi Na, Li Xumin, Guan Yong "Research on K-means clustering algorithm". Proc of Third International symposium on Intelligent Information Technology and Security Informatics, IEEE 2010.

[7] Ahamad Shafeeq and Hareesha "Dynamic clustering of data with modified K-mean algorithm", Proc. International Conference on Information and Computer Networks (ICICN 2012) IPCSIT vol. 27 (2012) © (2012) IACSIT Press, Singapore 2012.

[8] Kohei Arai,Ali Ridho Barakbah, Hierarchical K-means: an algorithm for centroids initialization for K-means.

[9] Data Mining Concepts and Techniques,Second edition Jiawei Han and Micheline Kamber.

[10] "Towards more accurate clustering method by using dynamic time warping" International Journal of Data Mining and Knowledge Management Process (IJDKP), Vol.3, No.2,March 2013.

[11] C. S. Li, "Cluster Center Initialization Method for K-means Algorithm Over Data Sets with Two Clusters", "2011 International Conference on Advances in Engineering, Elsevier", pp. 324-328, vol.24, 2011.

[12] A Review of Data Clustering Approaches Vaishali Aggarwal, Anil Kumar Ahlawat, B.N Panday. ISSN: 2277-3754 International Journal of Engineering and Innovative Technology (IJEIT) Volume 1, Issue 4, April 2012.

[13] Ali Alijamaat, Madjid Khalilian, and Norwati Mustapha, "A Novel Approach for High Dimensional Data Clustering" 2010 Third International Conference on Knowledge Discovery and Data Mining.

[14] Zhong Wei, et al. "Improved K-Means Clustering Algorithm for Exploring Local Protein Sequence Motifs Representing Common Structural Property" IEEE Transactions on Nanobioscience, Vol.4., No.3. Sep. 2005. 255-265.

[15] K.A.Abdul Nazeer, M.P.Sebastian, "Improving the Accuracy and Efficiency of the k-means Clustering Algorithm",Proceeding of the World Congress on Engineering, vol 1,london, July 2009.

[16] Mu-Chun Su and Chien-Hsing Chou "A Modified version of k-means Algorithm with a Distance Based on Cluster Symmetry".IEEE Transactions On Pattern Analysis and Machine Intelligence, Vol 23 No. 6 ,June 2001.

Authors

Dr. M.P.S Bhatia. Professor at Department of Computer Science & Engineering NETAJI SUBASH INSTITUTE OF TECHNOLOGY, UNIVERSITY OF DELHI, INDIA.

Deepika Khurana Mtech. II year in Computer Science & Engineering (Information Systems) from NETAJI SUBHASH INSTITUTE OF TECHNOLOGY, UNIVERSITY OF DELHI, INDIA