International Journal of Computing Academic Research (IJCAR)
ISSN 2305-9184, Volume 5, Number 2 (April 2016), pp.88-103

http://www.meacse.org/ijcar

An Improved K-Means Clustering Algorithm

Asmita Yadav and Sandeep Kumar Singh

Jaypee Institute of Information Technology, Noida, Uttar Pradesh, India

Abstract

Lot of research work has been done on cluster based mining on relational databases. K-means is a basic algorithm, which is used in many of them. The main drawback of k-means is that it does not give a high precision rate and results are affected by random initialization of cluster centroids. It may produce empty clusters depending on the initial centroids, which reduce the performance of the system. In this paper, we have proposed an Improved K-means algorithm, which improve data clustering by removing empty clusters. Further, it improves the computational time of the algorithm by reusing stored information of previous iterations. The results obtained from our experiments show improvement in accuracy, precision rate and efficiency of the algorithm. The complexity of the algorithm is also reduced from O(nlk) to O(nk).

Keywords: K-means algorithms, efficient enhanced k-means algorithm, m-k means

1. Introduction

The activity of shifting through huge files and databases to discover useful, non-obvious and often unexpected trends and relationships is called Data mining [1]. It helps to find patterns among the data, using predictive methods such as classification, clustering or regression analysis. The goals of data mining can be summarized in terms of two types of activities: discovery of new patterns and verification of a user's hypothesis about patterns. Data mining have two basic components - *cases and feature*. A *case* is a specific event, commonly represented as a record and *feature* is a particular measurement on the data, also called attributes [2]. Data mining can be performed by many ways, like: predicative modeling, clustering, data summarization and dependency modeling etc.

Clustering divides the data-objects into different groups or components. This phenomenon is used by data mining process to manage a huge or large data-set [3,4] according to similar attributes or features.

A general definition of clustering is, to group the similar featured data-objects into one cluster and dissimilar in other one. The meaning of similar feature is represented by the minimum

Euclidean distance between data-object and cluster centroid. Clustering can be done by various methods like: Partitioning method, Hierarchical method, Grid-based method, Density- based method and Model–Based method etc.

Clustering algorithm follows some important necessities [5,6,19], like:

1. Data collected for clustering process, should be in uniform manner.
2. Clustering Algorithm should be able to handle diverse types of feature.
3. Distribution of data clusters should be such that data objects in one cluster should be similar (related) to one other and dissimilar from the data objects in other cluster.
4. Clustering algorithm should be able to remove all noise and outliers from data sets. We treat empty cluster as outliers and proposed improved k-means algorithm that remove outliers (empty clusters).

1.1 K-Means

K- Means is the first clustering algorithm which is proposed by James MacQueen in 1967, though the idea goes back to 1957[7, 8]. It uses greedy approach to cluster data which may result in a non optimal solution. It groups the data-objects into different predefined number of clusters (k), according to minimum Euclidean Distance between data- objects and cluster centroids. The phenomenon of K-Means [7, 8] is depicted in below:

Inputs: (i) C_k is the set of points that belong to cluster k (ii) Group of Data-objects to be clustered (iii) Number of clusters (k)

Output: Clusters of the data-object.

Step 1: Randomly select initial cluster centroids value from C_1 to C_k.

Step 2: Repeat step 3 to 5.

Step 3: Calculate Euclidean distance for each data-objects between cluster centroids.

Step 4: Assign each data-object to the nearest cluster centroid.

Step 5: Recalculate the cluster centroids of each cluster.

Step 6: Until, to get convergence criterion.

1.2 Limitations of K-means algorithm

Following limitations of k-means algorithms are identified.[9,10,11,12]-

K-means is an expensive algorithm and takes more computational time to cluster the data-objects. Its complexity is O(nlk), where n is total number of data-objects, l represent the number of iteration and k is total number of cluster. This algorithm is sensitive to the selection of initial centroids, which heavily affects the quality of resulting clusters and fails to get optimal solution. Basic k-means [3] and efficient k-Means [13] affect the performance of the algorithm by producing empty clusters.

One of the main reasons for this problem is bad initialization of centroids. Because, if all clusters are having same centroids, the Euclidean distance computed for each data-object comes out to be the same. Therefore according to K-means and efficient k-mean algorithm all data-objects become part of single cluster and other clusters remain empty which produces anomalous behavior of the system [12]. Although this problem can be solved by repeating the initialization until the removable of empty cluster, but re-initialization takes too much time and reduces performance.

To overcome this problem, we proposed a refinement technique to eliminate the generation of empty clusters. Our proposed method produced better results as that of efficient k- means with minimum proportion of time (O(nk)). It has superior performance in terms of precision rate, computational speed and complexity.

The outline of the paper is as follows: Section 1 briefs about existing K-Means clustering algorithm, Section 2 discusses related work and section 3 presents improved K-Means algorithm. We demonstrate experimental results and compression with efficient enhanced k-means algorithm in section 4. Finally paper is concluded in section 5.

2. Literature Review

Research for clustering in the various techniques started in the beginning of 1990s. In this section, we review some of the improved clustering algorithms over the basic k-means algorithm.

Malay K. pakhira [12] proposed a modified version of k-means. Centroids are treated as a data-object and successfully avoid the generation of empty clusters. But semantically, it is equivalent to basic k- means algorithm in complexity O(nlk) and have same convergence rate. Our approach is better from this modified version of k-means in respect for computational speed and time complexity O(nk).

In this approach, Fahim A.M.,Salem A.M.,Torkey F.A.,Ramadan M.A., [13] presented a new way to reduce the number of iterations by storing the previous iteration data in a simple data structure. If initial cenroids assortment is done randomly. It suffered for optimum solution of the

problem because the generation of empty clusters. This drawback is reduced in our approach by treating cluster centroid also as a data-object to avoid empty cluster.

An algorithm is proposed by Rajeev Kumar, using min-heap and red black tree [14], which produced better performance as compared to basic k-means. This algorithm used two types of data-structure for storing data-objects which increased the space complexity. But our proposed algorithm uses only one type of data-structure to store the data-object and also has better computational speed.

A systematic method is proposed by Fang Yuan [15] to find out the initial centriods with the consistent distribution of data. This approach is good in terms of accuracy as compared to k-means algorithm. None of the improvement is done with respect to time complexity and accuracy. Both these issues are addressed in our approach.

Neha Aggarwal & Kirti Aggarwal [16] presented the way to find the initial centroids for the k-means and it produced same outcome by using mid-point based K-means algorithms. This algorithm tried to remove few limitations of k-means. These results closely depend on the selection of initial centroids which causes it to converge at local optimum. But suffered for poor computational speed and produce empty cluster

A semi- supervised k-means algorithm is proposed by Xue Sun [17] by using global optimization techniques. A voting rule is used to guide the cluster labeling in data –sets. This is better than k-means and improves the quality the cluster efficiently and provides an optimum solution of the problem. But number of clusters is pre-defined in the algorithm which may lead to empty clusters in the end of the process. The solution of this problem is provided in our approach with better computational speed and time complexity.

3. Improved K-Means Algorithm

The approach followed by us makes K-means algorithm more effective and efficient by removing the first limitation i.e. to reduce the number of iterations by using previous iteration data for clustering the data-objects. In K-Means algorithm, Euclidean distance between all data-objects and centroids are recomputed in each iteration and all the data-objects are redistributed into the nearest cluster. This process involves overhead in terms of the computational time of the algorithm. To overcome this problem, in phase 1, we stored previously computed iteration data for next iteration, in a multi-dimensional array. Through Backtracking method, previous iteration data is used in next iteration which reduces the computational time of the algorithm. Each iteration process, the nearest cluster distance with cluster number is stored in cluster-id[i]. In next iteration, we calculate the Euclidean distance from the previous nearest cluster. In phase 2, The data-object stays in its previous cluster, if the new Euclidean distance is less or equal from the previous distance. In this case, there is no need to calculate its Euclidean distance from the

remaining cluster centroids. In this process, we only calculate the Euclidean distance to (k-1) cluster centroid and computational time is reduced.

After data-objects assignment, we need to calculate the new centroids for each cluster. This can be done by calculating the mean of all data-objects within the same cluster. Centroid is also treated as a data- object in this process. This removes the generation of empty clusters and improves the rate of convergence of the algorithm.

This algorithm is divided into two phases.

Phase 1: Initial assignment of Data objects to its closest cluster

Input: n= total number of data-Objects., k= number of clusters., x_i= ith Data- objects

Output: Cluster-id= number of the closest centroid, Euclidendis=Euclidean distance to the closest centroid, $m_j^{(new)}$ =New Centroids.

Begin

1. For i = 1 to n
2. For j = 1 to k

Compute squared Euclidean distance $d_2(x_i, m_j^{(new)})$;

End

Find the closest centroid $m_j^{(new)}$ to x_i;

$m_j^{(new)} = m_j^{(old)} + x_i$; $n_j^{(new)} = n_j^{(old)} + 1$;

MSE = MSE + $d_2(x_i, m_j^{(new)})$;

Cluster-id[i] = number of the closest centroid;

Euclidendis[i] = Euclidean distance to the closest centroid;

End

3. For j = 1 to k

$m_j^{(new)} = m_j^{(new)} / n_j^{(new)}$;

End

4. End

As we know, n and k is total number of data – objects in data set and number of clusters respectively. Our purpose is to group all data- objects into different clusters based on similar features or attributes. From step 2 , the function calculate the Euclidean distance for each data

objects x_i from all cluster centroids k and calculate the closest cluster centroids j from each data-objects. So that data- objects are grouped in clusters on the basis of minimum distance. And then, we analyze the total number of data-objects n in cluster j and centroids is also treated as a data-object and update the total number of data-objects in one cluster. The information of closest cluster and closest Euclidean distance are kept in Cluster-id[i] and Euclidendis[i] respectively. This is important part of our proposed idea which reduces the computational time of the algorithm and further stored iteration information is also used in the allocation of data-objects. Now in step 3, we recalculate the new centroids for each cluster using information of the previous cluster centroid and data-objects.

Next step of algorithm is processed in Phase 2.

Phase 2 : Re-assignment of data-objects

Inputs : n, k, x_i , Clusterid , Euclidendis, $m_j^{(new)}$ =New Centroids.
Output: Clusters of data-Objects without empty cluster.

Begin

1. For i = 1 to n
2. Calculate squared Euclidean distance d_2 (x_i, Cluster-id[i]);
3. If (d_2 (x_i, Cluster-id[i]) <= Euclidendis[i])
   Data-point resides in its cluster;
4. Else
   For j = 1 to k
   Compute squared Euclidean distance $d_2(x_i, m_j^{(new)})$;
   End
   Find the closest centroid $m_j^{(new)}$ to x_i;
   $m_{j+1} = m_j + x_i$; $n_{j+1} = n_{j+1}$;
   MSE = MSE + $d_2(x_i, m_j)$;
   Cluster-id[i] = number of the closest centroid;
   Euclidendis[i] = Euclidean distance to the closest centroids;
   End
5. For j = 1 to k
   $m_j^{(new)} = m_j^{(new)} / n_j^{(new)}$;
6. End

Working process of phase 2 is same as phase 1. In step 2, calculate new Euclidean distance between new cluster centroids and current data-objects. In step 3, If the computed new distance is less than or equal to previous distance to the old cluster centroid, then data-object stays in its cluster that was assigned to it in previous iteration. There is no requirement to calculate the distance from rest of the k-1 cluster centroids. In step 4, i.e. If computed distance is larger than the previous distance to the centroids, because the data- objects may change its cluster, so We again computes the Euclidean distance between the all K cluster centroids and current data-object. We again search for closest cluster centroids for each data- objects. Assigns the current point to the closest cluster and increases the count of data-objects in the newly assigned cluster by one. The cluster centroid is also treated as a data- object, which avoid, empty cluster problem because one of the data-objects is still in cluster. MSE updates the means squared error for each data-object. Cluster-id[i] and Euclidendis[i] is again calculated, which keep the information for the current cluster assigned to it, and its Euclidendis to it to be used for the next iteration to reduce the recalculation to assign each point to the closest cluster. This reusable function makes the algorithm faster than the k-means algorithm.

Our proposed algorithm is similar to efficient k-means algorithm, only difference is that in our case, the cluster centroid is treated as data-objects of the respective cluster and this can be written as:

$$m_j^{(new)} \leftarrow 1/n_j \{ \sum_{xi \in k\,j} (x_i) + m_j^{(old)} \} \quad (1)$$

Due to which, at least one data-object will always be present in each cluster in the form of Centroids. Under the similar condition, Enhanced Efficient k-means algorithm fails to remove empty cluster problem but our proposed improved algorithm is able to remove the generation the empty cluster problem with high precision rate and sustain the complexity i.e. o(nk) and also the cost of improved k-means algorithm is lesser than the cost of basic k-means i.e. O(nlk).

4. Result Analysis and Discussion

We showed that our proposed algorithm is capable to solve the empty clusters problem. To validate our proposal, we took a real bank data set to demonstrate the applicability of our algorithm, that the improved k-means algorithm removes the existing limitations in efficient k-means with better quality of data clustering. Bank data set used for experimental work is taken from Delve [18]. This dataset contains 8192 details of bank employees and clients. Bank employees and bank clients are categorized on 9 and 33 attributes respectively, like emp-salary, working experience, residential area, account details, account balance etc. By using our proposed algorithm, we clustered clients as well as bank employees' data objects. Some of the client and employee attributes are shown in Table 1 and Table 2.

Table 1: Bank Client Attributes

Attribute Name	Description
Age	Numeric
Job	Type of job: admin.,blue-collar,entrepreneur,housemaid,management,retired,self-employed,services,student,technician,unemployed,unknown
Material	Marital status: divorced,married,single,unknown
Education	Categorical:basic.4y,basic.6y,'basic.9y,'igh.school,illiterate,professional.course,university.degree
Loan	Category: Yes or No
Contact-no.	Mobile Phone or Telphone
Month	Last contact month of year
day_of_week	Last contact day of the week
Duration	Last contact duration, in seconds
Pdays	Number of days that passed by after the client was last contacted from a previous campaign
Previous	Number of contacts performed before this campaign and for this client (numeric)
Poutcome	Outcome of the previous marketing campaign
emp.var.rate	Employment variation rate - quarterly indicator
nr.employed	Number of employees - quarterly indicator

Table 2: Bank Employee Attributes

Attributes Name	Description
Age	Numeric
Salary	Gross salary in rupees
Working Experience	Total (previous+ current) experience in year
Job-title	Designation: Manager, cashier etc

Authority	Which area is accessed by him or her.
Contact Number	Mobile or landline
Working_ mode	Traditional or internet
Education	Categorical:basic.4y,basic.6y,'basic.9y,'igh.school,illiterate,professional.course,university.degree
Office _location	Where he worked.

Precision and recall is calculated on the basics of clusters quality.

Precision: Number of similar attributes of data-objects over total retrieved attributes of data-objects.

Recall: number of similar attributes of data-objects over the total number of possible similar attributes of data objects.

Example: For demonstration of algorithm feasibility, we show it on a subset of very small 2-dimensional data set (8 Bank employees with two attributes). Our goal is to groups these objects into k=3 groups (no empty cluster) based on the two attributes (salary and experience).

Table 3: Employee data set

Employee	**Salary (Lakh)**	**Experience (Year)**
Emp1	1	4
Emp2	2	6
Emp3	4	7
Emp4	6	8
Emp5	7	10
Emp6	8	12

Emp7	9	14
Emp8	10	15

Like k-means our approach also uses random values for initial centroids. We chose the initial clusters and set to the same value (3, 6), as 3 and 6 shows the salary of employee and total experience of the employee respectively. i.e. m_1=(3,6), m_2=(3,6) and m_3=(3,6). Following iteration shown below demonstrate how our improved k-means algorithm classifies the data set into three clusters C(1), C(2) and C(3) by removing empty clusters and reduce number of iterations by using Employee data set from Table 3. Centroid distance of emp1 to emp8 data-objects from the centroids of cluster C1, C2 and C3 is shown in Table 4 to 7.

Iteration 0: Find the data- object centroid distance from Table 3 with initial centroids values of C1 (3, 6), C2 (3, 6), C3 (3, 6).

Table 4: Object centroid distance matrix 1

	Emp1	Emp2	Emp3	Emp4	Emp5	Emp6	Emp7	Emp8
C1	2.83	1	1.41	3.61	5.66	7.81	10	11.4
C2	2.83	1	1.41	3.61	5.66	7.81	10	11.4
C3	2.83	1	1.41	3.61	5.66	7.81	10	11.4

All data objects that have same distance from each cluster centroids assigned to any one of the cluster from C1, C2 or C3. In this situation, we assigned C1 cluster for all data-objects. That why rest of the two clusters C2 & C3 are empty as shown in Table 4. Now, we proceed to next iteration.

Iteration 1: By considering the Table 4 result**,** we find new centroids based on new data points distance and cluster centroids. Centroids of Cluster 2 & 3 shall be unchanged because these are empty. But for cluster 1, new centroids is = ((1 + 2 + 4 + 6 + 7 + 8 + 9 + 10+3) / (8+1), (4+6+7+8+10+12+14+15+6)/(8+1))= (5.56, 9.11), then object- centroids distance are:

Table 5: Object centroid distance matrix 2

	Emp1	Emp2	Emp3	Emp4	Emp5	Emp6	Emp7	Emp8
C1	6.85	4.72	2.62	1.19	1.69	3.785	5.98	7.38
C2	2.83	1	1.41	-	-	-	-	-

C3	2.83	1	1.41	-	-	-	-	-

Cluster C1 and C2 contains (emp4, emp5, emp6, emp7, emp8), and (emp1, emp2, emp3) respectively. In Table 5, C3 has no data-object.

Iteration 2: C3 have same centroid (3, 6), while C1 and C2 centroids have changed, then C1= (7.59, 11.35) & C2= (2.5, 5.75) and again find object centroids distance [data taken from Table 5].

Table 6: Object centroid distance matrix 3

	Emp 1	Emp 2	Emp 3	Em 4	Emp 5	Emp 6	Emp 7	Emp 8
C1	-	-	5.64 4	1.19	1.48	.77	2.99	4.37
C2	2.3	.56	1.95	4.16	-	-	-	-
C3	-	-	1.41	3.61	-	-	-	-

In Table 6, data object arrangement in clusters shall be C1 (emp5, emp6, emp7, emp8), C2 (emp1, emp2) and C3 (emp3, emp4). According to this, none of the cluster is empty but for finding stability of data objects, we need to compute an extra iteration.

Iteration 3: All three clusters changed its centroids as C1 (8.32,12.47), C2(1.83,5.25) & C3(4.33,7).

Table 7: Object centroid distance matrix 4

	Emp 1	Emp 2	Emp 3	Em 4	Emp 5	Emp 6	Emp 7	Emp 8
C1	-	9.04	-	-	2.8	.568	1.68	3.04
C2	1.5	.77	-	-	-	-	-	-
C3	-	2.53	.33	1.95	-	-	-	-

Table 7 shows the final clustering results. Data objects positions as C1 (emp5, emp6, emp7, emp8), C2 (emp1, emp2) and C3(emp3,emp4). Consider the iteration 2 and iteration 3, data objects are in same clusters, and none of the cluster is empty. This is stopping entire of our proposed improved k-means algorithm.

The formation of clusters in different iterations using proposed improved k-means as well as existing efficient k-means[13] algorithm is shown in Table 8.

Table 8: Comparison of Improved K-Means and Efficient K-means algorithm using Employee data-set

Iteration No.	**Clusters**	**$m^{(old)}$**	**Elements in clusters(Proposed Improved K- means algo)**	**$m^{(new)}$**	**Elements in clusters (Existing efficient K-means algo) [13]**
0	C(1)	(3,6)	1,2,3,4,5,6,7,8	(5.56, 9.11)	1,2,3,4,5,6,7,8
	C(2)	(3,6)	Empty	(3,6)	Empty
	C(3)	(3,6)	Empty	(3,6)	Empty
1	C(1)	(5.56,9.11)	4,5,6,7,8	(7.59, 11.35)	1,2,3,4,5,6,7,8
	C(2)	(3,6)	1,2,3	(2.5, 5.75)	Empty
	C(3)	(3,6)	Empty	(3,6)	Empty
2	C(1)	(7.59,11.35)	5,6,7,8	(8.32, 12.47)	NA
	C(2)	(2.5, 5.75)	1,2	(1.83, 5.25)	NA
	C(3)	(3,6)	3,4	(4.33,7)	NA
3	C(1)	(8.32, 12.47)	5,6,7,8	(8.47, 12.69)	NA
	C(2)	(1.83,5.	1,2	(1.61, 5.08)	NA

		25)			
	C(3)	(4.3 3,7)	3,4	(4.78, 7.33)	NA

The above experimental results show that the proposed improved k-means algorithm gives optimal number of clusters for a dataset without producing empty clusters. After reaching the convergence criteria, efficient k-means algorithm stopped at the first iteration, and have 2 empty clusters and for rest of the iteration, this algorithm is not applicable (NA).

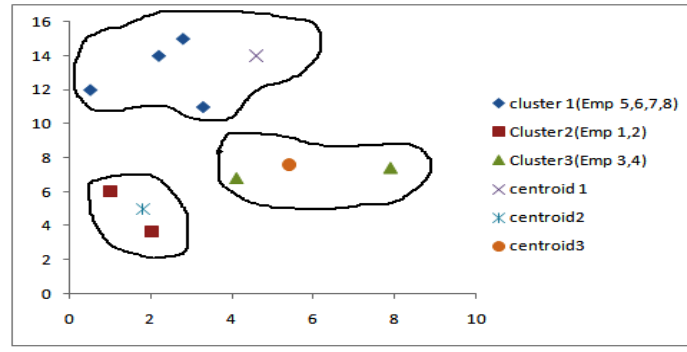

Figure 1: Improved k-means based graphical representation

For numerical analysis, simulation and experimental purpose we have used MATLAB-8 simulation environment.

According to proposed improved k-mean algorithm, the partitions become {5, 6, 7, 8}, {1,2} and {3,4}. That means, cluster C1 contains data objects (emp5,emp6,emp7 and emp 8) , C2 contains (emp 1 and emp 2) and cluster C3 have (emp 3 and emp 4), none of the cluster is empty as shown in Figure 1.

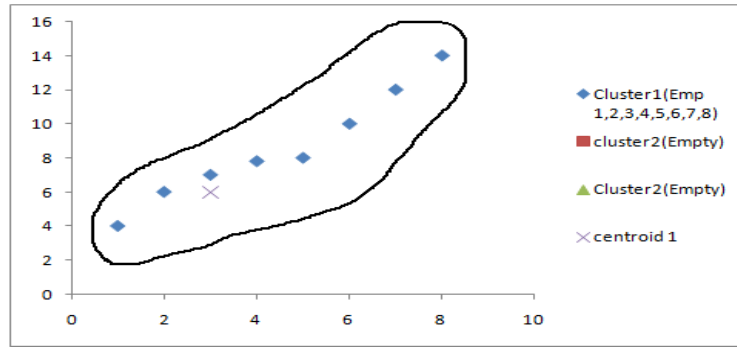

Figure 2: An efficient enhanced K-means clustering algorithm based graphical representation

But, under the similar initial condition, using existing efficient enhanced k-means [13] produced two empty clusters (C2 & C3) are produced because all data objects are in one cluster C1(Figure. 2). This case, the improved k-means algorithm is found to produce a good clustering while the efficient enhanced k-means algorithm fails.

Figure 3: Execution time of efficient enhanced k-means and Improved K-Means

We have also compared the time required for the data-object clustering execution. As shown in Figure 3, improved k-means took less time for data distribution in clusters as compare to efficient k-means algorithm.

Analysis of Proposed Algorithm

The complexity of basic k-means algorithm is O(nkl), where n, k and l represents the number of data-objects, total clusters and number of iteration respectively. After calculating the new centroids, data-objects are completely redistributed. The whole process increased the computational time of the algorithm. Our approach reduced computational time of the algorithm by using previous stored iteration data and requires O(nk) complexity. It holds some data-objects in the cluster, although some shift from one to other cluster depending on their minimum Euclidean distance from the new and old centroids. If data-objects are moved from the cluster than O(k) time is required otherwise O(1) for the algorithm to converges, It requires O(nk/2), when half of the data-objects are moved from the cluster. In each iteration, data-objects movement is decreased from one to other cluster. So the total estimated cost is $nk\sum 1/i$, where i=0 to l. Even for the large number of iteration,

i=1;

And the value of $nk\sum l/nk$ is less than $\sum nkl$. That means, improved k-means algorithm cost is approximately O(nk), which is much less than O(nkl). The accuracy and efficiency is shown is Figure 4 also.

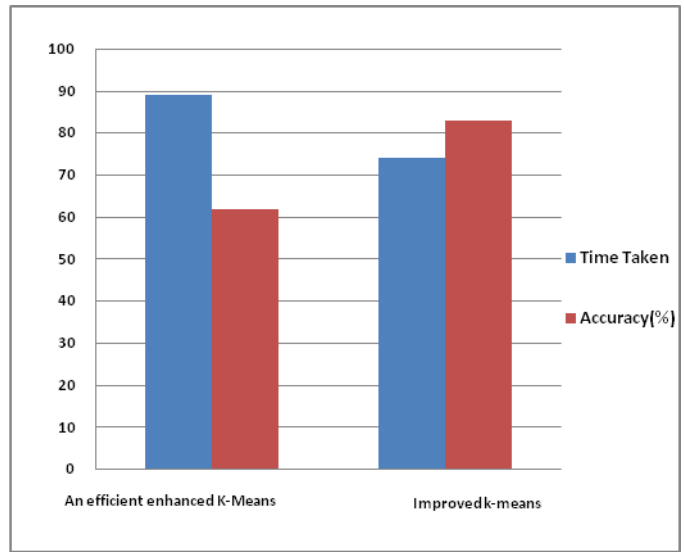

Figure 4: Accuracy and efficiency of Algorithms

5. Conclusion

In this paper, we improved the working efficiency of the existing efficient K-Means algorithm. This algorithm suffered the empty cluster problem and sometimes not able to give the optimal solution of the problem.

Our proposed algorithm is competent to eliminate these problems. We reduce computational complexity of the algorithm by reusing previous iteration data in current or next iteration for clustering the data-objects. In this algorithm, centroids is also treated as an data-object in that respective cluster, which bestow a help to avoid empty cluster problem and provide an optimal solution of the problem because all the data-objects are stored in its nearest cluster. From the result, it can be show that, our proposed improved K-Means Algorithm is the batter than the existing efficient k-means algorithm. In future, we want to do some work to reduce the space complexity and random selection of initial centroids. We are also planning to replicate our work on very large and high dimensional datasets for clustering.

References

[1] Vladan Devedzic, "Knowledge Discovery and Data Mining in Databases", Fon-School of Business Administration , University of Belgrade, Yugoslavia, page no 2-8, (2004)

[2] Peng Jin, Yun-long Zhu, Kun-yuan Hu , "A Clustering Algorithm for Data mining based on Swarm Intelligence", Proceedings of Sixth International Conference on Machin[e Learning Cybernetics, Hong Kong, 19-22, (2007)

[3] Vance Faber, "Clustering and the Continuous k-Means Algorithm" Los Alamos Science Number 22 (1994)

[4] H. G. Wilson, B. Boots, and A. A. Millward, "A Comparison of Hierarchical and partitional Clustering Techniques for Multispectral Image Classification", IEEE, 0-7803-7536-X (C), (2002)

[5] Bo Ji and Yangdong Ye, "An improved sIB algorithm for document clustering Using combination weighting measures" , IEEE , 978-1-4244-8728-8/11, (2011)
[6] Shi Na, Liu Xumin and Guan yong, "Research on k-means Clustering Algorithm An Improved k-means Clustering Algorithm", Third International Symposium on Intelligent Information Technology and Security Informatics, IEEE , 978-0-7695-4020-7/10, (2010)
[7] Chau, M., Cheng, R., and Kao, B., "Uncertain Data Mining: A New Research Direction", in Proceedings of the Workshop on the Sciences of the Artificial, Hualien, Taiwan, December 7-8, (2005)
[8] K.A. Abdul Nazeer, M. P. Sebastian, "Improving the Accuracy and Efficiency of the k. Means Clustering Algorithm" world congress on Engineering 2009 vol 1, (2009)
[9] M. Srinivas and C. Krishna Mohan, "Efficient Clustering Approach using Incremental and Hierarchical Clustering Methods", IEEE, 978-1-4244-8126-2/10, (2010)
[10] Ahamed Shafeeq B M and Hareesha K S, "Dynamic clustering of data with modified K-means algorithm", International conference on Information and Computer Networks(ICICN) (2012)
[11] Asmita Yadav, "Study of K-Means and Enhanced K-Means Clustering Algorithm", International Journal Of Advanced Research In Computer Science, 4 (10), 103-107, Sept–Oct, (2013)
[12] Malay K. Pakhira, "A Modified k-means Algorithm to Avoid Empty Clusters " International Journal of Recent Trends in Engineering, Vol 1, No. 1, 220, May (2009)
[13] Fahim A.M.,Salem A.M.,Torkey F.A.,Ramadan M.A., "An efficient enhanced k-means clustering algorithm", JZUS, 7(10):1626-1633, (2006)
[14] Rajeev Kumar, Rajeshwar Puran and Joydip Dhar, "Enhanced K-Means Clustering Algorithm Using Red Black Tree and Min-Heap", International Journal of Innovation, Management and Technology, Vol. 2, No. 1, February, 2011 , ISSN: 2010-0248
[15] Yuan F,Meng Z.H, Zhang H.X and Dong C.R, "A New Algorithm get the initial Centroids," Proc. Of the 3rd international Confrence on Machine learning and Cybernetics, pages 26-29, Aug 2004
[16] Neha Aggarwal et al., "A mid-point based k-means clustering algorithm" International Journal on Computer Science and Engineering (IJCSE), Vol 4 No. 06 June 2012, ISSN : 0975-3397
[17] Xue Sun, Kunlun li,Rui Zhao, Xikun Hu,"Global Optimaziation for semi-supervised k-means", Asia-pacific Confrence on information Processing(APCIP), vol 24,Pp.410-413,2009
[18] http://www.cs.toronto.edu/~delve/data/datasets
[19] Priyanka Sharma, anu aggarwal, "Modified dynamic algorithm of data clustering using fuzzy c mean algorithm", International Journal of Recent Technology and Engineering (IJRTE) ISSN: 2277-3878, Volume-1, Issue-3, August 2012