Journal of AL-Qadisiyah for computer science and mathematics
Vol.6 No.2 Year 2014

Modify Initialization k-means Clustering Algorithm to Generate Initial Centroids

Lamia AbedNoor Muhammed

College of Computer and Mathematics

University of Al-Qadissiyah

bon1491988@yahoo.com

Recived :22\7\2014 Revised : 27\8\2014 Accepted :4\9\2014

Modify Initialization k-means Clustering Algorithm to Generate Initial Centroids

Abstract

K-means is one of the most common clustering techniques used with numeric data. Different issues are conducted in k-means algorithm in order to reach the optimum solutions with best situations, weather producing good results or the ways used to produce the results efficiently. Initial centroids of this algorithm play important role, so the generation initial centroids attracting more work. However, this paper aims to discuss a new proposed step to improve the generation of initial centroids i.e. modification the first iteration of k-means algorithm. The experiment work of this paper would be applied with one of the famous data that is "iris", this data is suited with k-means algorithm. The experiments were tested with the origin k-means algorithm in two parameters: "execution time" and "cost function" that is represented by sum square error SSE. The results are promise work with this modification

Keywords: centroids, clustering, initial centroids k-means clustering

I. Introduction

Clustering algorithm has a broad attraction and usefulness in exploratory data analysis[1]. Thousands of clustering algorithms have been proposed in the literature in many different scientific disciplines[2]. The k- Means clustering algorithm is an old algorithm that has been intensely researched owing to its ease and simplicity of implementation[1]. The research works in this field have been varied in different issues. Initialize the k-means algorithm with center objects that are called centroid is a critical problem because this algorithm is sensitive to these values and may reach to local minima. So there are some research work to deal with this problem. One technique that is commonly used to address the problem of choosing initial centroids is to perform multiple runs, each with a different set of randomly chosen initial centroids, and then select the set of clusters with the minimum sum of the squared error (SSE)[3].

The work in [4], suggested the creation of initial centroids through generate iteratively these centroids from n objects. At each time select the centroid that has maximum distances from the previous centroids. Other

work in [5], the creation of initial centroids is performed through selection portion of data and then generate the best ones from. In [6], k-means algorithm was modified i.e. partitions the whole space into different segments and calculates the frequency of data point in each segment. The segment which shows maximum frequency of data point will have the maximum probability to contain the centroid of cluster. In[7] , compute the distance between each data point and all other data points. Then the put the closest data points in set data points in subsets that represent the cluster and compute the centroids for each one by averaging the points data in subset. In[8], the proposed algorithm search about the points data that has maximum distance between them in order to present as centroids.

II. Clustering

Clustering analysis is an important technique in the rapidly growing field known as exploratory data analysis and is being in a variety of engineering and scientific disciplines such as biology, psychology, medicine, marketing, computer vision, and remote sensing[9].
The process of grouping a set of physical or abstract objects into classes of similar objects is called clustering. A cluster is a collection of data objects that are similar to one another within the same cluster and are dissimilar to the objects in other clusters. A cluster of data objects can be treated collectively as one group and so may be considered as a form of data compression[10].

Cluster analysis organizes data by abstracting underlying structure either as a grouping of individuals or as a hierarchy of groups. The representation can then be investigated to see if the data group according to preconceived ideas or to suggest new experiments. Cluster analysis is a tool for exploring the structure of the data that does not require the assumptions common to most statistical methods. It is called "unsupervised learning" in the literature of pattern recognition and artificial intelligence[9].

III. K-Means Clustering

The most popular partitional algorithm among various clustering algorithms is K-mean clustering. K-means is an exclusive clustering algorithm, which is simple, easy to use and very efficient in dealing with large amount of data with linear time complexity[11]. 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[1].
K-means algorithm clusters observations into k groups, where k is provided as an input parameter. It then assigns each observation to clusters based upon the observation's proximity to the of the cluster. The cluster's mean is then recomputed and the process begins again[11] as shown in Algorithm(1) [12].

Algorithm(1): Traditional k-means Algorithm
a) Select k points as initial centroids b) Repeat c) From k clusters by assigning each point to its closest centroid d) Recompute the centroid of each cluster until centroid does not change

The objective function of K-mean algorithm is to minimize the intra-cluster distance and maximize the inter-cluster distance based on the Euclidean distance[11]. So to measure the quality of a clustering , sum square error (SSE) is used i.e. the error for each point is computed. The SSE is formally defined as follows[3]:

$$SSE = \sum_{i=1}^{c} \sum_{x \in C_i} d(x, m_i) \qquad (1)$$

In the above equation, m_i is the center of cluster C_i , while *d(x,mi)* is the Euclidean distance between a point *x* and *mi* . Thus, the criterion function *E* attempts to minimize the distance of each point from the center of the cluster to which the point belongs [13].

The K-means algorithm requires three user-specified parameters: number of clusters K, cluster initialization, and distance metric. Cluster initialize with one object that is center for each cluster. A cluster is a set of objects such that an object in a cluster is closer (more similar) to the "center" of a cluster, than to the center of any other cluster. The center of a cluster is often a centroid, the average of all the points in the cluster[12].

Different initializations can lead to different final clustering because K-means only converges to local minima. One way to overcome the local minima is to run the K-means algorithm, for a given K, with multiple different initial partitions and choose the partition with the smallest squared error[2].

IV. Work Algorithm

The work algorithm contains proposed steps that would be applied in the generation initial centroids. The first iteration in k-means algorithm is modified in order to generate the initial centroid . The random selection of points data are candidate as centroids. Then execute the first iteration with the proposed modification as shown in Algorithm(2) i.e. the initial random points are selected to be initial centroid and then compute the distance between each point and the centroid. The minimum distance will make the point would be assigned to that cluster with minimum distance. After that the specific fraction from the difference (between the point and this cluster centroid) will be added to this centroid as shown bellow.

$$c(i,j) = c(i,j) + r * \left(x(p,j) - c(i,j)\right) \qquad (2)$$

where c represents centroid for cluster (i), x(p,j) is the point (p) in data space that was assigned to cluster (i), r is the fraction value .

Algorithm(2): proposed steps for first iteration of k-means algorithm

---

1- Input k random points as cenetroids for ***k*** clusters
2- Assign parameter ***r*** with value range between 0.1and1.0
3- Repeat
4- Present data point ***x***
5- Assign point p to the cluster i with minimum distance
6- Recompute centriod ***c*** of cluster ***i*** as :
   c(i,j) = c(i,j) + r * (x(p.j) - c(i,j))
   j : features of data point

V. Practical Work

A. Experiment work

The practical work was applied with experiment data that is known "iris" and the code was performed in MATLAB9000a. The traditional algorithm as shown in Algorithm(1) first was executed then proposed algorithm as shown in Algorithm(2) would be executed with different fraction value (r) parameter that

range(0.1,0.2,…..,1.0). The aim of the practical work is extracting the SSE and execution time for different runs of traditional and proposed algorithms. The runs are subjects to the following issues:

1- Execute the traditional and proposed algorithms for different no. of cluster (k) i.e. k=2,k=3,k=4,k=5.
2- Assign initial centroid with random points.
3- For each cluster no. (k) , there are 5 different runs with different initial centroids for each algorithm .
4- The initial centroid that was applied in traditional algorithm would be used in proposed algorithm with different runs for different (R) parameter values.

B- Results and Discussion

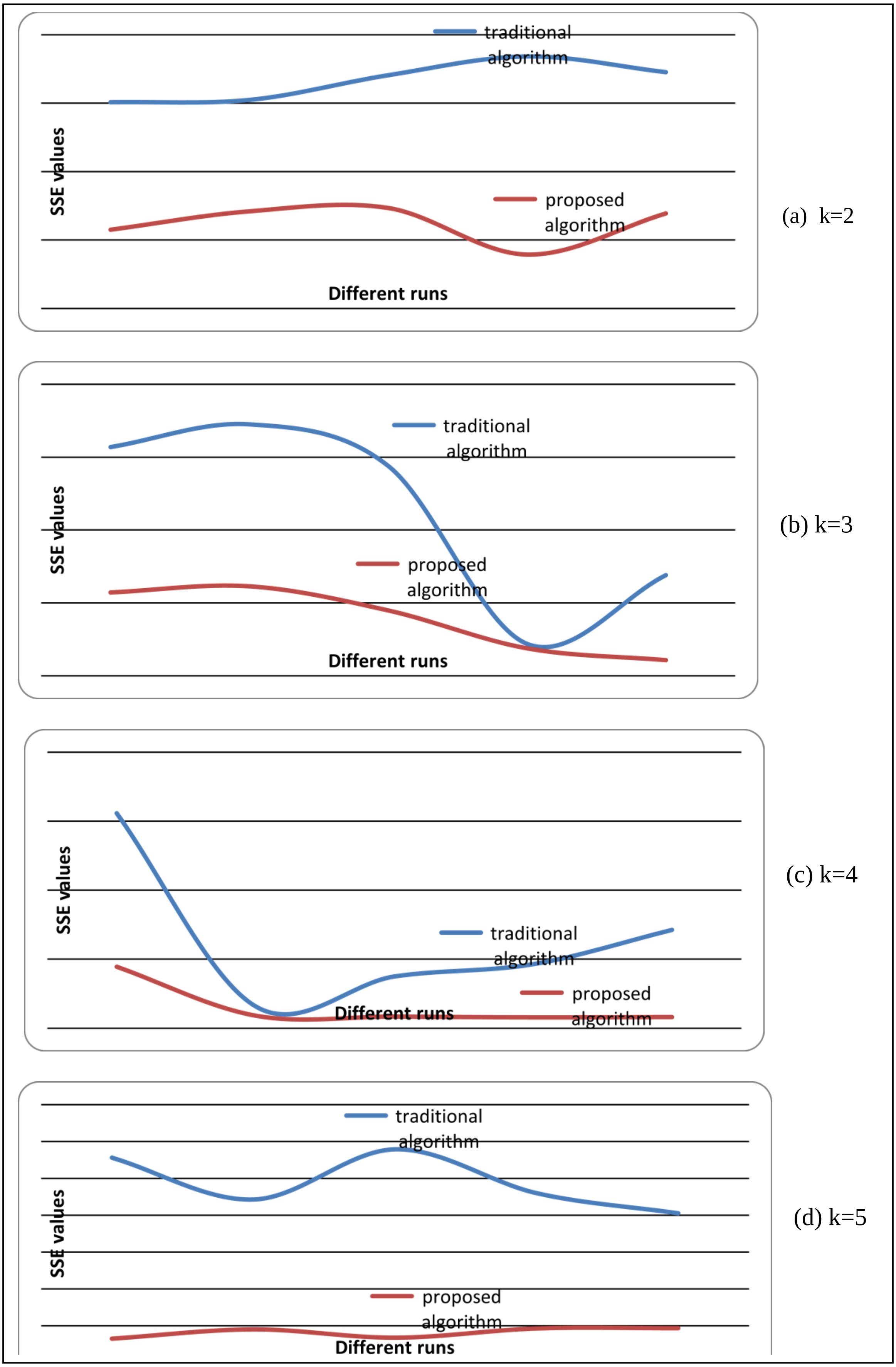
The results were produced from the experiment are in two sides, SSE as shown in Table(1) and execution time in Table(2). The values of SSE/execution time were computed for traditional k-means algorithm and proposed algorithm with different r range(0.1,…….1) . Each cell in these tables represent SSE/execution time for five runs with five different random initial centroids according to the cluster no. that use. In order to compare between the two algorithms, the initial centroids data itself were used in different algorithm with different clusters.
The results in Table(1) show the superior of the proposed algorithm for different parameters in contrast to traditional algorithm and this more clear in Figure(1) case(a,b,c,d) , where the two lines represent the two algorithms ; traditional algorithm and proposed algorithm with r=0.1. Other issue is concerned with the parameter r, the result reveal the better to use the less value for r parameter , however the good results.
In other side, the execution time is another measure that was used in comparing the two algorithms. The execution time results are shown in Table(2) that organized in the same form of table(1). The result consolidate the superior of the proposed algorithm. The execution time of the proposed algorithm in general about half of traditional algorithm execution time. Figure(2) in for case(a,b,c,d) shows graphically this comparing for traditional algorithm and proposed algorithm with parameter (r=0,1) .

Table(1) SSE values computed for first iteration of k-means and proposed algorithms

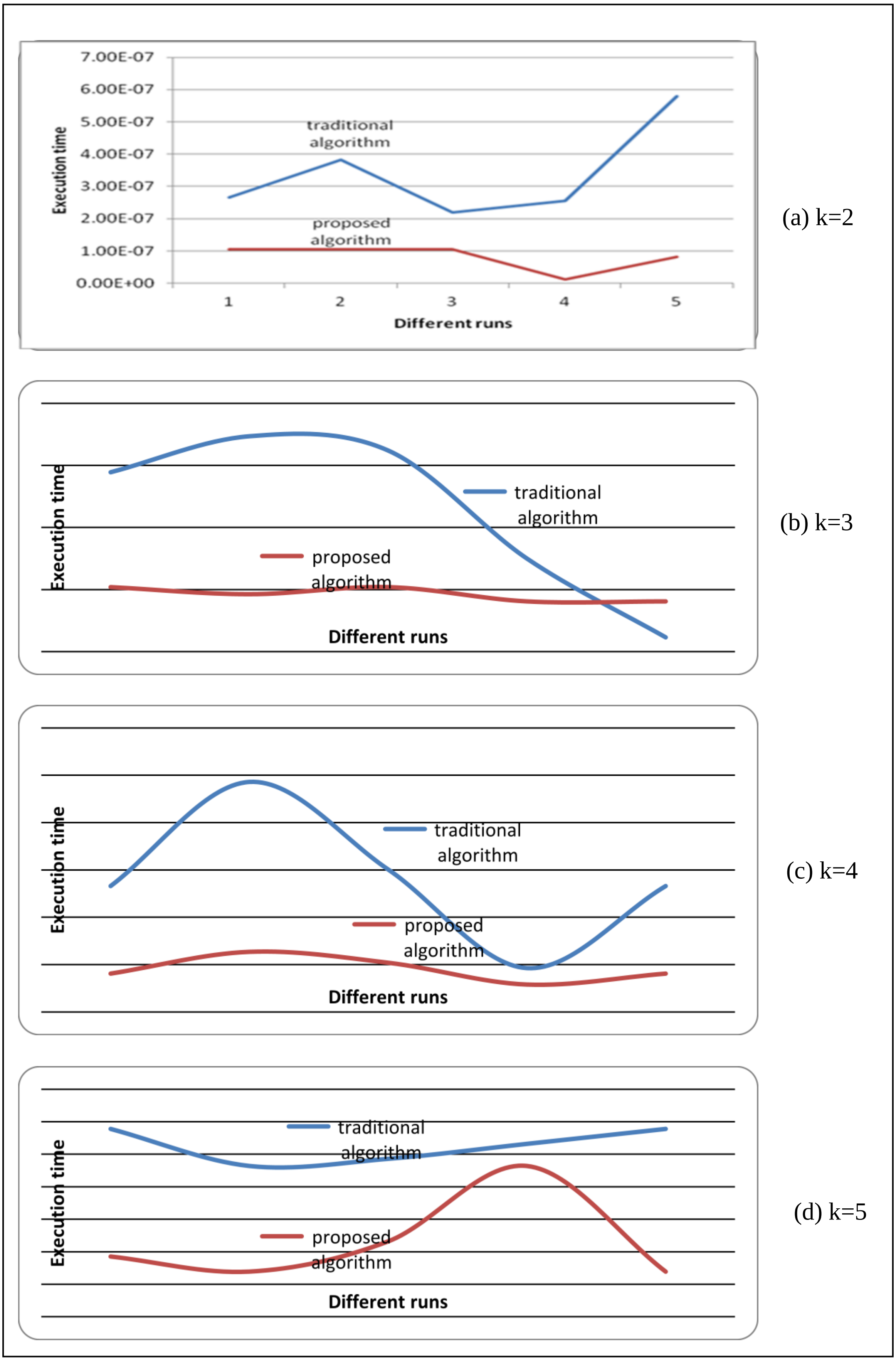
type of algorithm / no. of clusters	Traditional Algorithm	Proposed Algorithm									
		Parameter R									
		0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1.0
2 clusters	1507.08	575.2948	661.0528	653.9455	637.9423	626.3655	671.3656	662.4374	571.5157	562.8757	1125.75
	1524	709.8601	657.5448	653.9457	637.9423	626.3655	747.2174	662.4374	571.5157	562.8757	1125.75
	1704.41	734.3755	700.7154	712.3911	694.9327	664.0392	689.7483	680.5283	589.307	580.3306	1130.24
	1841.82	393.9104	505.7916	616.5472	619.2398	626.3655	652.7105	644.0859	553.4728	545.1761	1121.28
	1726.85	694.5393	680.9941	693.3376	676.3714	664.0392	689.7483	680.5283	589.307	580.3306	1130.24
3 clusters	1569.08	572.3233	601.9964	307.5448	566.0973	555.9183	606.5098	633.6863	644.4642	449.269	874.24
	1724.98	614.4341	566.4262	564.2385	570.842	675.7823	665.4427	692.059	701.6617	382.4497	823.88
	1439.81	449.0166	372.3693	287.6204	566.0871	615.511	606.5099	709.7597	718.9786	448.9048	870.59
	219.42	189.1254	183.2281	177.8091	174.5597	173.1504	173.7316	173.6116	173.8886	172.9411	644.37
	690.64	108.0001	136.2735	214.5974	209.4944	206.5426	199.7937	171.4942	185.3186	186.6066	191.12
4 clusters	1557.9	445.4929	504.5935	619.3812	604.0094	654.1584	682.8588	708.6897	697.9743	519.5312	506.46
	158.76	90.99603	115.7775	186.4591	181.5378	200.0613	194.4864	166.4821	181.2554	182.5566	187.08
	374.67	84.29855	96.60773	101.5839	96.86556	106.5662	109.676	181.4474	201.3745	171.9582	210.56
	463.94	78.68771	84.90167	107.999	112.4287	116.2303	138.6478	205.3383	213.0256	163.533	223.31
	712.64	81.02135	82.91577	89.36792	116.3125	102.2829	111.7945	189.0604	193.502	168.8694	210.56
5 clusters	556.04	64.84183	78.69251	70.74974	87.32712	95.89416	104.6176	123.9723	210.8832	212.2356	249.96
	442.47	90.10312	103.0188	150.383	148.8258	178.8137	184.2936	216.8326	200.2513	251.2702	276.25
	578.71	67.54941	94.88505	104.1979	111.0344	120.4475	130.1616	132.9083	202.1514	212.981	250.67
	459.7	92.66375	97.3787	111.004	162.137	176.1756	181.7391	195.3779	196.8997	241.7539	246.8
	405.18	93.26755	89.39872	115.9468	170.4585	174.1894	183.3982	210.5195	192.4846	241.7539	246.8

Table(2) execution time computed for first iteration of k-means and proposed algorithms

type of algorithm / no. of clusters	Traditional Algorithm	Proposed Algorithm									
		Parameter R									
		0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1.0
2 clusters	2.66E-07	1.04E-07	1.16E-08	1.16E-08	1.16E-08	1.16E-08	1.15E-08	2.32E-08	1.16E-08	9.15E-09	9.15E-09
	3.82E-07	1.04E-07	2.31E-08	1.15E-08	1.16E-08	1.15E-08	1.15E-08	1.15E-08	1.15E-08	1.15E-08	1.15E-08
	2.20E-07	1.04E-07	1.15E-08	1.16E-08	1.15E-08	1.16E-08	1.15E-08	1.16E-08	1.15E-08	1.15E-08	1.16E-08
	2.55E-07	1.16E-08	1.15E-08	8.35E-09	1.15E-08	8.35E-09	8.35E-09	8.35E-09	8.35E-09	1.15E-08	9.35E-09
	5.79E-07	8.10E-08	1.15E-08	1.16E-08	9.15E-09	1.16E-08	1.15E-08	2.32E-08	1.16E-08	1.15E-08	1.16E-08
3 clusters	2.89E-07	1.04E-07	1.15E-08	1.16E-08	1.15E-08	1.15E-08	1.15E-08	1.15E-08	1.15E-08	1.15E-08	1.15E-08
	3.47E-07	9.26E-08	2.31E-08	1.16E-08	1.15E-08	1.15E-08	1.16E-08	1.16E-08	1.15E-08	1.16E-08	1.15E-08
	3.24E-07	1.04E-07	1.16E-08	1.15E-08	1.16E-08	1.15E-08	1.15E-08	1.15E-08	1.15E-08	2.32E-08	1.15E-08
	1.50E-07	8.10E-08	1.15E-08	1.16E-08	1.15E-08	1.15E-08	1.16E-08	8.35E-09	1.16E-08	1.16E-08	1.04E-07
	2.31E-08	8.10E-08	2.32E-08	1.16E-08	1.16E-08	1.15E-08	8.35E-09	1.16E-08	1.15E-08	8.35E-09	1.15E-08
4 clusters	2.66E-07	8.10E-08	1.16E-08	1.16E-08	1.16E-08	1.16E-08	1.16E-08	1.15E-08	1.15E-08	1.16E-08	3.47E-08
	4.86E-07	1.27E-07	1.16E-08	1.15E-08	1.15E-08	1.15E-08	1.15E-08	1.15E-08	1.16E-08	1.16E-08	1.16E-08
	3.01E-07	1.04E-07	2.32E-08	2.31E-08	1.16E-08	1.15E-08	1.15E-08	1.16E-08	2.32E-08	1.16E-08	2.32E-08
	9.26E-08	5.79E-08	9.35E-09	1.15E-08	1.15E-08	8.35E-09	1.15E-08	8.35E-09	1.16E-08	1.15E-08	8.35E-09
	2.66E-07	8.10E-08	1.15E-08	1.15E-08	1.15E-08	1.15E-08	1.15E-08	1.16E-08	1.15E-08	1.15E-08	2.32E-08
5 clusters	2.89E-07	9.27E-08	1.15E-08	1.15E-08	1.16E-08	1.15E-08	1.15E-08	1.15E-08	1.15E-08	1.16E-08	1.15E-08
	2.32E-07	6.95E-08	1.15E-08	1.16E-08	1.15E-08	1.16E-08	1.16E-08	1.16E-08	1.15E-08	1.15E-08	1.15E-08
	2.43E-07	1.16E-07	1.15E-08	1.16E-08	1.16E-08	1.15E-08	1.15E-08	2.32E-08	1.15E-08	1.15E-08	1.16E-08
	2.66E-07	2.32E-07	1.15E-08	1.16E-08	1.15E-08	1.16E-08	1.16E-08	2.32E-08	2.32E-08	2.31E-08	2.31E-08
	2.89E-07	6.94E-08	1.15E-08	4.63E-08	1.16E-08	1.15E-08	2.32E-08	1.15E-08	1.15E-08	1.16E-08	2.32E-08



Figure(1) SSE values produced for (4 cases) , each one represents different no.of clusters k for traditional and proposed algorithm with r=0.1



Figure(2) execution time for (4 cases) , each one represents different no. of clusters k for traditional and proposed algorithm with r=0.1

VI- conclusion

The comparison of the two algorithms assessed that the proposed algorithm results are more prompt to use it. However achieving good results through two measurements; SSE and execution time can be exploited in developing the traditional k-means algorithm in various cases such as:

1- Generating the clusters with different trails using the proposed algorithm, then choose the best ones
2- Using the these initial centroids in traditional algorithm in order to access the better final step with less time.

Reference

[1] M.P.S Bhatia and Deepika Khurana, 2013, " Experimental study of Data clustering using k- Means and modified algorithms "International Journal of Data Mining & Knowledge Management Process (IJDKP) Vol.3, No.3, May 2013, p.p. 17-30.

[2] A.K. Jain, 2010, "Data clustering: 50 years beyond K-means", Pattern Recognition Letters 31 (2010) 651–666.

[3] T. Pang-Ning **,** S. Michael**,** and K**.** Vipin, 2006, **"** Introduction to Data Mining", ©2006 • Addison-Wesley •, p.p487-568.

[4] B. Anand M. and N. Prakash S., 2011, "Selection of Initial Centroids for k-Means Algorithm". IJCSMC, Vol. 2, Issue. 7, July 2013, pg.161 – 164.

[5] S. Raied and others, 2011, "Fast K-Means Algorithm Clustering", International Journal of Computer Networks & Communications (IJCNC) Vol.3, No.4, July 2011.

[6] V. Singh, R. Bhatia, and M. P S, 2011, "Data clustering with modified K-means algorithm," Recent Trends in Information Technology (ICRTIT), 2011 International Conference on , vol., no., pp.717,721, 3-5 June 2011.

[7] K. A. Abdul Nazeer, and M. P. Sebastian, 2009, "Improving the Accuracy and Efficiency of the k-means Clustering Algorithm", Proceedings of the World Congress on Engineering 2009 Vol I ,WCE 2009, July 1 - 3, 2009, London, U.K.

[8] C. Xuhui and X Yong, 2009, "K-Means Clustering Algorithm with Refined Initial Center," Biomedical Engineering and Informatics, 2009. BMEI '09. 2nd International Conference on , vol., no., pp.1,4, 17-19 Oct. 2009.

[9] J. Anil K. and D. Richard C., 1988, " Algorithms for Clustering Data", Prentice Hall, Englewood Cliffs, New Jersey 07632.

[10] H. Jiawei and K. Micheline, 2006, " Data Mining: Concepts and Techniques", Second Edition, Morgan Kaufmann Publishers, p.p. 383-389.

[11] M. R. Sindhu, S. Rachna, 2013, "Clustering Algorithm Technique", International Journal of Enhanced Research in Management and Computing Applications,vol. 2, Issue 3, March 2013.

[12] I. R. Swapnil, S. Amit M., 2013, "Clustering Techniques", International Journal of Advanced Research in Computer Science, Volume 4, No. 6, May 2013 (Special Issue), p.p.5-9.

[13] H. Maria, B. Yannis, and V. Michalis, 2001, " On Clustering Validation Techniques", Journal of Intelligent Information Systems, 17:2/3, 107–145, 2001, Kluwer Academic Publishers. Manufactured in The Netherlands