An Enhanced k-means algorithm to improve the Efficiency Using Normal Distribution Data Points

D.Napoleon

Assistant Professor
Department of Computer Science
School of Computer Science and Engineering
Bharathiar University
Coimbatore, Tamil Nadu, India

P.Ganga Lakshmi

Research scholar
Department of Computer Science
School of Computer Science and Engineering
Bharathiar University
Coimbatore, Tamil Nadu, India

***Abstract*— Clustering is one of the unsupervised learning method in which a set of essentials is separated into uniform groups. The k-means method is one of the most widely used clustering techniques for various applications. This paper proposes a method for making the K-means algorithm more effective and efficient; so as to get better clustering with reduced complexity. In this research, the most representative algorithms K-Means and the Enhanced K-means were examined and analyzed based on their basic approach. The best algorithm was found out based on their performance using Normal Distribution data points. The accuracy of the algorithm was investigated during different execution of the program on the input data points. The elapsed time taken by proposed enhanced k-means is less than k-means algorithm.**

Keywords- Data clustering, k-means, Enhanced k-means, cluster analysis

I. INTRODUCTION

Data mining involves the use of sophisticated data analysis tools to discover previously unknown, valid pattern and relationships in large datasets. Clustering is the process of partitioning or combination a given set of patterns into displaces clusters. A loose definition of clustering could be "the process of organizing objects into groups whose members are similar in some way". A cluster is therefore a collection of objects which are "similar" between them and are "dissimilar" to the objects belonging to other clusters[2,20] .Unlike classification, in which objects are assigned to predefined classes, clustering does not have any predefined classes[3]. The main advantage of clustering is that interesting patterns and structures can be found directly from very large data sets with little or none of the background knowledge [8, 16].

The k-means algorithm [6, 7] is successful 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. Moreover, this algorithm results in different types of clusters depending on the random choice of initial centroids. [10, 11]. Several attempts were made by researchers for improving the performance of the k-means clustering algorithm.

A. PROBLEM DESIGN

To compare the two algorithms using normal distribution data points. This investigate can be used two unsupervised clustering methods, namely K-Means, enhanced k-means are examined to analyze based on the distance between the input data points. The clusters are formed according to the distance between data points and cluster centers are formed for each cluster. The implementation plan will be in normal distribution of input data points. The implementation work was used in mat lab programming software. The number of clusters is chosen by the user [9, 22]. The data points in each cluster are displayed by different colors and the execution time is calculated in milliseconds. This paper deals with a method for improving efficiency of the k-means algorithm and analyze the elapsed time is taken by enhanced k-means is less than k means algorithm.

II. THE K-MEANS CLUSTERING ALGORITHM

The segment describes the original k-means clustering algorithm. The idea is to classify a given set of data into k number of transfer clusters, where the value of k is fixed in advance. The algorithm consists of two separate phases: the first stage is to define k centroids, one for each cluster [4, 17]. The next stage is to take each point belonging to the given data set and associate it to the nearest centroid.

Algorithm 1:

The 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:

1. arbitrarily choose k data-items from D as initial centroids;

2. Repeat Assign each item di to the cluster which has the closest centroid; Calculate new mean for each cluster; until convergence criteria is met.

The process, which is called "k-means", appears to give partitions which are reasonably efficient in the sense of within-class variance, corroborated to some extend by mathematical analysis and practical experience [18, 24]. Also, the k-means procedure is easily programmed and is computationally economical, so that it is feasible to process very large samples on a digital computer [13].

K-means algorithm is one of first which a data analyst will use to investigate a new data set because it is algorithmically simple, relatively robust and gives "good enough" answers over a wide variety of data sets [14, 19].

III. MODIFIED APPROACH

The k-means algorithm can be slightly modified to proposed method. In this algorithm can be used more effective than normal k-means algorithm [15].

The enhanced method is outlined as Algorithm 2.

Euclidean distance is generally considered to determine the distance between data points and the centroids. When all the points are included in some clusters, the first step is completed and an early grouping is done. At this point we need to recalculate the new centroids, as the inclusion of new points may lead to a change in the cluster centroids [1, 21]. Once we find *k* new centroids, a new binding is to be created between the same data points and the nearest new centroid, generating a loop. As a result of this loop, the *k* centroids may change their position in a step by step manner. Eventually, a situation will be reached where the centroids do not move anymore. This signifies the convergence criterion for clustering. Pseudo code for the k-means clustering algorithm is listed as Algorithm 1 [7].

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.

In the first phase, the initial centroids are determined systematically so as to produce clusters with better accuracy [5,12]. The second phase makes use of a variant of the clustering method discussed in [4]. It starts by forming the initial clusters based on the relative distance of each data-point from the initial centroids. These clusters are subsequently fine-tuned by using a heuristic approach, thereby improving the efficiency. The two phases of the enhanced method are described below in Algorithm 3 and 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:

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 Am (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 Am, Add it to Am and delete it from D;

5. Repeat step 4 until the number of data points in Am reaches 0.75*(n/k);

6. If m<k, then m = m+1, find another pair of datapoints 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 Am, these means will be the initial centroids [23].

Algorithm 3 describes the method for finding initial centroids of the clusters [12]. Initially, compute the distances between each data point and all other data points in the set of data points. 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. At that point go back to the second

step and 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. The distance between one vector X = (x1, x2, ...xn) and another vector Y = (y1, y2, …….yn) is obtained as (,) (1 1)2 (2 2)2 .... ( )2 *d X Y = x -y + x -y + + xn – yn* The distance between a data point X and a data-point set D is defined as d(X, D) = min (d (X, Y), where Y ∈D). The initial centroids of the clusters are given as input to the second stage, for assigning data-points to appropriate clusters. The steps involved in this phase are outlined as Algorithm 4[9].

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: 1. Compute the distance of each data-point *di* (1<=i<=n) to all the centroids *cj* (1<=j<=k) as *d(di, cj)*;

2. For each data-point *di*, find the closest centroid *cj* and assign *di* to cluster *j*.

3. Set Cluster Id[i]=j; // j:Id of the closest cluster

4. Set Nearest _Dist[i]= *d(di, cj)*;

5. For each cluster *j* (1<=j<=k), recalculate the centroids;

6. Repeat

7. for each data-point *di*,

a. Compute its distance from the centroid of the present 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(di, cj)*;

End for;

8. Assign the data-point *di* to the cluster with the nearest centroid *cj*

9. Set ClusterId[i]=j;

10. Set Nearest_Dist[i] = *d(di, cj)*;

Endfor (step(2));

11. For each cluster *j* (1<=j<=k), Recalculate the centroids until the convergence criteria is met.

IV. RESULTS

In this revise, the k-Means algorithm is explained with a paradigm first, followed by enhanced k-means algorithm. The enhanced k-means algorithm can be used determine the cluster centroids. The investigational results are discussed for the K-Means algorithm has to take the time complexity is greater for using different data sets.

The resulting clusters of the normal distribution of K-Means algorithm is presented in Fig. 1. The normal distribution data points can be taken to easily implement and take the results of convenient for our data sets. The number of clusters and data points is given by the user during the execution of the program. The number of data points is 1000 and the number of clusters given by the user is 10 (k = 10). The algorithm is repeated to allocate the user times to get efficient output. The cluster centers (centroids) are calculated for each cluster by its mean value and clusters are formed depending upon the distance between data points. For different input data points, the algorithm gives different types of outputs.

The enhanced k-means is better than k-means in experimental results. In cluster size has to be differing in different run. The k-means algorithm by taken elapsed time is 2343.3ms. And then the first cluster size is 84 in run1. It can be measures by the quality of clusters. The efficient method by taken elapsed time is 62.2ms.then the first cluster size in run1 is 99 based their quality of cluster size. The execution method can be executing five runs.The average time can be taken by 2116.26 in k means algorithm. The enhanced k-means method can be taken the average time is 43.194. The enhanced average time is less than k-means.

Table (1): Cluster results for Normal Distribution

Number of Data Points = 1000										**Number of Clusters=10**		
Cluster Size		**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**Time (ms)**
k-Means	**Run1**	**84**	**78**	**89**	**95**	**110**	**103**	**96**	**121**	**118**	**106**	**2343.3**
	Run2	**99**	**135**	**91**	**91**	**96**	**87**	**78**	**123**	**105**	**95**	**2145.5**
	Run3	**102**	**95**	**81**	**75**	**86**	**117**	**76**	**102**	**135**	**131**	**1528.5**
	Run4	**93**	**118**	**101**	**119**	**99**	**90**	**94**	**61**	**99**	**126**	**2166.0**
	Run5	**136**	**99**	**102**	**132**	**96**	**88**	**84**	**86**	**90**	**87**	**2397.8**
Enhanced k-means	**Run1**	**42**	**77**	**65**	**148**	**125**	**104**	**67**	**95**	**111**	**166**	**62.2**
	Run2	**102**	**112**	**108**	**74**	**103**	**121**	**102**	**97**	**87**	**94**	**45.97**
	Run3	**120**	**91**	**106**	**101**	**122**	**97**	**88**	**87**	**111**	**77**	**44.76**
	Run4	**92**	**133**	**71**	**95**	**95**	**101**	**74**	**101**	**121**	**117**	**34.45**
	Run5	**123**	**86**	**93**	**64**	**96**	**104**	**133**	**84**	**121**	**96**	**28.59**

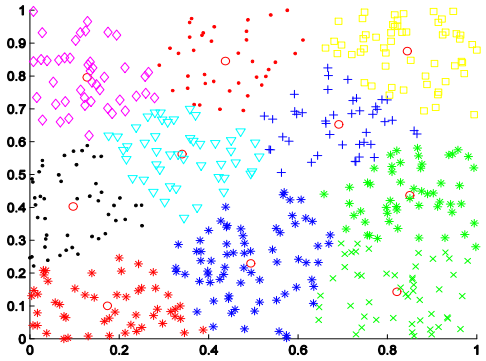

Fig (1). Normal Distribution output in k-means

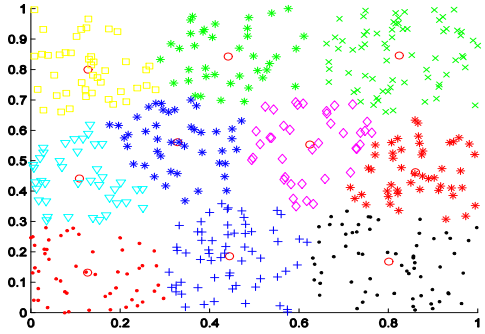

Fig (2) Normal Distribution output in Enhanced k-means

I. CONCLUSION

The time complexity can be calculated by CPU elapsed time for different two algorithms. As a rule the time complexity varies from one processor to another processor, which depends on the speed and the type of the system. The partitioning algorithms work well for decision spherical-shaped clusters in different type of data points. The advantage of the K-Means algorithm is its favorable execution time. Its drawback is that the user has to know in advance how many clusters are searched for. From the experimental results (by many execution of the programs), it is practical that K-Means algorithm is efficient for smaller data sets and enhanced k-means algorithm seems to be efficient for huge data sets than K-means algorithm.

REFERENCES

[1] Berkhin, P., 2002. "*Survey of clustering data mining techniques.*" Technical Report, Accrue Software,Inc.

[2] Chaturvedi J. C. A, Green P, "K-modes clustering," *J. Classification*,(18):35–55, 2001.

[3] Daxin Jiang, Chum Tong and Aidong Zhang, "Cluster Analysis for Gene Expression Data," *IEEE Transactions on Data and Knowledge Engineering*, 16(11): 1370-1386, 2004.

[4] Fahim A.M, Salem A. M, Torkey A and Ramadan M. A, "An Efficient enhanced k-means clustering algorithm," *Journal of Zhejiang University*, 10(7):1626–1633, 2006.

[5] Huang Z, "Extensions to the k-means algorithm for clustering large data sets with categorical values," *Data Mining and Knowledge Discovery*, (2):283–304, 1998.

[6] T. Velmurugan, T. Santhanam "*Computational complexity between k-means and k-medoid clustering algorithm for normal and uniform distribution of data points .*" journal of computer science 6(3):363-368,2010,ISSN 1549-3636.

[7] Margaret H. Dunham, *Data Mining- Introductory and Advanced Concepts*, Pearson Education, 2006.

[8] McQueen J, "*Some methods for classification and analysis of multivariate observations," Proc. 5th Berkeley Symp. Math. Statist.Prob.*, (1):281–297, 1967.

[9] K. A. Abdul Nazeer, M. P. Sebastian " *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.

[10] Pang-Ning Tan, Michael Steinback and Vipin Kumar, *Introduction to Data Mining*, Pearson Education, 2007.

[11] Stuart P. Lloyd, "Least squares quantization in pcm," *IEEE Transactions on Information Theory*, 28(2): 129-136.

[12] Yuan F, Meng Z. H, Zhang H. X and Dong C. R, "A New Algorithm to Get the Initial Centroids," *Proc. of the 3rd International Conference on Machine Learning and Cybernetics*, pages 26–29, August 2004.

[13]. MacQueen, J.: "*Some Methods for Classification and Analysis of Multivariate Observations.*" In Proceedings Fifth Berkeley Symposium Mathematics Statistics and Probability. Vol. 1. Berkeley, CA (1967) 281-297.

[14]. Wesan, Barbakh And Colin Fyfe. "*Local vs global interactions in clustering algorithms: Advances over K-means.*" International Journal of knowledge-based and Intelilligent Engineering Systems 12 (2008).83 – 99.

[15]. Zalik, Krista Rizman . "*An Efficient k`-means Clustering Algorithm.*" Pattern Reconition Letters, Vol. 29, I.9. Pag. 1385-1391. Elsevier 07/2008.

[16]. Borah, S. and M.K. Ghose, 2009"*Performance analysis of AIM-K-Means and K-Means in quality cluster generation.* J. Comput., 1: 175-178.

[17**]."***Data Mining: Introductory and Advanced Topics*." 1st Edn., Prentice Hall, USA., ISBN: 10: 0130888923, pp: 315.

[18].Han, J. and M. Kamber, 2006. "*Data Mining: Concepts and Techniques.*" Morgan Kaufmann Publishers,2nd Edn., New Delhi, ISBN: 978-81-312-0535-8.

[19].Jain, A.K. and R.C. Dubes, 1988. "*Algorithms for Clustering Data*". Prentice Hall Inc., Englewood Cliffs, New Jersey, ISBN: 0-13-022278-X, pp: 320.

[20].Jain, A.K., M.N. Murty and P.J. Flynn, 1999. "*Data clustering: A review*." ACM Comput. Surveys, 31: 264-323. DOI: 10.1145/331499.331504

[21].Khan, S.S. and A. Ahmad, 2004. "*Cluster center initialization algorithm for K-Means clustering*". Patt. Recog. Lett., 25: 1293-1302. DOI:10.1016/j.patrec.2004.04.007

[22].Park, H.S., J.S. Lee and C.H. Jun, 2006. *A K-means like algorithm for K-medoids clustering and its performance.*

[23]Rakhlin, A. and A. Caponnetto, 2007. "*Stability of k-Means clustering*". Adv. Neural Inform. Process.
Syst., 12: 216-222.

[24].Xiong, H., J. Wu and J. Chen, 2009. "*K-Means clustering versus validation measures: A data distribution perspective.*" IEEE Trans. Syst., Man, Cybernet. Part B, 39: 318-331.

Author Profile

D. Napoleon received the Bachelor's Degree in B.Sc Physics from Madurai Kamaraj University in 1999, Master's Degree in Computer Applications from Madurai Kamaraj University in 2002, and M.Phil degree in Computer Science from Periyar University in 2007.He is working as Assistant Professor in the Department of Computer Science ,School of Computer Science Engineering, Bharathiar University, Coimbatore, Tamilnadu, India. He has published articles in National and International Journals. He has presented papers in National and International Conferences. His research area is Data Mining.

P.Ganga Lakshmi received the Bachelor's Degreee in B.Sc Electronics & Communication in Sourashtra College, Madurai Kamarajar University , Madurai and M.Sc Computer Science and Information Technology, in Mannar Thirumalai Naicker College , Madurai Kamarajar University , Madurai. Now she is a Research in Department of Computer Science, Bharathiar University, Coimbatore, Tamilnadu. She has presented Papers in National and international Conferences.