Information Technology Journal 10 (3): 478-484, 2011
ISSN 1812-5638 / DOI: 10.3923/itj.2011.478.484

A Survey of Partition based Clustering Algorithms in Data Mining: An Experimental Approach

T. Velmurugan and T. Santhanam
PG and Research Department of Computer Science, D.G. Vaishnav College, Chennai-600106, India

Abstract: Clustering is one of the most important research areas in the field of data mining. Clustering means creating groups of objects based on their features in such a way that the objects belonging to the same groups are similar and those belonging in different groups are dissimilar. Clustering is an unsupervised learning technique. Data clustering is the subject of active research in several fields such as statistics, pattern recognition and machine learning. From a practical perspective clustering plays an outstanding role in data mining applications in many domains. 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. Clustering algorithms can be applied in many areas, for instance marketing, biology, libraries, insurance, city-planning, earthquake studies and www document classification. Data mining adds to clustering the complications of very large datasets with very many attributes of different types. This imposes unique computational requirements on relevant clustering algorithms. A variety of algorithms have recently emerged that meet these requirements and were successfully applied to real-life data mining problems. They are subject of this survey. Also, this survey explores the behavior of some of the partition based clustering algorithms and their basic approaches with experimental results.

Key words: Clustering algorithms, K-means clustering, K-medoids clustering, fuzzy C-means clustering

INTRODUCTION

The goal of this survey is to provide a comprehensive review of different partition based clustering algorithms in data mining. Clustering is a division of data into groups of similar objects. Each group, called cluster, consists of objects that are similar between themselves and dissimilar to objects of other groups. Representing data by fewer clusters necessarily loses certain fine details (akin to lossy data compression), but achieves simplification. It represents many data objects by few clusters and hence, it models data by its clusters. Data modeling puts clustering in a historical perspective rooted in mathematics, statistics and numerical analysis. From a machine learning perspective clusters correspond to hidden patterns, the search for clusters is unsupervised learning and the resulting system represents a data concept. Clustering is a method of unsupervised learning and a common technique for statistical data analysis used in many fields, including machine learning, data mining, pattern recognition, image analysis and bioinformatics. Besides the term clustering, there are a number of terms with similar meanings, including automatic classification, numerical taxonomy, botryology and typological analysis. Therefore, clustering is unsupervised learning of a hidden data concept (Berkhin, 2002; Dunham, 2003; Han and Kamber, 2006; Jain *et al*., 1999).

Data mining deals with large databases that impose on cluster analysis. Some of the challenges led to the emergence of powerful broadly applicable data mining clustering methods surveyed below. A variety of clustering algorithms have been used for research in the field of data mining (Berkhin, 2002; Dunham, 2003; Han and Kamber, 2006; Xiong *et al*., 2006; Park *et al*., 2009; Khan and Ahmad, 2004; Alexander and Caponnetto, 2007). They are organized into the following categories: partitioning methods, hierarchical methods, density-based methods, grid-based methods, model-based methods, methods for high dimensional data and constraint-based clustering. Some of the above methods use the distance measure for finding the clusters. The scope of this survey is modest: to provide an introduction to cluster analysis in the field of data mining, where, it is to define data mining to be the discovery of useful, but non-obvious, information or patterns in large collections of data. Much of this survey is necessarily consumed by providing a general background for cluster analysis, but also which

Corresponding Author: T. Velmurugan, PG and Research Department of Computer Science, D.G. Vaishnav College, Chennai-600106, India

discuss a number of partitions based clustering techniques that have recently been developed specifically for data mining. While, the survey strives to be self-contained from a conceptual point of view, many details have been omitted.

PARTITIONING METHODS

The term cluster does not have a precise definition. However, several working definitions of a cluster are commonly used. A cluster is a set of points such that any point in a cluster is closer (or more similar) to every other point in the cluster than to any point not in the cluster. Sometimes a threshold is used to specify that all the points in a cluster must be sufficiently close (or similar) to one another (Jain and Dubes, 1988). However, in many sets of data, a point on the edge of a cluster may be closer (or more similar) to some objects in another cluster than to objects in its own cluster. Consequently, 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 the cluster, or a medoid, the most representative point of a cluster.

A partitioning method first creates an initial set of k partitions, where, parameter k is the number of partitions to construct. It then uses an iterative relocation technique that attempts to improve the partitioning by moving objects from one group to another. These clustering techniques create a one-level partitioning of the data points. There are a number of such techniques, but this survey shall only describe three approaches namely K-means, K-medoids and fuzzy C-means. Except these three techniques, to deal with large data sets, a sampling-based method CLARA (Clustering LARge Applications) is used. To improve the quality and scalability of CLARA, a K-medoid type algorithm called CLARANS (Clustering Large Applications based upon RANdomized Search) was proposed, which combines the sampling technique with PAM (Partitioning Around Medoids) (Jain *et al*., 1999; Han and Kamber, 2006). All these techniques are based on the idea that a center point can represent a cluster. For K-means, the notion of a centroid is used, which is the mean or median point of a group of points. A centroid almost never corresponds to an actual data point. For K-medoid, the notion of a medoid is used, which is the most representative (central) point of a group of points. K-means is a simple algorithm that has been adapted to many problem domains. It can see that the K-means algorithm is a good candidate for extension to work with fuzzy feature vectors (Berkhin, 2002; Dunham, 2003; Han and Kamber, 2006; Jain *et al*., 1999; Kaufman and Rousseeuw, 1990). Therefore the algorithm with fuzzy feature is called the Fuzzy C-Means (FCM) algorithm.

As stated, this survey discusses only the partition based algorithms in data mining. Therefore, the algorithms K-means, K-medoids and fuzzy C-means are examined one by one to analyze based on the distance between the various input data points. The clusters are formed according to the distance between data points and cluster centers are formed for each cluster. The number of clusters (by giving the K-value) is specified by the user. The data points in each cluster are displayed by different colors (one color for one cluster) and the execution time for each cluster and the total time is calculated in milliseconds (Velmurugan and Santhanam, 2010a). This research work does not use any existing data set that they were available anywhere including the internet. The data points are created by own method by using a JAVA program for all the three types of algorithms discussed here.

K-MEANS ALGORITHM

K-Means is one of the simplest unsupervised learning algorithms that solve the well known clustering problem. The procedure follows a simple and easy way to classify a given data set through a certain number of clusters (assume k clusters) fixed a priori (Borah and Ghose, 2009; Alexander and Caponnetto, 2007). The main idea is to define k centroids, one for each cluster. These centroids should be placed in a cunning way because of different location causes different result. So, the better choice is to place them as much as possible far away from each other. The next step is to take each point belonging to a given data set and associate it to the nearest centroid. When, no point is pending, the first step is completed and an early group age is done. At this point it is necessary to re-calculate k new centroids as bar centers of the clusters resulting from the previous step. After obtaining these k new centroids, a new binding has to be done between the same data set points and the nearest new centroid. A loop has been generated. As a result of this loop, one may notice that the k centroids change their location step by step until no more changes are done (Xiong *et al*., 2006; Kanungo *et al*., 2003; Bradley and Fayyad, 1998). In other words centroids do not move any more. Finally, this algorithm aims at minimizing an objective function, in this case a squared error function. The objective function:

$$J = \sum_{j=1}^{k}\sum_{i=1}^{n}\left\|x_i^{(j)} - c_j\right\|^2$$

where, $\|x_i^{(j)}-c_j\|^2$ is a chosen distance measure between a data point $x_i^{(j)}$ and the cluster centre c_j, is an indicator of the distance of the n data points from their respective cluster centers. The algorithm is composed of the following steps:

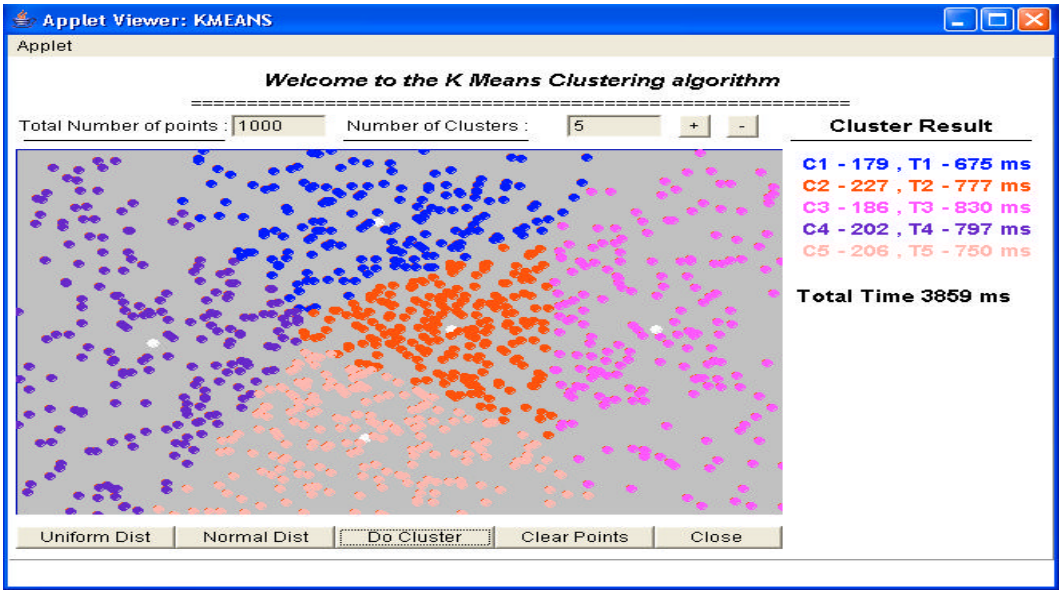

Fig. 1: K-means output

Step 1: Place K points into the space represented by the objects that are being clustered. These points represent initial group centroids

Step 2: Assign each object to the group that has the closest centroid

Step 3: When, all objects have been assigned, recalculate the positions of the K centroids

Step 4: Repeat steps 2 and 3 until the centroids no longer move

This produces a separation of the objects into groups from which the metric to be minimized can be calculated. Always the algorithm can be proved that the procedure will terminate, the K-means algorithm does not necessarily find the most optimal configuration, corresponding to the global objective function minimum. The algorithm is also significantly sensitive to the initial randomly selected cluster centers. The K-means algorithm can be run multiple times to reduce this effect (Dunham, 2003; Han and Kamber, 2006; Dhillon *et al*., 2005). K-means is a simple algorithm that has been adapted to many problem domains and it is a good candidate to work for a randomly generated data points. One of the most popular heuristics for solving the K-means problem is based on a simple iterative scheme for finding a locally minimal solution (Borah and Ghose, 2009; Khan and Ahmad, 2004; Alsabti *et al*., 1998). This algorithm is often called the K-means algorithm. There are some difficulties in using K-means for clustering data. This is proved by several times in this current as well as in the past research and an oft-recurring problem has to do with the initialization of the algorithm. K-means is a simple algorithm that has been adapted to many problem domains.

The experimental results for K-means clustering algorithm is shown in Fig. 1. The data points are created manually in the applet window by pressing the mouse buttons. Actually, three types of data points can be created in the applet window (Velmurugan and Santhanam, 2008). They are normal distribution, uniform distributions and the third one is manual creation of data points. This survey uses only the manual creation of data points. The other methods of creation of data points are used in other study (Velmurugan and Santhanam, 2010b). The resulting clusters of the distribution of data points for K-means algorithm is presented in Fig. 1. The number of clusters and the 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 5 (k = 5). The algorithm is repeated 1000 times (one iteration for each data point) 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 input data points are generated in red color and the output of the algorithm is displayed in different colors as shown in Fig. 1. The center point of each cluster is displayed in white color. The execution time of each run is calculated in milliseconds. The time taken for execution of the algorithm varies from one run to another run and

also it differs from one computer to another computer. The number of data points is the size of the cluster. If the number of data points are 1000 then the algorithm is repeated the same one thousand times. For each data point, the algorithm executes one time. From the results it is clear that the algorithm takes 3859 m sec for 1000 data points and 5 clusters. The execution time for each cluster is also calculated. The sum of the execution time of all clusters is 3829 m sec. The difference is 30 m sec. This time is the execution time for other codes in the program.

K-MEDOIDS ALGORITHM

The K-means algorithm is sensitive to outliers since an object with an extremely large value may substantially distort the distribution of data. How might the algorithm be modified to diminish such sensitivity? Instead of taking the mean value of the objects in a cluster as a reference point, a medoid can be used, which is the most centrally located object in a cluster. Thus, the partitioning method can still be performed based on the principle of minimizing the sum of the dissimilarities between each object and its corresponding reference point. This forms the basis of the K-medoids method (Han and Kamber, 2006; Jain *et al*., 1999; Kaufman and Rousseeuw, 1990). The basic strategy of K-medoids clustering algorithms is to find k clusters in n objects by first arbitrarily finding a representative object (the medoids) for each cluster. Each remaining object is clustered with the medoid to which it is the most similar. K-medoids method uses representative objects as reference points instead of taking the mean value of the objects in each cluster. The algorithm takes the input parameter k, the number of clusters to be partitioned among a set of n objects (Park *et al*., 2009; Dunham, 2003; Han and Kamber, 2006; Zeidat and Eick, 2004; Sheng and Liu, 2006).

A typical K-Mediods algorithm for partitioning based on medoid or central objects is as follows:

Input:
K: The number of clusters
D: A data set containing n objects
Output: A set of k clusters that minimizes the sum of the dissimilarities of all the objects to their nearest medoid
Method: Arbitrarily choose k objects in D as the initial representative objects
Repeat: Assign each remaining object to the cluster with the nearest medoid
randomly select a non medoid object O_{random}
compute the total points S of swaping object O_j with O_{random}
If S<0 then swap O_j with O_{random} to form the new set of k medoid
Until no change

Like this algorithm, a Partitioning Around Medoids (PAM) was one of the first K-medoids algorithms introduced. It attempts to determine k partitions for n objects. After an initial random selection of k medoids, the algorithm repeatedly tries to make a better choice of medoids. Therefore, the algorithm is often called as representative object based algorithm. Figure 2 is the output for one of the executions of K-medoids algorithm

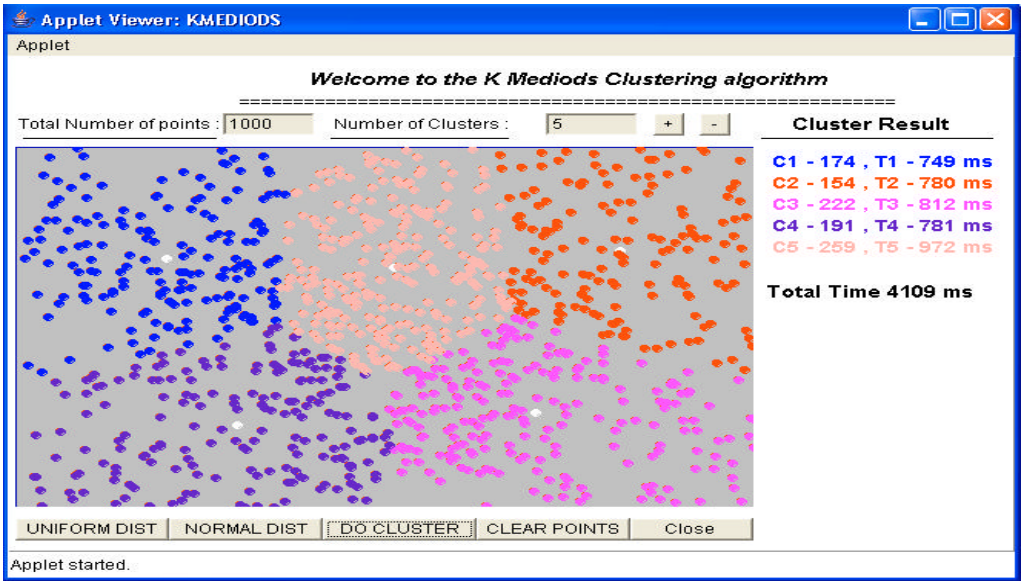

Fig. 2: K-medoids output

for the manual creation of input data points. Here, also the number of data points is 1000 and the number clusters chosen by the user is 5. By comparing the results of both the algorithms K-means and K-medoids (Velmurugan and Santhanam, 2010a), it can be easily understood that there is much difference in the execution time. The K-means algorithm takes 3859 m sec whereas the K-medoids algorithm takes 4109 m sec (Velmurugan and Santhanam, 2009a).

FUZZY C-MEANS ALGORITHM

Traditional clustering approaches generate partitions; in a partition, each pattern belongs to one and only one cluster. Hence, the clusters in a hard clustering are disjoint. Fuzzy clustering extends this notion to associate each pattern with every cluster using a membership function. The output of such algorithms is a clustering, but not a partition. Fuzzy clustering is a widely applied method for obtaining fuzzy models from data. It has been applied successfully in various fields including geographical surveying, finance or marketing. This method is frequently used in pattern recognition. It is based on minimization of the following objective function:

$$J_m = \sum_{i=1}^{N}\sum_{j=1}^{C} u_{ij}^{m} \left\| x_i - C_j \right\|^2,\ 1 \le m < \infty$$

where, m is any real number greater than 1, u_{ij} is the degree of membership of x_i in the cluster j, x_i is the ith of d-dimensional measured data, c_j is the d-dimension center of the cluster and ||*|| is any norm expressing the similarity between any measured data and the center (Al-Zoubi *et al*., 2007; Yong *et al*., 2004). Fuzzy partitioning is carried out through an iterative optimization of the objective function shown above, with the update of membership u_{ij} and the cluster centers c_j by:

$$u_{ij} = \frac{1}{\sum_{k=1}^{C}\left(\frac{\left\|x_i - C_j\right\|}{\left\|x_i - C_k\right\|}\right)},\ C_j = \frac{\sum_{i=1}^{N} u_{ij}^{m} . x_i}{\sum_{i=1}^{N} u_{ij}^{m}}$$

This iteration will stop when $\max_{ij} \{|u_{ij}^{(k+1)} - u_{ij}^{(k)}|\} < \varepsilon$, where, ε is a termination criterion between 0 and 1, whereas k are the iteration steps. This procedure converges to a local minimum or a saddle point of J_m. The algorithm is composed of the following steps:

Step 1: Initialize $U = [u_{ij}]$ matrix, $U^{(0)}$

Step 2: At k-step: calculate the centers vectors $C^{(k)} = [c_j]$ with $U^{(k)}$

Step 3: Update $U^{(k)}$, $U^{(k+1)}$

Step 4: If $\| U^{(k+1)} - U^{(k)} \| < \varepsilon$ then STOP; otherwise return to step 2

In this algorithm, data are bound to each cluster by means of a membership function, which represents the fuzzy behavior of the algorithm. To do that, the algorithm has to build an appropriate matrix named U whose factors are numbers between 0 and 1 and represent the degree of membership between data and centers of clusters. FCM clustering techniques are based on fuzzy behavior and provide a natural technique for producing a clustering where membership weights have a natural (but not probabilistic) interpretation. This algorithm is similar in structure to the K-means algorithm and also behaves in a similar way.

However, when the FCM algorithm is run on the given 1000 data points with C = 5, five clusters are identified as shown in Fig. 3. The result of the algorithm is displayed in the same figure. Here, also the number of data points is 1000 and the number of clusters chosen by the user is 5. This algorithm takes 4000 m sec to get the output. This is much better than the K-medoids algorithm, but not superior to K-means algorithm. The data points to all the three algorithms are created manually in this research in applet window, not by using any formula like box-muller formula. The normal and uniform distribution of data points are created by using the box-muller formula (Velmurugan and Santhanam, 2009b). They are not discussed in this study. Table 1 gives the comparative result of all these three algorithms. For all the three algorithms, the program is executed many times and the results are analyzed based on the number of data points and the number of clusters. The behavior of the algorithms is analyzed based on observations. The performance of the algorithms have also been analyzed for several executions by considering different data points (for which the results are not shown) as input (500, 1000 and 2000 data points etc.) and the number of clusters are from 5 to 10 (for which also the results are not shown), the outcomes are found to be highly satisfactory

Table 1: Results of algorithms

	Cluster						
Algorithm	1	2	3	4	5	Time (m sec)	Diff. time
K-means							
Size	179	227	186	202	206	3859	30
Time (m sec)	675	777	830	797	750	3829	
K-medoids							
Size	174	154	222	191	259	4109	15
Time (m sec)	749	780	812	781	972	4094	
FCM							
Size	143	192	183	232	250	4000	16
Time (m sec)	688	794	783	843	876	3984	

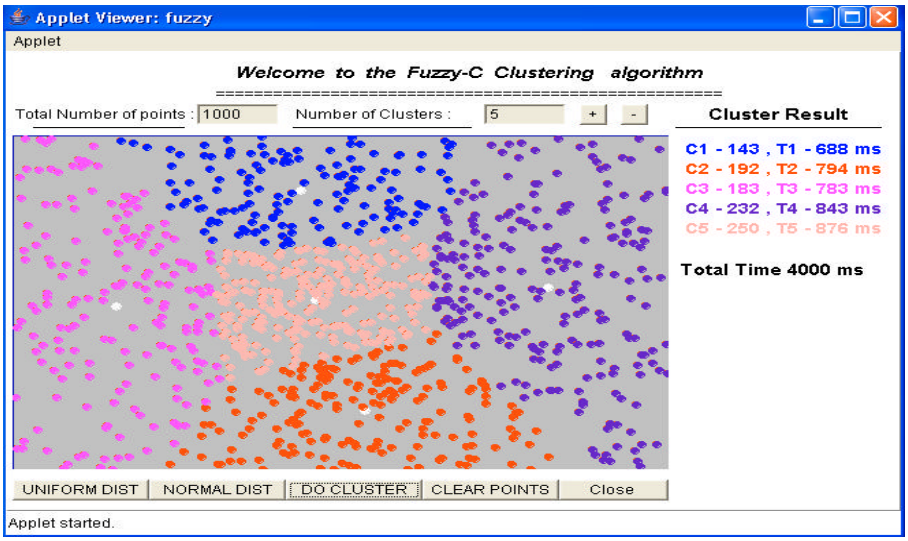

Fig. 3: Fuzzy C-means output

(Velmurugan and Santhanam, 2009b). From Table 1, it is easy to understand that K-means algorithm takes very less time and K-medoids algorithm takes more time than the K-means for clustering the data points, but the FCM algorithm act as intermediary between the previous two algorithms.

Cluster analysis is still an active field of development. Many cluster analysis techniques do not have a strong formal basis. Cluster analysis is a rather ad-hoc field (Berkhin, 2002; Dunham, 2003; Han and Kamber, 2006; Xiong *et al*., 2006; Park *et al*., 2009). There are a wide variety of clustering techniques. Comparisons among different clustering techniques are difficult. All techniques seem to impose a certain structure on the data and yet few authors describe the type of limitations being imposed. In spite of all these problems, clustering analysis is a useful (and interesting) field. In summary, clustering is an interesting, useful and challenging problem. It has great potential in applications like object recognition, image segmentation and information filtering and retrieval. However, it is possible to exploit this potential only after making several designs choices carefully. The advantage of the partition-based algorithms that they use an iterative way to create the clusters, but the drawback is that the number of clusters has to be determined in advance and only spherical shapes can be determined as clusters (Davies and Bouldin, 1979).

CONCLUSION

Usually the time complexity varies from one processor to another processor, which depends on the speed and the type of the system. The partition based algorithms work well for finding spherical-shaped clusters in small to medium-sized 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 several executions of the program for the proposed three algorithms, the following results were obtained. It is observed that K-means algorithm is efficient for smaller data sets and K-medoids algorithm seems to perform better for large data sets. The performance of FCM is intermediary between them. FCM produces close results to K-means clustering, yet it requires more computation time than K-means because of the fuzzy measures calculations involved in the algorithm. From Table 1, it is notorious that for 1000 manually created data points with five clusters, K-means algorithm is very consistent when compared with the other two algorithms. Further, it stamps its superiority in terms of its lesser execution time.

REFERENCES

Al-Zoubi, M.B., A. Hudaib and B. Al-Shboul, 2007. A fast fuzzy clustering algorithm. Proceedings of the 6th WSEAS International Conference on Artificial Intelligence, Knowledge Engineering and Data Bases, February 2007, Corfu Island, Greece, pp: 28-32.

Alexander, R. and A. Caponnetto, 2007. Stability of K-Means Clustering, Advances in Neural Information Processing Systems 12. MIT Press, Cambridge, MA, pp: 216-222.

Alsabti, K., S. Ranka and V. Singh, 1998. An efficient k-means clustering algorithm. Proc. First Workshop High Performance Data Mining. http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.110.903&rep=rep1&type=pdf.

Berkhin, P., 2002. Survey of Clustering Data Mining Techniques. Accure Software Inc., San Jose, CA., USA.

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.

Bradley, P.S. and U.M. Fayyad, 1998. Refining initial points for K-means clustering. Proceedings of the 15th International Conference on Machine Learning, July 24-27, Morgan Kaufmann, San Francisco, pp: 91-99.

Davies, D.L. and D.W. Bouldin, 1979. A cluster separation measure. IEEE. Trans. Pattern Anal. Mach. Intel., 1: 224-227.

Dhillon, I., Y. Guan and B. Kulis, 2005. A unified view of kernel k-means, spectral clustering and graph partitioning. Technical Report TR-04-25, UTCS Technical Report.

Dunham, M., 2003. Data Mining: Introductory and Advanced Topics. Prentice Hall, USA.

Han, J. and M. Kamber, 2006. Data Mining: Concepts and Techniques. 2nd Edn., Morgan Kaufmann Publisher, San Fransisco, USA., ISBN: 1-55860-901-6.

Jain, A.K. and R.C. Dubes, 1988. Algorithms for Clustering Data. Prentice Hall Inc., Englewood Cliffs, New Jerssy, ISBN: 0-13-022278-X.

Jain, A.K., M.N. Murty and P.J. Flynn, 1999. Data clustering: A review. ACM Comput. Surveys, 31: 264-323.

Kanungo, T., D.M. Mount, N.S. Netanyahu, C.D. Piatko, R. Silverman and A.Y. Wu, 2003. A local search approximation algorithm for k-means clustering. July 14, 2003. http://www.cs.umd.edu/~mount/Projects/KMeans/kmlocal-cgta.pdf.

Kaufman, L. and P.J. Rousseeuw, 1990. Finding Groups in Data: An Introduction to Cluster Analysis. John Wiley and Sons, New York.

Khan, S.S. and A. Ahmad, 2004. Cluster center initialization algorithm for K-Means clustering. Pattern Recognition Lett., 25: 1293-1302.

Park, H.S., J.S. Lee and C.H. Jun, 2009. A K-Means-Like Algorithm for K-Medoids Clustering and Its Performance. Department of Industrial and Management Engineering, POSTECH, South Korea.

Sheng, W. and X. Liu, 2006. A genetic k-medoids clustering algorithm. J. Heuristics, 12: 447-466.

Velmurugan, T. and T. Santhanam, 2008. Performance analysis of k-means and k-medoids clustering algorithms for a randomly generated data set. Proceedings of the International Conference on Systemics, Cybernetics and Informatics, Jan. 08, Hyderabad, India, pp: 578-583.

Velmurugan, T. and T. Santhanam, 2009a. Clustering of random data points using K-Means and fuzzy-c means clustering algorithms. Proceedings of the IEEE International Conference on Emerging Trends in Computing, Jan. 09, Virudhunagar, India, pp: 177-180.

Velmurugan, T. and T. Santhanam, 2009b. A practical approach of k-medoids clustering algorithm for artificial data points. Proceedings of the International Conference on Semantics, E-business and E-Commerce, Nov. 09, Trichirappalli, India, pp: 45-50.

Velmurugan, T. and T. Santhanam, 2010a. Computational complexity between K-means and K-medoids clustering algorithms for normal and uniform distributions of data points. J. Comput. Sci., 6: 363-368.

Velmurugan, T. and T. Santhanam, 2010b. Performance evaluation of k-means and fuzzy c-means clustering algorithms for statistical distributions of input data points. Eur. J. Sci. Res., Vol. 46, No. 3

Xiong, H., J. Wu and J. Chen, 2006. K-means clustering versus validation measures: A data distribution perspective. Proceedings of the 12th ACM SIGKDD International Conference Knowledge Discovery and Data Mining, (KDDM'06), USA., pp: 779-878.

Yong, Y., Z. Chongxun and L. Pan, 2004. A novel fuzzy c-means clustering algorithm for image thresholding. Measurement Sci. Rev., 4: 9-9.

Zeidat, N. and C.F. Eick, 2004. K-medoid-style clustering algorithms for supervised summary generation. Proceedings of the International Conference on Machine Learning. http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.128.9219&rep=rep1&type=pdf.