International Journal of Current Engineering and Technology
E-ISSN 2277 – 4106, P-ISSN 2347 - 5161
©2014 INPRESSCO®, All Rights Reserved
Available at http://inpressco.com/category/ijcet

Research Article

Optimizing K-Means by Fixing Initial Cluster Centers

Neeti Arora[Ȧ*] and Mahesh Motwani[Ȧ]

[Ȧ]Department of Computer Science and Engineering, Rajiv Gandhi Technical University, Madhya Pradesh, India

Accepted 05 June 2014, Available online 20 June 2014, **Vol.4, No.3 (June 2014)**

Abstract

Data mining techniques help in business decision making and predicting behaviors and future trends. Clustering is a data mining technique used to make groups of objects that are somehow similar in characteristics. Clustering analyzes data objects without consulting a known class label or category i.e. it is an unsupervised data mining technique. K-means is a widely used partitional clustering algorithm but the performance of K-means strongly depends on the initial guess of centers (centroid) and the final cluster centroids may not be the optimal ones. Therefore it is important for K-means to have good choice of initial centroids. By augmenting K-means with a technique of selecting centroids using criteria of sum of distances of data objects to all other data objects, we obtain an algorithm Farthest Distributed Centroids Clustering (FDCC) that result in better clustering as compared to not only the K-means partition clustering algorithm but also to the agglomerative hierarchical clustering algorithm and Hierarchical partitioning clustering algorithm. Unlike K-means FDCC algorithm does not perform random generation of the initial centers and does not produce different results for the same input data.

Keywords: *Initial centroids; Recall; Precision; Partitional clustering; Agglomerative hierarchical clustering and Hierarchical partitioning clustering.*

1. Introduction

Data mining tools and techniques allow an organization to make creative decisions and subsequently do proper planning. Clustering is the data mining technique that groups the data objects into classes or clusters, so that objects within a cluster have high similarity in comparison to one another(intra-class similarity) but are very dissimilar to objects in other clusters(inter-class similarity
Each of the different clustering methods available may give a different grouping of a dataset. These methods are divided into two basic types: hierarchical and partitional clustering (Dunham, 2006). Hierarchical clustering either merges smaller clusters into larger ones (Agglomerative) or splits larger clusters into smaller ones (Divisive). Agglomerative clustering starts with one point in each cluster and at each step selects and merges two most clusters that are most similar in characteristics. The process is repeated till the required number of clusters is formed. Divisive clustering starts with all data points in one cluster and splits them until the required number of clusters is formed. The agglomerative and divisive clustering methods further differ in the rule by which it is decided which two small clusters are merged or which large cluster is split. Partitional clustering, attempts to directly decompose the data set into a set of disjoint clusters. Each object is a member of the cluster with which it is most similar. K-means Lloyd S.(1982), (MacQueen 1967) is one of the most widely used partitional clustering algorithms. The algorithm clusters the n data points into k groups, where k is provided as an input parameter. It defines k centroids, one for each cluster. For this k data points are selected at random from D as initial centroids. The next step is to take each point belonging to the given data set and assign it to the nearest centroid. 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 mean of each cluster is calculated to find the new centroids for each cluster. It then assigns all the data points to clusters based upon their proximity to the mean of the cluster. The cluster's mean is then recomputed and the process begins again. In due course, a situation is reached where the clusters centres do not change anymore. This becomes the convergence criterion for clustering.

Algorithm K-means

Input: Dataset D of n data points $\{d_1, d_2,\ldots,d_n\}$ and k the number of clusters.

Output: Set of k Clusters.

```
1.  Choose k points at random from D as initial centroids.
2.  Repeat.

Assign each data point in D to the group that has the closest centroids.
Recalculate the positions of the k centroids by taking the means.
```

*Corresponding author: **Neeti Arora** is a Research Scholar and

Until the centroids no longer move.

The K-means algorithm has a drawback that it results in different types of clusters depending on the initialization process, which randomly generates the initial centers. Thus the performance of K-means depends on the initial guess of centers as different runs of K-means on the same input data might produce different results (Khan 2004) . Study and comparison of different clustering algorithms such as K-means algorithm and hierarchical clustering algorithm is provided in (Osama 2008). Researchers (Jain et. al. 1998), (Larsen 1999) have found that hierarchical clustering algorithms are better than partitional clustering algorithms in terms of clustering quality. But the computational complexity of hierarchical clustering algorithms is higher than that of partitional clustering algorithms (Steinbach 2000), (Xu et. al. 2005). The hybrid hierarchical partitioning algorithms (Zhao et. Al 2002) combine the advantages of the partitional and hierarchical clustering. Bisecting K-means (Murugesam et. al. 2011) is a hybrid hierarchical partitioning algorithm that produces the required k clusters by performing a sequence of $k - 1$ repeated bisections. In this approach, the input dataset is first clustered into two groups using K-means (2-way), and then one of these groups is selected and bisected further. This process continuous until the desired number of clusters is found. It is shown in (Murugesam et. al. 2011) that the hierarchical partitioning algorithm produces better clustering than the traditional clustering algorithms.

We have developed a new clustering algorithm Farthest Distributed Centroids Clustering (FDCC) that unlike K-means does not perform random generation of the initial centers and does not produce different results for the same input data. The proposed algorithm produce better quality clusters than the partitional clustering algorithm, agglomerative hierarchical clustering algorithm and the hierarchical partitioning clustering algorithm.

2. Related Work

Cluster center initialization algorithm CCIA (Khan et. al. 2004) is based on observation, that some of the patterns are very similar to each other and that is why they have same cluster membership irrespective to the choice of initial cluster centers. Also, an individual attribute may provide some information about initial cluster center. The experimental results show the improved performance of the proposed algorithm.

The method proposed in (Bradley et. al. 1998) refines the initial centroid points by analyzing distribution of data and probability of data density. The method operates over small subsamples of a given data base, hence requiring a small proportion of the total memory needed to store the full database. The clustering results obtained by refined initial centroids are superior to the randomly chosen initial centroids in K-means algorithm.

A global K-means clustering algorithm (Likas et. al. 2003) incrementally adds one cluster center at a time and performs n executions of the K-means algorithm from appropriate initial positions. An improved version of K-means (Yuan et al.2004) evaluates the distance between every pair of data points and then finds out those data points which are similar. This method finally chooses the initial centroids according to these data points found. This method produces better clusters as compared to the original K-means algorithm.

An optimized K-means algorithm proposed in (Barakbah et. al. 2005) spreads the initial centroids in the feature space so that the distances among them are as far as possible. An efficient enhanced K-means method is proposed in (Fahim A. M. et al. 2006) for assigning data points to clusters. The original K-means algorithm is having high time complexity because each iteration computes the distances between data points and all the centroids. This uses a distance functions based on a heuristics to reduce the number of distance calculations. This improves the execution time of the K-means algorithm. Initial centroids are determined randomly like K-means algorithm, therefore this method also produces different clusters in every run of the algorithm like in K-means algorithm. Thus the clustering results are same as K-means algorithm but the time complexity of this algorithm is lower than K-means. A new algorithm for finding the initial centroids by embedding Hierarchical clustering into K-means clustering is proposed in (Barakbah et. al. 2007).

An algorithm proposed in (Deelers et. al. 2007) computes initial cluster centers for K-means algorithm by partitioning the data set in a cell using a cutting plane that divides cell in two smaller cells. The plane is perpendicular to the data axis with the highest variance and is designed to reduce the sum squared errors of the two cells as much as possible, while at the same time keep the two cells far apart as possible. Also they partitioned the cells one at a time until the number of cells equals to the predefined number of clusters, k. In their method the centers of the k cells become the initial cluster centers for K-means algorithm.

A clustering approach inspired by the process of placing pillars in order to make a stable house is proposed in (Barakbah et. al. 2009). This technique chooses the position of initial centroids by using the farthest accumulated distance metric between each data point and all previous centroids, and then, a data point which has the maximum distance will be selected.

An algorithm for clustering is proposed in (Neeti et. al.2014) that selects initial centroids using criteria of finding sum of distances of data objects to all other data objects.

3. Proposed Algorithm

We have developed an algorithm Farthest Distributed Centroids Clustering (FDCC) for clustering that finds out optimal initial centroids from the given dataset. The selection of initial centroids is done such that the performance of K-means clustering is improved. The proposed algorithm produces better quality clusters than the partitional clustering algorithm, agglomerative hierarchical clustering algorithm and the hierarchical partitioning clustering algorithm.

3.1 Basic Concept

The K-means algorithm performs random generation of the initial centroids that may result in placement of initial centroids very close to each other. Due to this, these initial centroids may be trapped in local optima. The technique proposed in this paper selects a good set of initial centroids in such a way that they are spread out within the data space and are as far as possible from other initial centroids. Figure 1 illustrates the selection of four initial centroids C1, C2, C3 and C4 with the proposed technique. As is evident there are four clusters in the data space. The proposed technique selects a point *d* as the first initial centroid using a distance criteria explained in section 3.2. Once this point d is selected as initial centroid, the proposed technique avoids the points near to d from being selected as next initial centroids. This is how C1, C2, C3 and C4 are distributed as far as possible from each other. The refined initial centroid condition makes K-means to converge to a better local minimum.

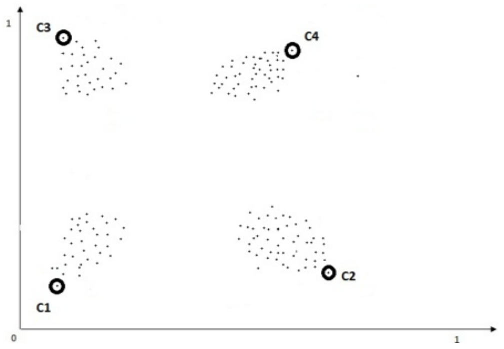

Fig. 1 Illustration of selecting a good set of initial centroids

3.2 Farthest Distributed Centroid Clustering

Let the clustering of *n* data points in the given dataset D is to be done into *k* clusters. In Farthest Distributed Centroids Clustering (FDCC) algorithm, we calculate the distance of each data point $d_{i=1 \text{ to } n}$ in the given dataset D from all other data points and store these distances in a distance matrix DM. We now calculate total distance of each data point $d_{i=1 \text{ to } n}$ with all other data points. The total distance for a point d_i is sum of all elements in the row of the DM corresponding to d_i. These sums are stored in a sum of distances vector SD. The vector SD is sorted in decreasing order of total distance values. Let P-SD is the vector of data points corresponding to the sorted vector SD, i.e. P-SD [1] will be the data point whose sum of distances (available in SD [1]) from all other data points is maximum and P-SD [2] will be the data point whose sum of distances (available in SD [2]) from all other data points is second highest. In general P-SD [i] will be the data point whose sum of distances from all other data points is in SD [i].

We now make the first point *d* of the vector P-SD as the first initial centroid. Put this initial centroid point in the set S of initial centroids. Now to avoid the points near to *d* from being selected as next initial centroids we define a variable *x* as defined in Eq. (1).

$$x = \text{floor}\ (n/k) \quad (1)$$

Here, the floor (n/k) function maps the real number (n/k) to the largest previous integer i.e. it returns the largest integer not greater than (n/k).

We now discard the next *x* number of points of the vector P-SD and define the next point left after discarding these x numbers of points from this vector P-SD, as the second initial centroid. Now discard the next *x* number of points from this vector P-SD and define the next point left after discarding these *x* numbers of points from this vector P-SD, as the third initial centroid. This process is repeated till *k* numbers of initial centroids are defined. These *k* initial centroids are now used in the K-means process as substitute for the *k* random initial centroids. K-means is now invoked for clustering the dataset D into *k* number of clusters using the initial centroids available in set S.

Algorithm FDCC

Input: Graph G of *n* data points $\{d_1, d_2, \ldots, d_n\}$ in D and *k* the number of clusters.
Output: Initial *k* cluster centroids and the set of *k* clusters.

```
1.  Find the distance matrix DM from G;
2.  Calculate the sum vector SD from DM;
3.  SD = Sort (SD, decreasing); // sort SD in decreasing order
4.  P-SD = data points corresponding to the sorted vector SD
5.  ck = 1; // ck indicates counter for number of initial centroids
6.  ic = 1; // ic is index of next element to be chosen as initial centroid from SD
7.  d = P-SD [ic]; // point corresponding to the SD[1] is chosen as first initial centroid
8.  S = d; // S is set of final initial centroids
9.  x = floor (n/k)
10. ic = ic + x + 1; // discard next x elements of SD
11. ck = ck +1;
12. While (ck <=k)
{
d = P-SD [ic]; // point corresponding to the SD [ic] is next initial centroid
S = S UNION d;
ck = ck + 1;
ic = ic + x + 1;
} End while
13. K-means(D, S, k);
```

3. Experimental Results

The experiments are conducted on synthetic and real data points to compute different sets of clusters using the partitional clustering algorithm, agglomerative hierarchical clustering algorithm, hierarchical partitioning clustering algorithm and the proposed FDCC algorithm. The experiments are performed on core i5 processor with a speed of 2.5 GHz and 4 GB RAM using Matlab. The comparison of the quality of the clustering achieved with FDCC is made with the quality of the clustering achieved with:

Table 1 Average Recall and Average Precision for synthetically generated datasets

Algorithm	R2400		R12000		R24000	
	Recall	Precision	Recall	Precision	Recall	Precision
FDCC	100 %	100 %	100 %	100 %	100 %	100 %
K-means	78.21 %	91.67 %	75.13 %	75 %	65.83 %	65.83 %
ClusterData	100 %	100 %	100 %	100 %	100 %	100 %
Cluto-rb	22.67 %	26.03 %	21.63 %	25.92 %	18.5 %	17.83 %

1. Partitional clustering technique (Dunham 2006), (Lloyd 1982), (MacQueen 1967) The K-means is the partitional clustering technique available as a built in function in Matlab (Mathworks 2009).

2. Hierarchical clustering technique (Dunham 2006), (Murugesam et.al. 2011) .The ClusterData is agglomerative hierarchical clustering technique available as a built in function in Matlab (Mathworks 2009). The single linkage is the default option used in ClusterData to create hierarchical cluster tree.

3. Hierarchical partitioning technique (Murugesam et.al. 2011). CLUTO Karypis(2003) is a software package for clustering datasets and for analyzing the characteristics of the various clusters. CLUTO contains both partitional and hierarchical clustering algorithms. The repeated bisections method available in CLUTO is a hierarchical partitioning algorithm that produces the required *k* clusters by performing a sequence of k − 1 repeated bisections. In this approach, the input dataset is first clustered into two groups, and then one of these groups is selected and bisected further. This process continuous until the desired number of clusters is found. This effectively is the bisect K-means divisive clustering algorithm and is the default option in CLUTO named as Cluto-rb.

Recall and Precision (Kowalski 1997) has been used to evaluate the quality of clustering achieved. Recall is defined as the proportion of data points that have been correctly put into a cluster among all the relevant points that should have been in that cluster. Precision is defined as the proportion of data points that have been correctly put into a cluster among all the points put into that cluster.

Recall = Data points correctly put into a cluster / All the points that should have been in that cluster

Precision = Data points correctly put into a cluster / Total points put into that cluster

The comparison of the quality of the clustering achieved with FDCC and other algorithms is made in terms of the percentage accuracy of grouping of objects in correct clusters. The recall accuracy is calculated by dividing the number of relevant data points correctly put in a particular group (by running the clustering algorithm) by the total number of data points that should have ideally been in that group. The precision accuracy is calculated by dividing the number of relevant data points correctly put in a particular group (by running the clustering algorithm) by the total number of all data points put in that group.

4.1 Experiments on Synthetic Datasets

4.1.1 R2400, R12000 and R24000

We have created the synthetic datasets by generating random data points in 2 dimensions. We have generated four synthetic datasets with 2400, 12000 and 24000 records in 2 dimensions. Let the names of these datasets are R2400, R12000 and R24000 respectively. These random data points are generated and distributed among 12 groups. Equal numbers of data points are generated in each group.

The range of points in the first group is chosen from [0,100] to [200,300]. The range of points in the second group is from [0,700] to [200,900]. The range of points in the third group is from [0, 1300] to [200, 1500]. The range of points in the fourth group is from [0, 1900] to [200, 2100]. The range of points in the fifth group is from [0, 2500] to [200, 2700]. The range of points in the sixth group is from [0, 3100] to [200, 3300]. The range of points in the seventh group is from [0, 3700] to [200, 3900]. The range of points in the eighth group is from [0, 4300] to [200, 4500]. The range of points in the ninth group is from [0, 4900] to [200, 5100]. The range of points in the tenth group is from [0, 5500] to [200, 5700]. The range of points in the eleventh group is from [0, 6100] to [200, 6300], and finally the range of points in the twelfth group is from [0, 6700] to [200, 6900].

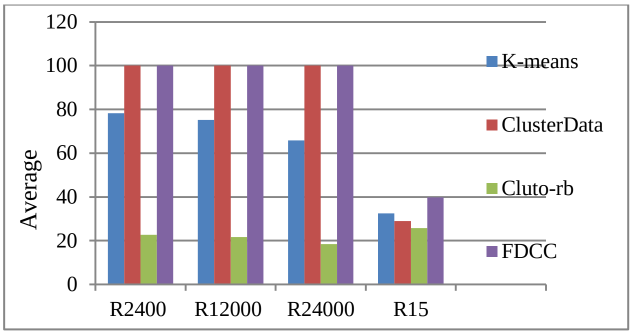

Fig. 2 Average Recall for synthetic datasets.

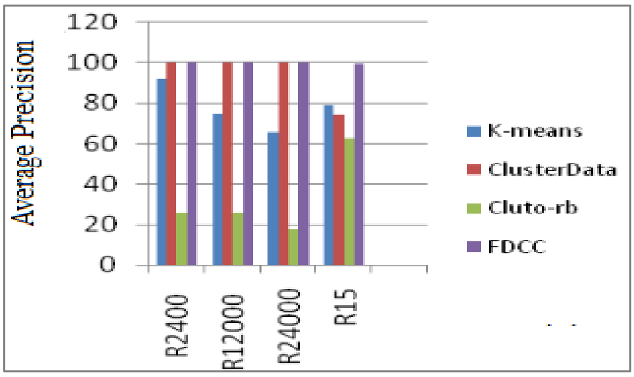

Fig. 3 Average Precision for synthetic datasets.

The recall and precision for creating twelve clusters using K-means, ClusterData, Cluto-rb and FDCC algorithms are found for datasets to see how correctly the data points are put into a cluster. The results of clustering using these algorithms are shown in Table 1. The graph plotted for average recall in percentage using these techniques is shown in Figure 2 and the graph plotted for average precision in percentage using these techniques is shown in Figure 3.

As can be seen from the results, FDCC algorithm performs 100 % correct clustering and is far better than the K-means and the Cluto-rb algorithms.

4.1.2 R15

R15 (Veenman et. al. 2002) is a 2 dimensional shape dataset with 600 vectors. These 600 vectors belong to 15 classes with 40 vectors per class. These 15 clusters of 40 points each are positioned in two rings (7 clusters in each ring) around a central cluster.

We have formed 15 clusters of these images using K-means, ClusterData, Cluto-rb and FDCC algorithms. The average recall and precision using these algorithms is determined and shown in Table 2. The graph plotted for average recall in percentage using these techniques is shown in Figure 2 and the graph plotted for average precision in percentage using these techniques is shown in Figure 3.

Table 2 Average Recall and Average Precision for R15 shape datasets.

Algorithm	**Average Recall**	**Average Precision**
FDCC	39.87 %	99.67 %
K-means	32.47 %	78.88 %
ClusterData	28.93 %	74.36 %
Cluto-rb	25.8 %	62.77 %

The results show that the recall and precision of FDCC algorithm is better than the recall and precision of K-means, ClusterData and Cluto-rb. Hence, the FDCC algorithm produces better quality clusters.

4.2 Experiments on real web datasets

The proposed algorithm FDCC has been implemented by us on web datasets of text and images. We have first processed and described the image datasets in terms of features that are inherent in the images themselves. The features extracted from the images are smaller in size as compared to the original image data and are being used in place of the images themselves for mining. These are represented in the form of real–valued components called the feature vectors. The colour moments (Stricker et. al. 1995) are used by us as the property in the feature extraction technique.

K-means, ClusterData, Cluto-rb and FDCC algorithms are applied to cluster the real datasets. The image datasets are clustered over the extracted features. Recall and Precision of the clusters formed using these techniques are calculated to determine and compare the quality of clustering. The datasets and experimental results achieved using K-means, ClusterData, Cluto-rb and FDCC are described below.

4.2.1 Results on Corel5K Dataset

Corel5K is a collection of 5000 images downloaded from website (Duygulu P., et al 2002). We have formed 10 clusters of these images using K-means, ClusterData, Cluto-rb and FDCC algorithm. The average recall and precision using these algorithms is shown in Table 3. The graph plotted for average recall in percentage using these techniques is shown in Figure 4 and the graph plotted for average precision in percentage is shown in Figure 5. The results show that the recall and precision of FDCC are better than that of K-means, ClusterData and Cluto-rb. Hence, the FDCC algorithm produces better quality clusters.

4.2.2 Results on Corel (Wang) Dataset

Corel (Wang) database consists of 700 images that are a subset of 1,000 images of the Corel stock photo database which have been manually selected and which form 10 classes of 100 images each. This database is downloaded from website (Wang et. al. 2001) and contains 1025 features per image.

We have formed 10 clusters of these images using K-means, ClusterData, Cluto-rb and FDCC algorithm. The average recall and precision using these algorithms is determined and shown in Table 3. The graph plotted for average recall in percentage using these techniques is shown in Figure 4 and the graph plotted for average precision in percentage using these techniques is shown in Figure 5. The results show that the recall and precision of FDCC are better than the recall and precision of K-means, ClusterData and Cluto-rb. Hence, the FDCC algorithm produces better quality clusters.

4.2.3 Wine Dataset

The wine recognition dataset (Aeberhard et. al. 1992) results from the chemical analysis of wines grown in the same region in Italy but derived from three different cultivars. The analysis determined the quantities of 13 constituents found in each of the three types of classes of wines. The dataset consists of 178 instances. The number of instances in class1, class2 and class3 are 59, 71 and 48 respectively. The wine dataset is downloaded from the website (Forina et. al. 1991).

We have formed 3 clusters of these images using K-means, ClusterData, Cluto-rb and FDCC algorithm. The average recall and precision using these algorithms is determined and shown in Table 3. The graph plotted for average recall in percentage using these techniques is shown in Figure 4 and the graph plotted for average precision in percentage using these techniques is shown in Figure 5. The results show that the recall and precision of FDCC are better than the recall and precision of K-means, ClusterData and Cluto-rb. Hence, the FDCC algorithm produces better quality clusters.

Table 3 Average Recall and Average Precision for real datasets.

Algorithm	Corel5K		Corel (Wang)		Wine	
	Recall	Precision	Recall	Precision	Recall	Precision
FDCC	30.02 %	38.2 %	100 %	100 %	41.67 %	72 %
K-means	20.02 %	33.3 %	60.7 %	86.71 %	33.67 %	61 %
ClusterData	10.08 %	41 %	100 %	100 %	25.37 %	47 %
Cluto-rb	23.02 %	30.32 %	33.29 %	27.91 %	39.67 %	69 %

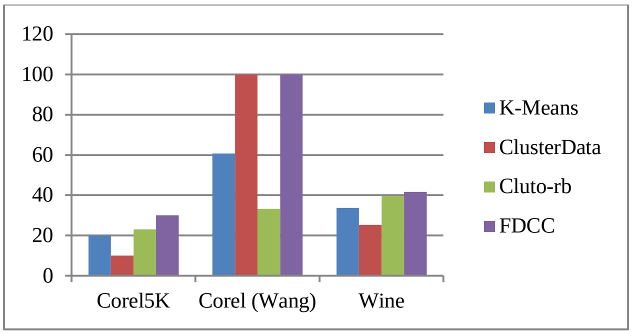

Fig. 4 Average Recall for real datasets.

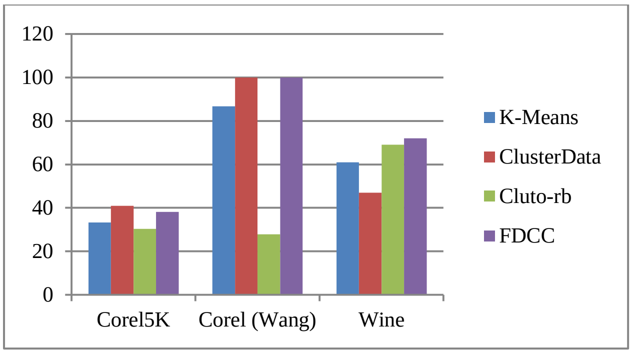

Fig. 5 Average Precision for real datasets

Conclusion

The quality of clustering of K-means algorithm depends on the initialization of *k* data points. These *k* initial centroids are chosen randomly by the K-means algorithm and this is the major weakness of K-means. The clustering algorithm proposed in this paper is based on the assumption that good clusters are formed when the choice of initial *k* centroids is such that they are as far as possible from each other. FDCC algorithm proposed here is based on computing the total distance of a node from all other nodes. The algorithm is tested on both synthetic database and real database from web. The experimental results show the effectiveness of the idea and the algorithm produce better quality clusters than the partitional clustering algorithm, agglomerative hierarchical clustering algorithm and the hierarchical partitioning clustering algorithm.

Developed clustering algorithm can be applied in many fields, for instance: In Marketing to find groups of customers with similar behavior given a large database of customer data; In City-planning to identify groups of houses according to their house type, value and geographical location; In WWW for document classification, clustering weblog data to discover groups of similar access patterns; In Medicine for clustering diseases, cures for diseases, or symptoms of diseases.

References

Aeberhard S., Coomans D. and Vel O. D.(1992), Comparison of Classifiers in High Dimensional Settings, *Tech. Rep. No. 92-02, Dept. of Computer Science and Dept. of Mathematics and Statistics*, James Cook University of North Queensland.

Barakbah A. R. and Arai K. (2007), Hierarchical K-means: An algorithm for centroids initialization for K-means, *Reports of the faculty of Science & Engineering*, Saga University, Japan, vol. 36 no.1.

Barakbah A. R. and Helen A. (2005), Optimized K-means: An algorithm of initial centroids optimization for K-means, *Proc. Soft Computing, Intelligent Systems and Information Technology (SIIT)*, Surabaya, Indonesia, pp. 263-266.

Barakbah A. R. and Kiyoki Y. (2009), A Pillar Algorithm for K-means optimization by distance maximization for initial centroid designation, *IEEE Symposium on Computational Intelligence and Data Mining (IDM)*, Nashville-Tennessee, pp. 61-68.

Bradley P. S. and Fayyad U. M. (1998), Refining initial points for K-Means clustering, *Proc. 15th International Conference on Machine Learning, Morgan Kaufmann*, San Francisco, CA, pp. 91-99.

Deelers S. and Auwatanamongkol S.(2007), Engineering k-means algorithm with initial cluster centers derived from data partitioning along the data axis with the highest variance, *Proc. World Academy of Science, Engineering and Technology*, 26, pp. 323-328.

Dunham M. (2006), *Data Mining – Introductory and advanced concepts*, Pearson Education.

Duygulu P., et al (2002), Object recognition as machine translation: Learning a lexicon for a fixed image vocabulary, *proc. 7th European Conference on Computer Vision, pp. 97-112.* Retrieved from http://sci2s.ugr.es/keel/dataset_smja.php?cod=230

Fahim A. M., et al.(2006), An efficient enhanced k-means clustering algorithm, *Journal of Zhejiang University Science*, vol. 7, no. 10, pp. 1626-1633.

Forina and Aeberhard.(1991),*UCI Machine Learning Repository*. Retrieved from http://archive.ics.uci.edu/ml/datasets/Wine

Jain A. K. and Dubes R. C. (1998), *Algorithm for Clustering in Data*, Prentice Hall.

Karypis (2003), Cluto: A Clustering Toolkit. Release 2.1.1, *Tech. Rep. No. 02-017 University of Minnesota, Department of Computer Science*, Minneapolis, MN 55455. Retrieved from www.cs.umn.edu/~karypis/cluto.

Khan S. S. and Ahmad, A. (2004), Cluster center initialization algorithm for k-means algorithm, *Pattern Recognition Letters*, vol. 25 no.11, pp. 1293-1302.

Kowalski G. (1997), *Information Retrieval Systems – Theory and Implementation*, Kluwer Academic Publishers.

Larsen B. and Aone C.(1999), Fast and Effective Text Mining using Linear-Time Document Clustering, *Proc. 5th ACM*

SIGKDD International Conference on Knowledge Discovery and Data, San Diego, CA, USA, pp. 16-22.

Likas A., Vlassis N. and Verbeek J. J.(2003), The global k-means clustering algorithm, *Journal of Pattern Recognition*, vol. 36, no. 2, pp. 451-461.

Lloyd S.(1982), Least Squares Quantization in PCM, *IEEE Transactions on Information Theory*, vol. 28 no. 2, pp. 129-137.

MacQueen J. B. (1967), Some Methods for Classification and Analysis of Multivariate Observations, *Proc. 5th Berkeley Symposium on Mathematical Statistics and Probability*, Berkeley, University of California Press, pp. 281-297.

MathWorks(2009) *MatLab: The Language of Technical Computing,* Retrieved from http://www.mathworks.in/products/matlab/

Murugesam K. and Zhang J. (2011), Hybrid hierarchical clustering: An experimental analysis, Tech. *Rep. No. CMIDA-HiPSCCS 001-11, Department of Computer Science*, University of Kentucky, Lexington, KY.

Osama Abbu Abbas(2008), Comparisons between Data Clustering Algorithms, *The International Arab Journal of Information Technology*, Vol. 5, No.3, pp. 320-325.

Steinbach M., Karypis G. and Kumar V.(2000), A comparison of document clustering techniques, *KDD Workshop on Text Mining,* Boston, MA, pp. 109-111.

Stricker M. and Orengo M. (1995), Similarity of color images. *Proc. Storage and Retrieval for Image and Video Databases III (SPIE)*, San Jose, CA, USA, pp. 381-392.

Veenman C. J., Reinders M. J. T. and Backer E. (2002), A maximum variance cluster algorithm, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 24 no. 9, pp. 1273-1280. Retrieved from http://cs.joensuu.fi/sipu/datasets/

Wang J. Z., Li J. and Wiederhold G. (2001), SIMPLIcity: Semantics-Sensitive Integrated Matching for Picture Libraries, *IEEE Transactions on Pattern Analysis and Machine Intelligence***,** vol. 23 no. 9, pp. 947-963, Retrieved from http://users.dsic.upv.es/~rparedes/english/research/rise/MiPRCV-WP6-RF/

Xu R. and Wunsch D. (2005), Survey of clustering algorithms, *IEEE Transactions on Neural Networks*, vol. 16 no. 3, pp. 645-678.

Yuan F., et al.(2004), A new algorithm to Get the Initial Centroids, *Proc. 13th International Conference on Machine Learning and Cybernetics*, Shanghai, pp. 26-29.

Zhao Y. and Karypis G. (2002), Evaluation of Hierarchical Clustering Algorithms for Document Datasets, *Proc. 11th International Conference on Information and Knowledge Management*, New York, USA, ACM press, pp. 515-524.

Neeti Arora and Mahesh Motwani (2014), Sum of Distance based Algorithm for Clustering Web Data, International Journal of Computer Applications, Volume 87, N0. 7, pp. 26-30.