International Journal of Computational and Experimental Science and Engineering (IJCESEN)
Vol. 2-No.1 (2016) pp. 9-18
http://dergipark.ulakbim.gov.tr/ijcesen

Copyright © IJCESEN

***ISSN:* 2149-9144**

Research Article

Improved K-means Clustering Algorithm for Biological Data using Voronoi Diagram#

Damodar REDDY[1]*, Pravin PAWAR[1]

[1]National Institute of Technology Goa, Farmagudi, Goa, India

* Corresponding Author : dr.reddy@nitgoa.ac.in

# Presented in "2nd International Conference on Computational and Experimental Science and Engineering (ICCESEN-2015)"

Keywords
Clustering
K-means
Voronoi Diagram
Biological Data

Abstract: As a simple clustering method K-means is known as an algorithm of choice for many clustering challenges due to its performance of clustering large data sets. However, it has two major drawbacks, the random selection of initial cluster centers and the pre estimation of 'K' value in advance. Here, we propose a method that overcomes these problems with the help of Voronoi diagram. To resolve the random selection of initial cluster centers, we use Voronoi diagram. The vertices in the Voronoi diagram are located first and then merged iteratively to converge to 'k' number of points which can be treated as initial cluster centers for K-means. The second problem of inputting 'K' value in advance is enhanced by taking a limit on the radius of Voronoi circle. The experimental results carried out on various synthetic and biological data sets are proved the efficiency of the proposed method..

1. Introduction

Data mining, or knowledge discovery in databases is the technique of analysing data to discover previously unknown information. Clustering [1] is one of the primary data analysis methods refers to a task of partitioning the given set of patterns into homogeneous disjoint groups called clusters and can be defined as regions in which the density of the objects is locally higher than in other regions. Therefore, a cluster is a collection of objects that are similar among themselves and dissimilar to the objects belonging to other clusters. Clustering do not make any statistical assumptions to data. Hence it is an example of unsupervised classification. It helps in finding hidden patterns and describes the underlying knowledge form a large data set. Various algorithms have been developed to solve different type of clustering problems. As a result clustering has variety of applications in various domains such as image processing [2], wireless sensor networks [5], bioinformatics [3] and knowledge discovery [4]. Clustering algorithms are mainly categorized into two types, Hierarchical algorithms and Partition algorithms. In hierarchical clustering [6] the given data set divided into smaller sub sets in hierarchical fashion. Hierarchical clustering does not require us to specify the number of clusters in advance and most hierarchical algorithms that have been used are deterministic. Hierarchical algorithms are divided into two types, agglomerative and divisive. Unlike hierarchical, partitional clustering [7] algorithms attempts to directly decompose the data set into a set of disjoint clusters. Much attention is paid in case of partitional clustering techniques and number of clustering algorithms have been proposed. A commonly used partitional clustering method is K-means [1]. It is one of the most used iterative clustering algorithm used in variety of domains because of its simplicity and effectiveness. The K-means algorithm attempts to find the cluster centres, (c1,…,ck), such that the sum of the squared distances (this sum of squared distances is termed the Distortion, D) of each data point (xi) to its nearest cluster centre (ck) is minimized. Here the distortion is defined as follows.

$$D = \sum_{i=1}^{n}\left[\min_{k=(1\ldots K)} d(x_i, c_k)\right]^2 \qquad (1)$$

However, K-means suffers from two major problems namely, the random selection of initial cluster centers and estimation of output number of clusters in advance. The in appropriate selection of initial cluster centers or the k value affects the clustering results significantly. Various attempts [8], [9], [10] have been made to select the initial cluster centers and to estimate the k value exactly. But they do not fulfil all the requirements in terms of efficiency, fastness and time complexity etc. Therefore, we propose a novel algorithm that finds the good seeds to act as initial clusters and to estimate the k value in advance with the help of Voronoi diagram [11]. It is a well known technique from computational geometry, especially popular for nearest neighbor problems. It has been used for cluster analysis and few algorithms [12], [13] have been developed. The Voronoi diagram is used to form the initial cluster centers with the help of voronoi vertices and circles. A threshold limit on the radius of the Voronoi circle is given to form k points that are treated as initial cluster centers to K-means. The proposed algorithm is tested on various synthetic and real world data sets and the results are compared with the classical K-means and improved K-means algorithms to show the efficiency of the proposed method. The rest of the paper is structured as follows. The useful terminologies are discussed in section 2 and the related work is described in section 3. We formulize the proposed algorithm in step 4. Finally experimental results are shown in section 5 followed by conclusion in section 6.

2. Basıc Termınologıes

We first discuss some terminologies that can help in understanding our proposed algorithm as follows.

2.1 K-means

K-means [1] algorithm finds the clusters by partitioning the given data set by minimizing the squared error between the empirical mean of a cluster and the points in the cluster. Let Ck denote the kth cluster of the data set: {x1, x2,…, xn}. Then if μj is the mean of the cluster Cj, the squared error between μk and the point xi within Cj is as follows.

$$S(C_j) = \sum_{x_i \in C_j} \| x_i - \mu_j \| \qquad (2)$$

The aim of K-means is to reduce the sum of squared error for all the ‘K’ clusters. i.e., to minimize S(C).

$$S(C) = \sum_{j=1}^{K} \sum_{x_i \in C_j} \| x_i - \mu_j \| \qquad (3)$$

The algorithm is as follows [14].

Step 1: Select K initial cluster centers $c_1, c_2,\ldots,c_K$ randomly from the given n data points $\{x_1, x_2,\ldots, x_n\}$, $K \le n$.

Step 2: Assign each point x_i, $i = 1, 2, \ldots, n$ to the cluster C_j corresponding to the cluster center c_j, for $j = 1, 2, \ldots, K$ iff $\|x_i - c_j\| \le \|x_i - c_p\|$ $p = 1, 2, \ldots, K$ and $j \neq p$.

Step 3: Compute new cluster centers $c_1^*, c_2^*,\ldots,c_K^*$ as follows

$$c_i^* = \frac{1}{n_i}\sum_{x_j \in C_i} x_j \quad \text{for } i = 1, 2,\ldots,K.$$

where n_i is the number of data points belonging to the cluster C_i.

Step 4: If $c_i^* = c_i$, $i = 1, 2,\ldots,K$, then terminate. Otherwise continue from step 2.

2.2 Voronoi Diagram

Given a set of points, Voronoi diagram [11] is a partition of space into cells, each of which consists of the points closer to one particular object than to any others. It is formally defined as follows.

Let $S = \{p_1, p_2,\ldots,p_n\}$ be a set of n points in a d-dimensional Euclidean space and $d(a, b)$ denotes distance between the points a and b in this space. Then the Voronoi diagram of S (see Fig. 1) is defined as the subdivision of the space into n cells, one for each point in S. A point u lies in the cell corresponding to the point p_i iff $d(u, p_i) < d(u, p_j)$ for each $p_j \in S$ and $j \neq i$. We denote the Voronoi diagram of S by Vor(S) and the cell corresponding to the point p_i by $V(p_i)$. We call the vertices of a Voronoi diagram as Voronoi vertices. There are maximum $2n$-5 Voronoi vertices in a Voronoi diagram of *n* points. It is obvious from the definition that for each point *pi* ϵ *S*, V(p_i) contains all the points that are closer to p_i than to other points of *S*. For a Voronoi vertex *v*, we define the largest empty circle of *v* (see Fig. 2) with respect to *S*, as the largest circle with *v* as its centre that contains no point of *S* in its interior. We denote this circle by CirS(*v*). The Voronoi vertices have the property that a point *q* is a vertex of Vor(*S*) iff CirS(*q*) contains three or more points of *S* on its boundary. A point *p* is said to be covered by a vertex *q* if and only if *p* lies on the boundary of CirS(*q*).

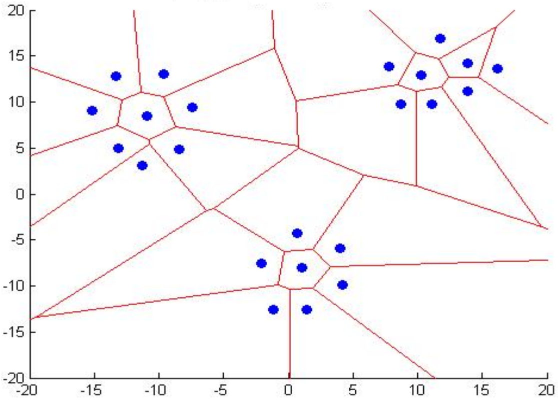


Figure 1. *Voronoi diagram of given points.*

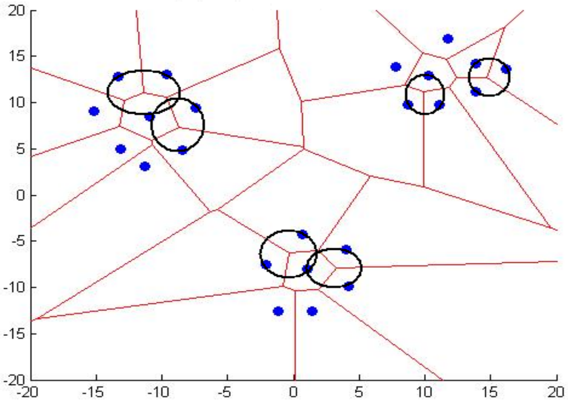


Figure 2. *Voronoi diagram with Voronoi circles (all circles are not shown)*

In our proposed algorithm we use the Vornoi vertices to represent all the given points and then these vertices are further merged to form 'k' number of points where each point represents a cluster center. The merging of the closer vertices is done by reconstructing the Voronoi diagram with these vertices. we input a limit on the radius of the Voronoi circle to terminate the process and the resultant points are the required initial cluster centers for K-means.

2.3 Dynamic Validity Index

Validity index is used to measure the quality of the clusters formed, especially in case of multi-dimensional data. Many validity indices have been proposed so far. In our algorithm, we use dynamic validity index (DVI) [15] defined as follows. Let N be the number of data point, K be the pre-defined upper bound number of clusters, and zi be the center of the cluster Ci. The dynamic validity index is given by

$$DVI = \min_{k=1,2,,,,,K} \{IntraRatio(k) + \gamma * InterRatio(k)\}$$

where

$$IntraRatio(k) = \frac{Intra(k)}{MaxIntra}, InterRatio(k) = \frac{Inter(k)}{MaxInter}$$

$$Intra(k) = \frac{1}{N}\sum_{i=1}^{k}\sum_{x-C_i}\left\|x - z_i\right\|^2, MaxIntra = \max_{i=1,2,...,k}(Intra(i))$$

$$Inter(k) = \frac{Max_{i,j}\left(\left\|z_i - z_j\right\|^2\right)}{Min_{i\neq j}\left(\left\|z_i - z_j\right\|^2\right)}\sum_{i=1}^{k}\left(\frac{1}{\sum_{j=1}^{k}\left\|z_i - z_j\right\|^2}\right)$$

and $$MaxInter = \underset{i=1,2,...,k.}{Max}(Inter(i))$$

Here, *Intra Ratio* stands for the overall compactness of clusters scaled from *Intra* term, where as *Inter Ratio* represents overall separation of clusters scaled from *Inter* term. The *Intra* term is the average distance of all the points within a cluster from cluster center. Then we have *Inter* term which is composed of two parts, both of them based on cluster centers. The value of *Inter* increases with the increment in *k*.

3. Related Work

3.1 K-means clustering algorithm

K-means algorithm developed by MacQueen [16] is one of the most popular nonhierarchical and squared error clustering technique that belongs to partitioning methods of clustering. It is a very robust technique and its convergence has always been proved. As we have discussed in Section 1 that it sometimes suffers from the global optima due to arbitrary selection of initial cluster centers. It also suffers from the estimation of correct number of clusters in advance. Many researches proposed various methods to overcome these problems, a good review of which can be seen from [9]. A recursive method for the initialization of cluster centre is proposed by Duda and Hart [17]. The algorithm proposed by Fisher [18] generates good seeds by constructing initial hierarchical clustering groups. Both Higgs et al., [19] and Snarey et al. [20] developed a method using MaxMin algorithm to choose a subset of the original database as initial cluster centers. Bradley et al., [21] formed the initial clusters based on the bilinear program, provided that the sum of the distances of each point to its nearest center is minimized. Khan and Ahmed [22] proposed an algorithm for resolving the problem of random selection. Considering all these issues, still there is no universal clustering technique that can initialize the cluster centers for K-means, due to the dissimilar characteristics of various problem domains. MacQueen, [16] introduced an online learning strategy that determines a set of cluster seeds based on the calculation of mean vector. But this method is costly in case of large data sets because of the repetitive calculation of mean vector every time a new point added. Tou and Gonzales [23] recommended another method based on the distance between the successive seeds and a threshold value. But this method entirely depends on the order of the points in the database and a user defined threshold value. Linde et al., [24] proposed a method based on Binary Splitting (BS) which splits the cluster centre using a small random vector. This method is computationally expensive as K-means is to be implemented after each split. Also the cluster quality mainly depends on the selection of random vector which determines the direction of the split. Kaufman and Rousseeuw [25] developed a method which is based on the reduction in the Distortion. Here the seeds that increase the reduction in the distortion are chosen for the next step. Babu and Murty [26] proposed a technique for the near optimal seed selection based on genetic programming. Although this method can find the optimal solution, yet it faces the problem of repetitive run of K-means until given number of clusters formed. This is also not robust in case of very large data bases. Huang and Harris [27] projected a method called Direct Search Binary Splitting (DSBS) which is same as the BS algorithm with a small change. Here the splitting is done efficiently through the Principle Component Analysis (PCA) which is based on the vector of Linde et al., [24] for the splitting. Thiesson et al., [28] introduced a method that depends on the mean value of the whole given data set which creates a set of K-points around the mean of the data. Bradley and Fayyad [29] proposed an initialization approach for K-means in which the given data points are randomly divided into few data sets and then K-means is applied on each set with the initial cluster centers chosen from Forgy’s method. They again apply K-means algorithm on the centers of the clusters formed and repeat the step. Finally the centre points left are for the initialization of K-means for the entire dataset.

3.2 Voronoi-based clustering algorithms

The Voronoi diagram [11] is an efficient technique from computational geometry that plays an eminent role for data clustering. It has been especially designed for nearest neighbor problems and applied extensively on cluster analysis. Few clustering algorithms [12], [13] have been developed based on Voronoi diagram. A brief survey of them is as follows. Haowen Yan et al.[30] proposed an algorithm to generate point clusters based on Voronoi diagram by considering four types of information. They are statistical, thematic, topological, and metric information’s. Jana et al. [31] proposed another clustering algorithm using Voronoi diagram and the cluster density proposed by Daxin Jiang [32]. In this method the clusters are formed by exploiting the Voronoi diagram as follows. 1) The Voronoi vertices are used as the initial centroids (cluster centres) and the points on its largest empty circles are used to form the initial clusters. 2) Only the neighbouring clusters that share a Voronoi edge are merged to produce the best clusters for the next iteration. Therefore, there is no need of searching the entire set and the overall run time of the proposed algorithm is reduced. Bishnu et al.[12] developed a method with the help of K-means and Voronoi diagram. In the first phase K-means algorithm is used to create a set of small clusters. Then in the next phase the actual clusters are formed with the help of Voronoi diagram. A novel clustering technique for uncertain data has been proposed by Ben kao et al.[33]. They developed few pruning techniques based on Voronoi diagram to reduce the number of expected distance calculation and then formed the clusters. Motivated with all these clustering methods, we propose a method based on K-means and Voronoi diagram.

4. Proposed Algorıthm

The main scheme behind the proposed technique is summarized as follows. Given the set of n data points, say S, the Voronoi diagram Vor(S) is first constructed. First of all, we find the minimum number of Voronoi vertices to cover all the given points. i.e., we represent all the given points by its closer Voronoi vertex. To find such useful Voronoi vertices, the Voronoi circles surrounding all the vertices are traced out. Then these Voronoi circles are sorted in ascending order with respect to the radius of their largest empty circle. We now consider the Voronoi circle with least radius and represent the points on its circumference with its vertex. Since we started with least radius Voronoi circle, this vertex is closer compared to all the other vertices. Then the next least radius circle is taken and the points on its boundary are taken to be represented by its vertex. If any point is already covered by a vertex, we ignore that point and proceed further. Because the Voronoi circles are sorted with respect to their radius in ascending order, hence, the points are covered by its closer vertices. We repeat the same step unless all the given ‘n’ points are covered by its closer vertices. Now the Voronoi diagram is reconstructed with the help of these Voronoi vertices. and repeat the same process. A limit on the radius of the Voronoi circle is given to terminate the procedure. If there is no Voronoi circle whose radius is less than that limit, then there is no further formation of new vertices to cover the points. We stop the process and store the resultant vertices (assume ‘k’) in a set. We then run K-means algorithm on the given data set with these ‘k’ points taken as the initial cluster centers. We now formalize the pseudo code as follows.

Algorithm *VK*-means:

Input: $X[n][d]$, μ

Output: $C_1, C_2,\ldots,C_k$

Functions and Notations used:

S: Given set of *n* points and d dimension

Vor(*S*): It constructs the Voronoi diagram for the data set S and stores the Voronoi vertices

v_i : Voronoi vertex i =1, 2,…, 2*n*-5 (max.)

CirS(*v*): It finds the largest empty circle of vertex *v*

R(*CirS*(*v*)): It finds the radius of the Voronoi circle *CirS*(*v*).

K-means(S, S´): It runs *K*-means algorithm for the set S of ‘*n*’ points and with the set S´ of ‘*k*’ initial cluster centers.

r: a temporary variable.

μ: Threshold value on the radius of the Voronoi circle to separate the cluster centers.

```
Step 1: Given a set S of ‘n’ points, construct the Voronoi diagram Vor(S).
Step 2: Sort all the Voronoi vertices $v_i$ in ascending order with respect to the radius of their largest empty Voronoi circle’s CirS($v_i$), $i = 1,2, \ldots, 2n-5$ (max.) and store them in an array V[].
Step 3: Repeat steps 4 through 7 for $i = 1, 2,\ldots, 2n - 5$
Step 4: Assign the radius of CirS(V[i]) to ‘r’, i.e, $r = R(CirS(V[i]))$
Step 5: If $r \le \mu$ then locate all the points lying on the boundary of CirS(V[i]). If any point is already covered by a circle then ignore that point.
Else go to step 7.
Step 6: Store the vertex V[i] in a set S´.  $i=i+1$
Step 7: If $r > \mu$ then store the uncovered points (if any) in the set S´ and exit the loop.
Step 8: If  $S = S´$ then go to step 9
        Else construct the Voronoi diagram Vor(S') for  the set S' and go to step 2.
Step 9: Call K-means(S, S´) to obtain the set of clusters, say $\{C_1, C_2, \ldots, C_k\}$.
Step 10: Stop
```

Complexity Analysis

Step 1 requires $O(n \log n)$ time for the construction of the Voronoi diagram of the *n* data points. Step 2 also requires $O(n \log n)$ time for sorting. Steps 4 through 6 are repeated at most 2n – 5 time in which each of the steps 4, 5, and 6 requires constant time and thus they require $O(n)$ time in total. Step 7 requires linear time. Therefore steps 2 through 7 require $O(n \log n)$ time in total. However, steps 2 through 8 are repeated a finite number of times, say *k* times in which construction of the Voronoi diagram is the dominating computation. Step 9 requires $O(n\tau)$ to run *K*-means clustering. Therefore the overall time complexity of the proposed algorithm is $O(kn \log n)+ O(n\tau)$.

5. Experimental Results

This section establishes the practical efficiency of the proposed algorithm. We tested its performance on a number of data sets. These included both synthetically generated data and data used in real applications taken form UCI machine learning repository. The useful experimental setup to implement the proposed scheme is as follows. We have used Intel Core 2 Duo Processor machine with T9400 chipset, 2.53 GHz CPU and 2 GB RAM running on the platform Microsoft Windows Vista.

We now briefly describe the data sets taken for the experiments.

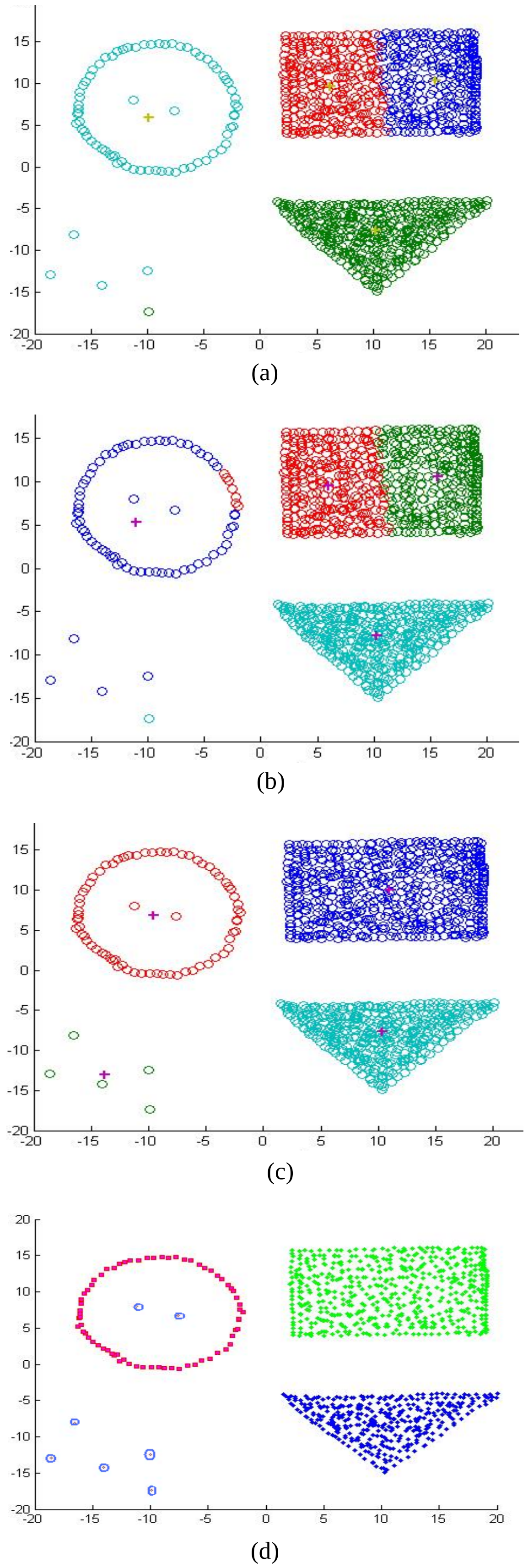
Synthetic Data

Triple-form data with outliers: There is a circular ring, a rectangle and a triangle. We insert seven outlier points. The size of this data set is 1007.
4–band data: There are four clusters in this data type where all the clusters are in the form of parallel bands. The size taken here is 600.

4-Ldata with outliers: There are four clusters in this data set and they are all of L-shaped. We insert here sixteen outlier points. Each cluster represents two perpendicular lines and size of this data set is 1216.

2-non-convex data: The two clusters of this data set are of non-convex shape with 250 points of size each.

The proposed method is applied on all these data sets and the results are compared with classical *K*-means [16], improved *K*-means [34] and Fuzzy C-means [35]. It can be observed that for the Triple-form data set, the K-means, and Fuzzy C-means are failed to obtain the desired clustering results whereas the improved K-means and our proposed algorithm are able to do so as depicted in Figs. 3(a)-3(d) in which all the points within a cluster are shown by same color. For the rest of the synthetic data sets, the K-means, Fuzzy C-means and the improved K-means all are failed to produce the desired clusters and also unable to detect the outliers. On the other hand the proposed algorithm works well on these data sets as shown in Figs. 4-6. The outliers are shown by small hollow circles in Fig. 3 and Fig. 5. It is important to note that the K-means, Fuzzy C-means and the improved K-means are unable to detect. They treat the outliers as the points of the other clusters as depicted by the same colors as the cluster points. For examples, in Fig. 3(a) all the outliers are the part of the ring cluster. Similarly, in Fig. 3(c), two outliers belong to the ring cluster and the remaining outliers belong to the triangle clusters. Whereas, our proposed algorithm successfully detects the outliers which are treated separately from the cluster points as depicted by the color different from any cluster point.



(a)

(b)

(c)

(d)

Figure 3. *Clustering results on Triple-form data of 1007 points: (a) result of K-means clustering; (b) result of Fuzzy C-means clustering; (c) result of Improved K-means clustering; (d) result of proposed algorithm.*

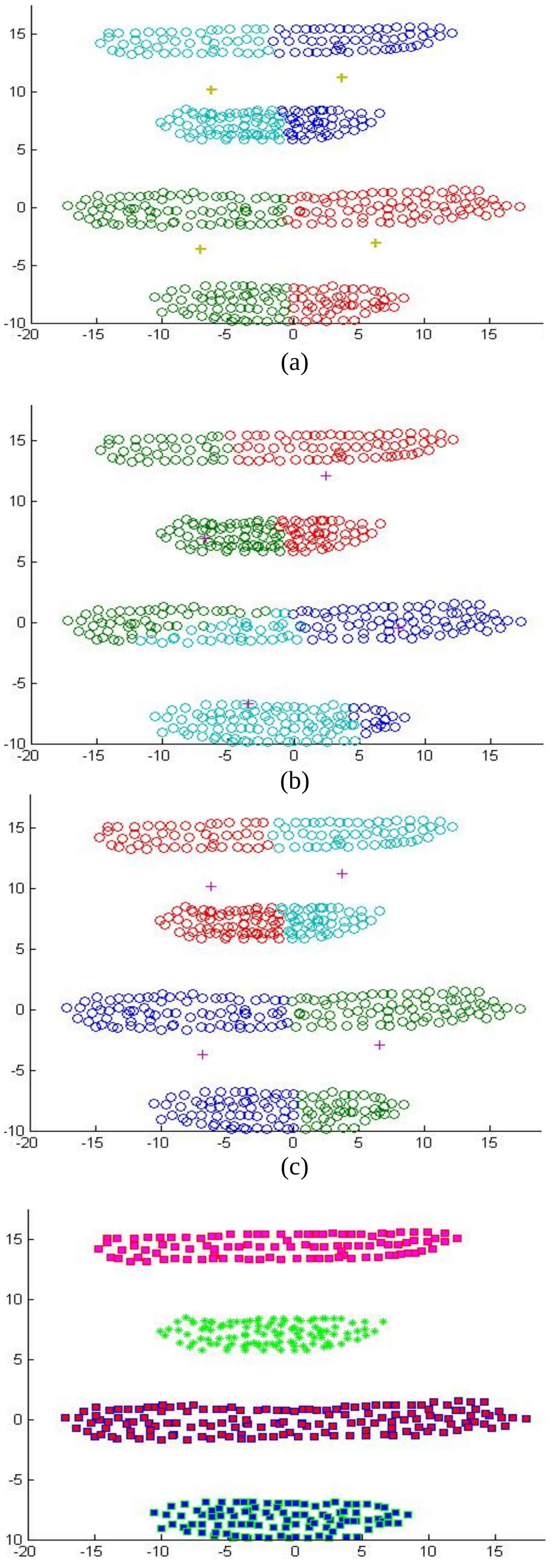


Figure 4. *Clustering results on 4-band data of 600 points: (a) result of K-means clustering; (b) result of Fuzzy C-means clustering; (c) result of Improved K-means clustering; (d) result of proposed algorithm.*

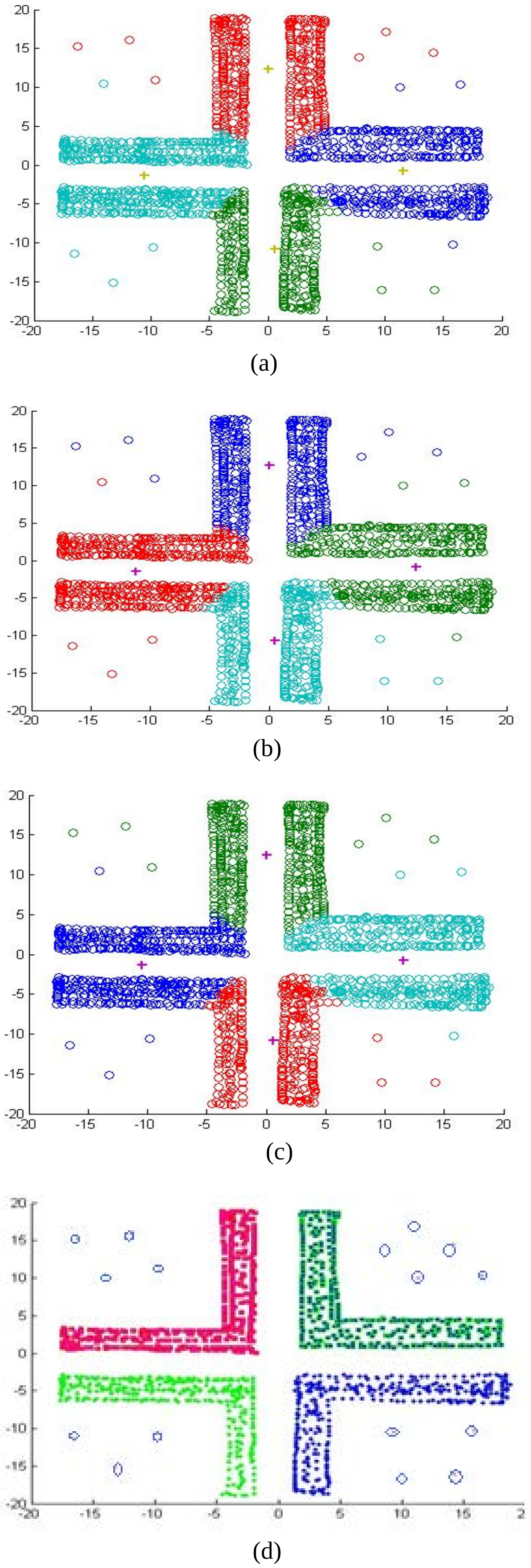
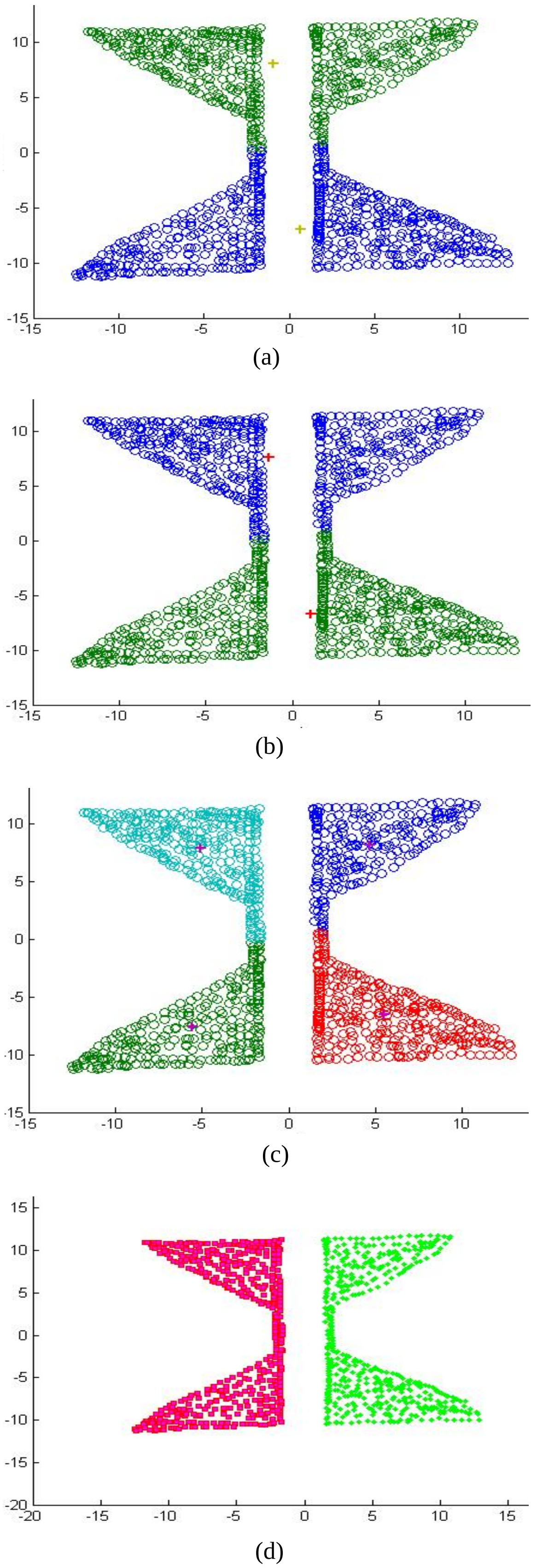


Figure 5. *Clustering results on 4-L data of 1216 points: (a) result of K-means clustering; (b) result of Fuzzy C-means clustering; (c) result of Improved K-means clustering; (d) result of proposed algorithm.*



(a)

(b)

(c)

(d)

Figure 6. *Clustering results on 2-non-convex data of 500 points: (a) result of K-means clustering; (b) result of Fuzzy C-means clustering; (c) result of Improved K-means clustering; (d) result of proposed algorithm.*

Real World Data

Here we consider few data sets from the UCI machine learning repository [36]. These data sets are multi-dimensional; hence we show the resultant clusters with the help of validity index.

Iris data: This is a famous data set used often in many clustering algorithms. It has three classes namely *Setosa*, *Versicolor* and *Virginica*. The set consists of 150 points with 50 instances for each class. Each point is described by a set of four attributes viz sepal length, sepal width, petal length and petal width. Our objective is to separate the points of different classes.

Spect heart data: This data set describes about cardiac Single Proton Emission Computed Tomography (SPECT) images. Each patient classified into two categories: normal and abnormal. There are 187 instances (SPECT image sets) taken with 22 attributes (binary feature patterns)

Wine data: This data is the result of a chemical analysis of wines to determine the origin of wines. We take the data set of 178 instances with 13 attributes and three classes. The analysis determined the quantities of 13 constituents found in each of the three types of wines. The three classes have 59, 71 and 48 instances respectively. All these classes are separable.

(Statlog) heart data: This dataset is a heart disease database similar to the Spect-Heart data set, but with small difference. Here the two classes represent the absence and presence of cost matrix. The data set taken here is of 270 points with 13 attributes each.

Pima-India-Diabetics (PID) data: This dataset donated by Vincent Sigillito, and is a collection of medical diagnostic reports of 768 examples from a population living near Phoenix, Arizona, USA. They used 576 training instances and obtained a classification of 76% on the remaining 192 instances. Here, the number of classes is two with 576 and 192 instances respectively. The number of attributes is 8.

Soybean (small) dataset: This is a Michalski's famous soybean disease database of 47 instances each of which has 35 attributes and belongs to four classes.

Breast Tissue: This is a dataset with electrical impedance measurements of freshly excised tissue samples from the breast. The number of instances taken here are 106 with 9 attributes per each. Here the numbers of classes is 2.

All these data sets are experimented by the proposed technique and the experimental results are compared with classical *K*-means and improved *K*-

means by means of dynamic validity index [15]. It is obvious to observe that the proposed scheme performs well in all the cases compared to the existing techniques *K*-means, fuzzy-c means and improved *K*-means. The comparison results are shown in table 1 (Appendix-1).

6. Conclusion

In this paper we proposed a novel *K*-means algorithm that solves the major problems faced by classical *K*-means. We solve the problem of random selection initial cluster centers with the help of Voronoi diagram. The initial cluster centers have been traced out iteratively by locating the nearest Voronoi circles of each point. Here we need not input the output number of clusters in advance as the ‘k’ initial cluster centers are automatically located with the help of threshold limit given on the radius of Voronoi circle. The experiments carried out many synthetic and multidimensional biological data sets show the efficient formation of clusters over the existing techniques.

References

[1] A. K. Jain, Algorithms for Clustering Data, New Jersey: Prentice Hall, Englewood Cliffs, 1988.

[2] Z. M. Wang, C.S. Yeng, Q. Song, and K. Sim, “Adaptive Spatial Information-theoretic Clustering for Image Segmentation”, Pattern Recognition, Vol. 42, 2009, pp. 2029-2044

[3] S. C. Madeira, and A. L. Oliveira, “Biclustering Algorithms for Biological Data Analysis: A Survey”, Computational Biology and Bioinformatics, Vol. 1, 2004, pp. 24-45.

[4] A.Y. Al-Omary, and M.S. Jamil, “A New Approach of Clustering based Machine Learning Algorithm”, Knowledge Based Systems, Vol. 19, 2006, pp. 248-258

[5] A. A. Abbasi, and M. Younis, “A Survey on Clustering Algorithms for Wireless Sensor Networks”, Computer Communications, 2007, pp.2826–2841.

[6] J. F. Lu, J. B. Tang, Z. M. Tang, and J. Y. Yang, “Hierarchical Initialization Approach for K-means Clustering”, Pattern Recognition Letters, Vol. 29, 2008, pp. 787-795.

[7] C. Fuyuan, J. Liang, and G. Jiang, “An Initialization Method for the K-means Algorithm using Neighborhood Model”,Computers and Mathematics with Applications, Vol. 58, 2009, pp. 474-483..

[8] S. Ray, and R. H. Turi, “Determination of Number of Clusters in K-means Clustering and Application in Colour Image Segmentation”, in fourth International Conference (ICAPRDT-99), Calcutta, India, 1999, pp. 137-143.

[9] A. K. Jain, “Data Clustering: 50 years beyond K-means*”, Pattern Recognition Letters, Vol. 31, 2010, pp. 651-666.

[10] S. Bandyopadhyay, and U. Maulik, “An Evolutionary Technique based on K-means Algorithm for Optimal Clustering in ℜN”, Information Science Applications, Vol. 146, 2002, pp. 221–237.

[11] Preparata FP, Shamos MI. Computational geometry-an introduction. Berlin Heidelberg, Tokyo: Springer-Verlag; 1985.

[12] Bishnu, P.S. and Bhattacherjee, V. (2009) ‘CTVN: Clustering technique using Voronoi diagram’, Journal of Recent Trends in Engineering, Vol. 2, No. 3, pp. 13-15.

[13] Koivistoinen, H., Ruuska, M. and Elomaa, T. (2006) ‘A Voronoi diagram approach to autonomous clustering’. Paper Presented at the International Conference. Discovery Science. Springer, Berlin. 2006. Spain.

[14] S. Bandyopadhyay, and U. Maulik, “An Evolutionary Technique based on K-means Algorithm for Optimal Clustering in ℜN”, Information Science Applications, Vol. 146, 2002, pp. 221–237.

[15] J. Shen, S.I. Chang, E.S. Lee, Y. Deng, S.J. Brown, Determination of cluster number in clustering microarray data, J. App. Math. and Comp.169 (2005) 1172-1185.

[16] J. B. MacQueen, “Some Methods for Classification and Analysis of Multivariate Observation”, in Berkeley symposium on Mathematical Statistics and Probability, University of California Press. 1967, pp. 281-297.

[17] R. O. Duda, and P. E. Hart, Pattern Classification and Scene Analysis, New York: John Wiley and Sons, 1973.

[18] D. Fisher, “Iterative Optimization and Simplification of Hierarchical Clusterings”, Artificial Intelligence Research, Vol. 4, 1996, pp. 147-179.

[19] R. E. Higgs, K. G. Bemis, I. A. Watson, and J. H. Wikel, “Experimental Designs for Selecting Molecules from Large Chemical Databases”, Chemical Information and Computer Sciences, Vol. 37, 1997, pp. 861-870.

[20] M. Snarey, N. K. Terrett, P. Willet, and D. J. Wilton, “Comparison of Algorithms for Dissimilarity-based Compound Selection”, Molecular Graphics and Modeling, Vol. 15, 1997, pp. 372-385.

[21] P. S. Bradley, O. L. Mangasarian, and W. N. Street, “Clustering via Concave Minimization,

in: M.C. Mozer, M.I. Jordan, and T. Petsche (Eds.).” Advances in Neural Information Processing System, Vol. 9, 1997, pp.368-374.

[22] S. S. Khan, and A. Ahmad, “Cluster Center Initialization Algorithm for K-means Clustering”, Patter Recognition Letters, Vol. 25, 2004, pp. 1293-1302.

[23] J. Tou, and R. Gonzales, Pattern Recognition Principles, MA: Addison Wesley, 1974.

[24] Y. Linde, A. Buzo, and R. M. Gray, “An Algorithm for Vector Quantizer Design”, IEEE Trans. on Commun, Vol. 28, 1980, pp. 84–95.

[25] L. Kaufman, and P. J. Rousseeuw, Finding Groups in Data. An Introduction to Cluster Analysis, Canada: Wiley, 1990.

[26] G. P. Babu, and M. N. Murty, “A Near-optimal Initial Seed Value Selection in K-means Algorithm using a Genetic Algorithm”, Pattern Recognition Letters, Vol. 14, No. 10, 1993, pp. 763–769.

[27] C. Huang, and R. Harris, “A Comparison of Several Codebook Generation Approaches”, IEEE Trans. on Image Process, Vol. 2, No. 1, 1993, pp. 108–112.

[28] B. Thiesson, B. Meck, C. Chickering, and D. Heckerman, “Learning mixtures of Bayesian Networks”, Microsoft Technical Report (TR-97-30), 1997.

[29] P. S. Bradley, and U. M. Fayyad, “Refining Initial Points for K-means Clustering”, in Fifteenth International Conference on Machine Learning, 1998,pp. 91–99.

[30] H. Yan and R. Weibel. An algorithm for point cluster generalization based on the Voronoi diagram. Journal of Computers & Geosciences, 34(8):939-954, 2008.

[31] P.K. Jana M.P. Misra and A. Raj. A Voronoi diagram based clustering algorithm. International Journal of Advanced Computer Engineering, 3(2):79-84, 2010.

[32] D. Jiang J. Pei and A. Zhang. DHC: A Density-based hierarchical clustering method for time series gene expression data. In 3 rd IEEE Symposium on Bioinformatics and Bioengineering (BIBE-2003), Bethesda, USA, pages 1 -8, March 10-12, 2003.

[33] B. Kao S.D. Lee F.K.F. Lee D.W. Cheung and W.S. Ho. Clustering uncertain data using Voronoi diagrams and R-tree index. IEEE Transactions on Knowledge and Data Engineering, 22(9):1219-1233, 2010.

[34] F. Geraci M. Leoncini M. Montengaro M. Pellegrini and M.E. Renda. FPFSB: A scalable algorithm for microarray gene expression data clustering. In International Conference on Digital Human Modeling (ICDHM-07), China, volume 4561, pages 606-615, July 22-27, 2007.

[35] J.C. Bezdek R. Ehrlich and W. Full. FCM: the fuzzy c-means clustering algorithm. Computers & Geosciences, 10(2-3):191-203, 1984.

[36] UCI Machine Learning Repository, http://archive.ics.uci.edu/ ml/ datasets.html.

Appendix 1:

Table 1: Comparison chart of the proposed scheme with *K*-means, Fuzzy *C*-means and improved *K*-means using Intra-Inter ratio validity index.

Name	No of Attributes	Data Size	Cluster No.	Val_index (*K*-means)	Val_index (Fuzzy *C*-means)	Val_index (Improved *K*-means)	Val_index (Proposed algorithm)
Iris	4	150	3	0.2930	0.3223	0.2888	0.0836
S. Heart	22	187	2	2.9122	5.8464	2.3846	0.5101
Wine	13	178	3	0.1895	0.1766	0.1895	0.1153
S. log (heart)	13	270	2	0.2632	0.2991	0.2611	0.0323
P.-India-Dia.	8	768	2	0.1549	0.1828	0.1549	0.0344
Soyabin	35	47	4	0.6324	1.9656	0.7010	0.2843
Breast Tissue	9	106	2	0.0522	0.1111	0.0057	0.0052