Initialization for K-means Clustering using Voronoi Diagram

Article · December 2012

DOI: 10.1016/j.protcy.2012.05.061

Prasanta K. Jana
Indian Institute of Technology (ISM) Dhanbad
141 PUBLICATIONS **1,171** CITATIONS

Available online at www.sciencedirect.com

SciVerse ScienceDirect

Procedia Technology 4 (2012) 395 – 400

Procedia Technology

C3IT-2012

Initialization for *K*-means clustering using Voronoi diagram

Damodar Reddy[a] and Prasanta K. Jana[a] , *IEEE Senior Member*

[a]*Department of Computer Science & Engineering*
Indian School of Mines, Dhanbad 826 004, India

Abstract

K-Means algorithm is one of the famous partitioning clustering techniques that has been studied extensively. However, the major problem with this method that it cannot ensure the global optimum results due to the random selection of initial cluster centers. In this paper, we present a novel method that selects the initial cluster centers with the help of Voronoi diagram constructed from the given set of data points. The initial cluster centers are effectively selected from those points which lie on the boundary of higher radius Voronoi circles. As a result, the proposed method automates the selection of the initial cluster centers to supply them for *K*-means. The proposed method is experimented on various artificial (hand-made) as well as real world data sets of various dimensions. It is observed that it is able to produce better clustering results than the traditional *K*-means and the improved *K*-means.

Keywords: Clustering; *K*-means; improved *K*-means; Voronoi diagram; error rate

1. Introduction and background

Clustering is one of the efficient data mining techniques to classify the intrinsic structure that lies behind the given data objects (points) [1]. It refers to the process of organizing the given objects into homogeneous classes called clusters whose members are similar to each other. Clustering has been applied in a wide variety of fields. For instance wireless sensor networks [2], economic science [3], medicine [4] etc. Based on various characteristics, clustering mainly divided into two models, namely, partitioning algorithms [1] and hierarchical algorithms [1]. Partitioning clustering algorithms have widely been applied because of its effectiveness and applicability for the large data sets. It mainly involves in partitioning the given data set into a number of disjoint clusters. *K*-means [1] is the most popular one among all the partitioning clustering techniques. Although it is very simple and robust in clustering large data sets, the method suffers from a few draw backs. The user needs to provide the number of clusters which is difficult to know in advance for many real world data sets. But the major problem it suffers that it is very sensitive for the selection of initial cluster centers. As a result, it cannot always produce global optimum results. A number of attempts [5], [6], [7] have been made to resolve this problem. In this paper,

2212-0173 © 2012 Published by Elsevier Ltd.
doi:10.1016/j.protcy.2012.05.061

we propose a new method to solve the same problem. However, our method is based on the Voronoi diagram [8] constructed from the given data set. We use the points lying on the higher radius Voronoi circles to find the initial cluster centers. We experiment our method on various synthetic and real world data sets from UCI machine learning repository [12] and observe that it is able to produce the results better than the *K*-means [1] and improved *K*-means [9] algorithms. The rest of the paper is organized as follows. The useful preliminaries are discussed in section 2. The proposed method is given in section 3. The experimental results are described in section 4 followed by the conclusion in sections 5.

2. Preliminaries

2.1. K-means clustering

The *K*-means algorithm [10] has the following steps:

Step 1: Select k initial cluster centers $c_1, c_2, \ldots, c_k$ randomly from the given n points $\{x_1, x_2, \ldots, x_n\}$, $k \leq n$.

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 $\left\| x_i - c_j \right\| \leq \left\| x_i - c_p \right\|$, $p = 1, 2, \ldots, k$ and $j \neq p$.

Step 3: Compute new cluster centers $c_1^*, c_2^*, \ldots, c_k^*$ as follows $c_i^* = \frac{1}{n_i} \sum_{x_j \in C_i} x_j$ for $i = 1, 2, \ldots, k$.

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

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

It is clear from the step 1 that the clustering results depend on the initial cluster centers which are chosen randomly.

2.2 Voronoi diagram

Given a set of n points $S = \{p_1, p_2, \ldots, p_n\}$ in a m-dimensional Euclidean space, the Voronoi diagram [8] of S is defined as the subdivision of the space into n cells such that each point belongs to only one cell. We denote the Voronoi diagram of S as $Vor(S)$. Let $d(a, b)$ denote the distance between the points a and b in this space. Then the definition implies that 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 S and $j \neq i$. The Voronoi diagram of twenty three points is shown in Fig. 1 (a) as an example. For a Voronoi vertex v, we define the largest empty circle of v (see Fig. 1(b)) with respect to S, as the largest circle with v as its center that contains no point of S in its interior. We denote it 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. There are a maximum of $2n-5$ Voronoi vertices in a Voronoi diagram of n points.

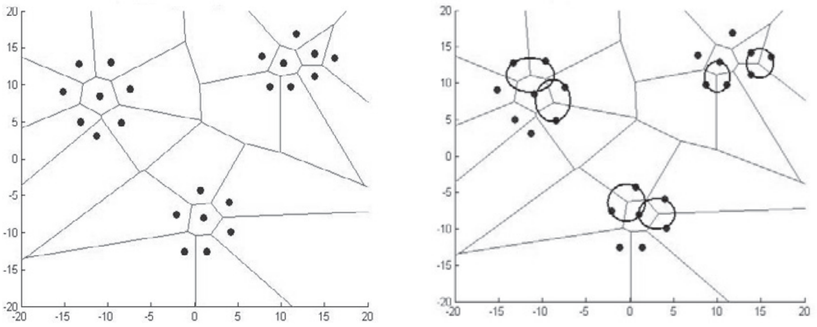

Fig. 1. (a) Voronoi diagram of 23 points; (b) Voronoi circles (all circles are not shown)

2.3 Error rate

We evaluate the quality of the proposed algorithm using error rate (*ER*) proposed in [11]. The error rate (*ER*) is measured by the percentage of total number of objects over the number of misclassified objects. It is given by

$$ER = \frac{Number\ of\ misclassified\ objects}{Total\ number\ of\ objects} \times 100\% \qquad (2.1)$$

It can be noted that lesser the error rate, the better will be the clustering results.

3. Proposed Initialization Method for *K*-means

The proposed method has the following two phases 1) Formation of initial cluster centers 2) Application of the *K*-means algorithm with these initial cluster centers. The basic idea is as follows. Given the set of *n* data points *S* and the required number of clusters *k*, the Voronoi diagram *Vor*(*S*) is constructed first. All the vertices v_i (i = 1, 2,…, 2n−5 maximum) of *Vor*(*S*) are then located. We next sort these vertices in non-increasing order of their radius of largest empty Voronoi circles *CirS*(v_i) and store them in *sorted_vertex*[*i*]. We maintain two arrays namely, *Test*[] and *ccenter*[] which are initially empty. *Test*[] is to used to store all the points on the circumference of Voronoi circle *CirS*(*v*) and *ccenter*[] holds the partial or final initial cluster centers. We start the iteration with the largest radius Voronoi circle, i.e., *CirS*(*sorted_vertex*[0]). In every iteration *i* (*i* = 1, 2,…, 2*n*−5 maximum), we perform the following three steps to form the initial cluster centers. 1) Store all the points on the boundary of *CirS*(*sorted_vertex*[i]) in *Test*[]. If the distance between any two points say P_1 and P_2 of *Test*[] is less than the radius of *CirS*(*sorted_vertex*[i]), remove any one of them from *Test*[]. 2) Check the distance between every point of *Test*[] and all the points of *ccenter*[]. If the distance between any two points of *Test*[] and *ccenter*[] is less than the radius of *CirS*(*sorted_vertex*[*i*] then also remove the point from *Test*[]. 3) Store all the remaining points of *Test*[] into *ccenter*[]. At the end of the final iteration, the set of initial cluster centers are stored in *ccenter*[] which is then provided to the *K*-means clustering algorithm. It is obvious to note that at some stage of the proposed method any one point belongs to the outlier region is assigned as an initial cluster center. This selection separates the outlier region form the other clusters during the process of *K*-means clustering. Once all such points are located in the form of separate clusters, we declare them as outliers based on the cardinality. Here, we use a limit on the cardinality of clusters for outlier detection. The pseudo code of the algorithm is as follows.

Algorithm Voronoi_*K*-means(*S*, *k*)

Input: A set S of n data points and the number of clusters k; ***Output***: The set C_i of clusters $i = 1, 2, \ldots, k$;
Functions and variables used:
VD(*S*): A function to construct the Voronoi diagram for the given data set S
$r(v)$: A function when called finds the radius of the Voronoi circle *CirS*(v)
sort(v_i): A function to sort the vertices v_i $i = 1, 2, \ldots, 2n{-}5$ in non-increasing order of the radiuses *CirS*(v_i)
sorted_vertex[i]: An array to store all the sorted vertices v_i $i = 1, 2, \ldots, 2n{-}5$
circumpoints(v): A function to find the points on the circumference of *CirS*(v)
KM(S, k): A function to return a set of k clusters for the given set S of n points using K-means clustering
$d(p, q)$: A function to calculate the Euclidean distance between the points p and q
ccenter[]: An array to store the already assigned cluster centers /* initially empty */
Test[]: An array to store the partially located cluster centers /* initially empty */
Temp[]: A temporary array /* initially empty */, $i \leftarrow$ a temporary variable /* initially zero*/

```
Step 1: Call VD(S)
Step 2: Find the radius of $v_i$ where $i = 1, 2, \ldots, 2n-5$ (max) using $r(v_i)$
Step 3: Call sort($v_i$) to sort all the vertices $v_i$ and store them in sorted_vertex[$i$] where $i = 1, 2, \ldots, 2n-5$
Step 4: $i \leftarrow 0$;
Step 5: Call circumpoints(sorted_vertex[$i$]) to find the points say, $P_i$ for some $i = 1, 2, \ldots, m$. where $m \geq 3$
            on the circumference of $(i+1)^{th}$ large Voronoi circle and store them in Test
            i.e., $Test \leftarrow Test \cup \{P_i\}$ $i = 1, 2, \ldots, m$
Step 6: Call $d(P_j, P_k)$ $\forall$ $P_j, P_k$     Test and $j \neq k$
Step 7: If $d(P_j, P_k) < r(CirS(sorted\_vertex[i]))$ for any $j$, $k$ then ignore either $P_j$ or $P_k$
            i.e., $Test \leftarrow Test - \{P_j$ or $P_k\}$
Step 8: If ccenter =     then {   ccenter ← Test;
                                  if (|ccenter| == k) then go to step 12;
                                  else{
                                        $i \leftarrow i + 1$;
                                        Test ←    ;
                                        go to step 5
                                      }
                                }
Step 9: For $j$ = 1 to |ccenter|
          For $k$ = 1 to |Test|
          {
        Call $d(c_j, P_k)$ where $c_j$     ccenter and $P_k$     Test
        If $d(c_j, P_k) < r(CirS(sorted\_vertex[i]))$ for any $j$, $k$ then store $P_k$ in Temp, i.e. $Temp \leftarrow Temp \cup \{P_k\}$;
          }
Step 10: ccenter ← ccenter ∪ Test – Temp;
Step 11: If (|ccenter| == k) then go to step 12;
            Else {
                  $i \leftarrow i + 1$;
                  Test && Temp←    ;
                  go to step 5
                 }
Step 12: Call KM(S, k) with the $k$ initial cluster centers $c_1, \ldots, c_k$ stored in the set ccenter
```

For the time complexity we proceed as follows. The Voronoi diagram is constructed in $O(n \log n)$ time. For sorting all the $2n$-5 (maximum) Voronoi vertices, it is needed $O(n \log n)$ time. K-means runs in $O(kn\tau)$ time which dominates the other computation time, where τ is the number of iterations. Hence the overall complexity of the proposed method is $O(kn\tau)$.

4. Experimental Design

In order to evaluate the performance of the proposed method we ran it on several artificial and real world data sets from UCI machine learning repository [12] and compared the results with that of the traditional *K*-means and improved *K*-means. We initially show the results on 3 synthetic data sets. The algorithm is applied on a five-cluster data set where three clusters are of very small size to represent the outliers. The Voronoi-based *K*-means produced two clusters shown in pink and green colors in the Fig. 2 (c). The outliers are also located by means of separate clusters which can be seen from the same Figure. But the same result is not obtained by any of *K*-means and improved *K*-means as shown in Figs. 2(a-b). Similarly, the proposed method works well in case of the ring-curve-parallelogram and four cluster data over the existing techniques as shown in Figs. 3(a-c) and 4(a-c). We show the results of the proposed method for real world data using the error rate defined in section 2.3. The comparison of the proposed method with *K*-means and improved *K*-means is shown in Table 1.

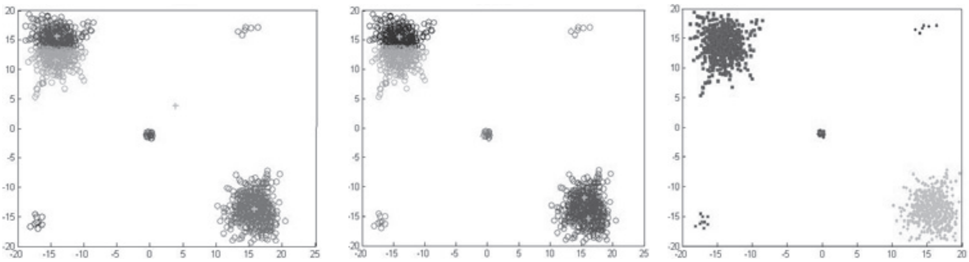

Fig. 2. Results on five-cluster data with outliers (a) *K*-means; (b) improved *K*-means; (c) proposed

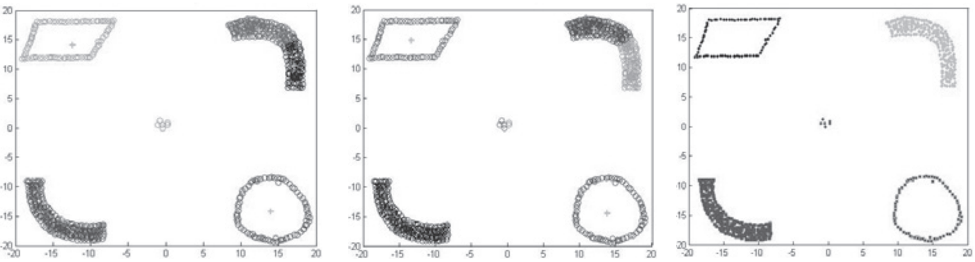

Fig. 3. Results on ring-curve-parallelogram data with outliers (a) *K*-means; (b) improved *K*-means; (c) proposed

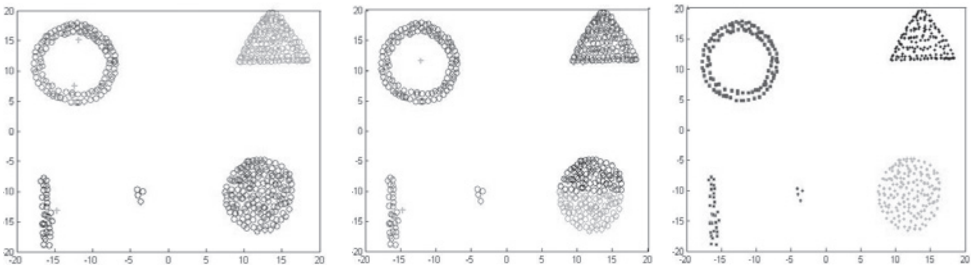

Fig. 4. Results on four-cluster data with outliers (a) *K*-means; (b) improved *K*-means; (c) proposed

Table 1. Comparison of the proposed scheme with *K*-means and improved *K*-means using *Error Rate* (***ER***).

Data set	**Total no. of**	**Misclassified patterns**			***ER***		
	Patterns	*K*-means	improved *K*-means	Ours	*K*-means	improved *K*-means	Ours
Iris	150	22	20	17	14.67	13.33	11.33
S. Heart	187	14	13	14	7.49	6.95	7.49
Wine	178	23	19	17	12.92	10.67	9.56
Ecoli	336	65	54	45	19.34	16.07	13.39
St. Heart	270	44	41	30	16.23	15.19	11.11
P.I.Diabetes	768	193	153	109	25.13	19.92	14.19
Soybean	47	11	8	8	23.40	17.02	17.02
B. Tissue	106	23	19	16	21.69	17.92	15.09

5. Conclusion

A novel initialization method has been proposed for *K*-means with the help of Voronoi diagram. The proposed method is experimented on various synthetic and real world data sets. The comparison results shows that the proposed initialization method produce better results than the traditional *K*-means as well as the improved *K*-means. In our future work, we attempt to devise a method to automate the number of clusters so that it will be a total package of a parameterless clustering algorithm.

Acknowledgements

We sincerely thank the Council of Scientific & Industrial Research (CSIR), New Delhi, for supporting this work under the grant (No. 25(0177)/ 09/EMR-II).

References

1. Jain AK, Dubes RC. *Algorithms for clustering data*. New Jersey: Prentice Hall; 1988.
2. Liu Z, Zheng Q, Xue L, Guan X. A distributed energy-efficient clustering algorithm with improved coverage in wireless sensor networks. *J. Future Generation Computer Systems* 2011.
3. Garibaldi U, Costantini D, Donadio S, Viarengo P. Herding and clustering in economics: the Yule-Zipf-Simon model. *J. Computational Economics* (*Springer*) 2006;**27**:115-134.
4. Villmann T, Albani C. Clustering of categoric data in medicine-application of evolutionary algorithms. *International Conference 7th Fuzzy Days on Computational Intelligence*, *Theory and Applications* 2001; 619-627.
5. Al-Daoud MB, Roberts SA. New methods for the Initialization of clusters. *J Pattern Recognition Letters* 1996;**7**:451-455.
6. Lu JF, Tang JB, Tang ZM, Yang JY. Hierarchical initialization approach for *K*-means clustering. *J Pattern Recognition Letters* 2008;**29**:787-795.
7. Fuyuan C, Liang J, Jiang G. An initialization method for the *K*-means algorithm using neighborhood model. *J Computers and Mathematics with Applications* 2009;**58**:474-483.
8. Preparata FP, Shamos MI. *Computational geometry-an introduction*. Berlin Heidelberg, Tokyo: Springer-Verlag; 1985.
9. Geraci F, Leoncini M, Montengaro M, Pellegrini M, Renda ME. FPF-SB: A scalable algorithm for microarray gene expression data clustering. *International Conference on Digital Human Modelling*, *ICDHM*-07 2007; 606-615.
10. Bandyopadhyay S, Maulik U. An evolutionary technique based on *K*-means algorithm for optimal clustering in $\Re^N$. *J Information Science Applications* 2002;**146**:221-237.
11. Khan SS, Ahmad A. Cluster center initialization algorithm for *K*-means clustering. *J Pattern Recognition Letters* 2004;**25**:1293-1302.
12. UCI Machine Learning Repository, http://archive.ics.uci.edu/ml/datasets.html.