International Journal of Information, Control and Computer Sciences
ISSN: 2517-9942
Vol:11, No:1, 2017

A Minimum Spanning Tree-Based Method for Initializing the K-Means Clustering Algorithm

J. Yang, Y. Ma, X. Zhang, S. Li, Y. Zhang

***Abstract*—**The traditional k-means algorithm has been widely used as a simple and efficient clustering method. However, the algorithm often converges to local minima for the reason that it is sensitive to the initial cluster centers. In this paper, an algorithm for selecting initial cluster centers on the basis of minimum spanning tree (MST) is presented. The set of vertices in MST with same degree are regarded as a whole which is used to find the skeleton data points. Furthermore, a distance measure between the skeleton data points with consideration of degree and Euclidean distance is presented. Finally, MST-based initialization method for the k-means algorithm is presented, and the corresponding time complexity is analyzed as well. The presented algorithm is tested on five data sets from the UCI Machine Learning Repository. The experimental results illustrate the effectiveness of the presented algorithm compared to three existing initialization methods.

***Keywords*—**Degree, initial cluster center, k-means, minimum spanning tree.

I. Introduction

THE goal of clustering is to partition data points into clusters according to the similarity between data points to maximize the similarity between the data points in the same cluster while minimizing the similarity between the data points in different clusters [1], [2]. Clustering algorithms can be broadly classified into hierarchical and nonhierarchical clustering algorithms [3]–[5]. The k-means algorithm's simplicity and efficiency render it the leading nonhierarchical clustering algorithm in various fields [6]. However, the k-means algorithm is especially sensitive to initial cluster centers, thus, after a bad initialization it easily gets trapped in poor local minima. To solve this problem, numerous improved methods have been presented. MacQueen took the first k data points from the data set as the centers [7]. An obvious drawback of this method is its sensitivity to data ordering. Gonzalez proposed the maximin method, which selects the data point that has the greatest minimum distance to the previously selected centers [8]. However, this method selects the first center arbitrarily, which leads to unstable clustering results. Khan [9] proposed Cluster Center Initialization Algorithm (CCIA) to solve cluster initialization problem. It initiates with calculating the mean and the standard deviation for data attributes, and then separates the data with a normal curve into a certain partition. The experimental results demonstrate the effectiveness and robustness of CCIA for several clustering problems. However, the time complexity of CCIA increases with the increase of the dimensionality of the data set. Redmond et al. [10] first constructs a kd-tree of the data points to perform density estimation and then uses a modified maximin method to select K centers from densely populated leaf buckets. Yet kd-tree is known to scale poorly with the dimensionality of the dataset. Like maximin method, David Arthur et al. [11] proposed k-means++ method which aims to avoid the unlikely event of choosing two centers that are close to each other. However, this method selects the first center arbitrarily, which leads to unstable clustering results. Cao et al. [12] selected the point with maximum density as the first initial cluster center. However, the point also may be a boundary point among clusters. Reddy [13] proposed an MST-based cluster initialization for k-means which bridges the k-means and the MST-based clustering algorithms. Huang et al. [14] used the Kruskal algorithm to generate the MST of all data points and then deletes k-1 edges according to the order of their weights. In summary, selecting proper initial cluster centers is an NP problem, and numerous improved methods have not yet been widely applied [15]. Therefore, the selection of initial cluster centers requires further research.

MST is a useful graph for detecting clusters of a given set of data points. MST has been well suited for clustering in the fields of pattern recognition, image processing, and computational biology. The MST-based clustering algorithm was initiated by Zahn [16]. In this paper, the MST on the given data set is constructed using prim algorithm. Then, the concept of skeleton point based on the MST is introduced. Furthermore, for the distance between skeleton points, a novel definition of distance metric is introduced instead of the traditional Euclidean distance. Finally, the initialization method based on MST is proposed to compute initial cluster centers for the k-means algorithm. The proposed algorithm is applied to five data sets with different dimensions to compute the initial cluster centers for the k-means algorithm. Compared with CCIA, kd-tree and k-means++ methods, the proposed algorithm demonstrates superior clustering performance.

II. MST-Based Initialization Method

A. Construction for Data Set

Let X be a data set with K clusters and n data points; that is, $X = \{x_i \mid x_i \in R^p, i = 1, 2, \ldots, n\}$. To apply the MST to the initialization problem, data set X should be represented by the undirected complete weighted graph $G = (V, E)$, where

Yan Ma, Jie Yang, Xiangfen Zhang and Yuping Zhang are with the College of Information, Mechanical and Electrical Engineering, Shanghai Normal University, Shanghai 200234, China (phone: 86-21-64322341, e-mail: ma-yan@shnu.edu.cn, 1372679490@qq.com, xiangfen@shnu.edu.cn, yp_zhang@shnu.edu.cn).

Shunbao Li is with the College of Mathematics & Science, Shanghai Normal University, Shanghai 200234, China (e-mail: lsb@shnu.edu.cn).

$V = \{v_1, v_2, \ldots, v_n\}$, $|E| = \dfrac{n(n-1)}{2}$. Each data point x_i in data set X corresponds to a vertex $v_i \in V$ in graph G, and there is a one-to-one correspondence between data point x_i (i = 1, 2,..., n) and vertex v_i (i = 1, 2,..., n). The number of vertices in graph G is equal to the number of data points x_i in data set X. Edge weights between any two vertices are the Euclidean distance between the corresponding two data points.

The MST of G can be generated by using the prim algorithm which can be described as performing the following steps:

Step 1. Pick any vertex v_i from the graph G to be the root of the tree.

Step 2. Grow the tree by one edge: of the edges that connect the tree to vertices not yet in the tree, find the minimum-weight edge from G and transfer it to the tree.

Step 3. Repeat Step 2 (until the tree contains all vertices in the graph G).

B. The Proposed Algorithm

Let $T = (V, E_T)$ be a MST of $G = (V, E)$, where $V = \{v_1, v_2, \ldots, v_n\}$, $E_T = \{e_1, e_2, \ldots, e_{n-1}\}$, $e_i \in E(G)$.

Definition 1. (Number of adjacent data points) Let U_i be the set of vertices of T with degree i and W_i be the complementary set of U_i, that is, $W_i = V \setminus U_i$. For U_i, the number of adjacent data points, denoted as f_i, is the number of vertex in W_i being adjacent to vertex in U_i. Note that, only add 1 to f_i under the circumstance of one vertex in W_i being adjacent to more than one vertices in U_i.

Lemma 1. If any one vertex in W_1 is adjacent to one and only one vertex in U_1, then $f_1 = |U_1|$, otherwise $f_1 < |U_1|$.

The proof of this lemma is obvious.

Definition 2. Skeleton point) Suppose the maximum degree of T be m, then, $V = U_1 \cup U_2 \cup \ldots \cup U_m$. Let $F = \arg\max_i (f_i)$. The skeleton point, denoted as s_i, is the vertex of T with degree being greater than or equal to F.

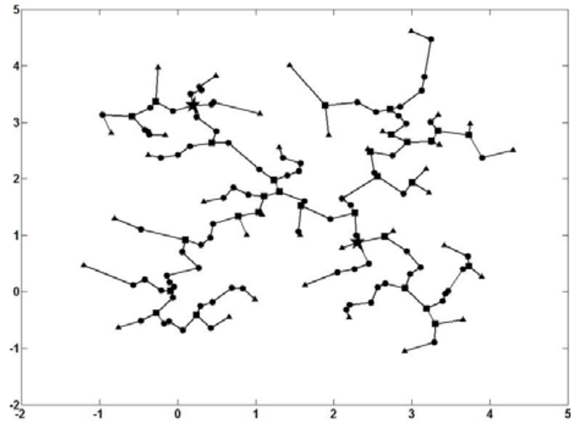

Fig. 1 MST with 150 vertices

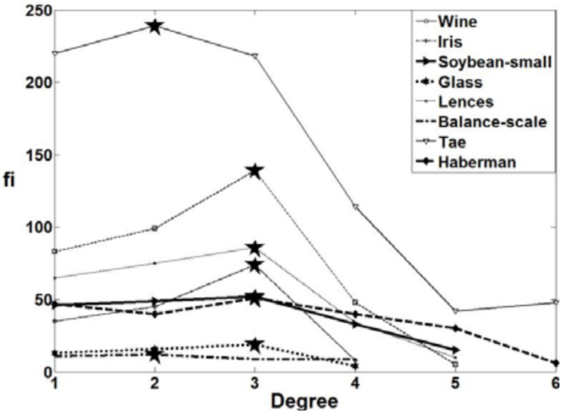

Fig. 2 Degree vs. f_i

The concept of the skeleton point is further explained in Fig. 1. As shown in Fig. 1, a dataset with 150 data points is randomly generated and constructed the corresponding MST $T = (V, E_T)$. The vertices with degree 1, 2, 3, 4 are marked by triangle, circle, rectangle and star, respectively. The maximum degree of T is 4. And the numbers of adjacent data points f1, f2, f3, f4 are 33, 51, 60, 8, respectively. Correspondingly, F=3. Therefore, the vertices of T with degree 3 or 4 are skeleton points marked by rectangle and star in Fig. 1. It can be seen from Fig. 1 that, the selected skeleton points, which represent the basic skeleton of the data set, can be seen as the candidates of the initial cluster centers. In contrast, the unselected data points correspond to the trivial detail of the data set. For instance, the data points with degree 1 are the leaves of T.

The skeleton points are chosen as the candidates of the initial cluster centers. Too many skeleton points will increase the time complexity of the presented algorithm, whereas a few skeleton points will influence clustering performance. Therefore, the number of skeleton points will influence both of running time and the clustering performance. Suppose two extreme cases, F=1 or F=m. As for F=1, the number of skeleton points is equal to the number of data points, which leads to too many candidates. As for F=m, the data points with maximum degree are chosen as skeleton points, which leads to a few skeleton points. However, the above two cases rarely happen. To illustrate, eight data sets are selected and their corresponding $f_1, f_2, \ldots, f_m$ are calculated, as shown in Fig. 2, in which the result of F is marked by star. Fig. 2 shows that in eight data sets, the values of F for six data sets are 3 and that for the other two data sets are 2. In summary, the value of F for eight data sets do not belong to one of the above two case.

Definition 3. (Distance measure) Let the Euclidean distance between two skeleton points s_i and s_j be $d(s_i, s_j)$. Let the degree of s_i and s_j be d_i and d_j, respectively. The distance measure between s_i and s_j is defined as $h(s_i, s_j) = (d_i + d_j) * d(s_i, s_j)$. Correspondingly, the set of distance is denoted as

$H = \{h(s_1, s_2), h(s_1, s_3), \ldots, h(s_1, s_m), h(r_2, r_3), \ldots, h(r_m, r_{m\text{-}1})\}$, where *m* is the number of skeleton points.

In the process of determining initial cluster centers for k-means, the main purpose of various initialization methods is to make the Euclidean distance between initial centers large enough, which avoids selecting data points among the same cluster as the cluster centers. Yet such methods will lead to the result that outliers may be selected as cluster centers; the main reason for this lies in that the factor of the density is not taken into consideration. To tackle this issue, a new distance measure in Definition 3 is defined, in which two factors of Euclidean distance and degree are considered in the calculation of the distance between two data points. Here, the concept of degree in graph *G* is similar to the density. For instance, given three skeleton points s_i , s_j and s_k , their corresponding degrees be d_i , d_j and d_k , which satisfies $d_j > d_k$, $d(s_i, s_j) = d(s_i, s_k)$. Since $d(s_i, s_j) = d(s_i, s_k)$, s_i 's distance to s_j and s_k is equal according to the traditional distance measure. Whereas, if the distance is calculated according to Definition 3, since $d_j > d_k$, it follows that $h(s_i, s_j) > h(s_i, s_k)$, which further improves the discriminative of distance.

Algorithm 1. Algorithm for determining the initial cluster centers.

```
Step 1. Input the data set $X = \{x_1, x_2, ..., x_n\}$ with $K$ clusters in
        which the initial cluster center need to be determined.
Step 2. Generate MST for data set $X = \{x_1, x_2, ..., x_n\}$ using
        prim algorithm.
Step 3. Calculate the skeleton point $s_i$ according to Definition
        2 and further constitutes the set of skeleton points
        $S = \{s_1, s_2, \ldots, s_m\}$, where $m$ is the number of skeleton
        points.
Step 4. Calculate the distance between any two skeleton points
        according to Definition 3 and further constitutes the set
        of distance $H$.
Step 5. Select the skeleton point $s_i$ with the highest degree
        from $S$ as the first initial cluster center. Denote the set of
        initial cluster centers as $C = \{s_i\}$ .
Step 6. Select the rest skeleton point $s_i$ from $S$ satisfying,
        $\max_{s_i \in s}(\min_{c_i \in c}(h(s_i, c_i)))$ , denotes the set of initial
        cluster centers as $C=\{r_i\} \cup C$. This step is repeated until
        the number of initial cluster centers is equal to $K$.
```

C. Time Complexity Analysis

The time complexity of the initialization method is analyzed as follows. In Step 2, the time complexity for generating MST through prim algorithm is $O(n^2)$. In Step 3, computation of the set of skeleton points *S* is $O(n)$. In Step 4, the time complexity for calculating the set of distance *H* is $O(\frac{m(m-1)}{2}) = O(m^2)$. In step 5 and step 6, the time complexity for determining the initial cluster centers is $O(km)$. Because $k \ll m$, $m < n$, the entire complexity of the presented algorithm is $O(n^2)$. This result illustrates that the time complexity of the presented algorithm is proportional to the square of *n*.

III. Experiments and Results

The proposed algorithm is evaluated on the Wine, Soybean-small, Iris, Glass and Haberman from the University of California at Irvine (UCI), as shown in Table I. The clustering results derived from the k-means algorithm are compared by using four initialization methods: The proposed algorithm, CCIA, kd-tree and k-means++. To test the effectiveness of the proposed algorithm, the following five evaluation indexes were adopted to obtain the clustering validity index (CVI): accuracy (AC), adjusted Rand index (ARI), Rand index (RI), Mirkin metric Index (MI), and Hurber's Γ index (HI). For all indexes except MI, higher CVI values indicate more favorable clustering; the opposite is true for MI. The experimental results are summarized in Tables II-VI.

TABLE I
Description of the Five Data Sets

Data set	# Points(N)	# Attributes(D)	# Classes(K)
Wine	178	13	3
Soybean-small	47	35	4
Iris	150	4	3
Glass	214	10	7
Haberman	306	3	2

A. Wine Data Set

This data set is the result of a 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 wines. There are overall 178 objects. There are 59, 71, 48 objects in class I, class II and class III, respectively. The experimental results are summarized in Table II.

B. Soybean-Small Data Set

The soybean-small data set has 47 objects, each of which is described by 35 attributes. Each object is labeled as one of the four diseases: Diaporthe Stem Canker, Charcoal Rot, Rhizoctonia Root Rot, and Phytophthora Rot. Except for Phytophthora Rot which as 17 objects, all other diseases have 10 objects each. The experimental results are summarized in Table III.

C. Iris Data Set

This data set has often been used as a standard for testing clustering algorithms. This data set has three classes that represent three different varieties of Iris flowers namely *Iris setosa*, *Iris versicolor* and *Iris virginica*. Fifty objects are obtained from each of the three classes, thus a total of 150 objects are available. Every object is described by four attributes, viz. sepal length, sepal width, petal length and petal width. The experimental results are summarized in Table IV.

D. Glass Data Set

This data set has 214 objects and 10 attributes. There are seven clusters (70 building windows, 17 vehicle windows, 76 building windows, zero vehicle windows, 13 containers, 9 tableware and 29 headlamps) that can be grouped in two bigger clusters (163 Window glass, 51 Non-window glass). In this experiment, suppose that the number of clusters is two. The experimental results are summarized in Table V.

E. Haberman Data Set

The Haberman data set contains cases from a study that was conducted on the survival of patients who had undergo surgery of breast cancer. It contains two clusters, 306 objects, and three attributes. The experimental results are summarized in Table VI.

TABLE II
WINE DATA SET

	AC	ARI	RI	MI	HI
CCIA	0.5674	0.3347	0.6855	0.3145	0.3709
kd-tree	0.5674	0.3347	0.6855	0.3145	0.3709
k-means++	0.5674	0.3347	0.6855	0.3145	0.3709
proposed	0.7022	0.3711	0.7187	0.2813	0.4373

TABLE III
SOYBEAN-SMALL DATA SET

	AC	ARI	RI	MI	HI
CCIA	0.7234	0.5452	0.8316	0.1684	0.6633
kd-tree	0.7234	0.5452	0.8316	0.1684	0.6633
k-means++	0.7021	0.5949	0.8205	0.1795	0.6411
proposed	0.7234	0.5452	0.8316	0.1684	0.6633

TABLE IV
IRIS DATA SET

	AC	ARI	RI	MI	HI
CCIA	0.8933	0.7302	0.8797	0.1203	0.7595
kd-tree	0.8933	0.7302	0.8797	0.1203	0.7595
k-means++	0.8933	0.7302	0.8797	0.1203	0.7595
proposed	0.8933	0.7302	0.8797	0.1203	0.7595

TABLE V
GLASS DATA SET

	AC	ARI	RI	MI	HI
CCIA	0.5421	0.2552	0.6659	0.3341	0.3318
kd-tree	0.4626	0.2096	0.7064	0.2936	0.4128
k-means++	0.4907	0.2446	0.5930	0.4070	0.1861
proposed	0.5421	0.2702	0.6764	0.3236	0.3527

TABLE VI
HABERMAN DATA SET

	AC	ARI	RI	MI	HI
CCIA	0.5196	-0.0037	0.4991	0.5009	-0.0017
kd-tree	0.5000	-0.0037	0.4984	0.5016	-0.0033
k-means++	0.5196	-0.0037	0.4991	0.5009	-0.0017
proposed	0.5196	-0.0037	0.4991	0.5009	-0.0017

Table II shows that, for the Wine data set, the performance of CCIA, kd-tree and k-means++ methods remain the same according to the five evaluation indexes, whereas the proposed method outperforms the other three initialization methods. For the Soybean-small data set, it can be seen from Table III that, the performance of the proposed algorithm is comparable to CCIA and kd-tree. Except for ARI, the performance of the proposed algorithm is better than k-means++ for AC, RI, MI and HI. Table IV shows that for the Iris data set, the performance of four initialization methods remains the same. For the Haberman data set, it can be seen from Table VI that, the performance of the proposed algorithm, CCIA and k-means++ is better than that of kd-tree. For the Glass data set, it can be seen from Table V, the proposed algorithm has better performance than the other three initialization method for AC and ARI. Yet for RI, HI and MI, kd-tree has better performance. The reason is that the distribution of the glass data set does not satisfy the principle of "small intra-cluster distances and large inter-cluster distances". Overall, the presented algorithm is superior to the other initialization methods.

IV. CONCLUSION

This paper presents MST-based initialization method for the k-means algorithm. The skeleton point is defined based on the feature of MST. Furthermore, a new distance measure between the skeleton data points is defined, which makes the distance discrimination between data points higher and takes consideration of both degree and distance. The clustering results of the k-means algorithm are compared for the presented algorithm, CCIA, kd-tree and k-means++ methods on the Wine, Soybean-small, Iris, Glass and Haberman data sets from UCI. Regarding the results of the five indexes (i.e., AC, ARI, RI, MI, and HI), the clustering performance of the k-means algorithm with the presented algorithm is the most favorable among the four methods.

ACKNOWLEDGMENT

This work is partially supported by the National Natural Science Foundation of China (no. 61373004), Shanghai Normal University Innovation Team Project (no. A700115001005).

REFERENCES

[1] G. Gan, C. Ma, J. Wu, *Data clustering: Theory, algorithms, and applications, ASA-SIAM series on statistics and applied probability*, SIAM, Philadelphia: McGraw-Hill, 2007.

[2] N. Abdelhamid, A. Ayesh, F. Thabtah, "Phishing detection based associative classification data mining," *Expert Syst. Appl.*, vol.41, no.13, pp. 5948–5959, 2014.

[3] O.A. Abbas, "Comparisons between data clustering algorithms," *Int. Arab J. Inform. Technol.*, vol.5, no.3, pp. 320–325, 2008.

[4] F. Höppner, F. Klawonn, R. Kruse, T. Runkler, *Fuzzy cluster analysis: methods for classification, data analysis and image recognition*, New York: John Wiley & Sons, USA, 1999.

[5] E. Boundaillier, G. Hebrail, "Interactive interpretation of hierarchical clustering," *Intell. Data Anal.*, vol.2, no.1, pp. 229–244, 1997.

[6] E. Forgy, "Cluster analysis of multivariate data: efficiency vs. interpretability of classification," *Biometrics*, vol.21, no.3, pp. 41–52, 1965.

[7] J.B. Macqueen, "Some methods for classification and analysis of multivariate observation," in *Proc. of Berkeley Symposium on Mathematical Statistics and Probability*, California, 1967, pp. 281–297.

[8] T. Gonzalez, "Clustering to minimize the maximum intercluster distance," *Theor. Comput. Sci.*, vol.38, no.2, pp. 293–306, 1985.

[9] Shehroz S. Khan, Amir Ahmad, "Cluster center initialization algorithm for k-means clustering," *Pattern Recognition Letters*, vol.25, no.11, pp. 1293–1302, 2004.

[10] Redmond, S. J., & Heneghan, C. “A method for initialising the k-means clustering algorithm using kd-trees,” *Pattern Recognition Letters*, vol.28, no.8, pp. 965–973, 2007.

[11] Arthur, D., & Vassilvitskii, S. “k-means++: The advantages of careful seeding,” in: *Proc. of the 18th annual ACM-SIAM symposium on discrete algorithms*, New Orleans, Louisiana, 2007, pp.1027–1035.

[12] Cao, F., Liang, J., & Bai, L. “A new initialization method for categorical data clustering,” *Expert Syst. Appl.*, vol.36, no.7, pp.10223–10228, 2009.

[13] Damodar Reddy, Devender Mishra, Prasanta K. Jana, *MST-based cluster initialization for k-means*, Berlin: Springer-verlag, 2010, pp.329-338.

[14] Lan Huang, Shixian Du, Yu Zhang, Yaolong Ju, Zhuo Li, “K-means initial clustering center optimal algorithm based on kruskal,” *Journal of Information & Computational Science*, vol.9, no.9, pp.2387–2392, 2012.

[15] M. Mahajan, P. Nimbor, K. Varadarajan, “The planar k-means problem is NP-hard,” *Theoretical Computer Science*, vol.442, no.8, pp.13–21, 2012.

[16] Zahn, C.T. “Graph-theoretical methods for detecting and describing gestalt clusters,” *IEEE Trans. on Computers*, vol.20, no.1, pp.68–86, 1971.