Hindawi
Mathematical Problems in Engineering
Volume 2018, Article ID 8451796, 14 pages
https://doi.org/10.1155/2018/8451796

Research Article

CciMST: A Clustering Algorithm Based on Minimum Spanning Tree and Cluster Centers

Xiaobo Lv,[1] Yan Ma ,[1] Xiaofu He,[2] Hui Huang,[1] and Jie Yang[3]

[1]*College of Information and Electrical Engineering, Shanghai Normal University, Shanghai 200234, China*
[2]*College of Physicians & Surgeons, Columbia University, New York, USA*
[3]*Computational Intelligence and Brain Computer Interface (CIBCI) Center, University of Technology Sydney, Australia*

Correspondence should be addressed to Yan Ma; ma-yan@shnu.edu.cn

Received 10 October 2018; Accepted 4 December 2018; Published 17 December 2018

Academic Editor: George A. Papakostas

The minimum spanning tree- (MST-) based clustering method can identify clusters of arbitrary shape by removing inconsistent edges. The definition of the inconsistent edges is a major issue that has to be addressed in all MST-based clustering algorithms. In this paper, we propose a novel MST-based clustering algorithm through the cluster center initialization algorithm, called cciMST. First, in order to capture the intrinsic structure of the data sets, we propose the cluster center initialization algorithm based on geodesic distance and dual densities of the points. Second, we propose and demonstrate that the inconsistent edge is located on the shortest path between the cluster centers, so we can find the inconsistent edge with the length of the edges as well as the densities of their endpoints on the shortest path. Correspondingly, we obtain two groups of clustering results. Third, we propose a novel intercluster separation by computing the distance between the points at the intersection of clusters. Furthermore, we propose a new internal clustering validation measure to select the best clustering result. The experimental results on the synthetic data sets, real data sets, and image data sets demonstrate the good performance of the proposed MST-based method.

1. Introduction

Clustering aims to group a set of objects into clusters such that the objects of the same cluster are similar, and objects belonging to different clusters are dissimilar. Clustering is an active research topic in statistics, pattern recognition, machine learning, and data mining. A wide variety of clustering algorithms have been proposed for different applications [1]. The different clustering methods, such as partitional, hierarchical, density-based, and grid-based approaches, are not completely satisfactory due to the multiplicity of problems and the data distributions [2–4]. For instance, as a well-known partitional clustering algorithm, the K-means algorithm often assumes a spherical shape structure of the underlying data, and it can detect clusters with irregular boundaries. Most of the hierarchical clustering algorithms cannot satisfy the requirement of clustering efficiency and accuracy simultaneously [5]. DBSCAN is a classical density-based clustering algorithm that can find clusters with arbitrary shapes. However, it needs to input four parameters which are difficult to determine [4]. CLIQUE combines grid-based and density-based clustering algorithms, and it works efficiently for small data sets. However, its cluster boundaries are either horizontal or vertical, owing to the nature of the rectangular grid [5]. Sufficient empirical evidence has shown that minimum spanning tree (MST) representation is invariant to detailed geometric changes in the boundaries of clusters. Therefore, the shape of the cluster boundary has little impact on the performance of the algorithm, which allows us to overcome the problems commonly faced by the classical clustering algorithms [6].

The MST-based clustering algorithm is able to achieve the clustering result provided that the inconsistent edges between the clusters have been determined and removed. Hence, defining the inconsistent edge is one of the main problems to be solved in this paper. If we tackle this issue from the view of the length of edges as well as the density of points, the MST method commonly requires a set of parameters whose tunings are problematic in practical cases, which will bring the clustering result instability. Furthermore,

many factors including the arbitrary shape of clusters and the different densities and noise make this problem more complex. We found that the shortest path between the cluster centers contains the inconsistent edge; that is, the search scope of inconsistent edges can be narrowed to the shortest path between the cluster centers. Based on this finding, we propose the cluster center initialization algorithm based on the geodesic distance and dual densities of points. In this method, the Euclidean distance between the vertices is modified with the geodesic distance in the MST. Global and local densities of the vertices are defined through adjusting the variance in the Gaussian function. Correspondingly, two groups of K cluster centers under different densities are achieved. Next, we find the K-1 shortest paths among the K(K-1)/2 paths between any pair of K cluster centers. Any K-1 inconsistent edges are determined and removed with consideration of the length of each edge as well as the densities of the two endpoints on the shortest path. Hence, we obtain two groups of clustering results. Then, we define a novel intercluster separation with the distance between the points at the intersection of clusters. The optimal clustering result is determined by combining intercluster separation and intracluster compactness. The key contributions of this paper include the following: (i) propose the use of cluster center initialization in MST-based clustering, (ii) give a cluster center initialization algorithm that takes advantage of geodesic distance, and (iii) develop a new intercluster separation.

The rest of this paper is organized as follows: in Section 2, we review some existing work on MST-based clustering algorithms. We next present our proposed cluster center initialization method in Section 3. In Section 4, we give the definition of inconsistent edges. Section 5 presents a new internal clustering validation measure. In Section 6, we analyze the time complexity of the algorithm. Section 7 presents the experimental evaluations. Finally, Section 8 concludes our work and discusses future work.

2. Related Work

A spanning tree is an acyclic subgraph of a graph G, which contains all the vertices from G. The minimum spanning tree (MST) of a weighted graph is the minimum weight spanning tree of that graph. The cost of constructing an MST is $O(m \log n)$ with the classical MST algorithm, where m is the number of edges in the graph and n is the number of vertices [7]. Enormous amounts of data in various application domains can be represented in a graph. The set of vertices in the graph represents the points in the data set and the edge connecting those vertices reveals the relationship between points. Usually, MST-based clustering algorithms consist of three steps: (1) construct a minimum spanning tree; (2) remove the inconsistent edges to get a set of connected components (clusters); (3) repeat step (2) until the terminating condition is satisfied. Since Zahn first proposed the MST-based clustering method, recent efforts focused on the definition of the inconsistent edges [8]. Under the ideal condition that the clusters are well separated and there exist no outliers, the inconsistent edges are the longest edges [8]. However, the longest edge does not always correspond to the inconsistent edge if there are outliers in the data set. Xu et al. used an MST to represent multidimensional gene expression data [9]. They describe three objective functions. The first algorithm removes the k-1 longest edges so that the total weight of the K subtrees is minimized. The second objective function is to minimize the total distance between the center and each point in a cluster. The third objective function is to minimize the total distance between the "representative" of a cluster and each point in the cluster. The clustering result is vulnerable to the outliers when removing the inconsistent edges according to the lengths of edges. To solve this problem, Laszlo et al. proposed an MST-based clustering algorithm that puts a constraint on the minimum cluster size rather than on the number of clusters [10]. Grygorash et al. proposed a hierarchical MST-based clustering approach (HEMST) that iteratively cuts edges, merges points in the resulting components, and rebuilds the spanning tree [11].

In addition to the inconsistent edges, the definition of the density of points is also one of the crucial factors that affect the performance of the clustering result. The traditional MST-based clustering algorithms only exploit the information of edges contained in the tree to partition a data set, which will make these algorithms more vulnerable to the outliers. The recent MST-based methods tend to define the inconsistent edges based on the local density around the point. Some methods define the density of points with the degree of the vertex. Chowdbury et al. proposed a density oriented MST-based clustering technique that assumes that the boundary between any two clusters must belong to a valley region (a region where the density of the data points is the lowest compared to those of the neighboring regions) and that the inconsistency measure is based on the finding of such valley regions [12]. Luo et al. proposed an MST-based clustering algorithm with neighborhood density difference estimation [13]. Wang et al. proposed to find a local density factor for each data point during the construction of an MST and discarding outliers [14]. Zhong et al. proposed a graph-theoretical clustering method based on two rounds of minimum spanning trees to deal with separated clusters and touching clusters [15]. For some specially distributed data, such as uniform distributed data, if only the local density of the point is taken into account, it cannot be guaranteed that the best clustering result will be achieved. To address this problem, we propose to calculate the global and local density of the point. Some MST-based algorithms are combined with other methods, such as information theory [16], k-means [17], and multivariate Gaussians [18].

3. The Proposed Cluster Center Initialization Method

3.1. Density Peaks Clustering. Among the recent cluster center initialization methods, density peaks clustering (DPC) has been widely used [19]. We propose a new cluster center initialization method based on DPC in this paper. Here, we briefly describe DPC.

It is assumed in DPC that the cluster centers are characterized by a higher density than their neighbors and by a relatively large distance from points with higher densities. DPC utilizes two quantities: one is the density ρ_i of point x_i and the other is its distance δ_i from points of higher densities.

The density ρ_i of point x_i is defined as

$$\rho_i = \sum_j \exp\left(-\frac{D\left(x_i, x_j\right)^2}{2\sigma^2}\right) \tag{1}$$

where $D(x_i, x_j)$ is the Euclidean distance between points x_i and x_j, and σ is variance. Algorithm 1 shows the definition of σ.

The distance between the point x_i and the other points with higher densities, denoted by δ_i, is defined as

$$\delta_i = \begin{cases} \min\limits_{j:\rho_i<\rho_j}\left(d\left(x_i, x_j\right)\right), & \text{if } \rho_i < \rho_j \\ \max\limits_{j}\left(d\left(x_i, x_j\right)\right), & \text{otherwise} \end{cases} \tag{2}$$

When the density ρ_i and the distance δ_i for each point have been calculated, the decision graph is further generated. The points with relatively high ρ_i and δ_i are considered as cluster centers.

3.2. Geodesic-Based Initialization Method. In the DPC method, the precondition to find the correct cluster centers is that the distribution of cluster centers conforms to the abovementioned assumptions. However, many studies show that the two assumptions have certain limitations in different scenarios. As can be seen from Figure 1(a), for the Three circles data set which is from [20], three cluster centers (represented as solid triangles) obtained by the DPC method lie in the red cluster and green cluster, respectively, yet none lie in the blue circle. As shown in Figure 1(b), there is only one point with a relatively large value of ρ_i and δ_i which lies in the red cluster. This is due to the fact that both the green cluster and the blue cluster are nonconvex shaped, and the densities of points in the blue cluster are smaller than that in the green cluster, which leads to the result that no cluster center lies in the blue cluster.

The DPC method exploits the Euclidean distance between the two points as the distance measure. This distance measure is suitable for the data sets with convex shape, yet is not suitable for the data sets with nonconvex shape. To address this issue, this paper adopts a new distance metric-geodesic distance.

Let X be a data set with K clusters and n data points, that is, $X = \{x_i,\ x_i \in R^P,\ i = 1, 2, \cdots, n\}$. Data set X is represented by an undirected completed graph $G = (V, E)$, where $V = \{v_1, v_2, \ldots, v_n\}$, $|E| = n(n-1)/2$. Each data point x_i in data set X corresponds to a vertex $v_i \in V$. For the sake of convenience, the vertex v_i in graph G is represented by x_i. Let $T = (V, E_T)$ denote the MST of $G = (V, E)$, where $E_T = \{e_1, e_2, \cdots, e_{n-1}\}$, $e_i \in E(G)$.

Lemma 1. *There is one and only one path between each pair of vertices in T.*

Definition 2 (geodesic distance). Suppose $p = \{p_1, p_2, \cdots, p_l\} \in V$ is the path between two vertices x_i and x_j in T, where edge $(p_k, p_{k+1}) \in E_T$, $1 \le k < l-1$. The geodesic distance between two vertices x_i and x_j is defined as

$$D_g\left(x_i, x_j\right) = \sum_{k=1}^{l-1} D\left(p_k, p_{k+1}\right) \tag{3}$$

where $D(p_k, p_{k+1})$ is the Euclidean distance between two points x_i and x_j.

The Euclidean distance between pairwise points is replaced by geodesic distance, which leads to the result that the distance between pairwise points in the same cluster becomes smaller, while the distance between pairwise points from the different cluster is larger. For example, we employ statistical tests for the Three circles data set. We divide the interval $[0, 1]$ for the normalized distance measure into ten subintervals of equal length. Then we count the number of pairwise points in the same or from different clusters whose Euclidean distance or geodesic distance drops into each subinterval, respectively. It can be seen from Figure 2 that, with respect to the Euclidean distance and geodesic distance, a large quantity of pairwise points in the same cluster drop into the first four subintervals, which implies that the difference between both of them is small. In contrast, as for the Euclidean distance and geodesic distance, the differences of distribution of pairwise points from the different clusters are significant. The former is concentrated in the 2nd-7th subintervals, while the latter is distributed among all of the subintervals. The reason is that the shape of the Three circles data set is nonspherical. For the distance metric between pairwise points from the different clusters, the corresponding result is smaller when provided with the Euclidean distance and larger when provided with the geodesic distance.

After the geodesic distance is defined, the density ρ_i of point x_i is redefined as

$$\rho_i = \sum_j \exp\left(-\frac{D_g\left(x_i, x_j\right)^2}{2\sigma^2}\right) \tag{4}$$

The size of the density ρ_i is related to σ in (4), and σ is proportional to s in Algorithm 1 mentioned in Section 3.1; that is, the larger s is, the larger σ will be, and vice versa.

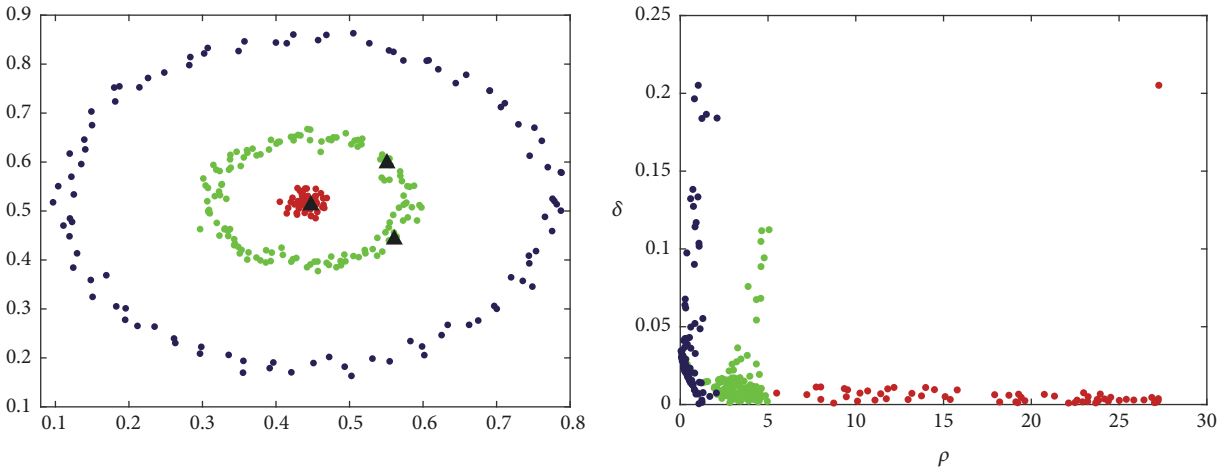
In addition, the distance δ_i between the points x_i and the other points with higher densities is redefined as

$$\delta_i = \begin{cases} \min\limits_{j:\rho_i<\rho_j}\left(D_g\left(x_i, x_j\right)\right), & \text{if } \rho_i < \rho_j \\ \max\limits_{j}\left(D_g\left(x_i, x_j\right)\right), & \text{otherwise} \end{cases} \tag{5}$$

For the purpose of adapting the selected cluster centers to data sets with arbitrary shape, we introduce the concept of global density and local density. The variance σ can be seen as the scale factor. The smaller the value of σ is, the smaller the scale is. Hence, the corresponding density ρ_i can be seen as the local density around the point x_i. In contrast, the larger the value of σ is, the larger the scale is. And the

(1) **Input:** Data set $X = \{x_i,\ x_i \in R^P,\ i = 1, 2, \cdots, n\}$, the total number of points n, a predefined parameter s
(2) **Output:** The value of σ
(3) **Begin**
(4) Calculate and sort the pairwise distance between points in ascending order, that is, $\{d_1, d_2, \cdots, d_{n(n-1)/2}\}$
(5) Calculate $th = [s * n(n-1)/2]$ ("[]" represents a rounding operation)
(6) Calculate $\sigma = d_{th}$
(7) **End**

Algorithm 1: Pseudocode of the Definition of σ.



(a) The cluster centers

(b) Decision graph under Euclidean distance

Figure 1: Three circles data set.

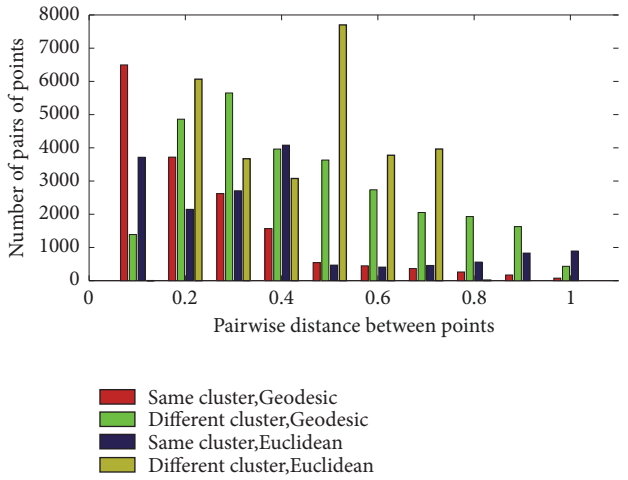


Figure 2: The histograms of two distance measures for pairwise points in the same and different clusters.

corresponding density ρ_i can be seen as the global density around the point x_i. The parameter s is set as 2% and 20% after a number of experiments in this paper, with which we can obtain the local density and global density of the point. The points with relatively higher ρ_i and δ_i are considered as cluster centers, and correspondingly two groups of cluster centers are achieved.

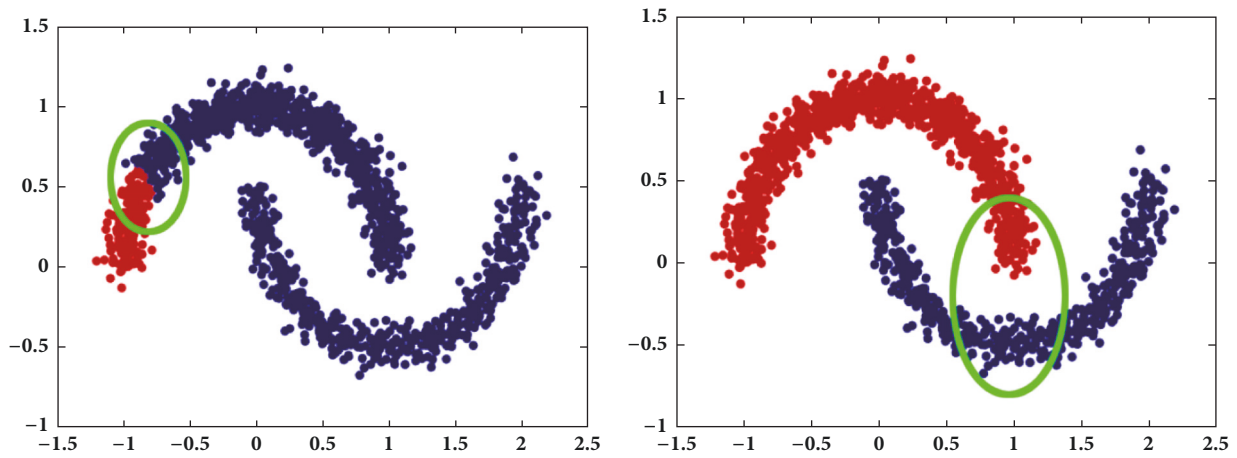
4. The Definition of Inconsistent Edge

For the data set with K categories, the MST-based clustering method attempts to partition the MST into K subtrees, $\{T_i\}_{i=1}^{K}$, by removing the K-1 inconsistent edges.

Lemma 3. *The inconsistent edge between two vertices must be in the path connecting two cluster centers of the different clusters to which the two vertices belong.*

Proof. Suppose data set X contains two clusters A and B whose cluster centers are C_a and C_b, respectively. Construct the MST $T = (V, E_T)$ for data set X. Given $e_{ab} \in E_T$ connecting a vertex $a \in A$ to a vertex $b \in B$, e_{ab} is an inconsistent edge. According to Lemma 1, there is one and only one path between points C_a and a, C_b and b, represented as $p_{C_a a} = \{p_{a1}, p_{a2}, \cdots, p_{al}\} \in A$, $p_{C_b b} = \{p_{b1}, p_{b2}, \cdots, p_{bm}\} \in B$. Correspondingly, the path between clusters C_a and C_b is $p_{C_a a} \cup e_{ab} \cup p_{C_b b}$. Thus, e_{ab} belongs to the path between C_a and C_b. □

There are $K(K-1)/2$ paths among K cluster centers. Next, we need to find K-1 paths from them. The inconsistent edge must lie in the intersection of each pair of adjacent clusters. Obviously, the geodesic distance between the cluster centers of the two adjacent clusters is smaller than that of two nonadjacent clusters. Therefore, the methodology for selecting K-1 paths is to construct the MST T_c, $T_c \subset T$,



(a) s=2% (b) s=20%

FIGURE 3: The clustering results of the Two moons data set.

according to the geodesic distances of $K(K-1)/2$ pairs of cluster centers in T, and, correspondingly, K-1 edges in T_c correspond to the paths in T.

After determining K-1 paths, the next task is to find the K-1 inconsistent edges on each of the K-1 paths. Generally, the inconsistent edge has two features: (1) Its length is longer. (2) The densities of the two end points are smaller. Based on this fact, we define a new parameter for the edge e_{ij} connecting x_i and x_j in the path.

$$\varsigma_{ij} = \frac{D\left(x_i, x_j\right)}{\rho_i + \rho_j} \tag{6}$$

where $D(x_i, x_j)$ is the Euclidean distance between points x_i and x_j and ρ_i and ρ_j are the local or global density of points x_i and x_j, respectively. For the K-1 paths, find and remove the edge with the largest value of ς_{ij}, and correspondingly we obtain K clusters.

5. Internal Clustering Validation Index

To adapt to the data sets with various characters, we obtain two groups of cluster centers under local density and global density, and then finally we achieve two groups of clustering results. We can exploit internal validation measures to determine the optimal result from the two clustering results when the external information is not available.

In general, the intercluster separation and intracluster compactness are used as the internal validation measures, where intercluster separation plays a more important role [21]. The calculation of intercluster separation can be categorized into two classes: one is to take the distance of a single pair of points as the intercluster separation. For example, the maximum or minimum distance between pairwise points or the distance between the cluster centers is taken as the intercluster separation. Another is based on the average pairwise distance between points in the different clusters. Let us analyze the two categories. In the first category, the distance between the single pair of points cannot represent the distance between two clusters. The result of intercluster separation in this method is unavoidably wrong if there exist outliers in the data set. And, for the second category, the average distance between pairwise points reflects the average value of pairwise distance of points, which cannot reflect accurately the distance between clusters. Yang et al. [22] proposed an internal clustering validation index based on the neighbors (CVN), which can be exploited to select the optimal result among the multiple clustering results. Similar to CVN, Liu et al. [21] proposed the internal clustering validation index (CVNN), which exploits the intracluster or intercluster relationship between the point and its neighbors. It is required to take the relation between each point and its neighbors into consideration to calculate the intercluster distance with CVN or CVNN. But in fact, we need not consider all points of the data set. Figure 3 illustrates the clustering results with the proposed method on the Two moons data set, respectively, where s=2% and 20%. According to Figure 3, we can see that the proposed method gives the optimal clustering result in Figure 3(b) and the undesirable clustering result in Figure 3(a). The main basis for judging by human eyes whether the cluster result is correct or not is the size of the distance between the points at the intersection of the two clusters, and the distance between the points far from the intersection of the two clusters is not considered. As shown in Figure 3, the green circles denote the points at the intersection of the two clusters. As shown in Figure 3(a), the distance between the points from the different clusters is smaller than the distance in Figure 3(b); that is, the intercluster distance in Figure 3(b) is greater than that in Figure 3(a). Thus, we select the clustering result in Figure 3(b) as the optimal solution.

Based on the above idea, we propose using the intercluster distance based on the distance between the points at the intersection of two different clusters. Let us consider the example in Figure 3(a). There are two clusters which we called the red cluster and the blue cluster. First, we calculate the minimum geodesic distance from each point in the red cluster to all of the points in the blue cluster. Then, we sort all of the

(1) **Input:** Data set $X = \{x_i,\ x_i \in R^P,\ i = 1, 2, \cdots, n\}$, the total number of points n, the number of clusters K, the clustering result $\{Cluster_1, Cluster_2, \cdots, Cluster_K\}$, K-1 inconsistent edges $E_{inc} = \{e_1, e_2, \cdots, e_{K-1}\}$, the geodesic distance $D_g(x_i, x_j)$ between points x_i and x_j
(2) **Output:** Intercluster separation *Sep*
(3) **Begin**
(4) Construct K-1 pairs of adjacent clusters $(Cluster_i, Cluster_j)$ according to $e_i \subset E_{inc}$ (The two end points of e_i belong to $Cluster_i$ and $Cluster_j$.)
(5) Calculate the intercluster distance sep_{ij} between adjacent clusters $Cluster_i$ and $Cluster_j$
(5.1) Select a pair of adjacent clusters $(Cluster_i, Cluster_j)$
(5.2) Calculate the minimum geodesic distance $\min_{x_i \in Cluster_i}\{D_g(x_i, x_j) \mid x_j \in Cluster_j\}$ from each point in the $Cluster_i$ to all of the points in the $Cluster_j$
(5.3) Sort all of the minimum geodesic distances $\min_{x_i \in Cluster_i}\{D_g(x_i, x_j) \mid x_j \in Cluster_j\}$ in ascending order
(5.4) Sum up the top 20% minimum geodesic distances $\{D_{g_{i1}}, D_{g_{i2}}, \cdots, D_{g_{i\chi_1}}\}$ (Here, suppose there are a total of χ_1 minimum geodesic distances)
(5.5) Similar to Step (5.4), for the adjacent clusters $(Cluster_j, Cluster_i)$, sum up the top 20% minimum geodesic distances $\{D_{g_{j1}}, D_{g_{j2}}, \cdots, D_{g_{j\chi_2}}\}$. (Here, suppose there are a total of χ_2 minimum geodesic distances)
(5.6) Calculate the distance $sep_{ij} = (\sum_{o=1}^{\chi_1} D_{gio} + \sum_{p=1}^{\chi_2} D_{gjp})/(\chi_1 + \chi_2)$ between $Cluster_i$ and $Cluster_j$
(6) Calculate the average of the K-1 sep_{ij}
(7) **End**

Algorithm 2: Pseudocode of Intercluster separation.

minimum geodesic distances in ascending order and sum up the top 20% minimum geodesic distances. Next, we exchange the red cluster and the blue cluster. And similarly, we sum up the top 20% minimum geodesic distances. The average of the two previous results is taken as the distance between the red cluster and the blue cluster. For the two clusters which are located at the end points of the inconsistent edge, we calculate the intercluster distance according to the above method. Finally, we take the average of all intercluster distances as the intercluster separation. The detailed algorithm is shown as in Algorithm 2.

Next, we define the intracluster compactness *CP*. Numerous measures estimate the intracluster compactness based on the average pairwise distance. Hence the compactness of $Cluster_i$ with n_i points can be defined as

$$cp_i = \frac{2}{n_i(n_i - 1)} \sum_{x_i, y_i \in Cluster_i} D_g(x_i, y_i) \tag{7}$$

The intracluster compactness of data set X is

$$CP = \frac{1}{K} \sum_{i=1}^{K} cp_i \tag{8}$$

where K is the cluster number.

The smaller the value of *CP* according to (7) and (8), the more compact the data set. We calculate the value of *CP* for the clustering results of Figure 3 with the above method. The value of *CP* for Figures 3(a) and 3(b) is 1.5835 and 1.8233, respectively, which indicates that the intracluster distance for Figure 3(a) is smaller than that of Figure 3(b). The value of cp_i for the red cluster and the blue cluster in Figure 3(a) is 0.3997 and 2.7673, respectively, and the value of cp_i for the red cluster and the blue cluster in Figure 3(b) is 1.9575 and 1.6892, respectively. For Figure 3(a), the value of *cp* of the blue cluster is greater than that of the red cluster. Thus, the value of *CP* is still smaller than the corresponding result of Figure 3(b). In conclusion, the previous method has its limitations.

This paper redefines the intracluster distance based on the greater pairwise geodesic distance between the points in the cluster; that is, the average of the greater pairwise geodesic distance is taken as the intracluster distance. For the intracluster compactness of the data set, we assign a weight to each intracluster distance before summing them up to avoid the aforementioned wrong result. The detailed algorithm is shown as in Algorithm 3.

We propose the internal clustering validation index ICV by combining intercluster separation *Sep* and intracluster compactness *CP*:

$$ICV = \frac{Sep}{CP} \tag{9}$$

In (9), the greater the value of *Sep* is, the smaller the value of *CP* is, and the greater the value of *ICV* is, which indicates the better clustering result. Hence, the clustering result corresponding to the greater value of *ICV* is taken as the optimal result.

6. Complexity Analysis

The flowchart of cciMST is illustrated in Figure 4. The computational complexity of cciMST is analyzed as follows.

Firstly, we do initialization work. We construct the MST for data set X with K clusters and n data points by using the Prim algorithm, which requires $O(n^2)$ calculations. In the calculations of all pairwise Euclidean distance and geodesic distance of data points, $O(n^2)$ and $O(n)$ are required.

(1) **Input:** Data set $X = \{x_i,\ x_i \in R^P,\ i = 1, 2, \cdots, n\}$, the total number of points n, the number of clusters K, the clustering result $\{Cluster_1, Cluster_2, \cdots, Cluster_K\}$, K-1 inconsistent edges $E_{inc} = \{e_1, e_2, \cdots, e_{K-1}\}$, the geodesic distance $D_g(x_i, x_j)$ between points x_i and x_j
(2) **Output:** Intracluster compactness CP
(3) **Begin**
(4) Sort the pairwise geodesic distances of all points from $Cluster_{i\in\{1,2,\cdots,K\}}$
(5) Extract the top 20% maximum geodesic distance $\{D_{g_{i1}}, D_{g_{i2}}, \cdots, D_{g_{i\omega_i}}\}$ (here, suppose there are a total of ω_i maximum geodesic distances)
(6) Calculate the average of the ω_i maximum geodesic distances
(7) Calculate the intracluster distance for the $Cluster_i$, $cp_i = (D_{gi1} + D_{gi2} + \cdots + D_{gi\omega_i})/\omega_i$
(8) Calculate $\Omega = \omega_1 + \omega_2 + \cdots + \omega_K$
(9) Calculate the intracluster compactness of data set X, $CP = \sum_{i=1}^{K}(\omega_i/\Omega)cp_i$ (ω_i/Ω is the weight of $Cluster_i$)
(10) **End**

Algorithm 3: Pseudocode of Intracluster compactness.

Next, we determine the cluster centers. The time complexity of calculating the densities ρ_i and distance δ_i of all data points is $O(n^2)$. It is required to sort all pairwise geodesic distances in ascending order to obtain the variance σ according to (4), which takes $O(n \log n)$ time. The time for the selection of K data points with larger values of ρ_i and δ_i as cluster centers can be ignored due to $K \ll n$.

Then, we determine the inconsistent edges. It takes $2O(K^2)$ to construct the MST T_c for two groups of K cluster centers and determine the edge with the largest value of ς_{ij}.

Finally, we select the optimal clustering result with internal validation measure. It will take $O(n^2)$ calculations for the calculation of Sep, as well as the calculation of CP. Both of the clustering results need to calculate the value of ICV, and hence the time complexity is $2O(n^2)$.

Therefore, the whole time complexity of the proposed algorithm is $7O(n^2) + O(n) + O(n \log n) + 2O(K^2)$.

7. Experimental Result

7.1. Experimental Setup. We evaluated cciMST on four synthetic data sets DS1-DS4, six real data sets, and seven images. The four synthetic data sets are taken from the literature [15, 17, 19]; see Figure 5. The six real data sets are taken from the UCI data sets [23], including Iris, Wine, Zoo, Liver-disorders, and Pendigits. The seven images are taken from the Berkeley image segmentation data set [24]. The descriptions of the four synthetic data sets and the six real data sets are shown in Table 1. The experiments were conducted with MATLAB 2016a which has offered convenient functions. CciMST is compared to the following five clustering algorithms:

(1) k-means [25].
(2) Single linkage [26].
(3) Spectral clustering [27].
(4) Density peaks clustering (DPC) [19].
(5) Spliting-and merg clustering (SAM) [17].

In the above five algorithms, k-means is one of the partitional clustering algorithms and single linkage is one of the hierarchical clustering algorithms. Both of them are traditional clustering algorithms. Spectral clustering is one of the graph-based clustering algorithms. DPC is a clustering algorithm by fast search and find of density peaks. SAM is a split-and-merge hierarchical clustering method based on MST. For k-means and spectral clustering, we take the best clustering result out of 1000 trial runs in terms of the external clustering validity index. The parameter of σ in spectral clustering is set as 0.

To evaluate the goodness of clustering results, we exploit four external clustering validation indices (CVI): accuracy (AC), precision (PR), recall (RE), and F1-measure ($F1$) [28]. The larger the values of AC, PR, RE, and $F1$, the better the clustering solution. Suppose that a data set contains K classes denoted by $C_1, C_2, \cdots, C_k$. Let p_i denote the number of points that are correctly assigned to class C_i. Let q_i denote the points that are incorrectly assigned to the class C_i. Let r_i denote the points that are incorrectly rejected from the class C_i. AC, PR, RE, and $F1$ are defined as follows:

$$AC = \frac{\sum_{i=1}^{K} p_i}{|D|} \tag{10}$$

$$PR = \frac{\sum_{i=1}^{K} \left(p_i / \left(p_i + q_i\right)\right)}{K} \tag{11}$$

$$RE = \frac{\sum_{i=1}^{K} \left(p_i / \left(p_i + r_i\right)\right)}{K} \tag{12}$$

$$F1 = \frac{2 \times PR \times RE}{PR + RE} \tag{13}$$

7.2. Experimental Results on the Synthetic Data Sets

DS1. This data set contains four parallel clusters with different densities. The clustering results are illustrated in Figure 6. Single linkage, SAM, and cciMST can identify the proper clusters. k-means can discover the sphere-shaped clusters properly, whereas it produces unsatisfactory partitions for the non-sphere-shaped clusters. For the spectral clustering algorithm, the similarity matrix is constructed by a Gaussian

Table 1: Description of the four synthetic data sets and the six real data sets.

Data set	Number of Instances	Number of Attributes	Number of Classes
DS1	512	2	4
DS2	299	2	3
DS3	1502	2	2
DS4	788	2	7
Iris	15	4	3
Wine	178	13	3
Zoo	101	16	7
Soybean	47	35	4
Liver-disorders	145	5	2
Pendigits	3498	16	10

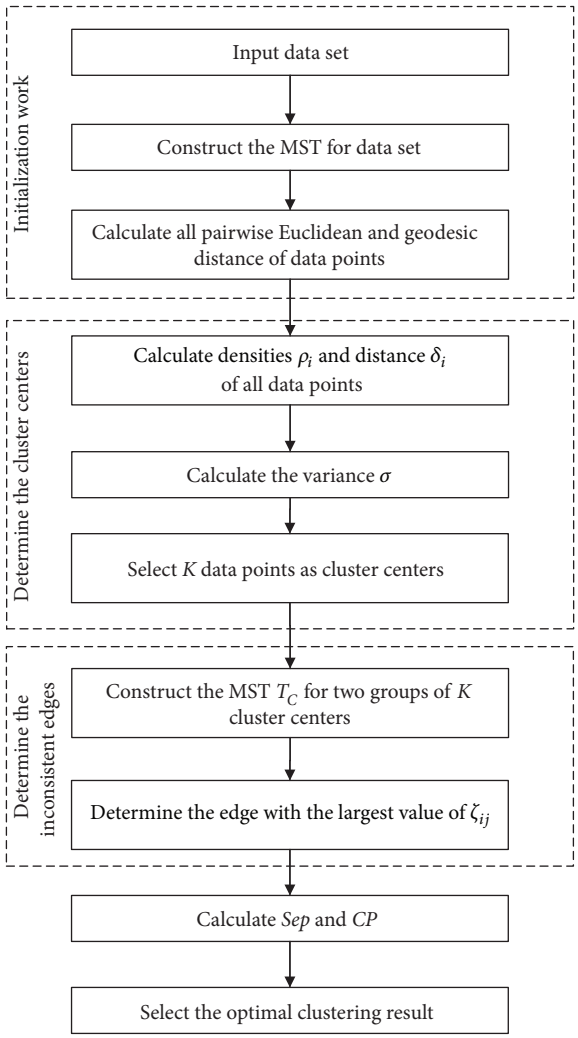


Figure 4: Flowchart of cciMST.

kernel function with Euclidean distance. However, its clustering result is similar to that of k-means. DPC determines the cluster centers through the decision graph constructed by ρ_i and δ_i. Wrong cluster centers will lead to the incorrect clustering result.

DS2. This data set is composed by one Gaussian distributed cluster and two ring clusters surrounding the first one. Figure 7 illustrates the clustering results. K-means, spectral clustering, and DPC cannot provide improper clustering results. Single linkage, SAM, and cciMST can identify the three clusters properly.

DS3. This data set contains two clusters shaped like crescent moons. The clustering results are illustrated in Figure 8. K-means, DPC, and SAM produce unsatisfactory partitions. In the clustering process of SAM, the data points in the subsets produced by k-means are reallocated to maintree. As shown in Figure 9, a data point in each of clusters C_2 and C_3 is redistributed into cluster C_1, which leads to the improper clustering result. Single linkage, spectral clustering, and cciMST can identify the two clusters properly.

DS4. This data set contains seven Gaussian distributed clusters. Figure 10 illustrates the clustering results. Except k-means and single linkage, the rest of the clustering algorithms can identify the clusters properly.

7.3. Experimental Results on the Real Data Sets. From Tables 2–7, the optimal result for the corresponding index is denoted in bold. For the Iris data set, Table 2 indicates that cciMST has the best performance and that the performance of SAM is slightly weaker than that of cciMST. In the case of the Wine data set, the corresponding clustering performances are shown in Table 3. Except for the PR index, the AC, RE, and F1 values of cciMST are higher than those of the other five methods. Moreover, the clustering performances of spectral clustering, DPC, and SAM are better than that of k-means and single linkage. For the Zoo data set, it can be seen from Table 4 that the performances of cciMST, SAM, and k-means are better than those of the other three methods. For the Soybean data set, Table 5 indicates that cciMST and DPC outperform the others. It can be seen from Table 6 that spectral clustering outperforms the other methods on the Liver-disorder, and the performance of cciMST is slightly lower than that of spectral clustering. For the Pendigits data set, Table 7 indicates that cciMST outperforms the other methods.

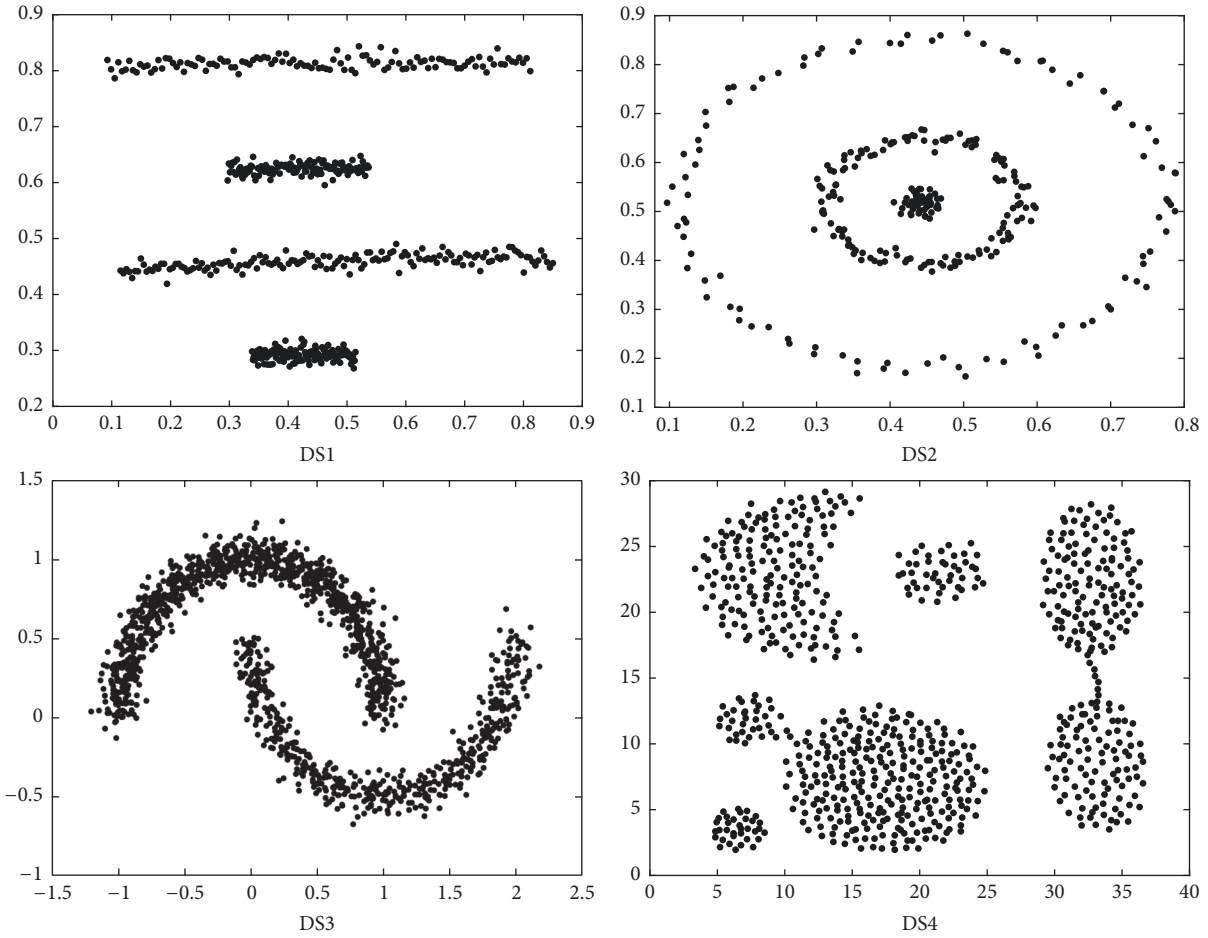


Figure 5: Four synthetic data sets.

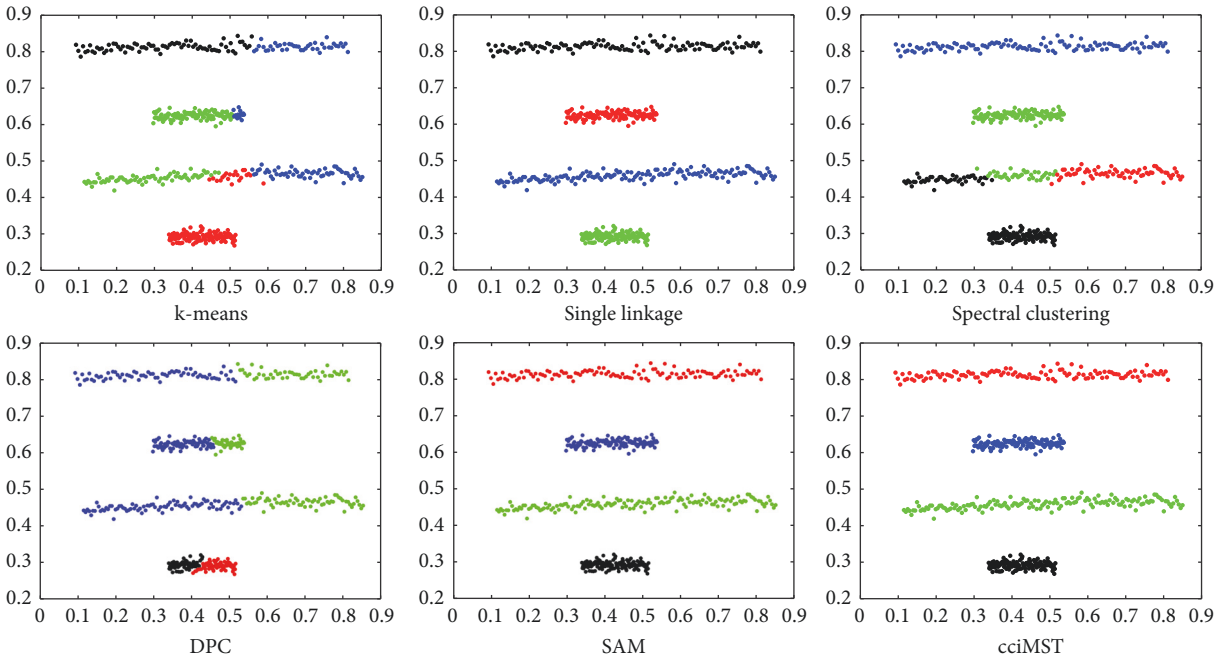


Figure 6: Clustering results on DS1.

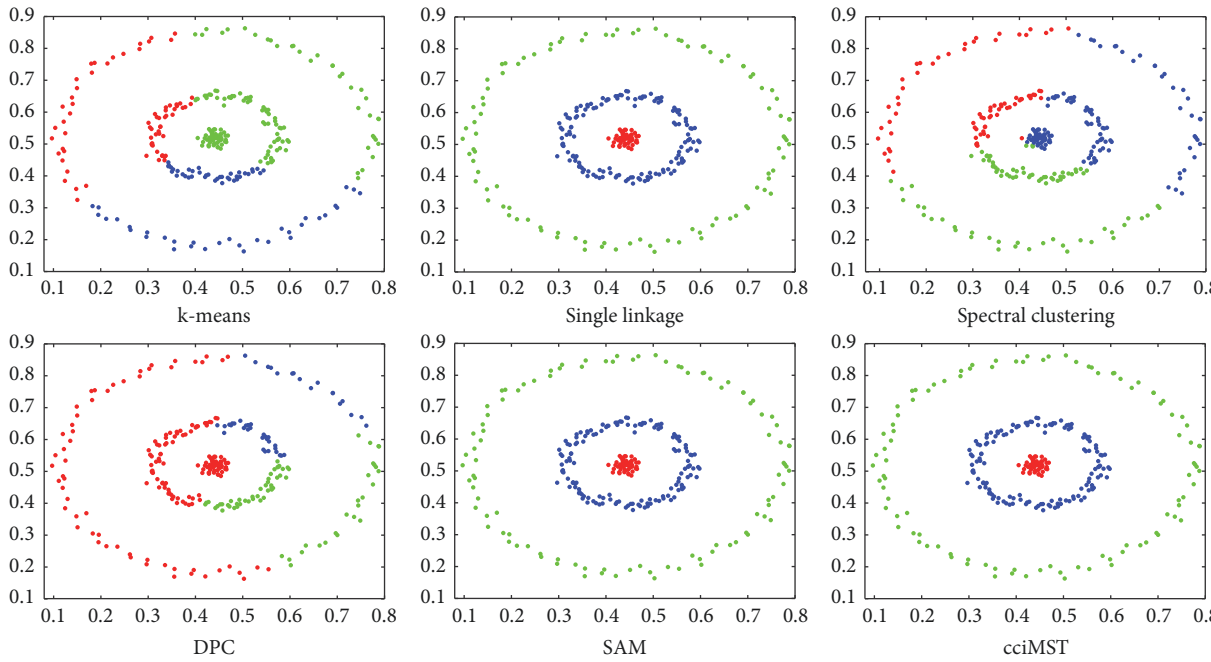


Figure 7: Clustering results on DS2.

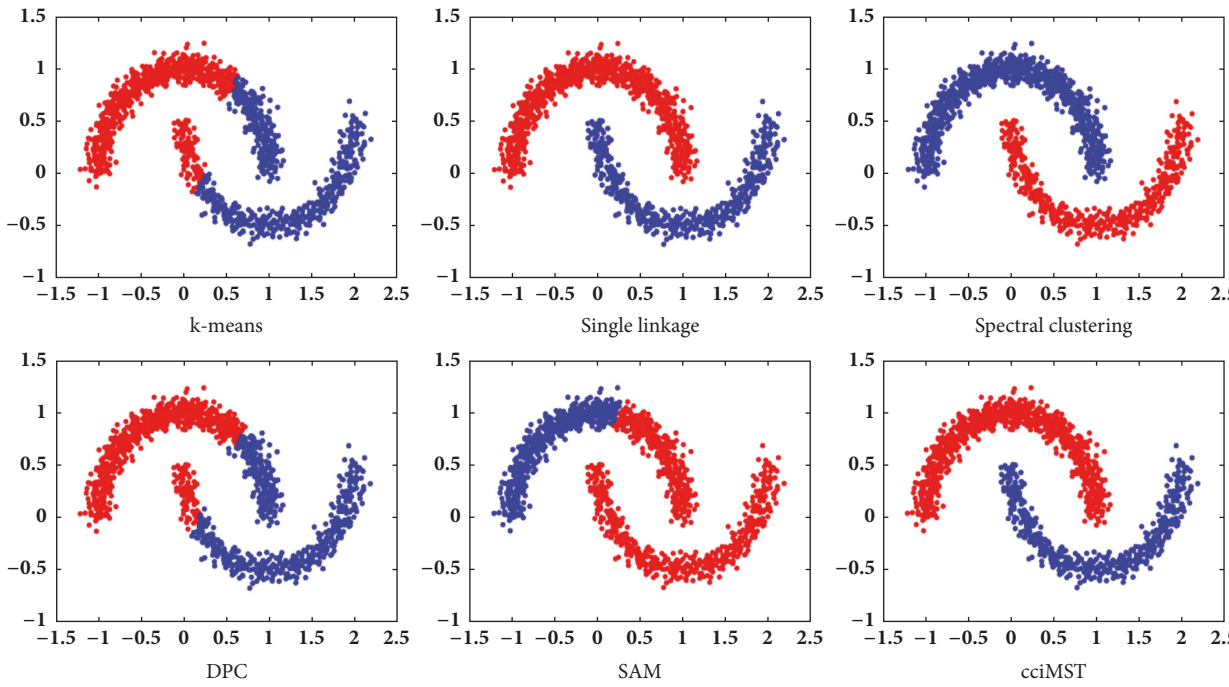


Figure 8: Clustering results on DS3.

Table 2: Clustering performances on Iris.

Index	k-means	Single linkage	Spectral clustering	DPC	SAM	cciMST
AC	0.8572	0.6800	0.8895	0.90667	0.9533	**0.9600**
PR	0.8572	0.6800	0.8895	0.90667	0.9533	**0.9600**
RE	0.8688	0.8367	0.8978	0.92708	0.9562	**0.9619**
*F*1	0.8630	0.7502	0.8936	0.91676	0.9548	**0.9609**

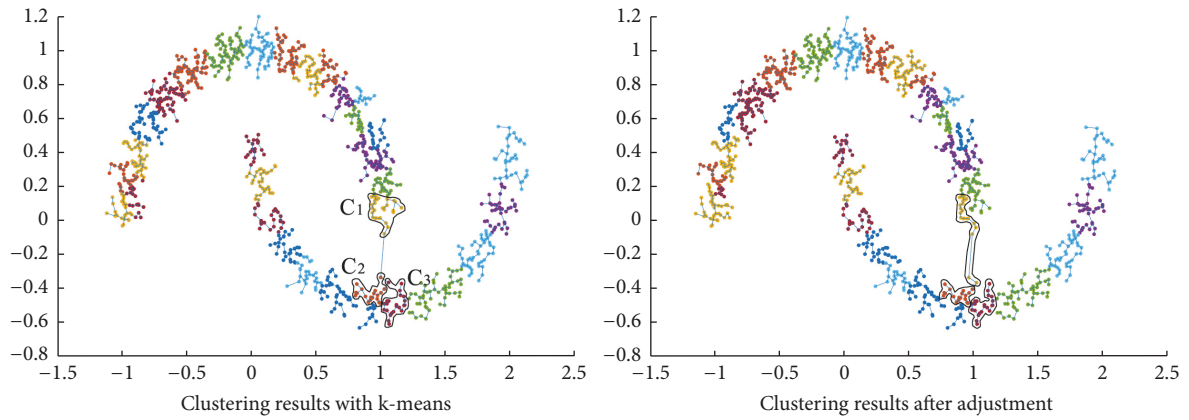


Figure 9: Clustering result with SAM.

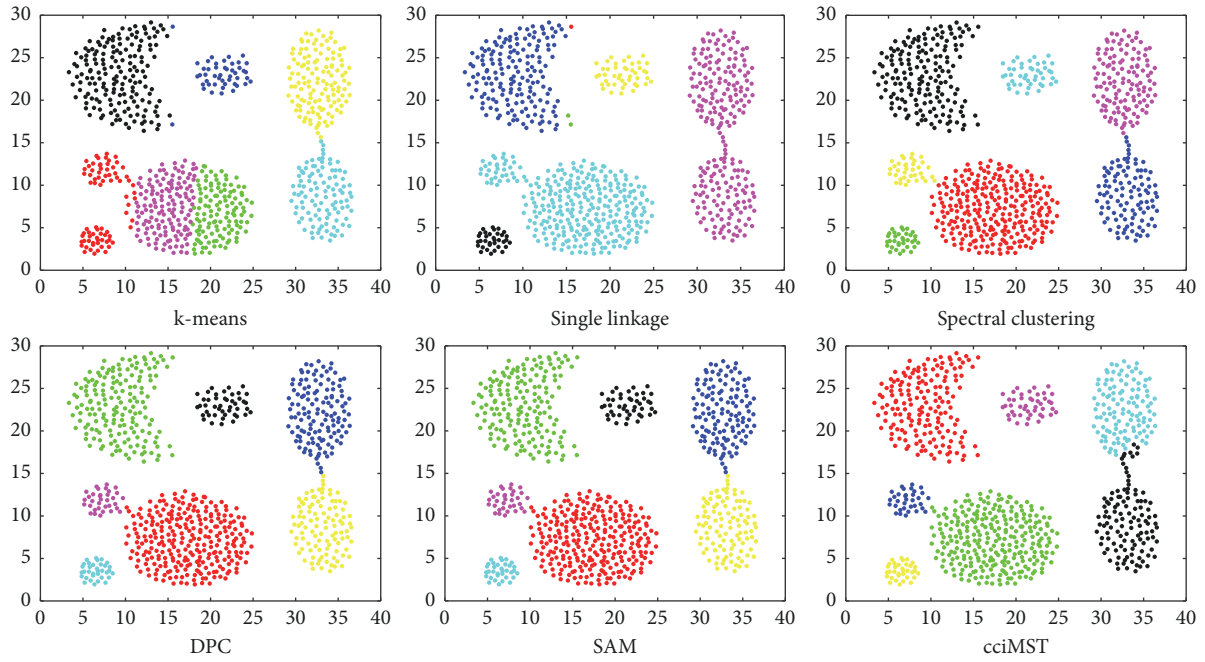


Figure 10: Clustering results on DS4.

Table 3: Clustering performances on Wine.

Index	k-means	Single linkage	Spectral clustering	DPC	SAM	cciMST
AC	0.6525	0.4269	0.7078	0.7079	0.6236	**0.7135**
PR	0.6321	0.3615	0.7029	0.7030	**0.7944**	0.7212
RE	0.6826	0.4709	0.7300	0.7258	0.6730	**0.7470**
*F*1	0.6559	0.4090	0.7162	0.7142	0.7287	**0.7338**

Table 4: Clustering performances on Zoo.

Index	k-means	Single linkage	Spectral clustering	DPC	SAM	cciMST
AC	0.7304	0.6733	0.5087	0.5644	0.6634	**0.8218**
PR	**0.6507**	0.4508	0.3941	0.5094	0.5130	0.6397
RE	**0.6151**	0.4738	0.4196	0.3941	0.6481	0.5875
*F*1	**0.6310**	0.4620	0.4064	0.4444	0.5727	0.6125

Table 5: Clustering performances on Soybean.

Index	k-means	Single linkage	Spectral clustering	DPC	SAM	cciMST
AC	0.7525	0.8085	0.7162	**0.8936**	0.7872	**0.8936**
PR	0.7599	0.775	0.6681	**0.9162**	0.8438	0.8750
RE	0.7647	0.9135	0.6407	0.9052	0.8015	**0.9297**
*F*1	0.7621	0.8386	0.6536	**0.9107**	0.8221	0.9015

Table 6: Clustering performances on Liver-disorders.

Index	k-means	Single linkage	Spectral clustering	DPC	SAM	cciMST
AC	0.6964	0.6345	**0.7117**	0.5310	0.6069	0.6966
PR	0.6105	0.5182	**0.6634**	0.5869	0.5788	0.6106
RE	0.7514	0.8147	0.7006	0.5994	0.5737	**0.7516**
*F*1	0.6737	0.6335	**0.6814**	0.5931	0.5753	0.6738

Table 7: Clustering performances on Pendigits.

Index	k-means	Single linkage	Spectral clustering	DPC	SAM	cciMST
AC	0.6631	0.1124	0.6800	0.7064	0.6635	**0.8385**
PR	0.6632	0.1086	0.6798	0.7037	0.7353	**0.8390**
RE	0.6760	0.6105	0.6855	0.6723	0.6613	**0.8607**
*F*1	0.6694	0.1844	0.6826	0.6876	0.6963	**0.8497**

7.4. Image Segmentation Results. To further evaluate the clustering performance of cciMST on real data sets, we perform image segmentation experiments on the Berkeley Segmentation Data set 300 (BSDS300) [24]. BSDS300 consists of 200 training and 100 testing natural images of size 481∗321. As shown in Figure 11, seven images are extracted from the BSDS300. The first image has various colors of peppers, broccoli, and wooden frames. The second image has a deer, grass, and trees. The third image contains the sky, houses, and grass. The fourth image is composed by flowerbeds and concrete. The fifth image contains bears and sea. The sixth image is composed by sky, mountains, and trees. The seventh image has two horses and grass with different colors.

First, the seven images are segmented by simple linear iterative clustering (SLIC) [29] and the number of superpixels is 250. Then, the image is transformed from RGB to Lab space. We compute the normalized 4-bins histogram for each color channel of Lab space. Next, we concatenate the three histogram vectors and take them as one data point in the data set. Hence, an image has 250 data points described by 12 attributes. The 250 data points of each image are clustered using the six methods: k-means, single linkage, spectral clustering, DPC, SAM, and cciMST, respectively. The clustering results are shown in Figure 11. For the first image, DPC, SAM, and cciMST can properly detect pepper, broccoli, and wooden frames, while single linkage cannot properly detect wooden frames. For the second image, the segmentation results of cciMST are the best. In the case of the third image, cciMST can segment houses, sky, and grass satisfactorily. The segmentation performance of SAM is slightly lower than that of cciMST, and SAM is unable to properly separate the houses from the grass. For the fourth image, the segmentation results of SAM and cciMST are consistent with the perception of the human vision. DPC and cciMST properly segment the bear's body, but improperly segment the legs of the bear. For the sixth image, DPC and cciMST can properly detect the sky, mountains, and trees. K-means, DPC, and cciMST can segment properly the horses in the seventh image.

8. Conclusions

Our MST-based clustering method tries to identify the inconsistent edges through the cluster centers. We exploit the geodesic distance between the two vertices in the MST as the distance measure. We also introduce the concept of global and local density of vertices. In addition, we propose the novel internal clustering validation index to select the optimal clustering result. The experimental results on synthetic data sets, real data sets, and image data illustrate that the proposed clustering method has the overall better performance. The future goal is to further improve the computational efficiency of the method.

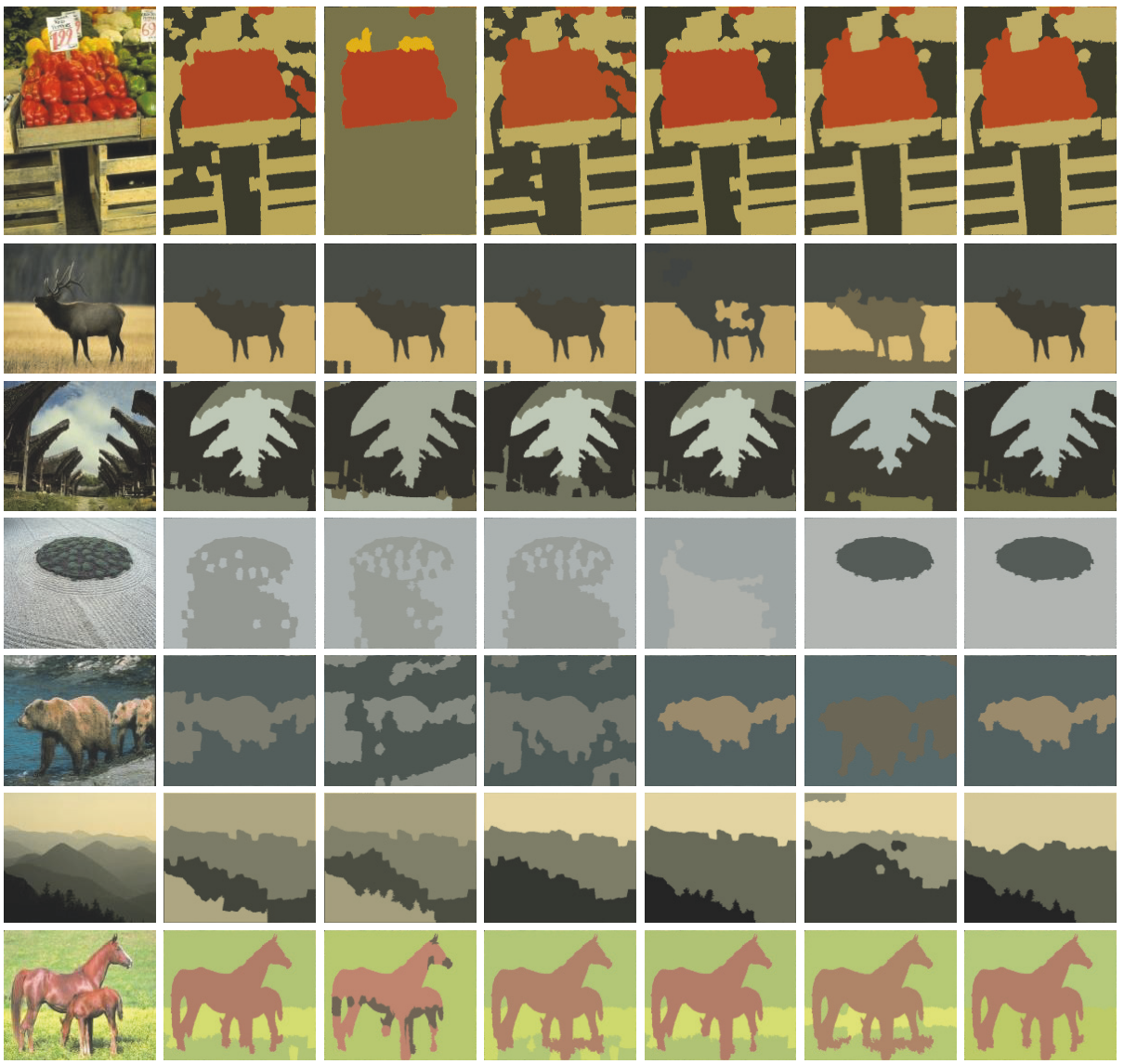


Figure 11: Image segmentation results (The first column displays the original images and the second-seventh columns display the segmentation results with k-means, single linkage, spectral clustering, DPC, SAM, and cciMST).

Data Availability

The code used in this paper is released, which is written in Matlab and available at https://github.com/Magiccbo/CciMST.git.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

Acknowledgments

The authors are grateful to the support of the National Natural Science Foundation of China (61373004, 61501297).

References

[1] A. K. Jain and R. C. Dubes, "Clustering methods and algorithms," in *Algorithms for Clustering Data*, pp. 55–141, 1988.

[2] S. P. Lloyd, "Least squares quantization in PCM," *IEEE Transactions on Information Theory*, vol. 28, no. 2, pp. 129–137, 1982.

[3] M. Ester, H. P. Kriegel, J. Sander, and X. Xu, "A densitybased algorithm for discovering clusters in large spatial databases with noise," in *Proceedings of the Second International Conference on Knowledge Discovery and Data Mining*, pp. 226–231, 1996.

[4] R. Agrawal, J. Gehrke, D. Gunopulos, and P. Raghavan, "Automatic subspace clustering of high dimensional data for data mining applications," in *Proceedings of the ACM SIGMOD International Conference on Management of Data*, pp. 94–105, 1998.

[5] J. Li, K. Wang, and L. Xu, "Chameleon based on clustering feature tree and its application in customer segmentation," *Annals of Operations Research*, vol. 168, no. 1, pp. 225–245, 2009.

[6] R. C. Prim, "Shortest connection networks and some generalizations," *Bell System Technical Journal*, vol. 36, no. 6, pp. 1389–1401, 1957.

[7] J. Kruskal, "On the shortest spanning subtree of a graph and the traveling salesman problem," *Proceedings of the American Mathematical Society*, vol. 7, pp. 48–50, 1956.

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

[9] Y. Xu, V. Olman, and D. Xu, "Clustering gene expression data using a graph-theoretic approach: An application of minimum spanning trees," *Bioinformatics*, vol. 18, no. 4, pp. 536–545, 2002.

[10] M. Laszlo and S. Mukherjee, "Minimum spanning tree partitioning algorithm for microaggregation," *IEEE Transactions on Knowledge and Data Engineering*, vol. 17, no. 7, pp. 902–911, 2005.

[11] O. Grygorash, Z. Yan, and Z. Jorgensen, "Minimum spanning tree based clustering algorithms," in *Proceedings of the 18th IEEE International Conference on Tools with Artificial Intelligence (ICTAI '06)*, pp. 73–81, October 2006.

[12] N. Chowdhury and C. A. Murthy, "Minimal spanning tree based clustering technique: Relationship with bayes classifier," *Pattern Recognition*, vol. 30, no. 11, pp. 1919–1929, 1997.

[13] T. Luo and C. Zhong, "A neighborhood density estimation clustering algorithm based on minimum spanning tree," in *Lecture Notes in Computer Science*, vol. 6401, pp. 557–565, 2010.

[14] X. Wang, X. L. Wang, C. Chen, and D. Wilkes, "Enhancing minimum spanning tree-based clustering by removing density-based outliers," *Digital Signal Processing*, vol. 23, no. 5, pp. 1523–1538, 2013.

[15] C. Zhong, D. Miao, and R. Wang, "A graph-theoretical clustering method based on two rounds of minimum spanning trees," *Pattern Recognition*, vol. 43, no. 3, pp. 752–766, 2010.

[16] A. C. Müller, S. Nowozin, and C. H. Lampert, "Information theoretic clustering using minimum spanning trees," in *Proceedings of the Symposium of the German Association for Pattern Recognition 34th*, pp. 205–215, 2012.

[17] C. Zhong, D. Miao, and P. Fränti, "Minimum spanning tree based split-and-merge: A hierarchical clustering method," *Information Sciences*, vol. 181, no. 16, pp. 3397–3410, 2011.

[18] A. Vathy-Fogarassy, A. Kiss, and J. Abonyi, "Hybrid minimal spanning tree and mixture of gaussians based clustering algorithms," in *Proceedings of the IEEE International Conference on Tools with Artificial Intelligence*, pp. 313–330, 2006.

[19] A. Laio and A. Rodriguez, "Clustering by fast search and find of density peaks," *Science*, vol. 344, no. 6191, pp. 1492–1496, 2014.

[20] A. Y. Ng, M. I. Jordan, and Y. Weiss, "On spectral clustering: analysis and an algorithm," in *Processing Systems: Natural and Synthetic*, pp. 849–856, MIT Press, 2001.

[21] Y. Liu, Z. Li, H. Xiong, X. Gao, J. Wu, and S. Wu, "Understanding and enhancement of internal clustering validation measures," *IEEE Transactions on Cybernetics*, vol. 43, no. 3, pp. 982–994, 2013.

[22] J. Yang, Y. Ma, X. Zhang, S. Li, and Y. Zhang, "An initialization method based on hybrid distance for k-means algorithm," *Neural Computation*, vol. 29, no. 11, pp. 3094–3117, 2017.

[23] UCI Machine Learning Repository, http://www.ics.uci.edu/mlearn/MLRepository.html, http://www.ics.uci/mlearn/ML-Respository.html, 2011.

[24] R. Unnikrishnan, C. Pantofaru, and M. Hebert, "Toward objective evaluation of image segmentation algorithms," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 29, no. 6, pp. 929–944, 2007.

[25] J. MacQueen, "Some methods for classification and analysis of multivariate observations," in *Proceedings of the Fifth Berkeley Symposium on Mathematics, Statistics and Probability*, pp. 281–297, 1967.

[26] P. H. A. Sneath and R. R. Sokal, "Numerical taxonomy. The principles and practice of numerical classification," *Taxon*, vol. 12, no. 5, pp. 190–199, 1963.

[27] J. Shi and J. Malik, "Normalized cuts and image segmentation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, no. 8, pp. 888–905, 2000.

[28] Y. J. Huang, R. Powers, and G. T. Montelione, "Protein NMR recall, precision, and F-measure scores (RPF scores): Structure quality assessment measures based on information retrieval statistics," *Journal of the American Chemical Society*, vol. 127, no. 6, pp. 1665–1674, 2005.

[29] R. Achanta, A. Shaji, K. Smith, A. Lucchi, P. Fua, and S. Süsstrunk, "SLIC superpixels compared to state-of-the-art superpixel methods," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 34, no. 11, pp. 2274–2281, 2012.