Pattern Recognition 43 (2010) 752 – 766

Contents lists available at ScienceDirect

Pattern Recognition

journal homepage: www.elsevier.com/locate/pr

A graph-theoretical clustering method based on two rounds of minimum spanning trees

Caiming Zhong[a,b,c], Duoqian Miao[a,b,*], Ruizhi Wang[a,b]

[a]Department of Computer Science and Technology, Tongji University, Shanghai 201804, PR China
[b]Key Laboratory of Embedded System & Service Computing, Ministry of Education of China, Shanghai 201804, PR China
[c]College of Science and Technology, Ningbo University, Ningbo 315211, PR China

ARTICLE INFO

Article history:
Received 22 January 2009
Received in revised form 19 July 2009
Accepted 24 July 2009

Keywords:
Graph-based clustering
Well-separated cluster
Touching cluster
Two rounds of MST

ABSTRACT

Many clustering approaches have been proposed in the literature, but most of them are vulnerable to the different cluster sizes, shapes and densities. In this paper, we present a graph-theoretical clustering method which is robust to the difference. Based on the graph composed of two rounds of minimum spanning trees (MST), the proposed method (2-MSTClus) classifies cluster problems into two groups, i.e. separated cluster problems and touching cluster problems, and identifies the two groups of cluster problems automatically. It contains two clustering algorithms which deal with separated clusters and touching clusters in two phases, respectively. In the first phase, two round minimum spanning trees are employed to construct a graph and detect separated clusters which cover distance separated and density separated clusters. In the second phase, touching clusters, which are subgroups produced in the first phase, can be partitioned by comparing cuts, respectively, on the two round minimum spanning trees. The proposed method is robust to the varied cluster sizes, shapes and densities, and can discover the number of clusters. Experimental results on synthetic and real datasets demonstrate the performance of the proposed method.

1. Introduction

The main goal of clustering is to partition a dataset into clusters in terms of its intrinsic structure, without resorting to any a priori knowledge such as the number of clusters, the distribution of the data elements, etc. Clustering is a powerful tool and has been studied and applied in many research areas, which include image segmentation [1,2], machine learning, data mining [3], and bioinformatics [4,5]. Although many clustering methods have been proposed in the recent decades, there is no universal one that can deal with all cluster problems, since in the real world clusters may be of arbitrary shapes, varied densities and unbalanced sizes [6,7]. In addition, Kleinberg [8] presented an impossibility theorem to indicate that it is difficult to develop a universal clustering scheme. However, in general, users have not any a priori knowledge on their datasets, which makes it a tough task for them to select suitable clustering methods. This is the dilemma of clustering.

Two techniques have been proposed and studied to alleviate the dilemma partially, i.e. clustering ensemble [9–11] and multiobjective clustering [12]. The basic idea of a clustering ensemble is to use different data representation, apply different clustering methods with varied parameters, collect multiple clustering results, and discover a cluster with better quality [13]. Fred and Jain [13] proposed a co-association matrix to depict and combine the different clustering results by exploring the idea of evidence accumulation. Topchy et al. [10] proposed a probabilistic model of consensus with a finite mixture of multinomial distributions in a space of clusterings, and used the EM algorithm to find the combined partitions. Taking advantage of correlation clustering [14], Gionis et al. [11] presented a clustering aggregation framework, which can find a new clustering that minimizes the total number of disagreements with all the given clusterings. Being different from a clustering ensemble which is limited to the posteriori integration of the solutions returned by the individual algorithms, multiobjective clustering considers the multiple clustering objective functions simultaneously, and trades off solutions during the clustering process [12]. Compared with the individual clustering approach, both clustering ensembles and multiobjective clustering can produce more robust partitions and higher cluster qualities. In addition, some of other clustering methods can automatically cope with arbitrary shaped and non-homogeneous clusters [15].

* Corresponding author at: Department of Computer Science and Technology, Tongji University, Shanghai 201804, PR China. Tel.: +86 21 69589867.
E-mail addresses: charman_zhong@hotmail.com (C. Zhong), miaoduoqian@163.com (D. Miao).

0031-3203/$ - see front matter
doi:10.1016/j.patcog.2009.07.010

Recently more attention has been paid to graph-based clustering methods. Being an intuitive and effective data representation approach, graphs have been employed in some clustering methods [16–25]. Obviously, the tasks of these kinds of methods include constructing a suitable graph and partitioning the graph effectively. Most graph-based clustering methods construct the graphs using *k* nearest neighbors [16,17]. Karypis et al. in CHAMELEON [16] represented a dataset with *k* nearest neighbor graph, and used relative interconnectivity and relative closeness to partition the graph and merge the partitions so that it can detect arbitrary shaped and connected clusters. This varies from data representation of CHAMELEON, in which a vertex denotes a data item, Fränti et al. employed a vertex to represent a cluster so as to speed up the process of clustering [17]. Other graph-based methods take advantage of minimum spanning trees (MST) to represent a dataset [18,19]. Zahn [18] divided a dataset into different groups in terms of their intrinsic structures, and conquered them with different schemes. Xu et al. [19] provided three approaches to cluster datasets, i.e. clustering through removing long MST-edges, an iterative clustering algorithm, and a globally optimal clustering algorithm. Although the methods of Zahn and Xu are effective for datasets with specific structures, users do not know how to select reasonable methods since they have no information about the structures of their datasets. From a statistical viewpoint, González-Barrios [20] identified clusters by comparing *k* nearest neighbor-based graph and the MST of a dataset. The limitation of González-Barrios's method is that only i.i.d. data are considered. Päivinen [21] combined a scale-free structure with MST to form a scale-free minimum spanning tree (SFMST) of a dataset, and determined the clusters and branches from the SFMST. Spectral clustering is another group of graph-based clustering algorithms [22]. Usually, in a spectral clustering, a fully connected graph is considered to depict a dataset, and the graph is partitioned in line with some cut off criterion, for instance, normalized cut, ratio cut, minmax cut, etc. Lee et al.[23]recently presented a novel spectral clustering algorithm that relaxes some constraints to improve clustering accuracy whilst keeping clustering simplicity. In addition, relative neighbor graphs can be used to cluster data [24,25].

For the purpose of relieving the dilemma of users such as choice of clustering method, choice of parameters, etc., in this paper, we propose a graph-theoretical clustering method based on two rounds of minimum spanning trees (2-MSTClus). It comprises two algorithms, i.e. a separated clustering algorithm and a touching clustering algorithm, of which the former can partition separated clusters but has no effect on touching clusters, whereas the latter acts in the opposite way. From the viewpoint of data intrinsic structure, since the concepts of separated and touching are mutually complement as will be discussed in Section 2.1, clusters in any dataset can be either separated or touching. As the two algorithms are adaptive to the two groups of clusters, the proposed method can partially alleviate the user dilemma aforementioned. The main contributions are as follows:

(1) A graph representation, which is composed of two rounds of minimum spanning tree, is proposed and employed for clustering.
(2) Two mutually complementary algorithms are proposed and merged into a scheme, which can roughly cope with clustering problems with different shapes, densities and unbalanced sizes.

The rest of this paper is organized as follows. Section 2 depicts the typical cluster problems. In terms of the typical cluster problems, a graph-based clustering method is presented in Section 3. Section 4 demonstrates the effectiveness of the proposed method on synthetic and real datasets. Section 5 discusses the method and conclusions are drawn in Section 6.

2. Typical cluster problems

2.1. Terms of cluster problems

Since there does not exist a universal clustering algorithm that can deal with all cluster problems [7], it is significant for us to clarify what typical cluster problems are and which typical cluster problem a clustering algorithm favors. Some frequently used terms about cluster problem in the paper are defined as follows.

Definition 1. For a given distance metric, a *well-separated cluster* is a set of points such that the distance between any pair of points in the cluster is less than the distance between any point in the cluster and any point not in the cluster.

The above definition of a *well-separated cluster* is similar to the one in [27]. However, it is also similar to the second definition of a *cluster* presented in [28]. That implies a *cluster* is well-separated for a given distance metric.

Definition 2. For a given density metric and a distance metric, a pair of *separated clusters* is two sets of points such that (1) the closest point regions between the two clusters are with high densities compared to the distance between the two closest points from the two regions, respectively, or (2) the closest point regions between the two clusters are different in density.

For the former situation the separated clusters are called *distance-separated clusters*, while for the later called *density-separated clusters*. Obviously, the separated clusters defined above are not transitive. For instance, if *A* and *B* are a pair of separated clusters, and *B* and *C* are another pair of separated clusters, then *A* and *C* are not necessarily a pair of separated clusters.

Definition 3. A pair of *touching clusters* is two sets of points that are joined by a small neck whose removal produces two separated clusters which are substantially large than the neck itself.

Generally, a threshold, which is dependent on a concrete clustering method, is employed to determine how small a small neck is.

Definition 4. For a given distance metric, a *compact cluster* is a set of points such that the distance between any point in the cluster and the representative of the cluster is less than the distance between the point and any representative of other clusters.

In general, a centroid or a medoid of a cluster can be selected as the representative. The difference between the two representative candidates is that a centroid of a cluster is not necessarily a member point of the cluster, while a medoid must be a member point.

Definition 5. For a given distance metric, a *connected cluster* is a set of points such that for every point in the cluster, there exists at least one other point in the cluster, the distance between them is less than the distance between the point and any point not in the cluster.

The definitions of a compact cluster and a connected cluster are similar to those of center-based cluster and contiguouscluster in [27], respectively.

2.2. Cluster problem samples described by Zahn

Some typical cluster problems are described in Fig. 1 by Zahn [18]. Fig. 1(a) illustrates two clusters with similar shape, size and density.

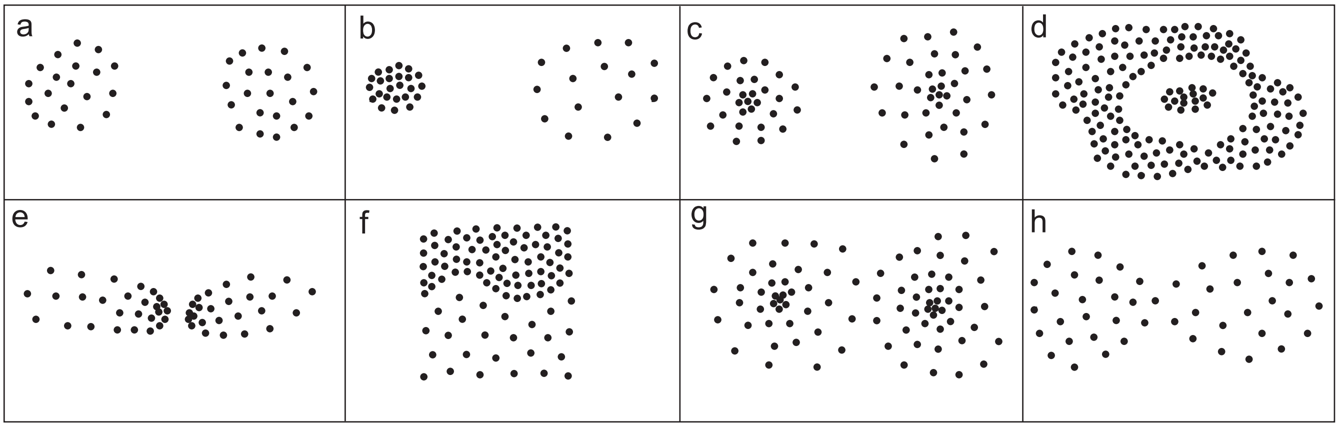

Fig. 1. Typical cluster problems from Ref. [18]. (a)–(e) are distance-separated cluster problems; (f) is density-separated cluster problem; (g) and (h) are density-separated cluster problems.

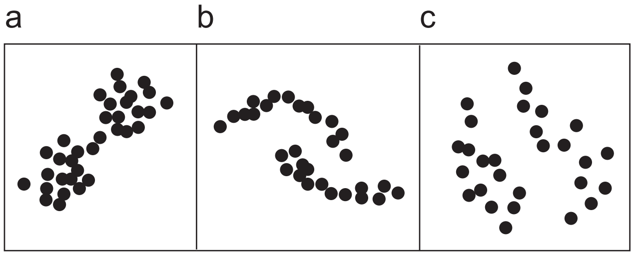

Fig. 2. The three patterns by Handl [12]. In (a), the compactness pattern illustrates the compactness objective which is suitable to deal with spherical datasets. In (b), the connectedness pattern depicts the connectedness objective which handles datasets of arbitrary shape; (c) is a spacial pattern.

It can be distinctly perceived that the two clusters are separated, since the inter-cluster density is very high compared to the intra-cluster pairwise distance. The principal feature of Fig. 1(b), (c) and (d) is still distance-separated, even if the shapes, sizes and/or densities of two clusters in each figure are diverse. The density of the two clusters in Fig. 1(e) are gradually varied, and become highest in their adjacent boundaries. From the global viewpoint of the rate of density variation, however, the separability remains prominent. Intuitively, the essential difference between the two clusters represented in Fig. 1(f) lies in density, rather than distance, and Zahn [18] called it density gradient. Fig. 1(g) and (h) are quite different from those aforementioned figures, because the two clusters touch each other slightly. Zahn [18] classified the cluster problems in Fig. 1(g) and (h) as touching cluster problems.

2.3. Cluster problems implied by clustering algorithms

Traditionally, clustering algorithms can be categorized into hierarchical, partitioning, density-based and model-based methods [3]. Being different from the traditional taxonomy, however, Handl and Knowles [12,26] classified clustering algorithms into three categories with different clustering criteria illustrated in Fig. 2: (1) algorithms based on the concept of compactness, such as k-means, average-linkage, etc., which make an effort to minimize the intra-cluster variation and are suitable for handling spherical datasets; (2) algorithms based on the concept of connectedness, such as path-based clustering algorithm [29], single-linkage, etc., which can detect the clusters with high intra-connectedness; (3) algorithms based on spatial separation criterion, which is opposed to connectedness criterion and generally considered incorporated with other criteria rather than independently. Actually, the clustering approach taxonomy in [12] is cluster-problem-based, as a clustering algorithm is categorized by the cluster problem which it favors, since the criteria of compactness, connectedness and spatial separation delineate the dataset structures instead of algorithms themselves. In accordance with the classification of clustering algorithm in [12], therefore, the cluster problems fall mainly into two groups: compact cluster problems and connected cluster problems.

2.4. Cluster problems classified in this work

In this paper, we classify cluster problems into two categories: separated problems and touching problems. The former includes distance-separated problems and density-separated problems. In terms of Definition 2, for example, we call the cluster problems depicted in Fig. 1(a)–(e) distance-separated, while the cluster problem depicted in Fig. 1(f) density-separated. Cluster problems in Fig. 1(g) and (h) are grouped, similarly in [18], as touching problems according to Definition 3. Since separated problem and touching problem are mutually supplemental, they may cover all kinds of datasets. This taxonomy of cluster problems ignores the compactness and connectedness. In fact, separated clusters can be compact or connected, and touching clusters can also be compact or connected. Based on our taxonomy, Fig. 3(a) and (b) are touching problems, Fig. 3(c) and (d) are separated problems; while in terms of clustering criteria in [12], Fig. 3(a) and (c) are compact problems, Fig. 3(b) and (d) are connected problems.

With the two-round-MST based graph representation of a dataset, we propose a separated clustering algorithm and a touching clustering algorithm, and encapsulate the two algorithms into a same method.

3. The clustering method

3.1. Problem formulation

Suppose that $X = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_i, \ldots, \mathbf{x}_N\}$ is a dataset, $\mathbf{x}_i = (x_{i1}, x_{i2}, \ldots, x_{ij}, \ldots, x_{id})^T \in \Re^d$ is a feature vector, and x_{ij} is a feature. Let $G(X) = (V, E)$ denote a weighted and undirected complete graph with vertex set $V = X$ and edge set $E = \{(\mathbf{x}_i, \mathbf{x}_j) | \mathbf{x}_i, \mathbf{x}_j \in X, i \neq j\}$. Each edge $e = (\mathbf{x}_i, \mathbf{x}_j)$ has a length $\rho(\mathbf{x}_i, \mathbf{x}_j)$, and generally the length can be Euclidean distance, Mahalanobis distance, City-block distance, etc. [7]. In this paper, Euclidean distance is employed. A general clustering algorithm attempts to partition the dataset X into K clusters: $C_1, C_2, \ldots, C_K$, where $C_i \neq \emptyset$, $C_i \cap C_j = \emptyset$, $X = C_1 \cup C_2 \cdots \cup C_K$, $i = 1 : K$,

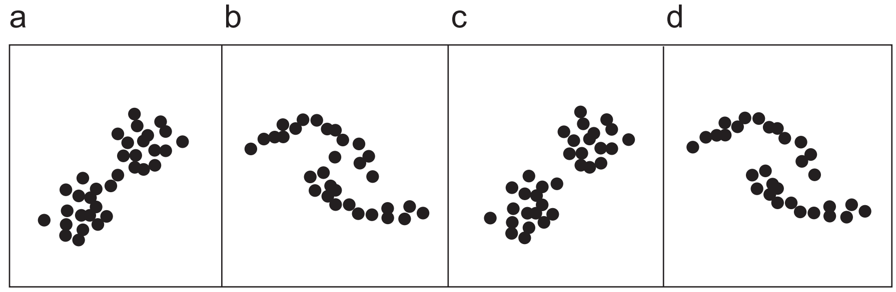

Fig. 3. The different taxonomies of cluster problems. The patterns in (a) and (b) are touching problems, and the patterns in (c) and (d) are separated problems in this paper. The patterns in (a) and (c) are compact problems, and the patterns in (b) and (d) are connected problems by Handl [12].

$j=1:K$, $i \neq j$. Correspondingly, the associated graph will be cut into K subgraphs.

A minimum spanning tree (MST) of graph $G(X)$ is an acyclic subset $T \subseteq E$ that connects all the vertices in V and whose total lengths $W(T)=\sum_{\mathbf{x}_i,\mathbf{x}_j \in T} \rho(\mathbf{x}_i,\mathbf{x}_j)$ is minimum.

3.2. Algorithm for separated cluster problem

As mentioned above, separated cluster problems are either distance-separated or density-separated. Zahn [18] employed two different algorithms to deal with the two situations, respectively. For the purpose of automatic clustering, we try to handle distance-separated problem and density-separated problem with one algorithm.

3.2.1. Two-round-MST based graph

Compared with KNN-graph-based clustering algorithms [16,17], MST-based clustering algorithms [18,19] have two disadvantages. The first one is that only information about the edges included in MST is made use of to partition a graph, while information about the other edges is lost. The second one is that for MST-based approaches every edge's removal will result in two subgraphs. This may lead to a partition without sufficient evidence. With these observations in mind, we consider using second round of MST for accumulating more evidence and making MST-based clustering more robust. It is defined as follows.

Definition 6. Let $T_1 = f_{mst}(V,E)$ denote the MST of $G(X)=(V,E)$. The second round MST of $G(X)$ is defined as

$$T_2 = f_{mst}(V, E - T_1) \tag{1}$$

where $f_{mst}:(V,E) \rightarrow T$ is a function to produce MST from a graph.

If there exists a vertex, say v, in T_1 such that the degree of v is $|V|-1$, v is isolated in $G(V, E-T_1)$. Hence T_2 cannot be generated in terms of Definition 6. To remedy the deficiency simply, the edge connected to v and with the longest length in T_1 is preserved for producing T_2.

Combining T_1 and T_2, a two-round-MST based graph, say $G_{mst}(X) = (V, T_1+T_2) = (V, E_{mst})$, is obtained. The two-round-MST based graph is inspired by Yang [30]. Yang used k MSTs to construct k-edge connected neighbor graph and estimate geodesic distances in high dimensional datasets. Fig. 4(a) and (b), respectively, represent the T_1 and T_2 of Fig. 1(c), in which the dataset is distance-separated. Fig. 4(c) represents the corresponding two-round-MST based graph.

The lengths of edges from T_1 and T_2 have a special relationship (see Theorem 3), which can be used to partition two-round-MST based graph.

Lemma 1. *Let $T(V_T, E_T)$ be a tree. If $T'(V'_T, E'_T)$ is maximum tree such that $V'_T \subseteq V_T$, $E'_T \cap E_T = \emptyset$, then either $|E'_T| = |E_T|-1$ or $|E'_T| = |E_T|$.*

Proof. If $|V_T|-1$ vertices of $T(V_T,E_T)$ have degree 1, and the other vertex, say v, has degree $|V_T|-1$. In T, from any vertex with degree 1, there exist at most $|V_T|-2$ edges connected to other vertices except its neighbor, i.e. v, and no more edge is available to construct $T'(V'_T, E'_T)$. At this moment, $V'_T = V_T \backslash \{v\}$, and $|E'_T| = |V_T|-2 = |E_T|-1$.

Otherwise, suppose vertex v_0 has degree of 1, its neighbor is v_1. From v_0, $|V_T|-2$ edges can be used to construct $T'(V'_T, E'_T)$. In addition, there must exist an edge between vertex v_1 and its non-neighbor vertex. At this moment, $V'_T = V_T$, and $|E'_T| = |V_T|-1 = |E_T|$. □

Corollary 2. *Let $F(V_F, E_F)$ be an acyclic forest. Suppose $F'(V'_F, E'_F)$ is maximum acyclic forest such that $V'_F \subseteq V_F$, $E'_F \cap E_F = \emptyset$, and for any $e \in E'_F$, $F(V_F, E_F \cup \{e\})$ is cyclic, then $|E'_F| \leq |E_F|$.*

Theorem 3. *Suppose T_1 and T_2 are first round and second round MST of $G(V,E)$, respectively. If edges of T_1 and edges of T_2 are ordered ascendingly by their weights as $e_1^1, e_1^2, \ldots, e_1^i, \ldots, e_1^{|V|-1}$, $e_2^1, e_2^2, \ldots, e_2^i, \ldots, e_2^{|V|-1}$, then $\rho(e_1^i) \leq \rho(e_2^i)$, where i denotes the sequence number of ordered edges, and $1 \leq i \leq |V|-1$.*

Proof. Suppose there exists j such that $\rho(e_1^j) > \rho(e_2^j)$. Obviously $\rho(e_1^j) > \rho(e_2^j) \geq \rho(e_2^{j-1}) \geq \cdots \geq \rho(e_2^1)$, in terms of Kruskal's algorithm of constructing a MST, the reason why $e_2^1, e_2^2, \ldots, e_2^j$ are not selected in the j th step of constructing T_1 is that the combination of any one of these edges with $e_1^1, e_1^2, \ldots, e_1^{j-1}$ would produce a cycle in T_1. Let $e_1^1, e_1^2, \ldots, e_1^{j-1}$ form $F(V_F, E_F)$ and $e_2^1, e_2^2, \ldots, e_2^j$ form $F'(V'_F, E'_F)$, then the two forests are acyclic since $e_1^1, e_1^2, \ldots, e_1^{j-1}$ and $e_2^1, e_2^2, \ldots, e_2^j$ are the part of T_1 and T_2, respectively. Because if any edge of $F'(V'_F, E'_F)$ is added into $F(V_F, E_F)$, $F(V_F, E_F)$ would be cyclic, we have $V'_F \subseteq V_F$. However, $|E_F| = j-1$ and $|E'_F| = j$, this contradicts Corollary 2. □

For a tree, any removal of edge will lead to a partition. Whereas to partition a two-round-MST based graph, at least two edges must be removed, of which at least one edge comes from T_1 and T_2, respectively. Accordingly, compared with a cut on MST, a two-round-MST based graph cut requires more evidence and may result in a more robust partition.

Generally, for a given dataset, MST is not unique because two or more edges with same length may exist. However, the non-uniqueness of MST does not influence the partition of a graph for clustering [18], and the clustering induced by removing long edges is independent of the particular MST [31].

3.2.2. Two-round-MST based graph cut

After a dataset is represented by a two-round-MST based graph, the task of clustering is transformed to partitioning the graph with a

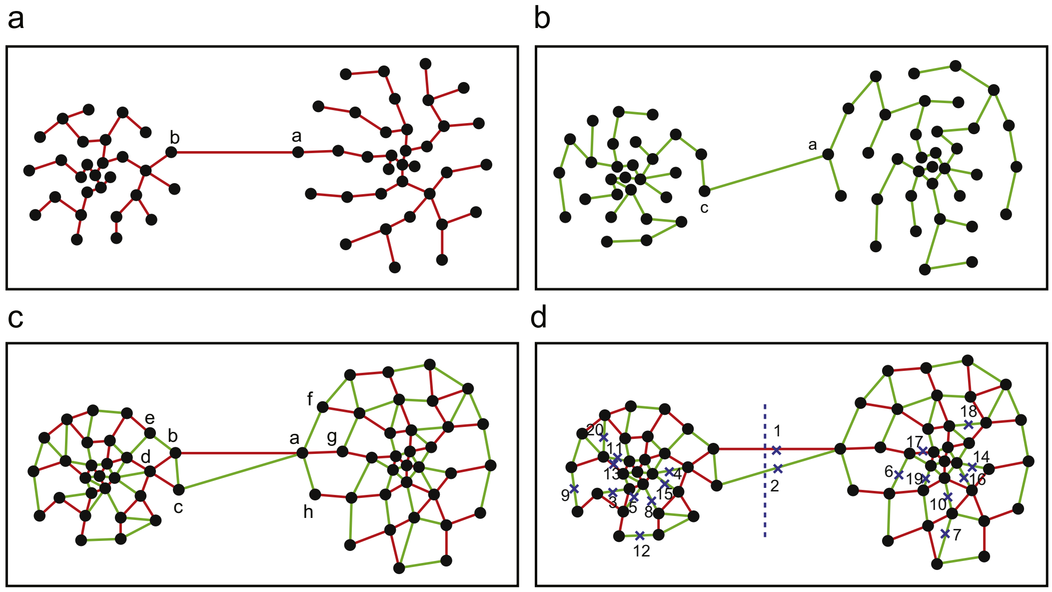

Fig. 4. Two-round-MSTs of the dataset in Fig. 1(c). (a) is the first round of MST, i.e. T_1, and *ab* is the significant and expected edge to be removed in traditional MST based clustering methods. (b) is the second round of MST, i.e. T_2, and *ac* is another significant edge. (c) illustrates the two-round-MST based graph. To partition the graph, *ab* and *ac* are expected edges to be cut. (d) depicts the top 20 edges with large weights based on Definition 2. The first two edge removals result in a valid graph cut. From the top 20 edges, 17 edges come from T_2, and 3 from T_1.

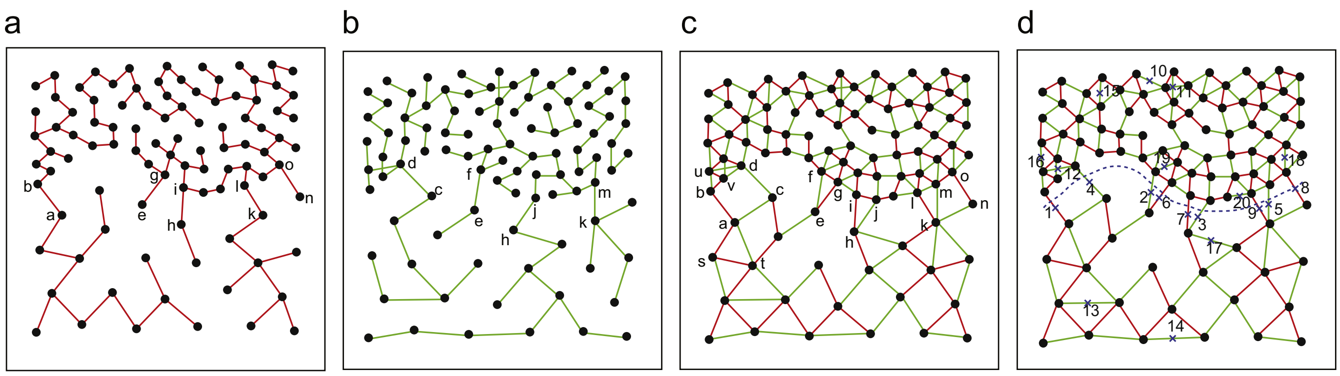

Fig. 5. Density-separated cluster problem taken from [18]. (a) illustrates the first round of MST; (b) depicts the second round of MST. In (c), the two-round-MST based graph is illustrated; *bu* and *bv* are edges connected to *ab* by vertex *b*, while *as* and *at* are connected to *ab* by *a*. The average lengths of the two groups are quite different. (d) illustrates the graph cut. The dashed curve is the graph cut which is achieved by the removals of the top nine edges based on Definition 2, the $Ratio(E_{gcut})$ is 0.444 and greater than the threshold λ.

partitioning criterion. In general, a partitioning criterion plays a pivot role in a clustering algorithm. Therefore, the next task is to define an effective partitioning criterion. Fig. 4(a) is the MST of a distance-separated dataset illustrated in Fig. 1(c). Obviously, *ab* is the very edge to be removed and lead to a valid partition for MST-based methods. Zahn [18] defined an edge inconsistency to detect the edge. That is, the edge, whose weight is significantly larger than the average of nearby edge weights on both sides of the edge, should be deleted. However, this definition is only relevant for the distance-separated cluster problem, for instance, Fig. 4(a). For density-separated cluster problem illustrated in Fig. 5(a), which is called density gradient problem in [18], Zahn first determined the dense set and the sparse set with a histogram of edge lengths, then singled out five inter-cluster edges *ab*, *eg*, *hi*, *kl* and *no*. Although Zahn's method for density-separated problem is feasible, it is somewhat complex. In brief, Zahn used two partitioning criteria to deal with distance-separated cluster problems and density-separated cluster problems, respectively. Our goal, however, is to handle the two situations with one partitioning criterion.

From Figs. 4(c) and 5(c), we observe that the main difference between a distance-separated cluster problem and a density-separated cluster problem is whether the average lengths of edges connected to two sides of an inter-cluster edge are similar or not. For distance-separated clusters in Fig. 4(c), the average length of edges connected to end point *a* of edge *ab* is similar to that of edges connected to the other end of *ab*, while for density-separated clusters in Fig. 5(c), the average lengths of two sets of edges connected, respectively, to two ends of *ab* are quite different. Accordingly, for the purpose of identifying an inter-cluster edge with one criterion for both distance-separated clusters and density-separated clusters, we compare the length of the inter-cluster edge with the minimum of the average lengths of the two sets of edges which are connected to its two ends, respectively. First, we define the weight of an edge as follows:

Definition 7. Let $G_{mst}(X)=(V,E_{mst})$ be a two-round-MST based graph, $e_{ab} \in E_{mst}$ and $a,b \in V$, $w(e_{ab})$ be the weight of e_{ab} as in

$$w(e_{ab}) = \frac{\rho(e_{ab}) - \min(avg(E_a - \{e_{ab}\}), avg(E_b - \{e_{ab}\}))}{\rho(e_{ab})} \tag{2}$$

where $E_a = \{e_{ij} | (e_{ij} \in E_{mst}) \wedge (i = a \vee j = a)\}$, $E_b = \{e_{ij} | (e_{ij} \in E_{mst}) \wedge (i = b \vee j = b)\}$,

$$avg(E) = \frac{1}{|E|} \sum_{e \in E} \rho(e)$$

and $\rho(e)$ is the Euclidean distance of edge e.

Analyzing two-round-MST based graphs of some separated datasets and the corresponding weights defined above, we find that two-round-MST based graphs and the weights have three good features: (1) generally, the weights of inter-cluster edges are quite larger than those of intra-cluster edges. (2) The inter-cluster edges are approximately equally distributed to T_1 and T_2. (3) Except inter-cluster edges, most of edges with large weights come from T_2, and this is supported by Theorem 3. Fig. 4(d) depicts the top 20 weights of the distance-separated dataset in Fig. 1(c). The two inter-cluster edges are those with top two weights, respectively, and one is from T_1 and the other one is from T_2. Among the next 18 edges, 16 edges come from T_2 and only two edges come from T_1. Fig. 5(d) describes the top 20 weights of the density-separated dataset in Fig. 1(f). The top nine weights are from the very nine inter-cluster edges, of which five are from T_1 and four are from T_2, and all of the remaining 11 edges belong to T_2.

In terms of the first feature, a desired two-round-MST based graph cut can be achieved by removing the edges with largest weight one by one. The next two features indicate that whether or not a graph cut is valid can be determined by analyzing the distribution of removed edges.

Definition 8. Let $Rank(E_{mst})$ be a list of edges ordered descendingly by corresponding weights as in

$$Rank(E_{mst}) = \langle edge\_topweight(E_{mst}) \circ Rank(E_{mst} - \{edge\_topweight(E_{mst})\}) \rangle \tag{3}$$

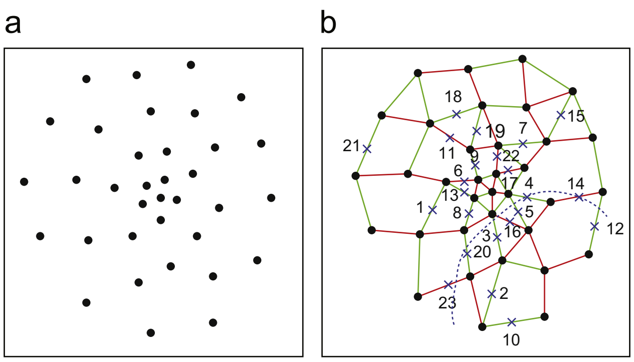

Fig. 6. A cluster cannot be partitioned any more. (a) illustrates the sub-dataset of Fig. 1(c); (b) separated clustering algorithm is applied to the sub-dataset. When a graph cut is obtained, which is indicated by the dashed curve, the $Ratio(E_{gcut})$ is 0.304 and less than the threshold λ.

where $edge\_topweight(E_{mst}) = \arg\max_{e \in E_{mst}}(w(e))$, $\circ$ is a concatenate operator.

Edge removing scheme: The edge with large weight has the priority to be removed, namely edges are removed in the order of $Rank(E_{mst})$. Since every removal of edge may lead to a graph cut (excluding the first removal), we must determine whether or not a new graph cut is achieved after each removal. The determination could be made by traversing the graph with either breadth-first search algorithm or depth-first search algorithm.

Definition 9. Let E_{gcut} be a set of removed edges when a graph cut on a two-round-MST based graph is achieved, if the following holds:

$$Ratio(E_{gcut}) = \frac{\min(|E_{gcut} \cap T_1|, |E_{gcut} \cap T_2|)}{|E_{gcut}|} \geq \lambda \tag{4}$$

where λ is a threshold, then the graph cut is valid, otherwise it is invalid. If the first graph cut is valid, the cluster is said to be separated, otherwise, non-separated.

Figs. 4(d) and 5(d) illustrate that both two first graph cuts are valid, because the $Ratio(E_{gcut})$'s are 0.500 and 0.440, respectively, greater than the threshold $\lambda = 0.333$ which is discussed in Section 4. Consequently, the datasets in Figs. 4 and 5 are separated, and are partitioned by removing the first two and first nine edges, respectively. Fig. 6(a) represents a subgraph produced by applying the scheme on the dataset in Fig. 4, while Fig. 6(b) indicates this subdataset is no-separated, since the $Ratio(E_{gcut})$ for the first cut is 0.304 and less than the threshold. However, the scheme is not always effective, because two exceptions exist.

Exception 1. In a density-separated dataset, there exist two (or more) inter-cluster edges which have a common vertex close to dense part. For example, in Fig. 7(a), inter-cluster edge e_{fg} and e_{hg} have a common vertex g which belongs to the dense part of the dataset. The dashed curve is the expected graph cut. But the weight of e_{fg} is less than those of e_{ab}, e_{cd} and e_{ef}, because when we compute the weight of e_{fg}, another inter-cluster edge e_{hg} is concerned according to Definition 7. As a result, more removed edges are from T_2 when the first graph cut is achieved, and the probability of the cut being valid decreases. The straightforward solution is to ignore the longest neighbor edge. For example, when the weight of e_{fg} is computed, edge e_{hg} should be ruled out from E_g.

Exception 2. The weight defined in Definition 7 is a ratio. If there exists an edge which is quite small in length, and the vertices connected to its one end are extremely close, then its weight is relatively large. In Fig. 7(c), vertices e, f, g are very close. For edge e_{hf}, because $avg(E_f - \{e_{hf}\})$ is extremely small, $w(e_{hf})$ is top 1 even though its length is far less than those of e_{ab} and e_{ac}. To remedy this exception, the edge length can be considered as a penalty.

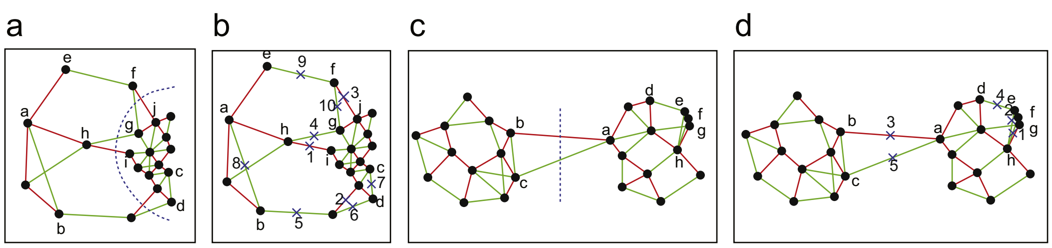

Fig. 7. Two exceptions for Definition 2. (a) is a density-separated cluster problem, the dashed curve is the expected graph cut. (b) illustrates the top 10 edges. As hg is considered for evaluating the weight of fg in terms of Definition 2, fg has a less weight than ab, ef and cd do. (c) is a distance-separated cluster problem, the dashed line is the expected graph cut. (d) illustrates another exception: since e, f, g are too close, he and hf have greater weights than ab and ac do.

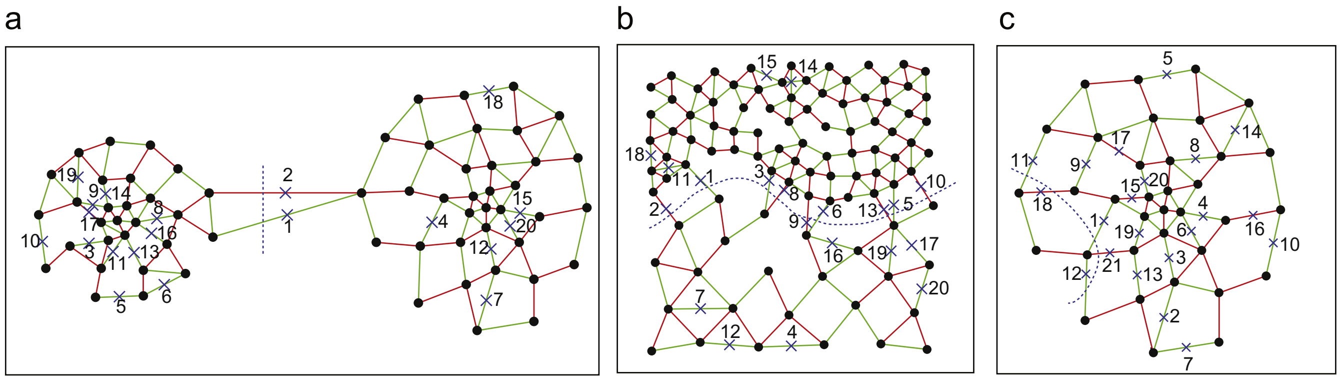

Fig. 8. Graph cuts with the improved definition of weight. In (a), the removals of the top two edges lead to the graph cut on dataset illustrated in Fig. 4, and corresponding $Ratio(E_{gcut})$ is 0.500. In (b), the graph cut on dataset presented in Fig. 5 is obtained by removing the top 13 edges, and the corresponding $Ratio(E_{gcut})$ is 0.385. In (c), the graph cut is different from that illustrated in Fig. 6, and the corresponding $Ratio(E_{gcut})$ is 0.238.

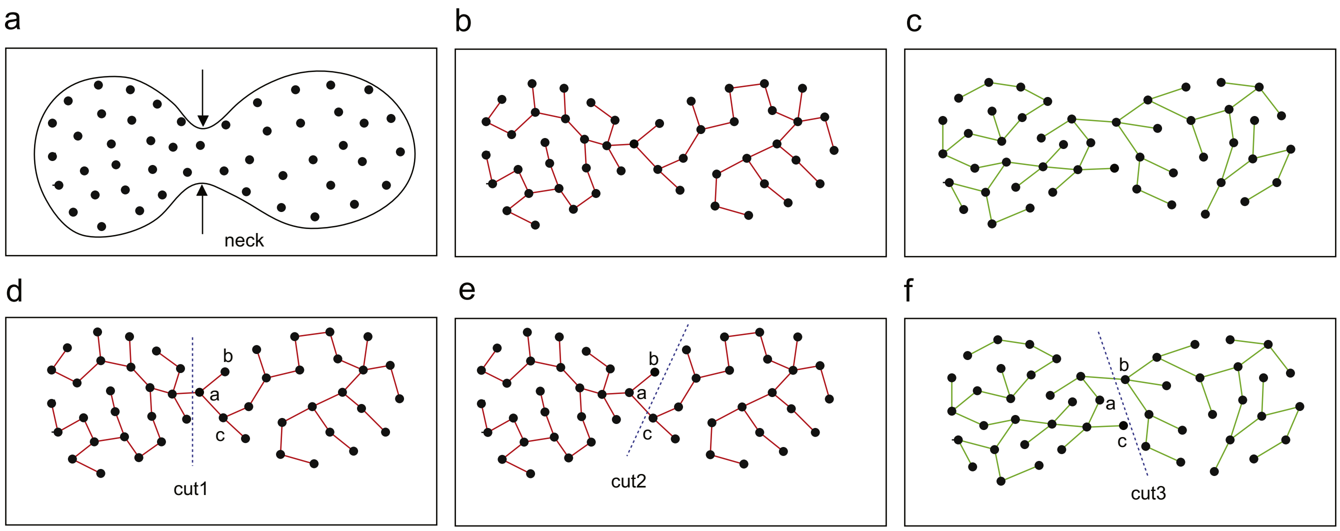

Fig. 9. Clustering a touching problem. (a) is the dataset from Fig. 1. (b) illustrates the first round of MST. (c) represents the second of MST. Comparing $cut1$ of T_1 in (d) and $cut3$ of T_2 in (f), two inconsistent vertices (a,c) exist, while between $cut2$ in (e) and $cut3$ in (f), there also exist two inconsistent vertices (b,c).

Therefore, the weight of e_{ab} in Definition 7 is redefined as

$$w(e_{ab}) = \delta \times \frac{\rho(e_{ab}) - \min(avg(E'_a - \{e'_a\}), avg(E'_b - \{e'_b\}))}{\rho(e_{ab})} + (1-\delta) \times \rho(e_{ab}) \tag{5}$$

where $E'_a = E_a - \{e_{ab}\}$, $e'_a = \arg\max_{e \in E'_a}(\rho(e))$, $E'_b = E_b - \{e_{ab}\}$, $e'_b = \arg\max_{e \in E'_b}(\rho(e))$, δ is a penalty factor and $0 \le \delta \le 1$. $E'_a - \{e'_a\}$ and $E'_b - \{e'_b\}$ ignore the longest neighbor edges, while penalty factor δ gives a tradeoff between the ratio and the edge length.

Fig. 8 illustrates the first graph cut of applying redefined weight on the three datasets in Figs. 4–6. The corresponding $Ratio(E_{gcut})$'s are 0.500, 0.380, 0.240, respectively. According to the discussion of λ in Section 4, the first two graph cuts are still valid and the third one is still invalid.

A subgraph partitioned from a separated problem may be another separated problem. Therefore, we must apply the graph cut method to every produced subgraph iteratively to check whether or not it can be further partitioned until no subgraphs are separated.

Algorithm 1. Clustering separated cluster problems

Input: $G(X) = (V, E)$, the graph of the dataset to be partitioned
Output: S, the set of partitioned subdatasets.
Step 1. Compute T_1 and T_2 of $G(X)$, and combine the two MSTs to construct the two-round-MST based graph $G_{mst}(X)$, and put it into a table named *Open*; create another empty table named *Closed*.
Step 2. If table *Open* is empty, sub-datasets corresponding to subgraphs in *Closed* table are put into S; return S.
Step 3. Get a graph $G'(X') = (V', E')$ out of *Open* table, calculate the weights of edges in $G'(X')$ with Eq. (5), and build the list $Rank(E')$.
Step 4. Remove the edges of $G'(X')$ in the order of $Rank(E')$ until a cut is achieved.
Step 5. If the cut is valid in terms of Definition 9, put the two subgraphs produced by the cut into *Open* table; otherwise put graph $G'(X')$ into *Closed* table.
Step 6. Go to Step 2.

Algorithm 1iteratively checks subgraphs, and partitions the separated ones until there exists no separated subgraphs. At the same

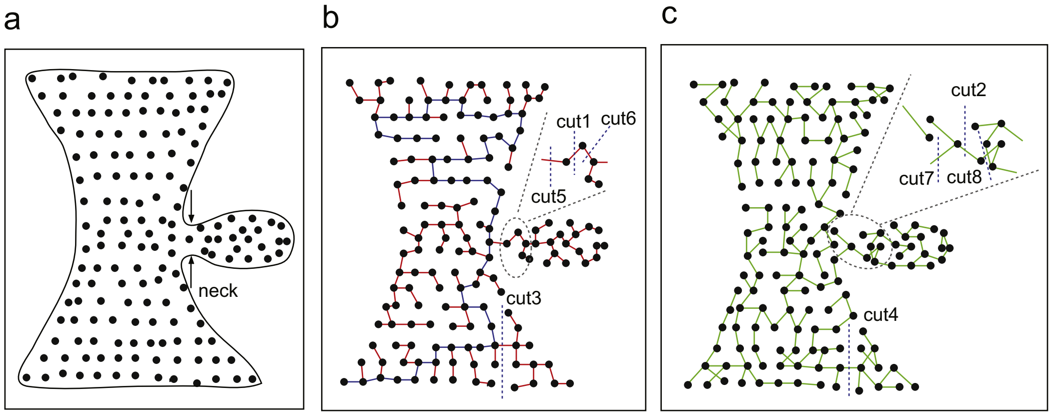

Fig. 10. A teapot dataset as a touching problem. (a) is a teapot dataset with a neck. In (b), the diameter defined by Zahn [18] is illustrated by the blue path. *cut*5 in (b) and *cut*7 in (c) are similar ($\beta = 1$), so are *cut*1 and *cut*2, *cut*6 and *cut*2, *cut*1 and *cut*8, *cut*3 and *cut*4.

time, the algorithm takes no action on non-separated subgraphs, namely touching subgraphs. In fact, the reason why Algorithm 1 can identify separated clusters is the definition of edge weight in Definition 7. The weight of an edge e_{ab} reflects the relation between the length of e_{ab} and two neighbor region densities of vertices a and b, respectively, where the neighbor region density is measured by the average length of the edges in the region. If the weight of e_{ab} is large, the densities of the two neighbor regions are high compared to the length $\rho(e_{ab})$, or the two densities are very different. For a pair of touching clusters, as a neck exists and lengths of edges in the neck are small compared to the neighbor region densities, namely the weights of edges in the neck are small, Algorithm 1 cannot detect the touching clusters.

3.3. Algorithm for touching cluster problem

Although Algorithm 1 can identify separated clusters, it becomes disabled for touching clusters. After the algorithm is applied to a dataset, each induced sub-dataset will be either a touching cluster or a basic cluster which cannot be partitioned further.

For a touching cluster, a neck exists between the two connected subclusters. The neck of touching cluster problem in Fig. 1(h) is illustrated in Fig. 9(a). Zahn [18] defined a diameter of MST as a path with the most number of edges, and detected the neck using diameter histogram. However, a diameter does not always pass through a neck. Fig. 10 illustrates an exception. We identify the neck by considering T_1 and T_2 simultaneously. The two-round-MSTs of Fig. 1(h) are depicted by Fig. 9(b) and (c), respectively. The task in this phase is to detect and remove these edges crossing the neck, and discover touching clusters. Based on the two-round-MSTs, an important observation is as follows:

Observation 1. A partition resulted from deleting an edge crossing the neck in T_1 is similar to a partition resulted from deleting an edge crossing the neck in T_2.

On the contrary, for the two cuts from T_1 and T_2, respectively, if one cut does not cross the neck, the two corresponding partitions will be generally quite different from each other. Comparing the partition on T_1 in Fig. 9(d) with the partition on T_2 in Fig. 9(f), we notice that only two vertices (a and c) belong to different group, and is called *inconsistent vertices*. Similarly, only two *inconsistent vertices* (b and c) exist between the cuts in Fig. 9(e) and (f).

For the purpose of determining whether two cuts are similar, the number of *inconsistent vertices* must be given out as a constraint, i.e. if the number of *inconsistent vertices* between two cuts is not greater than a threshold, say β, the two cuts are similar. For the previous example in Fig. 9, $\beta = 2$ is reasonable. However, some unexpected pairs of cuts which do not cross the neck of a dataset may conform to the criterion and are determined to be similar. For example, the *cut*3 in Fig. 10(b) and the *cut*4 in Fig. 10(c) are similar if $\beta = 1$, however, the two cuts are unexpected. Fortunately, the following observation can remedy this bad feature.

Observation 2. With a same threshold β, the number of pairs of similar cuts which cross the neck is generally greater than that of pairs of similar cuts which do not cross the neck.

In Fig. 10(b) and (c), suppose $\beta = 1$, it is easy to find another pair of similar cuts which cross the necks other than *cut*1 and *cut*2, for instance, *cut*5 and *cut*7, *cut*6 and *cut*2, *cut*1 and *cut*8, while there exists no other pair of similar cuts near *cut*3 and *cut*4. Therefore, *cut*3 and *cut*4 are discarded because the similar evidence is insufficient. With the observation in mind, we can design the algorithm for touching cluster problems as follows.

Definition 10. Let P^{T_1} be the list of $N-1$ partitions on T_1 as in

$$P^{T_1} = \langle (p_{11}^{T_1}, p_{12}^{T_1}), (p_{21}^{T_1}, p_{22}^{T_1}), \ldots, (p_{(N-1)1}^{T_1}, p_{(N-1)2}^{T_1}) \rangle \tag{6}$$

where pair $(p_{i1}^{T_1}, p_{i2}^{T_1})$ denotes the partition which results from removing the i th edge in T_1, $p_{i1}^{T_1}$ and $p_{i2}^{T_1}$ are subsets of vertices, $p_{i1}^{T_1} \cup p_{i2}^{T_1} = V$, $|p_{i1}^{T_1}| \le |p_{i2}^{T_1}|$. Similarly, the list of $N-1$ partitions on T_2, P^{T_2}, is defined as

$$P^{T_2} = \langle (p_{11}^{T_2}, p_{12}^{T_2}), (p_{21}^{T_2}, p_{22}^{T_2}), \ldots, (p_{(N-1)1}^{T_2}, p_{(N-1)2}^{T_2}) \rangle \tag{7}$$

Obviously, some partitions both on T_1 and T_2 can be very skewed. However, a valid partition is expected to generate two subsets with relatively balanced element numbers in some typical graph partition methods, such as ratio cut [32]. Therefore, the partition lists P^{T_1} and P^{T_2} can be refined by ignoring skewed partitions so as to reduce the number of comparisons.

Definition 11. Let RP^{T_1} and RP^{T_2} be the refined lists, all of the elements of RP^{T_1} come from P^{T_1} , and all of the elements of RP^{T_2} come

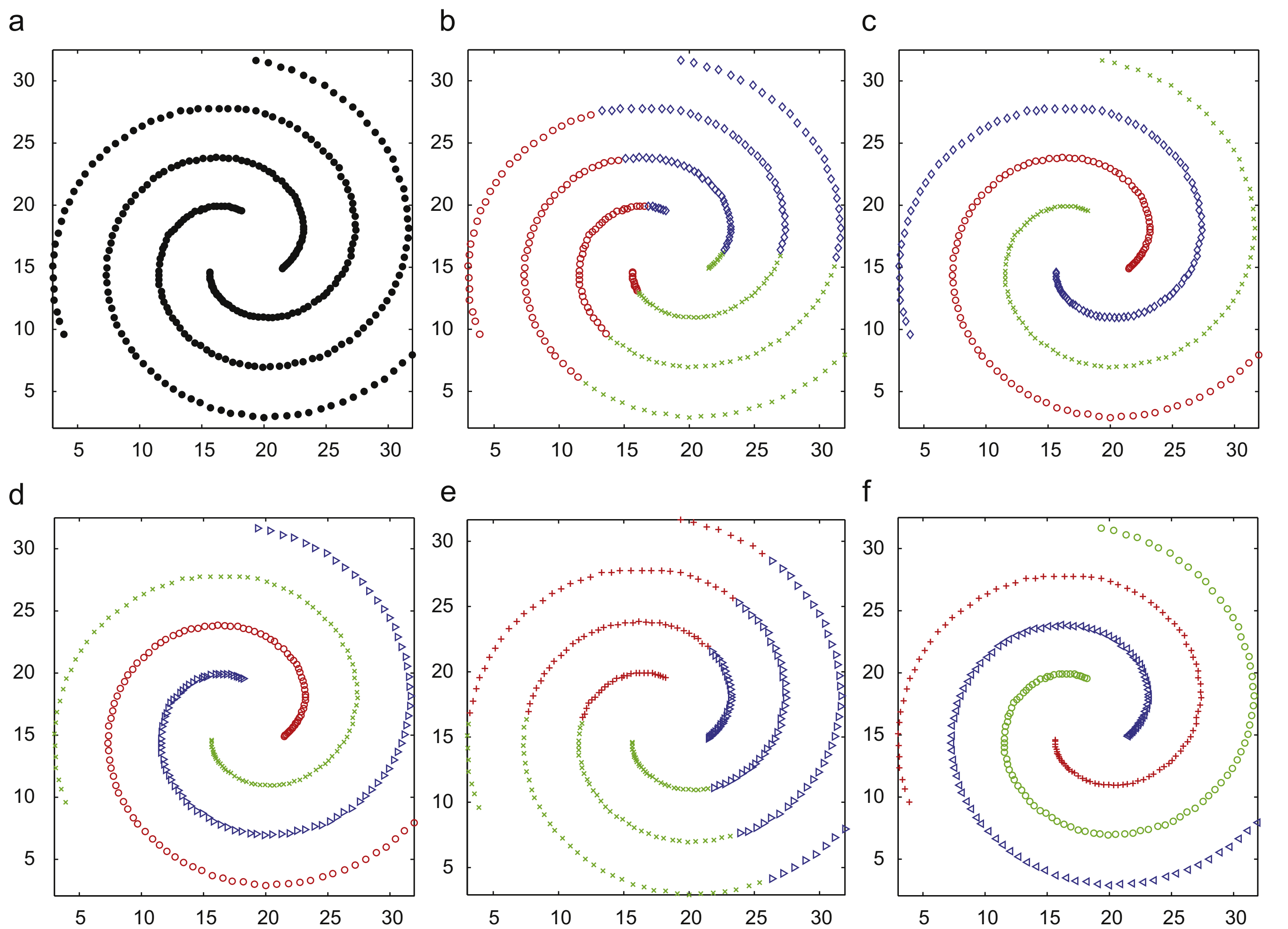

Fig. 11. Clustering results on DS1. (a) is the original dataset; (b) is the clustering result of *k*-means; (c) is the clustering result of DBScan (MinPts = 3, Eps = 1.6); (d) is the clustering result of single-linkage; (e) is the clustering result of spectral clustering; (f) is the clustering result of 2-MSTClus.

from P^{T_2}, as in

$$RP^{T_1} = \langle (rp_{11}^{T_1}, rp_{12}^{T_1}), (rp_{21}^{T_1}, rp_{22}^{T_1}), \ldots, (rp_{L1}^{T_1}, rp_{L2}^{T_1}) \rangle \quad (8)$$

$$RP^{T_2} = \langle (rp_{11}^{T_2}, rp_{12}^{T_2}), (rp_{21}^{T_2}, rp_{22}^{T_2}), \ldots, (rp_{M1}^{T_2}, rp_{M2}^{T_2}) \rangle \quad (9)$$

where $L \le N-1$, $\varepsilon \le \min(|rp_{i1}^{T_1}|, |rp_{i2}^{T_1}|)/\max(|rp_{i1}^{T_1}|, |rp_{i2}^{T_1}|)$; $M \le N-1$, $\varepsilon \le \min(|rp_{i1}^{T_2}|, |rp_{i2}^{T_2}|)/\max(|rp_{i1}^{T_2}|, |rp_{i2}^{T_2}|)$, $\varepsilon \ll N$, ε will be discussed in Section 4.

In the next step, partitions in RP^{T_1} will be compared with those in RP^{T_2}. As the number of *inconsistent vertices* between two cuts must be less than or equal to the threshold β, if $||rp_{i1}^{T_1}| - |rp_{j1}^{T_2}|| > \beta$, the comparison between two partitions $(rp_{i1}^{T_1}, rp_{i2}^{T_1})$ and $(rp_{j1}^{T_2}, rp_{j2}^{T_2})$ can be skipped.

For the purpose of saving the computational cost, we can further combine the two lists RP^{T_1} and RP^{T_2}, and order them ascendingly by the element numbers of the left parts of the pairs. Only pairs which come from different MSTs and of which element number of left parts have differences not more than β will be compared.

Definition 12. Let SP be a set which consists of all the elements of RP^{T_1} and RP^{T_2}:

$$SP = \{(rp_{11}^{T_1}, rp_{12}^{T_1}), \ldots, (rp_{L1}^{T_1}, rp_{L2}^{T_1}), (rp_{11}^{T_2}, rp_{12}^{T_2}), \ldots, (rp_{M1}^{T_1}, rp_{M2}^{T_2})\} \quad (10)$$

Definition 13. For a $sp \in SP$, let $left(sp)$ denote the left part of sp, the source of sp is defined as

$$source(sp) = \begin{cases} 1 & \text{if } sp \text{ comes from } T_1 \\ 0 & \text{otherwise} \end{cases} \quad (11)$$

For example, if $sp = (rp_{11}^{T_1}, rp_{12}^{T_1})$, then $left(sp) = rp_{11}^{T_1}$ and $source(sp) = 1$.

Definition 14. Let $CP(SP) = \langle (cp_{11}, cp_{12}), \ldots, (cp_{(L+M)1}, cp_{(L+M)2}) \rangle$ be the ordered list as in

$$CP(SP) = \langle part\_\min(SP) \circ CP(SP - \{part\_\min(SP)\}) \rangle \quad (12)$$

where $part\_\min(SP) = \arg\min_{sp \in SP} |left(sp)|$, and $\circ$ is a concatenate operator.

Definition 15. Two partitions (cp_{i1}, cp_{i2}) and (cp_{j1}, cp_{j2}) are said to be similar, where $i \le j$, if the followings hold:

(a) $source((cp_{i1}, cp_{i2})) \ne source((cp_{j1}, cp_{j2}))$;
(b) $|cp_{i1} - cp_{j1}| + |cp_{j1} - cp_{i1}| \le \beta$ or $|cp_{i1} - cp_{j2}| + |cp_{j2} - cp_{i1}| \le \beta$.

The first condition indicates that the two partitions come from different MSTs, while the second condition reveals that the number of inconsistent vertices between the two partitions is sufficient small.

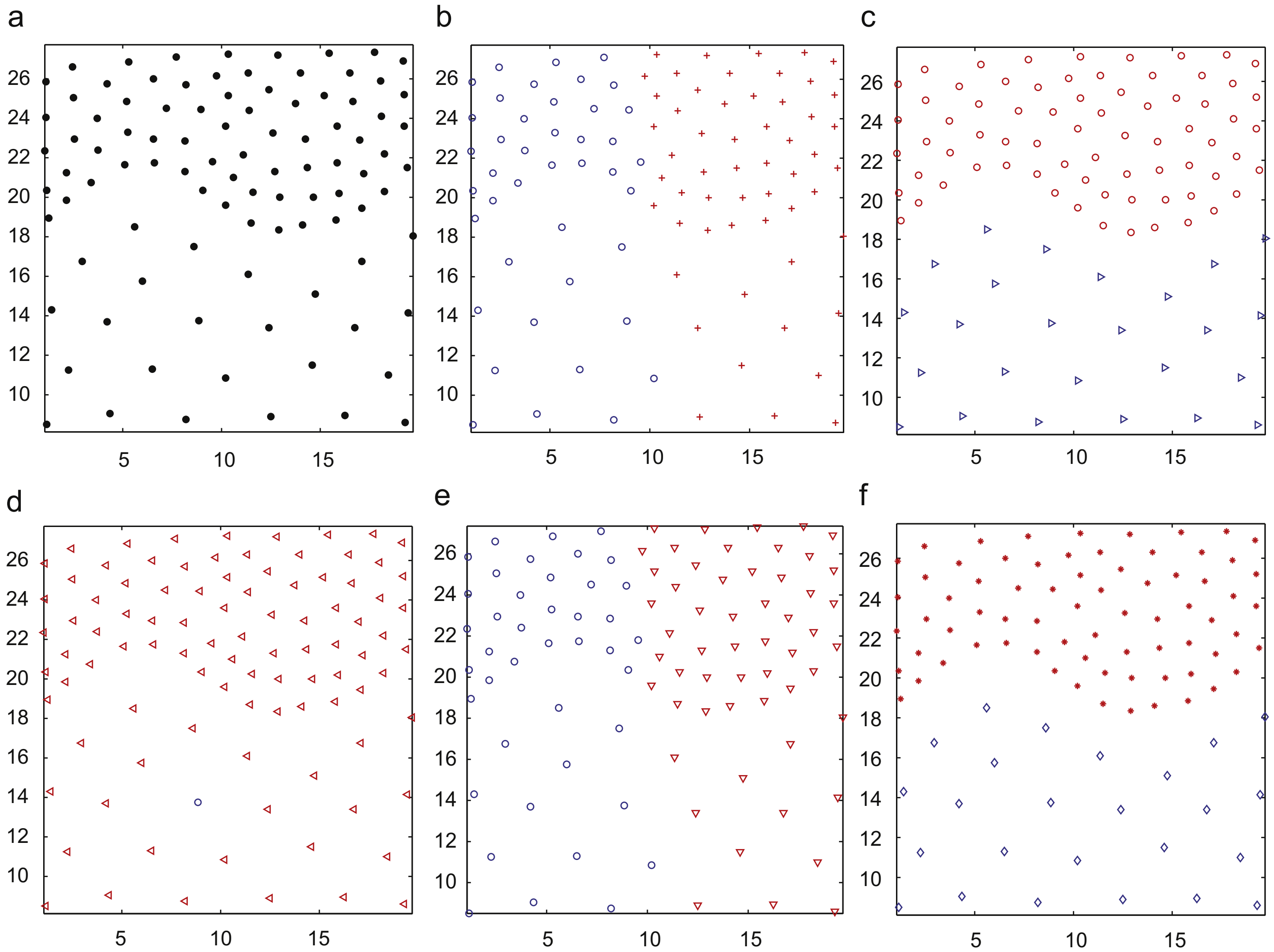

Fig. 12. Clustering results on DS2. (a) is the original dataset; (b) is the clustering result of k-means; (c) is the clustering result of DBScan (MinPts = 3, Eps = 1.9); (d) is the clustering result of single-linkage; (e) is the clustering result of spectral clustering; (f) is the clustering result of 2-MSTClus.

Algorithm 2. Clustering touching problems

Input: T_1 and T_2, the two rounds of MST of a sub-dataset generated by **Algorithm 1**.
Output: S, the set of expected partitions.
Step 1. Construct the ordered list $CP(SP)$ with T_1 and T_2; create two empty set S' and S.
Step 2. For each $(cp_{i1}, cp_{i2}) \in CP(SP)$, it is compared with $(cp_{j1}, cp_{j2}) \in CP(SP)$ and $j > i$. According to Definition 15, if there exists a partition (cp_{j1}, cp_{j2}) which is similar to (cp_{i1}, cp_{i2}), (cp_{i1}, cp_{i2}) is put into S'.
Step 3. For each $s \in S'$, if there exists a $t \in S'$, $t \neq s$, and t is similar to s, s is put into S.
Step 4. Combine similar partitions in S.

In Algorithm 2, Step 3 is to remove the unexpected partitions in terms of Observation 2. For simplicity, only those partitions without similar others are removed. In Step 3, when determining the similarity between t and s, we ignore whether they come from different MSTs or not, since at this stage only the number of inconsistent vertices are concerned. Step 4 combines the similar partitions. This can be achieved by assigning *inconsistent vertices* to two groups in terms of the evidence (support rate) accumulated from the similar partitions. Algorithm 2 can identify touching clusters except overlapping ones.

3.4. The combination of the two algorithms

As mentioned above, cluster problems are categorized into separated problems and touching problems in this paper, and the two cluster problems roughly cover all the cluster problems since they are mutual complementary. As Algorithm 1 automatically identifies separated clusters and has no effect on touching clusters, Algorithms 1 and 2 can be easily combined to deal with any cluster problem. When every subset partitioned by Algorithm 1 is fed to Algorithm 2, we will obtain the final clustering results. Therefore, the two algorithms can be easily combined to form the method 2-MSTClus.

Many traditional clustering algorithms are vulnerable to the different cluster sizes, shapes and densities. However, since the separated clusters and touching clusters can roughly cover all kinds of clusters (except overlapping clusters) in terms of definition of separated cluster and touching cluster regardless of cluster size, shape and density, the combination of Algorithms 1 and 2 is robust to diversifications of sizes, shapes and densities of clusters.

3.5. Computational complexity analysis

The computational complexity of Algorithm 1 is analyzed as follows. For a graph $G(X) = (V, E)$, if Fibonacci heaps are used to implement the min-priority queue, the running time of Prim's MST

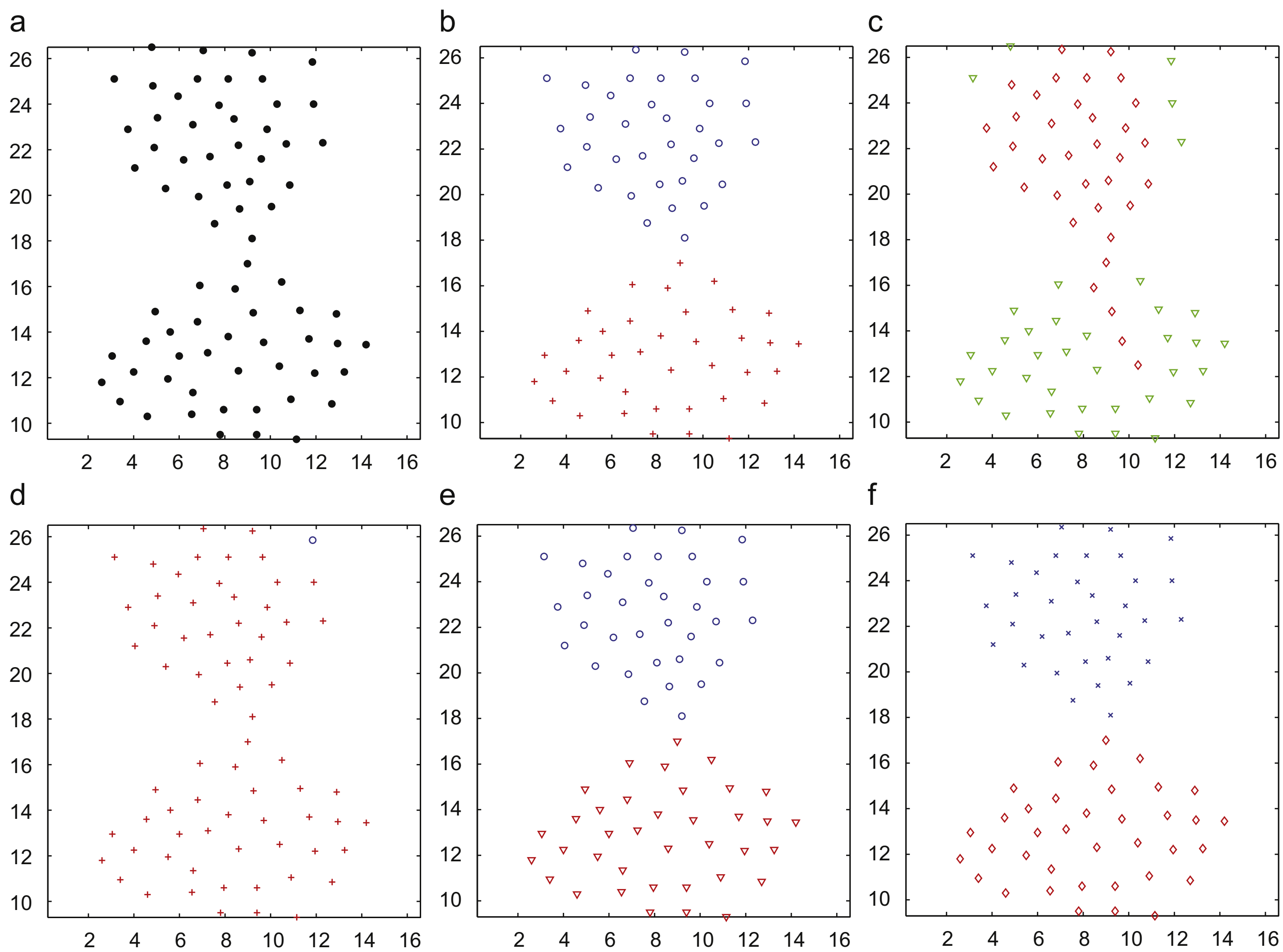

Fig. 13. Clustering results on DS3. (a) is the original dataset; (b) is the clustering result of k-means; (c) is the clustering result of DBScan (MinPts = 5, Eps = 1.5); (d) is the clustering result of single-linkage; (e) is the clustering result of spectral clustering; (f) is clustering result of 2-MSTClus.

algorithm is $O(|E| + |V|\log|V|)$ [35]. As the MST in a graph-based clustering method is generally constructed from a complete graph, $|E|$ is equal to $|V|^2$ and the computational complexity of Prim's algorithm is $O(|V|^2)$. In Step 1, accordingly, T_1 and T_2 are generated in $O(N^2)$, while Step 3 sorts the list $Rank(E')$ in $O(N\log N)$. Step 4 repeatedly removes an edge of $G'(X')$ and checks if a cut is achieved in $O(|X'|\log|X'|)$, where $|X'| \leq N$. The iteration time from Step 2 to Step 6 is the number of separated clusters in dataset, which is generally far less than N. Therefore, the time complexity of Algorithm 1 is $O(N^2)$.

In Step 1 of Algorithm 2, the constructing SP takes $O(N^2)$, and sorting $CP(SP)$ takes $O(N\log N)$. As a result, the Step 1 can be done in $O(N^2)$. The iteration in Step 2 of Algorithm 2 can be finished in $O(N\log N)$. Both Steps 3 and 4 in Algorithm 2 are executed in $O(N)$. The computational complexity of Algorithm 2 is $O(N^2)$.

Obviously, since the method 2-MSTClus is composed of Algorithms 1 and 2, its overall time complexity is $O(N^2)$.

4. Experimental results

4.1. Parameter setting

In the two proposed algorithms, although four parameters exist, they are all set to fixed values in all our experiments. The parameter λ determines whether a graph cut is valid or not when the framework deals with separated cluster problems. For the first graph cut on a separated problem, the numbers of edges removed from T_1 and T_2, respectively, are almost equal. The ideal situation is that two clusters are far away from each other, and when the first graph cut is achieved the number of the edges removed from T_1 is equal to that of the edges from T_2, i.e. $\lambda = 0.5$. For non-separated problems, on the contrary, the edges removed from T_2 are in the majority, and that leads to significantly skewed ratio. However, a change of local density may disturb the ratio. Accordingly the parameter λ is relaxed to some degree. In all of our experiments, the parameter is set to 0.330. Specially, suppose that three edge removals result in a cut, although the $Ratio(E_{gcut})$ is 0.333 and only slightly greater than the parameter, the absolute difference is very small (only 1).

When the weight of an edge is computed, the parameter δ is employed to balance the relative weight (the ratio of lengths) and the absolute weight (the length of the edge). As the relative weight is crucial, δ is generally set to 0.9.

In touching problem algorithm, the parameter β is the margin of the number of *inconsistent vertices*. If it is too small, some neck-crossed partitions may be identified as invalid, while if it is too large, some non-neck-crossed partitions may be regarded as valid. As overlapped clusters are beyond the scope of this paper, and only slightly touching clusters are considered, β is set to a relatively small value, for instance, 2 in all our experiments. In addition, some extreme skewed partitions can be ignored by the parameter ε to save

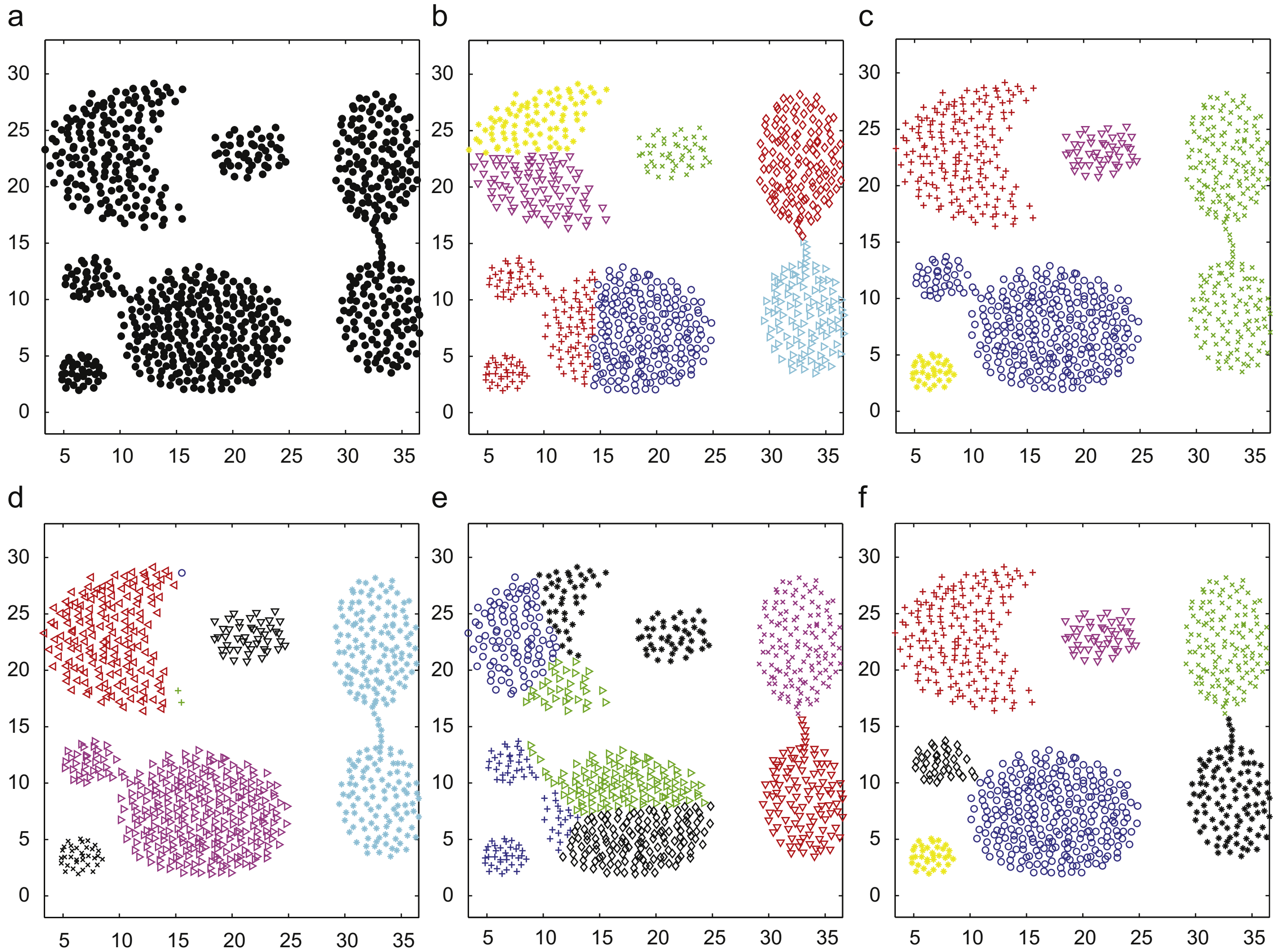

Fig. 14. Clustering results on DS4. (a) is the original dataset; (b) is the clustering result of k-means; (c) is the clustering result of DBScan (MinPts = 6, Eps = 1.5); (d) is the clustering result of single-linkage; (e) is the clustering result of spectral clustering; (f) is clustering result of 2-MSTClus.

the computational cost in touching problem algorithm. For example, cuts on some short branches (called hairs by [18]) are meaningless. Although the short branches can be identified adaptively, for simplicity we set $\varepsilon \in [0.01, 0.05]$ in the following experiments. If the number of vertices contained in a short branch is less than ε vertices, cuts on the short branch will be ignored.

4.2. Experiments on synthetic and real datasets

The proposed method 2-MSTClus is tested on five 2-D synthetic datasets, DS1–DS5, and two UCI datasets, Iris and Wine, and compared with four typical clustering algorithms, namely k-means, DBScan, single-linkage and spectral clustering, in which normalized cut is used. For DBScan, in the all experiments, the parameters are selected with the best clustering result.

The dataset DS1 is taken from [34] and illustrated in Fig. 11(a). It contains three spiral clusters, which are separated from each other in distance, therefore it is a distance-separated cluster problem. However, in terms of Handl's taxonomy [12], DS1 falls into the group of connectedness. Fig. 11(b)–(f) depict the clustering results of the four methods and 2-MSTClus. As DBScan and single-linkage prefer the datasets with connectedness, they can discover the three actual clusters, but k-means and spectral clustering cannot. 2-MSTClus can easily deal with DS1 as a separated problem and detect the three clusters.

Fig. 12 illustrates the clustering results of DS2, which is from [18]. It is a typical density-separated cluster problem. Since DBScan is a density-based and identifies clusters with the concept of density-reachable, it partitions DS2 well. However, k-means, single-linkage and spectral clustering are ineffective on DS2. While 2-MSTClus still produces ideal result by its separated algorithm.

The dataset DS3 is also taken from [18] and illustrated in Fig. 13. It is composed of two clusters, which are compact and slightly touched. k-means, which favors this kind of dataset, and spectral clustering have good performance on D3, whereas single-linkage and DBScan perform badly. Instead of the separated algorithm of 2-MSTClus, the touching algorithm of 2-MSTCLus can detect the two clusters.

In Fig. 14(a), the dataset DS4 is taken from [11]. Compared with the former three datasets DS1–DS3, this dataset is more complex. It consists of seven clusters, and is a composite cluster problem. All of the four algorithms, k-means, DBScan, single-linkage, spectral clustering fail on this dataset. However, 2-MSTClus identifies the seven clusters accurately.

In DS4, three clusters are distance-separated from others, while two pairs are internal touched. When the dataset is fed to 2-MSTClus, Algorithm 1 is first applied to it. Fig. 15(a) represents the first cut when top two weight edges are removed. As the $Ratio(E_{gcut})$ is 0.333, and greater than the threshold λ, the cut is valid. Next cut analysis is on the remaining six clusters. In Fig. 15(b), the removals of the top

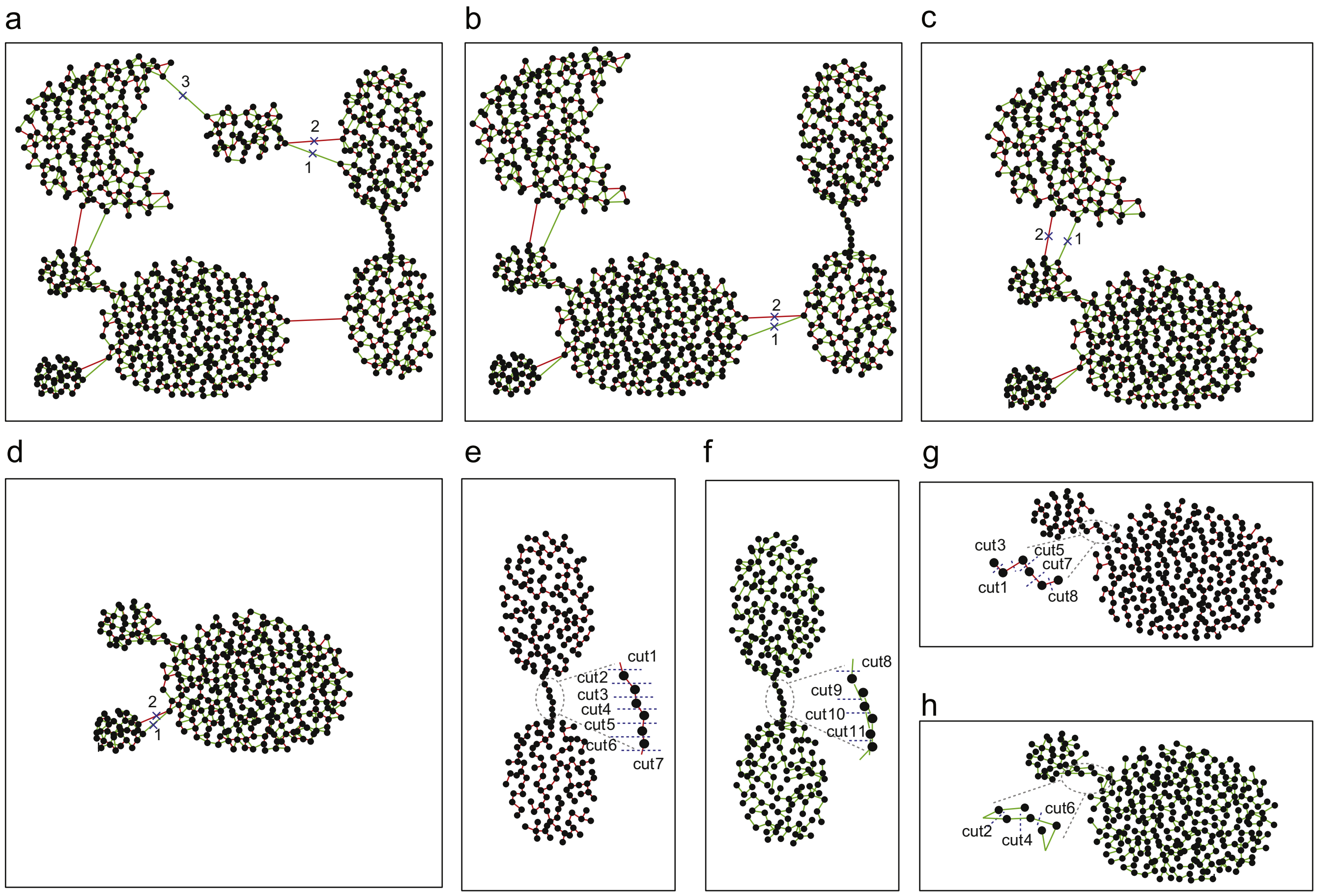

Fig. 15. Clustering process of 2-MSTClus on DS4. (a)–(d) illustrate the four graph cuts by the separated algorithm. In (e) and (f), Algorithm 2 is applied, *cut*1 and *cut*8, *cut*2 and *cut*9, *cut*3 and *cut*9, *cut*4 and *cut*9, *cut*5 and *cut*10, *cut*6 and *cut*11, *cut*7 and *cut*11,are similar ($\beta = 2$). In (g) and (h), *cut*1 and *cut*2, *cut*3 and *cut*6, *cut*5 and *cut*2, *cut*7 and *cut*2, *cut*4 and *cut*4, etc. are similar ($\beta = 2$).

two edges result in a valid cut, which partitions the six clusters into two groups. Similarly, the separated sub-dataset in Fig. 15(c) and (d) is further partitioned. Afterwards, Algorithm 1 could not partition the clusters any more. Then all the clusters are checked by Algorithm 2. Two touching clusters problems are figured out in Fig. 15(e)–(h), even though in Fig. 15(g) and (h) the two clusters have significant difference in their sizes.

The dataset DS5 is composed of three datasets from [18]. The left top dataset in Fig. 16(a) is a touching problem; the left bottom one is a distance-separated problem; while the right one is a density-separated problem. For this composite dataset, 2-MSTClus can identify the six clusters, but the four clustering methods *k*-means, single-linkage, DBScan and spectral clustering cannot.

In Fig. 17(a), (b) and (d), the distance-separated problems are identified with the removals of top two edges, respectively. With diverse densities, the two clusters in Fig. 17(c) are partitioned by Algorithm 1, and the corresponding $Ratio(E_{gcut})$ is 0.417, hence the graph cut is valid. As for the touching problem in the top left of Fig. 16(a), Algorithm 1 is ineffective. The similar cuts in Fig. 17(e) and (f) are detected by Algorithm 2.

Two real datasets from UCI are employed to test the proposed method. The first one is IRIS, which is a well-known benchmark for machine learning research. The dataset consists of three clusters with 50 samples each, and the one is well separated from the other two clusters, while the two clusters are slightly touched to each other. Similar to DS4 and DS5, it is also a composite clustering problem. When the dataset is fed to the 2-MSTClus, Algorithm 1 in the first round cuts off 50 samples, which constitute the separated cluster. Then the algorithm produces no clusters further. When Algorithm 2 is applied to the two subsets, only the cluster that is composed of 100 samples has some similar cuts between its T_1 and T_2, therefore, this cluster is further partitioned.

The performance comparison on IRIS is presented in Table 1. Four frequently-used external clustering validity indices are employed to evaluate the clustering results: Rand, Adjusted rand, Jaccard and FM. From Table 1, it is evident that 2-MSTClus performs best, since all of indices of 2-MSTClus are ranked first.

The second real dataset is WINE. It is composed of 178 samples, which fall into three clusters. From Table 2, 2-MSTClus performs only better than single-linkage. Compared with the former datasets, the performance of 2-MSTClus on WINE is slightly weakened. This is because some outliers exist in the dataset. In Algorithm 1, the graph cut criterion is a heuristic, however, the existence of outlier may affect the heuristic.

5. Discussion

Traditional MST-based clustering methods [18,19,24,33] make use of an MST to partition a dataset. A general way of partitioning is to remove the edges with relative large lengths, and one removal leads to a bipartition. Within an MST, although some crucial information of a dataset are collected, some are missed. T_1 and T_2 are combined to form a graph for the purpose of accumulating more evidence to partition datasets. In a two-round-MST based graph, a graph cut requires at least two edge removals. Only the evidence from T_1 and T_2 being consistent, is the graph cut valid. For analyzing

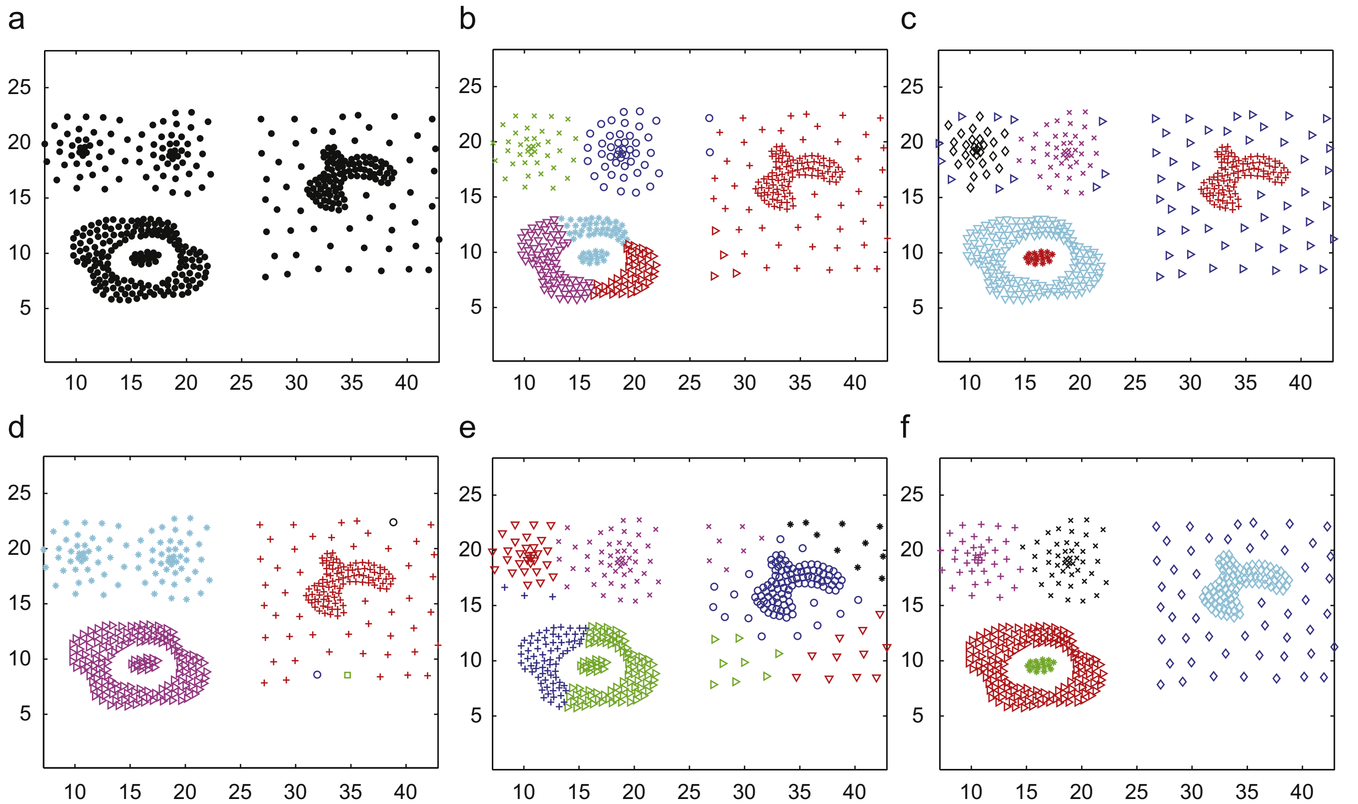

Fig. 16. Clustering results on DS5. (a) is the original dataset; (b) is the clustering result of *k*-means; (c) is the clustering result of DBScan (MinPts = 4, Eps = 1.4); (d) is the clustering result of single-linkage; (e) is the clustering result of spectral clustering; (f) is the clustering result of 2-MSTClus.

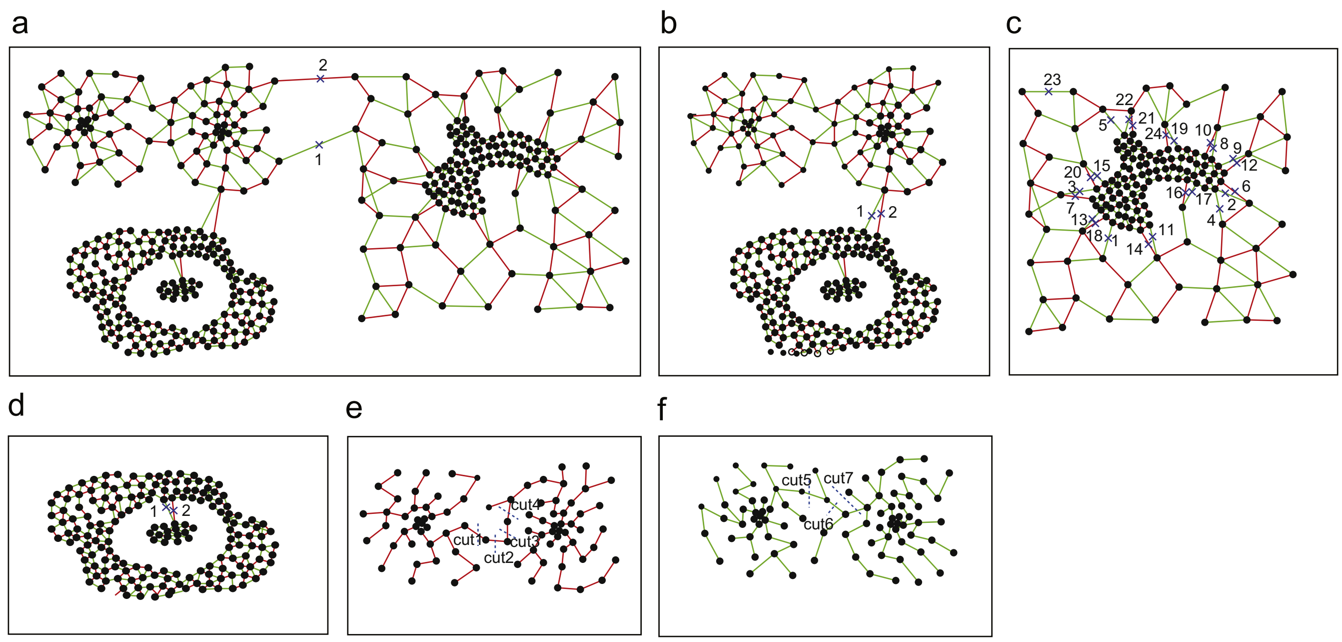

Fig. 17. Clustering process of 2-MSTClus on DS5. In (a), (b) and (d), the separated clustering method is applied. Only two edges are removed for each dataset and three valid graph cuts are achieved. (c) illustrates the partitioning process of the density-separated cluster problem. Totally 24 edges are removed for the graph cut, from which 14 edges come from T_2 and 10 from T_1. In (e) and (f), *cut*1 and *cut*5, *cut*1 and *cut*6, *cut*2 and *cut*7, *cut*3 and *cut*7, *cut*4 and *cut*7 are similar ($\beta = 2$).

the touching problems in 2-MSTClus, the concept of *inconsistent vertices* delivers the same idea.

The proposed method 2-MSTClus deals with a dataset in terms of which cluster problem it belongs to, separated problem or touching problem. The diversifications of sizes, shapes as well as densities of clusters have no effect on the clustering process.

A drawback of 2-MSTClus is that it is not robust to outliers. Although some outlier detection methods can be used to preprocess a dataset and remedy this drawback, we will discover more robust mechanism to outliers based on two-round-MST based graph in the future. In addition, the proposed method cannot detect the overlapping clusters. If a dataset composed of two overlapping clusters is

Table 1
Performance comparison on IRIS data.

Method	Rand	Adjusted rand	Jaccard	FM
k-Means	0.8797	0.7302	0.6959	0.8208
DBScan	0.8834	0.7388	0.7044	0.8268
Single-linkage	0.7766	0.5638	0.5891	0.7635
Spectral clustering	0.7998	0.5468	0.5334	0.6957
2-MSTClus	0.9341	0.8512	0.8188	0.9004

Table 2
Performance comparison on WINE data.

Method	Rand	Adjusted rand	Jaccard	FM
k-Means	0.7183	0.3711	0.4120	0.7302
DBScan	0.7610	0.5291	0.5902	0.7512
Single-linkage	0.3628	0.0054	0.3325	0.5650
Spectral clustering	0.7644	0.4713	0.4798	0.6485
2-MSTClus	0.7173	0.3676	0.4094	0.5809

dealt with 2-MSTClus, the two clusters will be recognized as one cluster.

If more MSTs are combined, for instance, $T_1, T_2, T_3, \ldots, T_k, k \leq N/2$, does the performance of the proposed method become better? In other words, how is a suitable k selected for a dataset? This is an interesting problem for the future work.

6. Conclusions

In this paper, a two-round-MST based graph is utilized to represent a dataset, and a clustering method 2-MSTClus is proposed. The method makes use of the good properties of the two-round-MST based graph, automatically differentiates separated problems from touching problems, and deals with the two kinds of cluster problem. It does not request user-defined cluster number, and is robust to different cluster shapes, densities and sizes. Our future work will focus on improving the robustness of 2-MSTClus to outliers and selecting a reasonable k for constructing k-MST.

Acknowledgments

We would like to thank the anonymous reviewers whose thoughtful comments and suggestions improved the quality of this paper. The paper is supported by the National Natural Science Foundation of China, Grant No. 60775036, No. 60475019, and the Research Fund for the Doctoral Program of Higher Education: No. 20060247039.

References

[1] W. Cai, S. Chen, D. Zhang, Fast and robust fuzzy c-means clustering algorithms incorporating local information for image segmentation, Pattern Recognition 40 (2007) 825–838.
[2] Z. Wu, R. Leahy, An optimal graph theoretic approach to data clustering: theory and its application to image segmentation, IEEE Transactions on Pattern Analysis and Machine Intelligence 15 (1993) 1101–1113.
[3] J. Han, M. Kamber, Data Mining: Concepts and Techniques, Morgan-Kaufman, San Francisco, 2006.
[4] Z. Yu, H.S. Wong, H. Wang, Graph-based consensus clustering for class discovery from gene expression data, Bioinformatics 23 (2007) 2888–2896.
[5] S. Bandyopadhyay, A. Mukhopadhyay, U. Maulik, An improved algorithm for clustering gene expression data, Bioinformatics 23 (2007) 2859–2865.
[6] A.K. Jain, M.C. Law, Data clustering: a user's dilemma, Pattern Recognition and Machine Intelligence, Lecture Notes in Computer Science, 3776, Springer, Berlin, Heidelberg, 2005, pp. 1–10.
[7] R. Xu, D. Wunsch II, Survey of clustering algorithms, IEEE Transactions on Neural Networks 16 (2005) 645–678.
[8] J. Kleinberg, An Impossibility Theorem for Clustering, MIT Press, Cambridge, MA, USA, 2002.
[9] H.G. Ayad, M.S. Kamel, Cumulative voting consensus method for partitions with variable number of clusters, IEEE Transactions on Pattern Analysis and Machine Intelligence 30 (2008) 160–173.
[10] A. Topchy, A.K. Jain, W. Punch, Clustering ensembles: models of consensus and weak partitions, IEEE Transactions on Pattern Analysis and Machine Intelligence 27 (2005) 1866–1881.
[11] A. Gionis, H. Annila, P. Tsaparas, Clustering aggregation, ACM Transactions on Knowledge Discovery from Data 1 (2007) 1–30.
[12] J. Handl, J. Knowles, An evolutionary approach to multiobjective clustering, IEEE Transactions on Evolutionary Computation 11 (2007) 56–76.
[13] A.L. Fred, A.K. Jain, Combining multiple clusterings using evidence accumulation, IEEE Transactions on Pattern Analysis and Machine Intelligence 27 (2005) 835–850.
[14] N. Bansal, A. Blum, S. Chawla, Correlation clustering, Machine Learning 56 (2004) 89–113.
[15] G.V. Nosovskiy, D. Liu, O. Sourina, Automatic clustering and boundary detection algorithm based on adaptive influence function, Pattern Recognition 41 (2008) 2757–2776.
[16] G. Karypis, E.H. Han, V. Kumar, CHAMELEON: a hierarchical clustering algorithm using dynamic modeling, IEEE Transactions on Computers 32 (1999) 68–75.
[17] P. Fränti, O. Virmajoki, V. Hautamäki, Fast agglomerative clustering using a k-nearest neighbor graph, IEEE Transactions on Pattern Analysis and Machine Intelligence 28 (2006) 1875–1881.
[18] C.T. Zahn, Graph-theoretical methods for detecting and describing gestalt clusters, IEEE Transactions on Computers C-20 (1971) 68–86.
[19] Y. Xu, V. Olman, D. Xu, Clustering gene expression data using a graph-theoretic approach: an application of minimum spanning tree, Bioinformatics 18 (2002) 536–545.
[20] J.M. González-Barrios, A.J. Quiroz, A clustering procedure based on the comparison between the k nearest neighbors graph and the minimal spanning tree, Statistics & Probability Letters 62 (2003) 23–34.
[21] N. Päivinen, Clustering with a minimum spanning tree of scale-free-like structure, Pattern Recognition Letters 26 (2005) 921–930.
[22] J. Shi, J. Malik, Normalized cuts and image segmentation, IEEE Transactions on Pattern Analysis and Machine Intelligence 22 (2000) 888–905.
[23] C.H. Lee, et al., Clustering high dimensional data: a graph-based relaxed optimization approach, Information Sciences 178 (2008) 4501–4511.
[24] S. Bandyopadhyay, An automatic shape independent clustering technique, Pattern Recognition 37 (2004) 33–45.
[25] G.T. Toussaint, The relative neighborhood graph of a finite planar set, Pattern Recognition 12 (1980) 261–268.
[26] J. Handl, J. Knowles, D.B. Kell, Computational cluster validation in post-genomic data analysis, Bioinformatics 21 (2005) 3201–3212.
[27] M. Steinbach, L. Ertöz, V. Kumar, The Challenges of Clustering High Dimensional Data, New Directions in Statistical Physics: Bioinformatics and Pattern Recognition, L.T. Wille (Ed.), Springer, Berlin, 2002, pp. 273–307.
[28] A.K. Jain, R.C. Dubes, Algorithms for Clustering Data, Prentice-Hall, Englewood Cliffs, NJ, 1988.
[29] B. Fischer, J.M. Buhmann, Path-based clustering for grouping of smooth curves and texture segmentation, IEEE Transactions on Pattern Analysis and Machine Intelligence 25 (2003) 513–518.
[30] L. Yang, K-edge connected neighborhood graph for geodesic distance estimation and nonlinear data projection, in: Proceedings of the 17th International Conference on Pattern Recognition (ICPR'04), 2004.
[31] U. Brandes, M. Gaertler, D. Wagner, Engineering graph clustering: models and experimental evaluation, ACM Journal of Experimental Algorithmics 12 (2007).
[32] L. Hagen, A.B. Kahng, New spectral methods for ratio cut partitioning and clustering, IEEE Transactions on Computer-Aided Design 11 (1992) 1074–1085.
[33] Y. Li, A clustering algorithm based on maximal θ-distant subtrees, Pattern Recognition 40 (2007) 1425–1431.
[34] H. Chang, D.Y. Yeung, Robust path-based spectral clustering, Pattern Recognition 41 (2008) 191–203.
[35] T.H. Corman, C.E. Leiserson, R.L. Rivest, C. Stein, Introduction to Algorithms, second ed., MIT press, Cambridge, MA, 2001.

About the Author—CAIMING ZHONG is currently pursuing his Ph.D. in Computer Sciences at Tongji University, Shanghai, China. His research interests include cluster analysis, manifold learning and image segmentation.

About the Author—DUOQIAN MIAO is a professor of Department of Computer Science and Technology at Tongji University, Shanghai, China. He has published more than 40 papers in international proceedings and journals. His research interests include soft computing, rough sets, pattern recognition, data mining, machine learning and granular computing.

About the Author—RUIZHI WANG is a Ph.D. candidate of Computer Sciences at Tongji University of China. Her research interests include data mining, statistical pattern recognition, and bioinformatics. She is currently working on biclustering algorithms and their applications to various tasks in gene expression data analysis.