Physica A 460 (2016) 188–200

Contents lists available at ScienceDirect

Physica A

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

Spatial clustering with Density-Ordered tree

Qing Cheng [a,*], Xin Lu [b,c,d,e], Zhong Liu [a], Jincai Huang [a], Guangquan Cheng [a]

[a] Science and Technology on Information Systems Engineering Laboratory, National University of Defense Technology, Changsha 410073, PR China

[b] College of Information System and Management, National University of Defense Technology, Changsha 410073, PR China

[c] Flowminder Foundation, Stockholm 17177, Sweden

[d] Department of Public Health Sciences, Karolinska Institutet, Stockholm 17177, Sweden

[e] Division of Infectious Disease, Key Laboratory of Surveillance and Early-Warning on Infectious Disease, Chinese Centre for Disease Control and Prevention, Beijing 102206, PR China

HIGHLIGHTS

- We develop a Density-Ordered tree to represent the original data. It efficiently integrates information on not only distance but also density between data points.
- Our method can effectively identify clusters with diverse shapes and densities for spatial dataset.
- Our method provides an innovative way to identify noise and cluster center.
- Experiments demonstrate that our method is more effective than DBSCAN and Chameleon.

ARTICLE INFO

Article history:
Received 26 October 2015
Received in revised form 28 March 2016
Available online 14 May 2016

Keywords:
Spatial clustering
Diverse shapes and densities
Density-Ordered tree
Partition-and-merge strategy

ABSTRACT

Clustering has emerged as an active research direction for knowledge discovery in spatial databases. Most spatial clustering methods become ineffective when inappropriate parameters are given or when datasets of diverse shapes and densities are provided. To address this issue, we propose a novel clustering method, called SCDOT (Spatial Clustering with Density-Ordered Tree). By projecting a dataset to a Density-Ordered Tree, SCDOT partitions the data into several relatively small sub-clusters with a box-plot method. A heuristic method is proposed to find the genuine clusters by repeatedly merging sub-clusters and an iteration strategy is utilized to automatically determine input parameters. Moreover, we also provide an innovative way to identify cluster center and noise. Extensive experiments on both synthetic and real-world datasets demonstrate the superior performance of SCDOT over the baseline methods.

1. Introduction

Spatial data mining becomes more and more important in the analysis of spatial databases since increasingly large amounts of data obtained from satellite images, X-ray crystallography or other automatic equipment are stored in spatial databases. Spatial clustering, which groups similar spatial objects into classes, is an important component of spatial data mining. Spatial clustering can be used as a tool to get insight into the distribution of data, to observe the characteristics of

* Corresponding author.
E-mail address: sgggps@163.com (Q. Cheng).

http://dx.doi.org/10.1016/j.physa.2016.05.041
0378-4371/

each cluster, and to focus on a particular set of clusters for further analysis. It has been applied in diverse fields, such as spatial epidemiology, landscape ecology, crime analysis, disease surveillance and population genetics [1–4].

Due to its wide applications in various areas, many studies on spatial clustering have been conducted [5,6], but most of them become ineffective when inappropriate parameters are given or when spatial datasets with diverse shapes, sizes and densities are provided. For example, partition clustering methods, such as K-means [7] and K-medoids [8], are very sensitive to noise and cannot detect arbitrary shaped clusters. Some advanced density-methods are applied in spatial clustering, such as DBSCAN [9]. DBSCAN relies on a density-based notion of clusters which is designed to discover clusters of arbitrary shape. However, the performance of DBSCAN depends on two specified parameters and it does not perform well for datasets with varying densities. Several variants of hierarchical clustering methods were proposed to effectively handle spatial dataset with different shapes and densities, such as Chameleon [10] and SAM [11], however, similar to DBSCAN, all these methods still require some pre-determined parameters as inputs. In fact, without enough prior knowledge, appropriate parameters are often not known in advance when dealing with large databases. This drawback also exists in some shared nearest neighbor approaches, such as SNN [12]. Thus, without specifying any parameters related to respective datasets, it remains a challenge to identify clusters of different shapes and densities in spatial clustering.

In this paper, we propose a novel clustering algorithm, namely, Spatial Clustering with Density-Ordered Tree (SCDOT), which is capable of detecting clusters of different shapes and densities, in detail, we construct a Density-Ordered tree (DOT) to represent original data by combining density and distance. Then, a split-and-merge strategy is applied to DOT for identifying clusters with diverse shapes and densities, and an iteration optimization is utilized to automatically determine input parameters. To the best of our knowledge, this is the first work that employs the density-based clustering, partition and hierarchal clustering principle. The main contributions of this paper are summarized as follows:

1. We develop a Density-Ordered tree method to represent the original data. The method efficiently reflects not only distance relation but also density relation among data points. Based on this method, we convert the cluster discovery problem into a Density-Ordered tree partitioning and merging problem.
2. Our method provides an innovative way to identify noise and cluster center. After Density-Ordered tree being divided into many sub-trees, cluster centers are recognized as roots of sub-trees, and noises are recognized as the anomalous leaves.
3. The new clustering method can effectively identify clusters with diverse shapes and densities for spatial dataset. Given the number of clusters to be detected, there is no need to tune other parameters by the user.

2. Related work

The problem of detecting clusters has been extensively studied for years, in this paper we focus on the research of spatial clustering. An distinct attribute of spatial data is that it contains spatial information and are composed of clusters with diverse shapes, sizes and densities. In this section, a brief review of spatial clustering methods is given.

The partition clustering method splits a dataset at once using an objective function, K-means [7] is one of the most popular examples of partition clustering, it employs mean-squared-error as its objective function: find partitions such that the sum of square error between empirical mean of a cluster and points in that cluster is minimized. Many variations of K-means were proposed, such as K-medoids [8], K-medians clustering [13], K-means++ [14], Fuzzy c-means [15]. CLARANS (clustering large applications based on randomized search) [5] is another type of partition clustering method which works well on large dataset. Partition clustering produces inaccurate results when the objective function used does not capture the intrinsic structure of the data [11]. This limitation makes partition clustering incapable of handling clusters of arbitrary shapes, distinct sizes and densities. Similar limitation exists for distribution-based methods, such as EM algorithm for clustering [16], one attempts to reproduce the observed realization of data points as a mix of predefined probability distribution functions, the accuracy of such methods depends on the capability of the trial probability to represent the data. In particular, for many real datasets, there may be no concisely predefined probability.

Density-based clustering methods assume clusters as dense regions of objects in the data space and are separated by regions of low density. They are much powerful for filtering outliers and discovering arbitrary shaped clusters compared with partition and distribution-based clustering methods. DBSCAN [9] is a well-known density-based clustering algorithm. It judges the density around the neighborhood of an object to be sufficiently dense if the number of data points within a distance *eps* of the object is greater than *MinPts* number of points. It can discover clusters of arbitrary shapes and sizes. The main weakness of DBSCAN is that the performance is poor when clusters have greatly varied densities and users need to set suitable parameters in order to get qualified result. Additionally, many varieties of DBSCAN, such as OPTICS (Ordering Points to Identify the Clustering Structure) [17], l-DBSCAN [18], rough-DBSCAN [19], S-T DBSCAN [20], DENCLUE [21], are proposed to improve clustering performance in different perspectives, but pre-defined parameters, such as *MinPts* and *eps*, are still needed from users.

More recently, Rodriguez and Laio [22] proposed a clustering method based on the idea that cluster centers are characterized by higher density than their neighbors and by relatively large distance from points with higher densities. Similar to the K-medoids method, it has its basis only in the distance between data points. Like DBSCAN, it is able to detect nonspherical clusters and to automatically find adequate number of clusters. It provided not only a new way for combining density and distance to identify clusters with diverse shapes and densities but also a novel criterion for the automatic choice of cluster centers. We have used this method to solve the problem of community discovery [23].

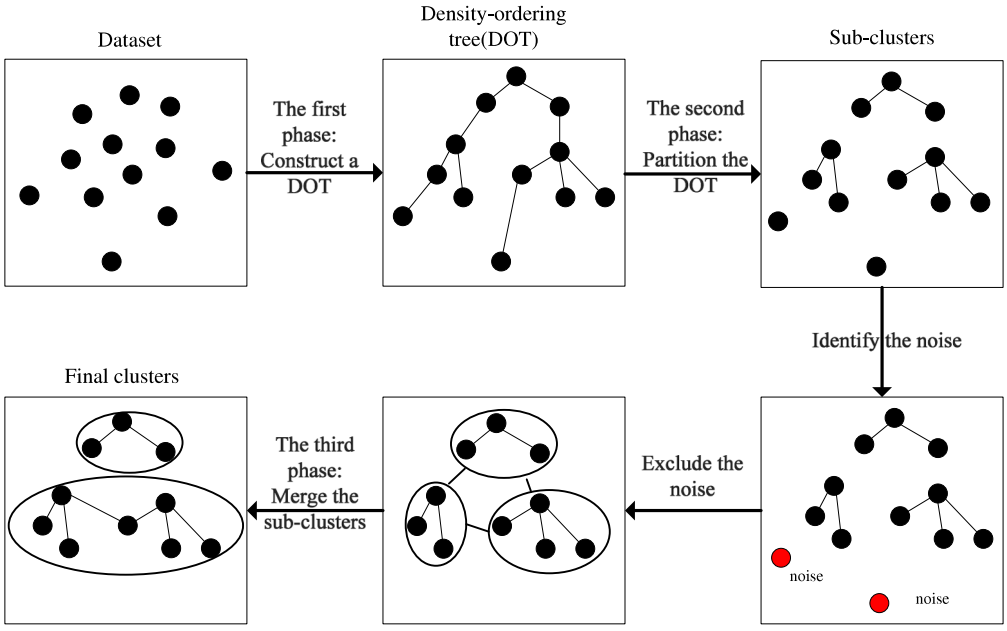

Fig. 1. SCDOT uses a three-phase algorithm, which first constructs a Density-Ordered tree and then partitions the Density-Ordered tree into sub-clusters. Finally, we use a heuristic method to find the genuine clusters by repeatedly merging these sub-clusters.

Hierarchical algorithms successively split (divisive) or merge (agglomerative) groups to form a hierarchy of clusters based on a specified measure of distance or similarity between objects. To enhance the effectiveness of hierarchical clustering, recent methods have adopted one of the following two approaches. The first approach represented by algorithms like BIRCH [24] is to obtain an initial result by using a hierarchical agglomerative algorithm and then refining the result using iterative relocation. The second approach represented by hybrid methods that unite the pros of hierarchical and partition clustering [10,11,25–27], such as CURE [25], Chameleon [10], SAM (minimum spanning tree based split-and-merge method) [11]. These hybrid methods analyze the dataset in two stages. In the first stage the dataset is divided by partitioning algorithm and in the second stage merging is done by similarity measures. However, many empirical parameters are involved, yet without enough prior knowledge, these parameters are difficult to be determined. This drawback also exists in some shared nearest neighbor approaches, such as ROCK [28], SNN [12]. Despite this limitation, the split-and-merge strategy is effective in discovering clusters with diverse shapes and densities, hereby, we integrate this strategy in our work; in addition, to overcome the limitation of these methods, i.e., model performance depends on many pre-specified parameters, we provide a heuristic method to select parameters automatically. A detailed description of our method is provided in the next section.

3. Materials and methods

In this section, we propose a novel clustering algorithm, called Spatial Clustering with Density-Ordered Tree (SCDOT), to overcome difficulties in detecting clusters with different shapes and densities. SCDOT requires no parameter except the number of clusters. There are three phases in SCDOT. In the first phase, for each data point, we use a Density-Ordered Tree (DOT) to represent the original dataset. Unlike the minimal spanning tree, DOT efficiently integrates information on not only distance but also density between data points; In the second phase, a box-plot method is introduced to partition the DOT into sub-clusters and to identify noises; In the third phase, similar to Chameleon [10], we provide a heuristic method to find the genuine clusters by repeatedly merging sub-clusters. The key difference between SCDOT and Chameleon is that we introduce the concept of disconnectivity to determine the similarity between pairs of clusters, rather than relative interconnectivity and relative closeness, and unlike Chameleon requires setting the user-specified thresholds, SCDOT is able to select parameters automatically through a heuristic method. The above process is displayed in Fig. 1.

3.1. Construct the Density-Ordered Tree

Definition 1. The local density ρ_i of data point i is defined as

$$\rho_i = 1 \Big/ \sum_{j=1}^{k} d(i, Ne_j(i)) \tag{1}$$

where $d(i, j)$ represents the distance between node i and j, $Ne_j(i)$ represents the jth-nearest neighbor of node i.

Remark 1. Intuitively, the local density of a point i is the inverse of the distance based on the k-nearest neighbors of i. Note that the local density can be ∞ if all distances in the summation are 0. This may occur for a point i if there are at least k points, different from i, but sharing the same spatial coordinates, i.e. if there are at least k duplicates of i in the dataset. For simplicity, we will not handle this case explicitly but simply assume that there are no duplicates. (To deal with duplicates, we can extend our notion of neighborhood on local density, with the additional requirement that there have to be at least k points with different spatial coordinates.)

With the definition of local density, the density order of points can be denoted as

$$\rho_1 \leq \rho_2 \leq \cdots \leq \rho_n. \tag{2}$$

Remark 2. If $\rho_{n-1} = \rho_n$, we randomly choose a node and add a small value to its density, to ensure that there is only one node with the maximum density. Thus, the density order of points is changed to

$$\rho_1 \leq \rho_2 \leq \cdots < \rho_n. \tag{3}$$

The distance δ_i of node i is measured by computing the minimum distance between the node i and any other node with higher density [22]:

$$\delta_i = \min_{j:\rho_j > \rho_i} (d(i, j)). \tag{4}$$

If the node is with highest density, we conventionally take $\delta_i = \max_j(d(i, j))$.

Based on the assumption that each point is assigned to the same cluster as its nearest neighbor of higher density [22]. A graph is built such that nodes are data points, and for each node i (excluding the node of maximal density), we connect it with its nearest neighbor of higher density and set the edge weight with δ_i according to Eq. (4). This graph is denoted as G.

Remark 3. For each data point i, if the number of its nearest neighbor of higher density is more than one, we randomly choose one to connect it with.

We will provide basic lemmas which are necessary for the understanding of G.

Lemma 1. *Let C be a collection of data points, and $n = |C|$ be the number of the data points. Let G be a clustering graph for the C. Thus G has $n - 1$ edges.*

Proof. For each point in the C, we connect it with its nearest neighbor of higher density, but for the point of maxima density, there is no point with the higher density than it, thus, the edges in the G is $n - 1$.

Lemma 2. *Let C be a collection of data points, G be a clustering graph for C. Thus G is connected.*

Proof. If G is not connected, assume that components of G are G_1 and G_2, so for G_1, there exists a node (v_i) with highest density, its density is denoted as ρ_i, similarly, there exists a node (v_j) with highest density in the G_2, its density is denoted as ρ_j. Based on Remark 2, there is only one node with highest density in G, such that $\rho_i \neq \rho_j$. Moreover, according to connecting a node with its nearest neighbor of higher density in the G, if $\rho_i > \rho_j$, in G_1 there must be a node connected to v_j; if $\rho_i < \rho_j$, in G_2 there must be a node connected to v_i, thereby, G_1 and G_2 belong to exactly one component of G, this claim is in contradiction with the assumption that G is not connected. Accordingly, it is impossible that the G has more than one component, so G is connected.

Lemma 3. *Let C be a collection of data points, G be a clustering graph for the C. Thus G has no simple cycles.*

Proof. If there exists a cycle C' for G such that C' consists of a sequence of nodes $v_1, v_2 \cdots v_m$, where m is the number of nodes in C'. If v_i is the node with smallest density in the C' ($i \leq m$), such that $\rho_i < \rho_{i+1}$ and $\rho_i < \rho_{i-1}$. It is impossible that v_{i+1} and v_{i-1} both connect with v_i due to the constraint that each node can only connect with one node of higher density. Meanwhile, it is impossible that v_i connects v_{i+1} and v_{i-1} simultaneously according to Remark 3. Thus, C' is not a closed walk since there exists a cut between v_i and v_{i+1} or v_i and v_{i-1}, which is in contradiction with C' is a cycle C'. Consequently, there is no cycle in G.

Lemmas 1–3 suggest that G is a tree, each node's density is smaller than its parent and larger than its children, the root node is the node with the highest density. We will call G the Density-Ordered Tree (DOT) because the construction of DOT not only depends on distance but also depends on density between nodes.

Since DOT is a tree, partition dataset into sub-clusters is equal to partition DOT into sub-trees by removing edges (see Section 3.2). Fig. 2(b) illustrates DOT construction with a simple example (let $k = 2$, node i is denoted as v_i). In this example, it is obvious that $\rho_1 > \rho_2 > \rho_5 > \rho_4 > \rho_3$, v_1, v_2, v_5 have higher density than node v_4 and $d(5, 4) = d(2, 4) > d(1, 4)$, so we connect v_4 and v_1.

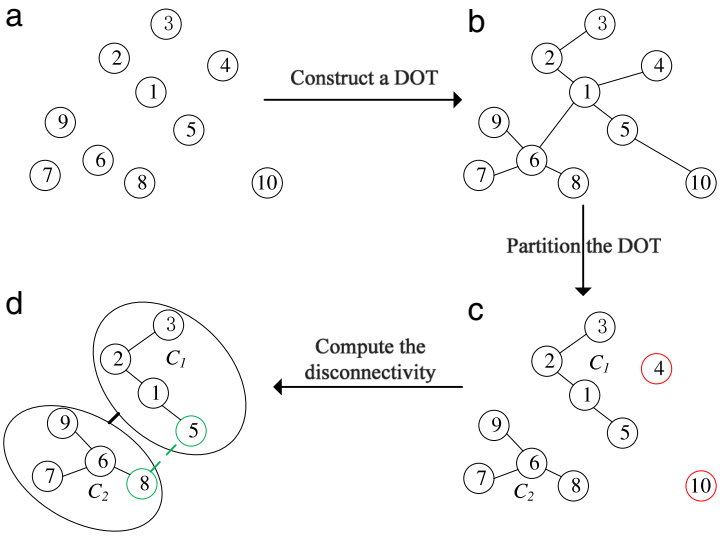

Fig. 2. A schematic illustration of SCDOT.

3.2. Partition Density-Ordered tree into sub-clusters

After the DOT has been constructed, SCDOT partitions DOT into several relatively small sub-clusters. For the DOT, it is noteworthy that δ (edge weight) is much larger than the typical nearest neighbor distance only for points that are local or global maxima in the density. Here, cluster centers are recognized as points for which the value of δ is anomalously large. After cluster centers have been found, each remaining point is assigned to the same cluster as its nearest neighbor of higher density. Actually, this process corresponds to partition DOT into sub-trees by removing edges with anomalously large weight, such that every sub-tree indicates a cluster and the root of the sub-tree is the cluster center. In this paper, we use a box-plot method to detect the anomalously large edges for partitioning DOT.

Box-plot displays variation in samples of a statistical population without making any assumptions of the underlying statistical distribution. We take edge weights of a tree (e.g. SCDOT) as a dataset box-plot displays the distribution of this dataset based on the five number summary: minimum, first quartile (denoted as Q1), median (also called Q2), third quartile (denoted as Q3), and maximum [29]. The interquartile range (IQR) is represented by the width of the box (Q3 minus Q1) as Fig. 3. Based on these, we define two types of *outlier-edges*:

1. *Extreme outlier*-edges are edges with weights equal or above $3 \times \text{IQR} + \text{Q3}$.
2. *Mild outlier-edges* are edges with weights equal or above $1.5 \times \text{IQR} + \text{Q3}$ and below the $3 \times \text{IQR} + \text{Q3}$.

Let g denote the desired number of clusters, and the DOT partition is likely to have more clusters than the desired number of clusters. The steps of DOT partition are as below:

step 1. We take edge weights of DOT as a dataset, by utilizing the box-plot method, extreme outlier-edges are found and removed from DOT to form sub-trees. Each sub-tree's edge weights is then considered as a dataset and the above process is repeated until no extreme outlier-edges can be found. If the sub-tree count is equal or larger than g, then stop. Otherwise, go to step 2.
step 2. For each sub-tree, we also use the box-plot method to find the mild outlier-edges for its edge weights, we get the new set of sub-trees after removing the mild outlier-edges. Similarly, for each new sub-tree, we repeat this process until no mild outlier-edges can be found in all sub-trees. If the number of sub-trees is more than or equal to g, then stop. Otherwise, go to step 3.
step 3. Repeatedly remove the edge with the maxima weight until the number of sub-tree count is g.

Remark 4. During steps above, if a sub-trees contains only one node, it indicates that this node has a relatively high distance to other clusters. Hence, it can be considered as a cluster composed of a single point, namely, noise. In fact, small sub-trees can be regarded as noise, however, for the purpose of this paper, we assume that sub-tree composed of a single point is noise.

In the above steps of SCDOT all nodes are partitioned into sub-trees (sub-clusters). In addition, the root node of each sub-tree is the center for each sub-clusters and has the highest density in sub-clusters. For example, in Fig. 2(b). We find that the edge(v_5, v_{10}) and edge(v_1, v_6) are extreme outlier-edges, after removing them, we can get two sub-clusters (the top sub-clusters denoted as C_1, and the bottom sub-clusters denoted as C_2), in which v_1 is the center of C_1, v_6 is the center of C_2,

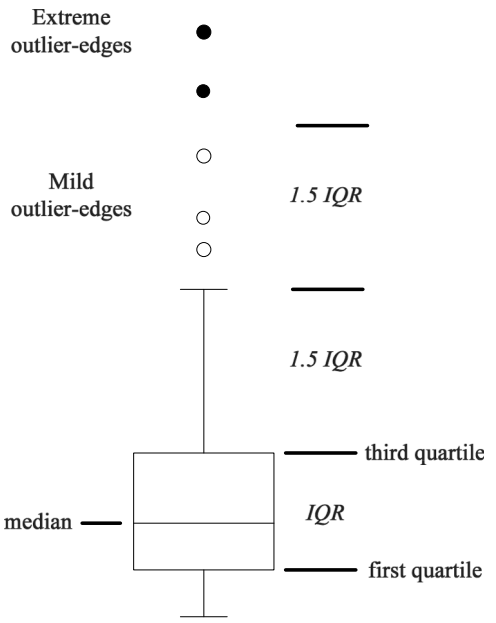

Fig. 3. Illustration of Box-plot with extreme outlier-edges and mild outlier-edges.

and v_5 is the noise. Similarly, we find an extreme outlier-edge edge(v_1, v_4) in C_1 and no extreme outlier-edges in C_2. After removing edge(v_1, v_4), we get a sub-cluster containing only one node v_4, thus, v_4 is also a noise. Thus, by using box-plot method to partition DOT in Fig. 2(b), we finally get two sub-clusters $C_1 = v_1, v_2, v_3, v_5$ and $C_2 = v_6, v_7, v_8, v_9$, and two noises v_4 and v_{10}, the result is shown in Fig. 2(c).

Partitioning the DOT is similar to the process of choosing cluster centers from a decision graph [22] (decision graph is the plot of δ_i as a function of ρ_i for each node). However, our method provides a clear criterion on how to choose cluster centers through the DOT constructing principle and how to get the sub-clusters through partitioning DOT. Take a synthetic dataset DS4 (taken from Ref. [30]) as an example. DS4 contains two datasets shaped like a crescent with different densities. Fig. 4(a) shows noises we discovered in decision graph and Fig. 4(d) shows noises in the 2-D space. Fig. 4(b) shows cluster centers in the decision graph, and Fig. 4(e) shows cluster centers in the 2-D space. It is obvious that cluster centers we find are points of relatively high δ in decision graph. Moreover, the 14 sub-clusters are correctly found though our method (Fig. 4(c)).

3.3. Merge sub-clusters through an iterative heuristic method

The procedure of merging sub-clusters is applied if the number of sub-clusters is more than the target number of clusters (g). In this phase, we merge sub-clusters by minimizing a measure which is called disconnectivity.

The disconnectivity between two sub-clusters C_i and C_j is defined as a penalty for the violation of both cluster k-NN consistency (k-Nearest Neighbor) and cluster k-MN consistency (cluster k-Mutual Nearest-Neighbor consistency) [31,32]. With this measure, we can see that if the neighbor node j of node i does not belong to the same cluster, then a penalty is imposed. This penalty is equal to the inverse of the distance between the two nodes. Furthermore, if the neighbor node i of node j is not grouped in the cluster to which node j belongs to, then the same amount of penalty will also be imposed [31]. Specifically, disconnectivity between sub-cluster C_i and sub-cluster C_j is defined as follows:

$$dis(i,j) = \frac{\sum_{i\in C_i}\sum_{j\in C_j}(b_{ij}^{(1)} + b_{ij}^{(2)})\frac{1}{d(i,j)}}{|C_i|\,|C_j|} \tag{5}$$

where $b_{ij}^{(1)} = \begin{cases}1 \text{ if } j \in Ne_k(i)\\ 0 \text{ otherwise}\end{cases}$, $b_{ij}^{(2)} = \begin{cases}1 \text{ if } i \in Ne_k(j)\\ 0 \text{ otherwise}\end{cases}$, and $|C|$ is the number of nodes in C. For example, in Fig. 2(d) ($k = 2$), for two sub-clusters denoted as $C_1 = v_1, v_2, v_3, v_5$ and $C_2 = v_6, v_7, v_8, v_9$, it is clear that $v_5 \in Ne_2(v_8)$ and $v_8 \in Ne_2(v_5)$, and $|C_1| = 4$, $|C_2| = 4$, thus, we can calculate $dis(C_1, C_2) = 2/(4 \times 4 \times d(v_5, v_8))$.

Moreover, given a partition $P_g = \{C_1, C_2, \ldots, C_m\}$, in which C_i denote the ith sub-cluster, the global disconnectivity of the partition is defined as

$$gdis = \sum_r \sum_{i\in C_r} \sum_{j\notin C_r} (b_{ij}^{(1)} + b_{ij}^{(2)})\frac{1}{d(i,j)}. \tag{6}$$

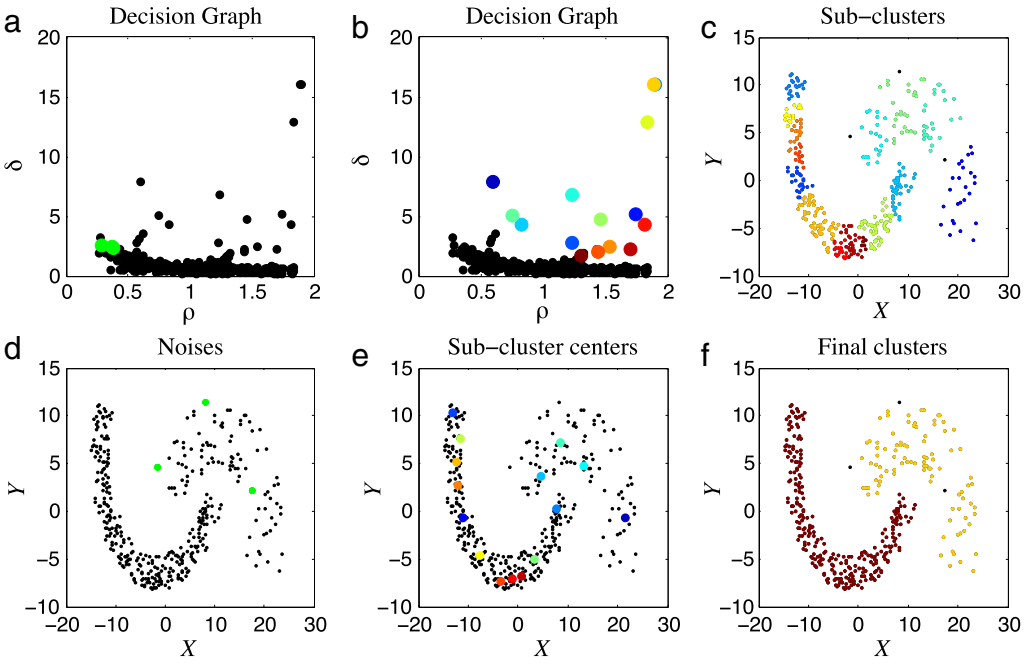

Fig. 4. Illustrations of sub-clusters found for DS4 dataset.

It is intuitive that global disconnectivity should be minimized for optimized clustering solution [31]. Here we provide a heuristic method to the merge problem that applies to a sub-clusters for with the objective is to minimize disconnectivity.

step 1. we select a pair of sub-clusters C_i and C_j such that

$$(i, j) \leftarrow \max_{i,j\in\{1,\ldots,m\}} dis(i, j) \tag{7}$$

s.t.

$$\max_{i,j\in\{1,\ldots,m\}} dis(i, j) > 0 \tag{8}$$

step 2. merge C_i and C_j and decrease the number of sub-clusters, i.e. $m \leftarrow m - 1$. If $m = g$, the target number of clusters has been obtained then stop. Otherwise, repeat step 1 and step 2.

It is noteworthy that the step 2 must guarantee $\max dis(i, j)_{i,j\in\{1,\ldots,m\}} > 0$. If $\max dis(i, j)_{i,j\in\{1,\ldots,m\}} > 0$, i.e. $dis(i, j) = 0,\ \forall i, j \in \{1, \ldots, m\}$, $dis(i, j) = 0$ implies no interactive among sub-clusters from the k-NN and k-MN perspective such that merge steps should be terminated. Therefore, every merge step must satisfy $\max dis(i, j)_{i,j\in\{1,\ldots,m\}} > 0$. Actually, SCDOT depends only on the selected number of nearest neighbors k, as it determines the density of points and disconnectivity. SCDOT uses an iteration strategy to choose k. Specifically, we start with $k = 4$ (similar to the DBSCAN advice), and repeat the algorithm with increasing values of k until the target number clusters is obtained. Therefore, we can automatically determine k based on this strategy. That is, except the number of clusters, there are no parameters left to be tuned by the user. Fig. 4(f) shows the final clusters found in DS4 dataset through this strategy. And details of the algorithm are described in the follow section.

3.4. Algorithm and performance analysis

Given n data points, we propose to solve the SCDOT by using a heuristic iteration algorithm shown below, based on four data structures: local density vector $\boldsymbol{\rho}$, where ρ_i is the local density of node i; nearest node vector ***nneigh***, where $nneigh_i$ is the position of node i's nearest neighbor of higher density; nearest neighbor matrix $\boldsymbol{K}$, an $n * (n - 1)$ matrix in which an entry K_{ij} is node i's j-nearest neighbor; ***dis*** is the disconnectivity matrix of sub-clusters.

Given dataset D, distance between node i and node j, d_{ij}, and the target number of clusters g. Let us initialize the $k = 4$. In order to build matrix $\boldsymbol{K}$, an efficient algorithm for nearest neighbor search such as KD-tree [33] can be used (step 1). The KD-tree has the complexity of $O(n \log(n))$ for building and $O(kn^{1-1/k})$ for searching in the worst case [34]. Therefore, the overall complexity for building $\boldsymbol{K}$ is equivalent to the one for building KD-tree, as the search takes much less time than the tree building when $n \gg k$. Step 2 takes $O(kn)$ to compute ρ according to Eq. (1). Step 3 takes $O(kn^{1-1/k})$ to compute ***nneigh***

by using $\boldsymbol{\rho}$ and KD-tree searching. This step aims to find the point of higher density in the tree that is nearest to a given input point. Based on ***nneigh***, step 4 takes $O(n-1)$ to construct DOT because ***nneigh*** is actually the edges of DOT. In step 5, the box-plot method is used to partition DOT, the complexity of this step is $O(2In)$ in the worst case, and I is the number of iterations. In fact, I is the number of sub-clusters. Finally, ***dis*** is obtained in $O(nkI^2)$ time in step 6, after that, in step 7, we merge the sub-clusters repeatedly and adjust ***dis*** accordingly. Therefore, the overall time complexity is $O(3n\log n + kn + 2In + nkI^2)$, since $I \ll n$, $k \ll n$, the time complexities is about $O(n\log(n))$, which is dominated by the construction of the KD-tree. The proposed algorithm is described as Algorithm 1.

Algorithm 1 SCDOT

Input: number of clusters g, dataset D.
Output: clusters C.

step 1. Initialize k = 4, build KD-tree for the dataset D, and compute $\boldsymbol{K}$ through nearest neighbor search.
step 2. Compute the $\boldsymbol{\rho}$ according to Eq. (1).
step 3. Compute ***nneigh*** according to Eq. (4).
step 4. Compute DOT based on the ***nneigh***. In fact, ***nneigh*** can be considered as the weight of edges of the DOT.
step 5. Use the box-plot method to partition DOT into sub-clusters $C = C_1, C_2, \cdots, C_m$.
step 6. Calculate ***dis*** according to Eq. (5).
step 7. Merge C according to Eq. (7), and get the final clusters $C = C_1, C_2, \cdots, C_c$, where c is the number of clusters.
step 8. If $c < g$, goto step 2 and set $k = k + 1$.

4. Results and discussion

In this section, we present three experimental results to demonstrate the effectiveness of the proposed algorithm. The synthetic dataset DS1, DS2, DS3 are taken from the literature [35–37], DS1 and DS2 are of diverse shape, and DS3 is 3-spiral dataset; The synthetic dataset DS4, DS5 are taken from Refs. [30,38], DS6 is generated by DataGenerator tool [39], they are composed of clusters with diverse shapes and densities. Finally, a real dataset containing all organized violence in African continent are taken from the Geo-referenced Event Dataset of the Uppsala Conflict Data Program (UCDP) [40].

In this paper, we focus on finding clusters with diverse shapes and densities in spatial data. Since partition clustering method and distribution-based method are not able to solve this problem, we will not discuss them in the experiments. DBSCAN method and Chameleon are outstanding density-based clustering method and hierarchical clustering method respectively, they are able to find clusters with diverse shapes or/and densities. SNN claims that it can deal with various shape and density in cluster discovering, but it highly depends on the parameter setting, which is difficult to set as a priori. Therefore, we compare SCDOT's performance against that of DBSCAN and Chameleon on three different datasets and we utilize the Adjust Rand Index (ARI) [41,42] and visual inspection to evaluate the accuracy of clustering methods.

With the goal of comparing SCDOT with DBSCAN, we first select $MinPts = 4$ (default value in Ref. [9]) for DBSCAN and then tune *Eps* manually to achieve the best result in terms of visual inspection. Chameleon is implemented through CLUTO package [43] (*-clmethod = graph*, *-sim = dist*, *-agglofrom = 30*).

Experiment 1. We benchmark SCDOT against DBSCAN and Chameleon on DS1, DS2, and DS3.

We demonstrate all results in Fig. 5. For these datasets, SCDOT and DBSCAN can both deal with varied shape of clusters, but DBSCAN clusters have more noise points than SCDOT, and it requires tuning *Minpts* and *eps* carefully. Chameleon performs well on DS2 dataset but it is the worst on DS1 dataset and DS3 dataset in terms of ARI values, it is likely that chameleon tends to merge clusters which are too close to each other. In addition, SCDOT can identify the sub-clusters, cluster centers and noises (Fig. 6).

Experiment 2. These datasets are composed of clusters with diverse shape and density, DS6 dataset is generated by DataGenerator tool, and it contains significant noises which are not suitable for ARI evaluation. As we can see from Fig. 7 and Table 2, in the case of DS4 and DS5, SCDOT and Chameleon outperform DBSCAN, because DBSCAN clusters too many noise points in DS5 and identifies three clusters in DS4.

In the case of DS6, it is clear that SCDOT find expected clusters and outperform the DBSCAN and Chameleon. DBSCAN discovers four prominent clusters, but it also finds some small clusters with too many noise points (Fig. 7(h)). Fig. 7(i) shows that Chameleon fails to detect clusters in the central region. Of course, in this case, DBSCAN and chameleon may achieve proper results by tuning the parameters, but if the dataset is not visible, it is impossible evaluate clustering results and is difficult to determine input parameters, this weakness is discussed in detail in section of related work. In addition, SCDOT can identify the sub-clusters, cluster centers and noises in DS4, DS5 and DS6 (Fig. 8). It can be seen from Table 1, Table 2, Fig. 5, and Fig. 7 that SCDOT can effectively identify clusters with various densities and shapes

Experiment 3. We further investigate the performance of SCDOT on a real dataset. For this purpose we use the Geo-referenced Event Dataset [40], which is a 2-dimensional dataset containing all organized violence in African continent. For the purpose of this study we use 2008 year dataset where 848 organized violence are included (the duplicate points are regard as one point). Clustering results by the three methods are shown in Fig. 9. In this experiment, we regard the result

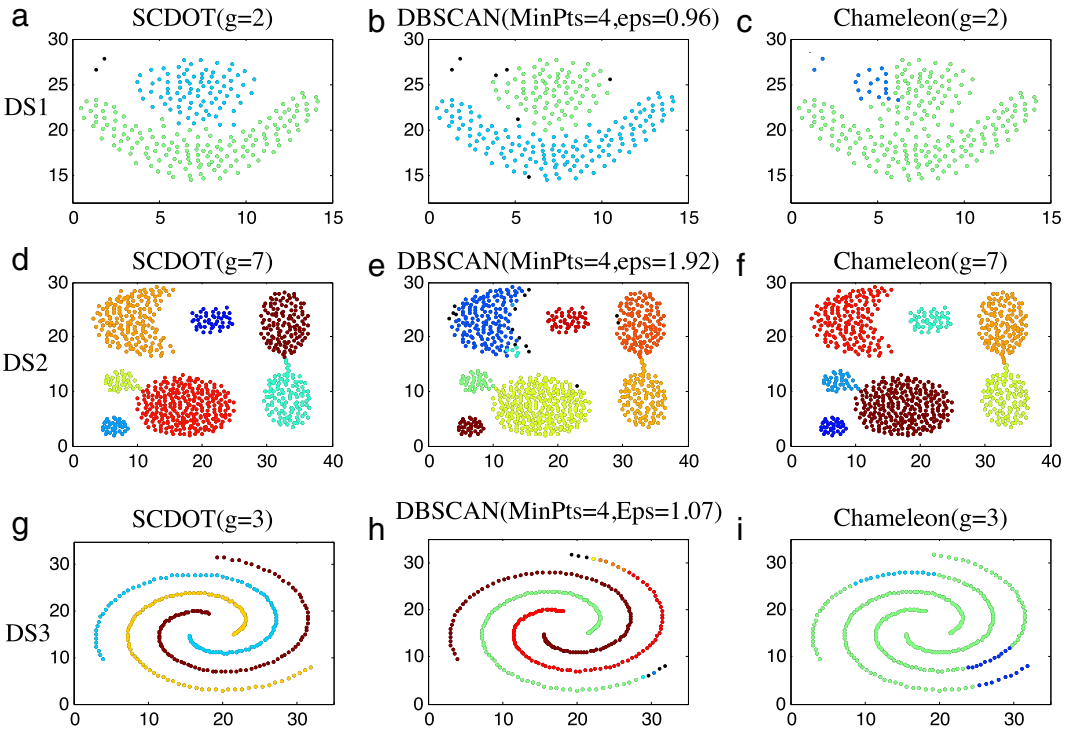

Fig. 5. Clusters discovered by SCDOT, DBSCAN and Chameleon for DS1, DS2, DS3. g is the number of desired cluster and black points are noises. Clusters are represented by different colors. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

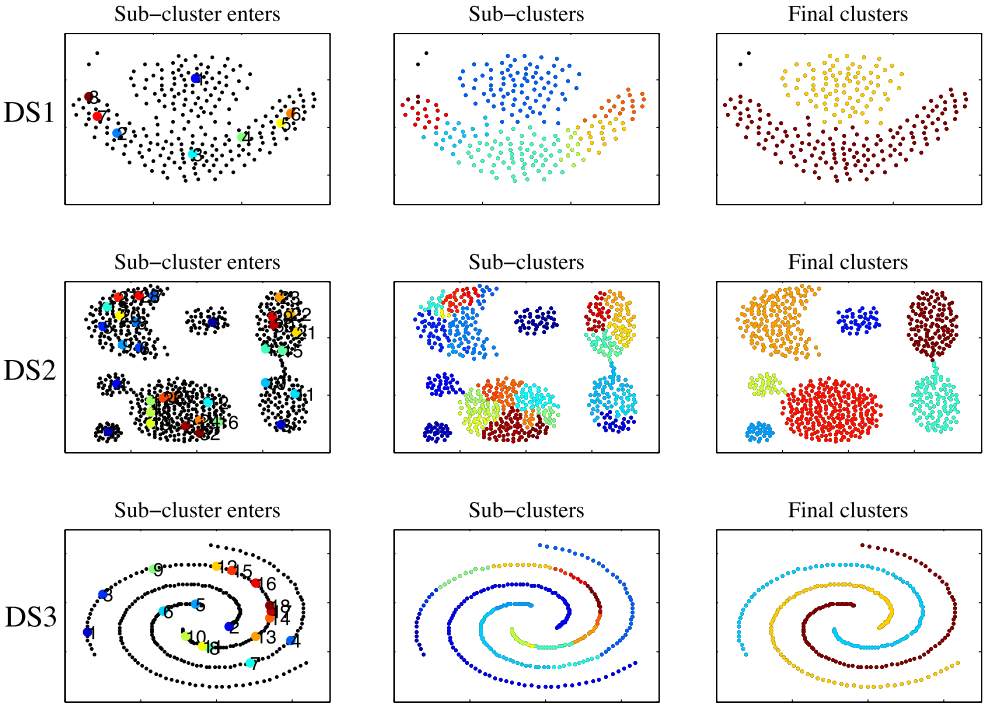

Fig. 6. For DS1, DS2 and DS3, SCDOT finds some sub-cluster centers and sub-clusters by partitioning the DOT, and it finds the final clusters by merging the sub-clusters.

which has 9 or 10 clusters as the accepted results though visual inspection. We can see that SCDOT outperforms DBSCAN (by tuning *Eps* to achieve the best result in terms of visual inspection) and Chameleon. Specially, for each of the results the

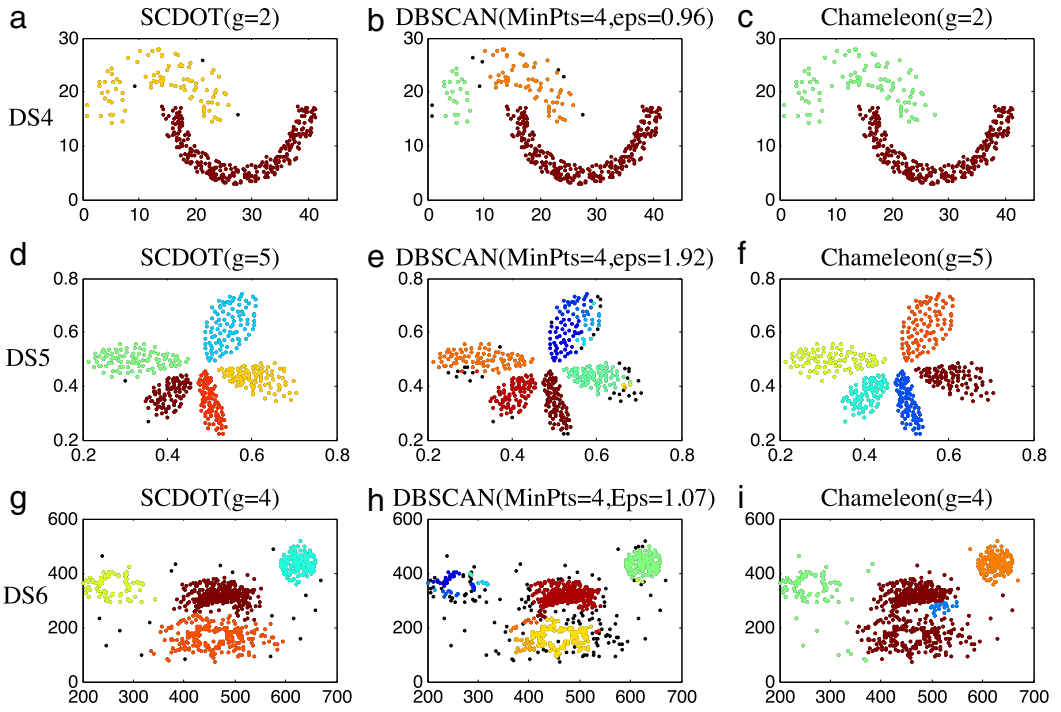

Fig. 7. Clusters discovered by SCDOT, DBSCAN and Chameleon for DS4, DS5, and DS6 datasets. g is the number of desired cluster and black points are noises, clusters are represented by different colors. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

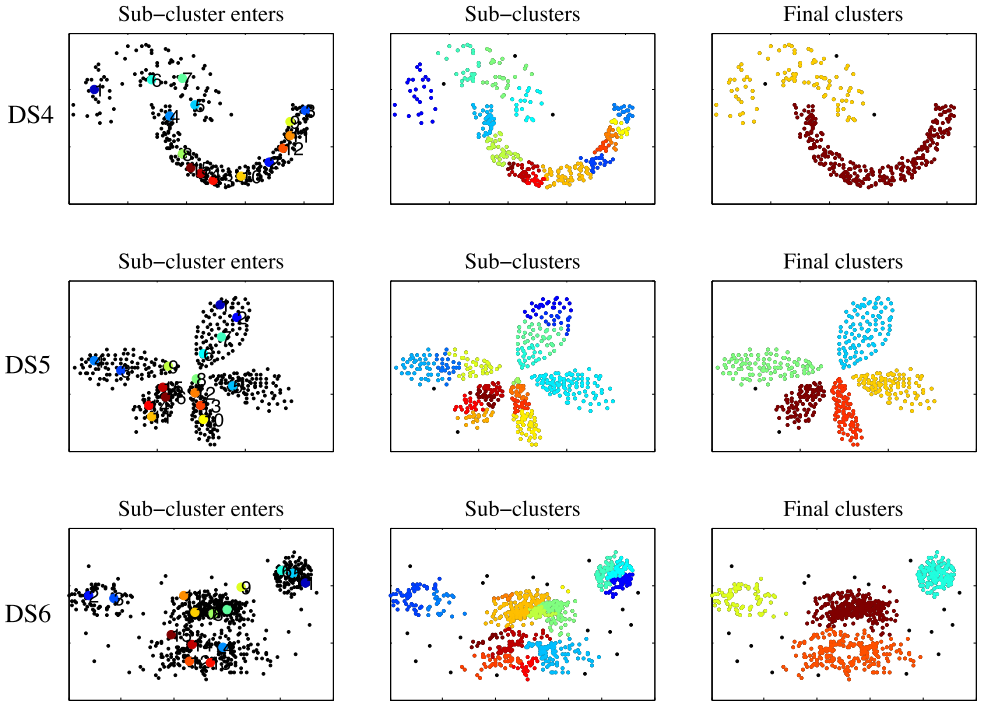

Fig. 8. For DS4, DS5 and DS6, SCDOT finds some sub-cluster centers and sub-clusters by partitioning the DOT, and it finds the final clusters by merging the sub-clusters.

clusters obtained by SCDOT appear reasonable, while DBSCAN regard lower density of cluster as the noise when the *Eps* tuned lower, as shown in Fig. 9(b). In order to discover clusters with lower density, it regards some distinct clusters with

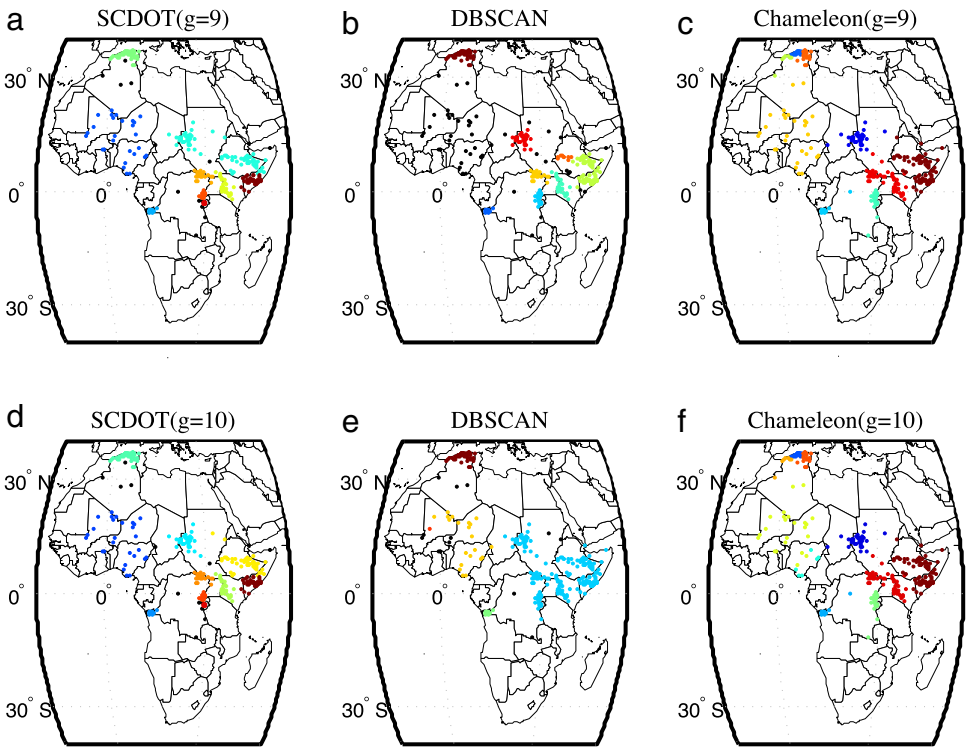

Fig. 9. Clusters discovered by SCDOT, DBSCAN and Chameleon. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

Table 1
Results in Adjust Rand Index for DS1, DS2 and DS3 dataset.

	SCDOT		DBSCAN		Chameleon	
	ARI	Noise number	ARI	Noise number	ARI	Noise number
DS1	0.988	2	0.901	7	0.132	0
DS2	0.988	0	0.958	15	0.992	0
DS3	1	0	0.935	6	0.007	0

Table 2
Results Adjust Rand Index for DS4, and DS5 datasets.

	SCDOT		DBSCAN		Chameleon	
	ARI	Noise number	ARI	Noise number	ARI	Noise number
DS4	0.991	3	0.933	8	1	1
DS5	0.995	2	0.805	15	1	0

higher density as one cluster (Fig. 9(e)). Chameleon identifies almost proper clusters, but it divides one distinct cluster (in Northern Algeria) into three small clusters, as shown in Fig. 9(c) and (f). Moreover, from Fig. 9(a) and (d), clusters found by SCDOT correspond to actual geographical distribution of organized violence (shown in Fig. 9(d)). For example, in Fig. 9(d), green cluster corresponds to northern Algeria, the organized violence in this area is the state-based armed conflicts between government of Algeria and AQIM (salafist group for preaching and combat); Yellow cluster corresponds to Ethiopia, mainly including the state-based armed conflicts between government of Ethiopia and ONLF (Ogaden National Liberation Front), OLF (Oromo Liberation Front); Brown cluster corresponds to southern Somalia, mainly including the conflicts between government of Somalia and Al-Shabaab; Light green cluster corresponds to western Kenya, mainly including the one-side violence of civilians and non-state conflicts between Kalenjin and Kikuyu; Cyan cluster corresponds to western Sudan, the organized violence in this area are some one-side violence and conflicts between government of Sudan and SLM/A (South Sudan Liberation Movement/ Army), JEM (Justice and Equality Movement); Red cluster corresponds to The Republic of Burundi, the organized violence in this area mainly include the state-based armed conflicts between Government of Burundi and Palipehutu-FNL (Party for the Liberation of the Hutu People-Forces for National Liberation—Lovers of Peace faction); Blue cluster with lower density correspond to the area around north of Gulf of Guinea including Mali, Niger and

Nigeria, etc., the organized violence in this area mainly includes conflicts between government of Mali and ATNMC (May 23 Democratic Alliance for Change—Ibrahim Bahanga faction), Government of Niger and MNJ(Niger Movement for Justice), and some one-side violence, and so on.

5. Conclusions

In this paper, a novel spatial clustering method, called SCDOT, is presented to detect clusters with diverse shapes and densities. Except the number of clusters, there are no parameters left to be tuned by the user. SCDOT is the first method that combines the density-based clustering, partition and hierarchical clustering principle. Inspired by the work of Rodriguez and Laio, we construct a DOT to model the data. Unlike the minimal spanning tree and k-nearest-neighbor graph, DOT efficiently integrates information on not only distance but also density between data points, which is helpful for identifying clusters with high density or low density. A partition-and-merge strategy is applied to DOT to identify genuine clusters, meanwhile, an iterative heuristic approach is utilized to select parameters automatically. Extensive experiments on both synthetic and real-world datasets demonstrate that SCDOT does effectively identify clusters of different shapes and densities. We provide a new way to identify noise and cluster centers, i.e., cluster centers are recognized as roots of sub-trees and noises are recognized as anomalous leaves. This method can identify meaningful local noises that previous approaches cannot find, for example, the anomalous leaves in DOT are global noises while the anomalous leaves in sub-trees are regarded as the local noises. An important direction of future research is not only to reduce the influence of noise but also to clearly identify them. In addition, our approach only requires measuring the distance between all the pairs of data points and does not require parameterizing a probability distribution. Therefore, its performance is not affected by the intrinsic dimensionality of the space in which the data points are embedded, SCDOT is flexible and extendible to different applications.

Acknowledgments

The authors would like to thank Alessandro Laio for guidance on the manuscript. This work was supported by the National Natural Science Foundation of China (no. 71471175, 71301165, 71522014 and 61201328) and Hunan Provincial Innovation Foundation for Postgraduate under Grant No. CX2013B024.

References

[1] J. Han, M. Kamber, A.K.H. Tung, Spatial clustering methods in data mining: A survey, in: H.J. Miller, J. Han (Eds.), Geographic Data Mining and Knowledge Discovery, in: Research Monographs in GIS, Taylor and Francis, 2001, pp. 1–29.
[2] F. Harvey, The handbook of geographic informatin science, Int. J. Geogr. Inf. Sci. 23 (5) (2009) 683–684.
[3] L. Wang, X. Li, Spatial epidemiology of networked metapopulation: an overview, Chinese Sci. Bull. 59 (28) (2014) 3511–3522. http://dx.doi.org/10.1007/s11434-014-0499-8.
[4] L. Wang, Z. Wang, Y. Zhang, X. Li, How human location-specific contact patterns impact spatial transmission between populations? Sci. Rep. 3 (2013) 1468.
[5] R.T. Ng, J. Han, Efficient and effective clustering methods for spatial data mining, in: Proceedings of the 20th International Conference on Very Large Data Bases, VLDB'94, Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1994, pp. 144–155.
[6] Tony H. Grubesica, Ran Weib, Alan T. Murraya, Spatial clustering overview and comparison: Accuracy, sensitivity, and computational expense, Ann. Assoc. Amer. Geogr. 104 (6) (2014) 1134–1115.
[7] J.B. MacQueen, Some methods for classification and analysis of multivariate observations, in: L.M.L. Cam, J. Neyman (Eds.), Proc. of the fifth Berkeley Symposium on Mathematical Statistics and Probability, Vol. 1, University of California Press, 1967, pp. 281–297.
[8] L. Kaufman, P. Rousseeuw, Clustering by means of medoids, in: Statistical Data Analysis Based on the L1-Norm and Related Methods, North–Holland, 1987.
[9] M. Ester, H.-P. Kriegel, J. Sander, X. Xu, A density-based algorithm for discovering clusters in large spatial databases with noise, in: Proc. of 2nd International Conference on Knowledge Discovery, 1996, pp. 226–231.
[10] G. Karypis, E.-H.S. Han, V. Kumar, Chameleon: Hierarchical clustering using dynamic modeling, Computer 32 (8) (1999) 68–75. http://dx.doi.org/10.1109/2.781637.
[11] C. Zhong, D. Miao, P. Fränti, Minimum spanning tree based split-and-merge: A hierarchical clustering method, Inform. Sci. 181 (16) (2011) 3397–3410. http://dx.doi.org/10.1016/j.ins.2011.04.013.
[12] L. Ertoz, M. Steinbach, V. Kumar, A new shared nearest neighbor clustering algorithm and its applications, in: Workshop on Clustering High Dimensional Data and its Applications at 2nd SIAM International Conference on Data Mining, 2002.
[13] P.S. Bradley, O.L. Mangasarian, W.N. Street, Clustering via concave minimization, in: M. Mozer, M.I. Jordan, T. Petsche (Eds.), NIPS, MIT Press, 1996, pp. 368–374.
[14] D. Arthur, S. Vassilvitskii, K-means++: The advantages of careful seeding, in: Proceedings of the Eighteenth Annual ACM–SIAM Symposium on Discrete Algorithms, SODA'07, Society for Industrial and Applied Mathematics, Philadelphia, PA, USA, 2007, pp. 1027–1035.
[15] R. Nock, F. Nielsen, On weighting clustering, IEEE Trans. Pattern Anal. Mach. Intell. 28 (8) (2006) 1223–1235.
[16] B.W. Silverman, Density Estimation for Statistics and Data Analysis, Chapman & Hall, London, 1986.
[17] M. Ankerst, M.M. Breunig, H.-P. Kriegel, J. Sander, Optics: Ordering points to identify the clustering structure, in: Proceedings of the 1999 ACM SIGMOD International Conference on Management of Data, SIGMOD'99, ACM, New York, NY, USA, 1999, pp. 49–60. http://dx.doi.org/10.1145/304182.304187.
[18] P. Viswanath, R. Pinkesh, l-dbscan: A fast hybrid density based clustering method, in: ICPR (1), IEEE Computer Society, 2006, pp. 912–915.
[19] P. Viswanath, V. Suresh Babu, Rough-dbscan: A fast hybrid density based clustering method for large data sets, Pattern Recognit. Lett. 30 (16) (2009) 1477–1488. http://dx.doi.org/10.1016/j.patrec.2009.08.008.
[20] D. Birant, A. Kut, St-dbscan: An algorithm for clustering spatial–temporal data, Data Knowl. Eng. 60 (1) (2007) 208–221. http://dx.doi.org/10.1016/j.datak.2006.01.013.
[21] A. Hinneburg, E. Hinneburg, D.A. Keim, An efficient approach to clustering in large multimedia databases with noise, in: Proc of 4th International Conference in Knowledge Discovery and Data Mining, KDD 98, 1998, pp. 58–65.
[22] A. Rodriguez, A. Laio, Clustering by fast search and find of density peaks, Science 344 (6191) (2014) 1492–1496. http://dx.doi.org/10.1126/science.1242072.

[23] Q. Cheng, Z. Liu, J. Huang, G. Cheng, Community detection in hypernetwork via density-ordered tree partition, Appl. Math. Comput. 276 (5) (2016) 384–393. http://dx.doi.org/10.1016/j.amc.2015.12.039.
[24] T. Zhang, R. Ramakrishnan, M. Livny, Birch: An efficient data clustering method for very large databases, in: Proceedings of the 1996 ACM SIGMOD International Conference on Management of Data, SIGMOD'96, ACM, New York, NY, USA, 1996, pp. 103–114. http://dx.doi.org/10.1145/233269.233324.
[25] S. Guha, R. Rastogi, K. Shim, Cure: An efficient clustering algorithm for large databases, in: Proceedings of the 1998 ACM SIGMOD International Conference on Management of Data, SIGMOD'98, ACM, New York, NY, USA, 1998, pp. 73–84. http://dx.doi.org/10.1145/276304.276312.
[26] D. Cheng, R. Kannan, S. Vempala, G. Wang, A divide-and-merge methodology for clustering, ACM Trans. Database Syst. 31 (4) (2006) 1499–1525. http://dx.doi.org/10.1145/1189769.1189779.
[27] H. Cui, G. Ruan, J. Xue, R. Xie, L. Wang, X. Feng, A collaborative divide-and-conquer k-means clustering algorithm for processing large data, in: Proceedings of the 11th ACM Conference on Computing Frontiers, CF'14, ACM, New York, NY, USA, 2014, pp. 20:1–20:10. http://dx.doi.org/10.1145/2597917.2597918.
[28] S. Guha, R. Rastogi, K. Shim, Rock: A robust clustering algorithm for categorical attributes, Inf. Syst. 25 (5) (2000) 345–366.
[29] R. McGill, J.W. Tukey, W.A. Larsen, Variations of box plots, Amer. Statist. 32 (1) (1978) 12–16. http://dx.doi.org/10.2307/2683468.
[30] A.K. Jain, M.H.C. Law, Data clustering: A user's dilemma, in: S.K. Pal, S. Bandyopadhyay, S. Biswas (Eds.), PReMI, in: Lecture Notes in Computer Science, vol. 3776, Springer, 2005, pp. 1–10. URL: http://dblp.uni-trier.de/db/conf/premi/premi2005.html#JainL05.
[31] J.-S. Lee, S. Olafsson, Data clustering by minimizing disconnectivity, Inform. Sci. 181 (4) (2011) 732–746. http://dx.doi.org/10.1016/j.ins.2010.10.028.
[32] J.-S. Lee, S. Olafsson, A meta-learning approach for determining the number of clusters with consideration of nearest neighbors, Inform. Sci. 232 (2013) 208–224. http://dx.doi.org/10.1016/j.ins.2012.12.033.
[33] J.L. Bentley, Multidimensional binary search trees used for associative searching, Commun. ACM 18 (9) (1975) 509–517. http://dx.doi.org/10.1145/361002.361007.
[34] D.T. Lee, C.K. Won, Worst-case analysis for region and partial region searches in multidimensional binary search trees and balanced quad trees, Acta Inf. 9 (1977) 23–29.
[35] L. Fu, E. Medico, Flame, a novel fuzzy clustering method for the analysis of dna microarray data, BMC Bioinformatics 8 (1) (2007) 3. http://dx.doi.org/10.1186/1471-2105-8-3.
[36] A. Gionis, H. Mannila, P. Tsaparas, Clustering aggregation, ACM Trans. Knowl. Discov. Data 1 (1) (2007) http://dx.doi.org/10.1145/1217299.1217303.
[37] H. Chang, D.-Y. Yeung, Robust path-based spectral clustering, Pattern Recognit. 41 (1) (2008) 191–203.
[38] X. Chen, W. Liu, H. Qiu, J. Lai, Apscan: A parameter free algorithm for clustering, Pattern Recognit. Lett. 32 (7) (2011) 973–986. http://dx.doi.org/10.1016/j.patrec.2011.02.001.
[39] Y. Pei, O. Zaiane, A synthetic data generator for clustering and outlier analysis, Tech. rep, Department of Computing Science, University of Alberta, 2006.
[40] R. Sundberg, E. Melander, Introducing the UCDP georeferenced event dataset, J. Peace Res. 50 (4) (2013) 523–532.
[41] W.M. Rand, Objective criteria for the evaluation of clustering methods, J. Amer. Statist. Assoc. 66 (336) (1971) 846–850.
[42] L. Hubert, P. Arabie, Comparing partitions, J. Classification 2 (1) (1985) 193–218. http://dx.doi.org/10.1007/BF01908075.
[43] G. Karypis, CLUTO: A Clustering Toolkit, 2002.