IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received February 24, 2019, accepted March 9, 2019, date of publication March 14, 2019, date of current version April 13, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2904995

A Novel Hybrid Clustering Algorithm Based on Minimum Spanning Tree of Natural Core Points

JINLONG HUANG[1], RU XU[1], DONGDONG CHENG[1], SULAN ZHANG[1], AND KEKE SHANG[2]
[1]College of Big Data and Intelligent Engineering, Yangtze Normal University, Chongqing 408100, China
[2]Computational Communication Collaboratory, School of Journalism and Communication, Nanjing University, Nanjing 210093, China

Corresponding author: Sulan Zhang (slzhang@cqu.edu.cn)

This work was supported in part by the Education Committee Foundation of Chongqing under Grant KJ1601210, and in part by the National Nature Science Foundation of China under Grant 61601060 and Grant 61803047.

ABSTRACT Clustering analysis has been widely used in pattern recognition, image processing, machine learning, and so on. It is a great challenge for most existing clustering algorithms to discover clusters with complex manifolds or great density variation. Most of the existing clustering needs manually set neighborhood parameter K to search the neighbor of each object. In this paper, we use natural neighbor to adaptively get the value of K and natural density of each object. Then, we define two novel concepts, natural core point and the distance between clusters to solve the complex manifold problem. On the basis of above-proposed concept, we propose a novel hybrid clustering algorithm that only needs one parameter M (the number of final clusters) based on minimum spanning tree of natural core points, called NCP. The experimental results on the synthetic dataset and real dataset show that the proposed algorithm is competitive with the state-of-the-art methods when discovering with the complex manifold or great density variation.

INDEX TERMS Clustering, complex manifold, natural neighbor, natural core points, minimum spanning tree.

I. INTRODUCTION

Clustering is one of primary research topic in data mining. Clustering has been widely used in many areas, such as pattern recognition, machine learning, face recognition and community detection. It aims at grouping N data points into M clusters so that data points in same cluster as similar as possible, while data points in different cluster as distinct as possible. Relative to classification, clustering is a unsupervised learning method, since the class label is unknown before clustering. Up to present, a number of clustering algorithms have been proposed. These algorithms can be roughly divided into distribution-based clustering [1]–[3], partitional clustering [4]–[6], density-based clustering [7], [8], hierarchical clustering [9]–[11] and so on.

Distribution-based algorithm assumes that the objects in a specified cluster are most likely to be derived from a unique distribution. Such as Expectation Maximization (EM)-based method [12]. Moreover, it is usually difficult for researcher to know the model or describe the distribution of real datasets before clustering. Given a dataset $X = \{x_1, x_2, \ldots, x_n\}$, the basic idea of a partitinal clustering method is to partition the dataset into K clusters($K < n$). This kind of clustering algorithm generally starts with an initial partition of X and then uses an iterative control strategy to optimize an objective function until get the optimal solution or meet the termination condition. K-means [4] and K-medoids [5] are the primary representatives of partitional clusteirng method. However, These two types of clustering algorithms are not applicable to non-spherical or nonconvex datasets [13].

The performance of Density-based clustering and hierarchical clustering algorithm is well on non-spherical datasets and nonconvex datasets. The key idea of density-based clustering is that the clusters are defined as areas with higher density. Density-based clustering algorithm also has a certain capacity to cluster the datasets with manifold. DBSCAN [14] is a primary representative of Density-based clustering algorithm. However, density-based clustering algorithms cannot successfully cluster datasets with great density variations or complex manifold.

Generally, a hierarchical clustering algorithm partitions a dataset into various clusters by an agglomerative or a divisive approach based on a dendrogram. Agglomerative clustering and divisive clustering are two main clustering strategies of hierarchical clustering. At first, agglomerative clustering considering each point as a cluster. Then agglomerative

The associate editor coordinating the review of this manuscript and approving it for publication was Chao Tong.

clustering iteratively combines two most similar clusters in terms of an objective function until gains the final clustering results. On the contrary, divisive clustering starts with only one cluster that including all data points of datasets. Then divisive clustering iteratively according to some strategy to selects a cluster and partitions it into two subclusters. The advantage of hierarchical clustering is that the dendrogram of clusters is more useful than the final clustering results for user. However, although hierarchical clustering is applicable to datasets with manifold, hierarchical clustering has a relatively high computational cost since it constructs the dendrogram on all points of datasets. Such as Single linkage [15] and complete linkage [16] that two well-known examples of hierarchical clustering algorithms take $O(N^2\, log^N)$ time.

Besides the above clustering algorithms, many hybrid clustering algorithms that combine the advantages of hierarchical and partitional clustering have been proposed in the literature [17]–[22]. The hybrid clustering algorithms mainly consist of two stages. In the first stage, hybrid clustering algorithms divide the datasets into many subsets with a partitioning criterion. In the second stage, hybrid clustering algorithms continually merge two subsets into one cluster in terms of a similarity measure until meet the termination condition. CHAMELEON [18] is the representation of hybrid clustering algorithm. Zhong and Miao [23] proposed a hybrid clustering method in which a minimum spanning tree (MST) and an MST-based graph are employed to guide the splitting and merging process.

Minimum spanning tree (MST) is a useful graph structure, which has been employed to capture perceptual grouping [24]. A number of MST-based clustering algorithms have been proposed. Xu *et al.* [25] proposed three MST-based algorithms: removing long MST-edges, a center-based iterative algorithm and a representative-based golbal optimal algorithm. Paper [26] propose MST-LOF algorithm. MST-LOF employs LOF [27] to discard noise points whose density factors are larger than a threshold during the construction of MST. Many other MST-based clustering algorithms that maximize or minimize the degrees of link of the vertices are proposed in paper [28], [29].

However, most of existing clustering algorithms face with two problem: (1) It is difficult to cluster the datasets with complex manifold; (2) In order to solve the first problem, clustering algorithms need so many parameters that manually set. For instance, K-means, K-medoids, AP [30], DP [31] etc. do not apply to cluster the datasets with complex manifold. Although DBSCAN, MST-LOF, DAAP is applicable to the datasets with manifold, these algorithms need many parameters that manually set. For example, MST-LOF need manually set three parameters: the number of neighbors of each object (K); the number of final cluster (M); the outlier rates (α). Even DBSCAN also need manually set two parameters: scan radius (eps) and the minimum points ($minpoints$). It is well known that if one of the parameters is not set appropriately, most of clustering algorithm can't effectively cluster the datasets. In other words, the value of each parameter may directly influence the clustering results. However, in fact, determination of each parameter is dependent on the knowledge of researches experience and a lot of experiment.

In order to solve the problem that mentioned above, in this paper, we combine the core idea of hybrid clustering algorithm with minimum spanning tree, and proposed a novel clustering algorithm, called NCP. Fist, NCP use the concept of Natural Neighbor to adaptively obtain the value of neighborhood parameter K, and find all Neural Core Points of datasets. Second, the proposed algorithm split the datasets into a number of subsets with a partitioning criterion that spreading from natural core point to sparse area. Therefore, one natural core point represents one initial cluster. Thirdly, the proposed algorithm cluster-distance (detailed explaining in section 3) of all adjacent initial clusters, and constructs the minimum spanning tree of natural core points according to cluster-distance. Finally, the proposed algorithm split the dataset into M clusters according to a cut scheme that iteratively cut the maximum edge of the minimum spanning tree of natural core points until gains M clusters. The proposed algorithm is applicable to datasets with complex manifold and great density variations, and only need one parameter M that the number of final cluster.

The remainder of this paper is organized as follows. Section 2 presents a brief overview of the related work, and describes the concept of Natural Neighbor. The proposed algorithm and its correlative definitions are presented in Section 3. The experimental results are presented in Section 4. Finally, the conclusion is provided in Section 5.

II. RELATED WORK

A. MST-BASED AND CENTER-BASED CLUSTERING ALGORITHM

For clustering algorithms, minimum spanning tree is a useful graph structure, since MST can effectively reflect the internal structure of datasets. And a number of MST-based clustering algorithms have been proposed. The core idea of MST-based clustering algorithms is that continually cut the maximum edge of minimum spanning tree of all points in datasets until gain the final M clusters. Therefore, MST-based clustering algorithm is applicable to arbitrary shape datasets. However, the result of MST-based clustering algorithm is easy to be affected by noise points. In other words, the existence of noise points may reduce the clustering effect of MST-based clustering algorithm, such as algorithms that proposed by paper [25], [28], [29], [32]. In order to avoid the influence of noise points, paper [26] propose a new MST-based algorithm (MST-LOF). Unlike the traditional MST-based clustering algorithms that directly construct MST on original datasets, MST-LOF firstly employs LOF [27] to remove the noise points of original datasets before constructing MST.

Fig.1(a) shows the minimum spanning tree of dataset without noise points. $p \in C1, q \in C2$. Form Fig.1(a) we can see that e(p,q) is the only edge, which link up C1 with C2. And e(p,q) is the maximum edge of minimum spanning tree. Therefore, we can easily cluster the dataset via

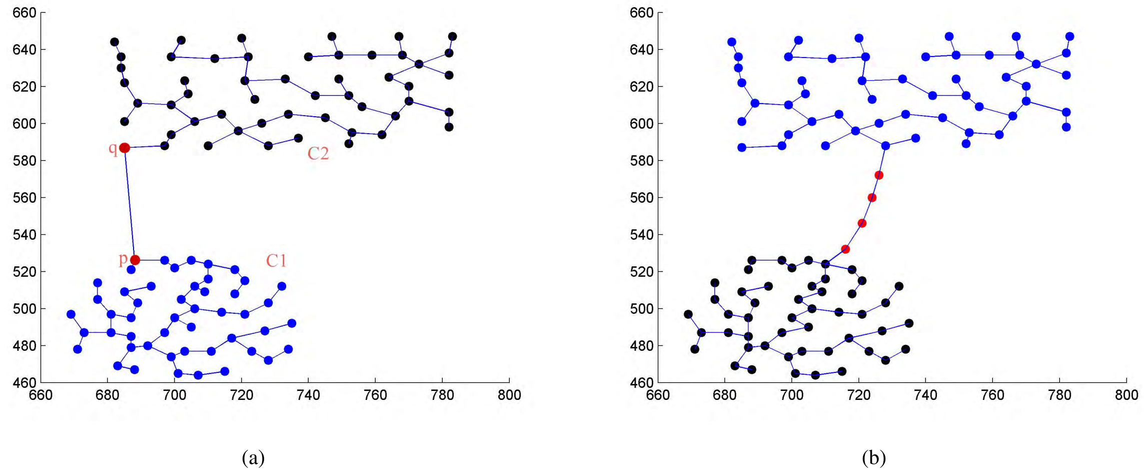


FIGURE 1. The effect of noise on minimum spanning tree.

cutting edge e(p,q). Fig.1(b) shown the minimum spanning tree of dataset with noise points (marked in red). From Fig.1(b) we can see that C1 is connected with C2 via noise points. Moreover, these noise points shorten the edge that link up C1 with C2. Therefore, the noise points may lead to edges that link up C1 with C2 are not the maximum edge of minimum spanning tree. As a result, traditional MST-based algorithm may not get the desirable clustering result via cutting maximum edge of MST. The idea of MST-LOF is that lengthen the edge that link up C1 and C2 via removing the noise of datasets. However, in order to gain the desirable clustering result, MST-LOF may regard some normal points as noise. Same as the traditional clustering algorithm, although MST-LOF remove the noise pints, MST-LOF still has a relatively high computational cost, since MST-LOF construct MST on all remainder points of datasets.

For many clustering algorithm, the key stage is to find the cluster centers, called center-based clustering algorithm, such as K-means and K-medoids. However, most of center-based clustering algorithm do not apply to cluster Non-spherical datasets, such as AP [30] and K-AP [33]. Although DP is applicable to non-spherical, DP do not apply to cluster datasets with complex manifold. In order to solve this problem, Jia *et al.* [34] proposed a clustering algorithm that can solve the complex manifold problem by computing the graph-based distance which is defined as the sum of the Edge-Weight of shortest path, called DAAP. However, the time complexity of DAAP is much higher than AP, K-AP and DP, and the number of parameter that needed by DAAP, AP and K-AP is more than DP. DP only needs one parameter that cutoff distance. DAAP needs too many parameters such as density factor ρ, maximal iteration *maxits*, convergence of iteration coefficient *convits* and the number of neighbors and final clusters.

B. NATURAL NEIGHBOR

For many of traditional clustering algorithms, the basic step is to find neighbors of each object. And the most widely used concept of neighbors is K-nearest neighbors and ε-nearest neighbors that proposed by Stevens [35]. The basic idea of K-nearest neighbors is that finding K nearest or most similar objects for each object. K is a parameter manually set. The smaller the K-distance that the distance between one object and its K-th neighbor is, the bigger the value of this object's density is. ε-nearest neighbors based on the idea: finding all neighbors that distance smaller than scan radius ε for each object. ε is a parameter manually set.

However, although the concept of K-nearest neighbors and ε-nearest neighbors have been widely used in clustering, there is an obvious problem that these two concept of neighbors need parameter K or parameter ε. In this paper, in order to solve the problem that need manually set parameter K or ε, we introduce a new neighborhood concept, Natural Neighbor, into clustering algorithm.

Compared with K-nearest neighbors and ε-nearest neighbors, Natural Neighbor [36] is a new neighbor concept. The great advantage of Natural Neighbor is that the searching procedure of Natural Neighbor do not need any parameter. And the concept of Natural Neighbor has been used in some area of data mining, such as clustering analysis [37], [38], outlier detection [39] and prototype reduction [40]. Unlike K-nearest neighbors and ε-nearest neighbors, the key idea of Natural Neighbor is that not all objects have same number of neighbors. Natural Neighbor suppose that the core objects should possess a more number of neighbors than other ordinary objects. The Natural Neighbor searching algorithm as the Algorithm 1.

Rnb(x) is the times that point x is contained by the neighborhood of other points, which the number of x's reverse neighbor. $NN_r(x)$ is the r-neighborhood of x. $RNN_r(x)$ is the r-reverse-neighborhood of x. sup_k is the average value of the number of each point's neighbors, called Natural Eigenvalue. The time complexity of KNN and RNN for each object in the database is $O(N^2)$. Since KD-tree is introduced into NaN-Searching, the time complexity of NaN-searching algorithm is O(N*lgN). N is the number of object in D.

Algorithm 1 RoughlyCluster(D,k) //D Is the Unclassified Dataset

```
• Output: Natural Eigenvalue $sup_k$
1) Initiallizing: $r = 0, Rnb(i) = 0; NN_r(i) = \emptyset, RNN_r(i) = \emptyset, NaN_r(i) = \emptyset$;
2) Kdtree=creatKDTree(D); //create a KD-tree
3) While
    a) Use kdtree to find the r-th neighbor y for each data point x;
    b) Rnb(y)=Rnb(y)+1;
    c) $NN_r(x) = NN_r(x) \cup y$;
    d) $RNN_r(y) = RNN_r(y) \cup x$;
    e) Compute the number(Num) of data point x that Rnb(x)=0;
4) Until Num has not changed;
5) $sup_k = r$ and output the max Rnb(i);
```

Definition 1 Natural neighbor-NaN: Based on the Natural Neighbor searching algorithm, if point x belongs to the neighbors of point y and y belongs to the neighbors of point x, then x is called as y's Natural Neighbor (NaN). In the same way, y is Natural Neighbor of x.

Compared with the concept of neighbor that proposed by Stevens, the great advantage of Natural Neighbor is that the search method of Natural Neighbor is non-parameter. Through the above NaN-Searching algorithm, we can obtain two eigenvalue Rnb(x) and sup_k.

III. THE PROPOSED ALGORITHM

In this section, the proposed algorithm and its related concepts will be introduced in detail. Let D be a database, p and q be some objects in D, and K be a positive integer to indicate the number of neighbors of each object. The value of K is adaptively obtained by NaN-searching algorithm but manually set in all following concept and definition.

In order to find Natural Core Point, the proposed algorithm needs to compute the density for every point. Traditional density measurement, like DBSCAN, DP and DAAP used, is not applicable to dataset with great inter-class density variations [38] and need set neighborhood parameter manually.

In order to solve the above problem, in this paper, we introduce Natural Neighbor in density measurement. First, we use NaN-searching algorithm to compute the adaptive value of $K = sup_k$. Then we proposed a new density measurement, named Natural Density (NDen), defined as follows:

$$NDen(p) = \frac{1}{Dist_K(p)} \tag{1}$$

Therefore, we can gain the value of NDen(p) without K that manually set. Afterwards we divide the neighbors of every point into Natural Dense Neighbors (NDN) and Natural Sparse Neighbors (NSN), which are defined in Definition2.

Definition 2 NDN and NSN: If $NDen(q) > NDen(p)$ and $q \in KNN(p)$, then the object q is called the Natural Dense Neighbor of p, denoted as NDN(p). On the contrary, if $NDen(q) \leq NDen(p)$ and $q \in KNN(p)$, then q is called the Natural Sparse Neighbor of p, denoted as NSN(p).

Definition 3 Natural Exemplar: If object q satisfies the following conditions at once, then we call q is the Natural Exemplar of p.

1) $q \in Q = \{q|NDen(q) = max\{NDen(KNN(p))\}$ *and* $p \neq q\}$
2) *and* $d(p, q) = min_{q_i \in Q}\{d(p, q_i)\}$

From the definition of Natural Exemplar, we can know that each point possesses at most one Natural Exemplar. There is a special case that the Natural Density of p is greater than the Natural Density of all K nearest neighbors and reverse K nearest neighbors of p. Obviously, in this situation, p is the Natural Exemplar of itself. Then we call p a Natural Core Point that defined as follows:

Definition 4 Natural Core Point: If object p satisfies one of the following two conditions, then we call p a Natural Core Point (NCP).

1) $\forall q \in KNN(p), NDen(p) \geq NDen(q)$ *or*
2) $\forall q \in RKNN(p), NDen(p) \geq NDen(q)$

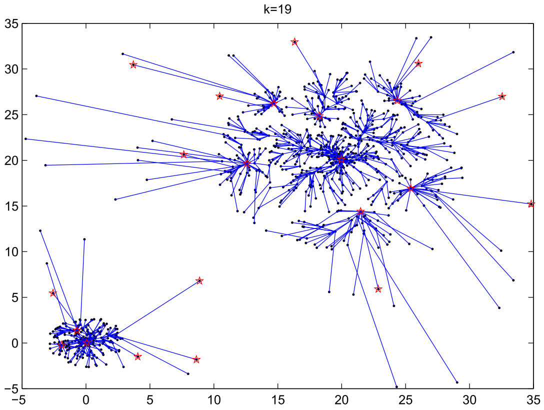


FIGURE 2. **Natural exemplar graph and natural core point.**

Fig.2 is the Natural Exemplar Graph (NEG) which can be comprised by connecting each point p to its Natural Exemplar. As shown in Fig.2, the value of K is 19 that adaptively gained by NaN-searching algorithm. The red points are Natural Core Point. Although some noise points are regarded as Natural Core Point, these red points will be regarded as noise points that will be explained in Algorithm 2. Natural Core Point is different from traditional cluster center. One traditional cluster center represent one final cluster. In other words, the number of cluster centers determines the number of final clusters, in traditional center-based clustering algorithm. Therefore, once obtain the wrong number of cluster centers, the final clustering result are unavailable. However, one cluster may have multiple Natural Core Points, that means the number of Natural Core Points does not directly determines the number of final clusters. The detailed clustering procedures will be described in the following content.

Algorithm 2 Initial-Clustering(D) //D Is the Unclassified Dataset

```
• Output: initial clustering results $C = \{c_1, c_2, \ldots, c_n\}$
1) Initiallizing: $r = 0, Dist_k(i) = 0, NDen(i) = 0, KNN(i) = \emptyset, RKNN(i) = \emptyset, NSN(i) = \emptyset, Exemplar(i) = i, NCP = \emptyset, visited(i) = false$;
2) K=NaN-searching(D);
3) for $\forall x \in D$
    a) find KNN(x) and RKNN(x);
    b) compute the $Dist_k(x)$ and $NDen(x)$;
    c) find the NSN(x);
4) endfor
5) for $\forall x \in D$
    a) y=max(NDen(KNN(x)));
    b) if $y \neq x$ then Exemplar(x)=y;
    c) else r=r+1 and NCP(r)=x;
    d) endif
    e) z=max(NDen(RKNN(x)));
    f) if $x == z$ then r=r+1 and NCP(r)=x;
6) endfor
7) for i=1 to r
    a) $c_i = NCP(i) \cup NSN(NCP(i))$;
    b) for $\forall x \in c_i$ i.
        i) if $visited(x) \neq true$ then visited(x)=true and $c_i = c_i \cup NSN(x)$;
    c) endfor
8) endfor
9) for i=1 to r
    a) if $|c_i| \leq K$
        i) all points of $c_i$ will be marked as noise;
        ii) delete $c_i$ from C;
    b) endif
10) endfor
```

As shown in Fig.2, we can roughly cluster the dataset and get initial clusters by Natural Exemplar Graph. The steps are descripted in Algorithm 2.

Firstly, Initial-Clustering algorithm uses NaN-searching algorithm to adaptively obtain the value of K. Secondly, Initial-Clustering algorithm finds KNN and RKNN of each point, and computes the natural density of each point. Thirdly, Initial-Clustering algorithm finds all Natural Core Points using Definition 4. After that, Initial-Clustering algorithm obtains the initial clusters via the following steps.

1) Initial-Clustering algorithm finds a unvisited Natural Core Point, and classifies it and its Natural Sparse Neighbors to the same cluster c_i.
2) Then Initial-Clustering algorithm arbitrarily finds a point p in c_i and classifies the Natural Sparse Neighbors of p to cluster c_i, until all points of this cluster have been visited.
3) Afterwards, Initial-Clustering algorithm repeats the above steps, until all Natural Core Points have been visited.
4) Finally, if $|c_i| <= k$, then delete c_i and mark all points of c_i as noise. Therefore, these noise points that regarded as Natural Core Points are still marked as noise points.

By doing so, we can obtain many initial clusters, but not final clusters. For example, as shown in Fig.2, the lower left cluster will be divided into 3 initial clusters since this cluster possess three Natural Core Points. In the same way, the upper right cluster will be divided into 7 initial clusters. In order to obtain the final clustering result, we construct a minimum spanning tree of Natural Core Point, that is minimum spanning tree of initial clusters. Each Natural Core Point represent an initial cluster. Therefore, we need to compute the distance between initial clusters. We define the distance between initial clusters as follows.

Definition 5 Distance between initial cluster: Distance between initial clusters C_i and C_j, denoted as $Dis(C_i, C_j)$, is defined as the ratio of $max(|C_i|, |C_j|)$ and $|C_i \cap C_j|$. The formulation of distance between clusters is shown as follows:

$$Dis\left(C_i, C_j\right) = \begin{cases} \frac{\max(|C_i|,|C_j|)}{|C_i \cap C_j|} & if\ C_i \cap C_j \neq \emptyset \\ \max\{Dis\left(C_i, C_j\right)\} + 1 & if\ C_i \cap C_j = \emptyset \end{cases} \tag{2}$$

Here, $|C_i|$ is the number of points in initial cluster C_i. We use Initial-Clustering algorithm to obtain the initial clusters that spread from dense areas to sparse areas. Therefore, some points that located in sparse areas will be classified to two or more initial clusters. As shown in Fig.3, the set of red points is the intersection of C_1 and C_2. The two blue points are the Natural Core Point of C_1 and C_2 respectively. $|C_1| = 169$, $|C_2| = 241$, $|C_i \cap C_j| = 17$. Therefore, based on the above formula, the distance between C_1 and C_2 is $Dis(C_1, C_2) = 241/17 = 14.1$.

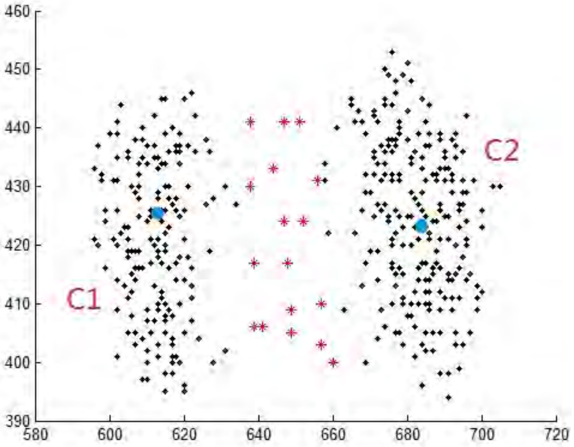


FIGURE 3. The intersection between C_1 and C_2.

After obtain the distance between all initial clusters, we construct minimum spanning tree of initial clusters, as shown in Fig.4. The red points are Natural Core Points. From Fig.4 we can see that two spherical classes have only one Natural Core Point, one spherical class have two Natural

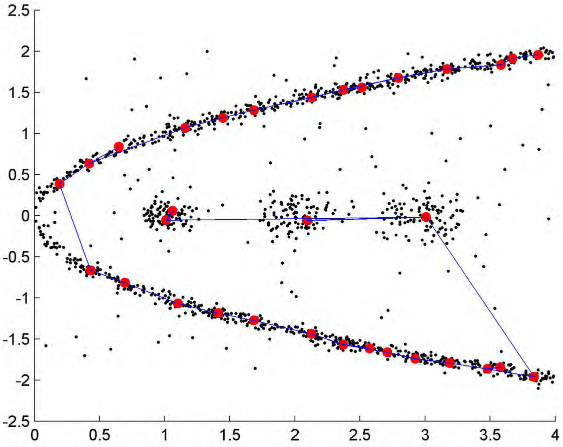


FIGURE 4. **Minimum spanning tree of initial clusters.**

Core Points. The complex manifold class have many Natural Core Points, and these Natural Core Points conform to the manifold distribution of this manifold class. It has to be noticed that these blue lines represent the distance between initial clusters but not Euclidean distance between points.

Based on the concept of Natural Core Point and the minimum spanning tree of initial clusters, we proposed a novel hybrid clustering algorithm, called NCP. The procedure of NCP algorithm is minutely described in Algorithm3.

First, the proposed clustering algorithm NCP uses Initial-clustering algorithm to roughly cluster the dataset and obtain the initial clusters. Secondly, NCP constructs the minimum spanning tree S of initial clusters. Then, as shown in Fig.5, NCP obtains the final cluster result via continually cut the maximum edge until the number of clusters is M that the only one manually set parameter needed by NCP.

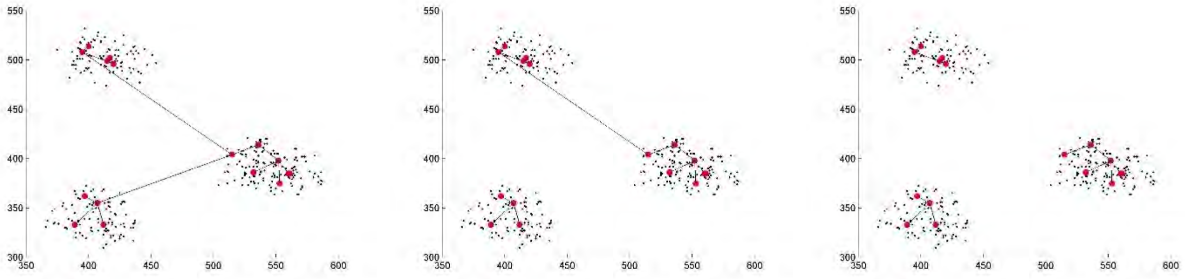
NCP require only one parameter that the number of clusters to cluster datasets. Moreover, NCP is applicable to datasets with complex manifold and great density variations. Different from the traditional MST-based clustering algorithms, NCP construct MST of natural core points instead of all points of datasets. Since NCP needs to get the neighbors for each point in dataset, the complexity of NCP is $O(N^2)$. If we use the K-D tree to search the neighbors of each point, the complexity of NCP would be decreased to $O(N * log^N)$. Fig.6 shows the running time of NCP when the number of the object in the uniformly distributed dataset increases (one thousand to twenty thousand).

Algorithm 3 NCP-Clustering(D,M) //D Is the Unclassified Dataset. M Is the Number of Final Clusters

```
• Output: Final clustering results $C = \{C_1, C_2, \ldots, C_M\}$
1) Initiallizing: r=1, C=D;
2) Obtaining initial clusters $\{c_1, c_2, \ldots, c_n\}$=Initial-Clustering(D);
3) Constructing minimum spanning tree S of initial clusters $\{c_1, c_2, \ldots, c_n\}$;
4) While $r < M$
    a) Find maximum edge $e(c_i, c_j)$ in S and cut it ($e(c_i, c_j) = 0$ );
    b) Find $C_a$ that $c_i, c_j \subset C_a$ and $C_a \subset C$;
    c) For $\forall p \in (c_i \cap c_j)$ i.
        i) If $Dis(p, NCP_i) < Dis(p, NCP_j)$ then p is classified into $c_i$;
        ii) Else p is classified into $c_j$
    d) endFor
    e) Merge $c_i$ and all initial clusters that connect with $c_i$ into $C_b$;
    f) Merge $c_j$ and all initial clusters that connect with $c_j$ into $C_d$;
    g) $C = C - C_a$;
    h) $C = C + C_b, C_d$;
    i) r=r+1;
5) endWhile
6) Output the final clustering results $C = \{C_1, C_2, \ldots, C_M\}$;
```

IV. PERFORMANCE EVALUATION

In order to evaluate the performance of NCP algorithm, we do experiments on synthetic data sets that contain clusters with various shapes, and Olivetti Face Database.



(a)

(b)

(c)

FIGURE 5. **Clustering procedure of NCP.**

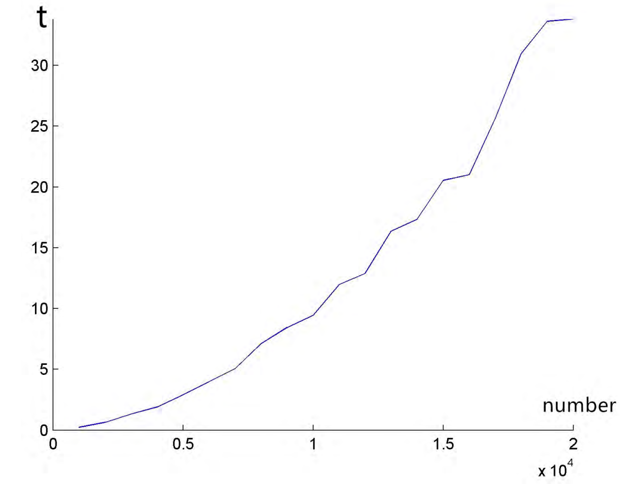


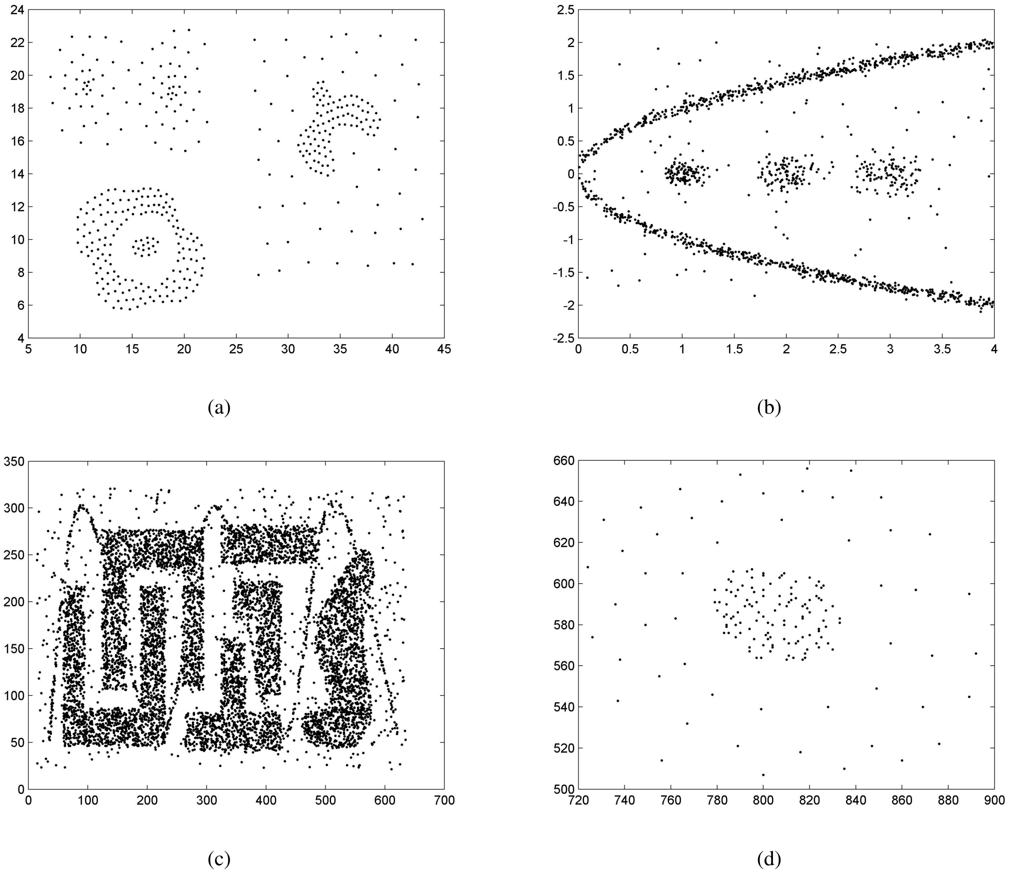
FIGURE 6. The running time of NCP when the number of object increases.

A. CLUSTERING ON SYNTHETIC DATA SETS

We demonstrate the effectiveness of NCP algorithm by comparing the proposed algorithm with DP, LOF-MST and DBSCAN on 4 complex synthetic data sets, illustrated in Fig.7. Data1 consists of three spherical classes, three manifold classes, a total of 399 points. Data2, taken from [41], consists of three spherical classes, one complex-manifold class and some noise points, a total of 1400 points. Data3, taken from [42], is composed of six high density manifold classes and some noise points, a total of 8000 points. Data4 consists of 168 points and has one dense spherical class and one sparse manifold class.

For DP, we show the best clustering result in repeated tests on Data1 and Data2, and decide on the right number of clusters to Data3 and Data4. Hence, we don't show the decision graph, deciding the number of the clusters, of DP in all results.

Fig.8 shows the clustering results of each approach on Data1. It reveals that DP and LOF-MST algorithm fail to discover the correct clusters. DP correctly cluster the spherical cluster of Data1. However, DP wrongly divides the manifold cluster into two clusters and merges the sparse cluster into adjacent dense cluster. Although LOF-MST ($K = 20, \alpha = 0.05, M = 6$) correctly cluster the manifold clusters and one spherical cluster, points in sparse cluster are regarded as noise points that marked with red color. Two spherical clusters located in upper left are incorrectly merged into one cluster by LOF-MST. DBSCAN (eps=0.4, minpoints=5) and NCP (M=6) algorithms correctly cluster on Data1.



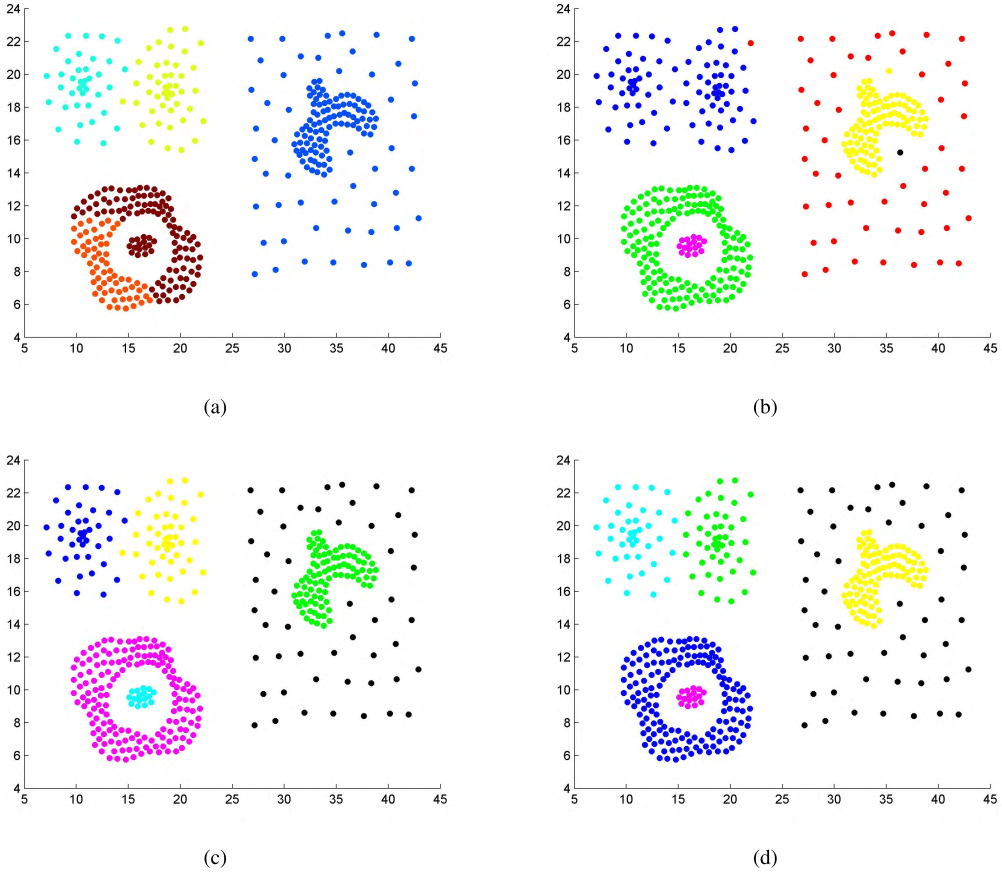
(a)

(b)

(c)

(d)

FIGURE 7. Four original synthetic datasets.



(a)

(b)

(c)

(d)

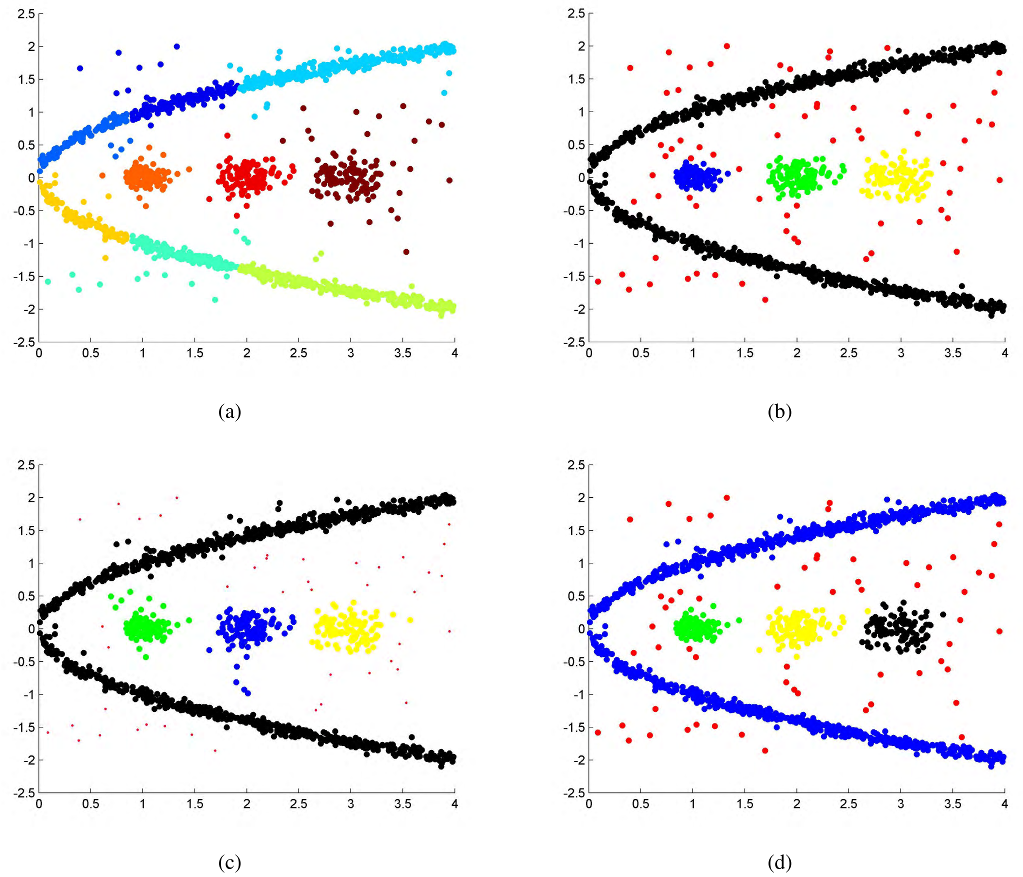
FIGURE 8. The clustering results of (a) DP, (b) LOF-MST, (c) DBSCAN, and (d) NCP algorithm on Data1.

Fig.9 shows the clustering results of each algorithm on Data2. From this figure, we can see that DP correctly clusters the spherical class, but fails to cluster the complex manifold cluster that is grouped into 6 clusters. LOF-MST ($K = 20, \alpha = 0.05, M = 4$) detect out the noise points in Data2 via manually set outlier parameter α, and correctly cluster Data2. DBSCAN ($eps = 0.2, minpoints = 20$) and NCP ($M = 4$) algorithms correctly cluster Data2 and detect out the noise points in Data2 without outlier parameter.

Fig.10 shows the clustering results of each algorithm on Data3. Although DP obtain the right number of clusters by manually select cluster centers in decision graph, three clusters are wrongly clustered by DP. LOF-MST ($k = 60, \alpha = 0.3, M = 6$) correctly clusters most of data points. However, many normal data points that located in the border of clusters are regarded as noise points by LOF-MST. Same as the result of LOF-MST, DBSCAN ($eps = 0.2, minpoints = 20$) correctly clusters most of data points, but some normal data points are regarded as noise points by DBSCAN. Although the proposed algorithm NCP ($M = 6$) regards some noise points as normal points, all normal points are correctly clustered by NCP. The performance of NCP is superior to DP, LOF-MST and DBSCAN on Data3.

Fig.11 shows the four algorithm's clustering results on Data4. Same as the result on Data3, although DP obtain the right number of clusters by manually select cluster centers in decision graph, the dense class is divided into two clusters by DP. Some normal points in sparse cluster are regarded as noise points by LOF-MST ($k = 10, \alpha = 0.06, M = 2$). Moreover, LOF-MST merges most of points in sparse cluster and dense cluster into one cluster, and wrongly regards two points, marked with blue, as a cluster. For the density variations of the two clusters in Data4 is great, DBSCAN ($eps = 17, minpoints = 3$) failed to correctly cluster Data4. DBSCAN wrongly regards most of normal points in sparse cluster as noise points that marked with red, and the rest points in sparse cluster are merged into dense cluster. If we decrease the value of scan radius eps, all points in sparse cluster may be regarded as noise points. The performance of NCP ($M = 2$) is obviously superior to DP, LOF-MST and DBSCAN on Data4.

From the above results and analysis, we can see that DP algorithms cannot deal with manifold datasets, although DP does not need parameters that manually set. LOF-MST algorithm increase the boundary distance between two adjacent clusters by removing noise points according to the value of LOF, so that it can discover complex manifold clusters.



(a)

(b)

(c)

(d)

FIGURE 9. The clustering results of (a) DP, (b) LOF-MST, (c) DBSCAN, and (d) NCP algorithm on Data2.

However, some normal points that density is very small or located in the boundary are recognized as noise points by LOF-MST. DBSCAN algorithm are able to cluster the datasets with arbitrary shapes. However, DBSCAN cannot deal with the datasets that the density variations of inter-clusters are great, and need manually set two parameter (*eps* and *minpoints*). So, from the result of artificial datasets, we can conclude that the scope of NCP's application is wider than DP, LOF-MST, DBSCAN algorithm. Based on the Natural Core Points and distance between initial clusters, no matter the datasets contain complex manifold clusters or great density variations of inter-clusters, the proposed algorithm NCP can get satisfactory clustering results without manually set parameter K that the number of neighbors. In order to demonstrate the effectiveness of NCP, we also experiment on real datasets as the follows section.

B. EXPERIMENTS ON REAL-WORLD DATA SETS

We also applied the proposed algorithm to real-world datasets that obtained from the University of California Irvine (UCI) machine learning repository, which include Iris, Cancer and Ecoli. The details are shown in Table 1.

TABLE 1. The characteristics real datasets.

Data sets	Number of data point	Dimension	Cluster
Iris	150	4	3
Cancer	569	30	2
Ecoli	336	7	8

In order to intuitively describe the efficiency of DP, MST-LOF, DBSCAN and NCP, we use the criteria of Purity, Recall, RI and F-measure to evaluate the clustering performance, which are defined as follows:

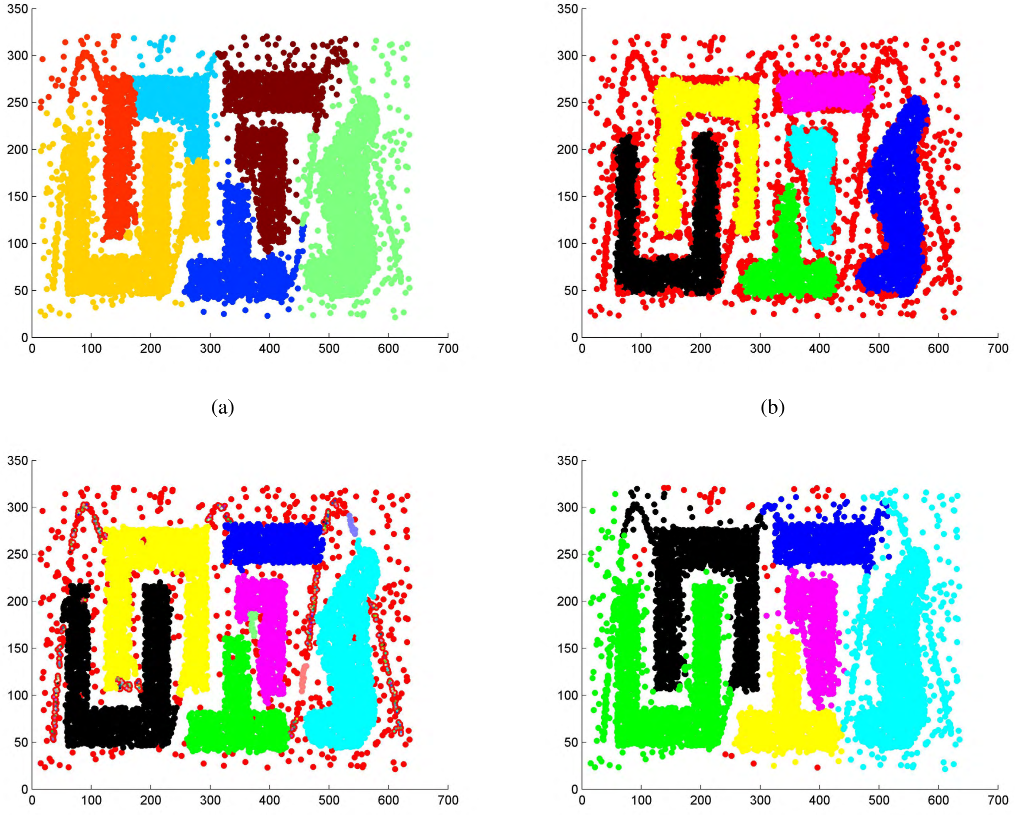
$$Purity = \frac{\sum_i^M \left(max_{tc_j \in TC}\left(\frac{|tc_j \cap c_i|}{|c_i|}\right)\right)}{M} \tag{3}$$

$$Recall = \frac{\sum_{i=1}^{TM} \left(max_{c_j \in C}\left(\frac{|tc_i \cap c_j|}{|tc_i|}\right)\right)}{TM} \tag{4}$$

$$RI = \frac{TP + TN}{TP + FP + FN + TN} \tag{5}$$

$$F - measure = \frac{2 * P * Recall}{P + Recall}, P = \frac{TP}{TP + FP} \tag{6}$$

Here, let D be a database and contains TM classes $TC = tc_1, tc_2, \ldots, tc_TM$. The result of clustering algorithm contains M clusters $C = c_1, c_2, \ldots, c_M$. $|c_i|$ is the number of points of c_i. TP is the number of point pair (p_i, p_j) that



(a)

(b)

(c)

(d)

FIGURE 10. The clustering results of (a) DP, (b) LOF-MST, (c) DBSCAN, and (d) NCP algorithm on Data3.

$p_i \in tc, p_j \in tc$ and $p_i \in c, p_j \in c$; TN is the number of point pair (p_i, p_j) that $p_i \in tc_i, p_j \in tc_j$ and $p_i \in c_i, p_j \in c_j$; FP is the number of point pair (p_i, p_j) that $p_i \in tc_i, p_j \in tc_j$ and $p_i \in c, p_j \in c$; FN is the number of point pair (p_i, p_j) that $p_i \in tc, p_j \in tc$ and $p_i \in c_i, p_j \in c_j$, $i \neq j$. P is the precision. The value of Purity, Recall, RI and F-measure is [0,1], the larger the value of Purity, Recall, RI and F-measure is, the better the clustering performance of the algorithm is.

TABLE 2. The performance comparison of four clustering algorithms on Iris.

	DP	MST-LOF	DBSCAN	NCP
Purity	0.89	0.83	0.75	0.92
Recall	0.74	0.59	0.6	0.81
RI	0.87	0.77	0.77	0.87
F-Measure	0.77	0.74	0.73	0.79

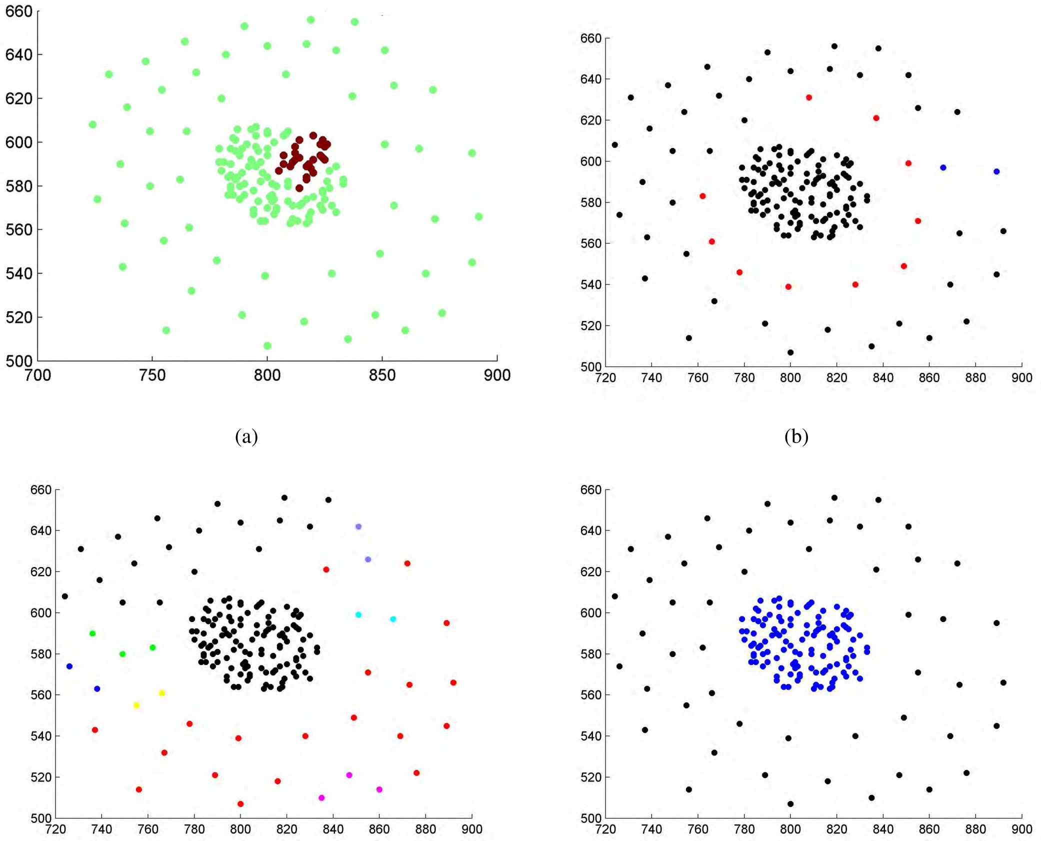
TABLE 3. The performance comparison of four clustering algorithms on Cancer.

	DP	MST-LOF	DBSCAN	NCP
Purity	0.81	0.81	0.64	0.83
Recall	0.53	0.53	0.53	0.75
RI	0.72	0.53	0.54	0.72
F-measure	0.69	0.69	0.69	0.72

TABLE 4. The performance comparison of four clustering algorithms on Ecoli.

	DP	MST-LOF	DBSCAN	NCP
Purity	0.80	0.88	0.44	0.61
Recall	0.38	0.30	0.28	0.62
RI	0.72	0.30	0.31	0.77
F-measure	0.52	0.44	0.43	0.60

Table 2, table 3 and table 4 show the Purity, Recall, RI and F-measure of various clustering algorithm on Iris, Cancer and Ecoli respectively. From table 2 and table 3, we can see that the value of Purity, Recall, RI and F-measure of NCP is the maximum on both two real-world datasets. Table 4 shows that the value of Recall, RI and F-measure of NCP is the maximum, although the value of Purity of NCP is not the maximum that obtained by MST-LOF. Although MST-LOF get well value of Purity, the value of RI, Recall and F-measure is undesirable. Although the value of Purity of NCP is not maximum, the value of NCP is not too bad. So, from the result of real-world datasets, we can conclude that the results of NCP is better than DP, LOF-MST, DBSCAN algorithm.



(a)

(b)

(c)

(d)

FIGURE 11. The clustering results of (a) DP, (b) LOF-MST, (c) DBSCAN, and (d) NCP algorithm on Data4.

C. CLUSTER ON OLIVETTI FACE DATABASE

In order to further demonstrate the effectiveness of NCP, we do experiments on the Olivetti Face Database. As a widely spread benchmark for machine learning algorithms, the dataset contains 400 faces images from 40 persons, taken at different time and with varying lighting, facial expressions and facial details. We select 100 faces, that is, 10 clusters to do the experiment. Same as the experiment on synthetic datasets, we compare NCP with DP, LOF-MST and DBSCAN algorithms in this experiment. We regard the correlation of picture A and B as the similarity between two images, denoted as S(A,B), the formulation as following equation.

$$S(A,B) = \frac{\sum_m \sum_n (A_{mn} - \overline{A})(B_{mn} - \overline{B})}{\sqrt{(\sum_m \sum_n (A_{mn} - \overline{A})^2)(\sum_m \sum_n (B_{mn} - \overline{B})^2)}} \tag{7}$$

Here A and B are the images in the Olivetti Face Database. A_{mn} and B_{mn} ($m = 1, 2, \ldots, 112$, $n = 1, 2, \ldots, 92$) represent the pixels of the subject images. The value of S is scaled between 0 and 1. The larger the value of S(A,B) is, the more similar A and B are. We define the distance between two images, denoted as d(A,B), as follows:

$$d(A,B) = 1 - S(A,B) \tag{8}$$

In order to intuitively describe the efficiency of DP, MST-LOF, DBSCAN and NCP on Olivetti Face Database, we also use the criteria of Purity, Recall and RI to evaluate the clustering performance.

Fig.12-15 show the clustering results of the DP, MST-LOF, DBSCAN and NCP algorithm on Olivetti Face Database respectively. In all results, faces with the same color belong to the same cluster. The value of Purity, Recall, RI and F-measure of the four algorithms is shown in Table 5.

TABLE 5. Comparison in terms of purity, recall and RI.

	DP	MST-LOF	DBSCAN	NCP
Purity	0.88	0.75	0.98	0.98
Recall	0.82	0.82	0.63	0.94
RI	0.93	0.79	0.87	0.98
F-measure	0.76	0.60	0.64	0.96

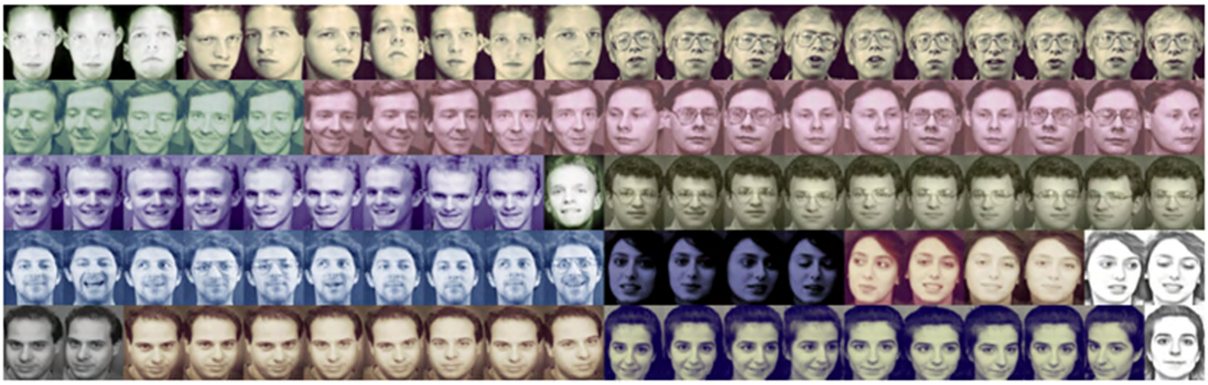


FIGURE 12. Clustering results of DP on Olivetti face database.

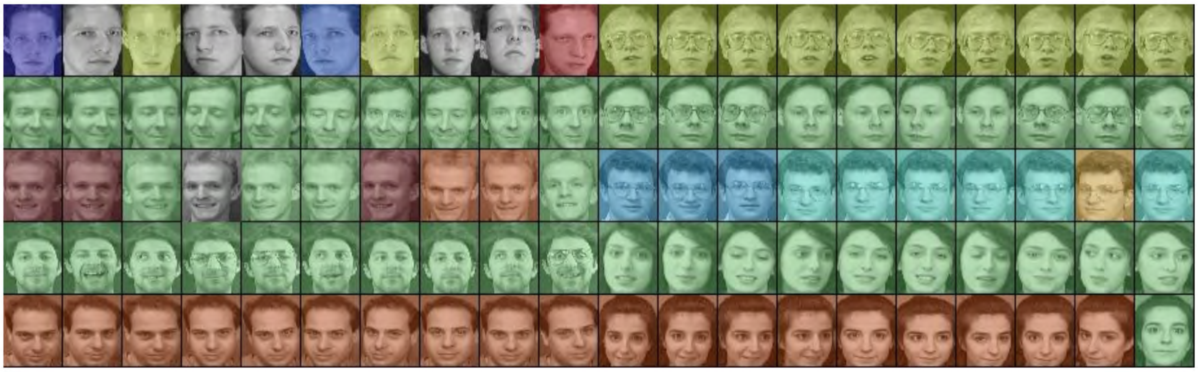


FIGURE 13. Clustering results of MST-LOF on Olivetti face database.

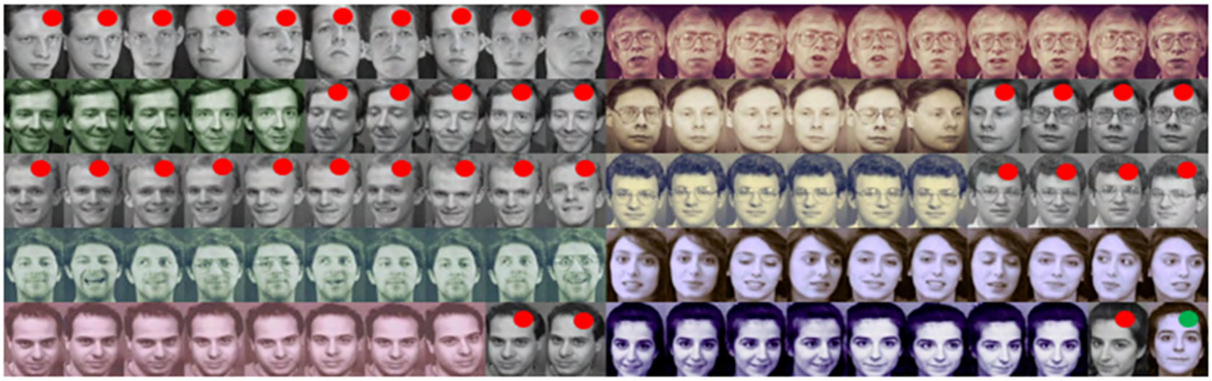


FIGURE 14. Clustering results of DBSCAN on Olivetti face database.

The results show that NCP get 11 clusters, because one face image that marked with yellow spot is wrongly regarded as one separate cluster. Three face images that within the red border are wrongly clustered. Two face images marked with red spot are considered as noise by NCP. Nonetheless, NCP clustering algorithm correctly cluster most of face images. The result of NCP is obviously superior to DP, MST-LOF and DBSCAN. As shown in Table 5, the value of Purity=0.98, Recall=0.94, RI=0.98, F-measure=0.96 of NCP are the maximum in the four clustering algorithms.

Through above experiments and analysis, we can get the conclusion that the proposed algorithm NCP outperforms DP, LOF-MST and DBSCAN algorithms. NCP is applicable to datasets containing complex-manifold clusters and clusters with great density variations. Therefore, NCP algorithm has broader application than DP, LOF-MST and DBSCAN algorithm.

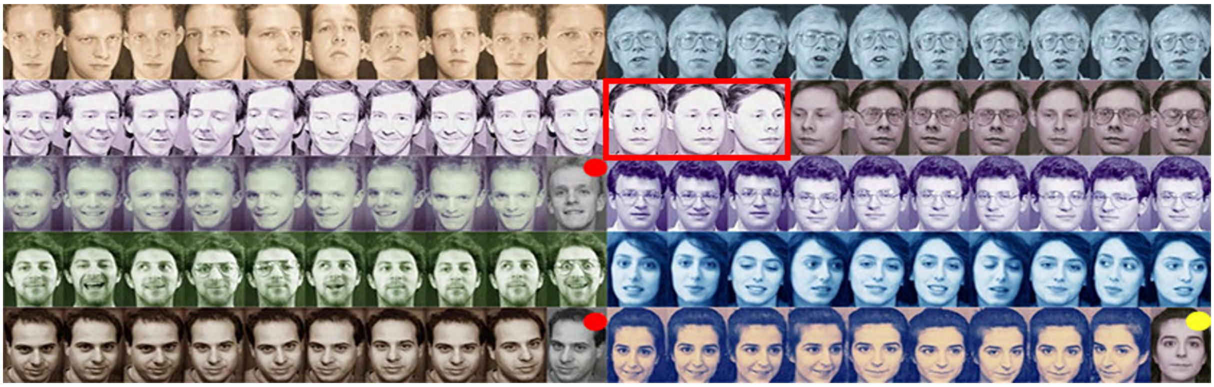


FIGURE 15. Clustering results of NCP on Olivetti face database.

V. CONCLUSIONS

In this paper, we propose a new hybrid clustering algorithm called NCP. The core idea of NCP is to search Natural Core Points and construct minimum spanning tree of Natural Core Points. In NCP, we first use natural neighbor to adaptively gain the neighborhood parameter K, and define a new concept that Natural Density to measure the local density of each object. Then we define the Natural Core Points, and obtain the initial clusters through spreading from natural core points to sparse areas. Each natural core point represent an initial cluster. After that, we define the distance between initial clusters, and construct minimum spanning tree of initial clusters (a.k.a. MST of natural core points). Afterwards, NCP obtain the final clusters via continually cut the maximum edges until the number of clusters is M that the only one manually set parameter needed by NCP. The experiments on synthetic data sets and the Olivetti Face Database demonstrate that NCP is applicable to datasets with complex-manifold patterns and great density variations, and NCP is more effective than DP, LOF-MST and DBSCAN.

REFERENCES

[1] A. W. Moore, "Very fast EM-based mixture model clustering using multiresolution KD-trees," *Adv. Neural Inf. Process. Syst.*, vol. 11, pp. 543–549, Jun. 1998.

[2] A. Doucet, N. de Freitas, and N. Gordon, Eds., *Sequential Monte Carlo Methods in Practice*. New York, NY, USA: Springer-Verlag, 2001.

[3] M. B. H. Rhouma and H. Frigui, "Self-organization of pulse-coupled oscillators with application to clustering," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 23, no. 2, pp. 180–195, Feb. 2001.

[4] H. Jiawei and K. Micheline, "Data mining: Concepts and techniques," *Data Mining Concepts Models Methods Algorithms*, vol. 5, no. 4, p. 1–18, 2006.

[5] L. Kaufman and P. J. Rousseeuw, "Finding groups in data: An introduction to cluster analysis," *J. Amer. Stat. Assoc.*, vol. 8, no. 2, pp. 277–279, 1990.

[6] R. T. Ng and J. Han, "CLARANS: A method for clustering objects for spatial data mining," *IEEE Trans. Knowl. Data Eng.*, vol. 14, no. 5, pp. 1003–1016, Sep. 2002.

[7] A. Hinneburg and D. A. Keim, "An efficient approach to clustering in large multimedia databases with noise," in *Proc. AAAI 4th Int. Conf. Knowl. Discovery Data Mining*, 1998, pp. 58–65.

[8] M. Ankerst, M. M. Breunig, H.-P. Kriegel, and J. Sander, "OPTICS: Ordering points to identify the clustering structure," *ACM Sigmod Rec.*, vol. 28, no. 2, pp. 49–60, 1999.

[9] T. Zhang, R. Ramakrishnan, and M. Livny, "BIRCH: An efficient data clustering method for very large databases," *ACM SIGMOD Rec.*, vol. 25, no. 2, pp. 103–114, 1996.

[10] S. Guha, R. Rastogi, and K. Shim, "CURE: An efficient clustering algorithm for large databases," *ACM SIGMOD Rec.*, vol. 27, no. 2, pp. 73–84, 1998.

[11] S. Guha, R. Rastogi, and K. Shim, "ROCK: A robust clustering algorithm for categorical attributes," in *Proc. 15th Int. Conf. Data Eng. (ICDE)*, Mar. 1999, pp. 512–521.

[12] D. Kushary, "The EM algorithm and extensions," *Technometrics*, vol. 40, no. 3, p. 260, 1998.

[13] A. K. Jain, "Data clustering: 50 years beyond K-means," *Pattern Recognit. Lett.*, vol. 31, no. 8, pp. 651–666, 2008.

[14] M. Ester, H.-P. Kriegel, and X. Xu, "A density-based algorithm for discovering clusters in large spatial databases with noise," in *Proc. Int. Conf. Knowl. Discovery Data Mining*, vol. 96, no. 34, pp. 226–231, 1996.

[15] W. W. Moss and J. A. Hendrickson, "Numerical taxonomy," *Encyclopedia Astrobiol.*, vol. 18, no. 1, pp. 227–258, 1973.

[16] B. King, "Step-wise clustering procedures," *J. Amer. Stat. Assoc.*, vol. 62, no. 317, pp. 86–101, 1967.

[17] D. Cheng, R. Kannan, S. Vempala, and G. Wang, "A divide-and-merge methodology for clustering," in *Proc. 24th ACM Sigmod-Sigact-Sigart Symp. Princ. Database Syst.*, 2005, pp. 1499–1525.

[18] G. Karypis, E.-H. Han, and V. Kumar, "Chameleon: Hierarchical clustering using dynamic modeling," *Computer*, vol. 32, no. 8, pp. 68–75, Aug. 1999.

[19] T. Kaukoranta, P. Franti, and O. Nevalainen, "Iterative split-and-merge algorithm for vector quantization codebook generation," *Opt. Eng.*, vol. 37, no. 10, p. 2726, 1998.

[20] J.-S. Lee and S. Olafsson, "Data clustering by minimizing disconnectivity," *Inf. Sci.*, vol. 181, no. 4, pp. 732–746, 2011.

[21] C.-R. Lin and M.-S. Chen, "Combining partitional and hierarchical algorithms for robust and efficient data clustering with cohesion self-merging," *IEEE Trans. Knowl. Data Eng.*, vol. 17, no. 2, pp. 145–159, Feb. 2005.

[22] M. Liu, X. Jiang, and A. C. Kot, "A multi-prototype clustering algorithm," *Pattern Recognit.*, vol. 42, no. 5, pp. 689–698, 2009.

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

[24] A. K. Jain and R. C. Dubes, "Algorithms for clustering data," *Technometrics*, vol. 32, no. 2, pp. 227–229, 1988.

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

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

[27] M. M. Breunig, "LOF: Identifying density-based local outliers," in *Proc. ACM SIGMOD Int. Conf. Manage. Data*, 2000, pp. 93–104.

[28] L. Caccetta and S. P. Hill, "A branch and cut method for the degree-constrained minimum spanning tree problem," *Netw. Int. J.*, vol. 37, no. 2, pp. 74–83, 2001.

[29] N. Päivinen, "Clustering with a minimum spanning tree of scale-free-like structure," *Pattern Recognit. Lett.*, vol. 26, no. 7, pp. 921–930, 2005.

[30] B. J. Frey and D. Dueck, "Clustering by passing messages between data points," *Science*, vol. 315, no. 5814, pp. 972–976, Feb. 2007.

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

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

[33] X. Zhang, W. Wang, K. Norvag, and M. Sebag, "K-AP: Generating specified K clusters by efficient affinity propagation," in *Proc. IEEE 10th Int. Conf. Data Mining*, Sydney, NSW, Australia, Dec. 2010, pp. 14–17.

[34] H. Jia, S. Ding, L. Meng, and S. Fan, "A density-adaptive affinity propagation clustering algorithm based on spectral dimension reduction," *Neural Comput. Appl.*, vol. 25, nos. 7–8, pp. 1557–1567, 2014.

[35] S. S. Stevens, "Mathematics, measurement, and psychophysics," in *Handbook of Experimental Psychology*, S. S. Stevens, Ed. Oxford, U.K.: Wiley, 1951, pp. 1–49.

[36] Q. Zhu, J. Feng, and J. Huang, "Natural neighbor: A self-adaptive neighborhood method without parameter K," *Pattern Recognit. Lett.*, vol. 80, pp. 30–36, Sep. 2016.

[37] D. Cheng, Q. Zhu, J. Huang, L. Yang, and Q. Wu, "Natural neighbor-based clustering algorithm with local representatives," *Knowl.-Based Syst.*, vol. 123, pp. 238–253, May 2017.

[38] J. Huang, Q. Zhu, L. Yang, D. Cheng, and Q. Wu, "QCC: A novel clustering algorithm based on quasi-cluster centers," *Mach. Learn.*, vol. 106, no. 3, pp. 337–357, 2017.

[39] J. Huang, Q. Zhu, L. Yang, and J. Feng, "A non-parameter outlier detection algorithm based on Natural Neighbor," *Knowl.-Based Syst.*, vol. 92, no. C, pp. 71–77, 2016.

[40] L. Yang, Q. Zhu, J. Huang, and D. Cheng, "Adaptive edited natural neighbor algorithm," *Neurocomputing*, vol. 230, pp. 427–433, Mar. 2017.

[41] J. Ha, S. Seok, and J.-S. Lee, "Robust outlier detection using the instability factor," *Knowl.-Based Syst.*, vol. 63, pp. 15–23, Jun. 2014.

[42] C. Cassisi, A. Ferro, R. Giugno, G. Pigola, and A. Pulvirenti, "Enhancing density-based clustering: Parameter reduction and outlier detection," *Inf. Syst.*, vol. 38, no. 3, pp. 317–330, 2013.

JINLONG HUANG received the bachelor's degree from Chongqing Jiaotong University, in 2011, and the master's and Ph.D. degrees from Chongqing University, in 2014 and 2017, respectively. He is currently a Lecturer with the College of Big Data and Intelligent Engineering, Yangtze Normal University. His research interests include data mining, clustering analysis, and outlier detection.

RU XU received the bachelor's degree from the Department of Computer, Yangtze Normal University, China, in 2006, and the master's degree in computer engineering from Chongqing University, China, in 2010. He has been a Software Developer with Shanghai Baosight Software Co., Ltd, China, from 2011 to 2012. He is currently an Experimentalist with the College of Big Data and Intelligent Engineering, Yangtze Normal University. His main research interests include algorithm and program design, data mining, database systems, and agricultural informationization.

DONGDONG CHENG received the bachelor's degree from Chongqing Normal University, in 2013, and the Ph.D. degree from Chongqing University, in 2018. She is currently a Lecturer with the College of Big Data and Intelligent Engineering, Yangtze Normal University. Her research interests include clustering analysis and data mining.

SULAN ZHANG received the B.S. degree from the Department of Computer Science and Technology, Southwest University, China, in 2006, the master's degree in computer software and theory and the Ph.D. degree in computer science and technology from Chongqing University, China, in 2009 and 2013, respectively. She was a Visiting Scholar with the Centre for Horticultural Science, The University of Queensland, Australia, from 2017 to 2018. She is currently an Associate Professor with the College of Big Data and Intelligent Engineering, Yangtze Normal University, China. Her main research interests include data mining, data analysis, and computer modeling and simulation.

KEKE SHANG received the Ph.D. degree from the School of Marine Science and Technology, Northwestern Polytechnical University, in 2017. He is currently an Assistant Researcher with the Computational Communication Collaboratory, School of Journalism and Communication, Nanjing University. His current research interest includes network science.