IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received April 21, 2019, accepted May 14, 2019, date of publication May 24, 2019, date of current version June 20, 2019.

*Digital Object Identifier 10.1109/ACCESS.2019.2918772*

# HCFS: A Density Peak Based Clustering Algorithm Employing A Hierarchical Strategy

**LINLIN ZHUO[1,2], KENLI LI[1,2], (Senior Member, IEEE), BO LIAO[1,2], HAO LI[1,2], (Student Member, IEEE), XIAOHUI WEI[1,2], AND KEQIN LI[3], (Fellow, IEEE)**

[1]College of Computer Science and Electronic Engineering, Hunan University, Changsha 410008, China
[2]National Supercomputing Center, Changsha 410082, China
[3]Department of Computer Science, State University of New York, New Paltz, NY 12561, USA

Corresponding authors: Kenli Li (lkl@hnu.edu.cn) and Bo Liao (dragonbw@163.com)

This work was supported in part by the National Key Research and Development Program of China under Grant 2016YFB0201303 and Grant SQ2018YFB020061, in part by the National Outstanding Youth Science Program of the National Natural Science Foundation of China under Grant 61625202, in part by the Natural Science Foundation of Hunan Province, China, under Grant 2018JJ2063, in part by the Program of National Natural Science Foundation of China under Grant 61751204, and in part by the International (Regional) Cooperation and Exchange Program of National Natural Science Foundation of China under Grant 61661146006 and Grant 61860206011.

**ABSTRACT** Clustering, which explores the visualization and distribution of data, has recently been widely studied. Although current clustering algorithms such as DBSCAN, can detect the arbitrary-shape clusters and work well, the parameters involved in these methods are often difficult to determine. Clustering using a fast search of density peaks is a promising technique for solving this problem. However, the current methods suffer from the problem of uneven distribution within local clusters. To solve this problem, we propose a new density peak based clustering algorithm employing a hierarchical strategy, namely, HCFS, which consists mainly of two stages. In the first stage, the HCFS estimates the density and distance of each point. The points with higher density and distance are selected as candidate centers, and then subclusters centered on them are further obtained. In the second stage, considering that adjacent subclusters based on certain candidate centers are highly similar and connected within the same cluster, we propose a new mechanism for measuring dissimilarity and connectivity between the subclusters. Those highly similar and connected subclusters are merged to increase the dissimilarity between different clusters and to obtain the final clustering results. The experiments conducted on a large number of datasets show that our method can effectively identify unevenly distributed clusters and yield better or comparable performance for different datasets.

**INDEX TERMS** Cluster, candidate center, density peak based, hierarchical, merge, subclusters, two-stage algorithm, uneven distribution within local clusters.

## I. INTRODUCTION

### A. MOTIVATION

Clustering plays an important role in data mining due to the existence of a large number of unlabeled datasets. It is an effective technique for discovering the potential structure of these datasets. Hence, clustering algorithms are used in many applications [1], [2]. The clustering algorithms that are used to solve different problems are usually based on different strategies. The most popular algorithm is K-Means [3], which detects the spherical clusters by minimizing the distance objective function iteratively. Although the implementation of the K-Means algorithm is easy, it has the obvious drawback that it cannot work with non-spherical datasets very well. Then, some density based [4] and hierarchy based algorithms [5]–[7] that can recognize the non-spherical clusters with multiple appropriate parameters were proposed. In addition, some attempts were made to reduce the number of parameters, such as employing the reverse nearest neighbors technique in [8]–[10] and finding the density peaks in [11]. But the performance of these algorithms suffers from the problem of uneven distribution within local clusters.

### B. RELATED RESEARCH

Chameleon [5] is a classical clustering algorithm based on hierarchy and graph partition, which consists of two parts. This method first employs the graph-cut technique to

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

construct subclusters. Then, the subclusters that are most likely to be in a same cluster are merged by considering their interconnectivity and closeness simultaneously. It can recognize non-spherical clusters. However, it is difficult to set multiple appropriate parameters in this method. The applications related to the Chameleon algorithm [12]–[14] are also affected by this limitation. Unfortunately, there is a lack of research on how to obtain the appropriate parameters at present.

In density based algorithms, high density regions are clustered together to construct clusters. A cluster is separated from the other clusters by a low density region. DBSCAN [4] can discover the non-spherical clusters and noise points at the same time. Current studies have shown that DBSCAN can obtain high quality clustering results with appropriate parameters [15], [16]. However, this algorithm requires two parameters to be set manually. One parameter is the neighbor radius *Eps* and the other is the minimum vertex number *MinPts* within radius *Eps*. Setting the appropriate values for these two parameters is difficult.

To reduce the number of parameters used in DBSCAN, different algorithms were proposed. RECORD [8] employs the reverse nearest neighbors technique and the strongly connected components to discover the non-spherical clusters. ISDBSCAN [9] and ISBDBSCAN [10] are variants of DBSCAN, and they also employ the reverse nearest neighbors technique to estimate the density of points. The above three algorithms only need to adjust one parameter $k$, which is the number of nearest neighbors. Although these algorithms reduce the number of parameters, they cannot effectively discover the underlying structure of some datasets (e.g., Flame [17], a kind of DNA dataset). There are two main reasons that contribute to this situation, which are as follows: 1) a fixed and predetermined threshold is used to determine the core observation from a global perspective ($Core = \left\{v \in V \mid |IS(v)| > \frac{2k}{3}\right\}$ used in ISDBSCAN and ISBDBSCAN, $Core = \{v \in V \mid outdegree(v) \geq k\}$ used in RECORD); and 2) these methods do not take the existence of uneven distribution within local clusters into account.

Clustering by fast search and find of density peaks (CFS) [11] is another density based algorithm and uses the decision graph to find the density peaks. CFS can recognize the non-spherical clusters. There are two basic assumptions for identifying centers as follows: 1) the cluster centers are surrounded by neighbors with lower density; 2) the cluster centers have relatively high distance from the nearest neighbor with higher density. In CFS, only one parameter, namely, the average percentage of neighbours, is involved, thus mitigating the effect of parameters to some extent. However, the CFS algorithm does not consider the problem of uneven distribution within local clusters, which causes its performance to still be affected.

A New Density Kernel in Density Peak Based Clustering (NCFS) [18] is a variant of the CFS algorithm. The NCFS algorithm points out that it is difficult to determine the centers using the decision graph, which is due to the difficulty of differentiating between the 'high' and 'low' $\rho$'s and $\delta$'s in this graph. As a result, the number of clusters can not be determined automatically. Fortunately, some methods [19] [20] have been proposed to solve the problem of finding the correct number of centers. Then the NCFS algorithm employs the techniques of the reverse nearest neighbors and density normalization to more accurately determine the centers. However, the NCFS algorithm does not consider the problem of uneven distribution within local clusters, which results in its performance being limited.

### C. OUR CONTRIBUTIONS

The performance of the algorithms related to CFS is limited in two ways, for example, the determination of the centers based on the decision graph and the uneven distribution within local clusters. To overcome the above problems, a new density peak based algorithm employing a hierarchy strategy is proposed. Our algorithm consists mainly of two stages. In the first phase, the points with high $\rho$'s and $\delta$'s on the decision graph are selected as candidate centers. Then, subclusters centered on these candidate centers are constructed. Unlike the CFS algorithm, at this stage, these candidate centers that are not real centers will also not be directly assigned to other clusters. Consequently, the subclusters centered on these candidate centers, which are not real centers, will also not be merged into the wrong clusters. In the second phase, a new strategy to estimate the dissimilarity and connectivity between two adjacent subclusters is adopted. If the two subclusters are highly similar and connected, they will be merged. This process is repeated until the number of the final clusters is decreased to the correct number of the centers. The contributions of our work are summarized as follows:

1) We represent a new density peak based clustering algorithm employing a hierarchy strategy to solve the problem of uneven distribution within local clusters. Our algorithm only needs to find a set of candidate centers without determining the real centers, which indirectly solves the problem of determining the real centers from the set of candidate centers.
2) We propose a new strategy to estimate the dissimilarity and connectivity simultaneously between two adjacent subclusters. In this strategy, only one equation is used to determine whether the pair of adjacent subclusters are similar and connected or not, which means that our algorithm only needs to adjust one parameter.
3) We test our algorithm on eleven datasets containing the non-spherical clusters and unevenly distributed local clusters. The experimental results show that our algorithm can obtain better or comparable clustering results when compared with other comparison algorithms, which proves the effectiveness of our algorithm.

The remainder of this paper is organized as follows. In section II, the related concepts of the density peak based algorithms is presented. Section III describes the details of

our proposed clustering algorithm. The experimental results and analysis on several datasets are provided in Section IV. The conclusion is summarized in Section V.

## II. DENSITY PEAK BASED CLUSTERING ALGORITHMS

To describe our algorithm more clearly, some related concepts of two density peak based clustering algorithms, such as CFS and NCFS, will be described next.

### A. CFS CLUSTERING ALGORITHM

CFS is a typical density peak based algorithm. It can find the centers on the decision graph and recognize the non-spherical clusters. CFS relies on two assumptions. The first is that the centers might have higher densities and the second is that the centers are at relatively large distance from the nearest neighbor with higher local density. Some detailed definitions are provided as follows.

1) Distance matrix $D$: The Gauss distance between any two points in a given dataset constitutes a distance matrix $D = \left(d_{ij}\right)$. The symbol $d_{ij}$ represents the distance between the $i$-th and $j$-th points and is calculated by the following equation:

$$d_{ij} = \left(\sum_{k=1}^{dim}(x_{ik} - x_{jk})^2\right)^{\frac{1}{2}} \tag{1}$$

where $dim$ represents the number of features of the point.

2) The average percentage of neighbours $p$ and cut-off distance $d_c$: $p$ denotes the average percentage of neighbours, and it is empirically set to be approximately 1% to 2%. The cut-off distance $d_c$ is the neighborhood radius and can be calculated as:

$$d_c = sort\,(X)_{cut} \tag{2}$$

where

$$X = \left\{d_{ij} \mid i, j \in S \text{ and } i < j\right\} \tag{3}$$

and

$$cut = round\,(\|S\| \times p) \tag{4}$$

In formulas (3) and (4), the name of the function is consistent with that of the function in MATLAB. $S$ represents the entire dataset, $\|S\|$ denotes the number of the points in $S$.

3) Density of the point $i$ $\rho_i$: There are two kernels to estimate the density of points, i.e., the cut-off kernel and the Gaussian kernel. For a point $i$, its cut-off kernel density is defined as:

$$\rho_i = \sum_{j \in S} \chi(d_{ij} - d_c) \tag{5}$$

where

$$\chi(x) = \begin{cases} 1, & x < 0 \\ 0, & \text{otherwise} \end{cases} \tag{6}$$

and its Gauss kernel density is defined as:

$$\rho_i = \sum_{j \in S} \left(-\frac{d_{ij}^2}{d_c}\right) \tag{7}$$

4) The distance of the point $i$ $\delta_i$: $\delta_i$ is defined as the distance between point $i$ and its nearest neighbor with higher density as follows:

$$\delta_i = \min_{j \in S, \rho_j > \rho_i} d_{ij} \tag{8}$$

Note that if the point $i$ has the highest density, then its distance $\delta_i$ should be calculated as:

$$\delta_i = \max_{j \in S} d_{ij} \tag{9}$$

5) Decision graph: The decision graph is constructed by the coordinate pairs $(\rho, \delta)$ of the points, and the centers are determined on the decision graph.

CFS first loads the distance matrix $D$ and calculates the cut-off distance $d_c$ by using an artificial parameter $p$. Based on the above distance matrix and cut-off distance, the density $\rho$'s of all points can be further obtained by Eqs. (5) or (7), and the distance $\delta$'s of all points can be obtained by Eq. (8). Then, the decision graph is constructed to find the centers. Finally, each remaining point is assigned to the same cluster as its nearest neighbor with higher density.

As shown in Fig. 1(c), most points either have small $\rho$'s or small $\delta$'s and are concentrated in a narrow region. Obviously, only a small number of points have high $\rho$'s and $\delta$'s, and they are apart from the narrow region and are suitable as candidate centers. Therefore, the centers are determined by employing two thresholds empirically, which is also the core of the CFS algorithm. However, in this small set, there are usually other candidate centers in addition to the real centers. How to choose the appropriate centers from this set is a big challenge for CFS, which is due to the difficulty of differentiating between the 'high' and 'low' on decision graph. The method further proposes the equation $\lambda_i = \rho_i \times \delta_i$ to represent the qualification of a point as the center. However, this equation is not able to determine which variable is more important. As a result, some points with high $\delta$'s but small $\rho$'s are selected as the centers by error.

Additionally, it should be mentioned that the impact of statistical errors (possibly data loss) on performance can be mitigated by increasing the value of the parameter $p$. For example, all points can be classified correctly when the parameter $p$ is increased to the range [2.7%, 4.0%] in the Flame dataset while many points are misclassified when the average percentage of neighbours $p$ is set to 2%. More importantly, the clustering results are not satisfactory no matter how the parameter $p$ is adjusted in the other datasets containing unevenly distributed local clusters. For example, CFS can not work well with some relatively complex datasets, such as UCI datasets Chameleon t4.8k [5] and Pathbased1 [21] even if $p$ is adjusted in the range of [0%, 4%]. Therefore, adjusting the value of the parameter is not a typical strategy for the datasets containing unevenly distributed local clusters.

### B. NCFS CLUSTERING ALGORITHM

NCFS is another density peak based clustering algorithm. In this method, the equation $\lambda_i = \rho_i \times \delta_i$ is considered as a

reasonable way to select the centers. According to Eq. (8) and the above equation, the local density $\rho$ plays a key role in the calculation of $\lambda$. Different estimation methods of $\rho$ will result in different $\delta$ and then different $\lambda$, which will further lead to the selection of different centers. NCFS proposed a density estimation method based on the reverse nearest neighbors technique to determine more appropriate centers. For a point $i$, its local density $\rho_i$ can be calculated as follows:

$$\rho_i = \frac{d_{max}}{\frac{1}{\|S_0\|}\sum_{j\in S_0} d_{ij}} \tag{10}$$

where

$$d_{max} = max_{m,n\in S} d_{mn} \tag{11}$$

$S_0$ represents a subset of top $h$ farthest points among $k$ nearest neighbors of point $i$. In this method, $h$ and $k$ are empirically determined as 4 and 30, respectively. By Eq. (10), the farthest neighbors of a point will determine if this point is suitable for being selected as the center.

The NCFS method asserts that if the density distribution of local clusters is different, then the density will not be accurately estimated since Eq. (10) is a global method obviously. Therefore, a density estimation method based on density normalization was further proposed. For a point $i$, its density $\rho_i'$ can be calculated as follows:

$$\rho_i' = \frac{\rho_i}{\frac{1}{k}\sum_{j\in S_{knn}} \rho_j} \tag{12}$$

where $S_{knn}$ is a subset of $k$ nearest neighbors of point $i$. According to Eq. (12), the density of a point relies on the density of its nearest neighbors. Then, the equation $\lambda_i = \rho_i \times \delta_i$ is used to determine the appropriate centers.

The NCFS clustering algorithm employed the reverse nearest neighbors and density normalization to estimate the local density of a point, and then further to determine more appropriate centers. As a result, its performance is improved compared with the original CFS algorithm. However, the NCFS algorithm does not consider the existence of uneven distribution within the local clusters. Hence, the method is not capable of identifying the underlying structure of some datasets containing unevenly distributed local clusters.

## III. OUR ALGORITHM

A Density Peak Based Clustering Algorithm Employing a Hierarchical Strategy (HCFS) will be discussed in this section. HCFS is an improved variant of the CFS algorithm, and the HCFS algorithm will be represented in the following aspects: 1) analyzing the reasons of the problem of uneven distribution within local clusters in detail; 2) determining the set of candidate centers and constructing subclusters based on these candidate centers; and 3) merging the subclusters based on a new dissimilarity estimation strategy. Finally, the overall process of the algorithm and a simple example are presented.

### A. PROBLEM DESCRIPTION

In this section, we first describe the procedure of misclassification. Then, the reasons for this situation will also be discussed in detail.

The points might be misclassified due to the existence of uneven distribution within local clusters, as shown in Fig. 1(a). Fig. 1(b) shows the corresponding ground truth of the clusters. To describe the misclassification procedure in detail, more in-depth analysis is conducted. First, 6 points that are marked as A, B, C, D, E, F are selected as candidate centers on the decision graph, as shown in Fig. 1(c). These candidate centers undoubtedly have the highest density in the respective local subclusters. After these 6 candidate centers are selected, the clustering results are shown as Fig. 1(d). The marked points in Fig. 1(c) correspond to the star points of the same color in Fig. 1(d), respectively. As the correct number of clusters is 2, and only the brown and green points in Fig. 1(c) are the real centers. Therefore, a further assignment of points is in need. For the star points C, D, E and F, their nearest neighbors with higher density are points I, K, H and G, respectively. With the Eq. (8), the distance $\delta_C$, $\delta_D$, $\delta_E$, and $\delta_F$ of points C, D, E, and F, are calculated respectively as follows:

$$\begin{aligned} \delta_C &= d_{CI} \qquad \delta_D = d_{DK} \\ \delta_E &= d_{EH} \qquad \delta_F = d_{FG} \end{aligned} \tag{13}$$

For the point C, its nearest neighbor with higher density is point I, and the point I is obviously located in the cluster centered on point A. Therefore, the point C is assigned to the cluster centered on point A. Then, the red cluster centered on point C is merged into the brown cluster centered on point A. Similarly, blue cluster centered on point D are merged into brown cluster centered on point A, and pale blue and yellow clusters centered on points E and F are merged into the green cluster centered on point B. However, points C and D along with the clusters centered on them should have been assigned to the same cluster as point B. Therefore, this leads to an error of the assignment as shown in Fig. 1(a). In the following parts, the reasons for this situation will be discussed in detail.

There are usually more points with high $\rho$'s and $\delta$'s on decision graph in addition to the real centers, which is due to some statistical errors. And it is also the fundamental reason for the uneven distribution of density in local clusters. Then, misclassification will occur in the assignment stage. Generally, the density of the point that is farther away from the center should also be smaller within a local cluster. However, if there is one or several unevenly distributed local clusters within a dataset, the rule will be broken. For example, all red points in region N are closer to the real center B than point C in Fig. 1(d), but their density is significantly lower than point C. Similarly, the density of all points in region M is also lower than the density of point D. Additionally and worse, for some points, their nearest neighbors with higher density are usually in the wrong cluster. For example, the nearest neighbor with higher density of point C is located in the cluster centered on

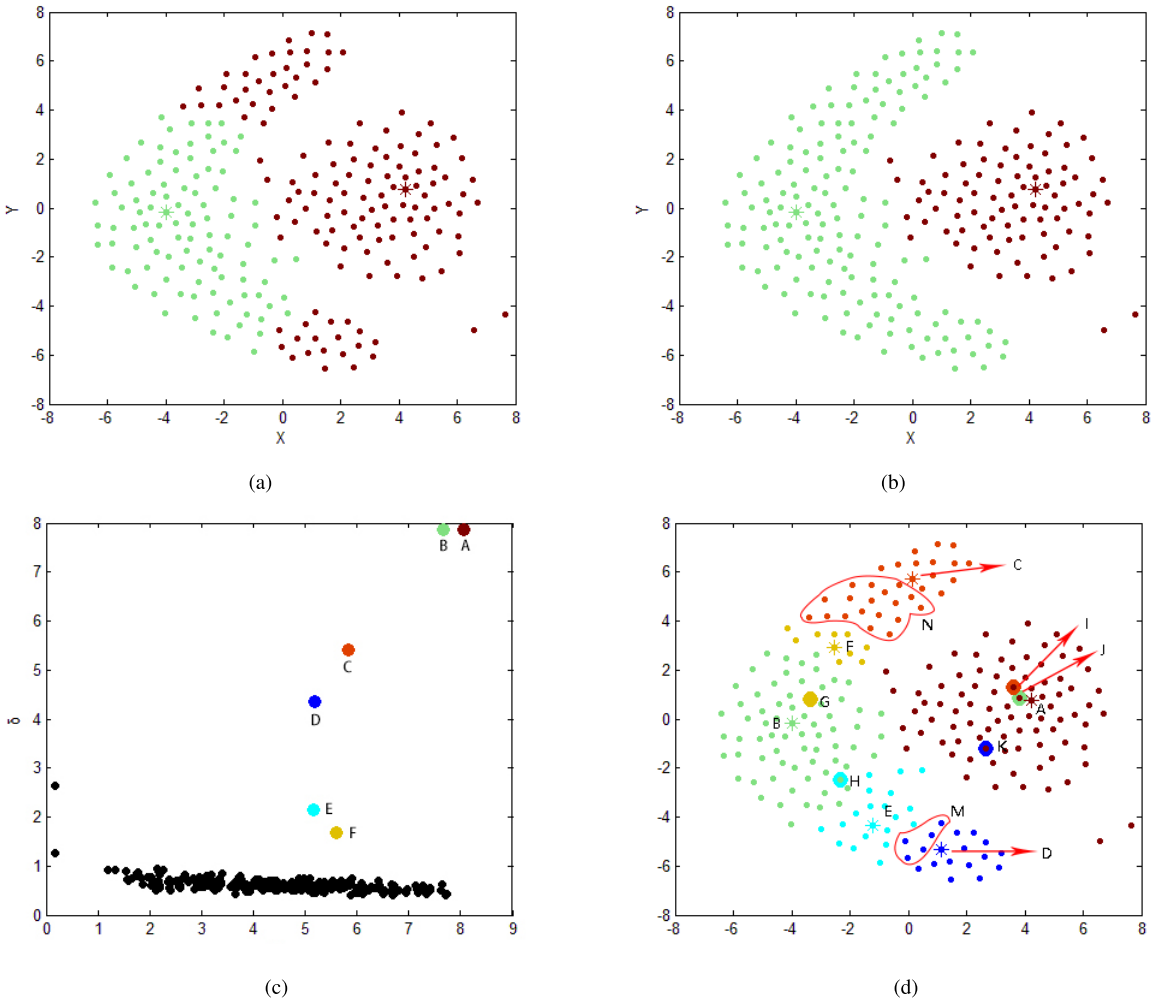



**FIGURE 1.** (a) The clustering results obtained performing CFS algorithm on the flame dataset when only two centers are selected. (b) Ground truth of the flame dataset; the star points are the real centers; and our algorithm can obtain results as demonstrated in this figure. (c) The decision graph obtained using CFS algorithm on the flame dataset. 6 central points are colored nonblack, while the remaining points are black. (d) The result if 6 central points are selected. The star points in the figure correspond to the center point of each subcluster. For star points B, C, D, E and F, their nearest neighbors with higher density are points J, I, K, H and G, respectively. The average percentage of neighbours $p$ is set to 2%, and there is no error in constructing subclusters.

point A, whereas the nearest neighbor with higher density of point C should have been within the same cluster as point B. The case of point D is similar to that of point C. And then, points C and D are mistakenly assigned to the same cluster as point A.

Although many datasets obey the Gauss distribution in general, the points might be unevenly distributed in some local clusters, which leads to assigning the points to a wrong cluster no matter how the parameter $p$ is adjusted. According to the above analysis, a conclusion can be drawn as follows: If there one or several local clusters that are not evenly distributed in a dataset, in addition to the real centers, there will be other noncenter points with high $\rho$'s and $\delta$'s. Some of these noncenter points will be assigned to the wrong clusters as their nearest neighbors with higher densities exist in the wrong clusters, which can lead to misclassification.

To alleviate this problem of misclassification caused by uneven distribution within local clusters, a feasible solution is presented as follows: All points with higher $\rho$'s and $\delta$'s within a local cluster should be selected as the candidate centers. Then the subclusters centered on these points will be constructed immediately. Note that each of these points is not assigned to the same cluster as its nearest neighbor with higher density. Fortunately, the adjacent subclusters are usually highly similar and connected within the same cluster (Even if an uneven distribution occurs in the local clusters). Based on this situation, if these adjacent subclusters are similar and connected, they should be merged into the same cluster.

### B. CONSTRUCTING THE SUBCLUSTERS

This section presents a method to find a set of candidate centers. After this set is determined, each remaining point should

be assigned to the same subcluster as its nearest neighbor with higher density. The subclusters will then be constructed centering on these candidate centers.

As analyzed in Section III.A, if each cluster within a dataset is approximately evenly distributed, only the real centers will have high $\delta$'s while the other noncenter points will have small $\delta$'s. In this case, the candidate centers are also the real centers. In contrast, in addition to the real centers, the other points that have higher $\rho$'s and $\delta$'s should also be selected as candidate centers. In other words, if there is an uneven distribution within a cluster, there will be multiple points with high $\rho$'s and $\delta$'s within this cluster besides one real center. This is also the inspiration for us to determine the set of candidate centers. Next, an approach is represented to determine the set of candidate centers.

On a decision graph, most of the points are concentrated in a narrow region, as shown in Fig. 1(c). At the same time, there are only a small set of the points with high $\rho$'s and $\delta$'s above the narrow region. In other words, almost all of the points of the dataset fall into a one dimensional low rank space, and the candidate centers can be recognized as the outliers. This situation also occurs in other datasets. The points within this set are apart from that narrow region and they are suitable candidate centers. The candidate centers are selected by two thresholds that can be determined based on the decision graph. Then, we will represent the reason why our algorithm can deal with the problem of uneven distribution within local clusters.

The number of candidate centers is always greater than or equal to the number of real centers. Note that each of these candidate centers is not directly assigned to the same cluster as its nearest neighbor with higher density. In contrast, all of these candidate points are used to construct the subclusters for later merging processes. This avoids assigning the candidate centers to the wrong clusters. Consequently, the subclusters centered on these candidate centers will also not be merged into the wrong clusters, which is also the core reason why our algorithm can deal with the problem of uneven distribution within local clusters. Next, a simple example are presented to describe the process.

As shown in Fig. 1(c), all colored nonblack points that are located above the narrow region are selected as candidate centers, and the remaining points that are located in a narrow region are noncenters. After the candidate centers are determined, each remaining point is assigned to the same subcluster as its nearest neighbor with higher density. Here, all the candidate centers are not assigned to their nearest neighbor with higher density, and the subclusters are also not merged into any other clusters. Then no misclassification occurs.

Additionally, compared with the other hierarchy based methods, our method only needs to construct subclusters through candidate centers that are easily determined. It is convenient for our method to construct the subclusters without complex computation. Note that $p$ is set to be a fixed number of 1.5% to construct the decision graph, and the experiment results indicate that strategy of constructing the subclusters will only lead to very few misclassifications and it will be discussed in Section IV.C.

### C. MERGING THE SUBCLUSTERS

This section presents a method to merge the subclusters constructed by the previous steps. Similar to the Chameleon algorithm, the connectivity and similarity between two adjacent subclusters are also taken into account during this procedure. The algorithm has its basis in the assumption that if two adjacent subclusters should have been in the same cluster, then the intersection region between them should be relatively dense and their distributions or some statistical information (such as density distribution) should be similar. Based on this assumption, we present a new strategy that can measure the connectivity and dissimilarity between subclusters simultaneously. As some methods [19], [20] are available to determine the number of clusters, we assume that the number of clusters is determined beforehand. Some definitions will be represented as follows:

1) Intersection set between two subclusters: the intersection set $C_{ij}$ between subcluster $C_i$ and subcluster $C_j$ can be calculated as:

$$\begin{aligned} &\forall x \in C_i, \quad \forall y \in C_j, \ if \ d_{xy} \leq \alpha d_c \\ &\text{then:} \\ &x \in C_{ij}, \quad y \in C_{ij} \end{aligned} \tag{14}$$

where $d_c$ is the cut-off distance that is calculated by the average percentage of neighbours $p$. $p$ is set to be a fixed number of 1.5% empirically. $\alpha$ is an adjustable coefficient. $\alpha d_c$ is used to determine the size of the intersection area of two subclusters.

2) Local set of a subcluster to its adjacent subcluster: For a subcluster $C_i$ with its adjacent subcluster $C_j$, the related local set $C_{i-j}$ can be calculated as:

$$\begin{aligned} &\forall x \in C_i, \quad \forall y \in C_j, \ if \ d_{xy} \leq d_{m_i m_j} \\ &\text{then:} \\ &x \in C_{i-j}, \quad y \in C_{j-i} \end{aligned} \tag{15}$$

where $m_i$ is the center of $C_i$ and $m_j$ is the center of $C_j$, and the related local set $C_{j-i}$ is also calculated at the same time. It can be seen that only the part of a subcluster that has points on the side of the center of this subcluster that is close to the center of the another subcluster will be considered. In other words, only those points between two central points are considered, which leads to the improvement of the robustness to a certain extent. The calculation process is symmetrical for another subcluster. This is different from the other hierarchy based algorithms where all of the points within a subcluster are taken into account when calculating the connectivity or the similarity [5].

Next, the average density of set $C_{ij}$, $C_{i-j}$ and $C_{j-i}$ will be calculated as follows:

$$\bar{\rho_{ij}} = \frac{\sum_{k \in C_{ij}} \rho_k}{\|C_{ij}\|} \tag{16}$$

$$\bar{\rho_{i-j}} = \frac{\sum_{k \in C_{i-j}} \rho_k}{\left\| C_{i-j} \right\|} \tag{17}$$

$$\bar{\rho_{j-i}} = \frac{\sum_{k \in C_{j-i}} \rho_k}{\left\| C_{j-i} \right\|} \tag{18}$$

where $\left\| C_{ij} \right\|$, $\left\| C_{i-j} \right\|$ and $\left\| C_{j-i} \right\|$ represent the number of the set $C_{ij}$, $C_{i-j}$ and $C_{j-i}$ respectively. And they are used to meansure the connectivity or the dissimilarity of two subclusters by the following equation:

$$M_{ij} = \left| \bar{\rho_{ij}} - \bar{\rho_{i-j}} \right| + \left| \bar{\rho_{ij}} - \bar{\rho_{j-i}} \right| \tag{19}$$

Note that if $C_{ij}$ is $\phi$, this indicates that the subcluster $i$ is not connected to the subcluster $j$. Then, the value of $M_{ij}$ does not exist and is not zero. We suppose that its value is $+\infty$ and it is called $M_\phi$.

Relatively small value of $M_{ij}$ indicates the following: 1) the region between the subcluster $C_i$ and the subcluster $C_j$ is relatively dense; 2) the distribution of these two subclusters is more similar. Then, the subcluster $C_i$ and $C_j$ should be merged; otherwise they should not be merged. During the merging process, all of $M_{ij}$ are sorted from low to high by its value, and the pair of subclusters are merged into a same cluster in the order of $M_{ij}$ repeatedly. There is also a stop criteria that the number of the remaining clusters decreases to the correct number of the clusters.

Eq. (19) is used to measure the dissimilarity between the two adjacent subclusters and determine whether they are connected. Our merging method reduces the hassle caused by multiple parameter settings, and it is very efficient as connectivity and dissimilarity between two adjacent subclusters are also taken into account.

The HFCS algorithm is proposed to solve the problem of uneven distribution within local clusters by employing a hierarchy strategy, and the general procedures of the presented HCFS clustering algorithm are shown as follows:

1) Calculate the Gauss distance among the points of the entire dataset to constitute the distance matrix $D$ by Eq. (1);
2) Calculate the local density $\rho$ and the distance $\delta$ respectively by Eq. (7) and Eqs. (8) and (9) for each point;
3) Construct $\delta - \rho$ decision graph, then determine the set of the candidate centers on the decision graph and construct the subclusters based on these candidate centers;
4) Calculate the dissimilarity between subclusters by Eq. (19). All $M_{ij}$ are sorted from low to high by its value. We assume that each subcluster is a separate cluster, then the subclusters $i$ and $j$ are merged into a same cluster in the order of $M_{ij}$ repeatedly. The stop condition is that the number of the remain clusters decreases to the correct number of the clusters. Note that if the subclusters $i$ and $j$ which meet the merging conditions are located in two different clusters, these two clusters should be merged.

An example is used to illustrate how the presented HCFS clustering algorithm works (Figs. 1(b), (c), (d)). Fig. 1(b) shows that the HCFS algorithm performs well on the Flame dataset. The main process is as follows: First, the colored non-black points are selected as the candidate centers, as shown in Fig. 1(c). Then, each remaining point is assigned to its nearest neighbor with higher density to construct the subclusters in Fig. 1(d). Finally, as $M_{DE} < M_{CF} < M_{BF} < M_{BE} < M_{AE} < M_{FA} < M_{AB} < M_\phi = +\infty$ and the correct number of the cluster is 2, the subclusters are merged in the following order:

Start $\longrightarrow \{\{A\}, \{B\}, \{C\}, \{D\}, \{E\}, \{F\}\}$
$\xrightarrow{Ope\,(M_{DE})} \{\{D, E\}, \{A\}, \{B\}, \{C\}, \{F\}\}$
$\xrightarrow{Ope\,(M_{CF})} \{\{D, E\}, \{C, F\}, \{A\}, \{B\}\}$
$\xrightarrow{Ope\,(M_{BF})} \{\{D, E\}, \{C, F, B\}, \{A\}\}$
$\xrightarrow{Ope\,(M_{BE})} \{\{D, E, C, F, B\}, \{A\}\} \longrightarrow$ Stop.

$Ope\,(M_{DE})$ represents the operation of merging adjacent subclusters centered on D and E. The final clustering results are shown in Fig. 1(b).

**TABLE 1. Datasets.**

| data set | Observations | Classes | Dimensions |
|---|---|---|---|
| Aggregation | 788 | 7 | 2 |
| D31 | 3100 | 31 | 2 |
| Flame | 240 | 2 | 2 |
| Jain | 373 | 2 | 2 |
| Pathbased | 300 | 3 | 2 |
| R15 | 600 | 15 | 2 |
| iris | 150 | 3 | 4 |
| Thyroid | 215 | 3 | 5 |
| seeds | 210 | 3 | 7 |
| chameleon (t4.8k) | 8000 | 6 | 2 |
| chameleon (t7.10k) | 8000 | 8 | 2 |

## IV. EXPERIMENTAL RESULTS

In this section, multiple datasets are used to evaluate our proposed algorithm as comprehensively as possible. In detail, Aggregation, D31, Jain, Pathbased, R15, chameleon t4.8k, and chameleon t7.10k are artificial datasets; Flame, iris, Thyroid, and seeds are real datasets. Table 1 provides an overall description of these datasets. Among them, two unlabeled chameleon datasets [5] and pathbased dataset contain unevenly distributed local clusters obviously. All datasets can be conventionally obtained from [22]. The experiment in this paper consists of two parts. First, the proposed HCFS clustering algorithm are compared with two other density peak based clustering algorithms, i.e, CFS and NCFS, on three typical datasets containing unevenly distributed local clusters. Second, the algorithms HCFS, NCFS, CFS, ISDBSCAN and ISBDBSCAN that need to adjust only one parameter are evaluated on all labeled datasets.

### A. DENSITY PEAK BASED CLUSTERING ALGORITHMS

To evaluate the performance of our method to deal with unevenly distributed local clusters, the pathbased dataset and two chameleon datasets t4.8k and t7.10k are used. These datasets contain unevenly distributed local clusters.

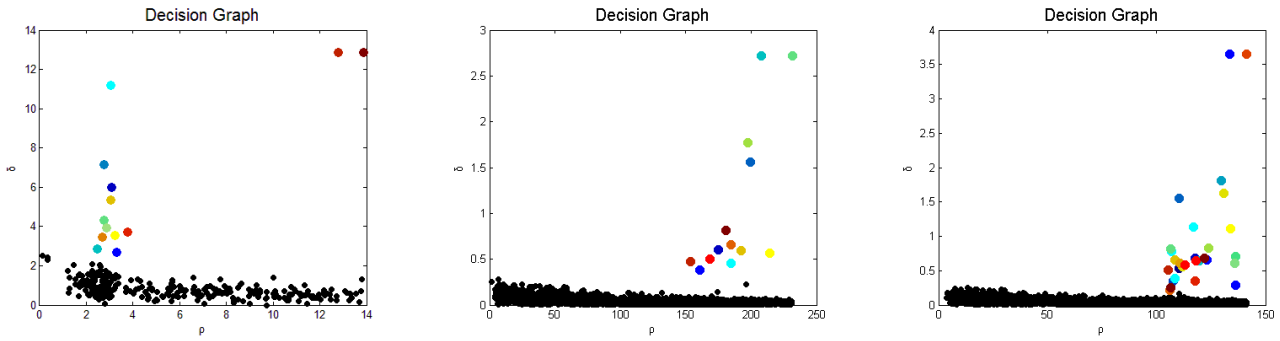



**FIGURE 2.** (a) Decision graph obtained on pathbased dataset by employing gauss kernel and $p$ is set to 2%. 6 candidate centers are coloured non-black while the rest points are coloured black. (b) Decision graph obtained on chameleon dataset t4.8k by employing gauss kernel and $p$ is set to 2%. 13 candidate centers are coloured non-black while the rest points are coloured black. (c) Decision graph obtained on chameleon dataset t7.10k by employing gauss kernel and $p$ is set to 1%. 29 candidate centers are coloured non-black while the rest points are coloured black.

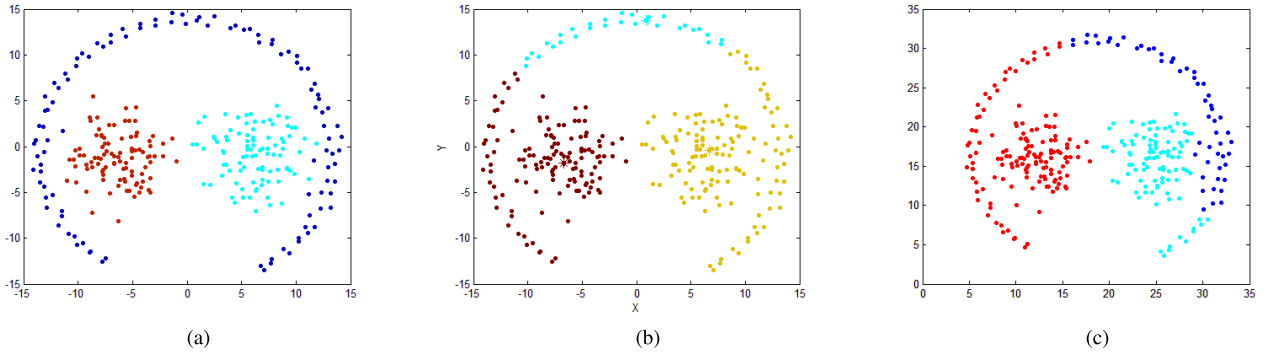



**FIGURE 3.** (a) The clustering result of pathbased dataset using the presented HCFS clustering algorithm by employing gauss kernel and $p$ is set to 2.0%. (b) The clustering result of pathbased dataset using CFS clustering algorithm by employing gauss kernel and $p$ is set to 2.0%. (c) The clustering result of pathbased dataset using NCFS clustering algorithm.

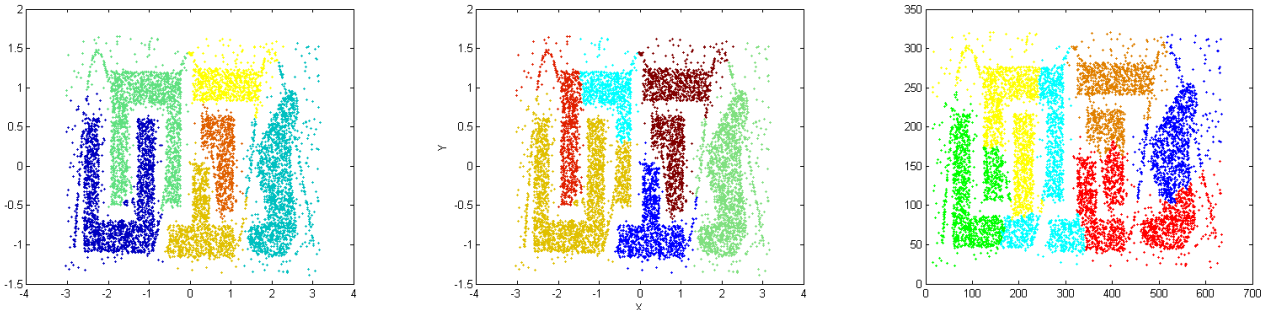
We compare our method with two density peak based algorithms CFS and NCFS. Each algorithm has its own appropriate parameter search space. For the CFS algorithm, the average percentage of neighbours is selected from the range [0%, 4%]. For the HCFS algorithm, the coefficient $\alpha$ is selected from the range [0, 4], and parameter k is selected from the range [1, 150] for NCFS.

As shown in Figs. 2(a), (b), and (c), there are 13, 13, 29 nonblack points located above the black narrow region corresponding to datasets pathbased, chameleon t4.8k and t7.10k, respectively. Obviously, the nonblack points have higher $\rho$'s and $\delta$'s. In addition, all of these points are suitable candidate centers. However, the correct numbers of these three datasets are 3, 6, 9 respectively. These results indicate that there are several unevenly distributed local clusters in these three datasets. This is because the number of the points with higher $\rho$'s and larger $\delta$'s is more than the correct number of the centers within these datasets.

When employing the original CFS algorithm on these datasets, as the nearest neighbors with higher densities of the partial points within the set of the candidate centers are in the wrong clusters, these partial candidate points are assigned to the wrong clusters. The subclusters in which these partial candidate points are located are also immediately merged into those wrong clusters. As a result, misclassification occurs, as shown in Fig. 3(b), 4(b), and 5(b). By employing the reverse nearest neighbor, although NCFS chooses more appropriate central points than CFS, it still achieve unsatisfactory results, as shown in Fig. 3(c), 4(c), and 5(c). The reason for this phenomenon is that NCFS does not have the ability to handle unevenly distributed local clusters. When employing the proposed HCFS algorithm on these datasets, all the candidate centers are not directly assigned to their nearest neighbor with higher density. Then, the adjacent subclusters are merged based on the dissimilarity estimation between them. As a result, the proposed HCFS algorithm is able to accurately distinguish between local clusters with uneven distribution, as shown in Fig. 3(a), 4(a), and 5(a).

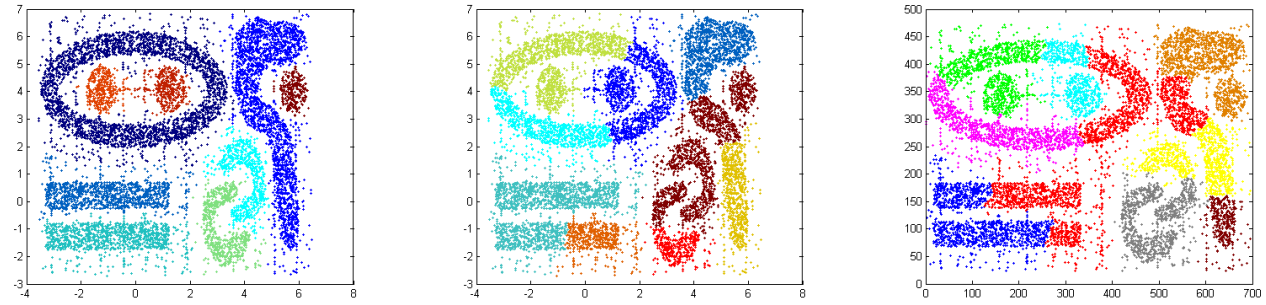
### *B. PARAMETER REDUCTION ALGORITHMS*

In this section, the algorithms HCFS, NCFS, CFS, ISDBSCAN and ISBDBSCAN that only need to adjust one parameters are compared on the datasets in Table 1, except for two unlabeled datasets chameleon t4.8k and t7.10k. Three density peak based algorithms, i.e., CFS, NCFS and HCFS, select the same parameters, as in Section IV.A. The ISDBSCAN and



(a) (b) (c)

**FIGURE 4.** (a) The clustering result of chameleon dataset t4.8k using HCFS clustering algorithm by employing gauss kernel and $p$ is set to 2%. (b) The clustering result of chameleon dataset t4.8k using CFS clustering algorithm by employing gauss kernel and $p$ is set to 2%. (c) The clustering result of chameleon dataset t4.8k using NCFS clustering algorithm.



(a) (b) (c)

**FIGURE 5.** (a) The clustering result of chameleon dataset t7.10k using HCFS clustering algorithm by employing gauss kernel and $p$ is set to 1%. (b) The clustering result of chameleon dataset t7.10k using CFS clustering algorithm by employing gauss kernel and $p$ is set to 1%. (c) The clustering result of chameleon dataset t7.10k using NCFS clustering algorithm.

ISBDBSCAN algorithms employ the reverse nearest neighbors to select the parameter $k$ from the range [1, 150]. To evaluate the performance of these algorithms, the Adjusted Rand Index (ARI) [23] and Normalized Mutual Information (NMI) [24] are used, which measure the agreement between the clustering results produced by an algorithm and the ground truth. We assumed that there were k clusters produced by an algorithm and $m$ real classes. The NMI is calculated as follows:

$$NMI = \frac{\sum_{c=1}^{k}\sum_{p=1}^{m} n_c^p log((n \cdot n_c^p)/(n_c \cdot n_p))}{\sqrt{(\sum_{c=1}^{k} n_c log(n_c/n))(\sum_{p=1}^{m} n_p log(n_p/n))}} \quad (20)$$

where $n$ denotes the total number of points, $n_c$ denotes the number of points in the $c$th cluster in experiment results, $n_p$ denotes the number of points in the $p$th class in ground truth, and $n_c^p$ denotes the number of common points in class $p$ and cluster $c$. ARI is calculated as follows:

$$ARI = \frac{\sum_{i=1}^{k}\sum_{j=1}^{m}\binom{n_{ij}}{2} - [\sum_{i=1}^{k}\binom{a_i}{2}\sum_{j=1}^{m}\binom{b_j}{2}]/(\binom{n}{2})}{\frac{1}{2}[\sum_{i=1}^{k}\binom{a_i}{2} + \sum_{j=1}^{m}\binom{b_j}{2}] - [\sum_{i=1}^{k}\binom{a_i}{2}\sum_{j=1}^{m}\binom{b_j}{2}]/(\binom{n}{2})} \quad (21)$$

**TABLE 2.** ARI performance.

| dataset | HCFS | NCFS | CFS | IS | ISB |
|---|---|---|---|---|---|
| Aggregation | 0.998 | 1.000 | 0.998 | 0.887 | 0.914 |
| D31 | 0.935 | 0.939 | 0.935 | 0.738 | 0.739 |
| Flame | 1.000 | 0.918 | 1.000 | 0.707 | 0.215 |
| Jain | 1.000 | 1.000 | 0.515 | 0.876 | 1.000 |
| Pathbased | 0.970 | 0.551 | 0.453 | 0.735 | 0.789 |
| R15 | 0.993 | 0.993 | 0.993 | 0.898 | 0.993 |
| iris | 0.886 | 0.868 | 0.759 | 0.735 | 0.789 |
| thyroid | 0.877 | 0.115 | 0.795 | 0.790 | 0.821 |
| seeds | 0.753 | 0.418 | 0.624 | 0.563 | 0.527 |

where $a_i$ denotes the sum of the common points in the $i$th cluster in experimental results and all classes in ground truth, $b_j$ denotes the sum of the common points in the $j$th classes in ground truth and all clusters in experimental results, and $n_{ij}$ denotes the number of common points in class $j$ and cluster $i$.

Table 2 shows the ARI performance for the HCFS, NCFS, CFS, ISDBSCAN and ISBDBSCAN clustering algorithms on the different datasets. The first six datasets in Table 2 are 2D datasets, and the last three datasets are non-2D datasets. From these results, the proposed HCFS algorithm is shown to not only performs well on the 2D datasets but also performs better than the other algorithms on the non-2D datasets.

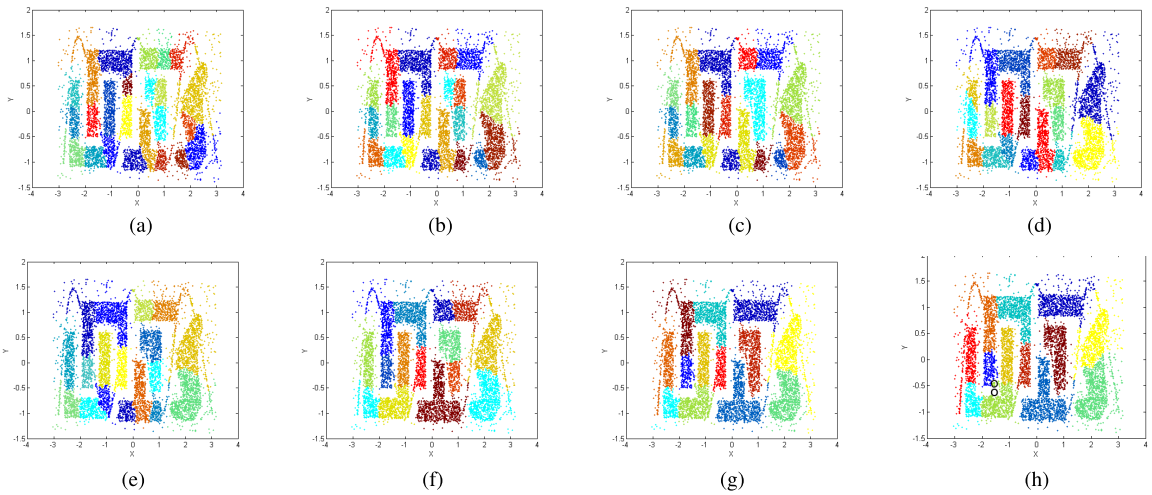


**FIGURE 6.** The clustering results in constructing subclusters by setting $p$ to 1.00(a), $1.05 \leqslant \alpha \leqslant 1.10$(b), $1.15 \leqslant \alpha \leqslant 1.20$(c), $1.25 \leqslant \alpha \leqslant 1.40$(d), 1.45(e), $1.50 \leqslant \alpha \leqslant 1.70$(f), $1.75 \leqslant \alpha \leqslant 1.95$(g), 2.00(h) on the chameleon dataset t4.8k.

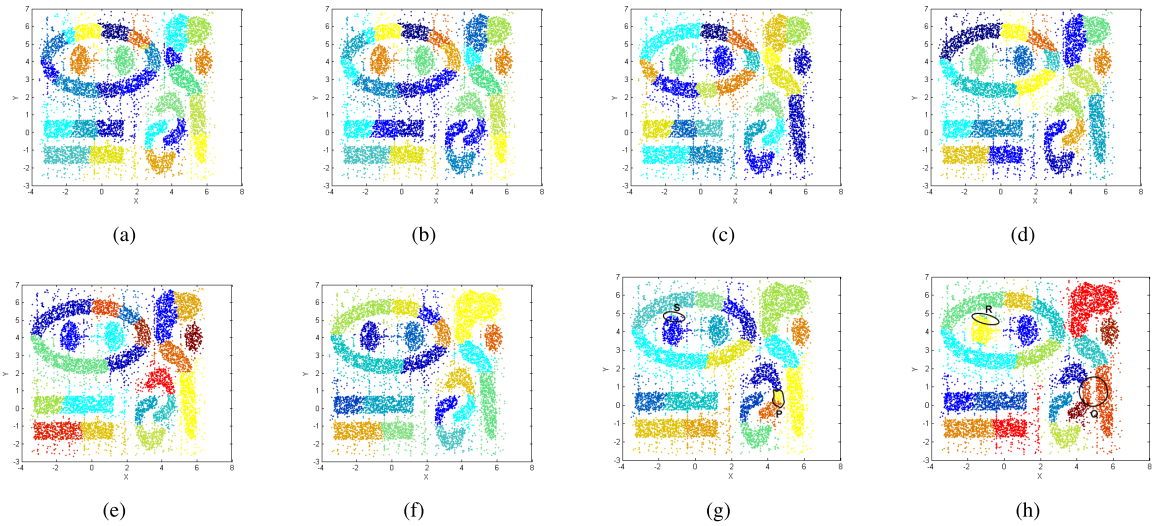


**FIGURE 7.** The clustering results in constructing subclusters by setting $p$ to 1.00(a), $1.05 \leqslant \alpha \leqslant 1.15$(b), $1.20 \leqslant \alpha \leqslant 1.25$(c), $1.30 \leqslant \alpha \leqslant 1.55$(d), $1.60 \leqslant \alpha \leqslant 1.65$(e), $1.70 \leqslant \alpha \leqslant 1.85$(f), $1.90 \leqslant \alpha \leqslant 1.95$(g), 2.00(h) on the chameleon dataset t7.10k.

**TABLE 3.** NMI performance.

| dataset | HCFS | NCFS | CFS | IS | ISB |
|---|---|---|---|---|---|
| Aggregation | 0.996 | 1.000 | 0.996 | 0.842 | 0.954 |
| D31 | 0.957 | 0.960 | 0.957 | 0.855 | 0.873 |
| Flame | 1.000 | 0.935 | 1.000 | 0.599 | 0.362 |
| Jain | 1.000 | 1.000 | 0.505 | 0.729 | 1.000 |
| Pathbased | 0.960 | 0.622 | 0.539 | 0.674 | 0.772 |
| R15 | 0.994 | 0.994 | 0.994 | 0.927 | 0.994 |
| iris | 0.871 | 0.857 | 0.806 | 0.611 | 0.568 |
| thyroid | 0.807 | 0.576 | 0.738 | 0.628 | 0.543 |
| seeds | 0.741 | 0.711 | 0.650 | 0.554 | 0.582 |

As the clusters of high dimensional datasets may exist in subspaces, then our algorithm does not perform as well on non-2D datasets as it does on 2D datasets. Overall, the three density peak based algorithms are shown to perform better than the ISDBSCAN and ISBDBSCAN algorithms. This is most likely due to the use of the fixed predetermined threshold for determining the core observation. We wondered if adjusting the predetermined threshold would be a solution for the problem.

Both CFS and NCFS algorithms attempt to use more information to reduce the statistical errors. In addition, the CFS algorithm attempts to reduce the statistical errors by increasing the value of the average percentage of neighbours $p$. In the same way, the NCFS algorithm attempts to reduce statistical errors by selecting the top four farthest neighbors rather than one neighbor. Unfortunately, regardless of how the parameters are adjusted, the CFS and NCFS algorithms cannot perform well on more complex datasets, such as the pathbased dataset. However, we can detect the more detail

**TABLE 4.** Number of subclusters and number of misclassification points in constructing subclusters.

| | Aggregation | | D31 | | Flame | | Jain | | Pathbased | | R15 | | t4.8k | | t7.10k | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| $p$ % | N1 | N2 | N1 | N2 | N1 | N2 | N1 | N2 | N1 | N2 | N1 | N2 | N1 | N2 | N1 | N2 |
| 1.00 | 17 | 1 | 31 | 34 | 7 | 0 | 25 | 0 | 21 | 0 | 15 | 5 | 24 | - | 29 | - |
| 1.05 | 17 | 1 | 31 | 37 | 7 | 0 | 25 | 0 | 21 | 0 | 15 | 5 | 21 | - | 28 | - |
| 1.10 | 17 | 1 | 31 | 33 | 7 | 0 | 25 | 0 | 21 | 0 | 15 | 2 | 21 | - | 28 | - |
| 1.15 | 17 | 1 | 31 | 38 | 7 | 0 | 25 | 0 | 21 | 0 | 15 | 2 | 19 | - | 28 | - |
| 1.20 | 17 | 1 | 31 | 38 | 7 | 0 | 25 | 0 | 21 | 0 | 15 | 2 | 19 | - | 25 | - |
| 1.25 | 17 | 1 | 31 | 38 | 7 | 0 | 25 | 0 | 19 | 3 | 15 | 2 | 19 | - | 25 | - |
| 1.30 | 16 | 1 | 31 | 38 | 7 | 0 | 23 | 0 | 19 | 3 | 15 | 2 | 19 | - | 22 | - |
| 1.35 | 14 | 1 | 31 | 34 | 7 | 0 | 22 | 0 | 19 | 3 | 15 | 2 | 19 | - | 22 | - |
| 1.40 | 13 | 1 | 31 | 34 | 7 | 0 | 22 | 0 | 17 | 3 | 15 | 2 | 19 | - | 22 | - |
| 1.45 | 13 | 1 | 31 | 34 | 7 | 0 | 22 | 0 | 17 | 3 | 15 | 2 | 18 | - | 22 | - |
| 1.50 | 13 | 1 | 31 | 34 | 7 | 0 | 21 | 0 | 17 | 3 | 15 | 2 | 15 | - | 22 | - |
| 1.55 | 13 | 1 | 31 | 34 | 7 | 0 | 21 | 0 | 17 | 3 | 15 | 2 | 15 | - | 22 | - |
| 1.60 | 13 | 1 | 31 | 33 | 7 | 0 | 21 | 0 | 17 | 3 | 15 | 2 | 15 | - | 21 | - |
| 1.65 | 11 | 1 | 31 | 33 | 7 | 0 | 20 | 0 | 17 | 3 | 15 | 2 | 15 | - | 21 | - |
| 1.70 | 11 | 1 | 31 | 33 | 7 | 0 | 19 | 0 | 17 | 3 | 15 | 2 | 15 | - | 20 | - |
| 1.75 | 11 | 1 | 31 | 37 | 6 | 0 | 19 | 0 | 15 | 3 | 15 | 2 | 13 | - | 20 | - |
| 1.80 | 10 | 1 | 31 | 37 | 6 | 0 | 19 | 0 | 15 | 3 | 15 | 2 | 13 | - | 20 | - |
| 1.85 | 10 | 1 | 31 | 37 | 6 | 0 | 19 | 0 | 15 | 3 | 15 | 2 | 13 | - | 20 | - |
| 1.90 | 10 | 1 | 31 | 37 | 6 | 0 | 15 | 0 | 15 | 3 | 15 | 2 | 13 | - | 20 | - |
| 1.95 | 10 | 1 | 31 | 37 | 6 | 0 | 15 | 0 | 14 | 3 | 15 | 2 | 13 | - | 20 | - |
| 2.00 | 10 | 1 | 31 | 37 | 6 | 0 | 15 | 0 | 13 | 3 | 15 | 2 | 13 | - | 19 | - |

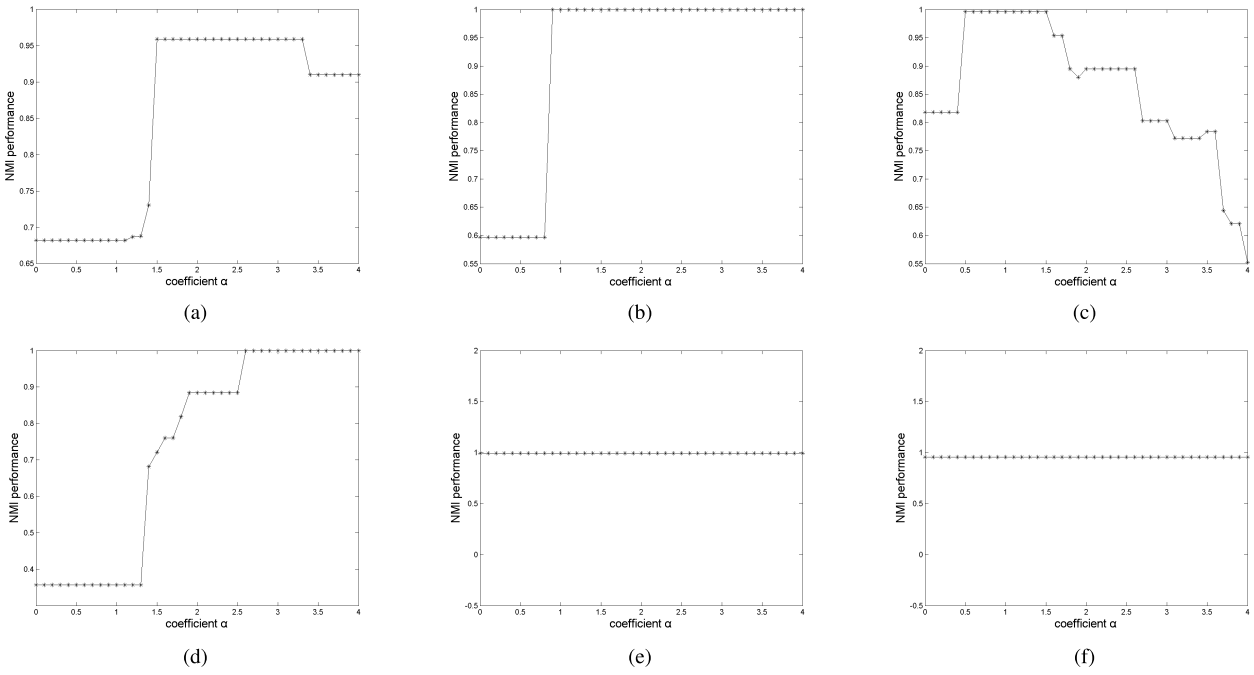



**FIGURE 8.** The coefficient $\alpha$ versus NMI performance for HCFS clusterings produced over the range $0 \leqslant \alpha \leqslant 4$ on the Pathbased(a), Flame(b), Aggregation(c), Jain(d), r15(e), d31(f) datasets.

distribution information about the datasets by adjusting the parameter to a smaller value. In addition, the HCFS algorithm then uses this information to construct the subclusters and to merge the adjacent subclusters that are similar and connected. The distribution of the local clusters are taken into account by the HCFS algorithm. Therefore, the HCFS algorithm performs better.

There is an obvious difference in the densities between the local clusters, as the average density of one cluster is 2.181 and that of the other cluster is 8.670 in the Jain dataset. The proposed HCFS algorithm can also perform very well on the Jain dataset, while the original CFS algorithm cannot, which indicates that our method is capable of classifying the datasets containing different clusters with different densities.

Table 3 shows the NMI performance results on the same set of datasets and clustering algorithms, and it indicates that the NMI performance results are similar to the ARI performance results.

### C. PARAMETER ANALYSIS

The proposed HCFS algorithm employs one fixed parameter (the average percentage of neighbours $p$) and one adjustable

parameter (the coefficient $\alpha$). Additionally, $d_c$, which is calculated though $p$, is used to compute the density of the points and to construct the decision graph. $\alpha d_c$ is used to determine the size of the intersection area of two subclusters. First, we will represent the reasons why the average percentage of neighbours $p$ can be set to a fixed decimal.

As shown in Table 4, N1 and N2 represent the number of subclusters and the number of misclassification points in the process of constructing subclusters, respectively. The results of column N2 on these labeled datasets indicate that at most only about 1% of the points are misclassified when the average percentage of neighbours $p$ is located at the range of [1%, 2%]. In addition, Figs. 6 and 7 show the effect of $p$ on performance to construct subclusters for unlabelled datasets. The results indicate that high quality subclusters are obtained on chameleon dataset t4.8k when the average percentage of neighbours $p$ is located at the range of [1%, 2%], and on chameleon dataset t7.10k when the average percentage of neighbours $p$ is located at the range of [1%, 1.85%]. This indicates that the average percentage of neighbours $p$ in the range of [1%, 1.85%] are reasonable. Here are some minor misclassification, such as region O in Fig. 6(h), regions P and S in Fig. 7(g) and regions R and Q in Fig. 7(h). This indicates that the misclassification rate in constructing subclusters will increase with the increase of the parameter $p$. The results of column N1 indicate that the number of subclusters decreases with the increase of parameter $p$. As a small number of subclasses can reduce the computational cost, then setting p to 1.85% will be a good compromise choice. However, in order to make our algorithm work with more complex datasets with fewer errors in constructing subclusters, we can set the parameter to a smaller fixed number of 1.5%.

Fig. 8 shows the effect of coefficient $\alpha$ on the performance of the algorithm. Meanwhile, The HCFS clustering algorithm has a probability of more than 25% to achieve maximum NMI performance when the coefficient $\alpha$ is located at the range of [0, 4], which indicates that it is convenient for us to find appropriate parameters for a dataset within this range. That also indicates that there is a great reduction in the dependence of our algorithm's performance on coefficient $\alpha$. In addition, if all candidate centers within a dataset are also real centers, then the NMI performance will not be affected by the coefficient $\alpha$, as shown in Figs. 8(e), (f).

## V. CONCLUSION

This paper describes the procedure of misclassification caused by uneven distributions of local clusters. The reason for this procedure was also analyzed in detail. Then, a density peak based algorithm employing the hierarchy strategy (HCFS) was proposed, which is capable of recognizing unevenly distributed local clusters. The HCFS algorithm consists mainly of two stages. First, the points with high $\rho$'s and $\delta$'s are selected as candidate centers. In this stage, each remaining candidate center is used to construct the subclusters instead of being assigned to the same cluster as their nearest neighbor with higher density. This not only avoids the misclassification caused by uneven distribution of local clusters, but also indirectly solves the problem of determining the real centers from the set of candidate centers. Second, the subclusters are merged by a new dissimilarity estimation strategy proposed in this paper. This new dissimilarity estimation method can measure the dissimilarity between two adjacent subclusters and determine whether two subclusters are connected simultaneously using one equation, which means that our algorithm only needs to adjust one parameter. Through these works, the problem of the misclassification due to uneven distribution within local clusters has been alleviated.

As the clusters of high dimensional datasets may exist in subspaces, our method does not perform as well on non-2D datasets as it does on 2D datasets. What's more, more and more applications produce large amounts of high dimensional data without labels. And there will be some unevenly distributed local clusters in these datasets. This inspire us to combine subspace clustering algorithms [25], [26] to extend our method to deal with these high dimensional datasets.

## ACKNOWLEDGMENT

The authors would like to thank the two anonymous reviewers for their valuable and helpful comments on improving the manuscript.

## REFERENCES

[1] H. Gao, C. Ding, C. Song, and J. Mei, "Automated inspection of E-shaped magnetic core elements using K-tSL-center clustering and active shape models," *IEEE Trans. Ind. Informat.*, vol. 9, no. 3, pp. 1782–1789, Aug. 2013.

[2] D. Wijayasekara, O. Linda, M. Manic, and C. Rieger, "Mining building energy management system data using fuzzy anomaly detection and linguistic descriptions," *IEEE Trans. Ind. Informat.*, vol. 10, no. 3, pp. 1829–1840, Aug. 2014.

[3] J. Macqueen, "Some methods for classification and analysis of multivariate observations," in *Berkeley Symposium on Mathematical Statistics and Probability*, 1965, pp. 281–297.

[4] M. Ester, H.-P. Kriegel, J. Sander, and X. Xu, "A density-based algorithm for discovering clusters a density-based algorithm for discovering clusters in large spatial databases with noise," in *Proc. 2nd Int. Conf. Knowl. Discovery Data Mining*, Aug. 1996, pp. 226–231.

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

[6] A. Y. Ng, M. I. Jordan, and Y. Weiss, "On spectral clustering: Analysis and an algorithm," in *Proc. Adv. Neural Inf. Process. Syst.*, 2001, pp. 849–856.

[7] C. Deng, "Compressed spectral regression for efficient nonlinear dimensionality reduction," in *Proc. Int. Conf. Artif. Intell.*, 2015, pp. 3359–3365.

[8] S. Vadapalli, S. R. Valluri, and K. Karlapalem, "A simple yet effective data clustering algorithm," in *Proc. 6th Int. Conf. Data Mining (ICDM)*, Dec. 2006, pp. 1108–1112.

[9] 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.

[10] Y. Lv, T. Ma, M. Tang, J. Cao, Y. Tian, A. Al-Dhelaan, and M. Al-Rodhaan,, "An efficient and scalable density-based clustering algorithm for datasets with complex structures," *Neurocomputing*, vol. 171, pp. 9–22, Jan. 2016.

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

[12] M. Hahsler and M. Bolaños, "Clustering data streams based on shared density between micro-clusters," *IEEE Trans. Knowl. Data Eng.*, vol. 28, no. 6, pp. 1449–1461, Jun. 2016.

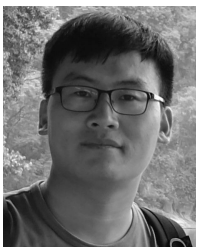
[13] A. Prasanth and S. Valsala, "Semantic chameleon clustering analysis algorithm with recommendation rules for efficient web usage mining," in *Proc. IEEE-GCC 9th Conf. Exhib. (GCCCE)*, May 2017, pp. 1–9.
[14] U. Gupta and N. Patil, "Recommender system based on hierarchical clustering algorithm chameleon," in *Proc. IEEE Int. Adv. Comput. Conf. (IACC)*, Jun. 2015, pp. 1006–1010.
[15] E. Schubert, J. Sander, M. Ester, H. P. Kriegel, and X. Xu, "DBSCAN revisited, revisited: Why and how you should (still) use DBSCAN," *ACM Trans. Database Syst.*, vol. 42, no. 3, p. 19, Aug. 2017.
[16] J. Hou, C. Sha, L. Chi, Q. Xia, and N.-M. Qi, "Merging dominant sets and DBSCAN for robust clustering and image segmentation," in *Proc. IEEE Int. Conf. Image Process.*, Oct. 2014, pp. 4422–4426.
[17] L. Fu and E. Medico, "FLAME, a novel fuzzy clustering method for the analysis of DNA microarray data," *BMC Bioinf.*, vol. 8, no. 1, p. 3, 2007.
[18] J. Hou and M. Pelillo, "A new density kernel in density peak based clustering," in *Proc. 23rd Int. Conf. Pattern Recognit.*, Dec. 2017, pp. 468–473.
[19] C. Fraley and A. E. Raftery, "How many clusters? which clustering method? answers via model-based cluster analysis," *Comput. J.*, vol. 41, no. 8, pp. 578–588, 1998.
[20] G. Evanno, S. Regnaut, and J. Goudet, "Detecting the number of clusters of individuals using the software structure: A simulation study," *Mol. Ecol.*, vol. 14, no. 8, pp. 2611–2620, 2005.
[21] H. Chang and D.-Y. Yeung, "Robust path-based spectral clustering," *Pattern Recognit.*, vol. 41, no. 1, pp. 191–203, 2008.
[22] M. Lichman, "Uci machine learning repository," Tech. Rep. 2013.
[23] L. Hubert and P. Arabie, "Comparing partitions," *J. Classification*, vol. 2, no. 1, pp. 193–218, 1985.
[24] A. Strehl and J. Ghosh, "Cluster ensembles—A knowledge reuse framework for combining multiple partitions," *J. Mach. Learn. Res.*, vol. 3, no. 3, pp. 583–617, 2003.
[25] I. Khan, J. Z. Huang, N. T. Tung, and G. Williams, "Ensemble clustering of high dimensional data with fastmap projection," in *Trends and Applications in Knowledge Discovery and Data Mining*, W.-C. Peng, H. Wang, J. Bailey, V. S. Tseng, T. B. Ho, Z.-H. Zhou, and A. L. Chen, Eds., Cham, Switzerland: Springer, 2014, pp. 483–493.
[26] I. Khan and J. Z. Huang, "Fastmap in dimensionality reduction: Ensemble clustering of high dimensional data," *Int. J. Data Sci.*, vol. 2, no. 1, pp. 15–28, 2017.

**LINLIN ZHUO** is currently pursuing the Ph.D. degree with Hunan University, China. His research interests include clustering algorithm, object detection, and multi-GPU computing.

**KENLI LI** received the Ph.D. degree in computer science from the Huazhong University of Science and Technology, China, in 2003. He was a Visiting Scholar with the University of Illinois at Urbana–Champaign, from 2004 to 2005. He is currently a Full Professor of computer science and technology with Hunan University and the Deputy Director of the National Supercomputing Center, Changsha. His current research interests include parallel computing, high-performance computing, and grid and cloud computing. He has published more than 130 research papers in international conferences and journals, such as the IEEE-TC, the IEEE-TPDS, the IEEE-TSP, JPDC, ICPP, and CCGrid. He is an Outstanding Member of CCF. He serves on the Editorial Board of the IEEE TRANSACTIONS ON COMPUTERS.

**BO LIAO** received the Ph.D. degree in computational mathematics from the Dalian University of Technology, Dalian, China, in 2004. From 2004 to 2006, he was a Postdoctoral Fellow with the University of Chinese Academy of Sciences, Beijing, China. He is currently a Full Professor of information engineering with Hunan University, Changsha, China. He has authored more than 100 papers in international conferences and journals. His current research interests include bioinformatics, image processing, and big data processing.

**HAO LI** is currently pursuing the Ph.D. degree with Hunan University, China. He has published six journal and conference papers in the IEEE-TPDS, InforSci, the IEEE-TII, CIKM, and ISPA. His research interests include large-scale sparse matrix and tensor factorization, recommender systems, social networks, data mining, machine learning, and GPU and multi-GPU computing.



**XIAOHUI WEI** is currently pursuing the Ph.D. degree in computer science and technology with Hunan University, Changsha, China. His research interests include subspace learning, multi-view clustering, and hyperspectral image classification.

**KEQIN LI** is currently a SUNY Distinguished Professor of computer science with the State University of New York. He has published more than 620 journal articles, book chapters, and refereed conference papers, and has received several best paper awards. He currently serves or has served on the Editorial Boards for the IEEE TRANSACTIONS ON PARALLEL AND DISTRIBUTED SYSTEMS, the IEEE TRANSACTIONS ON COMPUTERS, the IEEE TRANSACTIONS ON CLOUD COMPUTING, the IEEE TRANSACTIONS ON SERVICES COMPUTING, and the IEEE TRANSACTIONS ON SUSTAINABLE COMPUTING. His current research interests include cloud computing, fog computing and mobile edge computing, energy-efficient computing and communication, embedded systems and cyber-physical systems, heterogeneous computing systems, big data computing, high-performance computing, CPU-GPU hybrid and cooperative computing, computer architectures and systems, computer networking, machine learning, and intelligent and soft computing.