Chinese Journal of Electronics
Vol.32, No.6, Nov. 2023

Reverse-Nearest-Neighbor-Based Clustering by Fast Search and Find of Density Peaks

ZHANG Chunhao[1], XIE Bin[1,2,3], and ZHANG Yiran[1]

(1. *College of Computer and Cyber Security, Hebei Normal University, Shijiazhuang 050024, China*)
(2. *Hebei Provincial Engineering Research Center for Supply Chain Big Data Analytics and Data Security, Hebei Normal University, Shijiazhuang 050024, China*)
(3. *Hebei Provincial Key Laboratory of Network and Information Security, Hebei Normal University, Shijiazhuang 050024, China*)

Abstract — Clustering by fast search and find of density peaks (CFSFDP) has the advantages of a novel idea, easy implementation, and efficient clustering. It has been widely recognized in various fields since it was proposed in *Science* in 2014. The CFSFDP algorithm also has certain limitations, such as non-unified sample density metrics defined by cutoff distance, the domino effect for the assignment of remaining samples triggered by unstable assignment strategy, and the phenomenon of picking wrong density peaks as cluster centers. We propose reverse-nearest-neighbor-based clustering by fast search and find of density peaks (RNN-CFSFDP) to avoid these shortcomings. We redesign and unify the sample density metric by introducing reverse nearest neighbor. The newly defined local density metric and the K-nearest neighbors of each sample are combined to make the assignment process more robust and alleviate the domino effect. A cluster fusion algorithm is proposed, which further alleviates the domino effect and effectively avoids the phenomenon of picking wrong density peaks as cluster centers. Experimental results on publicly available synthetic data sets and real-world data sets show that in most cases, the proposed algorithm is superior to or at least equivalent to the comparative methods in clustering performance. The proposed algorithm works better on manifold data sets and uneven density data sets.

Key words — Density peaks, Reverse nearest neighbor, Clustering, Cluster fusion.

I. Introduction

In data mining, the unsupervised learning method is represented by clustering. Its training set does not need to be labeled in advance, and the samples can be grouped by comparing the similarity between samples [1], which has better generalization. Therefore, it has attracted more and more attention from researchers and has been widely used in search engines, social networks, image segmentation, and multi-modal data analytics [2]–[5].

Researchers have proposed a variety of clustering algorithms based on different ideas, which can be broadly classified as partition-based [6], [7], grid-based [8], hierarchy-based [9]–[11], density-based [12], [13], and graph-based [14]. Specifically, the algorithm of clustering by fast search and find of density peaks (CFSFDP) [15], proposed in *Science* in 2014, is a density-based clustering algorithm. It is widely recognized in various fields among researchers due to its novel idea, easy implementation, and efficient clustering.

As with other clustering algorithms, the CFSFDP algorithm also has some stand-out limitations in the clustering process. In detail, the CFSFDP algorithm adopts different metrics for data sets of different sizes in calculating sample density, but there is no criterion for distinguishing the size of data sets [16]. The remaining sample assignment is prone to the domino effect on some manifold data sets because of the poor fault tolerance [17]. In the process of density peaks selection, the selection of cluster centers may all be in high-density clusters, resulting in poor cluster results [18].

The critical issue is improving its clustering performance and generalization ability over various datasets by optimizing the process of the CFSFDP algorithm. Researchers have proposed various improved

Manuscript Received June 8, 2022; Accepted Sept. 26, 2022. This work was supported by the National Natural Science Foundation of China (62076088) and the Technological Innovation Foundation of Hebei Normal University (L2020K09).
 DOI:10.23919/cje.2022.00.165

CFSFDP algorithms to address these stand-out limitations. Combining sample K-nearest neighbors is a practical improvement direction. Xie *et al.* [19] proposed the FKNN-DPC algorithm, which unifies the sample density metric by combining the K-nearest neighbors, and gives two assignment strategies to detect the real distribution of data sets. However, the density peaks are manually found only by analyzing the decision graph, which still leads to the wrong selection of cluster centers. In [20], the DPC-KNN algorithm was proposed to address the shortcomings of the CFSFDP algorithm in dealing with some non-spherical data sets, which easily triggers the domino effect and incorporates the idea of K-nearest neighbors into the distance calculation and assignment process. However, the relationship between the cutoff distance d_c and K does not address. Not only the new parameter K is introduced, but the cutoff distance d_c still needs to be taken manually. Zhang *et al.* [21] proposed the DC-SKCG algorithm based on the shared K-nearest neighbors between samples. An automatic fusion mechanism of redundant high-density core regions is designed to reduce the sensitivity of the algorithm to parameters. However, new parameters are introduced, and the complexity of the algorithm is improved. Liu *et al.* [22] proposed an adaptive clustering algorithm ADPC-KNN, which introduces the concept of K-nearest neighbors to calculate the global parameter d_c and the local density ρ_i of each sample. Finally, the clusters with reachable density are aggregated. However, the defect of the domino effect in the assignment process is still not effectively solved because the assignment method in the CFSFDP algorithm is followed. Bai *et al.* [23] proposed an accelerated algorithm, CFSFDP+A, involving less calculation about distance, which can obtain the same clustering results as the CFSFDP algorithm and improve the running speed of the CFSFDP algorithm. However, problems such as the inconsistent sample density metric and the assignment method prone to joint errors still exist. Bryant *et al.* [24] proposed the RNN-DBSCAN algorithm, which verified the advantage of the reverse nearest neighbor reflecting the local distribution of the sample. Therefore, in this paper, to address these limitations of the CFSFDP algorithm, we combine reverse nearest neighbor with the CFSFDP algorithm and design an RNN-CFSFDP algorithm. The clustering performance and generalization ability of the CFSFDP algorithm are further improved.

Contributions of this paper can be summarized as follows: 1) To unify the local density metric of the CFSFDP algorithm on different size data sets and avoid the artificial value of the cutoff distance d_c, we redefine the local density metric of the samples by combining the reverse nearest neighbor of the samples; 2) To alleviate the CFSFDP algorithm's shortcoming, which is prone to the domino effect in the remaining sample assignment, we improve the assignment strategy by taking advantage of the nearest neighbor sample to detect the local distribution of samples; 3) We propose a cluster fusion algorithm to prevent density peaks from the wrong selection and further alleviate the domino effect; 4) Extensive experiments are conducted to verify the effectiveness of our techniques over both publicly available synthetic data sets and UCI real-world data sets.

The remainder of this paper is organized as follows. The related works are presented in Section II. The defects of the CFSFDP algorithm and the corresponding improvement strategies are analyzed in Section III. In Section IV, the further details of the RNN-CFSFDP algorithm proposed in this paper are introduced based on Section III. The experimental results on publicly available synthetic data sets and UCI real-world data sets are analyzed in Section V. The current works and prospect research are summarized in Section VI.

II. Related Works

1. K-nearest neighbors and reverse nearest neighbor

Definition 1 (K-nearest neighbors [24]) The set of K-nearest neighbors of sample $\boldsymbol{x}$ is defined by the function $\mathrm{KNN}(\boldsymbol{x}) = \boldsymbol{S}$, where $\boldsymbol{S}$ satisfies the following conditions:

$$\forall \boldsymbol{y} \in \boldsymbol{S}, \boldsymbol{z} \in \boldsymbol{X}/(\boldsymbol{S} + \{\boldsymbol{x}\}) : \mathrm{dist}(\boldsymbol{x}, \boldsymbol{y}) \leq \mathrm{dist}(\boldsymbol{x}, \boldsymbol{z}) \quad (1)$$

where $\boldsymbol{X}$ is the set of samples, $\boldsymbol{S} \subseteq \boldsymbol{X}/\{\boldsymbol{x}\}$ is the set of K-nearest neighbors of sample $\boldsymbol{x}$, $|\boldsymbol{S}| = K$, and $\mathrm{dist}(\boldsymbol{x}, \boldsymbol{y})$ is the Euclidean distance of sample $\boldsymbol{x}$ and sample $\boldsymbol{y}$.

The K-nearest neighbors (KNN) algorithm is one of the most fundamental, robust, and versatile algorithms [25]. As shown in Definition 1, the KNN of sample $\boldsymbol{x}$ is the set consisting of the K samples nearest to sample $\boldsymbol{x}$ in the data set $\boldsymbol{X}$. The method requires only the choice of K, the neighbors to be considered. Small values of K will select the closest samples that can best estimate the correct classification at sample $\boldsymbol{x}$. However, the estimation will be prone to large fluctuations due to density because of the small numbers. Further, a reverse nearest neighbor (RNN) method is proposed.

Definition 2 (Reverse nearest neighbor [24]) The reverse nearest neighbor of sample $\boldsymbol{x}$ is defined by the function $\mathrm{RNN}(\boldsymbol{x}) = \boldsymbol{R}$, where $\boldsymbol{R}$ satisfies the following conditions:

$$\forall \boldsymbol{y} \in \boldsymbol{R} : \boldsymbol{x} \in \mathrm{KNN}(\boldsymbol{y}) \quad (2)$$

where $\boldsymbol{R} \subseteq \boldsymbol{X}/\{\boldsymbol{x}\}$ is the set of reverse nearest neighbor of sample $\boldsymbol{x}$.

The RNN can not only be obtained directly from KNN but also better reflect the local distribution of the data. As shown in Definition 2, the RNN of a sample $\boldsymbol{x}$ is the set consisting of samples in the data set $\boldsymbol{X}$ that takes $\boldsymbol{x}$ as its KNN. Fig.1 shows the distribution of KNN and RNN for sample 1 and sample 2 in the randomly generated data set when $K = 3$. By comparison, it is found that KNN reflects the local density of the samples more rigidly because KNN does not consider the local distribution of the samples and requires each sample to find K neighbors. In contrast, RNN can adaptively adjust the number of nearest neighbors according to the local distribution of the samples so that it can reflect the local density of the samples better.

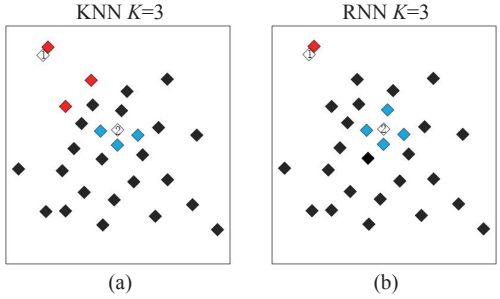

Fig. 1. Comparison of (a) K-nearest neighbors and (b) Reverse nearest neighbor.

2. Clustering by fast search and find of density peaks [15]

The CFSFDP algorithm relies on two crucial assumptions [26]: 1) The density of the sample at the cluster centers is higher than the density of the neighboring samples surrounding it. 2) The distance between the centers of different clusters is relatively far.

The density ρ_i of sample $\boldsymbol{x}_i$ is defined as

$$\rho_i = \sum_j \chi\left(d_{ij} - d_c\right) \tag{3}$$

$$\chi\left(d_{ij} - d_c\right) = \begin{cases} 1, & d_{ij} < d_c \\ 0, & \text{otherwise} \end{cases} \tag{4}$$

where d_{ij} is the Euclidean distance between sample $\boldsymbol{x}_i$ and sample $\boldsymbol{x}_j$, and the cutoff distance d_c needs to be given by the user depending on the specific details. The CFSFDP algorithm provides that one can choose d_c to make the average number of neighbors of a sample between 1% and 2% of the total number of samples.

In addition, for the case of insignificant density changes when dealing with small sample data, the Gaussian kernel function distance method is usually used to calculate the sample density by the formula

$$\rho_i = \sum_{j \neq i} \exp\left(-\left(\frac{d_{ij}}{d_c}\right)^2\right) \tag{5}$$

The relative distance δ_i of sample $\boldsymbol{x}_i$ is defined as

$$\delta_i = \min_{j:\rho_j > \rho_i} \left(d_{ij}\right) \tag{6}$$

If the sample $\boldsymbol{x}_i$ is the maximum density point, the relative distance δ_i is defined as

$$\delta_i = \max\left(d_{ij}\right) \tag{7}$$

By drawing the decision graph with ρ_i as the horizontal coordinate and δ_i as the vertical coordinate, the points from the decision graph where both ρ_i and δ_i are relatively large are selected as the density peak points. If the decision graph is not apparent, $\gamma_i = \rho_i \times \delta_i$ can be calculated for each sample, and the γ-decision graph can be drawn by arranging them in ascending or descending order as the vertical coordinate. The points with relatively larger γ_i are selected as the density peak points. Finally, the remaining samples other than the density peak points are assigned to the cluster where the nearest neighboring sample with a larger density is located.

III. Defect Analysis of the CFSFDP Algorithm

Although the experimental results [15] obtained for the CFSFDP algorithm show that it performs well in many cases, it still has some defects. In this section, the defects of the CFSFDP algorithm will be analyzed in detail, and solution strategies will be given.

1. Sample density metric

The CFSFDP algorithm is vulnerable to human intervention in calculating sample density. As suggested in [15], the cutoff distance method is used to calculate the sample density in data sets with larger sample sizes, and the Gaussian kernel function distance method is used to calculate the sample density in data sets with smaller sample sizes. However, since there are no standardized criteria for measuring the size of a dataset during the application, it becomes difficult for researchers to choose which sample density metric to use when faced with a real problem. In addition, even if a suitable density metric is chosen, the cutoff distance d_c still needs to be set artificially. Although the number of neighbors of a sample within the cutoff distance d_c is considered 1% to 2% of the total number of samples to achieve good results as suggested in [15], the way d_c is taken lacks theoretical proof. It does not yield good results on some data sets. Fig.2 shows the clustering results on the Flame dataset [27] using two cutoff meth-

ods and different cutoff distances. Among them, the best clustering result occurs when the $d_c = 3\%$ exceeds the values range between 1% and 2%. It can be found that different cutoff methods and different cutoff distances lead to different clustering results. This paper redefines the sample local density metric in combination with the reverse nearest neighbor of the sample. The local density of the sample can be calculated adaptively without considering the selection of the cutoff method and cutoff distance.

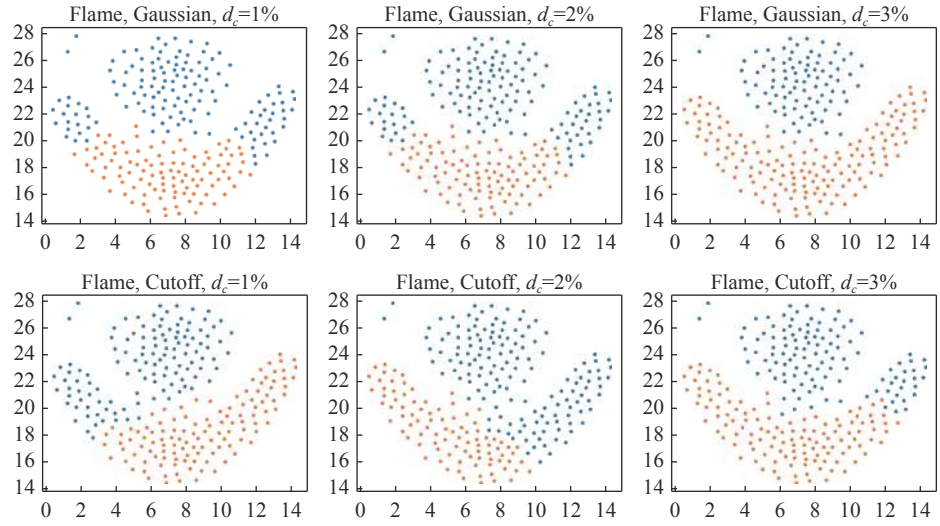

Fig. 2. Comparison of clustering results on Flame dataset using two cutoff methods and different cutoff distances.

2. Remaining samples assignment

The CFSFDP algorithm requires no iteration in the clustering process but only one assignment process. In the assignment process, the samples are sorted in descending order of density. Then the remaining samples are sequentially assigned to the cluster where the nearest neighbor sample with a larger density than itself is located. This also means that a wrong sample assignment during the assignment process will cause the neighbor samples with a smaller density to be incorrectly assigned, resulting in the wrong joint assignment, often referred to as the domino effect. This problem is particularly prone to occur on manifold data sets. Fig.3 shows the visualization of the Spiral dataset [28] and the clustering results of the Spiral dataset using the CFSFDP algorithm. By comparing and analyzing the experimental results, it is found that those wrong assignments from sample 1 to sample 13 are caused by the wrong assignment of sample 12. Since the density of sample 12 is relatively larger among the surrounding neighbors, sample 12 is assigned earlier than sample 13. Since the same-cluster sample with a greater density than sample 12 and the closest relative distance is sample 18, the correct assignment is to assign sample 12 to the cluster in which sample 18 is located. However, since sample 12 is closer to sample 150, a sample of a heterogeneous cluster with a greater density than it, it leads to the wrong assignment of sample 12 to the cluster where sample 150 is located. This eventually leads to the wrong assignment of all the near-neighbor samples that are smaller dense than sample 12, and thus the wrong joint assignment of sample 1 to sample 13 occurs.

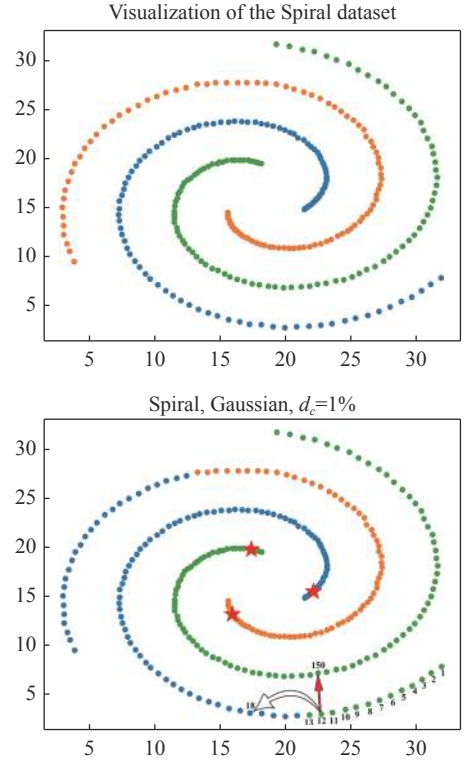

Fig. 3. The process of sample wrong assignment.

In this paper, we optimize the assignment strategy in combination with sample nearest neighbors to allevi-

ate the shortcomings of the CFSFDP algorithm, which is prone to the domino effect on manifold datasets.

3. Density peaks selection

Although the decision graph of the CFSFDP algorithm provides a good heuristic method for the selection of cluster centers, there are still cases where it is difficult to select or incorrectly selected. These cases weaken the clustering performance and are caused by two main reasons as follows:

1) Due to the difference in the size of the clusters in the dataset, there are multiple samples with higher density and greater relative distance, resulting in the inability to select the density peak points from the decision graph intuitively. As shown in the γ-decision graph of Aggregation dataset [29] in Fig.4(a), the relatively greater points from the decision graph are manually selected as the cluster centers according to the principle of cluster centers selection. As shown in Fig.4(b), the Aggregation dataset is a dataset consisting of seven clusters. However, it is difficult to directly select the seven density peak points just by observing its decision graph, which may be incorrectly selected as three or eight.

2) The density differences between different clusters influence the selection of cluster centers. There is a significant difference in the density and relative distance between the cluster center of the low-density cluster and the cluster center of the high-density cluster. This situation suggests that the selected cluster centers are all located in high-density clusters, while no cluster centers are found in low-density clusters. As shown in Fig.4(c) and (d), the Jain dataset [30] consists of two clusters with a significant difference in density. Although two density peak points can be selected from the decision graph intuitively, both are located in high-density clusters, thus leading to unsatisfactory clustering results.

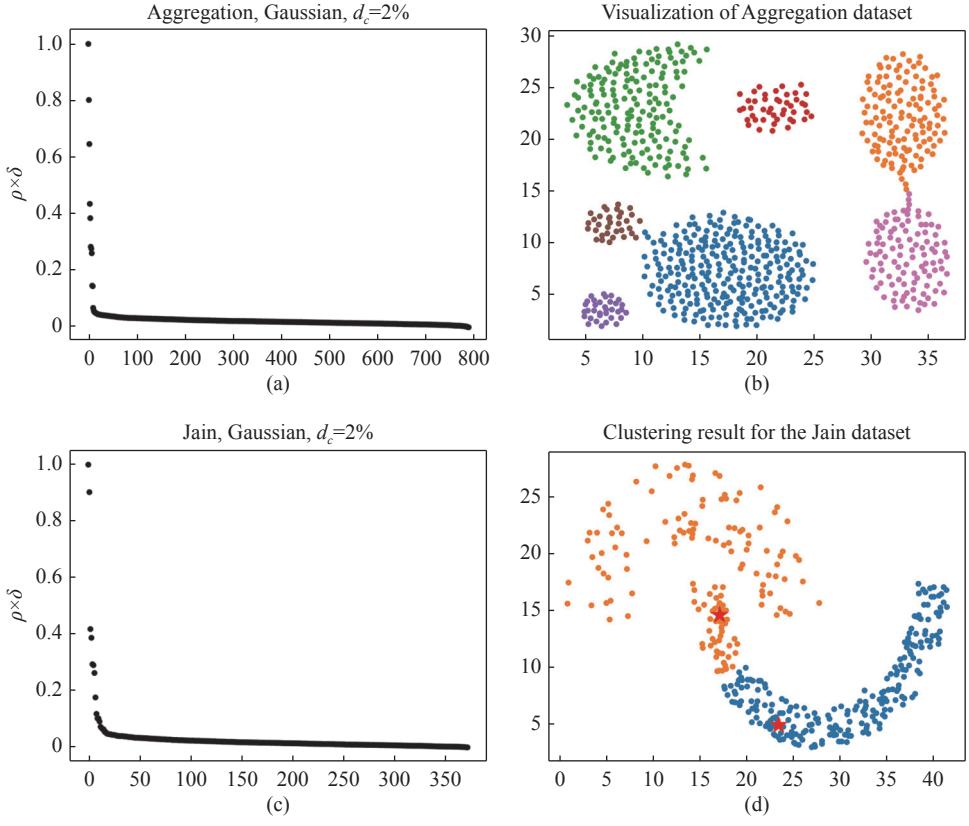

Fig. 4. Clustering results for (a) and (b) Aggregation dataset; (c) and (d) Jain datasets.

We re-specifies the rule for selecting density peak points from the γ-decision graph and propose a cluster fusion algorithm. The proposed method effectively solves the problem of the wrong selection of density peaks in the CFSFDP algorithm and further improves the clustering performance of the algorithm.

IV. The RNN-CFSFDP Algorithm

In order to address the CFSFDP algorithm's shortcomings, we propose a reverse-nearest-neighbor-based clustering by fast search and find of density peaks (RNN-CFSFDP). First, we define and unify the sample local density metric with the reverse nearest neighbor.

Then, we optimize the assignment strategy by combining the nearest neighbor samples. Finally, we propose that the cluster fusion algorithm further optimizes the clustering process by fusing similar clusters.

1. Sample local density metric combining reverse nearest neighbor

The new definition method uses the same metric of local density for different size data sets. It does not need to manually select the cutoff distance d_c, which improves the generalization, practicality, and operability of the CFSFDP algorithm. The local density ρ_i of sample $\boldsymbol{x}_i$ is defined as

$$\rho_i = \sum_{j \in \mathrm{RNN}(\boldsymbol{x}_i)} \exp\left(-d_{ij}\right) \tag{8}$$

where d_{ij} is the similarity between sample $\boldsymbol{x}_i$ and sample $\boldsymbol{x}_j$, and the Euclidean distance is used in this paper. When $\mathrm{RNN}(\boldsymbol{x}_i)$ is the empty set, $\rho_i = 0$.

The local density ρ_i of a sample $\boldsymbol{x}_i$ has the following property: as the value of K increases, the number of reverse nearest neighbor $\mathrm{RNN}(\boldsymbol{x}_i)$ of each sample $\boldsymbol{x}_i$ increases accordingly. After fixing the value of K, the set of reverse nearest neighbor $\mathrm{RNN}(\boldsymbol{x}_i)$ of each sample $\boldsymbol{x}_i$ also varies. The sample in denser locations has a larger number of reverse neighbors, and the local density of the sample is larger. The sample in sparse locations has fewer reverse neighbors, and the local density of the sample is smaller. The definition method fully considers the local information of the samples and can better reflect the local distribution of the samples.

2. Sample assignment strategy combining K-nearest neighbors

Combined with the analysis in Section III.2, it is found that considering only one sample is less fault-tolerant, so the assignment strategy is adjusted to assign sample $\boldsymbol{x}_i$ to the cluster which has a larger density than ρ_i and is closest to the sample $\boldsymbol{x}_i$ and its $\mathrm{KNN}(\boldsymbol{x}_i)$. The relative distance is defined as

$$\delta_i = \begin{cases} \min\left(d_{jk}\right), \boldsymbol{x}_j \in \boldsymbol{M}_i, \boldsymbol{x}_k \in \boldsymbol{H}_i \\ \max\left(d_{it}\right), \rho_i = \max(\rho), \boldsymbol{x}_t \in \boldsymbol{X} \end{cases} \tag{9}$$

where $\boldsymbol{M}_i = \mathrm{KNN}\left(\boldsymbol{x}_i\right) \cup \{\boldsymbol{x}_i\}$ is the union of sample $\boldsymbol{x}_i$ and its K-nearest neighbors. $\boldsymbol{H}_i = \{\boldsymbol{x}_k \mid \rho_k > \rho_i, \boldsymbol{x}_k \in \boldsymbol{X}, \boldsymbol{x}_k \neq \boldsymbol{x}_i\}$ is the set of samples with local density greater than ρ_i. $\max(\rho)$ is the maximum value of the local density of samples in the data set. The sample local density metric uses equation (8).

3. Cluster fusion combining shared reverse nearest neighbor

In this part, we define the concept of shared reverse nearest neighbors between samples and shared reverse nearest neighbors between clusters combing with reverse K-nearest neighbors, design similarity between clusters based on this concept, and propose a cluster fusion algorithm to merge the clusters with high similarity in turn.

Definition 3 (Sample shared reverse nearest neighbor set) For two samples $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$, if there exists sample $\boldsymbol{x}_k$, $\boldsymbol{x}_k \neq \boldsymbol{x}_i$ and $\boldsymbol{x}_k \neq \boldsymbol{x}_j$, so that $\boldsymbol{x}_k \in \mathrm{RNN}(\boldsymbol{x}_i)$ and $\boldsymbol{x}_k \in \mathrm{RNN}(\boldsymbol{x}_j)$, then the $\mathrm{SRNN}\left(\boldsymbol{x}_i, \boldsymbol{x}_j\right)$ consisting of sample $\boldsymbol{x}_k$ is called the sample shared reverse nearest neighbor set of sample $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$. The $\mathrm{SRNN}\left(\boldsymbol{x}_i, \boldsymbol{x}_j\right)$ is defined as

$$\mathrm{SRNN}\left(\boldsymbol{x}_i, \boldsymbol{x}_j\right) = \{\boldsymbol{x}_k \mid \boldsymbol{x}_k \in \mathrm{RNN}(\boldsymbol{x}_i) \text{ and } \boldsymbol{x}_k \in \mathrm{RNN}(\boldsymbol{x}_j)\} \tag{10}$$

The $\mathrm{SRNN}\left(\boldsymbol{x}_i, \boldsymbol{x}_j\right)$ of samples $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$ have the following properties: When fixing the value of K, if the number of shared reverse neighbors between samples $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$ is few, it means that the distance between samples $\boldsymbol{x}_i$ to $\boldsymbol{x}_j$ is far and the similarity is low. If the number of shared reverse neighbors between samples $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$ is many, it means that the distance between samples $\boldsymbol{x}_i$ to $\boldsymbol{x}_j$ is close, and the similarity is high.

Unlike KNN, the RNN of samples $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$ is related to the density of the samples' location. Therefore, under the same conditions, if the location where the samples $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$ are located is denser, the similarity between samples $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$ is higher. On the contrary, the similarity between the samples $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$ is lower. This gap can be further increased by adjusting the value of K.

Definition 4 (Cluster shared reverse nearest neighbor set) For two clusters $\boldsymbol{C}_m$ and $\boldsymbol{C}_n$, for any $\boldsymbol{x}_i \in \boldsymbol{C}_m$ and $\boldsymbol{x}_j \in \boldsymbol{C}_n$, if there exists $\boldsymbol{x}_k \in \mathrm{SRNN}\left(\boldsymbol{x}_i, \boldsymbol{x}_j\right)$ and $\boldsymbol{x}_k \in \boldsymbol{C}_m \cup \boldsymbol{C}_n$, the $\mathrm{SRNN}\left(\boldsymbol{C}_m, \boldsymbol{C}_n\right)$ consisting of $\boldsymbol{x}_k$ is called to be the cluster shared reverse nearest neighbor set of cluster $\boldsymbol{C}_m$ and $\boldsymbol{C}_n$. The $\mathrm{SRNN}\left(\boldsymbol{C}_m, \boldsymbol{C}_n\right)$ is defined as

$$\begin{aligned} \mathrm{SRNN}\left(\boldsymbol{C}_m, \boldsymbol{C}_n\right) = \{\boldsymbol{x}_k \mid \boldsymbol{x}_k \in (\boldsymbol{C}_m \cup \boldsymbol{C}_n) \cap \mathrm{SRNN}\left(\boldsymbol{x}_i, \boldsymbol{x}_j\right), \\ \boldsymbol{x}_i \in \boldsymbol{C}_m, \boldsymbol{x}_j \in \boldsymbol{C}_n\} \end{aligned} \tag{11}$$

The $\mathrm{SRNN}\left(\boldsymbol{C}_m, \boldsymbol{C}_n\right)$ of clusters $\boldsymbol{C}_m$ and $\boldsymbol{C}_n$ have the following properties: When fixing the value of K, if the number of shared reverse neighbors between clusters $\boldsymbol{C}_m$ and $\boldsymbol{C}_n$ is few, it means that the distance between clusters $\boldsymbol{C}_m$ to $\boldsymbol{C}_n$ is far and the similarity is low. If the number of shared reverse neighbors between clusters $\boldsymbol{C}_m$ and $\boldsymbol{C}_n$ is many, it means that the distance between clusters $\boldsymbol{C}_m$ to $\boldsymbol{C}_n$ is close, and the similarity is high. This gap can be further increased by adjusting the value of K.

Definition 5 (Cluster similarity) The similarity $\mathrm{Sim}(\boldsymbol{C}_m, \boldsymbol{C}_n)$ of clusters $\boldsymbol{C}_m$ and $\boldsymbol{C}_n$ is defined as

$$\begin{aligned}&\mathrm{Sim}\left(\boldsymbol{C}_m, \boldsymbol{C}_n\right)\\&= \frac{\left|\boldsymbol{C}_n\right|\left|\mathrm{SRNN}\left(\boldsymbol{C}_m, \boldsymbol{C}_n\right)_m\right| + \left|\boldsymbol{C}_m\right|\left|\mathrm{SRNN}\left(\boldsymbol{C}_m, \boldsymbol{C}_n\right)_n\right|}{2\left|\boldsymbol{C}_m\right|\left|\boldsymbol{C}_n\right|}\end{aligned} \tag{12}$$

where $|\boldsymbol{C}_m|$ is the number of samples in cluster $\boldsymbol{C}_m$, and $|\boldsymbol{C}_n|$ is the number of samples in cluster $\boldsymbol{C}_n$; $|\mathrm{SRNN}(\boldsymbol{C}_m, \boldsymbol{C}_n)_m|$ is the number of samples belonging to $\boldsymbol{C}_m$ in the cluster shared reverse nearest neighbors of cluster $\boldsymbol{C}_m$ and $\boldsymbol{C}_n$; $|\mathrm{SRNN}(\boldsymbol{C}_m, \boldsymbol{C}_n)_n|$ is the number of samples belonging to $\boldsymbol{C}_n$ in the cluster shared reverse nearest neighbors of cluster $\boldsymbol{C}_m$ and $\boldsymbol{C}_n$. The equation (12) can be expressed in the form of (13), from which the meaning of cluster similarity can be clearly observed.

$$\begin{aligned}&\mathrm{Sim}\left(\boldsymbol{C}_m, \boldsymbol{C}_n\right)\\&= \frac{1}{2}\left(\frac{\left|\mathrm{SRNN}\left(\boldsymbol{C}_m, \boldsymbol{C}_n\right)_m\right|}{\left|\boldsymbol{C}_m\right|} + \frac{\left|\mathrm{SRNN}\left(\boldsymbol{C}_m, \boldsymbol{C}_n\right)_n\right|}{\left|\boldsymbol{C}_n\right|}\right)\\&= \frac{\left|\boldsymbol{C}_n\right|\left|\mathrm{SRNN}\left(\boldsymbol{C}_m, \boldsymbol{C}_n\right)_m\right| + \left|\boldsymbol{C}_m\right|\left|\mathrm{SRNN}\left(\boldsymbol{C}_m, \boldsymbol{C}_n\right)_n\right|}{2\left|\boldsymbol{C}_m\right|\left|\boldsymbol{C}_n\right|}\end{aligned} \tag{13}$$

In the design process of the clusters similarity $\mathrm{Sim}(\boldsymbol{C}_m, \boldsymbol{C}_n)$, in order to prevent the situation that cluster $\boldsymbol{C}_m$ and cluster $\boldsymbol{C}_n$ have inaccurate similarity measures due to too disparate sizes, equation (13) is used to calculate the number of shared reverse nearest neighbors by calculating the arithmetic average of the number of shared reverse nearest neighbors located in the two clusters separately instead of directly calculating the number of shared reverse nearest neighbors of the two clusters similarity.

The process of the cluster fusion algorithm is shown in Algorithm 1.

Algorithm 1 The cluster fusion algorithm

Input: The initial clustering result $\boldsymbol{L} = \{\boldsymbol{C}_j\}_{j=1}^{l}, l \geq k$, the number of clusters k.
Output: Final clustering result $\boldsymbol{C} = \{\boldsymbol{C}_j\}_{j=1}^{k}$.

1: The similarity matrix $\boldsymbol{SM}^{l\times l} = \{\mathrm{Sim}(\boldsymbol{C}_i, \boldsymbol{C}_j)\}^{l\times l}$ is calculated according to equation (13);
2: while $l > k$;
3: Merge the two clusters with the highest similarity $\boldsymbol{C}_i$ and $\boldsymbol{C}_j$ into cluster $\boldsymbol{C}_{i,j}$, $i < j$, let i be the cluster label for the cluster $\boldsymbol{C}_{i,j}$;
4: Update the similarity between the remaining cluster $\boldsymbol{C}_m \in \boldsymbol{L}/\{\boldsymbol{C}_i, \boldsymbol{C}_j\}$ and cluster $\mathbf{C}_{i,j}$ as $\mathrm{Sim}(\mathbf{C}_m, \mathbf{C}_{i,j}) = \max(\mathrm{Sim}(\mathbf{C}_m, \mathbf{C}_i), \mathrm{Sim}(\mathbf{C}_m, \mathbf{C}_j))$;
5: end while;
6: Update the cluster labels as 1 to k.

4. The process of the RNN-CFSFDP algorithm

The RNN-CFSFDP algorithm still adopts the basic idea of the CFSFDP algorithm to quickly find the points with larger local density and relative distance as the clustering centers. Further, the metric of local density and relative distance is improved, and the final clustering results are fused. First, we find the KNN and RNN of each sample according to equations (1) and (2), calculate the ρ_i of each sample according to (8), calculate the δ_i of each sample according to (9), calculate the γ of each sample, and arrange them in descending order, and draw the γ-decision graph. In the process of selecting the density peak points in the decision graph, in order to prevent the case of wrong selection, we directly select all the points "floating" in the decision graph as potential density peak points and assign the remaining sample $\boldsymbol{x}_i$ to the cluster which has a larger density than ρ_i and is closest to the sample $\boldsymbol{x}_i$ and its KNN($\boldsymbol{x}_i$). Finally, cluster fusion is performed according to Algorithm 1 to obtain the final clustering results.

The process of the RNN-CFSFDP algorithm is shown in Algorithm 2. Fig.5 shows the clustering process for the Flame dataset, where the density peak points are selected as shown in Fig.5(a), the initial clustering results are generated as shown in Fig.5(b), and the similarity matrix is calculated as shown in Fig.5(c), and the final clustering results are generated as shown in Fig.5(d).

Algorithm 2 The RNN-CFSFDP algorithm

Input: The K for the reverse nearest neighbor, the number of clusters k.
Output: Final clustering result $\boldsymbol{C} = \{\boldsymbol{C}_j\}_{j=1}^{k}$.

1: Calculate ρ_i of each sample according to equation (8);
2: Calculate δ_i of each sample according to equation (9);
3: Calculate the $\gamma_i = \rho_i \times \delta_i$ of each sample, arrange them in descending order, and plot γ-decision graph;
4: Select the density peak points;
5: Assign the remaining samples according to the assignment strategy in Section IV.2;
6: Perform cluster fusion according to Algorithm 1.

V. Experiment

This section compares the proposed RNN-CFSFDP algorithm with five clustering algorithms, k-means, FCM, AGNES, CFSFDP, and DPC-KNN, on seventeen data sets. Five commonly used clustering evaluation indexes verify the effectiveness of the RNN-CFSFDP algorithm. To further verify the effectiveness of each improved part, we design ablation experiments. The experimental environment for all algorithms is Windows 10 64bit operating system, PyCharm Com-

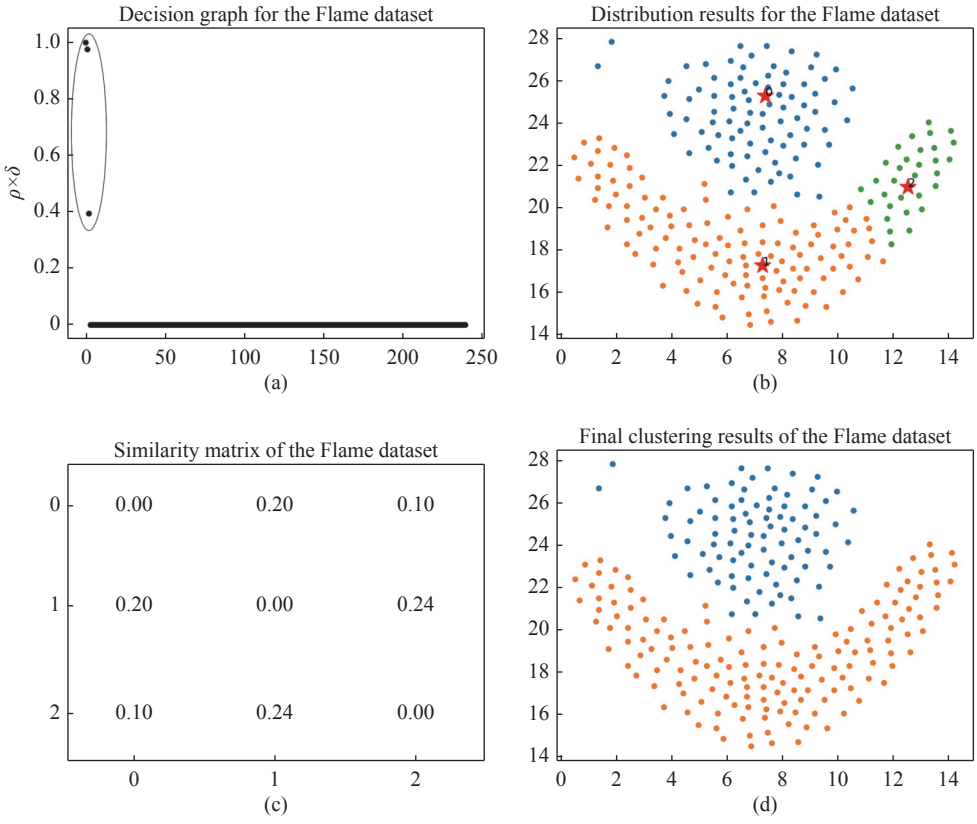

Fig. 5. Clustering process for the Flame dataset. (a) Decision graph; (b) Distribution results; (c) Similarity matrix; (d) Final clustering results.

munity 2020.3.2, 12 GB RAM, and Intel(R) Core(TM) i5-4210H CPU@2.90 GHz.

1. Data sets

The data sets used in the experiments are selected from publicly available synthetic data sets and UCI real-world data sets with different numbers of clusters, different sizes, different shapes, and different densities. For example, the Flame dataset is a semi-enveloped structure consisting of two clusters, one tightly surrounded by the other, and the two clusters are closely connected. The Aggregation dataset consists of seven clusters with a relatively uniform density distribution, in which two pairs of clusters are slightly connected. The Spiral dataset consists of three clusters, each of which is toroidal. The Jain dataset and the Banana dataset are two crescent-shaped clusters connected alternately. The Jain dataset has a large difference in density between the two clusters, and the Banana dataset has a more uniform density between the two clusters but a larger number of samples. The R15 dataset and the D31 dataset belong to the data sets with a larger number of samples and clusters.

The details and sources of the data sets used in the comparison experiments are shown in Table 1 [21], [27]–[33].

2. Evaluation indicators

The experimental results were evaluated by using the commonly used clustering evaluation indexes accuracy (Acc), adjusted mutual information (AMI) [17], normalized mutual information (NMI) [34], adjusted rand index (ARI) [35] and fowlkes-mallows index (FMI) [36].

Acc represents the number of correctly clustered samples among all samples as a percentage of the total. Acc is calculated as

Table 1. Datasets

Datasets	Instances	Attributes	Clusters	Source
Flame	240	2	2	[27]
Aggregation	788	2	7	[29]
Spiral	312	2	3	[28]
Jain	373	2	2	[30]
4k2_far	400	2	4	[21]
R15	600	2	15	[31]
D31	3100	2	31	[31]
Banana	4811	2	2	UCI
Spiral3D	318	3	3	[32]
Iris	150	4	3	UCI
Wine	178	13	3	UCI
Sonar	208	60	2	UCI
Movement_libras	360	90	15	UCI
Ionosphere	351	34	2	UCI
Ecoli	336	8	8	UCI
Leuk72_3k	72	39	3	[21]
Compound	399	2	6	[33]

$$\mathrm{Acc} = \frac{\sum_{i=1}^{N} \delta\left(cl_i, r_i\right)}{N} \tag{14}$$

where N denotes the total number of samples in the dataset, cl_i and r_i denote the labels obtained by the clustering algorithm and the true labels, respectively, and $\delta(\cdot)$ is the indicator function, which is calculated as

$$\delta(x, y) = \begin{cases} 1, & x = y \\ 0, & \text{otherwise} \end{cases} \tag{15}$$

The value of Acc is in the range [0, 1], the larger value, the better clustering effect.

ARI is commonly used in the evaluation of clustering algorithms, and its predecessor is the Rand Index (RI). Calculating the RI requires the true label information of dataset. Suppose the true label of the dataset is $\boldsymbol{U}$ and the predicted label after clustering is $\boldsymbol{V}$. Then a is expressed as the number of pairs of the data objects that belong to the same class in $\boldsymbol{U}$ and $\boldsymbol{V}$. b is expressed as the number of pairs of the data objects that belong to the same class in $\boldsymbol{U}$ and belong to different classes in $\boldsymbol{V}$. c is expressed as the number of pairs of the data objects that belong to the different classes in $\boldsymbol{U}$ and belong to the same class in $\boldsymbol{V}$. d is expressed as the number of pairs of the data objects that belong to different classes in $\boldsymbol{U}$ and $\boldsymbol{V}$. Then the formula for RI is defined as

$$\mathrm{RI} = \frac{a + b}{a + b + c + d} \tag{16}$$

where RI is a real number in [0, 1], the larger the RI is, the better the clustering effect is. The defect of RI is that for two random divisions, it is not guaranteed to make RI close to 0. To overcome this shortcoming, ARI is proposed. The formula for ARI is defined as

$$\mathrm{ARI} = \frac{\mathrm{RI} - E(\mathrm{RI})}{\max(\mathrm{RI}) - E(\mathrm{RI})} \tag{17}$$

where the value of ARI is in the range [−1, 1], and the closer ARI to 1, the better clustering quality.

The formula for FMI is defined as

$$\mathrm{FMI} = \frac{a}{\sqrt{(a + b)(a + d)}} \tag{18}$$

where the value of FMI is in the range [−1, 1], and the closer FMI to 1, the better clustering quality.

AMI is an improvement of mutual information (MI). MI takes values in [0, 1], but for random results, there is no guarantee that the MI value is close to 0. To solve this problem, it is proposed that AMI can better reflect the data distribution, and the formula is defined as

$$\mathrm{AMI} = \frac{\mathrm{MI} - E|\mathrm{MI}|}{\max(H(\boldsymbol{U}), H(\boldsymbol{V})) - E|\mathrm{MI}|} \tag{19}$$

where $H(\boldsymbol{U})$ is the edge entropy value of the sample and $E|\mathrm{MI}|$ is the mathematical expectation of mutual information. The value of AMI is in the range [−1, 1], and the larger value, the better clustering result.

The value of NMI is in the range [0, 1], and the larger value of NMI is, the better clustering result. The formula is defined as

$$\mathrm{NMI} = \frac{2\mathrm{MI}}{H(\boldsymbol{U}) + H(\boldsymbol{V})} \tag{20}$$

3. Experimental results and analysis

Table 2 shows the settings of the experimental parameters. Table 3 shows the results of k-means, FCM, AGNES, CFSFDP, and DPC-KNN algorithms compared with the RNN-CFSFDP algorithm on Acc, AMI, NMI, ARI, and FMI evaluation indexes. The experimental results show that the RNN-CFSFDP algorithm obtained the best results for all five evaluation indexes on the eight data sets. More than half of the evaluation indexes achieved the best results on the Spiral3D, Sonar, Ionosphere, and Leuk72_3k datasets. While only two evaluation indexes achieved the best results on the Movement_libras, Ecoli, and Compound datasets, they were the second to the best results on the other three. The evaluation results on the D31 dataset are lower than the k-means algorithm but higher than several other clustering algorithms. The main reason is that the D31 dataset consists of spherical clusters, which makes the k-means algorithm work better to its advantage.

The experimental results show that RNN-CFSFDP algorithm outperforms the commonly used clustering algorithms overall. It can weaken the manual intervention and enhances the robustness compared with the CFSFDP algorithm and its improvement algorithms. In most cases, the algorithm outperforms or is at least comparable to comparative methods in terms of clustering performance. In particular, it optimizes the assignment strategy of remaining samples by considering sample neighbors and cluster fusion. Furthermore, it shows better results on manifold and density inhomogeneous data sets. For a more visual presentation, Fig.6 visualizes the clustering results of the RNN-CFSFDP algorithm on manifold data sets.

4. Ablation experiment

To further verify the effectiveness of each improvement module, this section proposes three variants of the CFSFDP algorithm: 1) CFSFDP_1. The sample density metric of the CFSFDP algorithm is improved to equation (8). 2) CFSFDP_2. The relative distance cal-

Table 2. Experimental parameter setting situation

Data sets	k-means	FCM	AGNES	CFSFDP	DPC-KNN	RNN-CFSFDP
Flame	k=2	k=2, m=2	average, k=2	d_c=1.4, k=2	d_c=1.6008, K=4, k=2	K=20, k=2
Aggregation	k=7	k=7, m=2	average, k=7	d_c=1.1, k=7	d_c=3.1185, K=7, k=7	K=4, k=7
Spiral	k=3	k=3, m=2	complete, k=3	d_c=1.5, k=3	d_c=13.6041, K=7, k=3	K=6, k=3
Jain	k=2	k=2, m=2	complete, k=2	d_c=14, k=2	d_c=13.0124, K=9, k=2	K=8, k=2
4k2_far	k=4	k=4, m=2	ward, k=4	d_c=1, k=4	d_c=0.2170, K=10, k=4	K=1, k=4
R15	k=15	k=15, m=2	average, k=15	d_c=0.4, k=15	d_c=0.6551, K=8, k=15	K=15, k=15
D31	k=31	k=31, m=2	complete, k=31	d_c=0.6, k=31	d_c=1.4312, K=28, k=31	K=17, k=31
Banana	k=2	k=2, m=2	complete, k=2	d_c=0.03, k=2	d_c=0.0206, K=2, k=2	K=8, k=2
Spiral3D	k=3	k=3, m=2	ward, k=3	d_c=0.05, k=3	d_c=0.0511, K=21, k=3	K=22, k=3
Iris	k=3	k=3, m=2	average, k=3	d_c=0.3, k=3	d_c=0.3162, K=7, k=3	K=5, k=3
Wine	k=3	k=3, m=2	ward, k=3	d_c=0.5, k=3	d_c=96.4202, K=5, k=3	K=17, k=3
Sonar	k=2	k=2, m=2	average, k=2	d_c=0.2, k=2	d_c=0.7446, K=3, k=2	K=6, k=2
Movement_libras	k=15	k=15, m=2	ward, k=15	d_c=0.5, k=15	d_c=0.9406, K=3, k=15	K=8, k=15
Ionosphere	k=2	k=2, m=2	ward, k=2	d_c=0.5, k=2	d_c=0.6817, K=9, k=2	K=5, k=2
Ecoli	k=8	k=8, m=2	average, k=8	d_c=0.4, k=8	d_c=0.1300, K=14, k=8	K=2, k=8
Leuk72_3k	k=3	k=3, m=2	ward, k=3	d_c=1.6, k=3	d_c=4.2868, K=12, k=3	K=1, k=3
Compound	k=6	k=6, m=2	average, k=6	d_c=1.2, k=6	d_c=1.2500, K=8, k=6	K=12, k=6

Table 3. Comparison of experimental results of different algorithms on different data sets

Data sets	Evaluation indexes	Algorithms					
		k-means	FCM	AGNES	CFSFDP	DPC-KNN	RNN-CFSFDP
Flame	Acc	0.8375	0.8500	0.8333	**1.0000**	0.7830	**1.0000**
	AMI	0.3969	0.4403	0.4814	**1.0000**	0.8807	**1.0000**
	NMI	0.3988	0.4420	0.4831	**1.0000**	0.8824	**1.0000**
	ARI	0.4534	0.4880	0.4422	**1.0000**	0.7390	**1.0000**
	FMI	0.7364	0.7530	0.7311	**1.0000**	0.7952	**1.0000**
Aggregation	Acc	0.7843	0.6332	**0.9962**	0.7513	0.8503	**0.9962**
	AMI	0.8776	0.7598	**0.9894**	0.8736	0.9026	**0.9894**
	NMI	0.8792	0.7629	**0.9896**	0.8754	0.9039	**0.9896**
	ARI	0.7622	0.6113	**0.9935**	0.7084	0.7766	**0.9935**
	FMI	0.8158	0.6917	**0.9949**	0.7701	0.8237	**0.9949**
Spiral	Acc	0.3429	0.3397	0.3814	0.9487	**1.0000**	**1.0000**
	AMI	−0.0052	−0.0057	0.0071	0.8641	**1.0000**	**1.0000**
	NMI	0.0007	0.0002	0.0130	0.8649	**1.0000**	**1.0000**
	ARI	−0.0057	−0.0062	0.0046	0.8555	**1.0000**	**1.0000**
	FMI	0.3279	0.3272	0.3499	0.9038	**1.0000**	**1.0000**
Jain	Acc	0.7855	0.7748	0.9464	**1.0000**	**1.0000**	**1.0000**
	AMI	0.3677	0.3541	0.6956	**1.0000**	**1.0000**	**1.0000**
	NMI	0.3690	0.3555	0.6964	**1.0000**	**1.0000**	**1.0000**
	ARI	0.3241	0.3004	0.7792	**1.0000**	**1.0000**	**1.0000**
	FMI	0.7005	0.6894	0.9218	**1.0000**	**1.0000**	**1.0000**
4k2_far	Acc	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**
	AMI	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**
	NMI	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**
	ARI	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**
	FMI	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**
R15	Acc	**0.9967**	**0.9967**	0.9950	**0.9967**	**0.9967**	**0.9967**
	AMI	**0.9938**	**0.9938**	0.9916	**0.9938**	**0.9938**	**0.9938**
	NMI	**0.9942**	**0.9942**	0.9922	**0.9942**	**0.9942**	**0.9942**
	ARI	**0.9928**	**0.9928**	0.9893	**0.9928**	**0.9928**	**0.9928**
	FMI	**0.9932**	**0.9932**	0.9900	**0.9932**	**0.9932**	**0.9932**
D31	Acc	**0.9771**	0.8468	0.9619	0.9681	0.9687	0.9710
	AMI	**0.9660**	0.9196	0.9495	0.9548	0.9567	0.9589
	NMI	**0.9675**	0.9234	0.9519	0.9569	0.9587	0.9608
	ARI	**0.9535**	0.8236	0.9238	0.9358	0.9372	0.9414
	FMI	**0.9550**	0.8304	0.9262	0.9378	0.9392	0.9433

Table 3 (Continued)

Data sets	Evaluation indexes	Algorithms					
		k-means	FCM	AGNES	CFSFDP	DPC-KNN	RNN-CFSFDP
Banana	Acc	0.8285	0.8314	0.7774	0.6086	0.6142	**1.0000**
	AMI	0.3398	0.3470	0.2340	0.0330	0.0367	**1.0000**
	NMI	0.3399	0.3471	0.2342	0.0331	0.0368	**1.0000**
	ARI	0.4316	0.4392	0.3076	0.0469	0.0519	**1.0000**
	FMI	0.7178	0.7218	0.6551	0.5303	0.5333	**1.0000**
Spiral3D	Acc	0.3522	0.3522	0.3616	0.3648	0.3994	**0.4025**
	AMI	−0.0038	−0.0040	−0.0013	0.0046	0.0560	**0.0571**
	NMI	0.0021	0.0019	0.0046	0.0109	0.0619	**0.0643**
	ARI	−0.0042	−0.0044	−0.0023	−0.0009	**0.0218**	0.0091
	FMI	0.3385	0.3360	0.3409	0.3773	0.3853	**0.4771**
Iris	Acc	0.8933	0.8933	0.9067	0.9067	0.9067	**0.9600**
	AMI	0.7551	0.7465	0.8032	0.8032	0.8032	**0.8689**
	NMI	0.7582	0.7496	0.8057	0.8057	0.8057	**0.8705**
	ARI	0.7302	0.7294	0.7592	0.7592	0.7592	**0.8858**
	FMI	0.8208	0.8197	0.8407	0.8407	0.8407	**0.9234**
Wine	Acc	0.7022	0.6854	0.6966	**0.7416**	0.7135	0.6910
	AMI	0.4227	0.4106	0.4099	0.4181	0.4138	**0.4247**
	NMI	0.4288	0.4168	0.4161	0.4242	0.4199	**0.4308**
	ARI	0.3711	0.3539	0.3684	**0.4144**	0.3591	0.3910
	FMI	0.5835	0.5728	0.5821	**0.6127**	0.5762	0.6024
Sonar	Acc	0.5529	0.5529	0.5529	0.5625	0.5288	**0.6490**
	AMI	0.0053	0.0053	0.0031	0.0090	−0.0050	**0.0678**
	NMI	0.0088	0.0088	0.0081	0.0124	0.0000	**0.0713**
	ARI	0.0064	0.0064	0.0066	0.0109	−0.0013	**0.0845**
	FMI	0.5028	0.5028	**0.6510**	0.5056	0.6269	0.5845
Movement_libras	Acc	0.4583	–	0.4472	0.4361	**0.5139**	0.4917
	AMI	0.5583	–	0.5634	0.5400	0.6008	**0.6061**
	NMI	0.6112	–	0.6157	0.5949	0.6479	**0.6528**
	ARI	0.3204	–	0.3154	0.2989	**0.3957**	0.3647
	FMI	0.3689	–	0.3654	0.3500	**0.4426**	0.4144
Ionosphere	Acc	0.7123	0.7094	0.7179	0.6895	0.7236	**0.7350**
	AMI	0.1330	0.1280	**0.1368**	0.0873	0.1142	0.1319
	NMI	0.1349	0.1299	**0.1386**	0.0893	0.1161	0.1339
	ARI	0.1776	0.1727	0.1872	0.1388	0.1905	**0.2126**
	FMI	0.6053	0.6031	0.6108	0.5936	0.6343	**0.6413**
Ecoli	Acc	0.6012	0.4970	0.7649	0.6101	**0.8006**	0.7857
	AMI	0.5977	0.5322	**0.7074**	0.4166	0.6607	0.6772
	NMI	0.6144	0.5514	**0.7193**	0.4394	0.6740	0.6911
	ARI	0.4276	0.3682	0.7449	0.4203	0.7302	**0.7626**
	FMI	0.5617	0.5118	0.8210	0.6423	0.8119	**0.8264**
Leuk72_3k	Acc	**0.9583**	**0.9583**	**0.9583**	**0.9583**	**0.9583**	**0.9583**
	AMI	**0.8558**	**0.8558**	0.8381	0.8555	0.8555	0.8555
	NMI	**0.8596**	**0.8596**	0.8424	0.8593	0.8593	0.8593
	ARI	0.8803	0.8803	0.8805	**0.8809**	**0.8809**	**0.8809**
	FMI	0.9197	0.9197	0.9199	**0.9205**	**0.9205**	**0.9205**
Compound	Acc	0.6566	0.6566	**0.8622**	0.6316	0.6441	0.7218
	AMI	0.7135	0.7044	0.8314	0.7539	0.7308	**0.8327**
	NMI	0.7192	0.7103	0.8353	0.7589	0.7362	**0.8363**
	ARI	0.5379	0.5357	**0.8030**	0.5116	0.5435	0.6346
	FMI	0.6422	0.6404	**0.8616**	0.6251	0.6473	0.7223

culation of the CFSFDP_1 algorithm is improved to (9) to optimize the assignment strategy. 3) CFSFDP_3. It introduces the cluster fusion algorithm based on the CFSFDP_2 algorithm. The parameters of the three variants of the algorithm take the same values. Since the experimental results were similar on the five evaluation metrics, we show the results of the ablation experiment with Acc as a representative. The Acc of the CFSFDP algorithm and the three variants of the algorithm are shown in Table 4. The experimental res-

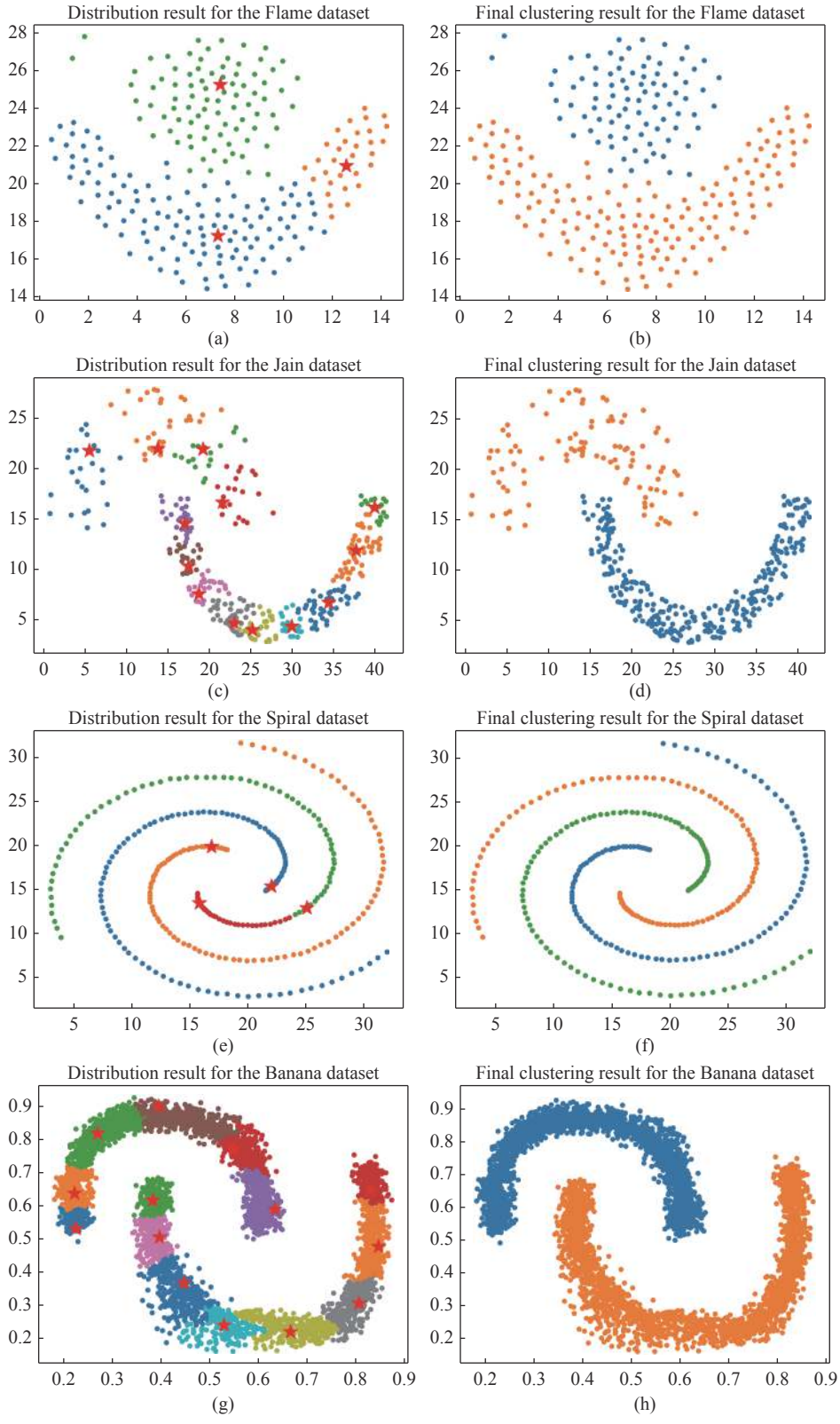

Fig. 6. Cluster fusion results of manifold dataset. (a) and (b) Flame dataset; (c) and (d) Jain dataset; (e) and (f) Spiral dataset; (g) and (h) Banana dataset.

ults show that the CFSFDP_1 algorithm further improves the accuracy of some data sets and avoids the artificial value of cutoff distance d_c, which verifies the effectiveness of the improved sample density metric. The CFSFDP_2 algorithm improves the clustering accuracy on some data sets again based on the CFSFDP_1 algorithm, which verifies the effectiveness of the improved remaining sample assignment method

by combining the nearest neighbor samples. The CFSFDP_3 algorithm further improves the accuracy on only two manifold data sets, Jain and Banana. However, it also meets the target expectation considering that the purpose of introducing the cluster fusion algorithm is to prevent density peaks misselection and improve the algorithm's effectiveness on manifold data sets.

Table 4. Acc of CFSFDP and three variant algorithms

Data sets	CFSFDP	CFSFDP_1	CFSFDP_2	CFSFDP_3
Flame	**1.0000**	**1.0000**	**1.0000**	**1.0000**
Aggregation	0.7513	**0.9962**	**0.9962**	**0.9962**
Spiral	0.9487	0.9679	**1.0000**	**1.0000**
Jain	**1.0000**	0.5657	0.5657	**1.0000**
4k2_far	**1.0000**	**1.0000**	**1.0000**	**1.0000**
R15	**0.9967**	**0.9967**	**0.9967**	**0.9967**
D31	0.9681	**0.9710**	**0.9710**	**0.9710**
Banana	0.6086	0.6306	0.6333	**1.0000**
Spiral3D	0.3648	**0.4025**	**0.4025**	**0.4025**
Iris	0.9067	0.9133	**0.9600**	**0.9600**
Wine	**0.7416**	0.5281	0.6910	0.6910
Sonar	0.5625	0.6202	**0.6490**	**0.6490**
Movement_libras	0.4361	0.4861	**0.4917**	**0.4917**
Ionosphere	0.6895	**0.7350**	**0.7350**	**0.7350**
Ecoli	0.6101	**0.8214**	0.7857	0.7857
Leuk72_3k	**0.9583**	**0.9583**	**0.9583**	**0.9583**
Compound	0.6316	0.6266	**0.7218**	**0.7218**

VI. Conclusions

In this paper, we propose reverse-nearest-neighbor-based clustering by fast search and find of density peaks (RNN-CFSFDP) by optimizing the CFSFDP algorithm. The RNN-CFSFDP algorithm redesigns and unifies the metric of sample density on data sets of different sizes by combining the reverse nearest neighbors of samples. Therefore, the ρ_i can reflect the local density of sample $\boldsymbol{x}_i$ more objectively and avoid the artificial value of cutoff distance d_c. In addition, the RNN-CFSFDP algorithm also improves the assignment strategy by using the advantage of nearest neighbor samples to detect the local distribution of samples. The method proposed effectively reduces the problem that the domino effect is prone to occur in the CFSFDP algorithm for manifold data sets. Finally, we propose a cluster fusion algorithm to solve when the cluster center is manually selected, the sparse cluster may not be selected to the density peak, and it may lead to cluster center wrong selection. Experimental results on publicly available synthetic data sets and UCI real-world data sets show that the RNN-CFSFDP algorithm can effectively reduce subjective intervention. In most cases, the algorithm outperforms or is at least comparable to comparative methods in terms of clustering performance. The RNN-CFSFDP algorithm is applicable to data sets of any dimension and size and is particularly robust to cluster shape and density differences.

However, the K for the reverse nearest neighbor in the RNN-CFSFDP algorithm still cannot be selected adaptively. For further research, we will focus on two points. One is to explore the local neighbor-based clustering algorithm and find a way to automatically determine the value of K to simplify the algorithm's parameters. The other is to combine the algorithm's advantages with those of other clustering algorithms.

References

[1] Z. W. Gu, P. Li, X. Lang, *et al.*, "A multi-granularity density peak clustering algorithm based on variational mode decomposition," *Chinese Journal of Electronics*, vol.30, no.4, pp.658–668, 2021.

[2] Y. Oktar and M. Turkan, "A review of sparsity-based clustering methods," *Signal Processing*, vol.148, pp.20–30, 2018.

[3] M. S. Chang, L. H. Chen, L. J. Hung, *et al.*, "Exact algorithms for problems related to the densest k-set problem," *Information Processing Letters*, vol.114, no.9, pp.510–513, 2014.

[4] Y. Shi, Z. S. Chen, Z. Q. Qi, *et al.*, "A novel clustering-based image segmentation via density peaks algorithm with mid-level feature," *Neural Computing and Applications*, vol.28, no.S1, pp.29–39, 2017.

[5] Y. Wang, "Survey on deep multi-modal data analytics: collaboration, rivalry, and fusion," *ACM Transactions on Multimedia Computing, Communications, and Applications*, vol.17, no.1s, article no.10, 2021.

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

[7] J. C. Bezdek, R. Ehrlich, and W. Full, "FCM: the fuzzy c-means clustering algorithm," *Computers & Geosciences*, vol.10, no.2-3, pp.191–203, 1984.

[8] W. Wang, J. Yang, and R. R. Muntz, "STING: A statistical information grid approach to spatial data mining," in *Proceedings of the 23rd International Conference on Very Large Data Bases*, San Francisco, CA, United States, pp.186–195, 1997.

[9] S. Guha, R. Rastogi, and K. Shim, "Cure: An efficient clustering algorithm for large databases," *Information Systems*, vol.26, no.1, pp.35–58, 2001.

[10] X. F. Wang and Y. F. Xu, "Fast clustering using adaptive density peak detection," *Statistical Methods in Medical Research*, vol.26, no.6, pp.2800–2811, 2017.

[11] J. Xu, G. Y. Wang, and W. H. Deng, "DenPEHC: Density peak based efficient hierarchical clustering," *Information Sciences*, vol.373, pp.200–218, 2016.

[12] M. Ester, H. P. Kriegel, J. Sander, *et al.*, "A density-based algorithm for discovering clusters in large spatial databases with noise," in *Proceedings of the Second International Conference on Knowledge Discovery and Data Mining*, Portland, OR, USA, pp.226–231, 1996.

[13] B. X. Zhao, S. L. Wang, and C. L. Liu, "STATE: A clustering algorithm focusing on edges instead of centers," *Chinese Journal of Electronics*, vol.30, no.5, pp.902–908, 2021.

[14] Y. Wang, W. J. Zhang, L. Wu, *et al.*, "Iterative views agreement: An iterative low-rank based structured optimiza-

tion method to multi-view spectral clustering," in *Proceedings of the 25th International Joint Conference on Artificial Intelligence*, New York, NY, USA, pp.2153–2159, 2016.

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

[16] J. Y. Xie, H. C. Gao, and W. X. Xie, "K-nearest neighbors optimized clustering algorithm by fast search and finding the density peaks of a dataset," *Scientia Sinica:Informationis*, vol.46, no.2, pp.258–280, 2016. (in Chinese)

[17] T. F. Xu and J. H. Jiang, "A Graph Adaptive Density Peaks Clustering algorithm for automatic centroid selection and effective aggregation," *Expert Systems with Applications*, vol.195, article no.116539, 2022.

[18] X. Xu, S. F. Ding, H. Xu, *et al.*, "A feasible density peaks clustering algorithm with a merging strategy," *Soft Computing*, vol.23, no.13, pp.5171–5183, 2019.

[19] J. Y. Xie, H. C. Gao, W. X. Xie, *et al.*, "Robust clustering by detecting density peaks and assigning points based on fuzzy weighted K-nearest neighbors," *Information Sciences*, vol.354, pp.19–40, 2016.

[20] J. H. Jiang, Y. J. Chen, X. Q. Meng, *et al.*, "A novel density peaks clustering algorithm based on k nearest neighbors for improving assignment process," *Physica A:Statistical Mechanics and its Applications*, vol.523, pp.702–713, 2019.

[21] R. Zhang, T. Du, S. N. Qu, *et al.*, "Adaptive density-based clustering algorithm with shared KNN conflict game," *Information Sciences*, vol.565, pp.344–369, 2021.

[22] Y. H. Liu, Z. M. Ma, and F. Yu, "Adaptive density peak clustering based on K-nearest neighbors with aggregating strategy," *Knowledge-Based Systems*, vol.133, pp.208–220, 2017.

[23] L. Bai, X. Q. Cheng, J. Y. Liang, *et al.*, "Fast density clustering strategies based on the k-means algorithm," *Pattern Recognition*, vol.71, pp.375–386, 2017.

[24] A. Bryant and K. Cios, "RNN-DBSCAN: a density-based clustering algorithm using reverse nearest neighbor density estimates," *IEEE Transactions on Knowledge and Data Engineering*, vol.30, no.6, pp.1109–1121, 2018.

[25] M. Chen, L. J. Li, B. Wang, *et al.*, "Effectively clustering by finding density backbone based-on kNN," *Pattern Recognition*, vol.60, pp.486–498, 2016.

[26] Y. W. Chen, L. L. Shen, C. M. Zhong, *et al.*, "Survey on density peak clustering algorithm," *Journal of Computer Research and Development*, vol.57, no.2, pp.378–394, 2020. (in Chinese)

[27] L. M. Fu and E. Medico, "FLAME, A novel fuzzy clustering method for the analysis of DNA microarray data," *BMC Bioinformatics*, vol.8, article no.3, 2007.

[28] H. Chang and D. Y. Yeung, "Robust path-based spectral clustering," *Pattern Recognition*, vol.41, no.1, pp.191–203, 2008.

[29] A. Gionis, H. Mannila, and P. Tsaparas, "Clustering aggregation," *ACM Transactions on Knowledge Discovery from Data*, vol.1, no.1, pp.4–es, 2007.

[30] A. K. Jain and M. H. C. Law, "Data clustering: A user's dilemma," in *Proceedings of the 1st International Conference on Pattern Recognition and Machine Intelligence*, Kolkata, India, pp.1–10, 2005.

[31] C. J. Veenman, M. J. T. Reinders, and E. Backer, "A maximum variance cluster algorithm," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol.24, no.9, pp.1273–1280, 2002.

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

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

[34] S. F. Ding, M. J. Du, T. F. Sun, *et al.*, "An entropy-based density peaks clustering algorithm for mixed type data employing fuzzy neighborhood," *Knowledge-Based Systems*, vol.133, pp.294–313, 2017.

[35] L. Hubert and P. Arabie, "Comparing partitions," *Journal of Classification*, vol.2, no.1, pp.193–218, 1985.

[36] R. Liu, H. Wang, and X. M. Yu, "Shared-nearest-neighbor-based clustering by fast search and find of density peaks," *Information Sciences*, vol.450, pp.200–226, 2018.

ZHANG Chunhao received the B.S. degree in information and computational science from Hebei University of Engineering in 2019. He is now an M.S. candidate of Hebei Normal University. His research interests include machine learning and data mining.
(Email: zhangchunhao_hebtu@163.com)

XIE Bin (corresponding author) received the B.S. degree in computational mathematics from Jilin University in 1998. He received the M.S. degree in computational mathematics from Jilin University in 2004. He received the Ph.D. degree in applied mathematics from Hebei Normal University in 2011. His research interests include granular computing, machine learning and approximate reasoning.
(Email: xiebin_hebtu@126.com)

ZHANG Yiran received the B.E. degree in software engineering from Hebei Normal University in 2019. She is now an M.S. candidate of Hebei Normal University. Her research interests include machine learning and data mining.
(Email: zhangyiran19@163.com)