Hindawi
Complexity
Volume 2020, Article ID 8864239, 11 pages
https://doi.org/10.1155/2020/8864239

*Research Article*

# Improved Density Peaks Clustering Based on Natural Neighbor Expanded Group

**Lin Ding**,[1] **Weihong Xu**,[1,2] **and Yuantao Chen**[1]

[1] *School of Computer and Communication Engineering and Hunan Provincial Key Laboratory of Intelligent Processing of Big Data on Transportation, Changsha University of Science and Technology, Changsha, Hunan 410114, China*
[2] *School of Computer Science and Engineering, Nanjing University of Science and Technology, Nanjing, Jiangsu 210094, China*

Correspondence should be addressed to Yuantao Chen; chenyt@csust.edu.cn

Received 23 August 2020; Revised 13 September 2020; Accepted 1 October 2020; Published 23 October 2020

Academic Editor: Shi Cheng

Density peaks clustering (DPC) is an advanced clustering technique due to its multiple advantages of efficiently determining cluster centers, fewer arguments, no iterations, no border noise, etc. However, it does suffer from the following defects: (1) difficult to determine a suitable value of its crucial cutoff distance parameter, (2) the local density metric is too simple to find out the proper center(s) of the sparse cluster(s), and (3) it is not robust that parts of prominent density peaks are remotely assigned. This paper proposes improved density peaks clustering based on natural neighbor expanded group (DPC-NNEG). The cores of the proposed algorithm contain two parts: (1) define natural neighbor expanded (NNE) and natural neighbor expanded group (NNEG) and (2) divide all NNEGs into a goal number of sets as the final clustering result, according to the closeness degree of NNEGs. At the same time, the paper provides the measurement of the closeness degree. We compared the state of the art with our proposal in public datasets, including several complex and real datasets. Experiments show the effectiveness and robustness of the proposed algorithm.

## 1. Introduction

Clustering algorithm, usually as unsupervised learning, is a type of fundamental technique of machine learning [1]. It aims to divide a dataset into several subsets, which are also called categories, clusters, groups, etc, according to similarity, dissimilarity, or distance of samples. Hence, unlike supervised learning [2–18], clustering methods implement classification tasks without any prior knowledge and have been applied to image processing, pattern recognition, bioinformatics, data mining, the Internet of things, and other fields.

Due to flexibility and validity, various clustering algorithms have been proposed one after another. Jain classified these methods into partitioning-based, model-based, hierarchical-based, grid-based, and density-based approaches [19]. Partitioning methods aim for grouping the dataset into a preset number of clusters via an iterative process. *K*-means [20, 21] and Fuzzy *c*-means [22, 23] are two famous partitioning-based clusterings. Although they are simple to understand and easy to implement, *K*-means is extremely sensitive to outliers and the selection of the initial cluster centers; besides, Fuzzy *c*-means approaches suffer from initial partition dependence [1]. Model-based clustering methods require one or more appropriate probability models to represent the dataset and often use the expectation-maximization approach to maximize the likelihood function [24]. Hierarchical-based approaches [25–28] partition the dataset into several categories using two opposite ways: top-down or bottom-up approach [23]. The first one considers the whole dataset as a cluster and split it into a suitable number of subclusters. Another regards each sample as a cluster and then merging these atomic clusters into more and more massive clusters. However, the effectiveness of hierarchical clustering algorithms depends on the type of distance measurement chosen for the clusters. Grid-based [29] and density-based [30, 31] approaches automatically determine the number of categories using suitable and preset parameters such as epsilon, min-pts, or others. While it is necessary to take a mass of argument adjustments to obtain optimal clustering results, these two types of algorithms generate noise at the cluster borders.

To overcome the above shortcomings, recently, density peaks clustering [32] is proposed and based on the assumption that cluster centers are relatively denser and are far from each other. Using a suitable value of cutoff distance (namely, dc, the only parameter of DPC), this approach manually selects the appropriate center of each cluster from a decision graph. It then assigns each of the remaining elements to the nearest denser point (NDP) that is the nearest one of neighbors possessing bigger density than the assigned sample. It has many advantages, including higher efficiency in finding cluster centers, fewer parameters, no iterations, and no noise around the cluster border. However, the algorithm is still affected by the following defects:

(1) It is challenging to determine suitable dc. It must also be mentioned that the original DPC algorithm does not cover a reliable and specific method to ensure proper dc. Besides, this was demonstrated in several studies [33, 34] that DPC is sensitive to its parameter, and even when being normalized or using the relative percentage method, a small change in dc will still cause a conspicuous fluctuation in the result.

(2) The formula of local density is too simple to find out suitable center(s) of the sparse cluster(s) and is only useful in datasets with balanced density [33]. As shown in Figure 1(a), the Jain dataset has two clusters: the upper one is sparse and the lower one is denser. However, DPC overlooks the center of the upper cluster, instead of a prominent density peak of the lower cluster.

(3) Its assignment strategy is not robust [35]. Each point is assigned to its NDP, which results in some prominent density peaks (PDP) that are relatively bigger on density and $\delta_i$ value but not cluster centers are mistakenly attributed to a denser superordinate but are far away from each other. Accordingly, the subordinates of the incorrect-assigned PDP are portioned to an incorrect group. Figure 1(b) shows that we manually modify the center to the densest point of the upper cluster. However, the prominent local peak of the top cluster is assigned to its NDP belonging to the lower cluster, which leads to the incorrect assignment of its subordinates. And there is a distinct gap between the assignment path.

To improve the performance of DPC and inspired by the idea of natural neighbor (NN) [36], we propose an improved density peaks clustering based on natural neighbor expanded group. The main innovations and improvements in our algorithm are as follows:

(1) Define natural neighbor expanded and natural neighbor expanded group based on the well-known $K$-nearest neighbor method and its optimal version named natural neighbor. The concept of natural neighbor expanded is to absorb those close neighbors overlooked by the NN method. And NNEG is able to overcome the shortcoming of the remote assignment of PDP and mine the potential structure of data.

(2) Provide a density metric formula based on NNE. With the aid of NNE, the new measurement adaptively calculates the local density for each sample without any arguments, unlike one of the original DPC.

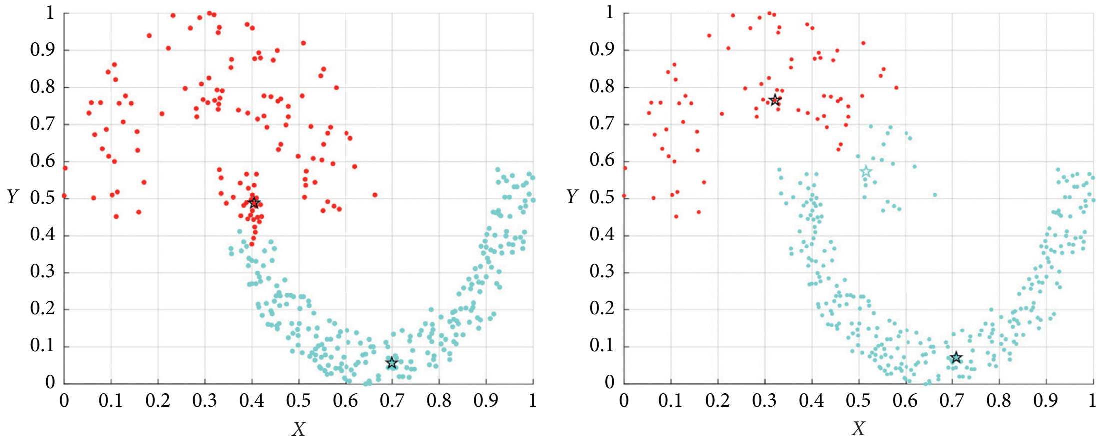
(3) Propose the measurement of the closeness degree of NNEGs that based on the mutual and pairwise neighbors which belonged to different NNEGs. Due to its application, all NNEGs are divided into the goal number of sets as the final clustering result.

(4) The time complexity is $O(Kn\log n)$, where $K$ is a constant, while the time complexity of all of the optimization algorithms and DPC is $O(n^2)$ [34].

The remainder of this paper comprises four sections. Section 2 describes the related works. Section 3 represents the DPC, NN method, and details of our algorithm. Section 4 presents the clustering results on our proposal and related works. In Section 5, we have a summary of the contributions and features of this paper.

## 2. Related Works

To improve the performance of the DPC algorithm, scholars proposed many optimization methods, as shown in Figure 2. Xie et al. modified the density metric formula using the $K$-nearest neighbor (KNN), which used the number of the nearest neighbors to replace dc. Besides, they devised an entirely new assignment scheme based on fuzzy weighted $K$-nearest neighbors (FKNN-DPC) [33]. Furthermore, this method is easier to determine the suitable value of parameter. Lotfi et al. proposed a technique called IDPC [37]. The algorithm sorts samples using the local density and then apportions the labels of centers to their KNN to develop cluster cores. Finally, IDPC implements a specific propagation strategy to attach the remaining points with labels. Guo et al. capitalized on the linear regression method to fit the decision values of DPC with a preset proper dc required (DPC-LRA) and then choose the instances above the fitting function as the centers [38]. Ding et al. proposed an algorithm based on the generalized extreme value distribution (GEV) to fit the DPC decision values in the descending order (DPC-GVE). To reduce the time complexity, they also represented a substitution method using Chebyshev inequality (DPC-CI) [39]. Ni et al. presented the definitions of density gap and the density path, as well as a new threshold [35]. Instead of the decision graph of DPC, the proper value of dc is determined by manually observing a summary graph incorporating the density gaps calculated by different dc. The method, named PPC, is able to reduce obviously the difficulty on threshold determination. Jiang et al. provided a novel density peaks clustering algorithm based on $K$-nearest neighbors (DPC-KNN) to overcome the issue of the assignment [40]. In this method, there are two sets for each sample $i$: the first one is $S_i$, which is composed of sample $i$ and its KNN, while the second is $H_i$, which covers the data points possessing



(a)

(b)

Figure 1: Clustering results of DPC on the Jain dataset. The diverse colors present different clusters, and the stars mark the cluster centers and prominent local peaks. (a) Clustering results of DPC on Jain. (b) Clustering results of DPC with modified center.

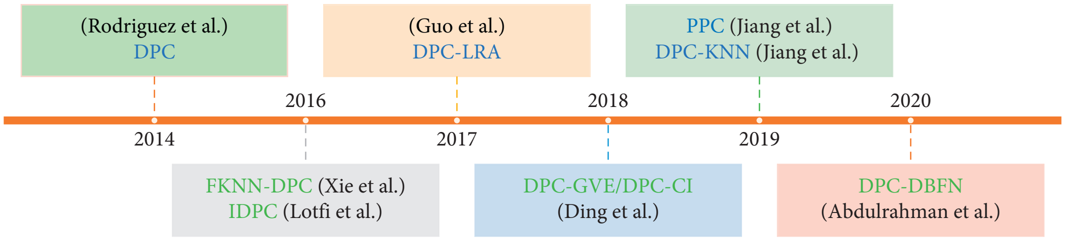


Figure 2: Improvement methods of DPC.

higher densities than sample $i$ in the whole dataset. The cluster centers are determined via the decision graph of DPC, DPC-KNN assigns each remaindering sample to an element of $H_i$, who has the smallest distance from any member of $S_i$ to any member of $H_i$. Lotfi et al. improve DPC using density backbone and fuzzy neighborhood (DPC-DBFN). They use a fuzzy kernel for improving the separability of clusters. DPC-DBFN uses a density-based KNN graph for labeling backbones and effectively assigns correct category labels to samples around the group borders to effectively cluster data with various shapes and densities [34].

However, FKNN-DPC, IDPC, DPC-KNN, PPC, and DPC-DBFN require manual operations. And a preset dc is necessary for DPC-LRA, DPC-GVE, and DPC-CI. Moreover, DPC and these algorithms require the time complexity of $O(n^2)$ [34].

## 3. Methods

This section aims to present the short versions of the original DPC algorithm and NN method and show a detailed description of our method.

*3.1. The Original DPC Algorithm.* DPC is the basis on which cluster centers are relatively denser and are distant from each other. For a given dataset $X = \{x_1, x_2, \ldots, x_n\}$, where $x_i = \{x_{i1}, x_{i2}, \ldots, x_{im}\}, i = 1, 2, \ldots, n$, cluster centers are manually picked from the decision graph, which is two-dimensional with $\delta_i$ as the ordinate and the local density as the abscissa. Local density is to measure the neighbor number and distances of each sample in its neighborhood, which is a crucial concept of DPC. The ordinate $\delta_i$ is the distance between the sample $i$ and its nearest denser point. Since the centers have relative lager density, each of them must be far away from their NDP, namely, has an enormous value of $\delta_i$. In the two-dimensional coordinate system, cluster centers simultaneously possess big values of $\delta_i$ and local density and appear in the upper right corner of the graph. To measure the local density of each element, the author provides two formulae expressed as equations (1) and (2). $\delta_i$ is calculated by equation (3):

$$\rho_i = \sum_j \aleph\left(d_{ij} - dc\right),$$

$$\aleph(\cdot) = \begin{cases} 1, & \cdot < 0, \\ 0, & \cdot \geq 0, \end{cases} \tag{1}$$

$$\rho_i = \sum_j \exp\left(-\left(\frac{d_{ij}}{dc}\right)^2\right), \tag{2}$$

where $d_{ij}$ is the distance between pairwise elements $i$ and $j$, $dc$ is the cutoff distance, the only argument of DPC. Therefore, The DPC algorithm inherits a defect, where Gaussian kernel is sensitive to bandwidth:

$$\delta_i = \begin{cases} \min_{j:\, \rho_i < \rho_j}\left(\mathrm{d}_{ij}\right), & \text{if } \exists j \text{ s.t. } \rho_i < \rho_j, \\ \max_j\left(\mathrm{d}_{ij}\right), & \text{otherwise,} \end{cases} \tag{3}$$

As shown in equation (3), $\delta_i$ is the minimum distance between elements $i$ and $j$ whose density is higher than $i$. For $i$ with the highest density, its $\delta_i$ is the maximum distance between $i$ and $j$. After the cluster centers have been found, each remaining point is assigned to the same cluster as its nearest neighbor of higher density.

*3.2. Natural Neighbor Method.* $K$-nearest neighbor is a popular method in machine learning to complete the tasks of classification and clustering. However, the crucial argument $K$ is preset manually. And natural neighbor is an adaptive method to find the relative near neighbors of each sample. The basic idea of NN is that samples of the dense regions have more neighbors; data points of the sparse area have relatively fewer neighbors; the outliers only have a few or no natural neighbors.

In the dataset $X$, the authors assume that $s_{ij}$ is the similarity between two points $x_i$ and $x_j$. With the help of comparing the similarity, let find KNN$(x_i, n)$ denote the function of KNN searching which returns the $r^{\text{th}}$ nearest neighbor of the point $x_i$, $\mathrm{KNN}_r(x_i)$ is a subset of $X$, and it is defined as follows:

$$\mathrm{KNN}_r(x_i) = \bigcup_{n=1}^{r} \{\text{find KNN}(x_i, n)\}. \tag{4}$$

*Definition 1.* (natural neighbor). Natural neighbor of $x_i$ is defined as

$$x_j \in \mathrm{NN}(x_i) \Leftrightarrow \left(x_i \in \mathrm{KNN}_\lambda\left(x_j\right) \wedge x_j \in \mathrm{KNN}_\lambda(x_i)\right). \tag{5}$$

*Definition 2.* (natural neighbor eigenvalue). When the algorithm reaches the Stable Searching State, Natural Neighbor Eigenvalue (NaNE) $\lambda$ is equal to the searching round $r$:

$$\lambda \triangleq r_{r\in N}\left\{r \mid (\forall x_i)\left(\exists x_j\right)(r \in N) \wedge \left(x_i \neq x_j\right) \longrightarrow \left(x_i \in \mathrm{KNN}_r\left(x_j\right) \wedge x_j \in \mathrm{KNN}_r(x_i)\right)\right\}. \tag{6}$$

*3.3. The Proposed Method.* In this section, the improved density peaks clustering based on natural neighbor expanded group is presented. Our method includes three major steps, including (1) calculating the local density of each sample according to the formula proposed, (2) determining natural neighbor expanded groups, and (3) grouping NNEGs into several sets as the final clustering result. The details of these steps are described in the remaining part of this section. To realize the above processing, we define the concept of natural neighbor expanded and then provide a straightforward but useful formula for local density. Besides, the definition of the natural neighbor expanded group is to reveal the structure of the dataset and divide the dataset into several local groups. For ensuring the grouping of NNEGs accuracy, we propose a measurement of closeness degree. And more details are presented in the rest content of this section.

*3.3.1. Basic Concepts.* The NN method only considers the relationship of mutual neighbors and overlooks the impact of distance between samples. And to fit the density metric and the searching of density peaks, we propose the concept of Natural Neighbor Expanded.

*Definition 3.* (natural neighbor expanded). Natural Neighbor Expanded is defined as the following equation:

$$\mathrm{NNE}\left(x_i\right) = \mathrm{KNN}_{2K_i}\left(x_i\right), \tag{7}$$

where we assume that the number of NN of $x_i$ is $|\mathrm{NN}_i|$ and the $|\mathrm{NN}_i|^{\text{th}}$ $\mathrm{NN}(x_i)$ is the $K_i^{\text{th}}$ $\mathrm{KNN}_r(x_i)$. Hence, $K_i < = r$. As shown in Figure 3, sample 1 is not the NN of sample 8, since it does not belong to the $\mathrm{KNN}_6(8)$. However, sample 1 is closer to sample 8 than 14. Hence, for calculating the density more wholly and accurately, we expand the natural neighborhood of sample 8 to include samples 1, 2, and 7.

Natural Neighbor is the set of close neighbors. Still, as shown in equation (2), the local density formula measures not only the close neighbors whose distances to sample $i$ are smaller than dc but also the rest samples of whole datasets. In the latter part, the distance to sample $i$ being approximate to dc and the corresponding sample also impacts the density of $i$. Therefore, $2K_i$ of equation (7) is to cover the more secondary-adjacent samples beside the close neighbors. And the new local density formula based on NNE is shown as

$$\rho_i = \sum_{j\in \mathrm{NNE}\left(x_i\right)} \frac{\max(\mathrm{distNNE}) - \mathrm{d}_{ij}}{\max(\mathrm{distNNE}) - \min(\mathrm{distNNE})}, \tag{8}$$

where $\mathrm{NNE} = \cup_{i=1}^{n}\mathrm{NNE}(x_i)$, $\mathrm{distNNE} = \cup_{i=1}^{n}\mathrm{distNNE}(x_i)$, and $\mathrm{distNNE}(x_i)$ is the set of the distances of $x_i$ to all of the elements in $\mathrm{NNE}(x_i)$. Inspired by the famous $K$-means method, equation (8) considers each point as core and calculates the sum of distances of it to its NNE. And the smaller the distance sum is, the more likely it is to be the local center.

Equation (2) maps the distances to similarities using the Gaussian kernel and calculates the accumulation sum of

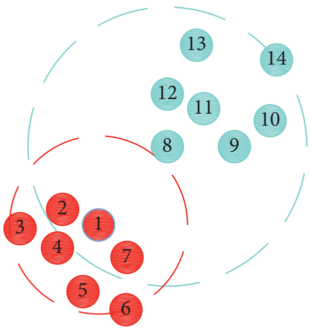


Figure 3: A schematic diagram with parts of samples and its $r$ equals 6. Sample 1 has six NNEs, including samples 2, 3, 4, 5, 6, and 7; sample 8 has six NNEs, including samples 9, 10, 11, 12, 13, and 14.

similarities linking to $x_i$ as $\rho_i$. Hence, equation (2) based on Gaussian kernel can resist the interference of outliers that possess vast distances to $x_i$. However, the equation covers too many negligible samples that have the distances to sample $i$ much bigger than dc because their contribution to density is tiny through the mapping of the Gaussian kernel. Moreover, it brings the original DPC to the time complexity of $O(n^2)$.

In contrast, our formula only considers NNE. It, therefore, also gets rid of the passive impact of outliers since they usually are distant from its nearest point and are not in any NNE(s) of other(s), at the same time, reduce the computational complexity. And unlike the Gaussian kernel mapping, equation (8) retains the original information of data, does not require any arguments, and avoids the sensitivity caused by dc.

*Definition 4.* (natural neighbor expanded group). Natural Neighbor Expanded Group consists of a prominent density peak and its subordinates.

In our method, each point is assigned to the nearest denser point of its NNE. The assignment process is stored in a list: the index numbers represent the samples in the given dataset, respectively; each unit stores the index number of its superordinate one, and if the density of a sample is bigger than all of its NDP, the related unit saves 0. Namely, zero samples are prominent density peaks. The assignment divides the dataset into several NNEGs, adaptively.

Essentially, NNEGs reveal the potential structure of the dataset analyzed and are relatively tighter subcluster and local groups in the cluster of the Ground Truth. Due to the application of NNEG, each sample only points to a neighbor, and our method could avoid the long-distance assignment of the PDP.

As shown in Figure 4, after NNEGs are determined, our method only needs to merge such local groups into the goal number of clusters and hence remove the operation of the center selection from the decision graph, which overcomes the mentioned issue of the density metric of DPC. To clarify the close relationship between NNEGs, we proposed the concept of the adjacent group graph.

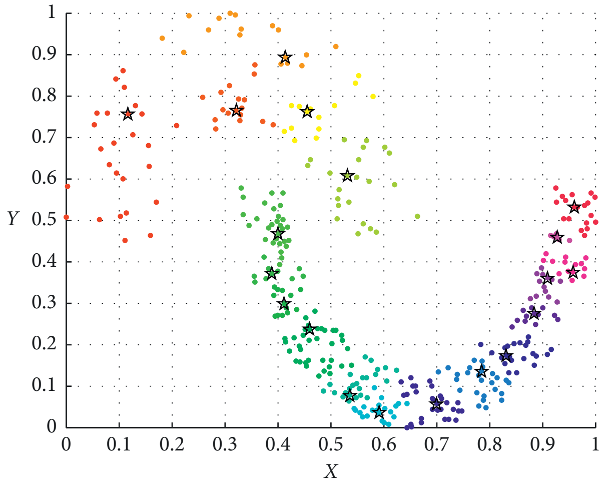


Figure 4: NNEGs of the Jain dataset. The diverse colors present different NNEGs, and the stars mark the zero samples.

*Definition 5.* (adjacent group graph). $\mathrm{AGG} = G(V, E)$, where $V = \{v_1, v_2, \ldots, v_k\}$ is a set of NNEGs, $E = \{E(v_t, v_\tau) | t \neq \tau,\ v_t,\ v_\tau \in V\}$, and $E(v_t, v_\tau)$ is a set of edges linked to NNEGs $v_t$ and $v_\tau$, and subject to

$$E(v_t, v_\tau) = \{e(x_i, x_j) | (x_i \in (v_t \wedge \mathrm{NNE}(x_j))) \wedge (x_j \in (v_\tau \wedge \mathrm{NNE}(x_i)))\}. \tag{9}$$

Adjacent Group Graph usually is a multigraph, since there could be several $e(x_i, x_j)$ between $v_t$ and $v_\tau$. And the more the edges are, the closer the two groups are. Obviously, in Figure 4, there are no edges between the upper and the lower clusters. Moreover, the degree of closeness (DC) of the neighboring pairwise NNEGs is calculated by

$$\mathrm{DC}(v_t, v_\tau) = \sum_{e(x_i, x_j) \in E(v_t, v_\tau)} w_i w_j \frac{\max(\mathrm{distNNE}) - d_{ij}}{\max(\mathrm{distNNE}) - \min(\mathrm{distNNE})}, \tag{10}$$

where $w_i = (|\mathrm{NNE}(x_i) \cap v_t|)/(|\mathrm{NNE}(x_i)|)$ and $w_j = (|\mathrm{NNE}(x_j) \cap v_\tau|)/(|\mathrm{NNE}(x_j)|)$. As shown in equation (10), the formula of closeness degree is constituted with two parts: the weight and the similarity normalized. It is based on an assumption where the more compact the endpoints and their respective NNEGs are, the more reliable the edge is. $w_i$ represents the compactness between the sample $x_i$ and the group $v_t$, viz., the bigger number of intersected elements of $\mathrm{NNEG}(x_i)$ and $v_t$ means the relationship between them is intenser. To ensure $w_i \in [0, 1]$, the number of the elements intersected divided by $|\mathrm{NNE}(x_i)|$.

### 3.3.2. *The Specific Processing*

Inputs: dataset $X$, the goal number of clusters.

Output: the clustering result.

Step 1: Create a *k-d* tree. Search NNE for each sample using the *k-d* tree.

Step 2: Calculate local density according to equation (8).

Step 3: Determine NNEG according to Definition 4.

Step 4: Generate the Adjacent Group Graph as in Definition 5, and find all edges of each pairwise NNEGs as equation (9).

Step 5: Calculate the degree of closeness, according to equation (10).

Step 6: Break up the original cluster containing all NNEGs into the goal number of sets, according to the closeness degree.

To clarify Step 6 in detail, we present an example in Table 1. As shown in Table 1 (A), there are five NNEGs in a dataset. And the closeness degrees of adjacent pairwise NNEGs are recorded. Assume the goal number is 2. Our method considers the whole dataset as a cluster, since $\mathrm{DC}(v_1, v_2), \mathrm{DC}(v_2, v_3), \mathrm{DC}(v_3, v_4), \mathrm{DC}(v_3, v_5), \mathrm{DC}(v_4, v_5) > 0$. We force the minimum $\mathrm{DC}(v_2, v_3) = 0$ as shown in Table 1 (B), which means those NNEGs are split into two parts: $\{v_1, v_2\}$ and $\{v_3, v_4, v_5\}$, i.e., split is a for-loop operation which let the minimum $\mathrm{DC} = 0$ until the cluster number equals to the goal one.

And more details are as shown in the pseudocode. In the 6th line, AGG is a matrix where each row and each column correspond to one of NNEGs. In the 16th line, inspired by the Top-down hierarchical clustering, we consider the whole dataset as a cluster containing all NNEGs and break the weakest $E(v_t, v_\tau)$ in the AGG until the cluster number equals the goal, which corresponds to the process, Table 1 (A) and (B).

Table 1: An example of Step 6.

| | $v_1$ | $v_2$ | $v_3$ | $v_4$ | $v_5$ |
|---|---|---|---|---|---|
| $v_1$ | — | 4 | 0 | 0 | 0 |
| $v_2$ | 4 | — | 1 | 0 | 0 |
| $v_3$ | 0 | 1 | — | 3 | 5 |
| $v_4$ | 0 | 0 | 3 | — | 4 |
| $v_5$ | 0 | 0 | 5 | 4 | — |

(a)

| | $v_1$ | $v_2$ | $v_3$ | $v_4$ | $v_5$ |
|---|---|---|---|---|---|
| $v_1$ | — | 4 | 0 | 0 | 0 |
| $v_2$ | 4 | — | 0 | 0 | 0 |
| $v_3$ | 0 | 0 | — | 3 | 5 |
| $v_4$ | 0 | 0 | 3 | — | 4 |
| $v_5$ | 0 | 0 | 5 | 4 | — |

(b)

*3.3.3. Time Complexity Analyses.* This section aims to analyze the computational complexity of our method, and suppose that the number of total samples in a dataset is $n$, the number of NNEG is equal to $n_{\mathrm{NNEG}}$, the goal number of clusters is $G$, the NDP of sample $i$ is the $n\mathrm{NDP}_i^{th}$ neighbor, and the biggest $K_i$ equals $\widehat{K}$. (Algorithm 1).

The time complexity of creating a k-d tree is $O(n \log n)$ [41]. It is demonstrated that determining NN for all samples also requires the cost of $O(n \log n)$ [36]. And for finding NNE, we can record the $K_i$ in the processing of searching NN. Hence, the searching NNE of a sample only needs to $2K_i$ times search operation, and its whole complexity for all samples is less than $O(2\widehat{K}n)$. Our local density metric is based on NNE, and it is not necessary to generate a distance matrix and only needs to $2K_i$ times plus operations for each sample. Therefore, it is required with at most $O(2\widehat{K}n)$ for the time cost to calculate local densities of all instances. For each sample, the method takes $n\mathrm{NDP}_i$ time search to find its NDP via the k-d tree in the round $2K_i$, and $n\mathrm{NDP}_i < \; = 2K_i$. In the process of generating each NNEG, we store the labels of its prominent density peak to a list where the first unit is any unallocated instance, and the end is an assigned one or prominent density peak. And the operation of storing labels of all samples only needs to the time cost of $O(n)$. And the cost required is $O(2\widehat{K}n)$ on dividing a dataset into $G$ NNEGs. In equations (9) and (10), $e(x_i, x_j)$ is requested and determined via searching the NNE of each sample to find the neighbors having different labels. Thus, for all edges, it is equal to $O(2\widehat{K}n)$ for the magnitude of how many times the searching operation is performed. Furthermore, the time complexity of grouping in the last step must be less than $G$. Overall, we can conclude that the time complexity of the entire algorithm is $O(Kn \log n)$.

## 4. Results

In this section, several datasets are used to evaluate the performance of our method in comparison with some state-of-the-art techniques such as DPC-DBFN [34], DPC-KNN [40], IDPC [37], and FKNN-DPC [33]. The experiments are performed on a computer with a Windows 10, Intel (R) Core (TM) i7-8750H, 16 GB memory, and Matlab 2016b. The results represented are measured by several performance metrics, including Normalized Mutual Information (NMI) [42], Rand Index (RI) [43], and the Adjusted Rand Index (ARI) [44]. In this section, the similarity between points is measured using the Euclidean distance metric.

*4.1. Datasets.* In this paper, all tested datasets include three low-dimensional datasets and five high-dimensional datasets, which are public and from UCI. The two-dimensional datasets have different numbers of samples and different objective distributions. The DMI512 dataset containing 1024 elements with 512-dimensional features, which belonged to 16 Gaussian clusters sampled from a Gaussian distribution, is often used to test algorithm performance in high-dimensional space. Experiments of the four datasets, including Statlog (Shuttle), Abalone, Wine Quality, and Libras Movement, are applications of our method on Physical (the positioning of radiators in the Space Shuttle), Population Biology, Model Wine Preferences, and Hand Movement Recognition, respectively. And more details are presented in Table 2.

To reduce the influence of dimension weights and ensure the validity of the experimental comparison, we processed each dataset and normalized all dataset tested. The normalization formula is as follows:

$$x'_{ij} = \frac{x_{ij} - \min\left(x_j\right)}{\max\left(x_j\right) - \min\left(x_j\right)}, \tag{11}$$

```
Require: Dataset $X = \{x_1, x_2, \ldots, x_n\}$, the goal number of clusters $G$
Ensure: The result of clustering: $C = \{C_1, C_2, \ldots, C_n\}$
Create a $k$-$d$ tree;
Search the $k$-$d$ tree;
Determine NN according to [34], and record $K_i$, which means NNE$(x_i)$ determined;
Calculate local density $\rho_i$ according to equation (8).;
Assign each point to its NDP of its NNE to generate several NNEGs;
Create a matrix AGG = (|NNEG|, |NNEG|);
for $i = 1 : n$ do
    for $t = 1 : 2K_i$ do
        if the tth NNE and sample $i$ belongs to different NNEGs do
            Calculate the closeness degree of this edge, referring to equation (10);
            Add the DC of this edge to the corresponding unit of AGG;
        end if
    end for
end for
while the number of clusters does not equal $G$ do
    Store zero in the unit with the min value but greater than zero;
    Count the number of clusters;
end while
```

Algorithm 1: DPC-NNEG.

Table 2: Detailed information on tested datasets.

| Dataset | #Instance | #Attribute | #Cluster |
|---|---|---|---|
| Jain | 373 | 2 | 7 |
| Flame | 240 | 2 | 2 |
| Spiral | 300 | 2 | 3 |
| Statlog (shuttle) | 58000 | 9 | 7 |
| Abalone | 4177 | 7 | 28 |
| Wine quality | 4898 | 11 | 7 |
| DIM512 | 1024 | 512 | 16 |
| Libras movement | 360 | 90 | 15 |

where $x_{\mathrm{ij}}$ is the $j$th feature value of the $i$th sample, while $\max(x_j)$ and $\min(x_j)$ represent the maximum and minimum values of the $j$th feature, respectively.

*4.2. Evaluation Measures.* We tested our algorithm and several related works on the above datasets. For intuitive comparison, we chose RI, ARI, and NMI to measure the clustering results.

The RI formula is shown in

$$\mathrm{RI} = \frac{\mathrm{TP} + \mathrm{TN}}{C_n^2}, \tag{12}$$

where TP indicates true positive, TN indicates real negative, and the denominator $C_n^2$ is the total number of sample pairs in a dataset consisting of $n$ samples.

The ARI formula is shown in

$$\mathrm{ARI} = \frac{\mathrm{RI} - E[\mathrm{RI}]}{\mathrm{MAX}\{\mathrm{RI}\} - E[\mathrm{RI}]}, \tag{13}$$

where $E[\mathrm{RI}]$ represents the expectations of RI.

The NMI formula is shown in

$$\mathrm{NMI} = \frac{-2\mathrm{MI}(A, B)}{H(A) + H(B)}, \tag{14}$$

where $H(A) = \sum_{i=1}^{|A|} P(i)\log_2 P(i)$, $H(B) = \sum_{i=}1^{|B|} P(j)\log_2 P(j)$, $E[\mathrm{MI}(A, B)]$ represents the expectations of $\mathrm{MI}(A, B)$, and $\mathrm{MI}(A, B)$ is expressed as

$$\mathrm{MI}(A, B) = \sum_{i=1}^{|A|}\sum_{j=1}^{|B|} P(i, j)\log_2 \frac{P(i, j)}{P(i)P(j)}, \tag{15}$$

where $P(i) = |A_i|/n$, $P(j) = |B_j|/n$, $P(i, j) = |A_i \cap B_j|/n$, $A = \{A_i | i = 1, 2, \ldots, |A|\}$, and $B = \{B_j | j = 1, 2, \ldots, |B|\}$. $A$ and $B$ represent two allocation methods for a dataset containing $n$ elements, and $A_i$ and $B_j$ are clusters. In experimental verification, let $A$ and $B$ be the original labels and the clustering results of an algorithm, respectively. If the clustering results are as same as the real labels, the three metrics take the value of 1, and if the clustering results are entirely different from the labels, the values will be equal to 0.

*4.3. Results.* This section aims to show the detailed clustering results and evaluate the performance of different clustering algorithms on the various datasets. Tables 3–5 compare the performance of our method with DPC-DBFN, DPC-KNN, IDPC, and FKNN-DPC in terms of NMI, RI, and ARI measures, respectively. All these methods are using the KNN method, and the number of nearest neighbors ($K$) can be set from 1 to $n$. In these tables, the numbers in the parenthesis are the value of $K$, where the corresponding algorithm obtains the results represented, and boldface marks the best results.

The Jain dataset has 373 points and two clusters: the upper one and the lower one. As shown in Figure 5, DPC-NNEG divides the dataset into nineteen NNEGs and then successfully and efficiently groups them into two sets since there are no edges between the two clusters. Homoplastically, as shown in Figure 6, our algorithm

Table 3: Clustering results measured by NMI.

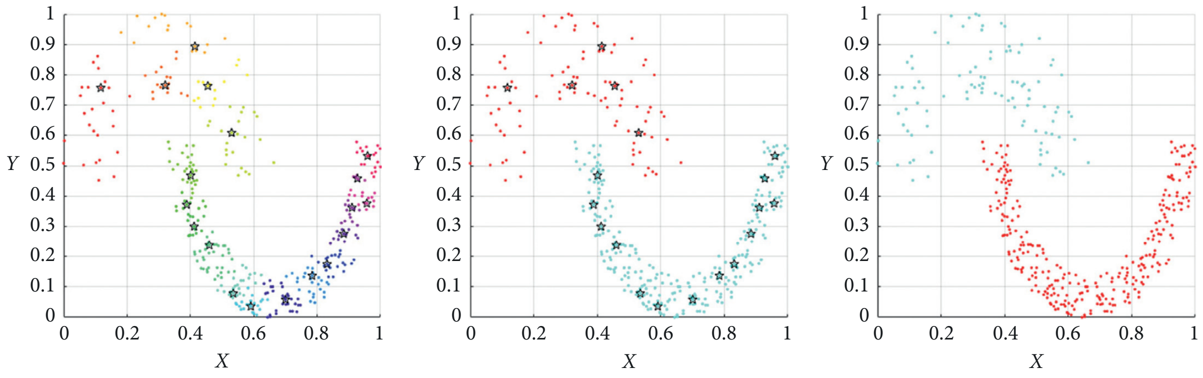
| Dataset | DPC-KNN | IDPC | FKNN-DPC | DPC-DBFN | DPC-NNEG |
|---|---|---|---|---|---|
| Jain | 1.0000 (9) | 1.0000 (9) | 1.0000 (10) | 1.0000 (9) | 1.0000 |
| Flame | 1.0000 (4) | 1.0000 (7) | 1.0000 (6) | 1.0000 (9) | 1.0000 |
| Spiral | 1.0000 (7) | 1.0000 (5) | 1.0000 (5) | 1.0000 (4) | 1.0000 |
| Statlog (shuttle) | 0.3734 (150) | 0.1552 (90) | 0.4226 (150) | 0.5490 (7) | 0.6101 |
| Abalone | 0.1780 (50) | 0.1791 (758) | 0.1828 (2) | 0.1846 (2) | 0.1852 |
| Wine quality | 0.0359 (28) | 0.0339 (68) | 0.0364 (244) | 0.0701 (1) | 0.0935 |
| DIM512 | 1.0000 (10) | 1.0000 (15) | 1.0000 (8) | 1.0000 (9) | 1.0000 |
| Libras movement | 0.5287 (53) | 0.5697 (5) | 0.5607 (11) | 0.5848 (6) | 0.5855 |

Table 4: Clustering results measured by RI.

| Dataset | DPC-KNN | IDPC | FKNN-DPC | DPC-DBFN | DPC-NNEG |
|---|---|---|---|---|---|
| Jain | 1.0000 (9) | 1.0000 (9) | 1.0000 (10) | 1.0000 (9) | 1.0000 |
| Flame | 1.0000 (4) | 1.0000 (7) | 1.0000 (6) | 1.0000 (9) | 1.0000 |
| Spiral | 1.0000 (7) | 1.0000 (5) | 1.0000 (5) | 1.0000 (4) | 1.0000 |
| Statlog (shuttle) | 0.6780 (150) | 0.4180 (90) | 0.7274 (150) | 0.7512 (7) | 0.7814 |
| Abalone | 0.8042 (14) | 0.8236 (60) | 0.7635 (5) | 0.8354 (500) | 0.8428 |
| Wine quality | 0.6277 (52) | 0.6063 (35) | 0.5107 (3) | 0.5578 (1) | 0.5751 |
| DIM512 | 1.0000 (10) | 1.0000 (15) | 1.0000 (8) | 1.0000 (9) | 1.0000 |
| Libras movement | 0.8839 (4) | 0.9089 (9) | 0.8995 (10) | 0.8945 (6) | 0.9187 |

Table 5: Clustering results measured by ARI.

| Dataset | DPC-KNN | IDPC | FKNN-DPC | DPC-DBFN | DPC-NNEG |
|---|---|---|---|---|---|
| Jain | 1.0000 (9) | 1.0000 (9) | 1.0000 (10) | 1.0000 (9) | 1.0000 |
| Flame | 1.0000 (4) | 1.0000 (7) | 1.0000 (6) | 1.0000 (9) | 1.0000 |
| Spiral | 1.0000 (7) | 1.0000 (5) | 1.0000 (5) | 1.0000 (4) | 1.0000 |
| Statlog (shuttle) | 0.3396 (150) | 0.1049 (90) | 0.3197 (150) | 0.3584 (7) | 0.5688 |
| Abalone | 0.0567 (3) | 0.0589 (10) | 0.0553 (7) | 0.0654 (2) | 0.0657 |
| Wine quality | 0.0275 (25) | 0.0356 (64) | 0.0214 (3) | 0.0575 (1) | 0.0511 |
| DIM512 | 1.0000 (10) | 1.0000 (15) | 1.0000 (8) | 1.0000 (9) | 1.0000 |
| Libras movement | 0.2492 (19) | 0.3337 (9) | 0.2846 (11) | 0.3130 (6) | 0.4862 |



(a)

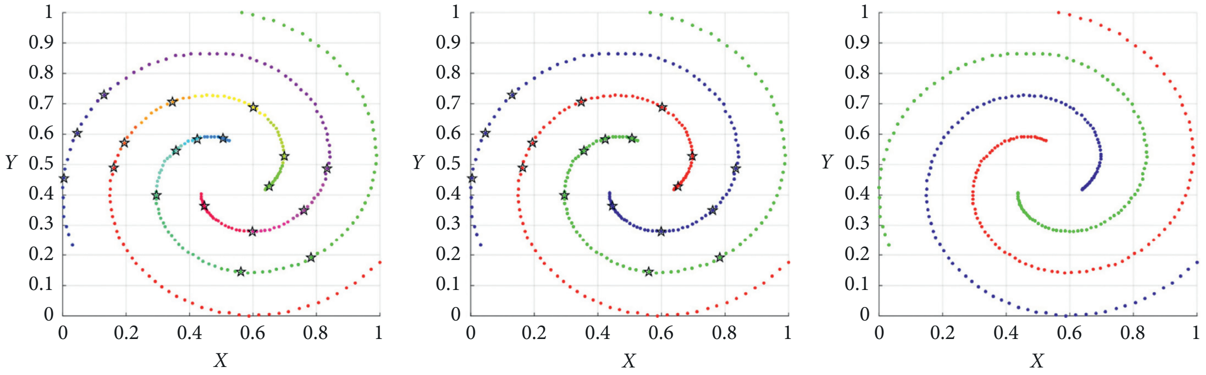
(b)

(c)

Figure 5: The clustering results of the Jain dataset. The diverse colors present different NNEGs and clusters, and the stars mark the zero samples. (a) NNEGs of Jain. (b) DPC-NNEG on Jain. (c) Ground truth.

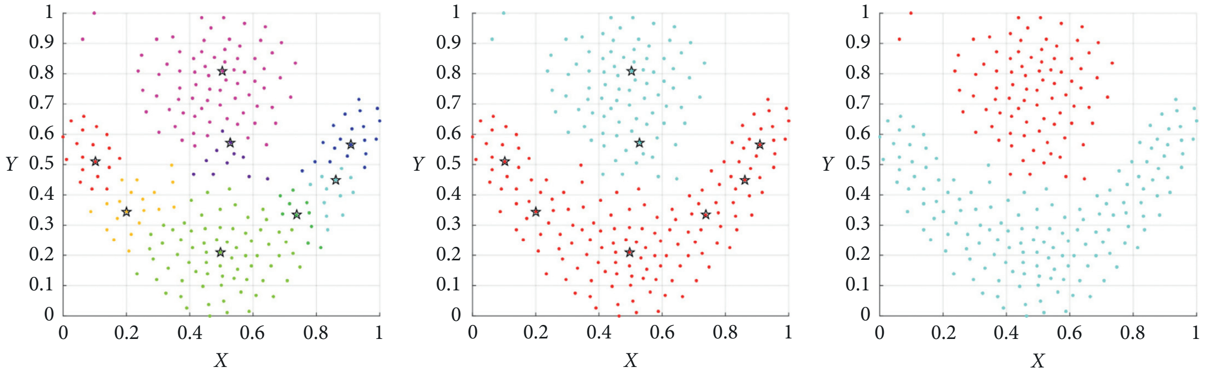
divides the Spiral dataset into several local groups and subsequently merges all NNEGs accurately into the goal number of clusters.

Unlike Jain and Spiral, as shown in Figure 7, the Flame dataset containing 240 data points has no clear gap between the two adjacent clusters. Hence, it is more sensitive to the value of dc of the DPC algorithm because a tiny change in dc will cause the border point is assigned to another cluster. However, our method not only partitions all samples into eight NNEGs but also measures the



(a) (b) (c)

Figure 6: The clustering results of the Spiral dataset. The diverse colors present different NNEGs, and the stars mark the zero samples. (a) NNEGs of spiral. (b) DPC-NNEG on spiral. (c) Ground truth.



(a) (b) (c)

Figure 7: The clustering results of the Flame dataset. The diverse colors present different NNEGs, and the stars mark the zero samples. (a) NNEGs of flame. (b) DPC-NNEG on spiral. (c) Ground truth.

tightness between different groups accurately, which realizes the correct grouping of those local groups. And Figure 7 shows that the clustering result of Flame by DPC-NNEG is consonant with the Ground Truth.

As shown in Tables 3–5, there is no difference in performance among our algorithm, DPC-DBFN, DPC-KNN, IDPC, and FKNN-DPC in three two-dimensional datasets. However, as shown in Table 2, the clustering results of more complex high-dimensional datasets show the out-performance of our method: DPC-NNEG gains the best marks measured by NMI in all datasets. For example, the results of DPC-NNEG in the Statlog (Shuttle), Abalone, Wine Quality, DIM512, and Libras Movement datasets are 0.6101, 0.1852, 0.0935, 1.0000, and 0.5855, respectively. Moreover, its improvements to the second-best method (in %) for Statlog (Shuttle), Abalone, Wine Quality, and Libras Movement datasets are respectively 11.13, 0.32, 33.38, and 0.12.

Tables 4 and 5 show similar results, respectively, measured by RI and ARI. These results also demonstrate that the proposed method, in most cases, obtains the biggest values of NMI except the Wine Quality dataset. Hence, based on these results, it can be concluded that DPC-NNEG has given an overall excellent performance in clustering.

## 5. Conclusions and Future Works

This paper proposed an efficient clustering algorithm called DPC-NNEG, which can easily split a dataset into local groups and then merge those groups into the goal number of clusters with various densities, shapes, and sizes. The proposed method aims at clustering the data by three major steps: calculating the local density of each sample, identifying natural neighbor expanded groups, and merge those groups into clusters. The first step utilizes the natural neighbor method in the local density calculation. And it is entirely different from the formula of the original DPC and could avoid the impact of outliners and reduce the sensitivity of dc. In the second step, the NNE defined is used to mine the potential structure of data, which is useful to divide the dataset into several relatively more compact local groups called NNEGs. And the last step groups all NNEGs into the goal number of clusters using the proposed formula of the closeness degree of local groups. And the application of the second and third steps not only overcomes the issue of remote assignment of the prominent density peaks but also removes the step of center selection in the original DPC. The effectiveness of the method proposed was verified on several datasets. The results show that our approach is more effective against the related improvement algorithms of DPC. In future work, we shall contribute to developing the concept of NNE to find a more suitable method for secondary-adjacent samples, instead of the given and fixated parameter $2K_i$ in equation (7). Fuzzy theory is a proper technique to mine relatively adjacent samples, in which NNE is used to construct the membership function of closeness, and then deduce the functions of secondary-adjacent samples and remote samples.

## Data Availability

All datasets in this paper are available in UCI.

## Conflicts of Interest

The authors declare that they have no conflicts of interest to report regarding the present study.

## Acknowledgments

This study was funded by the National Natural Science Foundation of China (61972056, 61772454, 61402053, and 61981340416), Hunan Provincial Natural Science Foundation of China (2020JJ4623), Scientific Research Fund of Hunan Provincial Education Department (17A007, 19C0028, and 19B005), Changsha Science and Technology Planning (KQ1703018, KQ1706064, KQ1703018-01, and KQ1703018-04), Junior Faculty Development Program Project of Changsha University of Science and Technology (2019QJCZ011), "Double First-class" International Cooperation and Development Scientific Research Project of Changsha University of Science and Technology (2019IC34), Practical Innovation and Entrepreneurship Ability Improvement Plan for Professional Degree Postgraduate of Changsha University of Science and Technology (SJCX202072), Postgraduate Training Innovation Base Construction Project of Hunan Province (2019-248-51 and 2020-172-48), and Beidou Micro Project of Hunan Provincial Education Department (XJT(2020) No.149).

## References

[1] A. Saxena, M. Prasad, A. Gupta et al., "A review of clustering techniques and developments," *Neurocomputing*, vol. 267, pp. 664–681, 2017.

[2] Y. Chen, W. Xu, J. Zuo, and K. Yang, "The fire recognition algorithm using dynamic feature fusion and IV-SVM classifier," *Cluster Computing*, vol. 22, no. 10, pp. 7665–7675, 2019.

[3] L. Sun, C. Ma, Y. Chen et al., "Low rank component induced spatial-spectral kernel method for hyperspectral image classification," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 30, no. 10, p. 3829, 2020.

[4] Y. Chen, J. Tao, Q. Zhang et al., "Saliency detection via improved hierarchical principle component analysis method," *Wireless Communications and Mobile Computing*, vol. 2020, Article ID 8822777, 12 pages, 2020.

[5] W. Lu, X. Zhang, H. Lu, and F. Li, "Deep hierarchical encoding model for sentence semantic matching," *Journal of Visual Communication and Image Representation*, vol. 71, Article ID 102794, 2020.

[6] Y. Chen, J. Wang, X. Chen et al., "Single-image super-resolution algorithm based on structural self-similarity and deformation block features," *IEEE Access*, vol. 7, pp. 58791–58801, 2019.

[7] Y. Luo, J. Qin, X. Xiang, Y. Tan, Q. Liu, and L. Xiang, "Coverless real-time image information hiding based on image block matching and dense convolutional network," *Journal of Real-Time Image Processing*, vol. 17, no. 1, pp. 125–135, 2020.

[8] Y. Chen, J. Wang, X. Chen, A. K. Sangaiah, K. Yang, and Z. Cao, "Image super-resolution algorithm based on dual-channel convolutional neural networks," *Applied Sciences*, vol. 9, no. 11, p. 2316, 2019.

[9] F. Yu, L. Liu, H. Shen et al., "Dynamic analysis, circuit design and synchronization of a novel 6D memristive four-wing hyperchaotic system with multiple coexisting attractors," *Complexity*, vol. 2020, Article ID 5904607, 17 pages, 2020.

[10] Y. Chen, L. Liu, J. Tao et al., "The improved image inpainting algorithm via encoder and similarity constraint," *The Visual Computer*, vol. 2020, 2020.

[11] L. Sun, F. Wu, T. Zhan, W. Liu, J. Wang, and B. Jeon, "Weighted nonlocal low-rank tensor decomposition method for sparse unmixing of hyperspectral images," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 13, pp. 1174–1188, 2020.

[12] Y. Chen, J. Wang, R. Xia, Q. Zhang, Z. Cao, and K. Yang, "The visual object tracking algorithm research based on adaptive combination kernel," *Journal of Ambient Intelligence and Humanized Computing*, vol. 10, no. 12, pp. 4855–4867, 2019.

[13] Y. Zhang, W. Lu, W. Ou et al., "Chinese medical question answer selection via hybrid models based on CNN and GRU," *Multimedia Tools and Applications*, vol. 79, no. 21-22, pp. 14751–14776, 2020.

[14] Y. Chen, J. Xiong, W. Xu, and J. Zuo, "A novel online incremental and decremental learning algorithm based on variable support vector machine," *Cluster Computing*, vol. 22, no. 8, pp. 7435–7445, 2019.

[15] J. Wang, J. Qin, J. Qin, X. Xiang, Y. Tan, and N. Pan, "CAPTCHA recognition based on deep convolutional neural network," *Mathematical Biosciences and Engineering*, vol. 16, no. 5, pp. 5851–5861, 2019.

[16] Y. Chen, J. Tao, L. Liu et al., "Research of improving semantic image segmentation based on a feature fusion model," *Journal of Ambient Intelligence and Humanized Computing*, vol. 2020, 2020.

[17] F. Yu, L. Liu, H. Shen et al., "Multistability analysis, coexisting multiple attractors and FPGA implementation of Yu-Wang four-wing chaotic system," *Mathematical Problems in Engineering*, vol. 2020, Article ID 7530976, 16 pages, 2020.

[18] Y. Chen, J. Wang, S. Liu et al., "Multiscale fast correlation filtering tracking algorithm based on a feature fusion model," *Concurrency and Computation: Practice and Experience*, vol. 2019, 2019.

[19] A. K. Jain, "Data clustering: 50 years beyond *K*-means," *Pattern Recognition Letters*, vol. 31, no. 8, pp. 651–666, 2010.

[20] D. Lam and D. C. Wunsch, *Academic Press Library in Signal Processing*Elsevier Press, Waltham, MA, USA, 2014.

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

[22] J. C. Dunn, "A fuzzy relative of the ISODATA process and its use in detecting compact well-separated clusters," *Journal of Cybernetics*, vol. 3, no. 3, pp. 32–57, 1973.

[23] R. Xu and D. Wunsch II, "Survey of clustering algorithms," *IEEE Transactions on Neural Networks*, vol. 16, no. 3, pp. 645–678, 2005.

[24] G. McLachlan and D. Peel, "Finite mixture models," in *Encyclopedia of Autism Spectrum Disorders*p. 1296, 1st edition, Springer Press, Manhattan, NY, USA, 2013.

[25] T. Zhang, R. Ramakrishnan, and M. Livny, "Birch," *ACM Sigmod Record*, vol. 25, no. 2, pp. 103–114, 1996.

[26] J. Zhong, W. T. Peter, and Y. Wei, "An intelligent and improved density and distance-based clustering approach for industrial survey data classification," *Expert Systems with Applications*, vol. 68, pp. 21–28, 2017.

[27] S. Guha, R. Rastogi, and S. Kyuseok, "Cure: an efficient clustering algorithm for large databases," in *Proceedings of the 1998 ACM SIGMOD International Conference on Management Of Data ACM*, pp. 73–84, Seattle, WA, USA, June 1998.

[28] S. Guha, R. Rastogi, and K. Shim, "Rock: a robust clustering algorithm for categorical attributes," in *Proceedings of the IEEE Conference on Data Engineering*, pp. 512–521, Sydney, Australia, March 1999.

[29] W. Wang, J. Yang, and R. Muntz, "Sting: a statistical information grid approach to spatial data mining," in *Proceedings of the 23rd International Conference on Very Large Data Bases*, pp. 186–195, Athens, Greece, August 1997.

[30] 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*, pp. 226–231, Portland, Oregon, August 1996.

[31] M. Ankerst, M. M. Breunig, H.-P. Kriegel, and J. Sander, "Optics: ordering points to identify the clustering structure," in *Proceedings of the ACM Sigmod Record*, pp. 49–60, Philadelphia, PA, USA, 1999.

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

[33] J. Xie, H. Gao, W. Xie, X. Liu, and P. W. Grant, "Robust clustering by detecting density peaks and assigning points based on fuzzy weighted $k$-nearest neighbors," *Information Sciences*, vol. 354, pp. 19–40, 2016.

[34] A. Lotfi, P. Moradi, and H. Beigy, "Density peaks clustering based on density backbone and fuzzy neighborhood," *Pattern Recognition*, vol. 107, Article ID 107449, 2020.

[35] L. Ni, W. Luo, W. Zhu, and W. Liu, "Clustering by finding prominent peaks in density space," *Engineering Applications of Artificial Intelligence*, vol. 85, pp. 727–739, 2019.

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

[37] A. Lotfi, S. A. Seyedi, and P. Moradi, "An improved density peaks method for data clustering," in *Proceedings of the 6th International Conference on Computer and Knowledge Engineering*, pp. 263–268, Mashhad, Iran, October 2016.

[38] P. Guo, X. Wang, Y. Wang et al., "Research on automatic determining clustering centers algorithm based on linear regression analysis," in *Proceedings of the 2017 2nd International Conference on Image, Vision and Computing*, pp. 1016–1023, Chengdu, China, August 2017.

[39] J. Ding, X. He, J. Yuan, and B. Jiang, "Automatic clustering based on density peak detection using generalized extreme value distribution," *Soft Computing*, vol. 22, no. 9, pp. 2777–2796, 2018.

[40] J. Jiang, Y. Chen, X. Meng, L. Wang, and K. Li, "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.

[41] J. L. Bentley, "Multidimensional binary search trees used for associative searching," *Communications of the ACM*, vol. 18, no. 9, pp. 509–517, 1975.

[42] D. Pfitzner, R. Leibbrandt, and D. Powers, "Characterization and evaluation of similarity measures for pairs of clusterings," *Knowledge and Information Systems*, vol. 19, no. 3, pp. 361–394, 2009.

[43] W. M. Rand, "Objective criteria for the evaluation of clustering methods," *Journal of the American Statistical Association*, vol. 66, no. 336, pp. 846–850, 1971.

[44] P. Fränti, M. Rezaei, and Q. Zhao, "Centroid index: cluster level similarity measure," *Pattern Recognition*, vol. 47, no. 9, pp. 3034–3045, 2014.