IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received 15 November 2022, accepted 4 December 2022, date of publication 9 December 2022,
date of current version 14 December 2022.

Digital Object Identifier 10.1109/ACCESS.2022.3227936

RESEARCH ARTICLE

Density Peaks Clustering Based on Potential Model and Diffusion Strength

JING CHE, WENKE ZANG, JINGWEN XIONG, AND XIYU LIU, (Member, IEEE)
School of Business, Shandong Normal University, Jinan 250014, China

Corresponding author: Wenke Zang (wink@sdnu.edu.cn)

This work was supported in part by the National Science Foundation of China under Grant 61876101 and Grant 61806114.

ABSTRACT Density peaks clustering (DPC) is a simple and efficient density-based clustering algorithm without complex iterative procedures. However, DPC needs to manually choose clustering centers via a decision graph, which often can't identify real centers and breaks the continuous flow of the algorithm. In addition, DPC is highly sensitive to the cut-off distance and suffers from the domino chain reaction. To surmount the aforementioned deficiencies, an improved density peaks clustering based on potential model and diffusion strength (DPC-PMDS) is proposed in this paper. Firstly, we utilize the potential and centrality of data points to calculate the density instead of the cut-off distance. Secondly, inspired by the information diffusion in social networks, we define the influence of data points and the diffusion strength between data points, and realize the diffusion of label from each center to the core data points while selecting clustering centers. Through this process, the core structure of each cluster is obtained and the centers are accurately identified. Finally, the distances from the boundary points to each cluster computed based on centrality are applied to assign boundary points to avoid chain reaction. Extensive experiments on synthetic, UCI and Olivetti Faces datasets demonstrate that DPC-PMDS can achieve excellent clustering results over other state-of-art algorithms, especially on datasets with complex shapes and uneven density distribution.

INDEX TERMS Density peaks clustering, potential model, diffusion strength, center detection.

I. INTRODUCTION

Clustering is an unsupervised method that divides a collection of data points into some non-empty groups based on the distance or similarity between data points. There are partition-based [1], hierarchy-based [2], grid-based [3], model-based [4], and density-based [5] Clustering algorithms. Clustering has many applications, such as image segmentation [6], [7], pattern recognition [8], recommender system [9], gene expression [10], and intrusion detection [11].

The density-based clustering method considers that the cluster center is surrounded by high-density points, and the points at the edge area of the cluster are low-density points. DBSCAN is representative of density-based clustering methods. It can find clusters with various shapes, which makes up for the shortcoming of K-means that can only find spherical clusters [12]. However, DBSCAN takes two parameters: the neighborhood radius *Eps* and the minimum number *MinPts* of points, the values of which exert tremendous influence on the algorithm results.

The density peaks clustering (DPC) algorithm [13] is a novel density-based clustering algorithm. DPC computes each data point's density and relative distance δ to construct a decision graph and selects the data points with high δ and relatively high density as the cluster centers. Each remaining point is assigned to the same cluster as its nearest neighbor with higher density. DPC has great advantages in dealing with non-spherical data distribution datasets. DPC algorithm does not require iteration, relies on few parameters, and operates efficiently. Nevertheless, DPC has the following limitations: (1) highly sensitive to the choice of cut-off distance parameter [14], (2) using decision graph to select cluster centers manually [15], and (3) affected by the problem of chain reaction [14].

To alleviate these problems, numerous improved density peaks clustering algorithms have been proposed. In order to

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

select the cut-off distance effectively, Jiang et al. [16] put forward a method to calculate the cut-off distance based on the Gini coefficient and k-nearest neighbors. Gao et al. [17] constructed an optimization function using the uncertainty of the target dataset to optimize the cut-off distance for various clustering tasks. There are also some researchers who design new density to avoid setting the cut-off distance. Lotfi et al. [18] proposed a method called DPC-DBFN that uses fuzzy kernel and k-nearest neighbors to compute the local density for improving the separability of clusters. Sun et al. [19] developed a new local density that utilizes KNN-based neighborhood and mutual neighbor degree. To enhance the precision of selecting cluster centers, Guo et al. [20] propounded DPC-CE that estimates the connectivity information between local centers with a graph-based strategy and re-evaluates the similarity between local centers by a distance punishment, which can ensure that the true cluster centers stand out in the decision graph. Li et al. [21] set two new thresholds to select candidate centers and proposed a new cluster fusion strategy to achieve the correct clustering of clusters with multiple density peaks. Flores et al. [15] came up with a method that can automatically select cluster centers by detecting gaps between data points in a one-dimensional decision graph. To eliminate the chain reaction, the FHC-LDP algorithm proposed by Guan et al. [22] uses the idea of hierarchical clustering to establish a hierarchical structure of sub-clusters by considering the association between data points. Xie et al. [14] presented two sample assignment strategies based on K-nearest neighbors, one is to assign non-outliers using a breadth-first search. The second is to assign outliers and points not assigned in the first assignment process using fuzzy weighted K-nearest neighbors.

The algorithms mentioned above have modified DPC in different aspects, but there is still a lot of work to be done when dealing with datasets with complex shapes and uneven density distribution. In this paper, we propose a novel density peaks clustering algorithm based on the potential model and diffusion strength called DPC-PMDS. Firstly, the potential of data points is computed, and the centrality of data points is obtained based on the *nneigh* information. Then a new density calculation method is proposed to better find the density peaks. Secondly, inspired by the information diffusion in social networks, the influence of data points and the diffusion strength between points are presented to select cluster centers accurately. And the initial clusters are generated by merging core points via label diffusion rule, which can well reflect the core distribution structure of the clusters and contributes to the correct assignment of the boundary points. Finally, the labels of the boundary points are obtained based on their distances from each cluster to avoid chain reaction. The main contributions of DPC-PMDS are the following three points:

1) Using the potential model and centrality without cut-off distance to calculate density. The density calculated by the new method can better reflect the structure of the dataset and make the density peaks stand out compared with the original potential.
2) The label diffusion rule is used for label propagation of core points, which can enhance the effectiveness of selecting cluster centers and avoid a cluster with multiple peaks from being split into multiple clusters.
3) To avoid the chain reaction, a new distance-based assignment method that efficiently assigns boundary points is presented.

The rest of this paper is as follows. Section 2 shows the DPC algorithm and the potential model. Section 3 presents the DPC-PMDS algorithm proposed in this paper. Section 4 shows the experiments and analysis. Finally, the conclusion in Section 5 summarizes our work.

II. RELATED WORKS

In this section, we briefly introduce DPC and the potential model and illustrate their weaknesses with examples.

A. DENSITY PEAKS CLUSTERING ALGORITHM

DPC first calculates the density and relative distance of points to find the density peaks [13]. For large-scale datasets, the local density ρ_i of data point i is estimated by the cut-off kernel:

$$\rho_i = \sum_j \chi\left(d_{ij} - d_c\right), \tag{1}$$

where d_c is the predefined cut-off distance and d_{ij} is the Euclidean distance between point i and point j. For $x = d_{ij} - d_c$, if $x < 0$, $\chi(x) = 1$, otherwise $\chi(x) = 0$. For small datasets, ρ_i is computed by Gaussian kernel:

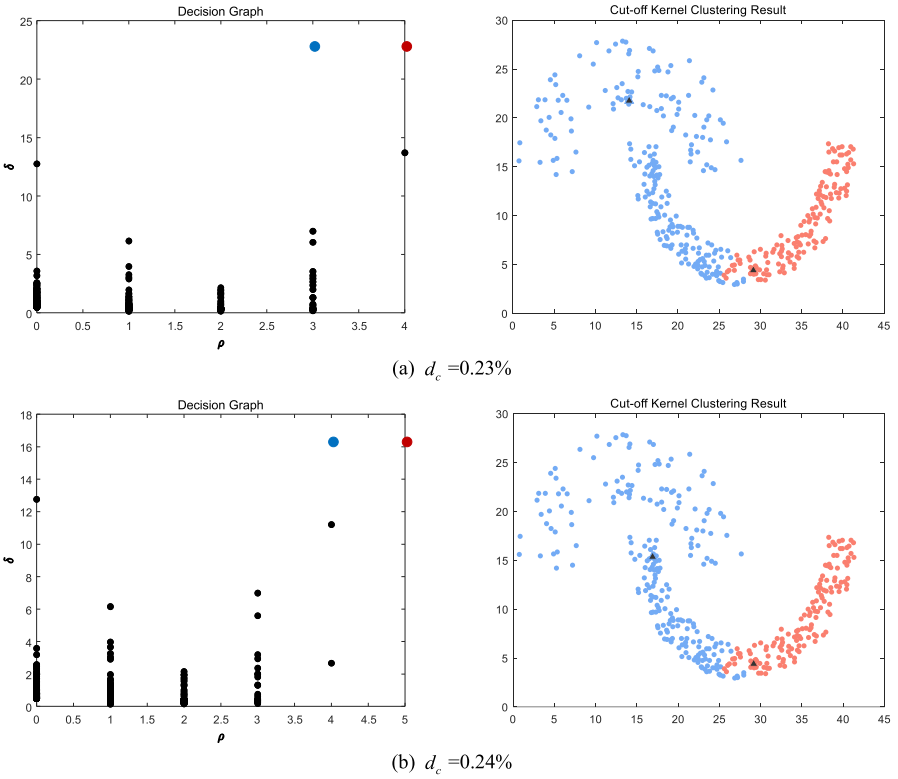
$$\rho_i = \sum_j e^{\left(-\frac{d_{ij}}{d_c}\right)^2}. \tag{2}$$

The relative distance δ_i is calculated as follows:

$$\delta_i = \begin{cases} \min\limits_{\exists j:\rho_j > \rho_i} \left(d_{ij}\right) \\ \max\limits_{\forall j:\rho_j \le \rho_i} \left(d_{ij}\right) \end{cases}. \tag{3}$$

After computing ρ_i and δ_i, DPC takes the (ρ_i, δ_i) values of all data points to build a decision graph, and manually finds the points with high δ and relatively high ρ as cluster centers. A second method is to select the data points with larger $\gamma_i = \rho_i \delta_i$ value as cluster centers. Finally, DPC assigns each remaining point to the same cluster as its nearest point with higher density.

Although DPC has great performance on a wide range of datasets, it still has several limitations. First of all, the accuracy of DPC algorithm is heavily dependent on the cut-off distance. Second, DPC manually selects the points with high density and high relative distance as cluster centers without considering the relationship between density peaks. On the dataset with uneven density distribution and complex shape, DPC may ignore the centers of low-density clusters and select redundant centers of high-density clusters. This subjective approach also breaks the continuity of the algorithm. Third, DPC is affected by the chain reaction problem,



(a) d_c =0.23%

(b) d_c =0.24%

FIGURE 1. Decision graph and clustering results obtained with the cut-off kernel for $d_c = 0.23\%$ and $d_c = 0.24\%$.

i.e., if a data point is incorrectly assigned, it may lead to the misallocation of its nearby data points, resulting in erroneous propagation of clustering labels. These limitations can be exemplified by the Jain dataset. Jain contains two clusters with significantly different density distributions. Fig.1 shows the decision graph and clustering results obtained with the cut-off kernel for $d_c = 0.23\%$ and $d_c = 0.24\%$. The colored points in the decision graph correspond to the centers of the corresponding colored clusters in the result graph. From Fig.1(a) and Fig.1(b), it can be seen that the value of d_c have a great influence on the results. Fig.1(b) shows that the cluster centers selected by DPC via decision graph are all in the bottom cluster, because the density peaks in the bottom cluster have much higher density and relative distance than the upper cluster. Furthermore, in Fig.1(a), it is obvious that the assignment of data points is affected by the chain reaction problem.

B. THE POTENTIAL MODEL

Lu et al. [23] put forward a clustering method called Clustering by Sorting Potential Values (CSPV) based on a potential model. The potential model considers the data points following Newton's law of universal gravitation and sets the mass of all points to 1. The gravitational force between point i and point j is obtained as:

$$\vec{F}_{ij}\left(\vec{r}_{ij}\right) = \begin{cases} G\dfrac{\hat{r}_{ij}}{r_{ij}^2} & if\ r_{ij} \geq \eta \\ 0 & if\ r_{ij} < \eta \end{cases}, \tag{4}$$

where $\vec{r}_{ij}$ and $\hat{r}_{ij}$ are the vector and unit vector from point i to point j, respectively, and r_{ij} is the Euclidean distance between i and j. G is the gravitational constant. The parameter η is used to avoid the problem of singularity when r_{ij} is zero.

In the potential model, only the relative value of the potential is considered, so G is set to 1 for the convenience of calculation. The simplified potential at point i from point j is calculated as:

$$\Phi_{ij}\left(r_{ij}\right) = \int_{r_{ij}}^{\infty} \vec{F}_{ij}\left(\vec{r}\right) \cdot \hat{r} dr = \begin{cases} -\frac{1}{r_{ij}} & if\ r_{ij} \geq \eta \\ -\frac{1}{\eta} & if\ r_{ij} < \eta \end{cases}. \tag{5}$$

The potential of point i is:

$$\Phi_i = \sum_{j \neq i} \Phi_{ij}\left(r_{ij}\right). \tag{6}$$

Lu et al. [24] used the distance matrix of the dataset to select the parameter η to satisfy the condition of Scale-Invariance:

$$MinD_i = \min_{r_{ij} \neq 0, j=1,\cdots n} \left(r_{ij}\right), \tag{7}$$

$$\eta = mean\left(MinD_i\right)/S, \tag{8}$$

where $MinD_i$ is the minimum distance from point i to all the other points, and n is the number of points. S is a scale factor, generally set to 10.

The Parzen window function, a nonparametric estimation method, is used to demonstrate that the total potential value is negatively proportional to the estimated probability density [24]. Thereby, the smaller the potential of a data point, the higher its density.

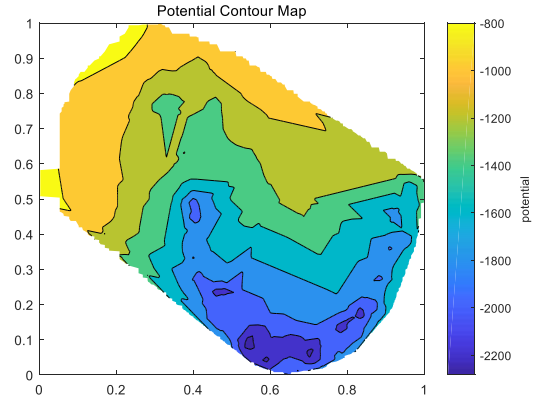


FIGURE 2. Potential contour map of the Jain dataset.

Fig.2 shows the contour map of potential of the Jain dataset. The darker the color in Fig.2, the lower the potential, i.e., the higher the density. The potential is calculated from a global perspective, so the potential does not display the data distribution of the dataset with uneven density well. As can be seen from Fig.2, the potential of high-density clusters is extremely high, while the potential of low-density clusters is extremely low. The boundaries of clusters are not clear, and density peaks of low-density clusters also do not stand out well in the figure.

III. THE PROPOSED METHOD

In this section, the detailed procedure of the DPC-PMDS algorithm is presented.

A. DENSITY CALCULATION

In this subsection, the density is calculated by considering potential and centrality. The values of potential are negative, and the smaller the potential, the higher the corresponding density. Therefore, we first calculate the density by changing the sign of the potential and normalizing it:

$$\rho_i = \frac{-\Phi_i}{\max\limits_{j=1,\cdots n}\left(\Phi_j\right)}. \tag{9}$$

Because low-density clusters can't be identified well with the density calculated by the original potential, we improve it by considering the centrality of data points. For any point j except the point with the highest density, we use $nneigh_j$ to denote the nearest neighbor with higher density of j:

$$nneigh_j = \left\{ i | \min_{\exists i:\rho_i > \rho_j} \left(d_{ji}\right) \right\}. \tag{10}$$

The centrality of point i is defined as follows:

$$c_i = |A_i| + 1, \quad A_i = \left\{ j | nneigh_j = i \right\}, \tag{11}$$

where $|\cdot|$ is the cardinality of the set. The centrality c_i is greater than or equal to 1. A large value of c_i means that i is the nearest neighbor with higher density of many points around it, i.e., point i has a relatively high density in its neighborhood. $c_i = 1$ means that no point has the nearest neighbor with higher density equal to i, which indicates that the density of point i is relatively small in its neighborhood. Therefore, the larger the c_i, the more likely that point i is a density peak. We take c_i as the weight and multiply it with ρ_i, then the calculation formula of new density is:

$$\rho_i' = c_i \times \rho_i. \tag{12}$$

Fig.3 is the contour map of the new density of the Jain dataset. The darker the color in Fig.3, the higher the density. Compared with Fig.2, it can be found that the new density can better reflect the data distribution and facilitate the selection of density peaks.

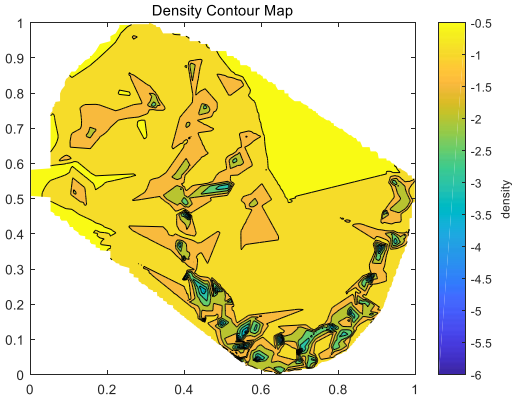


FIGURE 3. Density contour map of the Jain dataset.

B. DIFFUSION STRENGTH

In this subsection, the influence of data points and the diffusion strength of two data points will be introduced.

In social networks, information diffusion is carried out via interactions between nodes (links in the network), i.e., information of nodes is diffused through paths composed of edges. Each information diffusion network can be considered as a tree-like structure. The root node, the publisher of information, diffuses information to the leaf nodes. Generally, the influence of nodes and the relationship between nodes affect information diffusion [25]. Inspired by the information diffusion in social networks, we regard the assignment of clustering labels as a label diffusion process. The core points of each cluster are linked via label diffusion rule to avoid selecting excessive centers of high-density clusters and neglecting centers of low-density clusters. Before introducing the label diffusion rules, the definitions of influence and diffusion strength are given.

The influence of a point is related to its distance from its surrounding points. The definition of the influence of point i is:

$$I_i = \bar{X}_i + 2S_i, \tag{13}$$

where $\bar{X}_i$ and S_i are the mean and standard deviation of the distance from the point to its surrounding points, respectively.

They are calculated as:

$$\bar{X}_i = \frac{1}{s} \sum_{j \in sNN_i} d_{ij}, \tag{14}$$

$$S_i = \sqrt{\frac{1}{s-1} \sum_{j \in sNN_i} \left(d_{ij} - \bar{X}_i\right)^2}, \tag{15}$$

$sNN(x_i)$ is a set of nearest neighbors of i:

$$\mathrm{s}NN_i = \left\{ j | d_{ij} \leq d_{is} \right\}, \tag{16}$$

where d_{is} is the Euclidean distance between data points i and the sth nearest neighbor of i. s is equal to the maximum value of the centrality c of all data points plus a parameter α, i.e., $s = \max(c) + \alpha$. α is an integer greater than $-\max(c)$. The larger α, the larger $\bar{X}_i$. Thus the value of α is positively correlated with the influence of the data points. The greater the influence of data points, the greater their ability to diffuse labels to surrounding points.

Next, the computation formula for measuring diffusion strength from point i to point j will be described. The influence of a data point represents its label diffusion ability. The larger I_i and I_j, the larger diffusion strength from point i to point j. $\bar{X}_i$ represents the mean distance between data point i and its neighbors. If $\bar{X}_i$ and $\bar{X}_j$ are very different, it means that the distribution characteristics of points around them are very different and point i and j are very likely not close to each other. Hence, the diffusion strength between points i and j should be small. In addition, the diffusion strength from points in the boundary region to other points should be small to avoid propagating cluster labels to other clusters. Let ci denote the cluster center of the cluster where point i is located. If the difference between $\bar{X}_i$ and $\bar{X}_{ci}$ is large, point i is likely to be a boundary point far from the cluster center, so the diffusion strength of point i to other points should be small. Thereby, the diffusion strength from point i to point j is:

$$ds_{ij} = \frac{\min\left(\bar{X}_i, \bar{X}_j\right)}{\max\left(\bar{X}_i, \bar{X}_j\right)} \cdot \frac{\min\left(\bar{X}_i, \bar{X}_{ci}\right)}{\max\left(\bar{X}_i, \bar{X}_{ci}\right)} \cdot \left(\frac{I_i + I_j}{2}\right), \tag{17}$$

ds_{ij} indicates the ability of point i to diffuse labels to point j. The larger ds_{ij} is, the more likely point i is to diffuse its own label to point j.

C. CENTER IDENTIFICATION AND CORE POINT LABLE DIFFUSION

In this subsection, the label diffusion rule is defined, the process of automatically determining cluster centers and assigning core points is described.

The decision value γ is the probability of each data point becoming a cluster center and the decision value of data point i is calculated by:

$$\gamma_i = \frac{\rho_i'}{\max(\rho')} \times \frac{\delta_i}{\max(\delta)}. \tag{18}$$

The data points are sorted in descending order according to the γ value.

We assume that point i already has a cluster label. Based on the diffusion strength, the label diffusion rule is defined as follows:

$$CR_{ij} = \begin{cases} 1 & if\ d_{ij} \leq ds_{ij} \wedge j \in sNN_i \\ 0 & if\ d_{ij} > ds_{ij} \vee j \notin sNN_i \end{cases}, \tag{19}$$

where ds_{ij} is the diffusion strength from point i to point j, and d_{ij} is the Euclidean distance between point i and point j. $CR_{ij} = 1$ means that point i can propagate its label to j, point j is within the diffusion range of i. Otherwise, $CR_{ij} = 0$ means that i cannot propagate its label to j, i.e., j is not within the diffusion range of i.

Then the diffusion range of data point i can be obtained through the label diffusion rule:

$$DR_i = \left\{ j | CR_{ij} = 1 \right\}. \tag{20}$$

A point can propagate its label to points that are within its diffusion range.

As mentioned above, if there are two or more data points with high δ and high ρ in the same cluster, selecting the center by decision graph or γ value may split a cluster into multiple clusters. In addition, the cluster centers with lower density in the dataset are easily ignored. To address this problem, we propose a strategy to select centers by γ values and the label diffusion rule, which can select the centers of low-density clusters and connect core points of each cluster.

Firstly, we select the point with the largest γ value as the first cluster center, and then add the points within the diffusion range of the cluster center to the cluster. Next, we iterate through the newly added points and assigned the points within their diffusion range to the cluster. This traversal process continues until the diffusion ranges of all points assigned to the cluster are traversed and no new points can be assigned to the cluster. Then we select the unassigned point with the largest γ value as the new cluster center, and loop the above steps of assigning points according to the label diffusion rule until the number of cluster centers reaches the number of clusters we want. In this process, the points assigned by the label diffusion rule are called core points and the remaining points are boundary points. Finally, the cluster center and core points of each cluster are obtained, i.e., the initial clusters are generated. The specific steps are shown in Algorithm 1.

We take the Flame and Jain datasets as an example, the results of selecting the center and assigning core points of Flame and Jain according to Algorithm 1 are shown in Fig.4. Points colored black in the graph are boundary points that are not assigned through the label diffusion rule, and the points marked with black triangles are the cluster centers. The directed line segment shows the process of label diffusion. In each cluster, the labels start from the cluster center and spread to the surrounding points. As can be found in Fig.4, the algorithm correctly selects the cluster centers and connects the core points of each cluster. On the Flame dataset, the two clusters have intersection and the points at the junction of the two clusters are the boundary points. On the Jain dataset, the

Algorithm 1 Center Identification and Core Point Label Diffusion

Input: Dataset X, the number of clusters *npeak*.

Output: Cluster centers and the clustering results of core points C_{core}.

1. Let $NCLUST = 0$.
2. **while** $NCLUST \neq npeak$ and there are unassigned points in the dataset
3. Select the unassigned point with the largest γ value as the cluster center to create a new cluster.
4. $NCLUST = NCLUST + 1$.
5. Create a queue Q and put the clustering center into the queue Q.
6. **while** Q is not empty
7. Take the head node q1 of Q.
8. Find all the unassigned points within the diffusion range of q1, assign these points to the cluster where q1 is located and put them in the queue Q.
9. Delete q1 in Q.
10. **end**
11. Delete Q
12. **end**
13. Output the cluster centers and the clustering results of core points.

two clusters have no intersection and most of the points of both clusters are considered as core points.

D. BOUNDARY POINTS ASSIGNMENT

The core points of each cluster are connected by the label diffusion rule, there are still some boundary points that are not assigned because they do not conform to the label diffusion rule. In this subsection, a distance-based assignment method is used to obtain the labels of the boundary points to avoid the chain reaction.

For any boundary point i, we calculate its distance to each cluster through the sum of its distance to the c_i nearest points of each cluster separately. Then i is assigned to the cluster with the minimum distance. Fig.5 shows the final clustering results of the Flame dataset and the Jain dataset. From Fig.5, it can be seen that the algorithm proposed in this paper obtains the correct clustering results on the Flame and Jain datasets.

E. THE TIME COMPLEXITY

This section gives the computational complexity of the DPC-PMDS algorithm. The time complexity of DPC-PMDS depends on five main steps: 1) calculating the distance between data points and the potential of each data point, with a time complexity of $O(n^2)$, 2) calculating the centrality and density of data points with a time complexity of no more than $O(n^2)$, 3) searching for the nearest s points to calculate the diffusion strength of data points requires $O(n^2)$, 4) obtaining the cluster centers and connecting the core points within the same cluster according to the γ value and label diffusion rule, the time complexity of this process will not exceed $O(n^2)$, 5) assigning the boundary points according to the distance to the nearest c_i points of each cluster, the time complexity is $O(n^2)$. Therefore, the time complexity of the DPC-PMDS algorithm proposed in this paper is $O(n^2)$ as that of DPC algorithm.

Algorithm 2 DPC-PMDS

Input: Dataset $X = \{x_1, x_2, \cdots x_n\}$, cluster number *npeak*.

Output: Cluster centers and cluster label vector of data points.

1. Normalize the dataset X
2. Calculate the distance matrix using Euclidean distance
3. Calculate the density ρ_i from Eq.(9)
4. Calculate c_i from Eq. (11)
5. Calculate new density ρ_i' from Eq.(12)
6. Calculate the relative density δ_i from Eq. (3)
7. Calculate diffusion strength $ds_{ij}$from Eq. (17)
8. Calculate γ_i from Eq. (18) and sort the data points in descending order by γ value.
9. Select cluster centers and assign core points according to Algorithm 1.
10. Handle boundary points. Each boundary point is assigned to the cluster with the minimum distance.
11. Output the cluster centers and cluster label vector.

IV. EXPERIMENTS

In this section, for the sake of evaluating the performance of DPC-PMDS, we compare DPC-PMDS with DPC [13], PHA [24],[1] and state-of-the-art clustering methods including DPC-DBFN [18],[2] DPC-CE [20][3] and FHC-LDP [22][4] on a variety of datasets. The time complexity of the PHA, DPC-DBFN and DPC-CE algorithms is $O(n^2)$, and the time complexity of the FHC-LDP is $O(n\log(n))$. The Accuracy(ACC) [26], Normalized Mutual Information (NMI) [27], Rand Index(RI) [28] and Adjusted Rand Index (ARI) [29] are used to evaluate the performance of clustering algorithms. We implement the proposed PDC-PMDS and other five comparison algorithms in a desktop computer with Intel(R) Xeon(R) CPU E5-2430 0 @ 2.20 GHz 2.20 GHz, Windows 10 Professional Edition 64-bit OS. All the clustering methods' codes are written, run, and tested by MATLAB R2017b.

A. DATASETS

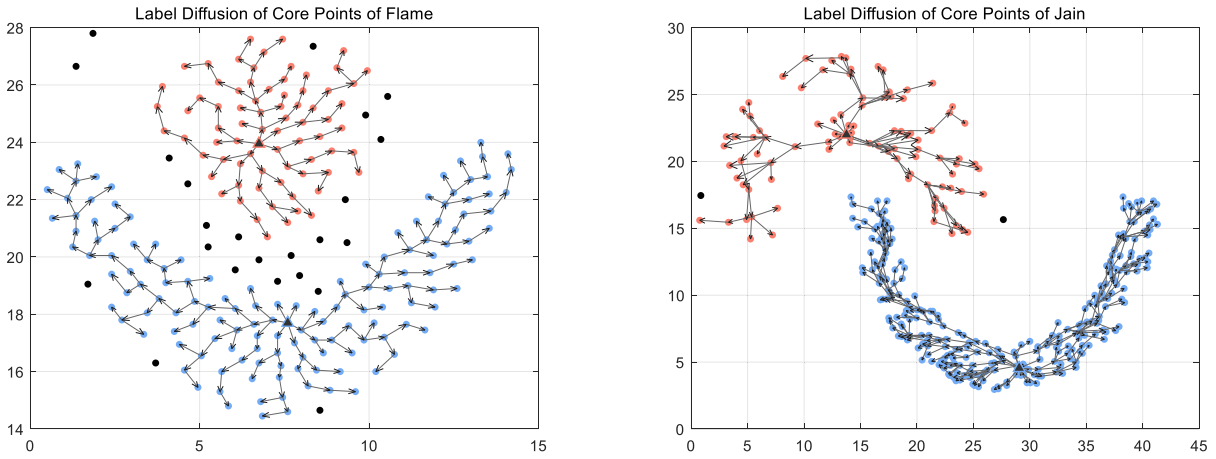
The synthetic datasets including Aggregation [30], Flame [31], Jain [32], Spiral [33], Pathbased [33], Compound [34], R15 [35], D31 [35], threecircles [36], CMC [37], S1 [38], and Unbalance [39] are used in this paper. These twelve synthetic

[1]https://ww2.mathworks.cn/matlabcentral/fileexchange/46134-fast-hierarchical-clustering-method-pha?s_tid=srchtitle_PHA_1

[2]https://github.com/abdulrahmanlotfi/DPC-DBFN

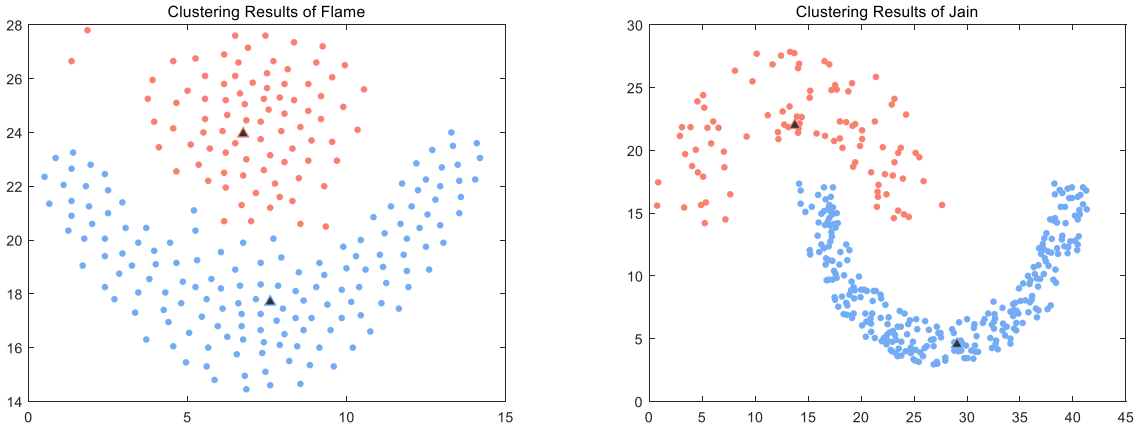
[3]https://github.com/WJ-Guo/DPC-CE

[4]https://github.com/Guanjunyi/FHC-LDP-a-variant-of-density-peak-clustering-DPC



(a) Flame (b) Jain

FIGURE 4. Results of selecting cluster centers and assigning core point of dataset Flame and Jain.



(a) Flame (b) Jain

FIGURE 5. Clustering results of Flame and Jain datasets.

datasets can evaluate the ability of DPC-PMDS to identify clusters of datasets with diverse shapes and uneven density distributions. Additionally, eight UCI datasets (available at https://archive.ics.uci.edu/ml/datasets.php) and the Olivetti Faces [40] dataset are used to further evaluate the performance of DPC-PMDS on real-world datasets with different data volumes and dimensions. The UCI datasets contain Iris, Seeds, DNA, Diabetes, Thyroid, Abalone, Cloud, and Robot navigation. Tables 1 and 2 show the number of instances, the number of attributes, and the number of clusters for each dataset.

TABLE 1. Synthetic datasets used in this paper.

Datasets	# of instances	#of features	# of clusters
Aggregation	788	2	7
Flame	240	2	2
Jain	373	2	2
Spiral	312	2	3
Pathbased	300	2	3
Compound	399	2	6
D31	3100	2	31
R15	600	2	15
threecircles	299	2	3
CMC	1002	2	3
s1	5000	2	15
Unbalance	6500	2	8

B. PARAMETERS

In order to obtain a fair comparison, the parameters are set according to the description of the parameters in the original paper of the comparison algorithms. For DPC algorithm, d_c is usually chosen so that the average number of neighbors is around 1% to 2% of the total number of points in the dataset. We expand this scope to 0.5% to 3% and run the algorithm multiple times in steps of 0.5 to take the optimal value. The PHA algorithm has only one parameter S, which defaults to 10. For DPC-DBFN algorithm with parameter k, the value of

TABLE 2. UCI datasets used in this paper.

Datasets	# of instances	#of features	# of clusters
Iris	150	4	3
Seeds	210	7	3
DNA	2000	180	3
Diabetes	768	8	2
Thyroid	215	5	3
Abalone	4177	7	29
cloud	1024	10	2
Robot navigation	5456	24	4

k is selected from 1 to 40 to get the optimal value. DPC-CE contains three parameters, which are set to $dc = 2\%$, $Tr = 0.25$, and $Pr = 0.3$ in the original paper. The parameter k of FHC-LDP is also set according to the original paper. When the number of data points $n < 500$, we set $5 \leq k \leq 20$, when $500 \leq n < 10000$, we set $1\%n \leq k \leq 3\%n$, when $n \geq 10000$, we set $20 \leq k \leq 2\%n$. The specific parameter settings of each algorithm are shown in Table 3.

TABLE 3. Parameters setting of algorithms used in this paper.

Algorithms	Parameters setting
DPC	$d_c = 0.5\% \sim 3\%$
PHA	$S = 10$
DPC-DBFN	$k = 1 \sim 40$
DPC-CE	$dc = 2\%, Tr = 0.25, Pr = 0.3$
FHC-LDP	$if\ n < 500,\ 5 \leq k \leq 20$ $if\ 500 \leq n < 10000,\ 1\%n \leq k \leq 3\%n$ $if\ n \geq 10000,\ 20 \leq k \leq 2\%n$
DPC-PMDS	$(-\max(c), 30]$

C. RESULTS ON SYNTHETIC DATASETS

In this subsection, experiments are conducted on 12 synthetic datasets. The visual clustering results of DPC, PHA, DPC-DBFN, DPC-CE, FHC-LDP and the method proposed in this paper (DPC-PMDS) are shown in Fig.6-17. The ACC, NMI, RI, and ARI metrics of all algorithms are given in Table 4. The optimal values of the evaluation metrics on each dataset are bolded.

Fig.6 and Fig.7 show the clustering results on the Aggregation and Flame datasets. The points marked with black triangles are the cluster centers. On the Aggregation dataset, the clustering results of PHA, FHC-LDP, and DPC-PMDS are completely correct with the value of each evaluation metric is 1. DPC, DPC-DBFN, and DPC-CE have errors for the assignment of a few points. Flame is a dataset with overlapping area between clusters. From Fig.7, it can be seen that DPC and PHA obviously cannot separate the two clusters, DPC-DBFN has inaccurate assignment for the points in the junction part of two clusters. The clustering results of DPC-CE, FHC-LDP, and DPC-PMDS are completely correct.

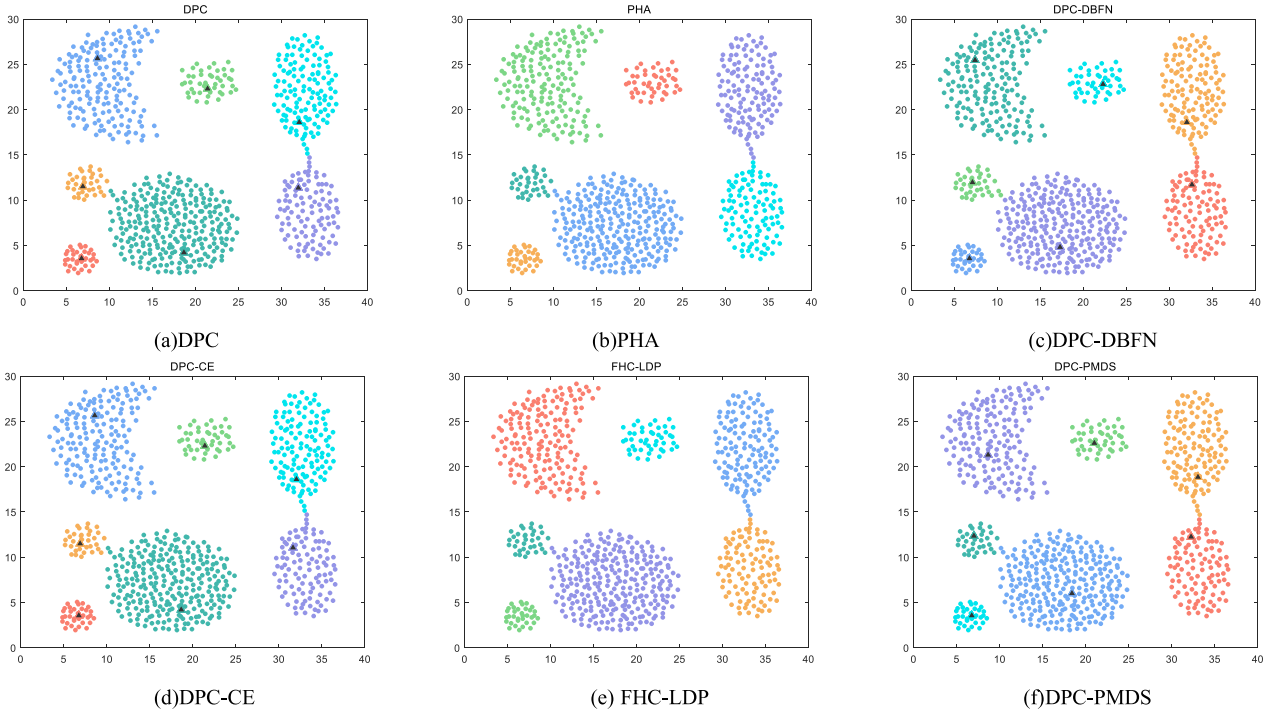
Fig.8 and Fig.9 show the clustering results on the Jain and Spiral datasets. These two datasets have clusters with irregular shape and data points with uneven density distribution. And there is no intersection between clusters. The clustering results of DPC-CE, FHC-LDP, and DPC-PMDS on these two datasets are correct because DPC-CE is based on the local central connectivity information estimation strategy of the graph, FHC-LDP considers the association between adjacent points, and the DPC-PMDS algorithm proposed in this paper considers the diffusion strength when finding the clustering centers. On the Jain dataset, DPC and DPC-DBFN cannot identify the correct cluster centers, and PHA also cannot separate the two clusters. On the Spiral dataset, although DPC and DPC-DBFN can identify the correct cluster centers, there is a problem with the assignment strategy. The potential-based hierarchical clustering method PHA also obtains the correct clustering results.

Fig.10 and Fig.11 show the clustering results on the Pathbased and Compound datasets. These two datasets have clusters with large density differences and intersections. The clustering result of DPC-PMDS on these two datasets is significantly better than other algorithms. On Pathbased dataset, the evaluation metrics of DPC-PMDS is 1, while that of all other algorithms is far less than 1. On the Compound dataset, the DPC-PMDS algorithm identifies cluster centers and correctly assigns most of the points. DPC, PHA, DPC-DBFN and FHC-LDP cannot identify the correct clusters. DPC-CE has erroneous assignment for low-density clusters.

Fig.12 and Fig.13 show the clustering results on the D31 and R15 datasets. These two datasets have more instances and clusters than the previous datasets, and the clusters in these two datasets are mainly spherical in shape. The clustering results of the algorithms on these two datasets are similar. On the D31 dataset, DPC-PMDS does not perform as well as DPC-DBFN, DPC-CE and FHC-LDP, but it is not much different from them, and it outperforms the other two algorithms. On the R15 dataset, the clustering results of DPC-PMDS and DPC-DBFN are the best.

Fig.14 and Fig.15 show the clustering results on the Three-circles and CMC datasets. The points in the central region of these two datasets have a higher density than the surrounding points. From Fig.14, it can be seen that DPC and DPC-DBFN cannot separate clusters correctly, which is because just choosing points with high density and relative distance as centers on this dataset will choose the wrong cluster centers. PHA incorrectly combines two clusters in the central region into one cluster. The clustering results of DPC-CE, FHC-LDP, and DPC-PMDS are completely correct. On the CMC dataset, the clustering results of DPC-PMDS and FHC-LDP are correct. PHA cannot distinguish different clusters and DPC cannot identify the correct cluster centers. DPC-DBFN and DPC-CE can identify the correct cluster centers, but the points are inaccurately assigned.

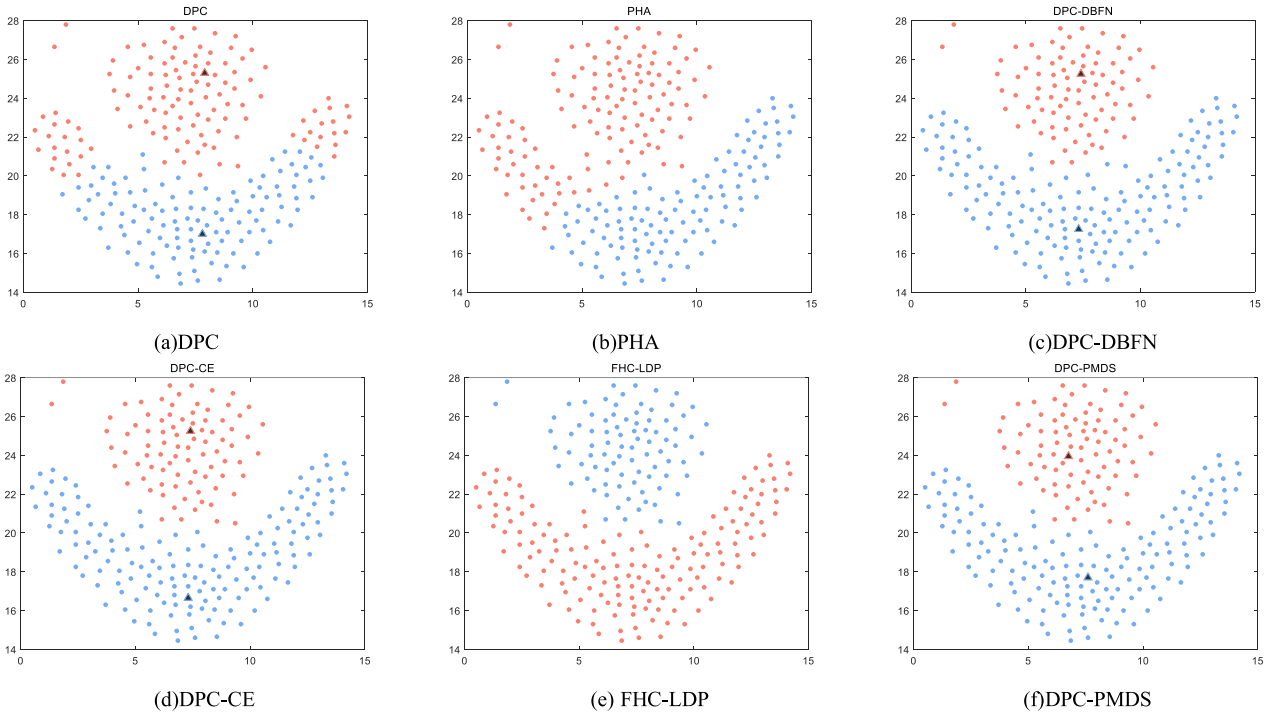
Fig.16 and Fig.17 show the clustering results on S1 and Unbalance datasets. The number of instances for these two datasets is 5000 and 6500, respectively, which can verify the performance of the algorithm on large-scale datasets. On the S1 dataset, the result of DPC-PMDS is optimal among all algorithms. The shape of clusters on the Unbalanced dataset is simple, and there is no intersection between clusters. All the algorithms except the PHA algorithm have correct clustering results on this dataset.



(a)DPC (b)PHA (c)DPC-DBFN

(d)DPC-CE (e) FHC-LDP (f)DPC-PMDS

FIGURE 6. Clustering results of (a) DPC, (b)PHA, (c)DPC-DBFN, (d)DPC-CE, (e)FHC-LDP and (f) the proposed method (DPC-PMDS) clustering methods on the Aggregation dataset.



(a)DPC (b)PHA (c)DPC-DBFN

(d)DPC-CE (e) FHC-LDP (f)DPC-PMDS

FIGURE 7. Clustering results of (a) DPC, (b)PHA, (c)DPC-DBFN, (d)DPC-CE, (e)FHC-LDP and (f) the proposed method (DPC-PMDS) clustering methods on the Flame dataset.

Table 4 shows the optimal clustering results of all algorithms on the 12 synthetic datasets. DPC-PMDS is optimal among all algorithms on all datasets except the D31 dataset. The values of ACC, NMI, RI, and ARI for DPC-PMDS on Aggregation, Flame, Jain, Spiral, Pathbased, Threecircles, CMC, and Unbalance datasets are all 1. On the Compound

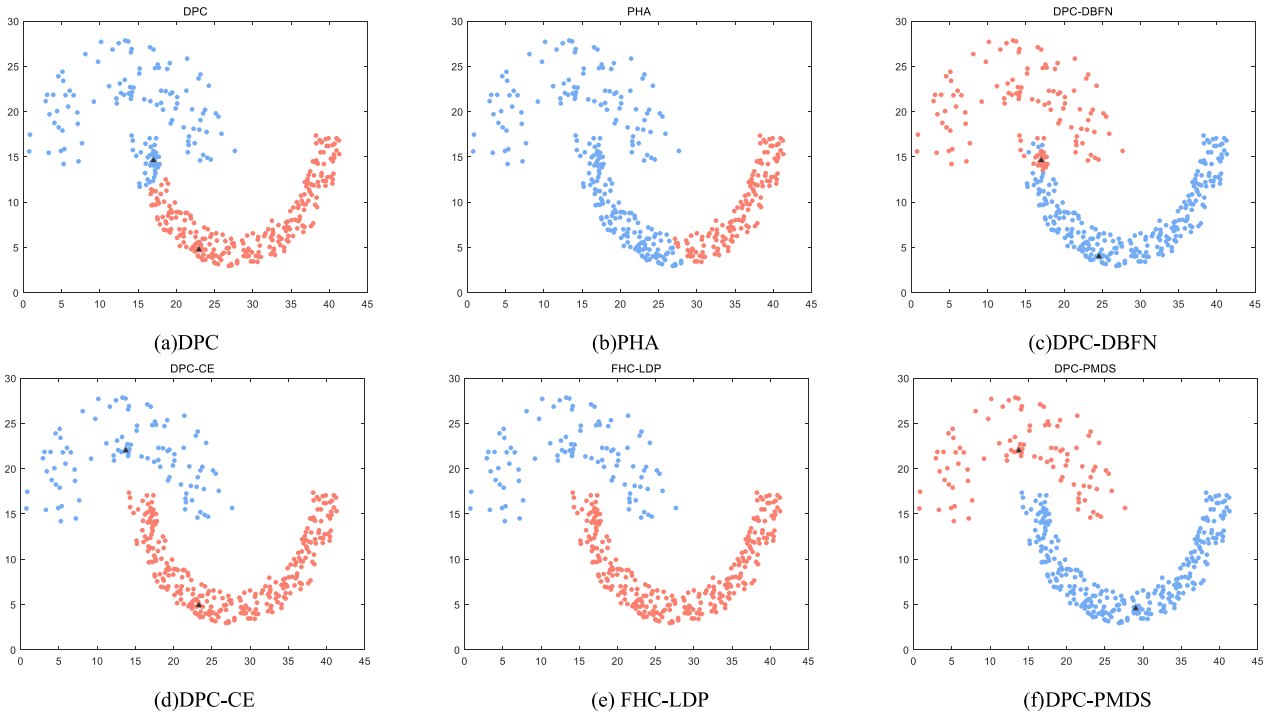


FIGURE 8. Clustering results of (a) DPC, (b)PHA, (c)DPC-DBFN, (d)DPC-CE, (e)FHC-LDP and (f) the proposed method (DPC-PMDS) clustering methods on the Jain dataset.

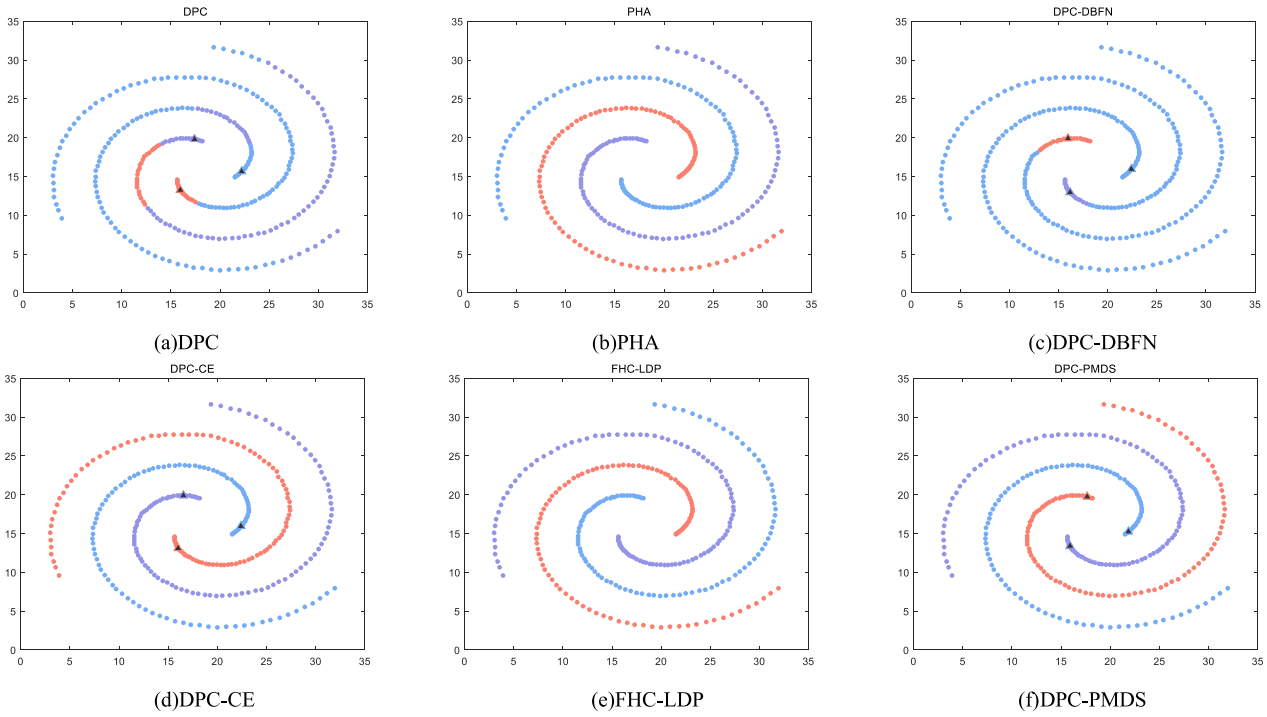


FIGURE 9. Clustering results of (a) DPC, (b)PHA, (c)DPC-DBFN, (d)DPC-CE, (e)FHC-LDP and (f) the proposed method (DPC-PMDS) clustering methods on the Spiral dataset.

dataset, ACC, NMI, RI, and ARI of DPC-PMDS are significantly greater than other algorithms. On the R15 dataset, the evaluation metrics of DPC-PMDS and DPC-DBFN are marginally higher than the other algorithms. On the S1

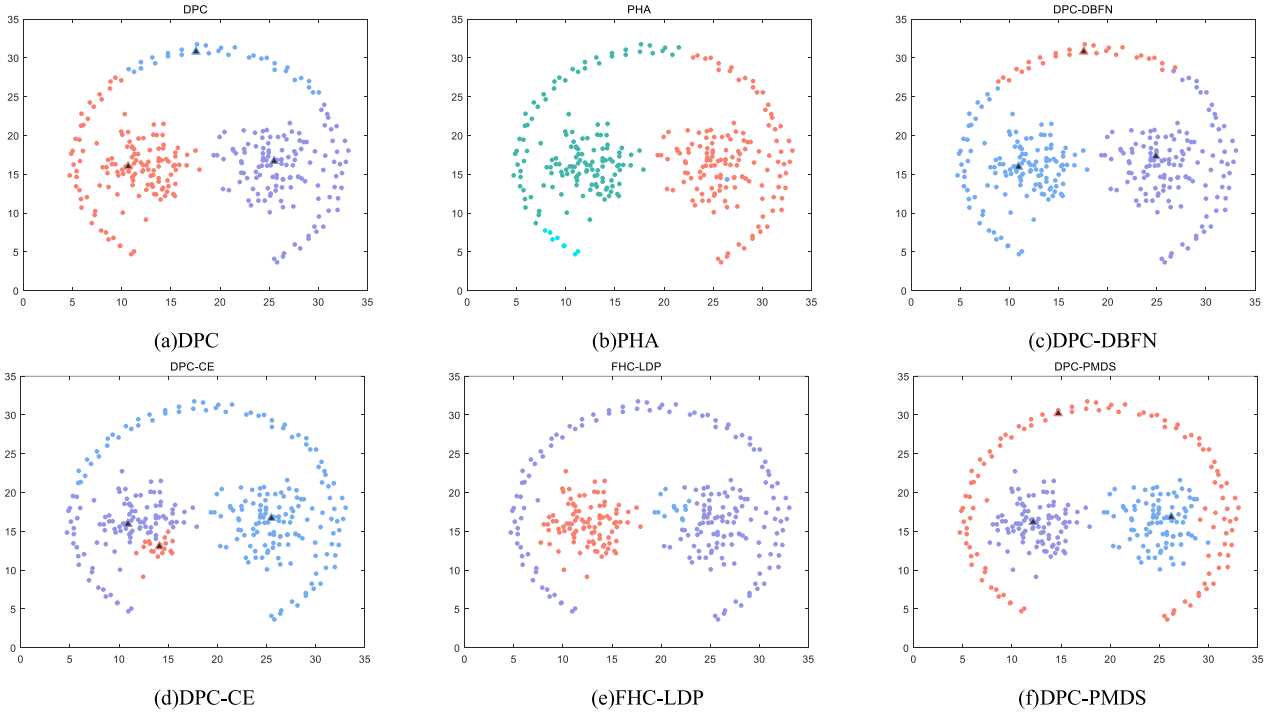


FIGURE 10. Clustering results of (a) DPC, (b)PHA, (c)DPC-DBFN, (d)DPC-CE, (e)FHC-LDP and (f) the proposed method (DPC-PMDS) clustering methods on the Pathbased dataset.

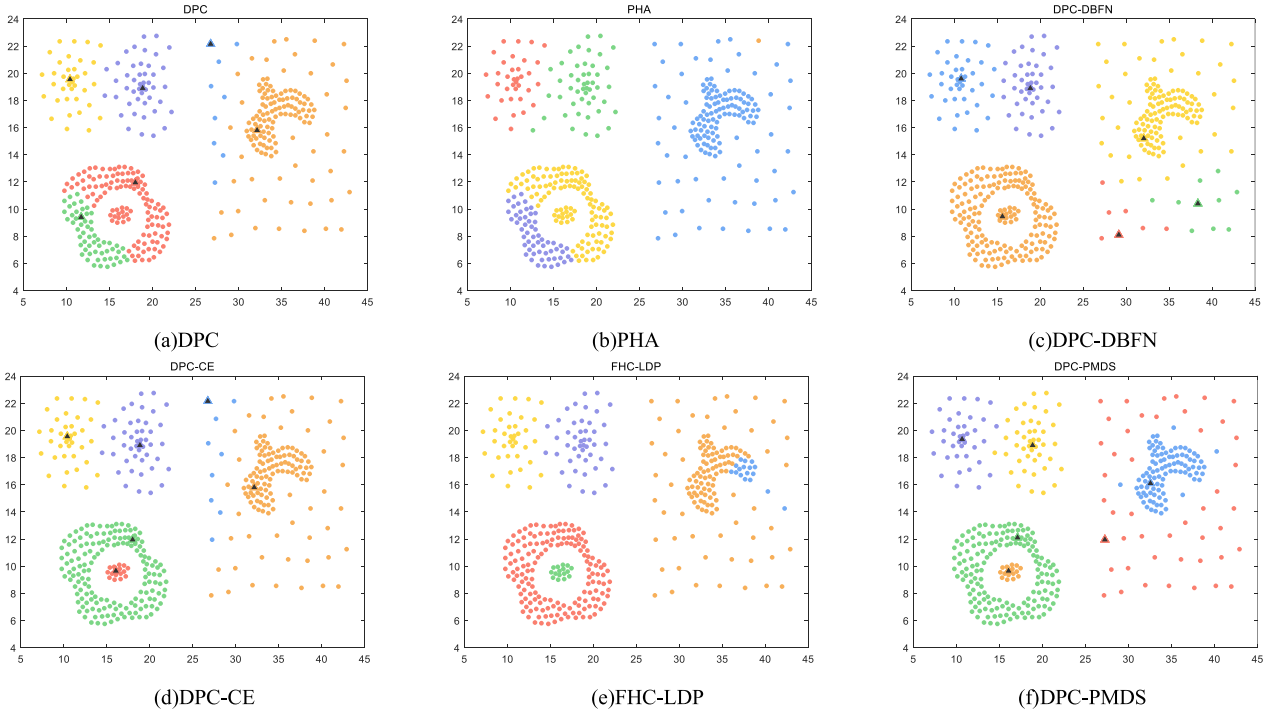


FIGURE 11. Clustering results of (a) DPC, (b)PHA, (c)DPC-DBFN, (d)DPC-CE, (e)FHC-LDP and (f) the proposed method (DPC-PMDS) clustering methods on the Compound dataset.

datasets, DPC-PMDS also has slightly better performance than the other algorithms. On the D31 dataset, the clustering results of DPC-PMDS are also acceptable.

In general, the DPC-PMDS algorithm proposed in this paper obtains the best clustering result on 12 synthetic datasets, and DPC-PMDS performs better on datasets with

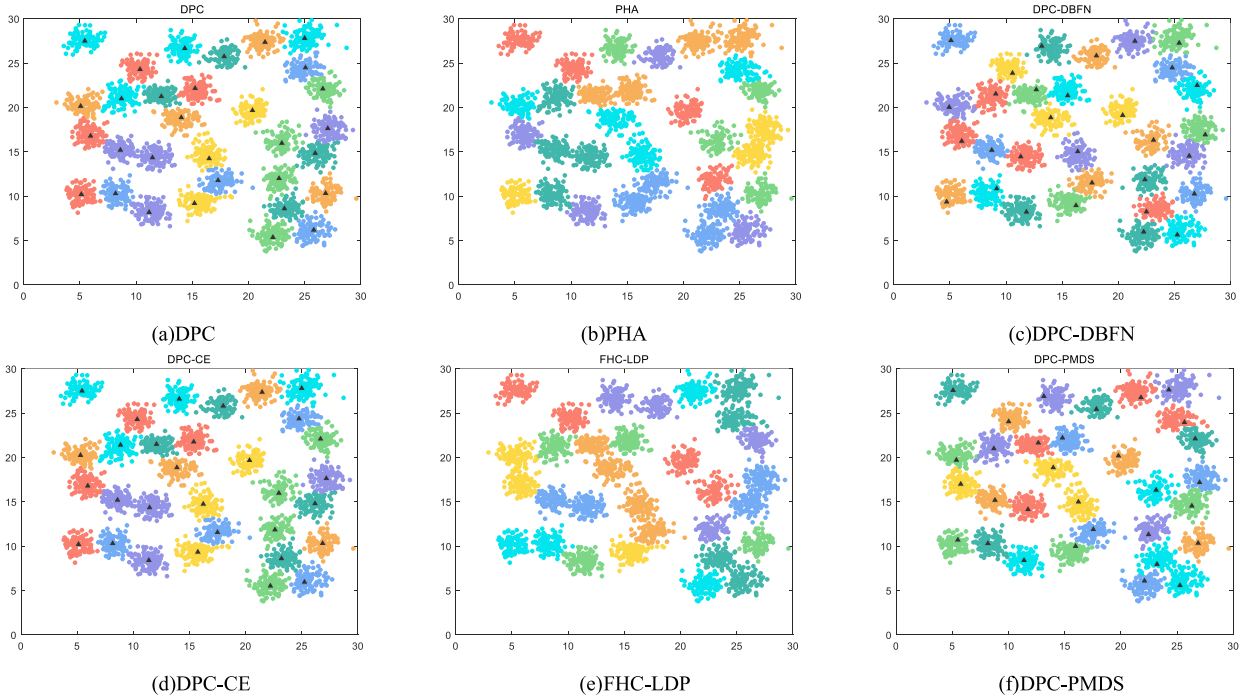


FIGURE 12. Clustering results of (a) DPC, (b)PHA, (c)DPC-DBFN, (d)DPC-CE, (e)FHC-LDP and (f) the proposed method (DPC-PMDS) clustering methods on the D31 dataset.

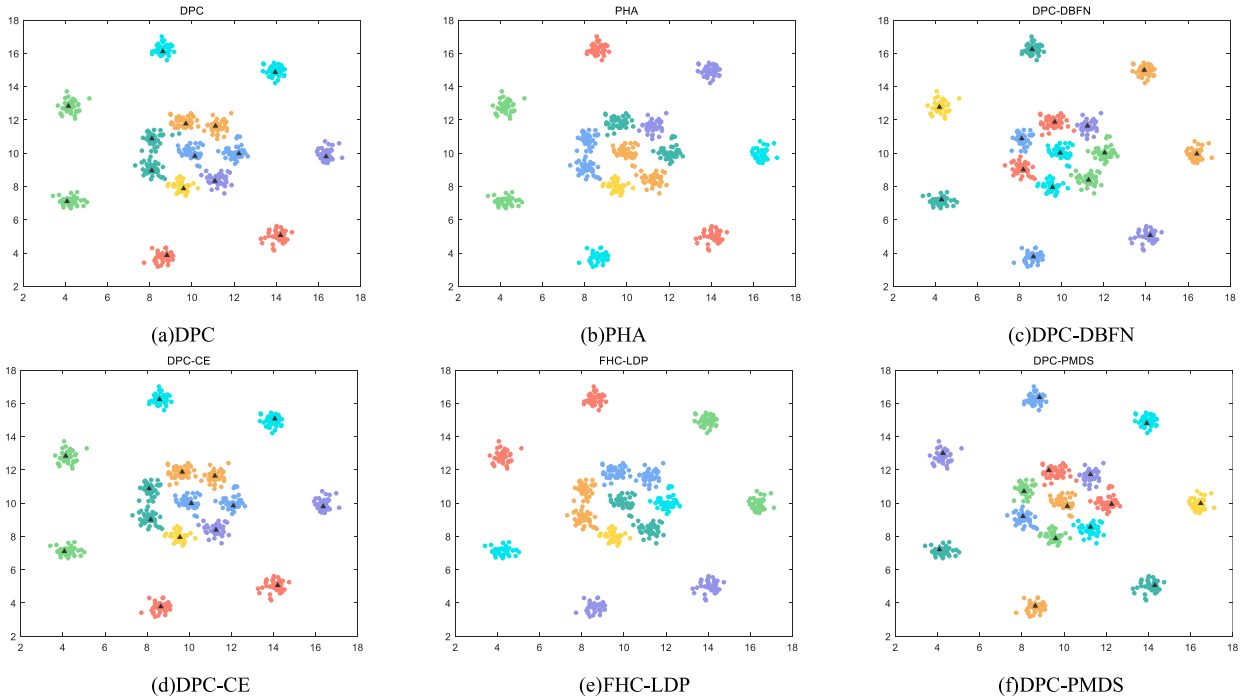


FIGURE 13. Clustering results of (a) DPC, (b)PHA, (c)DPC-DBFN, (d)DPC-CE, (e)FHC-LDP and (f) the proposed method (DPC-PMDS) clustering methods on the R15 dataset.

complex shapes and uneven density distribution compared to the other algorithms.

D. RESULTS ON UCI DATASETS

In this subsection, experiments are conducted on eight UCI datasets. Table 5 shows the best clustering results of all algorithms on the eight UCI datasets, and the optimal values of the evaluation metrics are in bold.

These UCI datasets contain different dimensions and numbers of instances, among which Seeds and Thyroid have 210 and 215 instances with 7 and 5 features, respectively. Diabetes has 768 instances with 8 features, and Cloud has

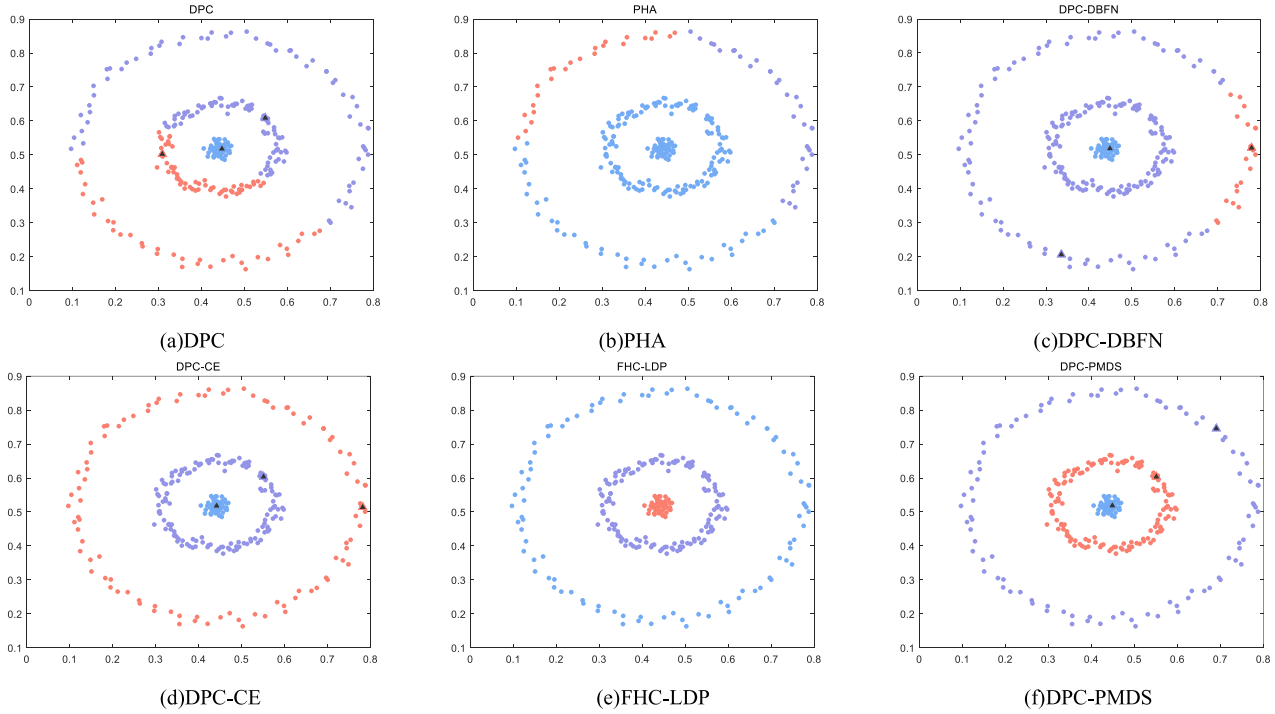


FIGURE 14. Clustering results of (a) DPC, (b)PHA, (c)DPC-DBFN, (d)DPC-CE, (e)FHC-LDP and (f) the proposed method (DPC-PMDS) clustering methods on the Threecircles dataset.

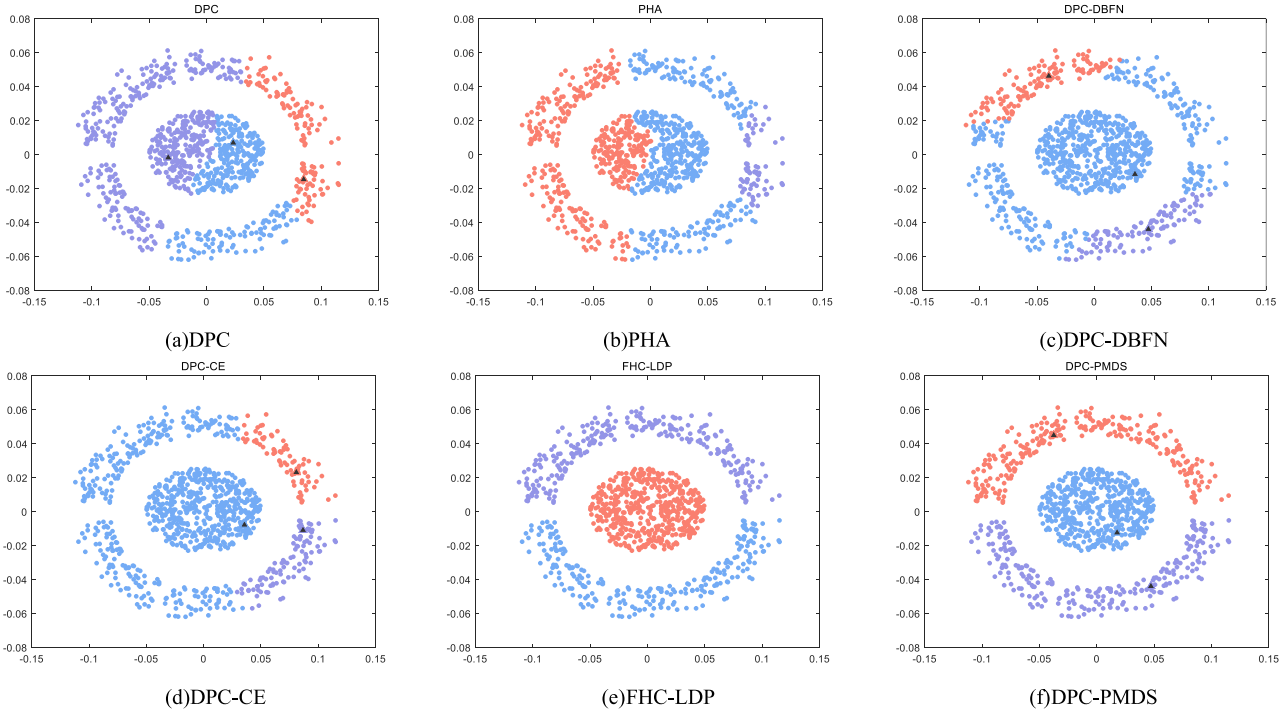


FIGURE 15. Clustering results of (a) DPC, (b)PHA, (c)DPC-DBFN, (d)DPC-CE, (e)FHC-LDP and (f) the proposed method (DPC-PMDS) clustering methods on the CMC dataset.

1024 instances and 10 features. On Seeds, Diabetes, Thyroid, and Cloud datasets, the values of ACC, NMI, RI, and ARI of DPC-PMDS are significantly higher than those of other algorithms, which indicates that DPC-PMDS has better clustering result compared to other algorithms. The Iris dataset contains 4 features and 150 instances. On the Iris dataset,

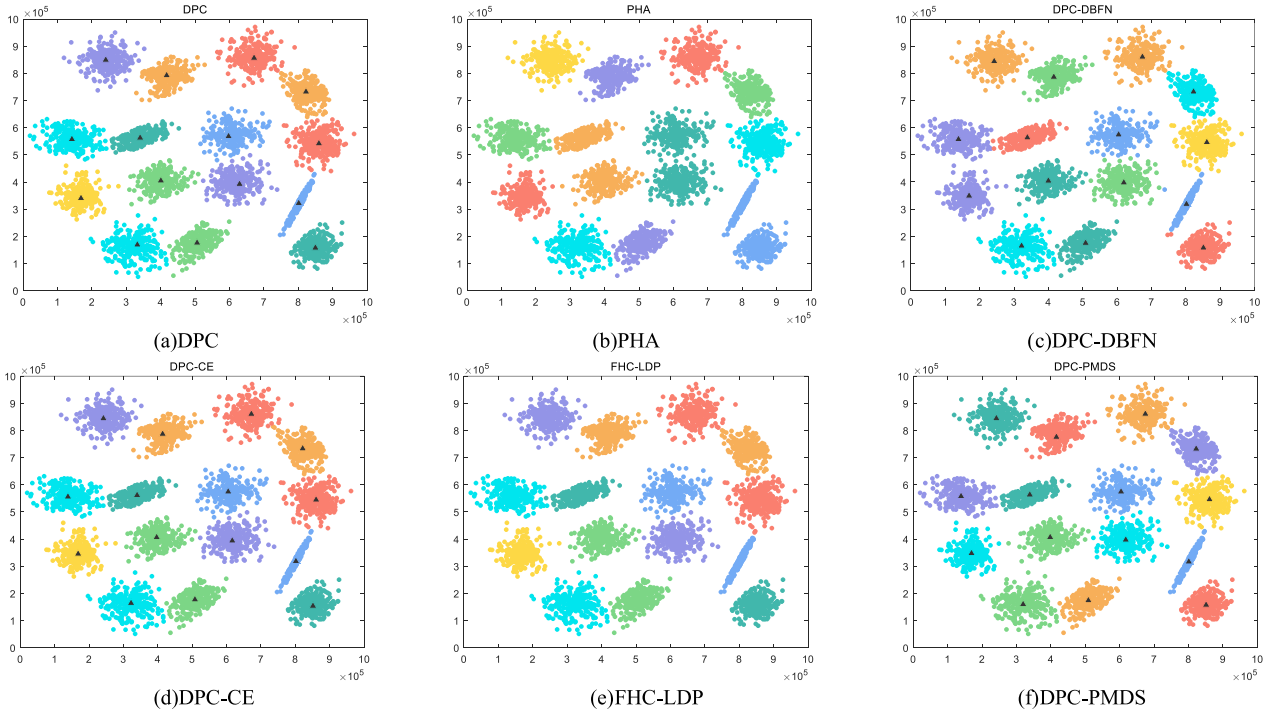


FIGURE 16. Clustering results of (a) DPC, (b)PHA, (c)DPC-DBFN, (d)DPC-CE, (e)FHC-LDP and (f) the proposed method (DPC-PMDS) clustering methods on the S1 dataset.

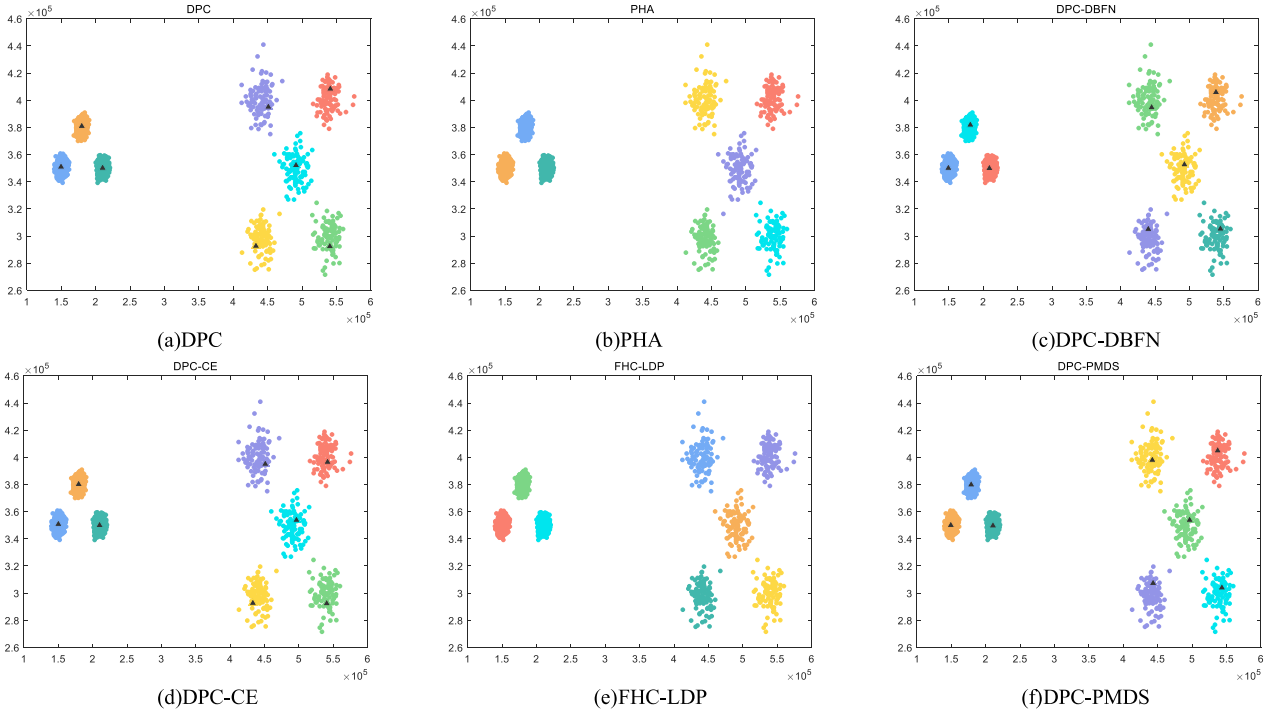


FIGURE 17. Clustering results of (a) DPC, (b)PHA, (c)DPC-DBFN, (d)DPC-CE, (e)FHC-LDP and (f) the proposed method (DPC-PMDS) clustering methods on the Unbalance dataset.

the clustering result of DPC-PMDS is slightly lower than that of FHC-LDP but better than that of other algorithms. DNA is a high-dimensional dataset with 2000 instances and 180 features. On the DNA dataset, the NMI value of DPC-PMDS is much higher than that of other algorithms, and the maximum values of ARI, RI, and ACC are

TABLE 4. Clustering results of algorithms on twelve synthetic datasets.

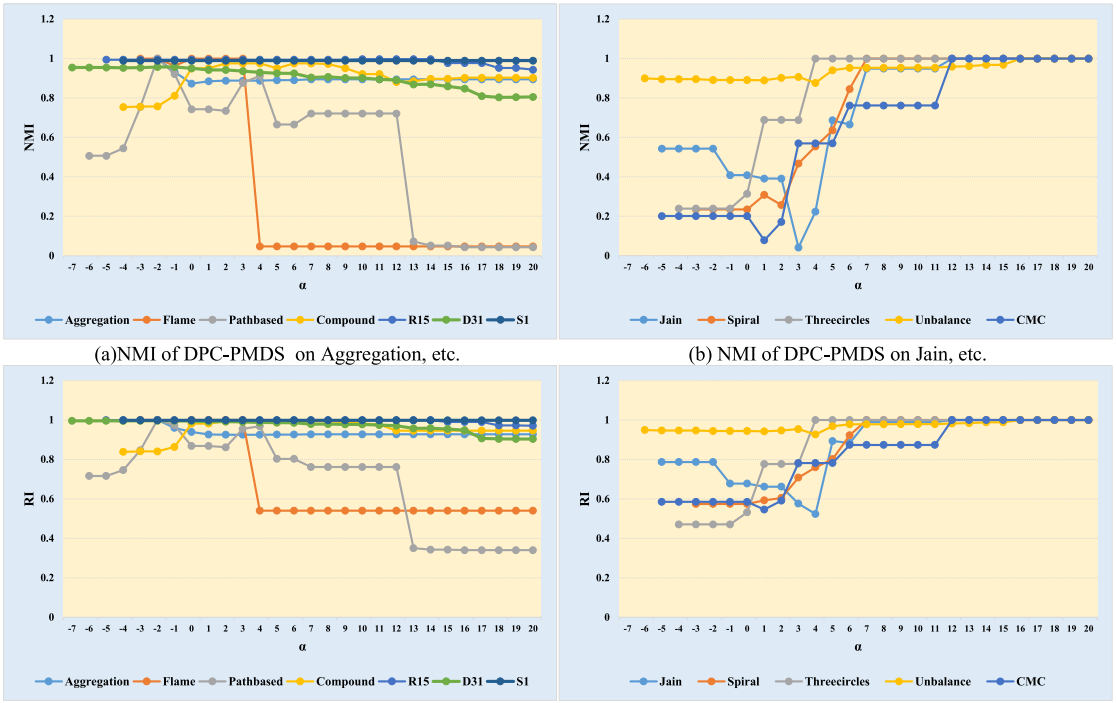
Algorithm	Datasets	ACC	NMI	RI	ARI	Par	Datasets	ACC	NMI	RI	ARI	Par
DPC	*Aggregation*	0.9987	0.9957	0.9993	0.9978	2%	*Flame*	0.8458	0.5048	0.7381	0.4763	3%
PHA		**1**	**1**	**1**	**1**	10		0.8292	0.4763	0.7155	0.4310	10
DPC-DBFN		0.9962	0.9884	0.9973	0.9920	33		0.9917	0.9355	0.9834	0.9666	27
DPC-CE		0.9987	0.9957	0.9993	0.9978	2%/0.25/ 0.3		**1**	**1**	**1**	**1**	2%/0.25/ 0.3
FHC-LDP		**1**	**1**	**1**	**1**	15		**1**	**1**	**1**	**1**	15
DPC-PMDS		**1**	**1**	**1**	**1**	-2		**1**	**1**	**1**	**1**	-2
DPC	*Jain*	0.9035	0.5972	0.8251	0.6438	1.5%	*Spiral*	0.5513	0.3635	0.6685	0.2992	1%
PHA		0.6220	0.2315	0.5285	0.0408	10		**1**	**1**	**1**	**1**	10
DPC-DBFN		0.9383	0.6932	0.8840	0.7611	16		0.4679	0.2040	0.4270	0.0292	19
DPC-CE		**1**	**1**	**1**	**1**	2%/0.25/ 0.3		**1**	**1**	**1**	**1**	2%/0.25/ 0.3
FHC-LDP		**1**	**1**	**1**	**1**	15		**1**	**1**	**1**	**1**	9
DPC-PMDS		**1**	**1**	**1**	**1**	13		**1**	**1**	**1**	**1**	13
DPC	*Pathbased*	0.7400	0.5437	0.7452	0.4572	2%	*D31*	0.9697	0.9597	0.9962	0.9390	2%
PHA		0.6567	0.4946	0.7072	0.4021	10		0.9639	0.9537	0.9955	0.9278	10
DPC-DBFN		0.7267	0.5354	0.7386	0.4470	20		**0.9752**	**0.9654**	**0.9969**	**0.9497**	11
DPC-CE		0.5700	0.5070	0.6979	0.3830	2%/0.25/ 0.3		0.9690	0.9587	0.9961	0.9378	2%/0.25/ 0.3
FHC-LDP		0.7333	0.7144	0.7806	0.5629	8		0.9700	0.9599	0.9963	0.9396	38
DPC-PMDS		**1**	**1**	**1**	**1**	-2		0.9677	0.9569	0.9960	0.9352	-2
DPC	*Compound*	0.7118	0.7872	0.8540	0.6026	2%	*R15*	0.9933	0.9893	0.9983	0.9857	1.5%
PHA		0.6867	0.7805	0.8467	0.5906	10		0.9967	0.9942	0.9991	0.9928	10
DPC-DBFN		0.8571	0.8490	0.9209	0.8041	3		**0.9983**	**0.9971**	**0.9996**	**0.9964**	39
DPC-CE		0.8947	0.9043	0.9468	0.8650	2%/0.25/ 0.3		0.9967	0.9942	0.9991	0.9928	2%/0.25/ 0.3
FHC-LDP		0.8396	0.8779	0.9363	0.8358	8		0.9967	0.9942	0.9991	0.9928	10
DPC-PMDS		**0.9900**	**0.9755**	**0.9930**	**0.9814**	3		**0.9983**	**0.9971**	**0.9996**	**0.9964**	12
DPC	*Threecircles*	0.6154	0.4854	0.6825	0.3156	0.5%	*CMC*	0.4800	0.2093	0.5847	0.1278	2%
PHA		0.5753	0.3307	0.5537	0.1815	10		0.4701	0.0641	0.5406	0.0547	10
DPC-DBFN		0.7391	0.6489	0.7199	0.4606	18		0.7275	0.4131	0.6289	0.3006	22
DPC-CE		**1**	**1**	**1**	**1**	2%/0.25/ 0.3		0.6776	0.3413	0.5692	0.2140	2%/0.25/ 0.3
FHC-LDP		**1**	**1**	**1**	**1**	15		**1**	**1**	**1**	**1**	11
DPC-PMDS		**1**	**1**	**1**	**1**	13		**1**	**1**	**1**	**1**	13
DPC	*Unbalance*	**1**	**1**	**1**	**1**	2%	*s1*	**0.9952**	0.9896	**0.9987**	**0.9897**	2.5%
PHA		0.9998	0.9994	**1**	**1**	10		0.9942	0.9882	0.9985	0.9877	10
DPC-DBFN		**1**	**1**	**1**	**1**	23		0.9944	0.9878	0.9985	0.9880	28
DPC-CE		**1**	**1**	**1**	**1**	2%/0.25/ 0.3		**0.9952**	0.9895	**0.9987**	**0.9897**	2%/0.25/ 0.3
FHC-LDP		**1**	**1**	**1**	**1**	65		0.9950	0.9893	**0.9987**	0.9893	63
DPC-PMDS		**1**	**1**	**1**	**1**	17		**0.9952**	**0.9898**	**0.9987**	**0.9897**	3

FIGURE 18. Clustering results of DPC-PMDS on the Olivetti Faces dataset.

achieved by DPC-DBFN, DPC-CE, and FHC-LDP, respectively. Abalone is a large-scale dataset with 4177 instances and 7 features. On the Abalone dataset, DPC-PMDS has the highest NMI and ARI values. Although the values of ACC and RI of DPC-PMDS are slightly lower than those of FHC-LDP, they are higher than those of other algorithms. Robot navigation is a large-scale high-dimensional dataset containing 5456 instances and 24 features, and the NMI value of DPC-PMDS is the highest on this dataset. The experimental results show that the clustering result of the DPC-PMDS algorithm proposed in this paper is overall optimal on the UCI dataset, and the DPC-PMDS algorithm can

TABLE 5. Clustering results of algorithms on eight UCI datasets.

Algorithm	Datasets	ACC	NMI	RI	ARI	Par	Datasets	ACC	NMI	RI	ARI	Par
DPC	*Iris*	0.9067	0.8058	0.8923	0.7592	1%	*Diabetes*	0.6589	0.0187	0.5499	0.0196	0.5%
PHA		0.6800	0.7355	0.7766	0.5638	10		0.6523	0.0171	0.5458	0.0023	10
DPC-DBFN		0.9467	0.8366	0.9341	0.8510	21		0.6380	0.0135	0.5375	-0.0104	3
DPC-CE		0.8533	0.7544	0.8464	0.6634	2%/0.25/ 0.3		0.5169	0.0009	0.4999	-0.0000	2%/0.25/ 0.3
FHC-LDP		**0.9667**	**0.8851**	**0.9575**	**0.9038**	12		0.6497	0.0052	0.5443	0.0143	11
DPC-PMDS		0.9600	0.8642	0.9495	0.8857	0		**0.6901**	**0.0845**	**0.5717**	**0.1393**	7
DPC	*Seeds*	0.8952	0.7126	0.8766	0.7227	2%	*Thyroid*	0.7209	0.1851	0.5794	0.1114	2%
PHA		0.8381	0.6362	0.8226	0.6026	10		0.7023	0.0843	0.5441	0.0313	10
DPC-DBFN		0.9143	0.7343	0.8964	0.7664	2		0.8605	0.5323	0.7752	0.5383	9
DPC-CE		0.9000	0.6857	0.8799	0.7288	2%/0.25/ 0.3		0.7209	0.1851	0.5794	0.1114	2%/0.25/ 0.3
FHC-LDP		0.9000	0.6946	0.8797	0.7289	14		0.6558	0.3868	0.6077	0.2215	9
DPC-PMDS		**0.9286**	**0.7524**	**0.9113**	**0.7995**	0		**0.9442**	**0.7280**	**0.9073**	**0.8130**	29
DPC	*DNA*	0.4940	0.0226	0.5169	0.0398	0.5%	*Abalone*	0.2183	0.1911	0.4827	0.0553	1%
PHA		0.5245	0.0129	0.3888	-0.0009	10		0.2004	0.1915	0.3507	0.0373	10
DPC-DBFN		0.5270	0.0782	0.5299	**0.1049**	3		0.1882	0.1167	0.3042	0.0139	4
DPC-CE		0.4850	0.0095	**0.5403**	0.0289	2%/0.25/ 0.3		0.1700	0.0541	0.1363	0.0023	2%/0.25/ 0.3
FHC-LDP		**0.5490**	0.0500	0.4399	0.0401	32		**0.2456**	0.1794	**0.7428**	0.0659	71
DPC-PMDS		0.4960	**0.2136**	0.4933	0.0333	-5		0.2356	**0.1979**	0.6930	**0.0723**	-4
DPC	*Cloud*	0.6143	0.0191	0.5256	0.0023	2%	*Robot navigation*	**0.4833**	0.1578	0.5537	**0.0852**	0.5%
PHA		0.6143	0.0191	0.5256	0.0023	10		0.3794	0.0527	0.4656	-0.0174	10
DPC-DBFN		0.5234	0.1422	0.5006	-0.0257	2		0.3673	0.0968	**0.5949**	0.0507	21
DPC-CE		0.6143	0.0191	0.5256	0.0023	2%/0.25/ 0.3		0.4597	0.1448	0.5529	0.0757	2%/0.25/ 0.3
FHC-LDP		0.8223	0.4422	0.7074	0.4145	23		0.4054	0.1082	0.4698	0.0205	65
DPC-PMDS		**0.8613**	**0.4612**	**0.7609**	**0.5218**	-3		0.4307	**0.1625**	0.4816	0.0680	16



(a)NMI of DPC-PMDS on Aggregation, etc.

(b) NMI of DPC-PMDS on Jain, etc.

(c)RI of DPC-PMDS on Aggregation, etc.

(d) RI of DPC-PMDS on Jain, etc.

FIGURE 19. NMI and RI values of DPC-PMDS with different α values on synthetic datasets.

handle large-scale and high-dimensional real-world datasets relatively well.

E. RESULTS ON OLIVETTI FACES DATASET

The Olivetti Faces dataset has a total of 400 different face images, with 92×112 features per instance. It is a commonly used dataset for clustering. We selected its top 100 images for experiments to verify the performance of the proposed algorithm on the image dataset. The values of ACC, NMI, RI, and ARI of DPC-PMDS on the top 100 face images of the Olivetti Faces dataset are 0.9900, 0.9857, 0.9962, and 0.9768, respectively. Fig.18 shows the visualization results. Different colors represent different clusters. From Fig.18, it can be seen that only one face is not successfully assigned. Thereby

DPC-PMDS algorithm can obtain valid clustering results on the Olivetti Faces dataset.

F. SENSITIVITY ANALYSIS OF THE PROPOSED METHOD

In this subsection, the effect of the parameter α on the clustering result of the DPC-PMDS algorithm will be analyzed. Fig.19 shows the NMI and RI values of DPC-PMDS with different values of α on the synthetic datasets.

According to the previous section, it can be known that the value of α determines the diffusion strength value, and the diffusion strength determines the size and structure of the initial clusters. Therefore, if the value of α is too small, i.e., the diffusion strength is too small, the initial clusters generated according to the label diffusion rule cannot contain enough core points. Then some points will be assigned incorrectly on non-spherical datasets with complex shapes, which is because the boundary points are assigned by a distance-based strategy. In Fig. 19, we can see that when the α takes a small value, the clustering results on the Pathbased, Compound, Jain, Spiral, Threecircles, and CMC datasets are not optimal.

Through experiments, we find that the parameter α has different effects on datasets with different characteristics. From Fig.19(a) and Fig.19(c), it can be found that for the dataset with intersections between clusters, a slightly smaller value of α in general leads to better clustering results. This is because if the value of α is large, the large diffusion strength will cause excessive label diffusion and easily connect the points of multiple clusters. This situation is especially obvious on the dataset with intersections between clusters. From Fig. 19(b) and Fig. 19(d), it can be found that for the dataset with no intersection between clusters, better results are generally obtained with a slightly larger value of α. This is because on datasets without clusters intersection, a slightly larger α will facilitate label diffusion so that the initial cluster can better reflect the cluster structure.

In summary, the parameter α is less sensitive on spherical clusters. On a dataset that the clusters do not intersect, a slightly larger α can get better clustering results. On a dataset with clusters intersection, the value of α should be slightly smaller.

V. CONCLUSION

In this paper, an improved density peaks clustering algorithm based on the potential model and diffusion strength is proposed. The main purpose of this paper is to avoid the dependence of DPC on the parameter d_c, to improve the accuracy of the selection of cluster centers on datasets with complex shapes and uneven density distribution, and to reduce the chain reaction. The potential and centrality of data points are used to calculate the density. We present the concept of diffusion strength and the label diffusion rule. By considering the diffusion strength, DPC-PMDS can accurately select the cluster centers of datasets with uneven density distribution. The initial clusters consisting of centers and core points can reflect the core structure of clusters well. The motivation for this study is that obtaining the core structure of clusters usually leads to great clustering results.

The performance of DPC-PMDS is compared with DPC, PHA, DPC-DBFN, DPC-CE, and FHC-LDP algorithms on synthetic and UCI datasets, and the performance of DPC-PMDS on image dataset is examined with the first 100 face images from the Olivetti Faces dataset. The experimental results indicate that the proposed DPC-PMDS algorithm exhibits good clustering effectiveness on all datasets.

This work can be further improved in the following two directions. The first is the adaptive selection of parameter since the optimal value of parameter is different for datasets with different characteristics. DPC-PMDS uses Euclidean distance for similarity calculation of data points, which is not suitable for high-dimensional datasets. Therefore, we will look for new similarity measure suitable for high-dimensional datasets to further improve the effectiveness of the algorithm.

REFERENCES

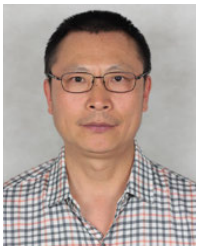
[1] K. P. Sinaga and M.-S. Yang, "Unsupervised K-Means clustering algorithm," *IEEE Access*, vol. 8, pp. 80716–80727, 2020, doi: 10.1109/ACCESS.2020.2988796.
[2] Q. Xu, Q. Zhang, J. Liu, and B. Luo, "Efficient synthetical clustering validity indexes for hierarchical clustering," *Expert Syst. Appl.*, vol. 151, Aug. 2020, Art. no. 113367, doi: 10.1016/j.eswa.2020.113367.
[3] M. Tareq, E. A. Sundararajan, A. Harwood, and A. A. Bakar, "A systematic review of density grid-based clustering for data streams," *IEEE Access*, vol. 10, pp. 579–596, 2021, doi: 10.1109/ACCESS.2021.3134704.
[4] M.-S. Yang, S.-J. Chang-Chien, and Y. Nataliani, "Unsupervised fuzzy model-based Gaussian clustering," *Inf. Sci.*, vol. 481, pp. 1–23, May 2019, doi: 10.1016/j.ins.2018.12.059.
[5] R. J. G. B. Campello, P. Kröger, J. Sander, and A. Zimek, "Density-based clustering," *WIREs Data Mining Knowl. Discovery*, vol. 10, no. 2, Mar. 2020, Art. no. e1343, doi: 10.1002/widm.1343.
[6] T. Lei, P. Liu, X. Jia, X. Zhang, H. Meng, and A. K. Nandi, "Automatic fuzzy clustering framework for image segmentation," *IEEE Trans. Fuzzy Syst.*, vol. 28, no. 9, pp. 2078–2092, Sep. 2020, doi: 10.1109/TFUZZ.2019.2930030.
[7] L. Zhou and W. Wei, "DIC: Deep image clustering for unsupervised image segmentation," *IEEE Access*, vol. 8, pp. 34481–34491, 2020, doi: 10.1109/ACCESS.2020.2974496.
[8] B. Gohain, R. Chutia, and P. Dutta, "Distance measure on intuitionistic fuzzy sets and its application in decision-making, pattern recognition, and clustering problems," *Int. J. Intell. Syst.*, vol. 37, no. 3, pp. 2458–2501, Mar. 2022, doi: 10.1002/int.22780.
[9] H. H. Wu, G. Ke, Y. Wang, and Y. T. Chang, "Prediction on recommender system based on bi-clustering and moth flame optimization," *Appl Soft Comput.*, vol. 120, May 2022, doi: 10.1016/j.asoc.2022.108626.
[10] Y. Yang, P. Yin, Z. Luo, W. Gu, R. Chen, and Q. Wu, "Informative feature clustering and selection for gene expression data," *IEEE Access*, vol. 7, pp. 169174–169184, 2019, doi: 10.1109/ACCESS.2019.2952548.
[11] C. Zhang, M. Ni, H. Yin, and K. Qiu, "Developed density peak clustering with support vector data description for access network intrusion detection," *IEEE Access*, vol. 6, pp. 46356–46362, 2018, doi: 10.1109/ACCESS.2018.2866128.
[12] M. Ester, H.-P. Kriegel, J. Sander, and X. Xu, "A density-based algorithm for discovering clusters in large spatial databases with noise," in *Proc. KDD*, 1996, pp. 226–231.
[13] A. Rodriguez and A. Laio, "Clustering by fast search and find of density peaks," *Science*, vol. 344, no. 6191, pp. 1492–1496, Jun. 2014, doi: 10.1126/science.1242072.
[14] 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," *Inf. Sci.*, vol. 354, pp. 19–40, Aug. 2016, doi: 10.1016/j.ins.2016.03.011.
[15] K. G. Flores and S. E. Garza, "Density peaks clustering with gap-based automatic center detection," *Knowl.-Based Syst.*, vol. 206, Oct. 2020, Art. no. 106350, doi: 10.1016/j.knosys.2020.106350.

[16] D. Jiang, W. Zang, R. Sun, Z. Wang, and X. Liu, "Adaptive density peaks clustering based on K-nearest neighbor and Gini coefficient," *IEEE Access*, vol. 8, pp. 113900–113917, 2020, doi: 10.1109/ACCESS.2020.3003057.
[17] T. Gao, D. Chen, Y. Tang, B. Du, R. Ranjan, A. Y. Zomaya, and S. Dustdar, "Adaptive density peaks clustering: Towards exploratory EEG analysis," *Knowl.-Based Syst.*, vol. 240, Mar. 2022, Art. no. 108123, doi: 10.1016/j.knosys.2022.108123.
[18] A. Lotfi, P. Moradi, and H. Beigy, "Density peaks clustering based on density backbone and fuzzy neighborhood," *Pattern Recognit.*, vol. 107, Nov. 2020, Art. no. 107449, doi: 10.1016/j.patcog.2020.107449.
[19] L. Sun, X. Qin, W. Ding, J. Xu, and S. Zhang, "Density peaks clustering based on k-nearest neighbors and self-recommendation," *Int. J. Mach. Learn. Cybern.*, vol. 12, no. 7, pp. 1913–1938, Jul. 2021, doi: 10.1007/s13042-021-01284-x.
[20] W. Guo, W. Wang, S. Zhao, Y. Niu, Z. Zhang, and X. Liu, "Density peak clustering with connectivity estimation," *Knowl-Based Syst*, vol. 243, May 2022, Art. no. 108501, doi: 10.1016/j.knosys.2022.108501.
[21] F. Li, M. Zhou, S. Li, and T. Yang, "A new density peak clustering algorithm based on cluster fusion strategy," *IEEE Access*, vol. 10, pp. 98034–98047, 2022, doi: 10.1109/ACCESS.2022.3205742.
[22] J. Guan, S. Li, X. He, J. Zhu, and J. Chen, "Fast hierarchical clustering of local density peaks via an association degree transfer method," *Neurocomputing*, vol. 455, pp. 401–418, Sep. 2021, doi: 10.1016/j.neucom.2021.05.071.
[23] Y. Lu and Y. Wan, "Clustering by sorting potential values (CSPV): A novel potential-based clustering method," *Pattern Recognit.*, vol. 45, no. 9, pp. 3512–3522, Sep. 2012, doi: 10.1016/j.patcog.2012.02.035.
[24] Y. Lu and Y. Wan, "PHA: A fast potential-based hierarchical agglomerative clustering method," *Pattern Recognit.*, vol. 46, no. 5, pp. 1227–1239, May 2013, doi: 10.1016/j.patcog.2012.11.017.
[25] Z.-K. Zhang, C. Liu, X.-X. Zhan, L. Lu, C.-X. Zhang, and Y.-C. Zhang, "Dynamics of information diffusion and its applications on complex networks," *Phys. Rep.*, vol. 651, no. 7, pp. 1–34, Jul. 2016, doi: 10.1016/j.physrep.2016.07.002.
[26] D. Cai, X. He, and J. Han, "Document clustering using locality preserving indexing," *IEEE Trans. Knowl. Data Eng.*, vol. 17, no. 12, pp. 1624–1637, Dec. 2005, doi: 10.1109/TKDE.2005.198.
[27] D. Pfitzner, R. Leibbrandt, and D. Powers, "Characterization and evaluation of similarity measures for pairs of clusterings," *Knowl. Inf. Syst.*, vol. 19, no. 3, pp. 361–394, Jun. 2009, doi: 10.1007/s10115-008-0150-6.
[28] W. M. Rand, "Objective criteria for the evaluation of clustering methods," *J. Amer. Statist. Assoc.*, vol. 66, no. 336, pp. 846–850, 1971, doi: 10.1080/01621459.1971.10482356.
[29] P. Fränti, M. Rezaei, and Q. Zhao, "Centroid index: Cluster level similarity measure," *Pattern Recognit.*, vol. 47, no. 9, pp. 3034–3045, Sep. 2014, doi: 10.1016/j.patcog.2014.03.017.
[30] A. Gionis, H. Mannila, and P. Tsaparas, "Clustering aggregation," in *Proc. 21st Int. Conf. Data Eng. (ICDE)*, Apr. 2005, pp. 341–352, doi: 10.1109/ICDE.2005.34.
[31] L. Fu and E. Medico, "FLAME, a novel fuzzy clustering method for the analysis of DNA microarray data," *BMC Bioinf.*, vol. 8, no. 1, pp.1–15, Jan. 2007, doi: 10.1186/1471-2105-8-3.
[32] A. K. Jain and M. H. C. Law, "Data clustering: A user's dilemma," in *Pattern Recognition and Machine Intelligence*. Berlin, Germany: Springer, 2005, pp. 1–10.
[33] H. Chang and D.-Y. Yeung, "Robust path-based spectral clustering," *Pattern Recognit.*, vol. 41, no. 1, pp. 191–203, Jan. 2008, doi: 10.1016/j.patcog.2007.04.010.
[34] C. T. Zahn, "Graph-theoretical methods for detecting and describing gestalt clusters," *IEEE Trans. Comput.*, vol. C-20, no. 1, pp. 68–86, Jan. 1971, doi: 10.1109/T-C.1971.223083.
[35] C. J. Veenman, M. J. T. Reinders, and E. Backer, "A maximum variance cluster algorithm," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 24, no. 9, pp. 1273–1280, Sep. 2002, doi: 10.1109/TPAMI.2002.1033218.
[36] L. Zelnik-Manor and P. Perona, "Self-tuning spectral clustering," Presented at the of the 17th Int. Conf. Neural Inf. Process. Syst., Vancouver, BC, Canada, 2004.
[37] S. A. Seyedi, A. Lotfi, P. Moradi, and N. N. Qader, "Dynamic graph-based label propagation for density peaks clustering," *Expert Syst. Appl.*, vol. 115, pp. 314–328, Jan. 2019, doi: 10.1016/j.eswa.2018.07.075.
[38] P. Fränti and O. Virmajoki, "Iterative shrinking method for clustering problems," *Pattern Recognit.*, vol. 39, no. 5, pp. 761–775, May 2006, doi: 10.1016/j.patcog.2005.09.012.
[39] M. Rezaei and P. Fränti, "Set matching measures for external cluster validity," *IEEE Trans. Knowl. Data Eng.*, vol. 28, no. 8, pp. 2173–2186, Aug. 2016, doi: 10.1109/TKDE.2016.2551240.
[40] F. S. Samaria and A. C. Harter, "Parameterisation of a stochastic model for human face identification," in *Proc. IEEE Workshop Appl. Comput. Vis.*, Dec. 1994, pp. 138–142, doi: 10.1109/ACV.1994.341300.

JING CHE was born in 2000. She is currently pursuing the M.S. degree with the School of Business, Shandong Normal University, Jinan, China. Her current research interests include machine learning and data mining.

WENKE ZANG received the M.S. and Ph.D. degrees from Shandong Normal University, China, in 2005 and 2018, respectively. He is currently a Professor and the Doctoral Supervisor with Shandong Normal University. His research interests include machine learning, data mining, and optimization algorithm.

JINGWEN XIONG was born in 1999. She is currently pursuing the master's degree with the School of Business, Shandong Normal University, China. Her research interests include artificial intelligence, genetic algorithm, data mining, and machine learning.



XIYU LIU (Member, IEEE) received the Ph.D. degree in mathematical sciences from Shandong University, in 1990. He is currently a Professor, the Doctoral Supervisor, and the Dean of the Institute of Management Science, Shandong Normal University, China. His current research interests include membrane computing and data mining.