IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 20 April 2024, accepted 14 May 2024, date of publication 24 May 2024, date of current version 5 June 2024.

Digital Object Identifier 10.1109/ACCESS.2024.3404917

RESEARCH ARTICLE

SM-DPC: Clustering by Fast Search and Find of Density Peaks Based on SNN With Multi-Cluster Fusion Strategy

SHIBO ZHOU, BINGBING PENG, WENPENG XU, AND LÜZHEN REN
Navigation College, Jimei University, Xiamen 361021, China
Corresponding author: Lüzhen Ren (lzren@jmu.edu.cn)

ABSTRACT The Clustering by Fast Search and Find of Density Peaks (DPC) algorithm is a clustering method that automatically identifies clustering centers based on density and relative distance. It has several advantages, including the ability to identify arbitrarily shaped clusters and requiring few input parameters. However, the density measure used in DPC does not consider the spatial distribution characteristics of the sample points in the data set. The clustering performance is suboptimal for datasets with significant differences in cluster density. Additionally, its non-central sample point assignment method is less error-tolerant, which can result in successive assignment errors and a domino effect, ultimately leading to poor clustering accuracy. To address these shortcomings, we propose an improved DPC algorithm based on shared nearest neighbor and multi-cluster fusion (SM-DPC). The local density of sample points is redefined using K-nearest neighbors, which makes the density metric more consistent with the local structural characteristics of the dataset. A two-step allocation strategy for non-central sample points based on shared nearest neighbors is proposed to improve the accuracy of allocation of non-central sample points. A multi-cluster fusion strategy is used to correct the centroid selection bias for datasets where sample points are not uniformly distributed. The experimental results demonstrate that SM-DPC is capable of clustering datasets with arbitrary shape and density distributions effectively. Furthermore, it exhibits superior performance and broader adaptability to different types of datasets compared to DBSCAN, K-means algorithms, and other DPC optimization algorithms.

INDEX TERMS K-nearest neighbor, local density, multi-cluster fusion, density peaks.

I. INTRODUCTION

The extraction of useful knowledge from the vast amount of data generated by the rapid development of information technology and the application of the Internet in industry and daily life is a current research hotspot in data mining. Kaushik et al. [1], through meticulous review and analysis of extensive literature, provide a profound exploration and examination of NARM. Such in-depth investigation equips researchers with a deeper understanding of the essence and potential of data mining technology, furnishing them with abundant reference information and research perspectives in the domain of data mining. Furthermore, it offers valuable guidance and insights for future research and practice endeavors. After delving into the profound insights elucidated by Minakshi et al., it becomes apparent that a comprehensive comprehension of the intricate nature inherent in data mining techniques serves as a robust cornerstone for subsequent investigations. Armed with this comprehension, we redirect our focus towards elucidating the pivotal role occupied by cluster analysis in elucidating patterns and interrelationships embedded within datasets.

Cluster analysis is an essential unsupervised learning method that typically involves partitioning a dataset into different clusters by computing the similarity between sample points. This ensures that sample points within the same cluster exhibit similar features, while those in different clusters demonstrate distinct features. Kaushik et al. [2] has

The associate editor coordinating the review of this manuscript and approving it for publication was Juan A. Lara.

also proposed discretizing numerical attributes, aiding in simplifying data processing and enhancing computational efficiency, thereby facilitating better exploration of potential connections among sample points within the dataset. Such techniques find widespread applications in various fields, including bioinformatics [3], image processing [4], [5], and pattern recognition [6].

Moreover, beyond its scholarly utility, cluster analysis finds broad-ranging applications in practical realms such as traffic safety. For example, Shahin et al. [7] leveraged cluster analysis to discern clusters of akin accidents, thereby unveiling underlying patterns among diverse incidents and formulating corresponding preventative strategies. This pragmatic instantiation underscores the expansive potential of cluster analysis across varied domains.

The density-based clustering algorithm is a typical clustering algorithm that has received considerable attention from researchers owing to its complete theoretical foundation and wide application [8]. The DPC algorithm [9] is a novel density-based clustering method that constructs a visualised ρ-δ decision diagram by calculating the local density ρ and relative distance δ of each sample point in the dataset, selects the sample point with larger local density ρ and distance δ as the center of each cluster in the dataset, and then assigns the other non-center sample points to the clusters in which the center was located [10], [11].

The DPC algorithm is a novel approach that can cluster arbitrary-shaped data sets. However, it has two obvious shortcomings. Firstly, it performs poorly on datasets with uneven density distribution. If the distribution of sample points in a dataset is uneven, there may be multiple density peaks in one cluster. This can lead to misselection of cluster centroids by decision diagrams. Additionally, the non-central sample points assignment strategy assigns sample points to clusters where sample points with greater density and closest distance are located. It is a one-step assignment principle, which can result in successive assignment errors and reduced clustering accuracy. Therefore, solving the problem of poor clustering effect of DPC on variable density datasets has become the research focus of related scholars.

Fang et al. [12] used grid partitioning to divide the data space into grid cells. They then determined the clustering centers adaptively by using the density of grid cells instead of the local density of DPC. Li and Zhang [13] defined local relative densities to identify clustering centers of non-uniformly distributed datasets by considering information about the nearest neighbors of sample point truncation distance d_c. Hou et al. [14] introduced the concept of sample point affiliation to describe the relative density relationship and used the number of affiliated sample points as a criterion to determine the clustering centers. In their study on the effect of different density measures in DPC on clustering results, Hou and Zhang [15] proposed a new kernel that addresses the deficiency of DPC in effectively clustering variable density datasets through normalization and other methods. Mehmood et al. [16] proposed a nonparametric method to estimate the probability distribution of a given dataset. The method is based on the idea of thermal diffusion to optimize the truncation distance d_c and detect cluster boundaries, which improves the clustering quality of the DPC algorithm. Zhu et al. [17] used density ratios instead of kernel densities in the DPC algorithm to overcome the deficiency that global kernel densities are not fully adaptable to variable density datasets. Wu et al. [18] proposed an efficient clustering method based on density peaks with symmetric domain relations. They calculated the K-nearest neighbors and reverse K-nearest neighbors of each sample point to establish a symmetric neighborhood graph. Then, they used reverse K-nearest neighbors to calculate local densities and distinguish density peaks of sample points. Finally, they applied clustering using the symmetric neighborhood graph. Xu and Jiang [19] constructed sparse graphs based on truncated distance d_c and automatically selected clustering centers based on the connectivity of the graphs to reduce the effect of uneven distribution of the dataset on the clustering results. Wang et al. [20], [21] used data fields and fuzzy theory to optimize the calculation of local densities to solve the deficiency of difficult decision making of clustering centroids due to multiple density peaks. The scholars mentioned above proposed solutions for DPC's lack of adaptability to variable density datasets and achieved better results. However, they did not effectively address the shortcomings of DPC in the non-centroid assignment strategy.

In terms of optimizing DPC assignment strategy for non-central sample points, Liu et al. [22], [23], [24] optimized the assignment of non-central sample points by calculating the shared nearest neighbor information of sample points to avoid the possible cascading errors encountered in DPC non-central point assignment. Lei et al. [25] proposed a multi-cluster merging strategy by defining the similarity between clusters and performing multi-cluster merging according to the metric criterion of cluster similarity to avoid cascading errors when assigning non-central sample points. Zhao et al. [26] first assigned the K-nearest neighbors of the density peak to their corresponding clusters, and then based on the proximity of the sample points, the non-central sample points were assigned to the cluster where the sample points with their highest proximity and have been assigned were located, and this assignment strategy can effectively improve the correct assignment rate of non-central points. Seyedi et al. [27], [28], assign the neighboring nodes of the central sample point to the label of the central sample point to form a local backbone sample points after determining the center of clustering, and then use the dynamic label propagation process for node update to achieve the delineation of the labels of non-central sample points. Long et al. [29] used the local density of the DPC algorithm to capture the density structure graph of the dataset to construct the cluster spectrum of the dataset, designed a new similarity measure between clades, and used the normalized cut objective function to cut the connection graph between these clades to achieve the partitioning of the dataset and avoid the shortcomings of the DPC algorithm in the allocation of sample points. All of the above proposed methods for optimizing the DPC arithmetic allocation strategy have demonstrated the effectiveness of their optimization strategies on the relevant datasets and improved the clustering quality of the DPC algorithm in terms of accuracy.

The mentioned above related scholars have used different methods to optimize the shortcomings of DPC, all of which have improved its clustering effect in one aspect. In this paper, we propose a density peak clustering algorithm SM-DPC based on nearest neighbor relationship optimization and multi-cluster fusion to address the shortcomings of DPC in terms of poor adaptability to density inhomogeneous data sets and non-central sample point assignment which can easily result in consecutive assignment errors. The main innovative points are as follow:

(1) The local density calculation employs the K-nearest neighbor method to define a new measure of local density. By using K-nearest neighbor and Gaussian kernel, the local density considers the spatial distribution characteristics of sample points. This approach can identify the local structure of the data set and improve the local density values of sample points in sparse clusters. It also effectively reduces the influence of uneven density distribution among clusters on the selection of clustering centers.

(2) When assigning non-central sample points, use the two-step assignment strategy of the shared nearest neighbor and inter-sample similarity. Divide the non-central sample points into core-connected and boundary-connected sample points. Initially assign the core-connected sample points based on the cluster center. Then, gradually assign the boundary-connected sample points based on the classification of the core-connected sample points and their inter-sample similarity. This assignment strategy improves the clustering accuracy of the algorithm and prevents chain reactions caused by DPC.

(3) In the selection of centroids, it is inevitable that the distance between the sample points of the same class of clusters in a special data set is too large, that means that the Euclidean distance from the sample point i to the kth nearest neighbor is large, causing an biased selection of centroids and a decrease in clustering effectiveness. Therefore, designing a new multi-cluster fusion strategy based on a new measure criterion of similarity between clusters enables the algorithm to have an opportunity for correction, even if the centroids are selected incorrectly.

II. BASIC PRINCIPLE OF DPC ALGORITHM

The DPC algorithm is a density-based clustering algorithm. Its core idea is to construct a visual ρ-δ decision diagram to select the cluster centers. These are sample points with relatively large values of both ρ and δ. The algorithm calculates the local density ρ of the sample point and the distance δ between the point and the sample point with larger local density than it and the closest distance to it. The remaining non-central sample points are clustered in the cluster of the nearest neighbor with high-density value sample points.

DPC offers two methods for calculating the local density ρ of sample points. The cutoff kernel, shown in equation(1), is used for large data sets, while the exponential kernel, shown in equation(2), is used for small data sets.

$$\rho_i = \sum_{i \neq j} \chi(d_{ij} - d_c),\ \chi(x) = \begin{cases} 1, x<0 \\ 0, x\geq 0 \end{cases} \tag{1}$$

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

where d_{ij} represents the Euclidean distance between sample point i and sample point j, d_c is the truncated distance, which is defined as the value at the 2% position after the distance between any two sample points is arranged from smallest to largest, and needs to be set artificially. From equation(1), it can be seen that the local density of sample points is equal to the sum of the number of all sample points whose distance from the sample point is less than the truncated distance d_c. When the size of the data set is small there may be a large number of the same local density values, in order to avoid this situation, choosing the exponential kernel to calculate the local density can reduce the impact of the truncated distance d_c on the local density of the sample.

The relative distance δ of sample point i is defined in equation (3), and for the sample point with the largest local density, the relative distance δ is defined in equation (4).

$$\delta_i = \min_{j:\rho_j > \rho_i} (d_{ij}) \tag{3}$$

$$\delta_i = \max_{i \neq j}(\delta_j) \tag{4}$$

After calculating the local density ρ and relative distance δ for each sample point, a visualized ρ-δ decision diagram is constructed, and the sample points with larger local density and distance are selected as the centers. DPC also gives the automatic extraction of clustering centers using the decision value γ_i, see equation (5).

$$\gamma_i = \rho_i \times \delta_i \tag{5}$$

The points with large decision values γ_i are usually the ones with relatively high local densities ρ and relatively long distances δ, which are selected as clustering centers. Unassigned centroids will be allocated to the cluster where the sample point with greater local density and closest distance is located.

III. SM-DPC ALGORITHM

The SM-DPC algorithm aims to address the limitations of DPC in clustering datasets with uneven cluster densities and non-central sample point assignment strategies that can lead to consecutive assignment errors. The improvement method comprises three main parts:

The first improvement is to optimize the density calculation method. By applying the concept of k-nearest neighbors, the local density value of a sample point is determined solely by the relative closeness of the sample points within its set of k-nearest neighbors. This effectively characterizes the samples in low-density regions, thus reducing the impact of uneven density distribution among clusters on the selection of clustering centers. The second improvement introduces a two-step allocation strategy that combines shared nearest neighbors and sample similarity to allocate non-central sample points. This approach addresses the chaining effect produced by the original allocation method through a point-to-point approach, enhancing the accuracy of non-central sample point allocation. The third improvement employs a multi-cluster fusion strategy to further improve clustering effectiveness when the selection of clustering centers deviates. This strategy divides the dataset into multiple micro-clusters, which are then merged based on the calculation of inter-cluster affinity to refine the clustering results.

A. OPTIMIZATION OF THE CENTROIDS SELECTION

The definition of density in DPC is directly related to the selection of density peak points, and once the density peak as the center of clustering is selected incorrectly, it will lead to wrong clustering results. When there is a large difference in density between clusters in the dataset, the sample points with peak density calculated by the DPC algorithm are most likely to be in the high-density clusters, During the selection of clustering centers, they will be selected in the high-density clusters in preference to the sample points in the low-density clusters, which will lead to the wrong selection of clustering centers. Taking the Unbalance dataset shown in Figure 1 as an example, Figure 1 shows the heat map of the density values of each sample point calculated by the DPC algorithm according to the definition of density in equation (1).

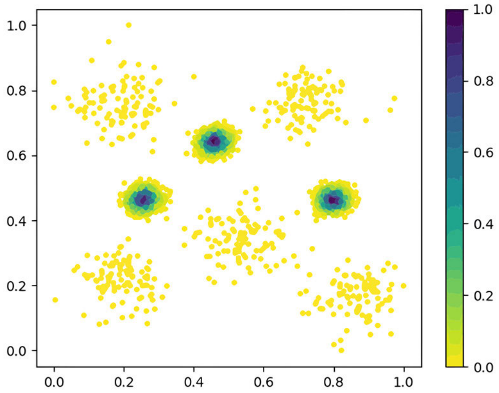


FIGURE 1. Heat map of sample point density.

It can be seen that the sample points with high density values are concentrated in the central area of the three high-density clusters, and the cluster centers are selected by choosing the sample points with greater density and relative distance, so it will result in the high-density clusters may have more than one cluster center point, while the low-density clusters have no cluster center points. Figure 2 shows the 8 clustering centers (pentagons in the figure) selected according to the decision value equation (5) of the DPC algorithm and the clustering results. It can be seen from the figure that two more clustering centers were incorrectly selected in the high-density cluster, while the sample points in the other two low-density clusters had no clustering centers, and the error in the selection of clustering centers led to the final false clustering.

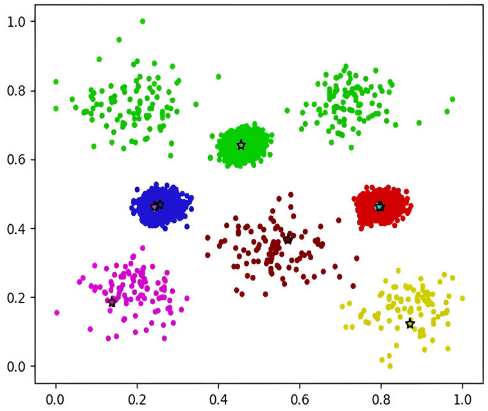


FIGURE 2. Selected clustering centroids and clustering results.

To address the problem that DPC is not applicable to datasets with large density differences among clusters, the K-nearest neighbor is introduced to redefine the local density metric. For the peak density points, their values are not the sample points with the highest density in the global range, but the sample points with higher density in the local range, so as long as they are the sample points with higher density values in the cluster, they are likely to be selected as the cluster center. The density measure of K-nearest neighbor optimization takes into account the information of the nearest neighbors of sample points, which better reflects the local structural characteristics of the dataset, amplifies the local density values of samples in low-density clusters, reduces the gap of density values among clusters, and makes it possible for sample points with higher local density values in low-density clusters to be selected as clustering centers.

SM-DPC introduces K-nearest neighbors into the density calculation and redefines the calculation of the density of sample points in combination with a Gaussian kernel.

Definition 1 (K-nearest neighbors): Given a data set X, for any sample point i, the K-nearest neighbors of sample point i are the set of the nearest K points in the distance of that sample point i to other sample points [21], which is defined in equation (6).

$$KNN\ (i) = \{j \in X | index\_dist(i, j) \leq k\} \tag{6}$$

where $index\_dist(i, j)$ is the index value of the distance of sample point i to other sample points in ascending order.

Definition 2 (Shared Nearest Neighbor): Given a dataset X, $KNN(i)$ and $KNN(j)$ are the sets of K-nearest neighbors of sample point i this and sample point j, and the sets of shared nearest neighbors of sample point i and sample point $j$is defined in equation (7).

$$SNN(i, j) = KNN(i) \cap KNN(j) \tag{7}$$

Definition 3 (Local density): For sample point i in data set X, its local density is defined in equation (8).

$$\rho_i = \sum_{q \in KNN(i)} \exp(-\frac{1}{k} \sum_{j \in KNN(q)} d_{qj}) \tag{8}$$

where $KNN(i)$ and $KNN(q)$ are the set of K-nearest neighbors of sample point i and sample point q, d_{qj} is the Euclidean distance from sample point q to sample point j in the set of its K-nearest neighbors.

For the new local density metric, it is divided into the following steps:

(1) Calculate the Euclidean distance $index\_dist(i, j)$ between sample points in the dataset;

(2) Find the set $KNN(i)$ of the k-nearest neighbors of a single sample point i and the set $KNN(q)$ of the nearest neighbors of point q in $KNN(i)$ according to equation (6);

(3) Apply equation (8) to obtain the local density of sample point i based on the relationship between $KNN(i)$ and $KNN(q)$.

It can be seen that new local density metric formula only considers the K-nearest neighbor sample points of the sample point, therefore, the local density value of a sample point is only related to its K-nearest neighbour sample points and the K-nearest neighbors of its K-nearest neighbors, while other sample points at a farther distance have no effect on the local density value.

Taking the Unbalance dataset as an example, equation (8) is used as a measure of local density, and the heat map of the local density value of each sample point is shown in Figure 3. It can be seen from Figure 3 that the density is not even among the clusters in the dataset, there are also localized sample points with high density values in the clusters with sparse distribution of sample points after using the density calculation of equation (8).

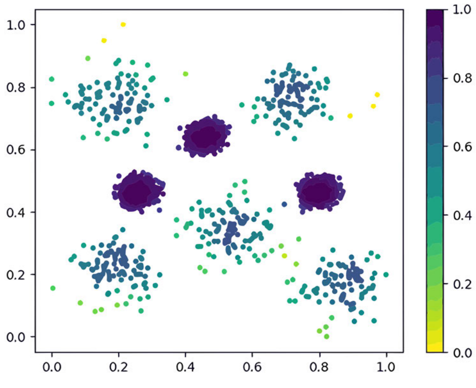


FIGURE 3. **Sample point density heat map of SM-DPC algorithm.**

Figure 4 shows the eight clustering centers (pentagons in the figure) selected according to the decision value equation (5) of DPC and the clustering results. It can be seen Figure 4 that all eight clusters have locally higher sample points, the clustering centers are correctly selected, and the final clustering results according to this density metric are also correct. This indicates that the density metric proposed by the paper takes into account the spatial distribution characteristics of sample points, and can more accurately describe the relationship among samples by using the nearest neighbor information of sample points, which can not only identify the local structural characteristics of the dataset, but also improve the local density values of sample points of sparse cluster species, which effectively reduce the influence of uneven density distribution among clusters on the selection of clustering centers, and improve the shortcomings of the density calculation method in DPC. In addition, the choice of the number of true cluster centers is not randomly entered manually, but is determined based on the structure of the dataset being clustered.

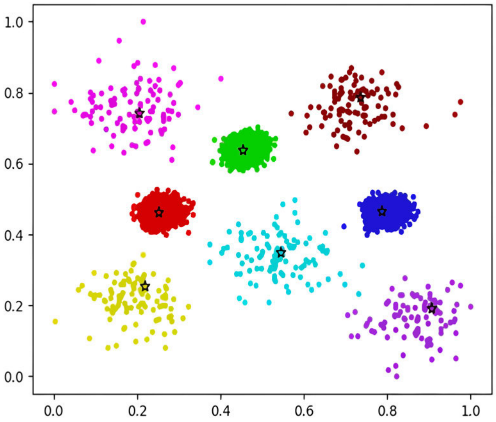


FIGURE 4. **SM-DPC algorithm selected cluster center points and cluster results.**

B. NON-CENTRAL SAMPLE POINTS ALLOCATION STRATEGY

The definition of non-central sample points encompasses all sample points except for the clustering centers. In this paper, a two-step assignment strategy for non-central sample points is proposed based on the shared neighbor count among sample points, which categorizes sample points into core-connection points and boundary-connection points. Core-connection points typically reside in the core regions of clustering clusters, representing the characteristic features of the cluster, whereas boundary-connection points lie between core regions and noise regions, with their k-nearest neighborhood points insufficient to reach the threshold but still directly connected to core points. By classifying sample points into these two categories, a better understanding of the internal structure and boundaries of clustering clusters can be achieved. Moreover, the introduction of core-connection points aids in the algorithm's quicker identification of the core parts of clusters, thereby reducing the processing of noise data points during iterations. As core-connection points concentrate in high-density areas, they can effectively detect and form clustering clusters in the initial stages, thereby enhancing the efficiency of the algorithm.

Definition 4 (Core-connection point): Suppose sample point i has been assigned, sample point j has not been assigned, and sample point p is a sample point in the shared nearest neighbor data set of sample point i and sample point j. If equation (9) is satisfied, then sample point j is a core-connection point of sample point i.

$$|\{\rho|\rho \in SNN(i,j) = KNN(i) \cap KNN(j)\}| \geq k/2 \tag{9}$$

From equation (9), it is evident that if sample point j is the core-connection point of sample point i, then at least half of their respective sets of K-nearest neighbors are shared, and sample point j is deemed to belong to the same cluster as sample point i. In other words, when the number of shared nearest neighbors between two points is sufficient, it indicates that they are densely and closely connected in space, suggesting that the two points have similar local density distributions and a strong correlation between them. Consequently, they can be considered as data points with similar density distributions and classified into the same clustering cluster.

Definition 5 (Boundary-connection point): Suppose sample point i has been assigned, sample point j has not been assigned, and sample point p is a sample point in the shared nearest neighbor data set of sample point i and sample point j. If equation (9) is satisfied, then sample point j is a boundary-connection point of sample point i.

$$0 < |\{\rho|\rho \in SNN(i,j) = KNN(i) \cap KNN(j)\}| < k/2 \tag{10}$$

Definition 6 (Similarity between sample points): The similarity between samples $Sim\left(x_i, x_j\right)$. It is defined [30] as follows:

$$\omega(i,j) = \begin{cases} e^{-d_{ij}^2}, & j \in KNN(i) \\ 0, & others \end{cases} \tag{11}$$

$$A(x_i, x_j) = \frac{\sum_{v \in |KNN(i),i|} \omega_{vj} + \sum_{v \in |KNN(j),j|} \omega_{vi}}{k} \tag{12}$$

$$Sim(x_i.x_j) = |SNN(i,j)| \cdot A(x_i, x_j) \tag{13}$$

where, $\omega(i,j)$ represents the proximity of sample x_i to sample x_j, classifying the relationship between sample x_i and other samples into k-nearest neighbor and non-nearest neighbor cases, and $\omega(i,j)$ numerically considers only the Euclidean distance between sample point x_i and its k-nearest neighbor. $A\left(x_i, x_j\right)$ represents the mutual proximity of sample x_i and sample x_j, which is the sum of ω from sample x_i and its k-nearest neighbors to x_j, and ω from sample x_j and its k-nearest neighbors to x_i, then normalized. This reflects the density of the environment in which the sample is located. $Sim\left(x_i, x_j\right)$ is the similarity between sample x_i and sample x_j, and $|SNN(i,j)|$ denotes the number of shared nearest neighbors between sample x_i and sample x_j. In the process of calculating the sample similarity, the distribution characteristics between samples are taken into account, and the more shared nearest neighbors between samples, the higher the sample similarity, so that the algorithm can correctly allocate the remaining samples.

In fact, the sample points that do not satisfy the condition of core-connection points are boundary-connection points. After dividing the non-central sample points into core-connection points and boundary-connection points, the assignment is carried out in two steps. Firstly, the core-connection sample points are assigned, starting from the cluster centroid, and using breadth-first search for its K nearest neighbor sample points, and if the number of shared nearest neighbors between them is greater than half of the value of K, the sample point is subordinated to the cluster where the current centroid is located, and it is assigned to the cluster where the current sample point is located, which is the assignment strategy 1. After assigning the core connected sample points, find the similarity of the corresponding unassigned points from the similarity matrix to form a new similarity matrix, and find the unassigned sample with the greatest similarity to the assigned sample in the new similarity matrix. Assigning the unassigned samples to the clusters where the assigned samples are located, which is the assignment strategy 2.

Step 1:Assignment strategy 1: Assign core-connection points

Input: set of clustering center, number of sample nearest neighbors K.

Output: initial assignment result C

(1)Initialize the set Q, select the clustering centroids sequentially and give the cluster label C_i, and add them to the set Q;

(2)Take the sample point p at the head of the set Q and remove it from the set Q;

(3)The unassigned sample point r in the K-nearest neighbor set *KNN(p)* of sample point p, if it satisfies $|SNN(p, r)| \geq K/2$, the sample point r will be grouped into the cluster where sample point p is located, and the sample point r will be added to the end of set Q;

(4)If the set Q is not empty, then turn (2), otherwise strategy 1 end.

Step 2: Allocation strategy 2: Assign boundary-connection points

Input: The number of sample nearest neighbors K, the initial assignment result C.

Output: The final clustering result C.

(1) Find all unassigned points and renumber them;

(2)Calculate the sample similarity according to equation(7) and equation(10)-equation(12), and construct the sample similarity matrix;

(3)Finding the data corresponding to unassigned sample points in the similarity matrix to form a new similarity matrix M;

(4)Find the maximum value in the matrix M, record the unassigned points q_i corresponding to the maximum value and the corresponding cluster C_i;

(5)Assign the sample point q_i to cluster c_i and remove the sample point q_i data from the matrix;

(6)Repeat (2) until all points are assigned; strategy 2 end.

C. MULT-CLUSTER FUSION

When selecting centroids for the data, it is possible that the distance gap between sample points of the same type of cluster is too large, resulting in a large Euclidean distance from sample point i to the k-th nearest neighbor, and a selection bias of centroids, which means that two and more centroids appear in a cluster. To optimize the clustering effect, a multi-cluster fusion strategy is proposed.

The concept of multi-cluster fusion [22] has gradually become a hot topic in cluster research. The final clustering result is expected to be achieved by merging potential clusters, which can more accurately reflect the intrinsic structure and characteristics of data. In this paper, we design a multi-cluster fusion strategy based on a new measure criterion of similarity between clusters. The similarity between clusters is evaluated based on the distribution information of data points in clusters, and which clusters should be merged to form new and larger clusters are determined.

Definition 7 (Proximity of samples to clusters): Using the inter-sample proximity obtained from equation (11), the proximity of samples to clusters is defined, as shown in equation (14).

$$p_i^m = \sum\nolimits_{j \in C_m} \frac{\omega_{ij}^4}{|C_m|} \tag{14}$$

C_m is the set of samples in cluster m, $|C_m|$ is the number of samples in cluster m, p_i^m is the proximity of sample point i to cluster m, and the larger the value of ω_{ij}, the larger the contribution of sample point j to the weight of p_i^m. the larger the value of ω_{ij}, the greater the contribution of sample point j to the weight of p_i^m. If there are more samples belonging to C_m in the K-nearest neighbors of sample point i and the closer they are to sample point i, the larger the value of ω_{ij}^4, the larger the proximity p_i^m of sample point i to C_m.

Definition 8 (Similarity among clusters): The similarity among clusters is calculated in equation (15).

$$SIM(C_m, C_n) = \sum\nolimits_{j \in C_n} P_j^m \tag{15}$$

SIM (C_m, C_n) is the similarity between two clusters, and the larger the sum of the proximity of sample $j \in C_n$ to C_m, the greater the similarity between C_m and C_n.

Multi-cluster fusion strategy: The first n samples are selected as the density peaks of the final generated clusters, and the first m ($n \leq m$) samples are selected as the potential density peaks. where n is the number of true clusters, determined by the structure of the dataset. Firstly, the similarity among clusters is calculated, and a cluster similarity matrix is built. Secondly, the two clusters with the highest similarity are merged until the number of potential clusters is equal to the number of real clusters.

Using the Jain dataset as an example, it is observed that the dataset comprises of two crescent-shaped clusters, one at the top and one at the bottom. Figure 5 illustrates the initial clustering results of the Jain dataset when the algorithm does not employ the multi-cluster fusion strategy. The clustering algorithm incorrectly assigns both clustering centers to the lower cluster (sample point A and sample point B) due to the larger distance between sample points of the upper cluster compared to that in the lower cluster, resulting in clustering errors.

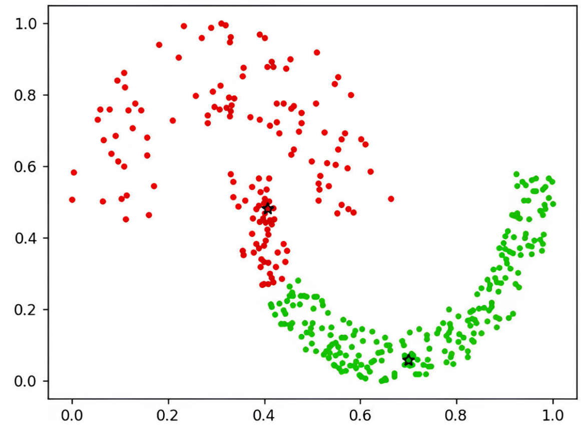


FIGURE 5. Initial clustering results for the Jain dataset.

The multi-cluster fusion strategy was used to optimize the clustering effect, as shown in Figures 6 and 7. In Figure 6, three sample points were selected as potential density peaks, and three potential clusters with sample points A, B, and C as clustering centers were obtained. Based on equations (14) and (15), the clusters with sample point A and sample point B as cluster centers are fused, resulting in the final clustering shown in Figure 7.

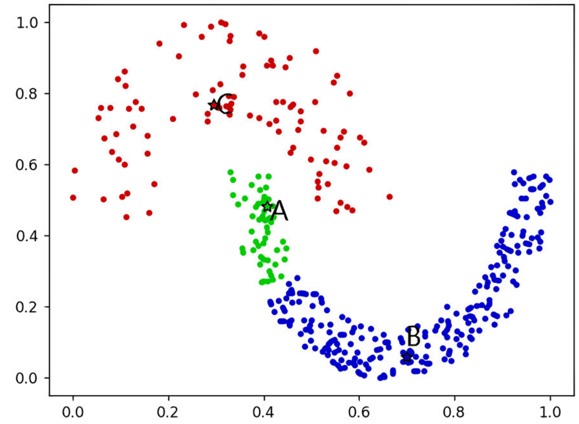


FIGURE 6. Initial clustering results for the Jain dataset(m=3).

D. SM-DPC ALGORITHM STEPS

Input: dataset X, number of nearest neighbors of sample points K, number of clusters n.

Output: clustering result C.

Setp1: normalized dataset X;

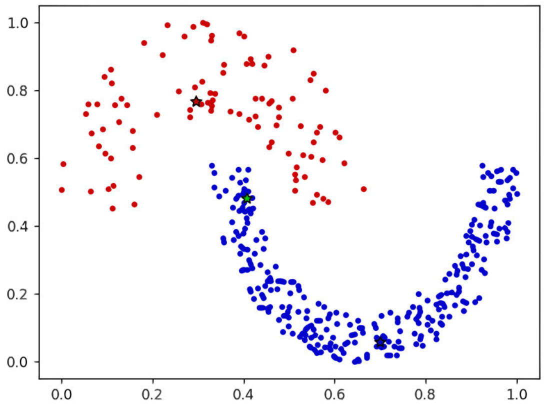


FIGURE 7. The final clustering result for the Jain dataset.

Setp2: Calculate the Euclidean distance between each sample point, and the set of K-nearest neighbors;

Setp3: Calculate the local density ρ of the sample points according to equation (8);

Setp4: Calculate the relative distance δ according to equations (3) and (4);

Setp5: Select the clustering centers according to the local density ρ and relative distance δ using equation (5);

Setp6: Assign the core connection sample points according to assignment strategy 1;

Setp7: Assign the boundary-connection sample points according to assignment strategy 2;

Setp8: Calculate the proximity p_i^m between samples and clusters according to equation (13);

Setp9: Select the number of potential clusters m,calculate the similarity $SIM\ (C_m, C_n)$ between clusters according to equation(14). Build the similarity matrix and merged potential clusters until the number of potential clusters is equal to the number of real clusters.

The flowchart of the algorithm is shown in Figure 8.

E. TIME COMPLEXITY

The time complexity of SM-DPC is determined by four main components. (1)The complexity of computing the distance among sample points is $O(n^2)$. (2)The computation of the local density of sample points is restricted to their K-nearest neighbors, and the time complexity of searching their K-nearest neighbors is $O(n)$. Therefore, the time complexity of searching the K-nearest neighbors of n sample points is $O(n^2)$. (3)For each sample point, calculate the distance δ value to the nearest sample point with a higher local density, with a time complexity of $O(n^2)$. (4)Allocation strategy 1 involves searching the k-nearest neighbors of the core-connected sample points. Since the k-nearest neighbors of the samples have already been calculated, and assuming that the size of the core connection sample points is n_1, the time complexity is $O(kn_1)$. Allocation strategy 2 should traverse all unallocated sample points, assuming their size is n_2. The time complexity of the similarity matrix for the n_1 sample points that have been allocated is $O(n_1 \times n_2)$, where n_1 and n_2 are smaller quantities for n. The total time complexity of the allocation strategy is less than $O(n^2)$. Additionally, potential clusters should be merged. To merge potential clusters, the algorithm calculates the similarity between samples, the proximity of samples to clusters, and the similarity between clusters. Each part has a time complexity of $O(n^2)$, resulting in a total time complexity of $O(n^2)$. The proposed algorithm in this paper has a time complexity of $O(n^2)$, which is similar to that of DPC.

IV. EXPERIMENTAL RESULT AND ANALYSIS

A. DATASETS AND EVALUATION INDICES

For the experiments, eight synthetic datasets and ten UCI datasets [31] were selected to test the performance of clustering algorithms. These datasets differ in size, number of features, density distribution, and categories, which verifies the adaptability and clustering effectiveness of SM-DPC on different types of datasets. Table 1 and Table 2 show the detailed properties of the synthetic and UCI datasets used for the experiments.

TABLE 1. Synthetic datasets.

Datasets	Number of points	Number of features	Number of categories
Jain	272	2	2
Aggregation	788	2	3
Spiral	312	2	3
Path-based	300	2	3
Unbalance	6300	2	6
R15	600	2	15
Longquare_1	900	2	6
D31	3100	2	31
Noise_1[32]	3673	2	7
Noise_2[32]	8000	2	8

Three external evaluation indices were used to evaluate the clustering results, namely Adjusted Mutual Information (AMI) [33], Adjusted Rand index (ARI) [33] and Fowlkes-Mallows Index, FMI) [34], all three indices take the upper value of 1, and the closer to 1 indicates the better clustering effect. “Arg-” indicates the optimal value of the parameters when the algorithm processes the data set, and “-” indicates that the algorithm does not need to be tuned. The bolded index value indicates that the result is the maximum value when processing the same dataset.

B. THE PARAMETER SELECTION

In order to objectively obtain the optimal clustering effect of each algorithm on different data sets, we make a parametric process for the algorithms that require parametric tuning. The SM-DPC algorithm requires setting the number of nearest

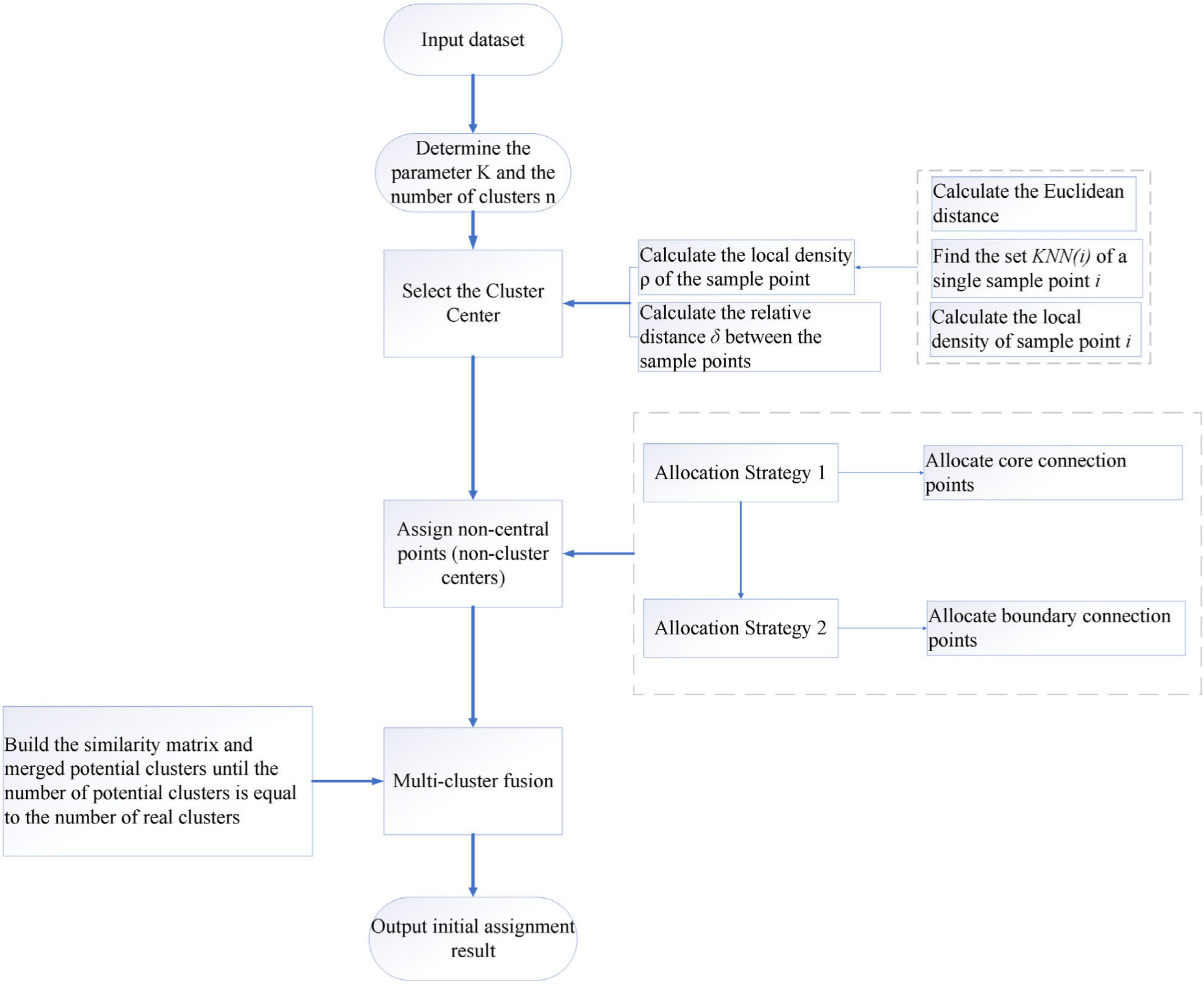

Input dataset
Determine the parameter K and the number of clusters n
Select the Cluster Center
Calculate the local density ρ of the sample point
Calculate the relative distance δ between the sample points
Calculate the Euclidean distance
Find the set KNN(i) of a single sample point i
Calculate the local density of sample point i
Assign non-central points (non-cluster centers)
Allocation Strategy 1
Allocate core connection points
Allocation Strategy 2
Allocate boundary connection points
Build the similarity matrix and merged potential clusters until the number of potential clusters is equal to the number of real clusters
Multi-cluster fusion
Output initial assignment result

FIGURE 8. Flow chart of SM-DPC algorithm.

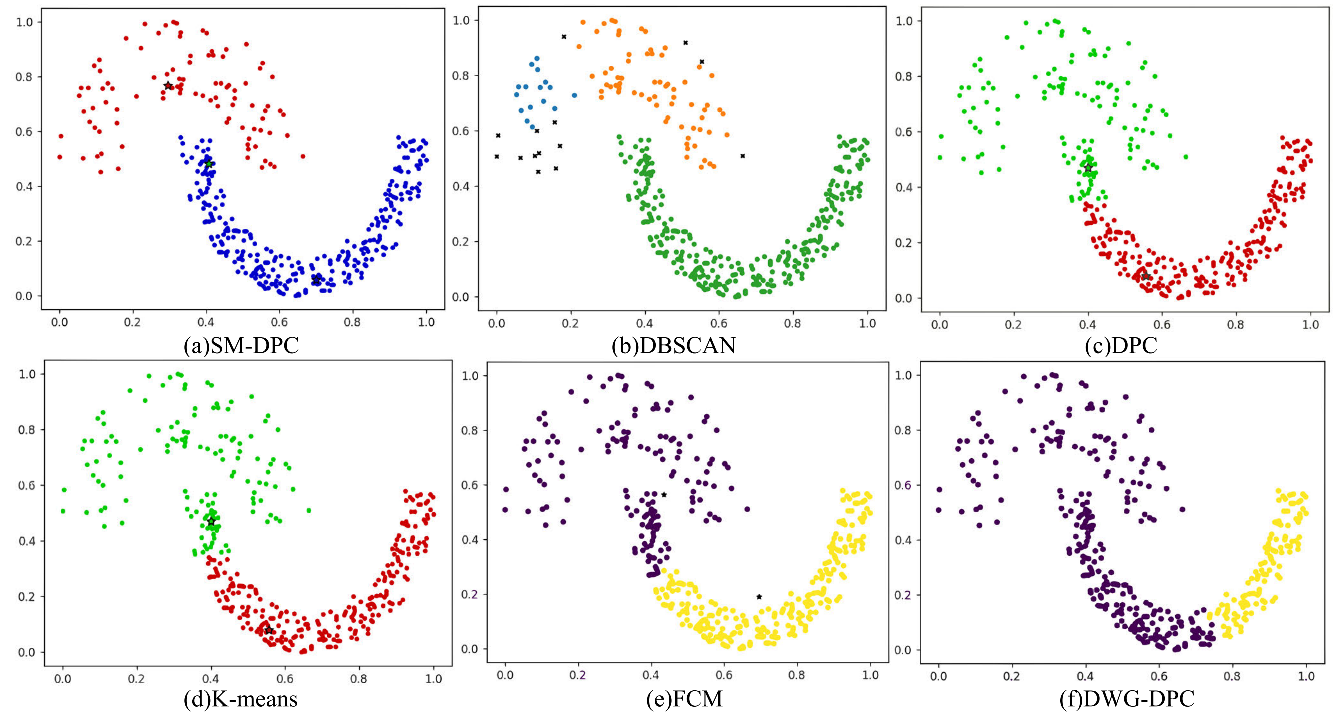
TABLE 2. UCI real-word datasets.

Datasets	Number of points	Number of features	Number of categories
Seeds	210	7	3
Iris	150	4	3
Ecoli	336	8	8
WDBC	569	30	2
Landsat	2000	36	6
Zoo	101	16	7
Dermatology	366	33	6
Wine	178	13	3
Ionosphere	351	34	2
BalanceScale	625	3	3

neighbors K of the samples, ranging from 5 to 20, and the number of potential clusters m, which takes the optimal value from 1 to 100; The DBSCAN algorithm requires tuning the domain radius ε and the minimum number of samples $minPts$. Different parameter values are selected for tuning in order to obtain the best clustering results; The DPC algorithm needs to set the truncation distance d_c, which is determined by the percentage of the distance matrix, chosen between 0.01 and 5; The FCM algorithm needs to reconcile the number of clusters c and the parameter m of the control algorithm flexibility; The DWG-DPC algorithm [34] tunes the parameter k with the optimal value between the number of parameters from 2 to the ratio of the number of sample points to the number of clusters in the dataset.

C. ANALYSIS OF EXPRIMENTAL RESULTS OF SYNTHETIC DATASETS

The paper selects ten synthetic datasets with varying sizes and cluster structures to effectively test the adaptability and clustering effects of each algorithm when faced with different types of datasets. The Jain dataset comprises of two crescent-shaped clusters that intersect, which is a common



(a)SM-DPC

(b)DBSCAN

(c)DPC

(d)K-means

(e)FCM

(f)DWG-DPC

FIGURE 9. The clustering results of each algorithm on the Jain dataset.

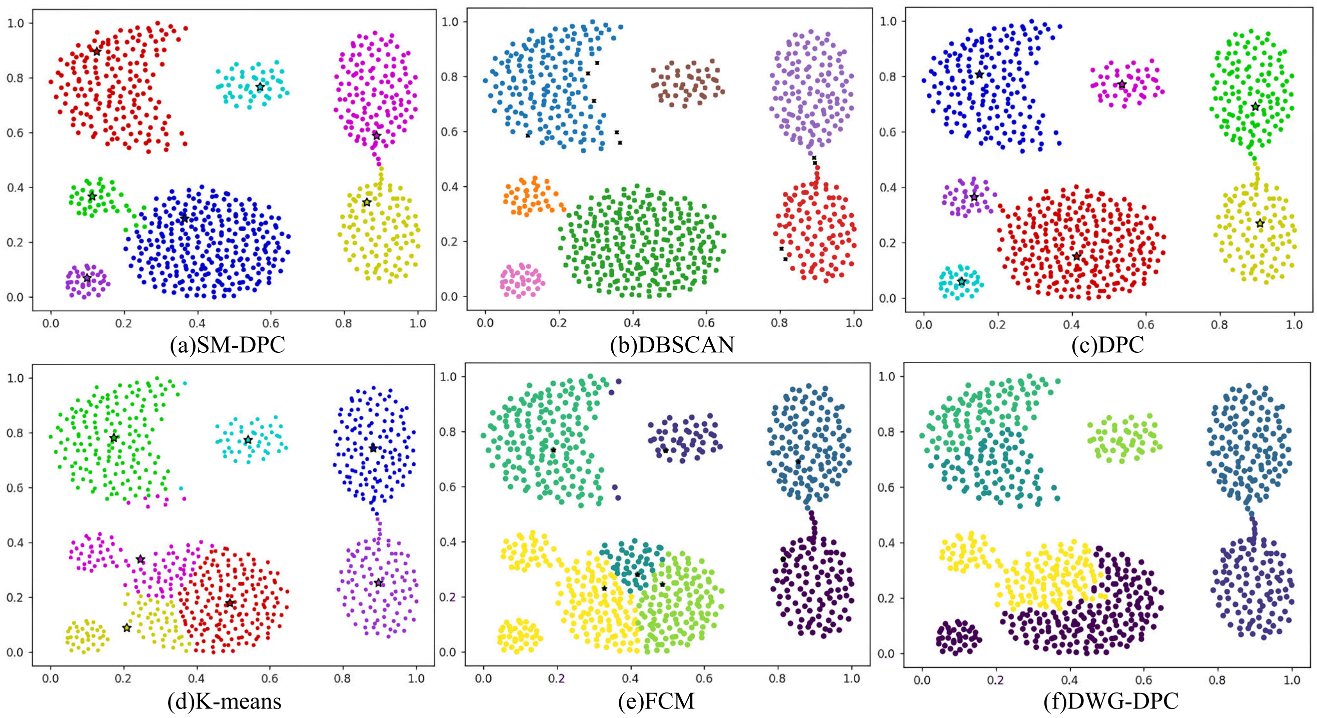
characteristic of datasets with uneven density distribution. The Aggregation dataset has a relatively uniform density distribution, and some of the clusters are connected. The Spiral dataset is typical of streamlined data, with no interference between the spirals. The Path-based dataset is a complex dataset of ring clusters surrounding block clusters. The Unbalance dataset is large and has significant differences in density between clusters. The R15 dataset has multiple categories and sticky conditions between individual clusters. The clusters in the Longquare_1 dataset have varying shapes and disturbances. The D31 dataset is a larger dataset with many categories and a disorderly distribution. The Noise_1 and Noise_2 datasets, selected by Cheng et al. [32] for the LDP-MST algorithm study, are complex datasets with notable characteristics. These datasets not only contain a large number of clusters but also exhibit extremely dense inter-cluster distributions, with connections even among noise points. This renders the clustering task more challenging and complex.

Figures 9 to 18 display the clustering outcomes of ten synthetic datasets using six algorithms. In the clustering image of SM-DPC, the pentagrams represent the centroids of clusters, not the true centroids. In K-means and DPC algorithms, the '+' symbolizes the cluster centroids. The black sample points indicate the noise points in DBSCAN.

The results of clustering each algorithm on the Jain dataset are presented in Figure 9. The Jain dataset exhibits uneven density distribution and a large distance between sample points within the same cluster due to its stream-like crescent-like cluster structure. The clustering results of DPC, K-means, FCM, and DWG-DPC are suboptimal. DBSCAN can cluster correctly for the most part, but there are still some individual points that cannot be correctly classified and are considered as noisy points. The Jain dataset is correctly classified and achieves optimal clustering through SM-DPC.

Figure 10 displays the clustering results of six algorithms on the Aggregation dataset, which comprises of seven clusters with distinct features, but there are cross-tangles between them. SM-DPC, DBSCAN, DWG-DPC, and DPC can accurately classify the clusters on the Aggregation dataset, with only a few individual sample points not being correctly clustered. Both K-means and FCM algorithms successfully identify the correct number of clusters, but the overall clustering results are unsatisfactory.

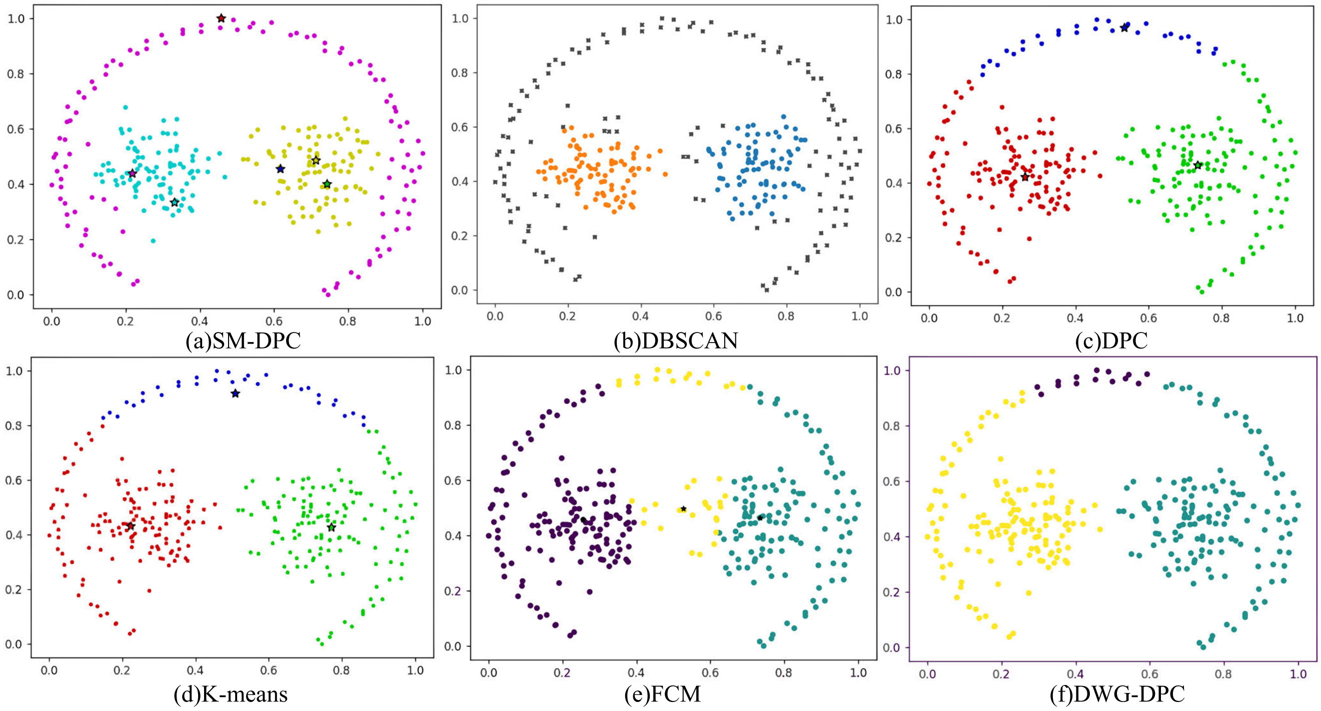
Figure 11 shows the Path-based dataset, which has a complex flow shape. It comprises three clusters: two spherical clusters surrounded by a ring-shaped cluster. The clusters are interconnected in a complex manner. The ring-shaped clusters are closely connected, which can lead to errors when assigning sample points. The figure shows that only SM-DPC can achieve correct clustering. DBSCAN can divide the clusters correctly, but it identifies the entire toroidal cluster as a noisy point. DPC can select the correct clustering center, but its one-step assignment rule based on distance and density results in the lower end of the toroidal cluster being incorrectly assigned to the two spherical clusters. The other algorithms are ineffective.



(a)SM-DPC (b)DBSCAN (c)DPC

(d)K-means (e)FCM (f)DWG-DPC

FIGURE 10. Clustering results of each algorithm on aggregation dataset.

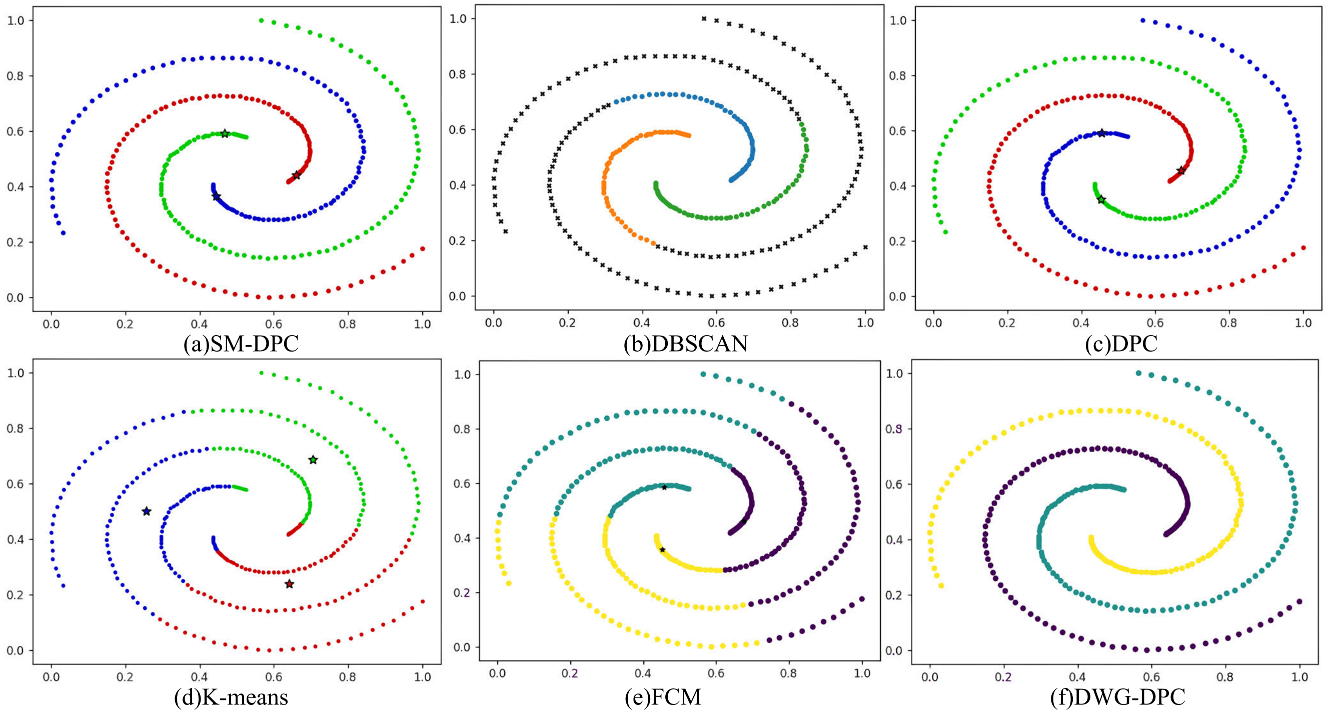


(a)SM-DPC (b)DBSCAN (c)DPC

(d)K-means (e)FCM (f)DWG-DPC

FIGURE 11. Clustering results of each algorithm on path-based dataset.

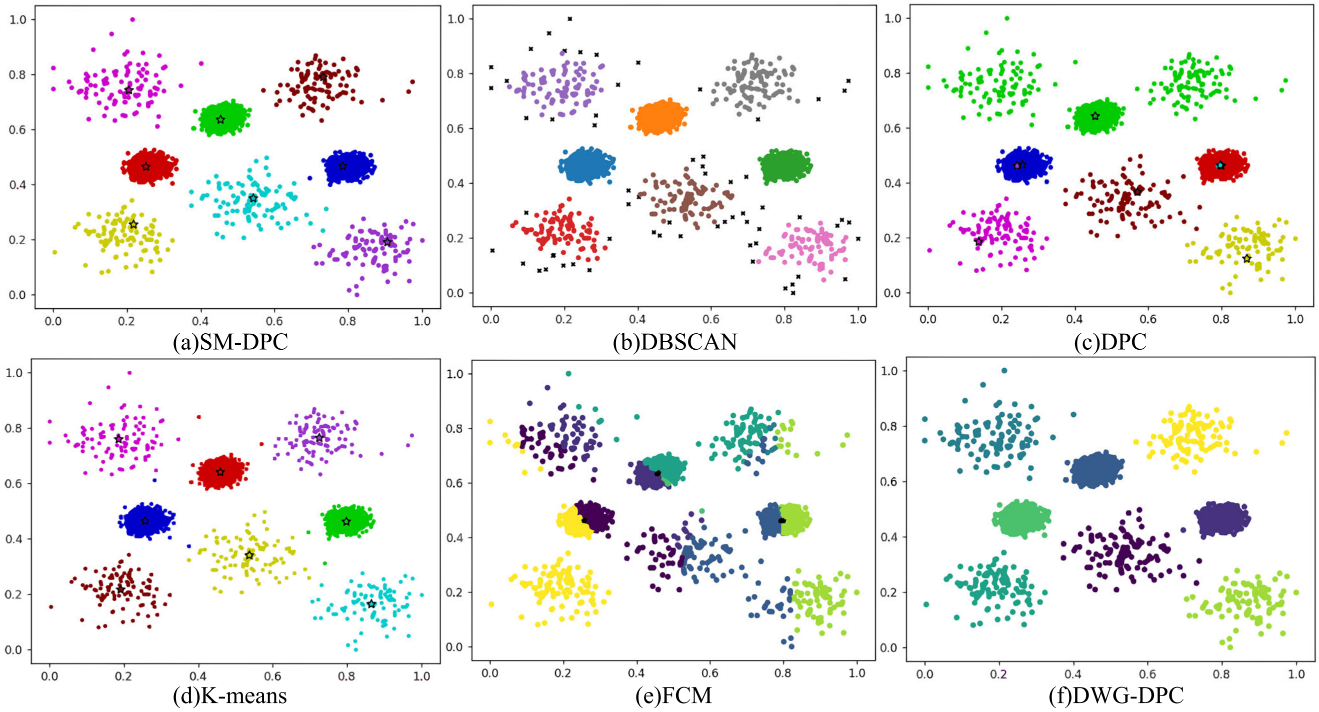
Figure 12 shows the clustering results of the six algorithms for the Spiral dataset, which is a streaming dataset consisting of three helix-like clusters. It can be seen that SM-DPC, DPC and DWG-DPC can correctly assign the sample points, DBSCAN is less effective in clustering the sample points for the Spiral dataset, while K-means and FCM do not have satisfactory clustering results.



(a)SM-DPC (b)DBSCAN (c)DPC

(d)K-means (e)FCM (f)DWG-DPC

FIGURE 12. Clustering results of each algorithm on the spiral dataset.

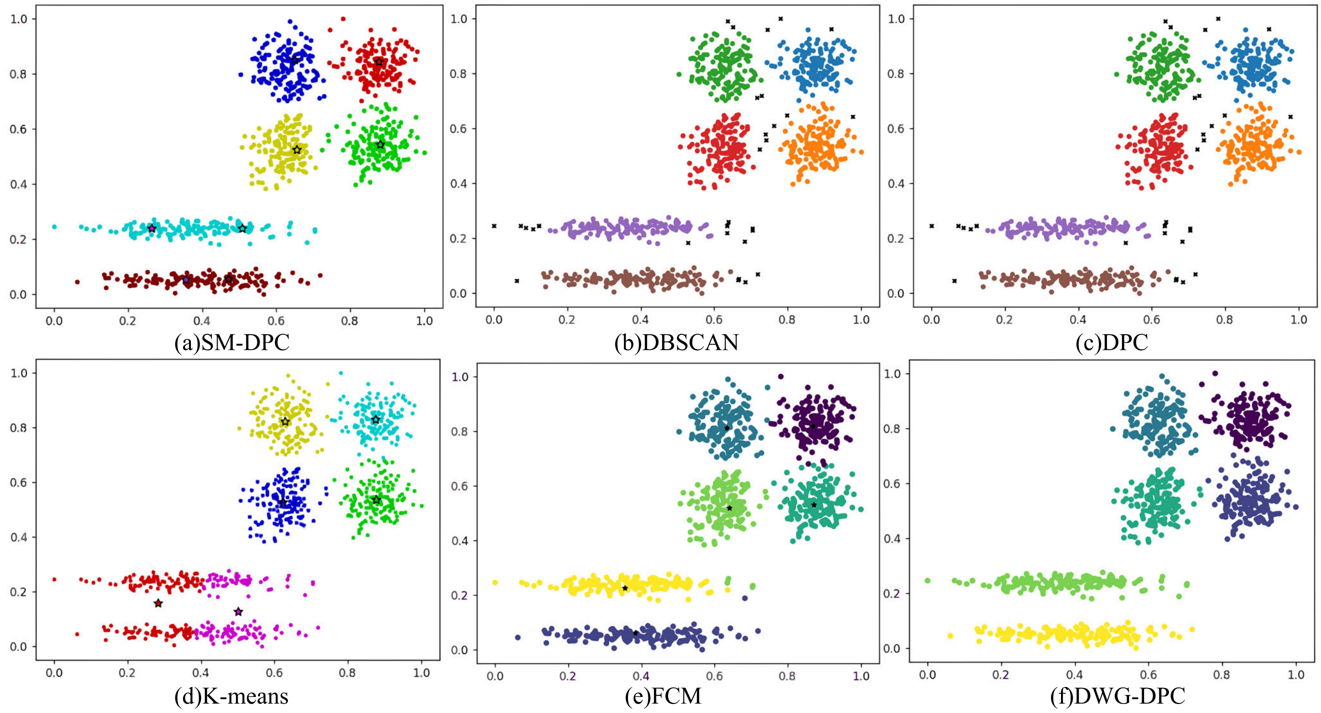


(a)SM-DPC (b)DBSCAN (c)DPC

(d)K-means (e)FCM (f)DWG-DPC

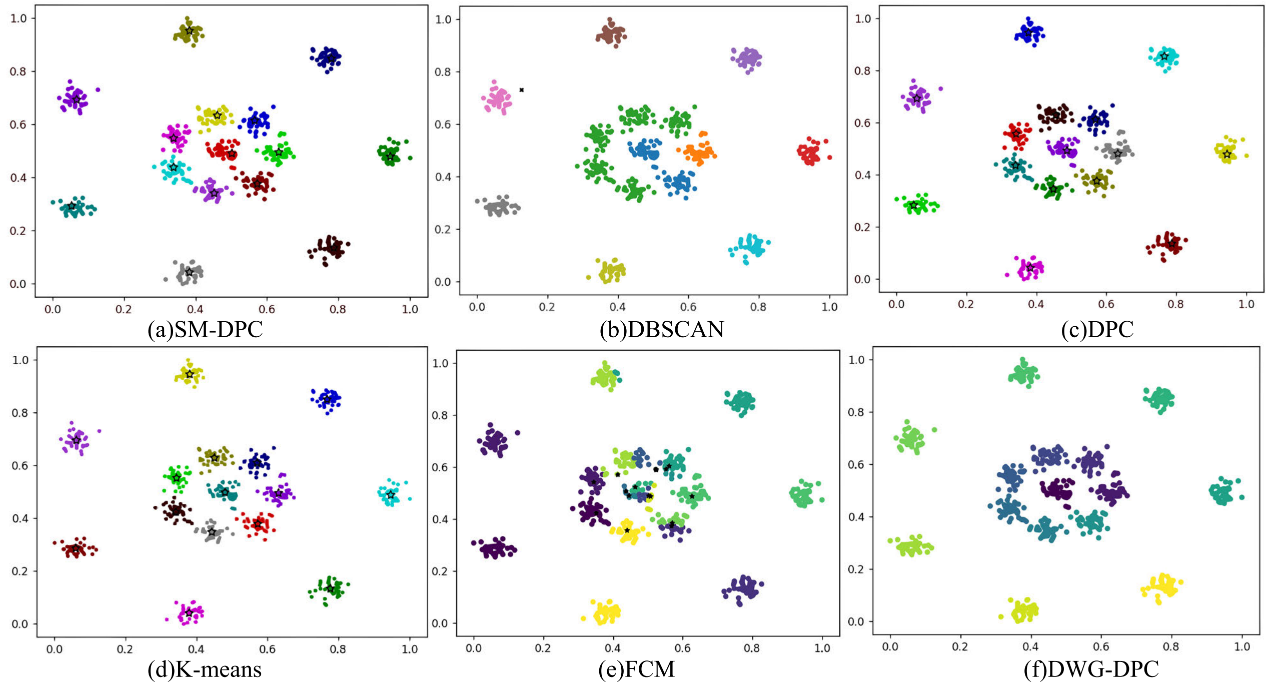
FIGURE 13. Clustering results of each algorithm on unbalance dataset.

The Unbalance dataset shown in Figure 13 is a dataset with uneven distribution of density among clusters. It can be seen from the figure that SM-DPC, K-means and DWG-DPC can correctly discover the cluster centers and correctly assign the non-central sample points; DBSCAN also can obtain good results; DPC and FCM appear to classify low-density clusters into high-density clusters, resulting in an incorrect clustering result.



(a)SM-DPC (b)DBSCAN (c)DPC (d)K-means (e)FCM (f)DWG-DPC

FIGURE 14. Clustering results of each algorithm on longquare_1 dataset.



(a)SM-DPC (b)DBSCAN (c)DPC (d)K-means (e)FCM (f)DWG-DPC

FIGURE 15. The clustering results of each algorithm on the r15 dataset.

Figure 14 shows the clustering results of six algorithms for the Longquare_1 dataset, which comprises two stream-like clusters and four clump-like clusters. The clump-like clusters are adjacent to each other, making them prone to assignment errors. The figure indicates that all algorithms perform well, with SM-DPC, DWG-DPC, and FCM having the best clustering effect.

The results of six algorithms for R15, which consists of 15 clusters, are presented in Figure 15. The outer seven clusters are well separated, while the inner eight clusters are

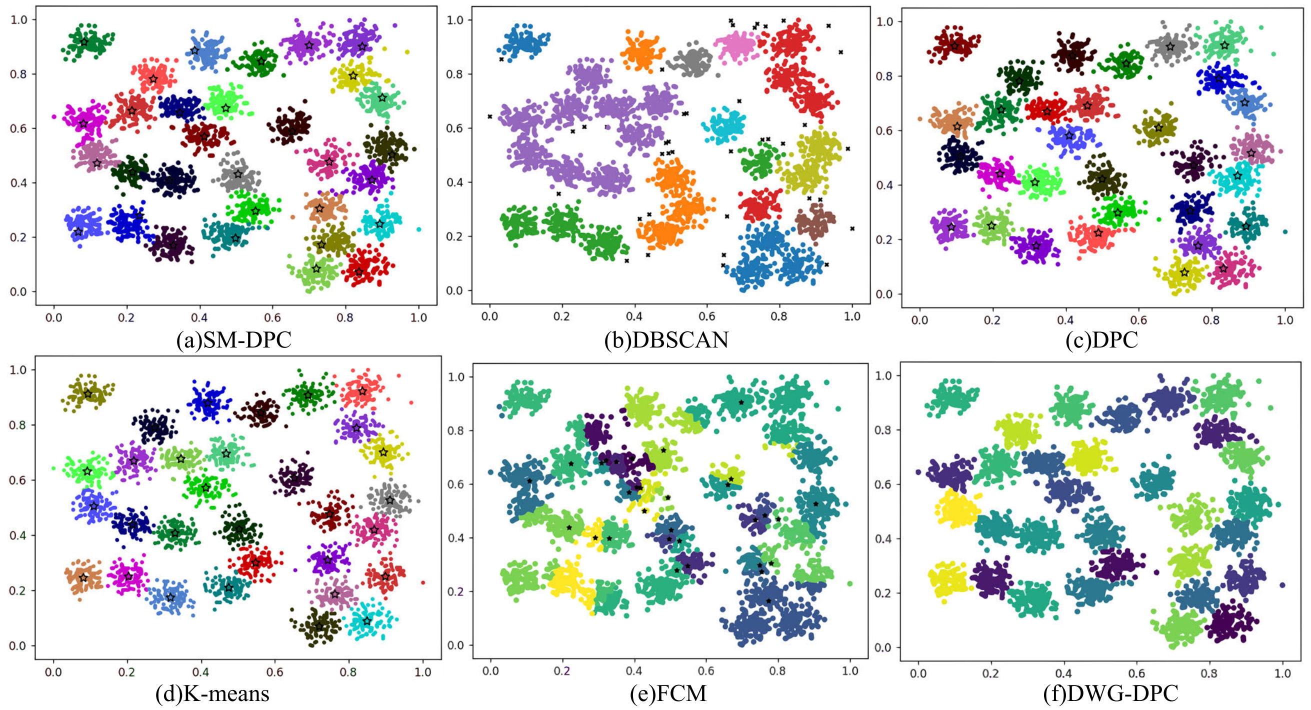


FIGURE 16. The clustering results of each algorithm on the D31 dataset.

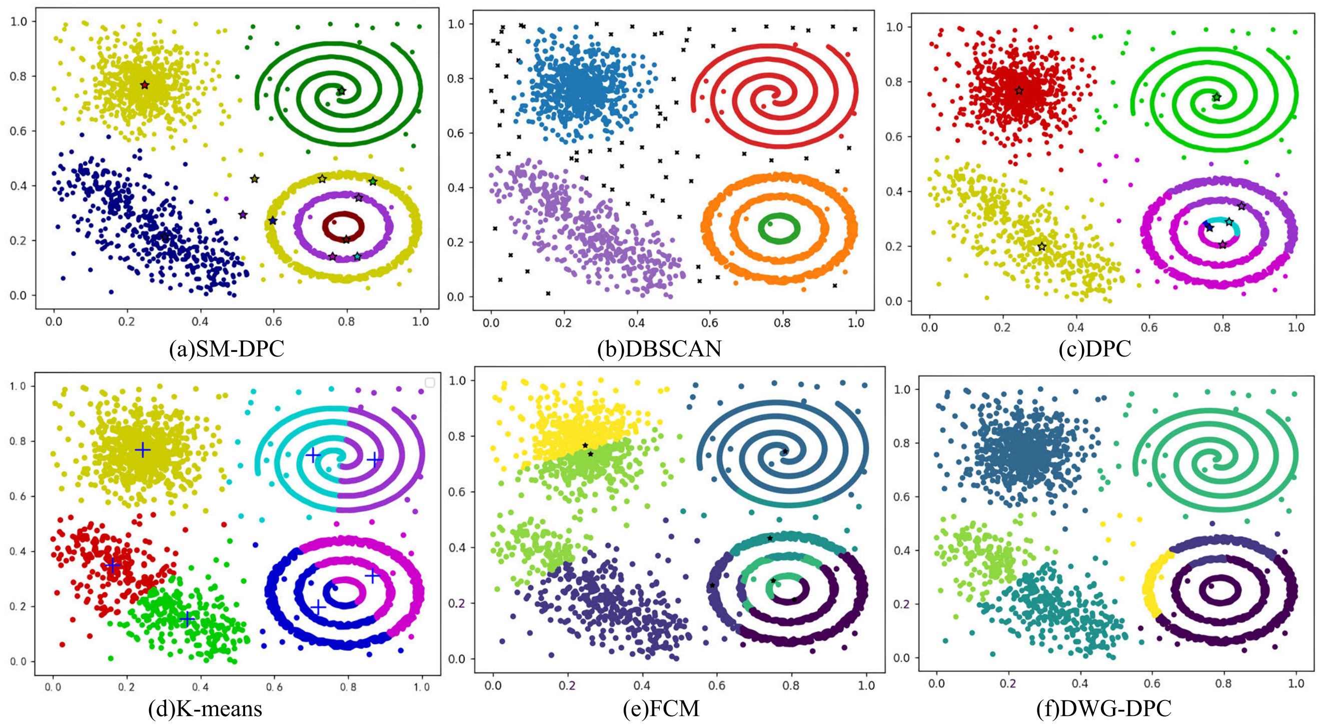


FIGURE 17. The clustering results of each algorithm on the Noise_1 dataset.

adjacent and intertwined, leading to potential misassignment of sample points. SM-DPC, DPC, and DWG-DPC produce the most accurate clustering results, correctly assigning most sample points with only a few exceptions. However, DBSCAN incorrectly groups all the inner clusters into one cluster, while K-means and FCM also miscluster the inner clusters.

The results of the six algorithms on the D31 dataset, a disordered distribution of clustered data, are presented in Figure 16. It is evident that, with the exception of the FCM

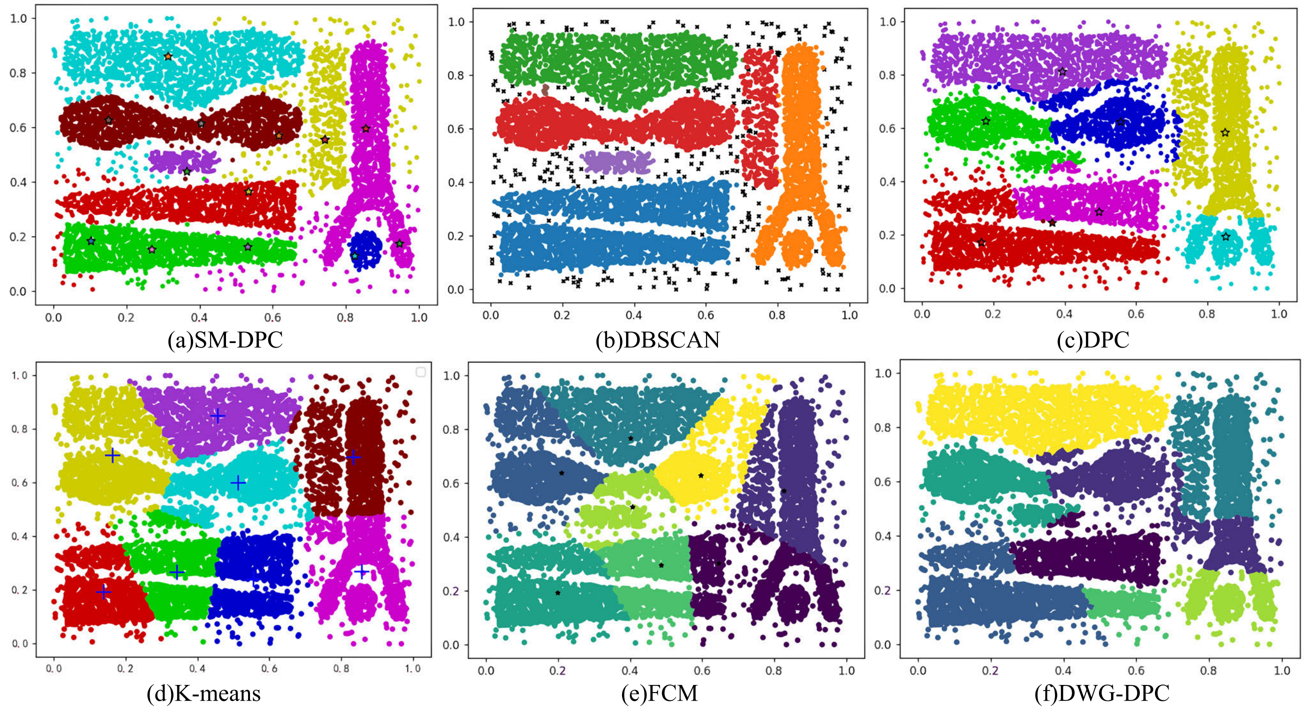


FIGURE 18. The clustering results of each algorithm on the Noise_2 dataset.

algorithm, which makes errors in clustering when individual clusters are adhered to each other, all other algorithms cluster the D31 dataset correctly. SM-DPC and the K-means have the most outstanding effect.

Figure 17 illustrates the performance of six algorithms on the Noise_1 dataset. Due to the complex internal distribution and interference from noise points, none of the algorithms achieve optimal clustering results. To varying degrees, they all exhibit errors in cluster allocation. Among these algorithms, DBSCAN performs the best, followed by the clustering effectiveness of the SM-DPC algorithm, which can essentially identify each cluster.

Figure 18 demonstrates the clustering performance of six algorithms on Noise_2 dataset. This dataset comprises clusters of various shapes and is significantly disrupted by a substantial amount of noisy points, posing significant challenges for clustering tasks. Among the six algorithms tested, all except for SM-DPC and DBSCAN failed to accurately identify the boundaries of specific clusters, leading to clustering errors. While the DBSCAN algorithm was able to recognize cluster boundaries, it incorrectly merged clusters together. In contrast, the SM-DPC algorithm performed significantly better, successfully partitioning the Noise_2 dataset into distinct clusters.

The AMI, ARI and FMI are the common straight metrics used to evaluate the performance of clustering algorithms. Table 3 shows the clustering results of SM-DPC, DBSCAN, DPC, K-means, FCM, and DWG-DPC algorithms on eight synthetic datasets, with bolded and weighted values indicating optimal experimental results. It can be seen that the SM-DPC algorithm performs better on all test data, especially for the dataset with uneven density distribution and the dataset with more complex cluster structure is significantly better than other algorithms.

D. ANALYSIS OF EXPERIMENTAL UCI DATA SETS

UCI datasets are commonly used real-world datasets specifically for testing the performance of machine learning and data mining algorithms. These datasets have exact classifications, but there are large differences in the number of attributes and sample size, etc.

The evaluation metric values of the clustering results for each algorithm on the 10 UCI datasets are shown in Table 4. It can be observed that SM-DPC significantly outperforms the other algorithms in most of thet datasets, excluding Ionosphere and Lansat. In the clustering results for the landsat dataset, SM-DPC did not achieve the best clustering results, but it is still the best algorithm besides the DWG-DPC algorithm. When dealing with the Ionosphere dataset, SM-DPC is slightly inferior to DBSCAN but performs better than the remaining algorithms. By comparing the clustering results of each algorithm on the UCI datasets, we can find that SM-DPC has good clustering performance, and the clustering performance is generally better than the other compared algorithms.

TABLE 3. Performance of 6 algorithms on synthetic datasets.

Algorithms	AMI	ARI	FMI	Arg-
Jain				
SM-DPC	1.0	1.0	1.0	10/3
DBSCAN	0.84029	0.93310	0.97343	0.08/7
DPC	0.57610	0.61832	0.83864	2
K-means	0.52644	0.57667	0.81997	-
FCM	0.50917	0.52222	0.79402	10
DWG-DPC	0.18282	-0.02958	0.58646	180
Aggregation				
SM-DPC	0.97628	0.97765	0.98250	10/7
DBSCAN	0.96752	0.97786	0.98269	0.04/10
DPC	0.99232	0.99563	0.99657	4
K-means	0.83612	0.72964	0.78812	-
FCM	0.81696	0.65676	0.72752	10
DWG-DPC	0.82378	0.64331	0.71658	250
Path-based				
SM-DPC	0.90070	0.92937	0.95292	9/6
DBSCAN	0.76520	0.76338	0.84446	0.04/10
DPC	0.53597	0.45300	0.65853	2
K-means	0.54284	0.46133	0.66168	-
FCM	0.38183	0.33824	0.57984	20
DWG-DPC	0.90070	0.92937	0.95292	9/6
Spiral				
SM-DPC	1.0	1.0	1.0	10/3
DBSCAN	0.68689	0.64371	0.75262	0.06/10
DPC	1.0	1.0	1.0	2
K-means	-0.00554	-0.00603	0.32745	-
FCM	-0.00455	-0.00492	0.32800	10
DWG-DPC	1.0	1.0	1.0	200
Unbalance				
SM-DPC	0.99635	0.99942	0.99959	10/8
DBSCAN	0.98366	0.99893	0.99924	0.04/6
DPC	0.94009	0.94495	0.96144	2
K-means	0.99348	0.99825	0.99875	-
FCM	0.67662	0.48120	0.61335	23
DWG-DPC	0.99635	0.99942	0.99959	500
Longquare_1				
SM-DPC	0.98702	0.98931	0.99109	11/8
DBSCAN	0.94494	0.95498	0.96269	0.04/6
DPC	0.99249	0.99467	0.99555	2
K-means	0.86546	0.79938	0.83268	-
FCM	0.96326	0.96350	0.96955	4
DWG-DPC	0.99249	0.99467	0.99555	400
R15				
SM-DPC	0.99381	0.99277	0.99324	15/15
DBSCAN	0.86018	0.52520	0.63023	0.04/6
DPC	0.99381	0.99277	0.99324	2
K-means	0.74417	0.40469	0.54383	7
FCM	0.80630	0.59294	0.64634	7
DWG-DPC	0.99381	0.99277	0.99324	200
D31				
SM-DPC	0.96021	0.94386	0.94566	15/31
DBSCAN	0.77863	0.34295	0.47499	0.04/25
DPC	0.95398	0.93323	0.93536	2
K-means	0.96565	0.95291	0.95441	-
FCM	0.78024	0.47498	0.52091	4
DWG-DPC	0.95566	0.93639	0.93842	100
Noise_1				
SM-DPC	0.86465	0.72807	0.80736	12/80
DBSCAN	0.87199	0.75851	0.81766	0.04/10
DPC	0.75920	0.66281	0.72899	4
K-means	0.68054	0.58795	0.65583	-
FCM	0.62099	0.46348	0.54958	5
DWG-DPC	0.73748	0.60860	0.68648	500
Noise_2				
SM-DPC	0.87813	0.89679	0.90417	14/50
DBSCAN	0.85545	0.75036	0.81365	0.02/7
DPC	0.74105	0.66083	0.71467	4
K-means	0.60056	0.42797	0.51153	-
FCM	0.58000	0.43360	0.51741	5
DWG-DPC	0.68970	0.58293	0.64637	700

E. ANALYSIS OF EXPERIMENTAL RESULTS WITH OTHER IMPROVED DPC ALGORITHMS

In order to further validate the performance of SM-DPC, SM-DPC is compared with other improved DPC algorithms, and due to the different evaluation metrics chosen in various literatures, this paper chooses ARI, which is the most frequently used among the evaluation metrics, as the metrics for evaluation. The experimental dataset in Table 5 and Table 6 was obtained from the corresponding literature. The dash indicates that the dataset was not provided in the original literature. It can be seen from Tables 5 to 6 that the improved algorithms selected are relatively excellent, with SM-DPC standing out particularly across multiple datasets.

TABLE 4. Performance of 6 algorithms on UCI datasets.

TABLE 4. *(Continued.)* Performance of 6 algorithms on UCI datasets.

Algorithms	AMI	ARI	FMI	Arg-
Seeds				
SM-DPC	0.77416	0.81376	0.87533	20/3
DBSCAN	0.5031	0.4408	0.6679	0.23/6
DPC	0.7213	0.7341	0.8231	2
K-means	0.6714	0.7049	0.8026	-
FCM	0.68787	0.72910	0.81878	10
DWG-DPC	0.65331	0.67831	0.78497	300
Iris				
SM-DPC	0.89988	0.92221	0.94778	10/3
DBSCAN	0.6396	0.5344	0.7333	0.2/10
DPC	0.6483	0.4531	0.6856	2
K-means	0.7387	0.7163	0.8112	-
FCM	0.67783	0.65252	0.76704	10
DWG-DPC	0.80322	0.75919	0.84072	200
Ecoli				
SM-DPC	0.66453	0.69659	0.79759	10/3
DBSCAN	0.4546	0.4751	0.6319	0.2/22
DPC	0.4570	0.3086	0.4747	2
K-means	0.5825	0.4458	0.5765	-
FCM	0.50200	0.58427	0.7241	10
DWG-DPC	0.57603	0.42779	0.56147	100
WDBC				
SM-DPC	0.69026	0.79911	0.9057	10/2
DBSCAN	0.3367	0.4540	0.7469	0.46/38
DPC	0.0063	-0.0056	0.7222	2
K-means	0.6226	0.7302	0.8770	-
FCM	0.56937	0.68951	0.85353	15
DWG-DPC	0.49306	0.52839	0.80372	200
Landsat				
SM-DPC	0.67453	0.63607	0.70468	30/6
DBSCAN	0.60892	0.44881	0.62437	0.45/10
DPC	0.61561	0.46809	0.56829	2
K-means	0.56856	0.46520	0.55997	-
FCM	0.32372	0.28003	0.47153	15
DWG-DPC	0.67096	0.66691	0.73199	100
Zoo				
SM-DPC	0.70658	0.56341	0.65742	8/7
DBSCAN	0.69504	0.50413	0.62909	0.2/18
DPC	0.678480	0.52952	0.62974	1
K-means	0.63459	0. 45244	0.56761	-
FCM	0.5669	0.44798	0.67412	10
DWG-DPC	0.59653	0.41508	0.58250	180
Dermatology				
SM-DPC	0.7934	0.7762	0.8330	8/6
DBSCAN	0.5721	0.4165	0.5395	0.99/3
DPC	0.784	0.776	0.8221	1
K-means	0.8748	0.7426	0.7947	-
FCM	0.4390	0.3255	0.5801	10
DWG-DPC	0.85408	0.7632	0.8090	250
Wine				
SM-DPC	0.8912	0.9024	0.9338	8/6
DBSCAN	0.5484	0.5292	0.7121	0.99/3
DPC	0.7065	0.6724	0.7835	2
K-means	0.8473	0.8685	0.9126	-
FCM	0.4669	0.3631	0..6310	10
DWG-DPC	0.8594	0.8707	0.9140	250
Ionosphere				
SM-DPC	0.4776	0.5703	0.8169	10/2
DBSCAN	0.5947	0.7226	0.874	0.78/9
DPC	0.1504	0.2357	0.6491	0.5
K-means	0.1294	0.1776	0.6053	-
FCM	0.1473	0.1455	0.5128	10
DWG-DPC	0.1441	0.1923	0.6125	300
Balance Scale				
SM-DPC	0.1772	0.2186	0.5166	22/3
DBSCAN	0.0902	0.1394	0.151	0.03/1
DPC	0.1154	0.1394	0.5024	1.1
K-means	0.0132	0.0015	0.044	3
FCM	0.1294	0.1552	0.4776	15
DWG-DPC	0.1471	0.1939	0.5210	200

As is shown in Tables 5, compared to the other six improved DPC algorithms, SM-DPC achieved the best clustering performance on the six artificial datasets (Jain, R15, Spiral, Path-based, Noise_1 and Noise_2). Although it did not achieve the optimal performance on another four artificial datasets (Aggregation, Unbalance, Longquare_1 and D31), the ARI was close to the best indicators and demonstrated stability.

It can be seen from Tables 6 that SM-DPC exhibited the best clustering results on five real-world datasets(Iris, Seeds, Zoo, Ionosphere, and Balance Scale), when compared to the other six improved DPC algorithms. While it did not achieve the optimal performance on the other five datasets (Dermatology, WDBC, Landsat, Ecoli, and Wine), its ARI

TABLE 5. The ARI of 7 improved algorithms on 10 synthetic datasets.

Algorithms	Jain	Aggregation	Path-based	Spiral	Unbalance	Longquare_1	R15	D31	Noise_1	Noise_2
SM-DPC	1.0	0.9776	0.9776	1.0	0.9994	0.9893	0.9928	0.9438	0.7280	0. 8967
FKNN-DPC[36]	1.0	0.9855	0.960	1.0	0.9892	0.9275	0.9915	0.935	0.6532	0.6954
SNN-DPC[22]	1.0	0.9594	0.9294	1.0	0.9994	0.9919	0.9928	0.9165	0.6572	0.5794
DPC-MND[26]	1.0	0.9878	0.9296	1.0	0.9928	0.9275	0.9816	0.9503	0.7045	0.7204
FastDEC[38]	0.7145	0.7622	0.3674	1.0	0.8530	0.9946	0.9819	0.9623	0.6070	0.6175
3W-PEDP[37]	0.0441	0.9742	0.9595	1.0	1.0	0.9973	0.9928	0.9484	0.7039	0.5793
DWG-DPC[35]	0.1928	0.6576	0.4150	1.0	0.9994	0.9946	0.9927	0.9363	0.6086	0.5829

TABLE 6. The ari of 7 improved algorithms on 10 real-world datasets.

Algorithms	Iris	Seeds	Ecoli	WDBC	Landsat	Zoo	Dermatology	Wine	Ionosphere	Balance Scale
SM-DPC	0.9222	0.8137	0.6966	0.7991	0.6363	0.5634	0.7762	0.9024	0.5703	0.2186
FKNN-DPC[36]	0.9038	0.7422	0.5323	0.4009	0.5223	0.4876	0.812 7	0.852	0.1321	0.0236
SNN-DPC[22]	0.9222	0.7890	0.3476	0.8503	0.5888	0.4545	0.2528	0.8992	0.3111	0.1742
DPC-MND[26]	0.9037	0.8011	0.728 3	0.7857	0.5516	0.5435	0.7869	0.914 9	0.5272	0.1985
FastDEC[38]	0.8509	0.7764	0.3472	0.7794	0.5215	0.5633	0.9036	0.8536	0.1665	0.1662
3W-PEDP[37]	0.9222	0.8125	0.4339	0.7731	0.6514	0.4791	0.8106	0.9149	0.5524	0.1173
DWG-DPC[35]	0.6181	0.6783	0.4278	0.4914	0.6669	0.4151	0.7632	0.8707	0.1919	0.1938

scores were very close to the optimal, attesting to the stability and efficiency of SM-DPC. In summary, the SM-DPC algorithm can obtain good clustering results on different types of datasets, showing its excellent clustering performance and adaptability, and its clustering performance is slightly better compared to the other six algorithms.

V. CONCLUSION

To address the limitations of DPC in clustering datasets with uneven density assignment due to the bias in centroid selection and its single non-central sample point assignment strategy, which is prone to successive assignment errors, we propose SM-DPC algorithm. The local density of sample points is measured using the K-nearest neighbor information to effectively reduce the influence of uneven density distribution among clusters on the selection of cluster centroids. The assignment strategy for non-central sample points is improved based on shared nearest neighbors and the similarity between sample points to enhance its accuracy. The multi-cluster fusion strategy is employed to correct for the bias in centroid selection for data with uneven distribution of sample points. The experimental results demonstrate that SM-DPC outperforms classical clustering algorithms, including DPC, DBSCAN, and K-means, as well as improved DPC algorithms such as FKNN-DPC and SNN-DPC, on synthetic datasets and UCI datasets. Additionally, SM-DPC is more adaptable to datasets of various forms and distributions. However, there are still some shortcomings in SM-DPC. Firstly, the algorithm requires the number of nearest neighbors (k) to be set artificially, and the size of k directly affects the clustering effect. Future work will focus on developing an adaptive method for selecting k based on different datasets. Secondly, although the algorithm produces good clustering results for large-scale data, it is time-consuming. Therefore, future research will aim to improve the algorithm's operational efficiency.

REFERENCES

[1] M. Kaushik, R. Sharma, I. Fister Jr., and D. Draheim, "Numerical association rule mining: A systematic literature review," 2023, *arXiv:2307.00662*.

[2] M. Kaushik, R. Sharma, A. Vidyarthi, and D. Draheim, "Discretizing numerical attributes: An analysis of human perceptions," in *New Trends in Database and Information Systems*, vol. 1652, S. Chiusano, T. Cerquitelli, R. Wrembel, K. Norvag, B. Catania, G. Vargas-Solar, E. Zumpano, Eds. Cham, Switzerland: Springer, 2022, pp. 188–197.

[3] B. I. Priyadarshini and D. K. Reddy, "Modified remora optimization based matching pursuit with density peak clustering for localization of epileptic seizure onset zones," *Evolving Syst.*, vol. 15, no. 2, pp. 249–265, Feb. 2023.

[4] S. S. Kumar, S. T. Ahmed, Q. Xin, S. Sandeep, M. Madheswaran, and S. M. Basha, "Unstructured oncological image cluster identification using improved unsupervised clustering techniques," *Comput., Mater. Continua*, vol. 72, no. 1, pp. 281–299, 2022.

[5] Z. Ma, Y. Cao, L. Song, F. Hao, and J. Zhao, "A new smoke segmentation method based on improved adaptive density peak clustering," *Appl. Sci.*, vol. 13, no. 3, p. 1281, Jan. 2023.

[6] Z. Zhang, S. Li, W. Liu, Y. Wang, and D. X. Li, "A new outlier detection algorithm based on fast density peak clustering outlier factor," *Int. J. Data Warehousing Mining*, vol. 19, no. 2, pp. 1–19, Jan. 2023.

[7] M. Shahin, S. Saeidi, S. A. Shah, M. Kaushik, R. Sharma, S. A. Peious, and D. Draheim, "Cluster-based association rule mining for an intersection accident dataset," in *Proc. Int. Conf. Comput., Electron. Electr. Eng. (ICE Cube)*, Oct. 2021, pp. 1–6.

[8] J. Guan, S. Li, X. Chen, X. He, and J. Chen, "DEMOS: Clustering by pruning a density-boosting cluster tree of density mounts," *IEEE Trans. Knowl. Data Eng.*, vol. 35, no. 10, pp. 10814–10830, Oct. 2023.

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

[10] X. Xu, S. Ding, Y. Wang, L. Wang, and W. Jia, "A fast density peaks clustering algorithm with sparse search," *Inf. Sci.*, vol. 554, pp. 61–83, Apr. 2021.

[11] W. Tong, Y. Wang, and D. Liu, "An adaptive clustering algorithm based on local-density peaks for imbalanced data without parameters," *IEEE Trans. Knowl. Data Eng.*, vol. 35, no. 4, pp. 3419–3432, Apr. 2023.

[12] X. Fang, Z. Xu, H. Ji, B. Wang, and Z. Huang, "A grid-based density peaks clustering algorithm," *IEEE Trans. Ind. Informat.*, vol. 19, no. 4, pp. 5476–5484, Apr. 2023.

[13] C. Li and Y. Zhang, "Density peak clustering based on relative density optimization," *Math. Problems Eng.*, vol. 2020, pp. 1–8, Jun. 2020.

[14] J. Hou, A. Zhang, and N. Qi, "Density peak clustering based on relative density relationship," *Pattern Recognit.*, vol. 108, Dec. 2020, Art. no. 107554.

[15] J. Hou and A. Zhang, "Enhancing density peak clustering via density normalization," *IEEE Trans. Ind. Informat.*, vol. 16, no. 4, pp. 2477–2485, Apr. 2020.

[16] R. Mehmood, G. Zhang, R. Bie, H. Dawood, and H. Ahmad, "Clustering by fast search and find of density peaks via heat diffusion," *Neurocomputing*, vol. 208, pp. 210–217, Oct. 2016.

[17] Y. Zhu, K. M. Ting, and M. J. Carman, "Density-ratio based clustering for discovering clusters with varying densities," *Pattern Recognit.*, vol. 60, pp. 983–997, Dec. 2016.

[18] C. Wu, J. Lee, T. Isokawa, J. Yao, and Y. Xia, "Efficient clustering method based on density peaks with symmetric neighborhood relationship," *IEEE Access*, vol. 7, pp. 60684–60696, 2019.

[19] T. Xu and J. Jiang, "A graph adaptive density peaks clustering algorithm for automatic centroid selection and effective aggregation," *Expert Syst. Appl.*, vol. 195, Jun. 2022, Art. no. 116539.

[20] S. Wang, D. Wang, C. Li, Y. Li, and G. Ding, "Clustering by fast search and find of density peaks with data field," *Chin. J. Electron.*, vol. 25, no. 3, pp. 397–402, May 2016.

[21] R. Mehmood, R. Bie, H. Dawood, and H. Ahmad, "Fuzzy clustering by fast search and find of density peaks," in *Proc. Int. Conf. Identificat., Inf., Knowl. Internet Things (IIKI)*, Oct. 2015, pp. 258–261.

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

[23] M. Du, S. Ding, and H. Jia, "Study on density peaks clustering based on k-nearest neighbors and principal component analysis," *Knowl.-Based Syst.*, vol. 99, pp. 135–145, May 2016.

[24] D. Cheng, J. Huang, S. Zhang, and H. Liu, "Improved density peaks clustering based on shared-neighbors of local cores for manifold data sets," *IEEE Access*, vol. 7, pp. 151339–151349, 2019.

[25] C. Lei, R. X. Wu, and P. W. Li, "Weighted K-nearest neighbors and multi-clusters merge density peaks clustering algorithm," *J. Frontiers Comput. Sci. Technol.*, vol. 16, no. 9, pp. 2163–2176, 2022.

[26] J. Zhao, Z. F. Yao, and L. Yu, "Density peaks clustering based on mutual neighbor degree," *Control. Decis.*, vol. 36, no. 3, pp. 543–552, 2021.

[27] 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.

[28] C. W. Wu, Y. F. Jiang, and N. Ma, "Density peak clustering algorithm combined with KNN and label propagation," *J Northwest Univ., Nat. Sci. Ed.*, vol. 50, no. 6, pp. 979–986, 2020.

[29] Z. Long, Y. Gao, H. Meng, Y. Yao, and T. Li, "Clustering based on local density peaks and graph cut," *Inf. Sci.*, vol. 600, pp. 263–286, Jul. 2022.

[30] W. C. Chen, J. Zhao, and R. B. Xiao, "Density peaks clustering algorithm with nearest neighbor optimization for data with uneven density distribution," *Control. Decis*, vol. 39, no. 3, pp. 919–928, 2024.

[31] *M. Lichman, UCI Machine Learning Repository*. [Online]. Available: http://archive.ics.uci.edu/ml

[32] D. Cheng, Q. Zhu, J. Huang, Q. Wu, and L. Yang, "Clustering with local density peaks-based minimum spanning tree," *IEEE Trans. Knowl. Data Eng.*, vol. 33, no. 2, pp. 374–387, Feb. 2021.

[33] N. X. Vinh, J. Epps, and J. Bailey, "Information theoretic measures for clusterings comparison: Variants, properties, normalization and correction for chance," *J. Mach. Learn. Res.*, vol. 11, pp. 2837–2854, Jan. 2010.

[34] E. B. Fowlkes and C. L. Mallows, "A method for comparing two hierarchical clusterings," *J. Amer. Stat. Assoc.*, vol. 78, no. 383, p. 553, Sep. 1983.

[35] H. Z. Lv, Y. Y. Yang, G. P. Yang, and Z. G. Gong, "K-NN density dominator component delegations based density peaks clustering," *Comput. Eng. Appl.*, vol. 59, no. 24, pp. 78–87, 2023.

[36] 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.

[37] H. Ju, Y. Lu, W. Ding, J. Cao, and X. Yang, "Three-way evidence theory-based density peak clustering with the principle of justifiable granularity," *Appl. Soft Comput.*, vol. 152, Feb. 2024, Art. no. 111217.

[38] G. Yang, H. Lv, Y. Yang, Z. Gong, X. Chen, and Z. Hao, "FastDEC: Clustering by fast dominance estimation," in *Proc. Joint Eur. Conf. Mach. Learn. Knowl. Discovery Databases*, vol. 13713, 2023, pp. 138–156.

SHIBO ZHOU received the B.S. degree from the Navigation College, Jimei University, in 2003, the M.S. degree from Shanghai Maritime University, in 2005, and the Ph.D. degree from Beijing Jiaotong University, in 2018. He is currently a Professor with Jimei University. His extensive research contributions, broad research interests, and eagerness to apply his findings to real-world applications make him a valuable asset to the academic and professional community. His main research interests include data mining, system analysis, and integration.

BINGBING PENG received the bachelor's degree in traffic engineering from Nantong University, Jiangsu, China, in 2022. He is currently pursuing the master's degree in transportation with Jimei University. He is a dedicated master's student with a strong passion for research in the field of transportation engineering. He is eager to deepen his understanding of these fields and apply his findings to practical applications, particularly in the domains of transportation and data mining. His current research interests include data mining, deep learning, and artificial intelligence.

WENPENG XU received the bachelor's degree in traffic engineering, in 2020, and the M.S. degree from the Navigation College, Jimei University, in 2022. His main research interests include data mining, maritime big data analytics, and ship trajectory analysis.

LÜZHEN REN received the B.S. degree from the Navigation College, Jimei University, in 1986, and the M.S. degree from Shanghai Maritime University, in 1996. He is currently an Associate Professor with Jimei University. His main research interests include data mining, maritime big data analytics, and ship trajectory analysis.