frontiers | Frontiers in Applied Mathematics and Statistics

TYPE Original Research
PUBLISHED 01 September 2025
DOI 10.3389/fams.2025.1598165

OPEN ACCESS

EDITED BY
Daniel Potts,
Chemnitz University of Technology, Germany

REVIEWED BY
Kai Bergermann,
Chemnitz University of Technology, Germany
Michael Quellmalz,
Technical University of Berlin, Germany

*CORRESPONDENCE
Chaorong Li
lichaorong88@163.com

RECEIVED 25 March 2025
ACCEPTED 31 July 2025
PUBLISHED 01 September 2025

CITATION
Ren C, Li C, Yu Y, Yang W and Guo R (2025) Density peak clustering algorithm based on weighted mutual K-nearest neighbors. *Front. Appl. Math. Stat.* 11:1598165. doi: 10.3389/fams.2025.1598165

Density peak clustering algorithm based on weighted mutual K-nearest neighbors

Chunhua Ren[1], Chaorong Li[1]*, Yang Yu[2], Wanan Yang[1] and Ruiqi Guo[1]

[1]School of Computer Science and Technology, Yibin University, Yibin, China, [2]School of Computer and Software, Southwest Petroleum University, Chengdu, China

Ever since Density Peak Clustering (DPC) was published in Science, it has been widely favored and applied in various fields due to its concise and efficient computational theory. However, DPC has two major flaws. On the one hand, it fails to find cluster centers of low-density clusters in datasets with uneven density distribution. On the other hand, its single assignment strategy, which only assigns points to high-density clusters, can lead to incorrect clustering due to a chain reaction. To address these weaknesses, a new density peak clustering algorithm based on weighted mutual K-nearest neighbors called WMKNNDPC is proposed in this paper. WMKNNDPC offers two significant advantages: (1) It introduces the concept of mutual K-nearest neighbors by using K-nearest neighbors and inverse K-nearest neighbors, allowing for the identification of cluster centers in clusters with uneven density distribution through a new local density calculation method. (2) It includes a remaining points assignment method based on weighted mutual K-nearest neighbors, which involves two stages: first, the initial assignment of data points is done by combining mutual K-nearest neighbors and breadth-first search, and second, the membership degree of data points is calculated based on weighted mutual K-nearest neighbors for remaining points assignment. This method allows for efficient assignment based on the local distribution of points, avoiding the disadvantages of using a fixed K-value in DPC-derived algorithms based on K-nearest neighbors. The WMKNNDPC algorithm has been extensively tested on two-dimensional synthetic datasets, real datasets, facial recognition dataset and parameter analysis. The experimental results indicate that our algorithm performs the best on most datasets.

KEYWORDS

K-nearest neighbors, inverse K-nearest neighbors, weighted mutual K-nearest neighbors, local density, remaining points assignment, density peak clustering

1 Introduction

Nowadays, due to the explosive growth of social data, data mining has become widespread in various industries, helping people understand data and make informed decisions. Clustering technology is an unsupervised learning method in data mining that focuses on a large amount of unlabeled data and uses data point similarity to classify. This results in highly similar data being grouped into the same category, while there is low data similarity between different categories [1]. Clustering algorithms can be categorized into partition clustering [2], hierarchical clustering [3], density clustering [4], grid clustering [5], model clustering [6], and graph clustering [7] based on different partition theories. These algorithms have been successfully applied in areas such as customer segmentation

[8], image processing [9–11], product recommendation [12], social networks [13], big data application [14] and data security [15].

So far, several traditional clustering algorithms have been developed, such as K-means [16], DBSCAN [17, 18], Spectral Clustering (SC) [19], and Expectation-Maximization Clustering (EM) [20], along with numerous enhanced methods. However, they are unable to effectively handle datasets of various shapes, sizes, and densities.

In 2014, Rodriguez and Laio [21] published a paper in Science titled "Clustering by fast search and find of density peaks." The paper introduced a clustering algorithm called DPC, which gained significant attention from researchers. DPC is a density-based clustering algorithm that operates on two key assumptions: low-density data points tend to cluster near the center point, and different cluster centers are usually far apart. This algorithm requires only one parameter and demonstrates efficient clustering center identification and data point assignment for most datasets, delivering strong clustering performance.

While the DPC process is straightforward and clustering is effective, there are several issues, with two notable shortcomings. One problem is that when dealing with manifold datasets, the strategy for assigning remaining points can easily cause a chain reaction, leading to significant assignment errors, as depicted in Figure 1a. In this dataset, comprising three clusters, the data points on both sides of the semicircle are incorrectly assigned to the other two clusters. Another issue arises when dealing with datasets featuring uneven density distribution. The lack of consideration for sparsity between data points when calculating local density can result in errors in selecting cluster centers, as shown in the dataset in Figure 1b. This dataset contains two clusters with uneven density distribution, leading to issues with multiple peaks when calculating candidate cluster centers.

In response to the shortcomings of DPC, numerous researchers have conducted a series of work, primarily focusing on two aspects: optimizing local density calculation and improving assignment strategy. In the meantime, several research findings have been produced. These include concepts such as K-nearest neighbors [22], fuzzy K-nearest neighbors [23], shared K-nearest neighbors [24], layered K-nearest neighbors [25], reverse K-nearest neighbors [26], and so on, which have been incorporated into DPC. However, these improved algorithms still use a fixed K-value parameter and lack consideration for the local distribution of the dataset.

Therefore, this paper proposed a new density peak clustering algorithm WMKNNDPC based on weighted mutual K-nearest neighbors, which, like most derived-DPC algorithms, requires a mutual K-nearest neighbors hyper-parameter. The WMKNNDPC algorithm can identify clusters with arbitrary shapes, densities, and sizes, and it offers two major contributions: (1) It defines mutual K-nearest neighbors based on K-nearest neighbors and inverse K-nearest neighbors, and redesigns the local density of mutual K-nearest neighbors. This method is adaptable, no longer using a fixed parameter K, and can calculate more accurate local density based on the local points distribution, which makes it easier to select the correct clustering center and avoid the issue of multiple peaks. (2) It introduces a remaining points assignment method based on weighted mutual K-nearest neighbors, utilizing mutual K-nearest neighbors, breadth-first search, and points membership probability for assigning. This approach helps in avoiding point assignment errors and stopping the domino effect.

The paper is organized as follows: In Section 2, we analyze the latest progress of the DPC algorithm. In Section 3, we provide a detailed introduction to the proposed WMKNNDPC algorithm. In Section 4, we conduct a large number of experiments and compare our algorithm with seven classic algorithms for analysis. In the Section 5, we provide a summary of the works and looks forward to the future.

2 Related works

This section explains the basic theory of the DPC algorithm and analyzes the latest research status of DPC-derived algorithms.

2.1 DPC analysis

The DPC clustering process is straightforward and easy to comprehend [21]. Firstly, it defines two methods for local density calculation, which are calculated using piecewise functions and Gaussian kernels, as shown in Equations 1, 2.

$$\rho_i = \sum_{x_j} \chi(d_{ij} - d_c), \chi(z) = \begin{cases} 1, & z < 0 \\ 0, & z \geq 0 \end{cases} \tag{1}$$

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

Where ρ_i represents the local density of data point x_i, d_{ij} is the Euclidean distance from data point x_i to x_j, and d_c is the only truncation distance parameter. $\chi(.)$ is a piecewise function defined as follows: If the parameter z is less than 0, $\chi(z)$ equals 1, otherwise $\chi(z)$ equals 0. According to DPC, local density calculation for large-scale datasets is more suitable for Equation 1, otherwise Equation 2 is used. However, this presents a problem: researchers do not know which method of local density calculation is more accurate and effective for different datasets. As a result, only two methods are employed for calculation, leading to additional computational workload.

Secondly, DPC defines another important variable, the relative distance δ_i , which represents the shortest distance from data point x_i to data point x_j with higher density. The calculation method is shown in Equation 3.

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

When data point x_i has the maximum local density, DPC considers it a peak point and sets the relative density to the maximum value using Equation 4.

$$\delta_i = \max_{x_j}(d_{ij}) \tag{4}$$

Once the local density and relative distance are calculated, DPC identifies potential cluster centers using a decision graph. These

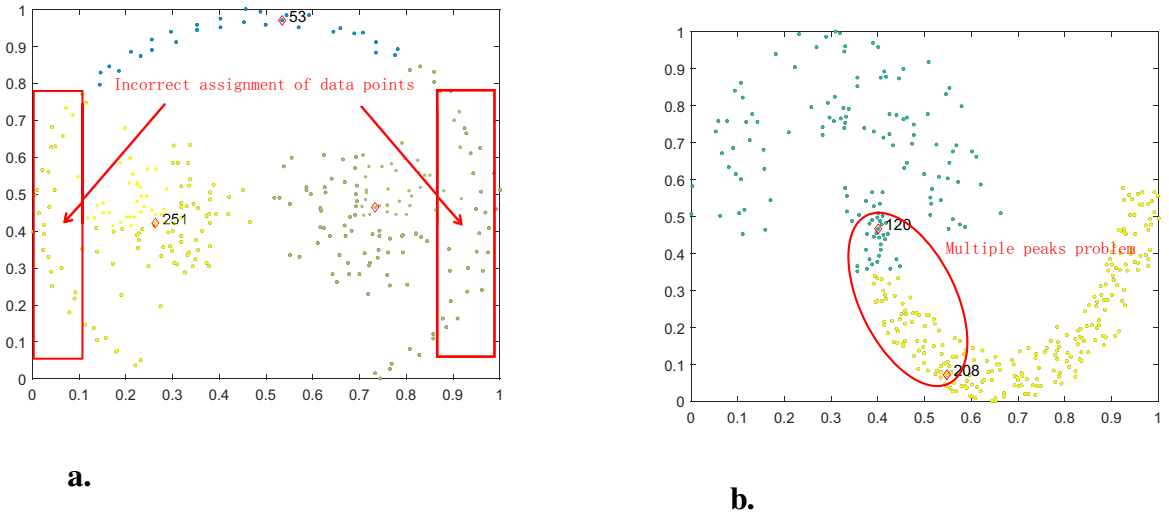

FIGURE 1
Problems faced by DPC in pathbased and jain datasets. **(a)** DPC on pathbased. **(b)** DPC On Jain.

potential cluster centers are characterized by having the highest local density and relative distance simultaneously for data point x_i. The process for calculating potential cluster centers is outlined in Equation 5.

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

DPC usually selects points with high local density and relative distance as actual clustering centers, or manually identifies significant outliers by drawing decision graph to determine the number of clustering centers. Finally, assign the remaining points sequentially to the nearest and denser data points.

2.2 Research progress of DPC

Since the density peak clustering algorithm was published in 2014, numerous researchers have conducted extensive research in the past decade to address the shortcomings of DPC.

The first stage is between 2016 and 2019. At the beginning, the density peaks clustering based on K-nearest neighbors (DPC-KNN) algorithm [27] redefined a new local density using the average distance from the data point to K-nearest neighbors, while considering the distribution differences between points, avoiding the limitation of DPC using a unified truncation distance when defining local density, and achieving good clustering performance. However, DPC-KNN has two shortcomings. Firstly, it still uses the percentage parameter to obtain K-nearest neighbors, which is very sensitive and makes it difficult to determine the optimal value. Secondly, it cannot accurately obtain cluster centers when dealing with datasets with uneven density distribution. Subsequently, the fuzzy weighted K-nearest neighbors density peak clustering algorithm (FKNN-DPC) was proposed by Xie et al. [23], which has better robustness compared to DPC and DPC-KNN. FKNN-DPC adopts the K-nearest neighbors method and fuzzy set theory to design a new local density and develops a two-stage remaining point assignment strategy. However, this method uses a fixed K-value each time to calculate local density and assign sample points, without considering the local distribution of sample points. To address the issue of DPC dependency and truncation distance parameters, the Natural Neighbor-based clustering algorithm with density peaks (NaNDP) [28] was first formed by introducing the idea of natural neighbors. NaNDP does not require additional parameters and can expand from the cluster center by searching for the natural neighborhood of the cluster midpoint. Finally, extension rules are defined to determine the boundaries of the cluster. However, NaNDP still adopts the assignment principle of DPC, which is not good for handling boundary points. To solve the problem of DPC not being able to correctly select cluster centers, the adaptive density peak clustering based on K-nearest neighbors with aggregating strategy (ADPC-KNN) [29] designed a method for automatically selecting initial cluster centers and improved the clustering performance by using the idea of cluster density reachability. However, the unique parameter K of this method needs to be manually preset. The shared-nearest-neighbor-based clustering by fast search and find of density peaks (SNN-DPC) algorithm was proposed by Liu et al. [24], which redefines local density based on nearest neighbors and shared neighbors, which can better adapt to the local environment of sample points. At the same time, a two-stage remaining point assignment strategy was also proposed based on shared neighbors. SNN-DPC can effectively handle various datasets, but its problems are consistent with FKNN-DPC, which requires the use of fixed K-nearest neighbors parameter in the clustering process. Based on two assumptions in DPC, the comparative density peaks clustering (CDP) [30] is a clustering algorithm based on comparative quality and density measurement, and experiments have shown that CDP has better

performance than DPC in most cases. A feasible density peaks clustering algorithm with a merging strategy (FDPC) [31] has been proposed to address the issue of DPC being unable to handle multiple peaks. It uses support vector machines to calculate the feedback values between clusters after finding the initial cluster center, and clusters based on the feedback values. In response to the dilemma of DPC being unable to distinguish overlapping clusters, Parmar et al. [32] designed a residual error-based density peak clustering algorithm (REDPC), which uses residuals to calculate local density and identify low-density sample points. This method can generate a decision graph that is more conducive to clustering, but it has multiple process parameters and poor autonomy.

The second stage, from 2020 to the present, has further generated a series of DPC-derived algorithms. In 2020, a novel systematic density based clustering method using anchor points (APC) [33] was proposed, which uses anchor points as the center to obtain intermediate clusters and automatically selects appropriate clustering strategies. The experimental results show that APC has good clustering performance in most cases. Although APC combines the advantages of DPC and DBSCAN, the four custom parameters incur significant time overhead. Ren et al. [25] proposed an improved density peaks clustering algorithm based on the layered K-nearest neighbors and subcluster merging (LKSM-DPC) to address the issue of multiple peaks in DPC. This algorithm uses layered K-nearest neighbors to define local density, which is beneficial for extracting cluster centers. At the same time, a subcluster merging strategy based on shared neighbors and universal gravitation is designed, and comparative experiments show that LKSM-DPC has some advantages. But LKSM-DPC also has obvious shortcomings, that is, selecting how many sub-clusters to merge requires a lot of trying, which increases the time cost. To cope with imbalanced datasets, a novel clustering algorithm related to density based clustering algorithm for identifying diverse density clusters effectively named IDDC [34] has emerged. It determines the local density of sample points by defining relative density and searches for unassigned points from a clustering perspective, designing a new assignment strategy. However, IDDC requires two parameters. A graph adaptive density peaks clustering algorithm based on graph theory (GADPC) [35] was proposed in 2022. GADPC based on the turning angle and graph connectivity automatically selects cluster centers, and the remaining points move closer to the corresponding cluster centers. The algorithm is more feasible when dealing with datasets with different densities such as Jain and Spiral, but it ignores the issue of selecting parameters for the same DPC. Similarly, for imbalanced datasets, Zhao et al. [36] proposed a new density peaks clustering algorithm based on fuzzy and weighted shared neighbor (DPC-FWSN), which redefines local density using the nearest neighbors fuzzy kernel function and designs a weighted shared neighbors similarity assignment strategy. Experimental tests have shown that DPC-FWSN can effectively handle datasets with uneven density distribution, but its drawback is that it still uses a manual setting of the nearest neighbors parameter K. Based on FKNN-DPC, Xie et al. [37] further proposed the standard deviation weighted distance based density peak clustering algorithm (SFKNN-DPC) with better robustness, considering the contribution of each feature to the distance between data points, adopting the standard deviation weighted distance, and designing a divide and conquer assignment strategy. For DPC, which cannot find cluster centers in sparse clusters, an adaptive nearest neighbors density peak clustering algorithm (ANN-DPC) [38] was proposed. Firstly, the adaptive nearest neighbors of points are introduced to accurately define the local density of points. Then, the sample points are divided into super score, core, linked, and slave points to check for suitable cluster centers. Finally, a new assignment strategy is designed using the adaptive nearest neighbors algorithm combined with breadth-first search and fuzzy weighted adaptive nearest neighbors algorithm. For ANN-DPC, the performance is excellent, but it is still necessary to specify the number of clusters in advance. Fan et al. [39] designed a density peak clustering based on improved mutual K-nearest neighbor graph (MKNNG-DPC) by defining K-nearest-neighbor sample set, distance-upper-bound point, distance level and mutual K-nearest-neighbors set. However, this method requires consideration of constructing a mutual K-nearest neighbor graph, and at the same time requires two parameters: truncation distance d_c and K-nearest neighbors. In 2023, Li et al. [40] constructed a new local density based on MKNNG-DPC and designed a two-stage sub cluster merging method, thus proposing a fast density peaks clustering algorithm based on improved mutual K-nearest-neighbors and sub-cluster merging (KS-FDPC). However, calculating the similarity of sub-clusters and merging them both require additional storage and time overhead.

3 WMKNNDPC algorithm

In this section, we will provide a detailed introduction to our proposed density peak clustering algorithm based on weighted mutual K-nearest neighbors (WMKNNDPC). For the specific technical details, please refer to the following sections.

3.1 Mutual K-nearest neighbors local density

In DPC, local density calculation depends on the truncation distance d_c. However, determining the optimal value for d_c can be challenging when working with different datasets. KNN-DPC addresses this issue by using the K-nearest neighbors method to calculate local density [27]. While this approach eliminates the need to determine d_c, it overlooks the distribution around data points, and using a fixed K-value can result in calculation errors when identifying cluster centers. Therefore, this section introduces a new method for local density calculation called mutual K-nearest neighbors local density.

First, calculate the K-nearest neighbors set $KNN(x_i)$ of data point x_i using Equation 6 and sort it in ascending order by Euclidean distance.

$$KNN(x_i) = \{x_j \in D \,|\, d_{iK} - d_{ij} \geq 0\} \tag{6}$$

where D represents the dataset, and d_{iK} represents the distance between data point x_i and its K-th nearest neighbor.

The inverse K-nearest neighbors set $RNN(x_i)$ of data point x_i is defined as shown in Equation 7. $RNN(x_i)$ indicates the influence of data point x_i in the dataset. If point x_i is in a high-density region, it is usually surrounded by more data points. Conversely, if it is in a sparse region, the number of inverse neighbors is less. The size of the inverse K-nearest neighbors set is more likely to provide feedback on the local distribution of the data points.

$$RNN(x_i) = \{x_j \in D | x_i \in KNN(x_j)\} \tag{7}$$

For a given data point x_i, the mutual K-nearest neighbors $MKNN(x_i)$ are defined based on the K-nearest neighbors and inverse K-nearest neighbors. It represents the bidirectional local relationship between data point x_i and its neighboring points. The set of mutual K-nearest neighbors for individual points is initially empty. However, this paper suggests that the mutual K-nearest neighbors set of these points should be the top 50% ($Top\lceil K/2\rceil$, $K \in \mathbb{N}^+$ and $1 < K < |D| - 1$) of the K-nearest neighbor set. The calculation method can be found in Equation 8.

$$MKNN(x_i) = \begin{cases} KNN(x_i) \cap RNN(x_i), \{KNN(x_i) \cap RNN(x_i)\} \neq \varnothing \\ Top_{\lceil K/2 \rceil}\left(KNN(x_i)\right), \{KNN(x_i) \cap RNN\,(x_i)\} = \varnothing \end{cases} \tag{8}$$

Experimental analysis indicates that the number of mutual K-nearest neighbors for each data point varies, suggesting that it is more appropriate to describe the local situation of points using mutual K-nearest neighbors rather than fixed K-nearest neighbors. Using mutual K-nearest neighbors provides the benefit of accurately describing the local nearest neighbors structure of each data point, particularly in distinguishing low-density areas. Additionally, mutual K-nearest neighbors not only consider the calculation of local density but also facilitate the assignment of remaining points.

Equation 9 describes the number of mutual K-nearest neighbors of data point x_i, denoted as $MK(x_i)$, where $|.|$ represents the number of data points in the set $MKNN(x_i)$.

$$MK(x_i) = |MKNN(x_i)| \tag{9}$$

Based on mutual K-nearest neighbors, we have designed a new method of local density ρ_i for any data point x_i, as shown in Equation 10. This method is reconstructed from the original local density calculation method, where the numerator represents the number of mutual K-nearest neighbors. Due to the inconsistency of mutual K-nearest neighbors for each data point, it has adaptability, which is completely different from the K-nearest neighbors method. The denominator in the equation is the contribution of the K-nearest neighbors of the data point x_i to the local density. If the sum of the distances between x_i and the K-nearest neighbors is smaller, the local density will be larger.

$$\rho_i = \frac{MK(x_i)}{\frac{1}{MK(x_i)} \sum_{x_j \in MKNN(x_i)} d_{ij}} \tag{10}$$

As per Equation 10, having more mutual K-nearest neighbors between data points implies that the data point is surrounded by more points, which significantly impacts the local density. When multiple data points have the same mutual K-nearest neighbors, it's important to consider the local distribution around the data points, specifically the Euclidean distance between the data points and their K-nearest neighbors. This helps to better identify data with uneven density distribution in different regions and select the correct clustering center.

For example, Figure 2 contains 20 data points, with two diamond center points labeled A and B, and the remaining 18 as circular points. The black line represents the distance between the data points and the center point. If the K-nearest neighbors parameter K is set to 7, the circular points in the dashed ellipse represent the K-nearest neighbors of A and B. We can conclude that the actual local density at center point A is lower than that at center point B, which is $\rho_A < \rho_B$. For data point x_i with $\rho_i = \frac{K}{\frac{1}{K}\sum_{x_j \in KNN(x_i)} d_{ij}}$, we can obtain a result where $\rho_A = 4.05$ is greater than $\rho_B = 3.04$, which is inconsistent with reality. This is the influence of using a fixed K-value in K-nearest neighbors, and data point 17 also serves as the K-nearest neighbor of center point B. If the local density calculation method designed in this paper is used, point 17 will not be recognized as the mutual K-nearest neighbors of center B. This method has adaptability and can obtain $\rho_B = 6.43$ greater than ρ_A, which is more in line with actual expectations.

3.2 A remaining points assignment method based on weighted mutual K-nearest neighbors

The incorrect assignment of remaining points in DPC can trigger a chain reaction. The strategy used in the assignment should prioritize high-density clusters which can lead to points from sparse clusters being incorrectly assigned to dense clusters. In this paper, a new method for assigning remaining points based on weighted mutual K-nearest neighbors is proposed, inspired by FKNN-DPC [23]. FKNN-DPC uses a fixed K-value for each assignment without considering the local distribution around data points. Instead, our proposed method utilizes weighted mutual K-nearest neighbors to assign these remaining points. This approach, in contrast to FKNN-DPC, is more suitable for capturing the local distribution of points due to the flexibility of the weighted mutual K-nearest neighbors for each data point.

The data points assignment method we propose consists of two algorithms. Algorithm 1 uses mutual K-nearest neighbors and breadth-first search of data points for initial priority assignment. Building on Algorithms 1, 2 calculates the membership probability of the remaining unassigned points using weighted mutual K-nearest neighbors and then assigns high-probability data points to the most suitable cluster. If there are still unassigned points, they are classified based on their nearest neighbors. This method can quickly assign remaining points in dense areas, where the number of mutual K-nearest neighbors of data points is relatively large. In sparse areas, the number of mutual K-nearest neighbors can be used for small-scale assignments to reduce errors. It effectively adapts to local conditions and demonstrates strong robustness.

The similarity S_{ij} between data points x_i and x_j is defined as follows: if the distance between data points is smaller and the S_{ij}

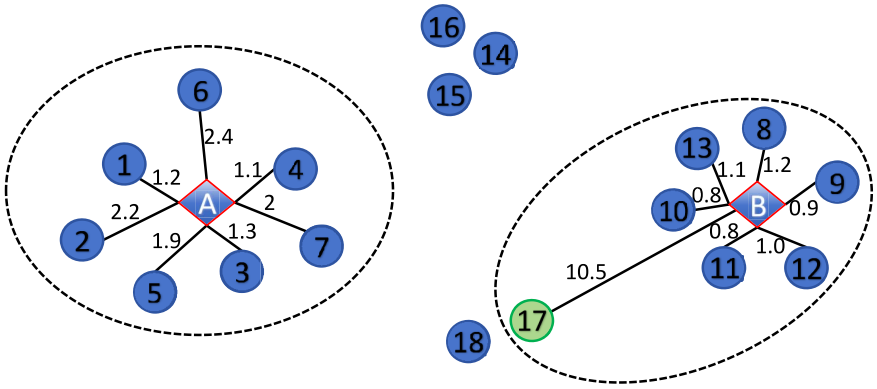

FIGURE 2
The influence of the K-nearest neighbors method on local density.

value is larger, it indicates that the two data points are more similar, as shown in Equation 11.

$$S_{ij} = 1 - \frac{d_{ij}}{d_{ij} + 1} \tag{11}$$

To calculate the assignment probability $p(x_i, c_t)$ of a data point x_i belonging to cluster c_t, we define w_{ij} as a weight. This weight is obtained by the similarity S_{ij} and the mutual K-nearest neighbors of the data point x_j, as shown in Equation 12.

$$w_{ij} = \frac{S_{ij}}{\sum_{x_q \in MKNN(x_j)} S_{qj}} \tag{12}$$

The value of $w_{ij}S_{ij}$ represents the weighted contribution of data point x_j to $p(x_i, c_t)$. This consideration takes into account the distance between data points x_i and x_j, as well as the weighted mutual K-nearest neighbors distribution of data points x_j. This approach helps assign the remaining points to the most suitable cluster. The calculation method for $p(x_i, c_t)$ is shown in Equation 13.

$$p(x_i, c_t) = \sum\nolimits_{x_j \in MKNN(x_i), l_{x_j} = c_t} w_{ij} S_{ij} \tag{13}$$

where c_t is a cluster center and belongs to the cluster center set C. $l_{x_j} = c_t$ indicates that the label of data point x_j belongs to c_t. $p(x_i, c_t)$ is explicitly normalized using $p(x_i, c_t) = p(x_i, c_t) / \sum_{c_r \in C} p(x_i, c_r)$.

The following are the remaining points assignment Algorithms 1, 2.

In Algorithm 1, there is a key criterion for initial assignment of data point x_p, which is Step 7. The purpose of step 7 is to ensure the accuracy of initial assignment as much as possible. In other words, when data point x_p is unassigned, x_p is the mutual K-nearest neighbor of x_q and the distance between data point x_p and x_q is less than half of the sum of all mutual K-nearest neighbor distances of x_q.

Require: Cluster center set C, the set $MKNN$ in dataset D, distance matrix Dist.
Ensure: Clustering results of D.
1: **for** each $c_i \in C$ **do**
2: Set the label of $MKNN(c_i)$ as l_{c_i};
3: AddQue(Que, $MKNN(c_i)$); ▷ Que is a queue, and AddQue represents the enqueue operation
4: **while** $Que \neq \phi$ **do**
5: GetHead(Que, x_q); ▷ GetHead represents the operation of getting the head elements of the queue
6: **for** each $x_p \in MKNN(x_q)$ **do**
7: **if** $l_{x_p} == 0$ **and** $d_{pq} \leq (\sum_{x_j \in MKNN(x_q)} d_{qj})/MK(x_q)$ **then**
8: Set the label of x_p as l_{c_i};
9: AddQue(Que, x_p));
10: **end if**
11: **end for**
12: RemoveHead(Que, x_p)); ▷ RemoveHead represents the operation of removing the head element of the queue
13: **end while**
14: **end for**

Algorithm 1. Initial assignment algorithm based on mutual K-nearest neighbors and breadth-first search.

3.3 The procedure of WMKNNDPC

This section provides a detailed description of the WMKNNDPC algorithm process. To illustrate the proposed algorithm, Figure 3 presents a simple algorithm flowchart. The WMKNNDPC algorithm can be divided into several steps: (1) Input the dataset and calculate the Euclidean distance matrix between the data. (2) Calculate the mutual K-nearest neighbors for each data point by combining K-nearest neighbors and inverse K-nearest neighbors. (3) Calculate the local density and relative distance of each point. (4) Construct a two-dimensional decision

Require: the unassigned remaining points set $D_n \subseteq D$, the set $MKNN$ in dataset D, C.
Ensure: Clustering results of D.
for each $x_i \in D_n$ **do**
  Calculate $p(x_i, c_t)$ using Equations 11–13 for all $t = 1, 2, \cdots, m$;
  Construct membership P, where $P[i,t] = p(x_i, c_t)$; ▷ P is a n-by-m matrix
  Construct max membership MP, where $MP[i,0] = \max_{t=1,\cdots,m} P[i,t]$ and $MP[i,1] = \arg\max_{t=1,\cdots,m} P[i,t]$; ▷ MP is a n-by-2 matrix, $MP[i,0]$ stores maximum membership probability, $MP[i,1]$ stores corresponding cluster label
end for
Select x_p where $p = \arg\max_i MP[i,0]$;
if $P[p, MP[p,1]] \neq 0$ **then**
  $flag$=1;
end if
while $flag$ **do**
  Assign label $l_{x_p} = MP[p,1]$;
  Update $P[p,t] = 0$ for all t;
  Update $MP[p,0] = 0$ and $MP[p,1] = 0$;
  for each unassigned point $x_q \in MKNN(x_p)$ **do**
    Update $P[q, MP[q,1]] = P[q, MP[q,1]] + w_{pq}S_{pq}$;
    Recompute $MP[q,0] = \max_{t=1..m} P[q,t]$;
    Recompute $MP[q,1] = \arg\max_{t=1..m} P[q,t]$;
  end for
  Select x_p where $p = \arg\max_i MP[i,0]$;
  if $P[p, MP[p,1]] == 0$ **then**
    $flag$=0;
  end if
end while
while $x_j \in D_n$ **and** $l_{x_j} == 0$ **do**
  Assign label $l_{x_j} = l_{x_1}$, where $x_1 \in Top(MKNN(x_j))$ and $l_{x_1} \neq 0$;
end while

Algorithm 2. Membership assignment algorithm based on weighted mutual K-nearest neighbors.

graph. (5) Perform initial priority assignment by Algorithm 1. (6) Calculate the membership degree of the remaining points and classify them using Algorithm 2. (7) Obtain the final clustering result. Algorithm 3 provides a detailed implementation process.

3.4 Time complexity analysis

This section focuses on evaluating the time complexity of WMKNNDPC. Assuming the size of the dataset is n, where k represents the number of nearest neighbors. According to the algorithm process, the time complexity is composed of the following parts: (1) The time consumption for calculating the Euclidean distance matrix is $O(n^2)$. (2) Calculate local density, including calculating K-nearest neighbors, inverse K-nearest neighbors, and mutual K-nearest neighbors, with time complexity of $O(k\log n)$, $O(kn)$, and $O(n^2)$, respectively. (3) The time complexity of calculating the relative density between data points is $O(n^2)$. (4) The key to assignment Algorithm 1 is using the breadth-first search, which requires one queue. In the worst-case scenario, all data points are queued, with a time complexity of $O(n)$. (5) In Algorithm 2, two aspects need to be considered: first, calculating the membership matrix of all unassigned points in the dataset and selecting the data point with the highest membership for category assignment, with a time complexity of $O(kn)$; second, updating the membership matrix and calculating the membership probability of the data point, in the worst-case scenario is $O(n^2)$.

Based on the above analysis, we can conclude that the time complexity of WMKNNDPC is approximately $O(n^2)$, which is consistent with the original DPC algorithm.

4 Experiments and analyses

In this section, we will assess the performance of the WMKNNDPC algorithm and compare it with several other clustering algorithms, including K-means [16], DBSCAN [18], DPC [21], and DPC-derived algorithms such as DPC-KNN [27], FKNN-DPC [23], the density peaks clustering based on weighted local density sequence and nearest neighbor assignment (DPCSA) [41], and the density peaks clustering based on local fair density and fuzzy K-nearest neighbor membership allocation strategy (LF-DPC) [42]. To ensure the consistency of the experiment, all tests were carried out in a consistent software and hardware environment, utilizing an i5-11400H @ 2.70GHz CPU, 16GB RAM, WIN10 x64 OS, and MATLAB 2015b programming software.

4.1 Test datasets

The test datasets consist of eight classic artificially synthetic 2D datasets (Abbreviation: synthetic datasets) [43] and ten real datasets from the UCI database [44]. Testing the performance of the proposed algorithm from different perspectives will be helpful due to variations in data size, dimensions, and category numbers in the test datasets. Detailed information about the datasets can be found in Tables 1, 2.

4.2 Evaluation criteria and experimental parameters

To assess the effectiveness of the proposed algorithm in this paper, three classic external clustering evaluation indicators were chosen as the metrics for the algorithm. These are the Adjusted Rand Index [ARI [45]], Adjusted Mutual Information [AMI [46]], and Folkes-Mallows Index [FMI [47]]. The closer the values of these three evaluation indicators are to 1, the better the clustering performance will be.

To compare different clustering algorithms, we fine-tuned their parameters to achieve the best clustering performance. For K-means clustering, the results are averaged over 10 runs due to the initial center's instability. The only parameter for K-means is the number of clusters. DBSCAN requires two parameters: the radius distance e and the minimum number $MinPts$. The parameter

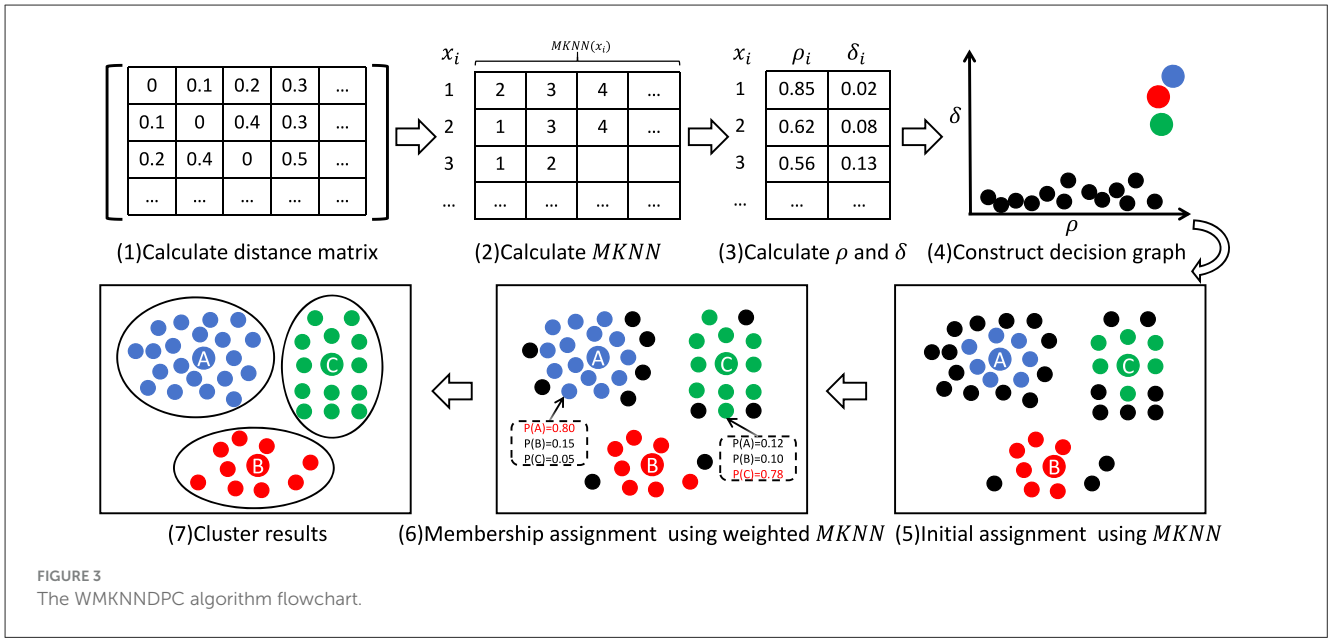

FIGURE 3
The WMKNNDPC algorithm flowchart.

Require: Dataset $D = \{x_1, \ldots, x_i, \ldots, x_n\}$, preset parameter K.
Ensure: Label of D.
1: The dataset D is normalized and the Euclidean distance between the data points is calculated;
2: Calculate the mutual K-nearest neighbors set of data point x_i using Equations 6–9;
3: Calculate the MKNN local density ρ_i of data point x_i by using Equation 10;
4: Calculate the relative distance δ_i of data point x_i by using Equations 3, 4;
5: The new local density and relative distance are used to construct the decision graph by using Equation 5 and select the most suitable clustering center (the number of clusters);
6: The unassigned data points are initially assigned using mutual K-nearest neighbors and breadth-first search by Algorithm 1;
7: The remaining unassigned points are assigned by the weighted mutual K-nearest neighbors membership by Algorithm 2;
8: Output a label for each point in dataset D.

Algorithm 3. WMKNNDPC.

of DPC is the truncation distance d_c, while the parameter for DPC-KNN is a percentage p controlling the number of K-nearest neighbors. Determining the optimal parameters for DPC and DPC-KNN is challenging. DPCSA does not need any parameters. WMKNNDPC, like FKNN-DPC and LF-DPC, only requires one parameter for K-nearest neighbors, making it easier to determine parameters compared to DPC. That is because the value of K is an integer, while the parameter d_c of DPC is a percentage, K is easier to obtain than d_c, especially for small datasets.

TABLE 1 The details of artificially synthetic 2D datasets.

Dataset	Records	Attributes	Clusters
Jain	373	2	2
Flame	240	2	2
Pathbased	300	2	3
Aggregation	788	2	7
Spiral	312	2	3
Compound	399	2	6
Smile	266	2	3
D31	3,100	2	31

4.3 Experiments on synthetic datasets

In this section, we will focus on exploring the clustering results of various algorithms (WMKNNDPC, K-means, DBSCAN, DPC, DPC-KNN, FKNN-DPC, DPCSA, and LF-DPC) on synthetic datasets. We will visualize the best clustering results obtained by each algorithm on eight datasets, as illustrated in Figures 4–11. All algorithms, except DBSCAN, will be represented by red diamonds for their cluster centers. Additionally, the performance of each algorithm on the synthetic datasets will be compared based on the ARI, AMI, and FMI clustering results in Table 3. The best evaluation indicators have been highlighted in bold.

In the given dataset, named Jain, there is an uneven density distribution, consisting of two semicircles. The cluster of data points in the upper semicircle is much sparser than the cluster in the lower semicircle. The clustering results of each algorithm are presented in Figure 4. It is evident that DPC, DPC-KNN, and DPCSA fail to find the correct cluster center primarily because they do not take into account the local distribution of data points when calculating local density. Though FKNN-DPC and LF-DPC

TABLE 2 The details of real datasets from UCI.

Dataset	Records	Attributes	Clusters
Seeds	210	7	3
Libras	360	91	15
WDBC	569	30	2
Parkinsons	195	23	2
Glass	214	9	6
Wine	178	13	3
SCADI	70	206	7
Ecoli	336	8	8
Dermatology	366	33	6
Banknote	1,372	4	2

can accurately locate cluster centers, some sparse data points in the upper half circle are incorrectly assigned to the lower half circle cluster due to the assignment strategy. K-means can approximately find the cluster center, but due to the assignment strategy considering only distance, some points have been assigned incorrectly. DBSCAN is the best-performing algorithm among the comparison algorithms, except for WMKNNDPC, which can correctly identify two clusters, but there are individual boundary points with category attribution errors. In contrast, our algorithm not only accurately finds cluster centers but also perfectly assigns remaining points. This is mainly due to the proposed new local density calculation method and remaining points assignment algorithm.

The Flame dataset consists of two evenly distributed clusters. According to the experimental results shown in Figure 5, both the original DPC and its derived algorithms (DPC-KNN, DPCSA, LF-DPC, and WMKNNDPC) can accurately identify two class centers and correctly assign the remaining points. However, FKNN-DPC correctly identifies cluster centers but has an assignment error in one data point on each side of the upper cluster. K-means can find suitable clustering centers but incorrectly attribute the data points in both clusters, resulting in the worst clustering performance. DBSCAN can correctly classify most of the data points, but it identifies 14 boundary points as noise.

The dataset Pathbased is a manifold dataset that consists of one circular cluster wrapping around two spherical clusters. Due to the proximity of data points on both sides of the spherical cluster to the circular cluster, assignment errors are easily caused. In Figure 6, both DPC and its DPC-derived algorithms can find cluster centers. The data points on both sides of the circular cluster in K-means, DPC, and DPC-KNN are incorrectly assigned to the spherical cluster, which is due to the assignment strategy that only considers distance. FKNN-DPC and DPCSA improved the assignment strategy, but there was an error in dividing the data points on the right side of the circular cluster. The clustering performance of DBSCAN is similar to that of DPCSA, but six spherical cluster boundary points are identified as noise. LF-DPC and WMKNNDPC can correctly partition the data points on both sides of a circular cluster, but there are still a few adhesive boundary point assignment errors, and our algorithm's performance is only inferior to LF-DPC.

Figure 7 illustrates the clustering performance of each algorithm on the Aggregation dataset. DBSCAN, DPC, DPC-KNN, FKNN-DPC, DPCSA, LF-DPC, and WMKNNDPC all demonstrate the ability to achieve good clustering results and identify seven clusters with different shapes. Our algorithm better than other comparative algorithms and attains the highest evaluation values for ARI, AMI, and FMI. K-means exhibits the poorest clustering performance on this dataset, which may be a common issue in partitioning clustering algorithms.

Spiral is composed of three spiral clusters, which are typical nonspherical clusters. The accompanying Figure 8 illustrates that, except K-means, all other algorithms such as DBSCAN, DPC, and DPC-derived algorithms are able to cluster perfectly, with only slight variations in cluster centers.

In Figure 9, the performance of WMKNNDPC and seven other comparison algorithms on the Compound dataset is demonstrated. The Compound dataset is a manifold dataset with an uneven density distribution and six arbitrary shapes. Detecting clusters in this dataset is a challenging task for many clustering algorithms. For example, K-means failed to identify two cluster centers on Compound, resulting in clustering errors for most points. DBSCAN only found four clusters and identified a large number of sparse boundary points as noise. DPC mistakenly identified two cluster centers from one cluster in the bottom left corner, mainly due to the local density calculation method. DPC-KNN could only find one cluster center in the bottom left of two clusters, while the sparse cluster on the right recognized two cluster centers, showcasing typical challenges faced by local density calculation methods. FKNN-DPC, DPCSA, and LF-DPC showed improved performance, but still faced challenges in finding cluster centers when dealing with sparse clusters and in identifying two cluster centers in the geese-shaped cluster in the upper right corner. One of the main reasons for these issues is the use of a fixed K-value, which does not adapt to the local distribution of data points. Although WMKNNDPC also faced challenges in identifying cluster centers in geese-shaped clusters, it managed to find the cluster centers of sparse clusters and correctly allocate sparse cluster data points. Additionally, the evaluation indicators of WMKNNDPC were significantly better than those of other algorithms.

The Smile dataset is surrounded by one semi-circular cluster and two square clusters, similar to the Pathbased dataset. In Figure 10, we can see the clustering results of each algorithm. Although K-means identifies three cluster centers, it incorrectly classifies the points of circular clusters into two square clusters. Similar issues are observed with DPC and DPC-KNN, as they also focus on assigning data points to the nearest neighbors in high-density areas. Our WMKNNDPC algorithm has the same performance as DBSCAN, FKNN-DPC, DPCSA, and LF-DPC, all of which can achieve perfect clustering.

The D31 dataset consists of 31 clusters containing a total of 3,100 data points. Several of these clusters are very close to each other, with some even partially overlapping. Results from the experiment shown in Figure 11 indicate that the K-means, DPC, and DPC-derived algorithms demonstrate strong clustering performance and are able to identify boundary points of different

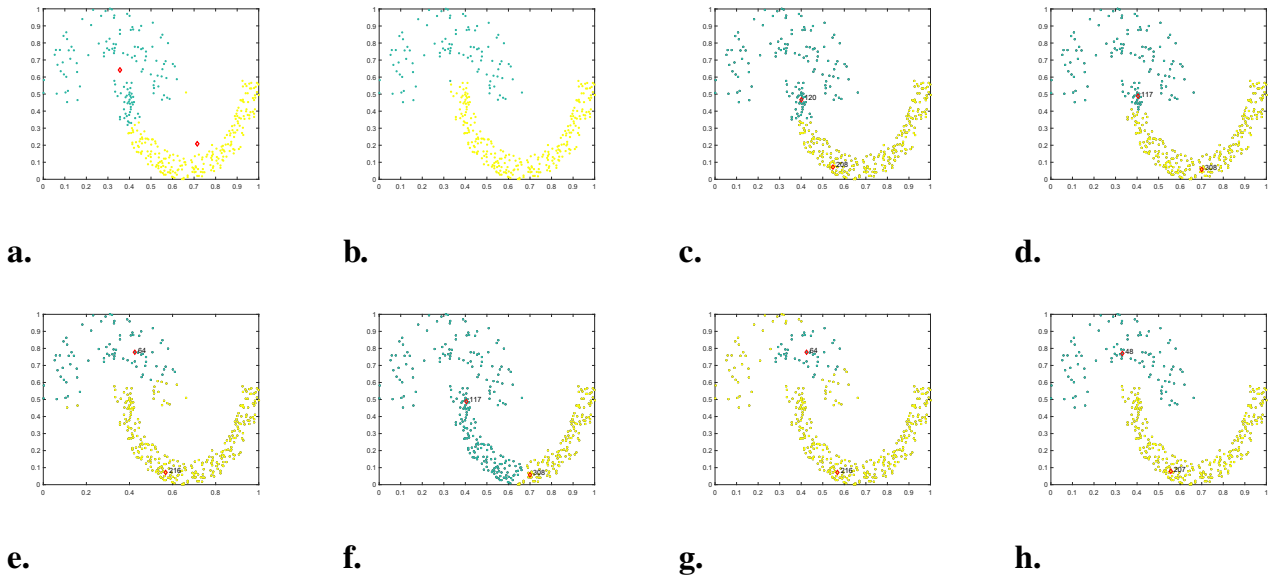

FIGURE 4
The clustering results of each algorithm on the Jain dataset. **(a)** K-means. **(b)** DBSCAN. **(c)** DPC. **(d)** DPC-KNN. **(e)** FKNN-DPC. **(f)** DPCSA. **(g)** LF-DPC. **(h)** Our.

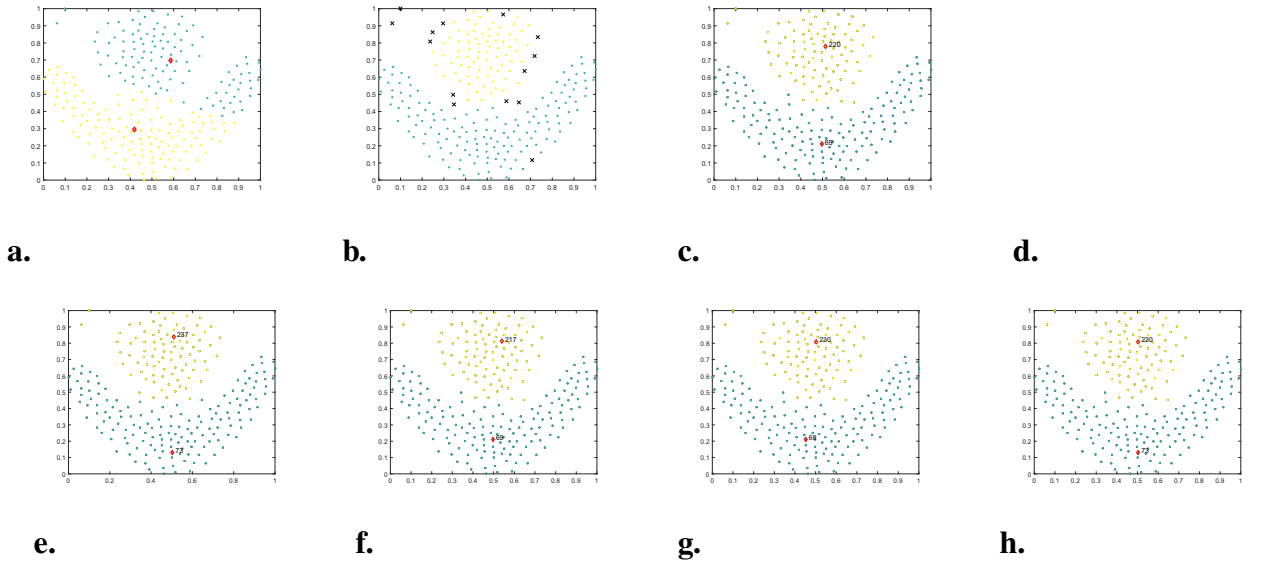

FIGURE 5
The clustering results of each algorithm on the Flame dataset. **(a)** K-means. **(b)** DBSCAN. **(c)** DPC. **(d)** DPC-KNN. **(e)** FKNN-DPC. **(f)** DPCSA. **(g)** LF-DPC. **(h)** Our.

categories to a certain extent. According to various evaluation indicators, FKNN-DPC exhibits the best clustering performance, with our algorithm being only slightly inferior to FKNN-DPC and LF-DPC, but better than the other five algorithms.

In summary, the algorithm presented in this paper demonstrated excellent clustering performance on six artificial synthesis datasets. Our WMKNNDPC algorithm performed second best on the Pathbased dataset, trailing only LF-DPC. However, it still achieved impressive results, with an ARI of 0.9299, AMI of 0.9004, and FMI of 0.9532. The results on the D31 dataset were similar to those on the Pathbased dataset. Therefore, we have confidence in the proposed algorithm, which is based on mutual K-nearest neighbors local density and remaining points assignment, and its strong clustering performance.

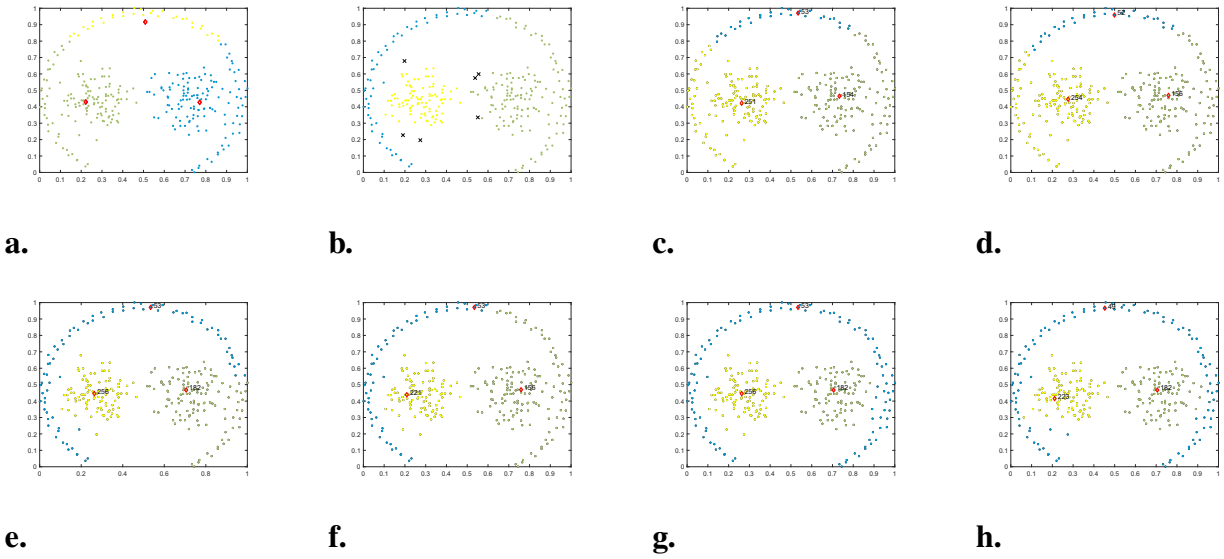

FIGURE 6
The clustering results of each algorithm on the Pathbased dataset. **(a)** K-means. **(b)** DBSCAN. **(c)** DPC. **(d)** DPC-KNN. **(e)** FKNN-DPC. **(f)** DPCSA. **(g)** LF-DPC. **(h)** Our.

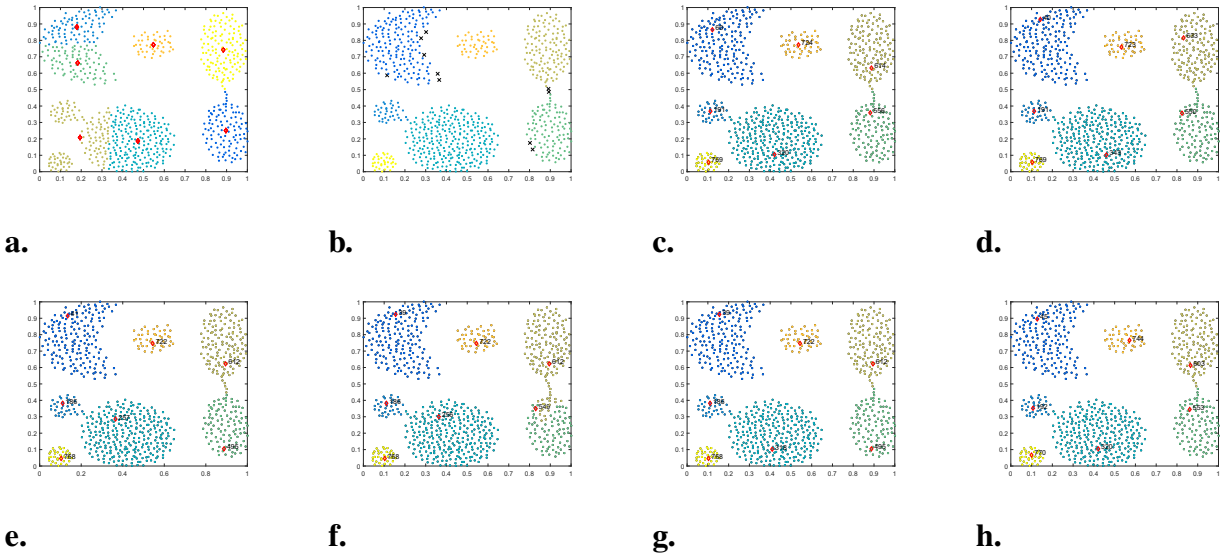

FIGURE 7
The clustering results of each algorithm on the Aggregation dataset. **(a)** K-means. **(b)** DBSCAN. **(c)** DPC. **(d)** DPC-KNN. **(e)** FKNN-DPC. **(f)** DPCSA. **(g)** LF-DPC. **(h)** Our.

4.4 Experiments on real datasets

In this section, we will evaluate the clustering performance of WMKNNDPC using real datasets from various research fields. These datasets vary in size, dimensions, and cluster numbers, allowing for a comprehensive assessment of the proposed algorithm's adaptability. Table 4 displays the clustering results of eight different algorithms applied to ten real datasets.

Based on the data in Table 4, it is evident that the proposed WMKNNDPC algorithm is better than the other seven algorithms in ARI, AMI, and FMI evaluation indicators across the Libras, WDBC, Parkinsons, SCADI, Ecoli, and Dermatology datasets. Additionally, our algorithm shows superior performance in the ARI and AMI indicators compared to other algorithms in the Glass dataset.

In the Seeds dataset, the WMKNNDPC algorithm ranks second among all algorithms and is very close to FKNN-DPC in the

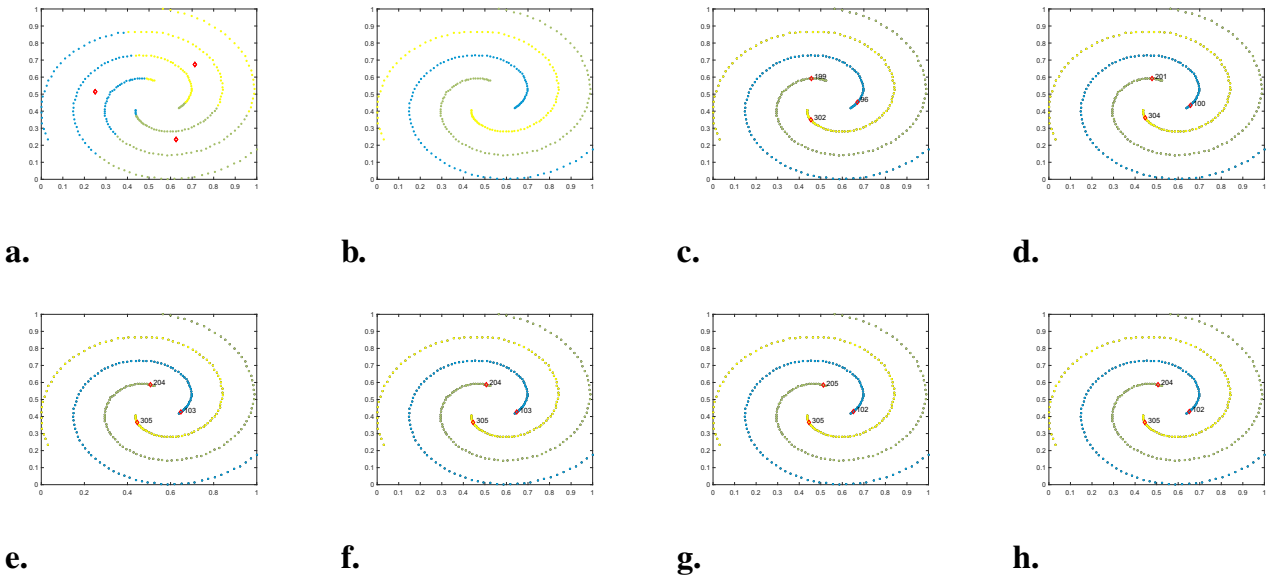

FIGURE 8
The clustering results of each algorithm on the Spiral dataset. **(a)** K-means. **(b)** DBSCAN. **(c)** DPC. **(d)** DPC-KNN. **(e)** FKNN-DPC. **(f)** DPCSA. **(g)** LF-DPC. **(h)** Our.

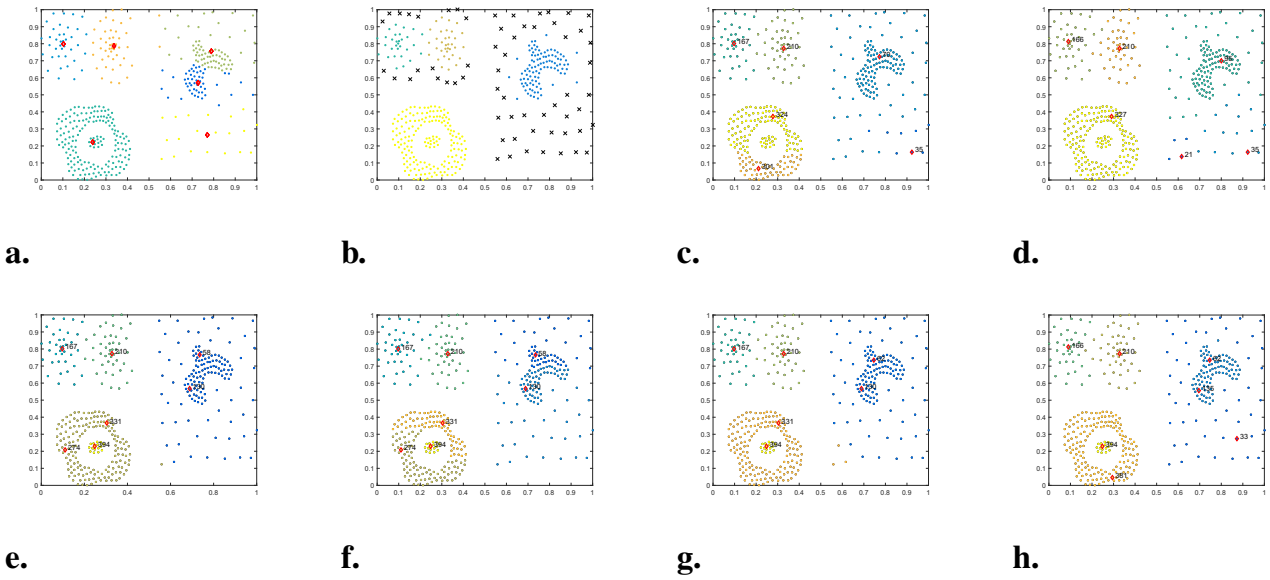

FIGURE 9
The clustering results of each algorithm on the Compound dataset. **(a)** K-means. **(b)** DBSCAN. **(c)** DPC. **(d)** DPC-KNN. **(e)** FKNN-DPC. **(f)** DPCSA. **(g)** LF-DPC. **(h)** Our.

evaluation indicators. FKNN-DPC's ARI value is 0.8024, while our ARI value is 0.7913, indicating that they are very comparable. Our algorithm performs far superior to the other six comparative algorithms. This is because FKNN-DPC obtains the best clustering result after parameter tuning, and our algorithm selects a K-value that is slightly larger to ensure that each point has mutual K-nearest neighbors with each other.

In comparing the Wine dataset, we observed that LF-DPC achieved the best clustering performance based on various indicators through local fair density, closely followed by FKNN-DPC and our algorithm.

In the Banknote dataset, DPCSA without any parameters achieved the best clustering performance, outperforming other algorithms. Our algorithm ranked second in all comparison results,

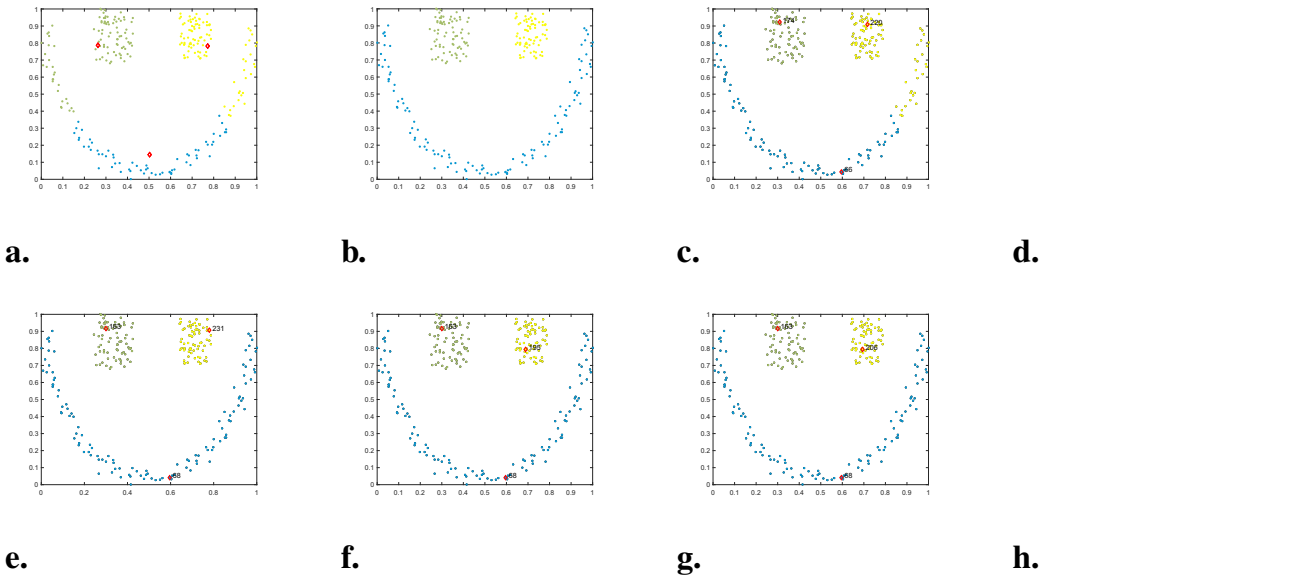

FIGURE 10
The clustering results of each algorithm on the Smile dataset. **(a)** K-means. **(b)** DBSCAN. **(c)** DPC. **(d)** DPC-KNN. **(e)** FKNN-DPC. **(f)** DPCSA. **(g)** LF-DPC. **(h)** Our.

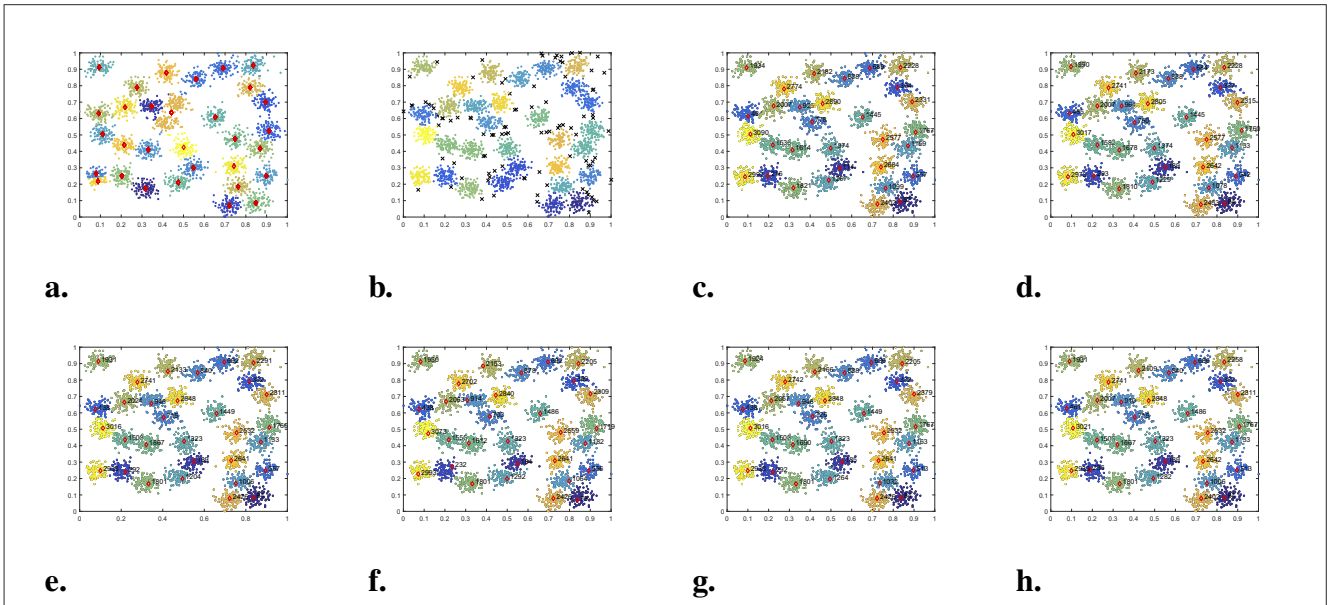

FIGURE 11
The clustering results of each algorithm on the D31 dataset. **(a)** K-means. **(b)** DBSCAN. **(c)** DPC. **(d)** DPC-KNN. **(e)** FKNN-DPC. **(f)** DPCSA. **(g)** LF-DPC. **(h)** Our.

showing that DPCSA's remaining points assignment algorithm is well-suited for this dataset.

According to Table 4, the WMKNNDPC algorithm performs better in clustering than other algorithms in most cases. It achieved the highest ARI and AMI indicators on the 7/10 dataset and the highest FMI on the 6/10 dataset. Even though the comparison algorithms were optimized, they did not perform as well as our WMKNNDPC algorithm. This is mainly due to the unique method for calculating the density of mutual K-nearest neighbors and the remaining points assignment algorithm based on weighted mutual K-nearest neighbors. Therefore, we can conclude that the WMKNNDPC algorithm is not only effective in discovering cluster centers but also in correctly assigning remaining points. It also demonstrates good adaptability on both manifold datasets and datasets with uneven density distribution.

TABLE 3 Cluster results on synthetic datasets.

Algorithm	ARI	AMI	FMI	Parm	ARI	AMI	FMI	Parm
	Jain				Flame			
K-means	0.5767	0.4916	0.8200	2	0.5116	0.4442	0.7646	2
DBSCAN	0.9887	0.9691	0.9956	0.05/9	0.9081	0.7570	0.9561	0.065/4
DPC	0.6183	0.5396	0.8386	1%	**1**	**1**	**1**	5%
DPC-KNN	0.7146	0.6183	0.8819	2%	**1**	**1**	**1**	0.50%
FKNN-DPC	0.8224	0.7092	0.9359	43	0.9666	0.9267	0.9845	5
DPCSA	0.0442	0.2167	0.5924	–	**1**	**1**	**1**	–
LF-DPC	0.4059	0.2936	0.8270	40	**1**	**1**	**1**	2
WMKNNDPC	**1**	**1**	**1**	30	**1**	**1**	**1**	3
	Pathbased				Aggregation			
K-means	0.4613	0.5098	0.6617	3	0.7136	0.8048	0.7742	7
DBSCAN	0.5890	0.6884	0.7317	0.065/4	0.9779	0.9529	0.9827	0.04/6
DPC	0.4530	0.4997	0.6585	2%	0.9956	0.9922	0.9966	2%
DPC-KNN	0.4602	0.5080	0.6617	2%	0.9935	0.9892	0.9949	0.50%
FKNN-DPC	0.7323	0.7744	0.8226	8	0.9949	0.9907	0.9960	8
DPCSA	0.6133	0.7073	0.7511	–	0.9581	0.9537	0.9673	–
LF-DPC	**0.9699**	**0.9525**	**0.9799**	8	0.9949	0.9905	0.9960	7
WMKNNDPC	0.9299	0.9004	0.9532	10	**0.9978**	**0.9955**	**0.9983**	23
	Spiral				Compound			
K-means	-0.0057	-0.0052	0.3277	3	0.7687	0.7853	0.8276	6
DBSCAN	**1**	**1**	**1**	0.04/2	0.8402	0.7839	0.8850	0.08/14
DPC	**1**	**1**	**1**	2%	0.5989	0.7798	0.6963	2%
DPC-KNN	**1**	**1**	**1**	4%	0.8087	0.7913	0.8661	0.50%
FKNN-DPC	**1**	**1**	**1**	6	0.8426	0.8337	0.8898	8
DPCSA	**1**	**1**	**1**	–	0.5738	0.7117	0.6714	–
LF-DPC	**1**	**1**	**1**	5	0.8409	0.8231	0.8891	10
WMKNNDPC	**1**	**1**	**1**	5	**0.9867**	**0.9703**	**0.9900**	11
	Smile				D31			
K-means	0.4875	0.5828	0.6650	3	0.9125	0.9501	0.9156	31
DBSCAN	**1**	**1**	**1**	0.08/5	0.8078	0.8895	0.8186	0.04/40
DPC	0.7210	0.7799	0.8166	4%	0.9332	0.9539	0.9354	2%
DPC-KNN	0.7179	0.7794	0.8148	6%	0.9357	0.9549	0.9378	2%
FKNN-DPC	**1**	**1**	**1**	6	**0.9516**	**0.9653**	**0.9531**	23
DPCSA	**1**	**1**	**1**	–	0.9353	0.9552	0.9374	–
LF-DPC	**1**	**1**	**1**	6	0.9473	0.9620	0.9490	20
WMKNNDPC	**1**	**1**	**1**	10	0.9362	0.9553	0.9382	30

Bold values indicate that the corresponding algorithm achieved optimal performance on specific evaluation metric (ARI, AMI, FMI) of the synthetic dataset.

4.5 Experiments on Olivetti face dataset

To further evaluate the performance of WMKNNDPC, we conducted experiments on the Olivetti face dataset to detect density peaks, complete clustering and compare with DPC, FKNN-DPC, and DPCSA. The Olivetti face dataset [48, 49] is a widely used test dataset in density peak clustering and machine learning. It consists of 40 types of faces, with each type having 10 different images.

To reduce experimental costs and computational load, we randomly selected 100 pictures of 10 types of faces for the experiment. In Figure 12, it is observed that DPC and DPCSA identified 11 and 12 density peaks, respectively, but did not

10.3389/fams.2025.1598165

TABLE 4 Cluster results on real datasets.

Algorithm	ARI	AMI	FMI	Parm	ARI	AMI	FMI	Parm
	Seeds				Libras			
K-means	0.7049	0.6705	0.8026	3	0.3032	0.5230	0.3528	15
DBSCAN	0.5291	0.5302	0.6711	0.24/16	0.2348	0.3598	0.2799	0.9/1
DPC	0.7341	0.7172	0.8231	2%	0.2984	0.5138	0.3682	0.40%
DPC-KNN	0.7448	0.7144	0.8297	1%	0.3051	0.5471	0.3666	1%
FKNN-DPC	**0.8024**	**0.7684**	**0.8680**	4	0.3211	0.5367	0.3943	10
DPCSA	0.6873	0.6609	0.7918	–	0.2683	0.4939	0.3572	–
LF-DPC	0.7777	0.7381	0.8516	8	0.3437	0.5406	0.3996	5
WMKNNDPC	0.7913	0.7607	0.8605	8	**0.4137**	**0.6016**	**0.4754**	11
	WDBC				Parkinsons			
K-means	0.7302	0.6110	0.8770	2	0.0520	0.2129	0.5957	2
DBSCAN	0.4786	0.3581	0.7570	0.46/38	0.0252	0.0071	0.5775	0.5/17
DPC	0.4705	0.4146	0.7860	0.40%	0.2686	0.1772	0.8140	0.20%
DPC-KNN	0.4552	0.4017	0.7813	1%	0.2686	0.1772	0.8140	2%
FKNN-DPC	0.4452	0.3932	0.7783	6	0.2686	0.1772	0.8140	5
DPCSA	0.3771	0.3361	0.7595	-	0.2686	0.1772	0.8140	–
LF-DPC	0.4756	0.4189	0.7875	4	0.2686	0.1772	0.8140	6
WMKNNDPC	**0.7613**	**0.6423**	**0.8894**	2	**0.3632**	**0.2151**	**0.8190**	5
	Glass				Wine			
K-means	0.1363	0.2113	0.3766	6	0.8471	0.8301	0.8984	3
DBSCAN	0.0697	0.167	0.3197	0.1/2	0.5292	0.5484	0.7121	0.5/21
DPC	0.1337	0.1758	0.5379	2%	0.6724	0.7065	0.7835	2%
DPC-KNN	0.1809	0.1652	0.5395	2%	0.6990	0.7228	0.8006	8%
FKNN-DPC	0.2209	0.2385	0.5165	7	0.8819	0.8566	0.9215	8
DPCSA	0.1818	0.1646	**0.5402**	-	0.7414	0.7480	0.8283	-
LF-DPC	0.1806	0.1669	0.5311	7	**0.9150**	**0.8800**	**0.9436**	7
WMKNNDPC	**0.2316**	**0.2836**	0.5317	10	0.8685	0.8473	0.9126	8
	SCADI				Ecoli			
K-means	0.4583	0.4910	0.5805	7	0.4957	0.5192	0.6174	8
DBSCAN	–	–	–	–	0.0947	0.0696	0.5203	0.2/6
DPC	0.5618	0.4966	0.6684	2%	0.7054	0.5816	0.7983	1%
DPC-KNN	0.5627	0.4759	0.6690	2%	0.6913	0.5817	0.7939	5%
FKNN-DPC	0.6191	0.5319	0.7122	6	0.5914	0.5596	0.7071	7
DPCSA	0.5939	0.4988	0.6932	-	0.4883	0.4229	0.6787	-
LF-DPC	0.6953	0.5872	0.7736	6	0.7060	0.5877	0.8014	6
WMKNNDPC	**0.7621**	**0.6419**	**0.8216**	4	**0.7185**	**0.6067**	**0.8037**	6
	Dermatology				Banknote			
K-means	0.7312	0.8741	0.7856	6	0.0223	0.0168	0.5139	2
DBSCAN	0.4161	0.5176	0.5293	0.98/2	0.8260	0.7542	0.9099	0.1/5
DPC	0.6622	0.7167	0.7487	0.40%	0.8008	0.7751	0.8968	1%
DPC-KNN	0.6349	0.7731	0.7089	1%	0.3955	0.3575	0.6524	0.8%

(Continued)

TABLE 4 (Continued)

Algorithm	ARI	AMI	FMI	Parm	ARI	AMI	FMI	Parm
FKNN-DPC	0.8654	0.8741	0.8994	6	0.7702	0.7576	0.8793	20
DPCSA	0.6062	0.7451	0.6896	–	**0.9653**	**0.9359**	**0.9828**	–
LF-DPC	0.8288	0.8345	0.8704	8	0.7702	0.7576	0.8793	10
WMKNNDPC	**0.8703**	**0.8747**	**0.9035**	9	0.9321	0.8717	0.9664	15

Bold values indicate that the corresponding algorithm achieved optimal performance on specific evaluation metric (ARI, AMI, FMI) of the real dataset.

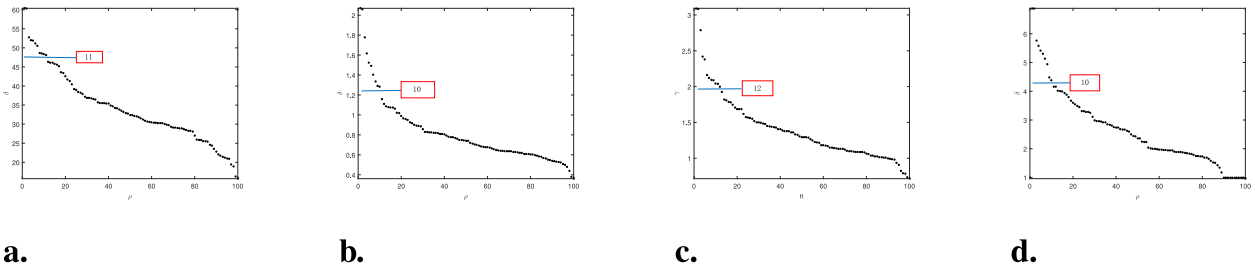

FIGURE 12
Select clustering centers in the Olivetti face dataset. **(a)** DPC (p = 3%). **(b)** KNN-DPC (k = 3). **(c)** DPCSA. **(d)** WMKNNDPC (k = 4).

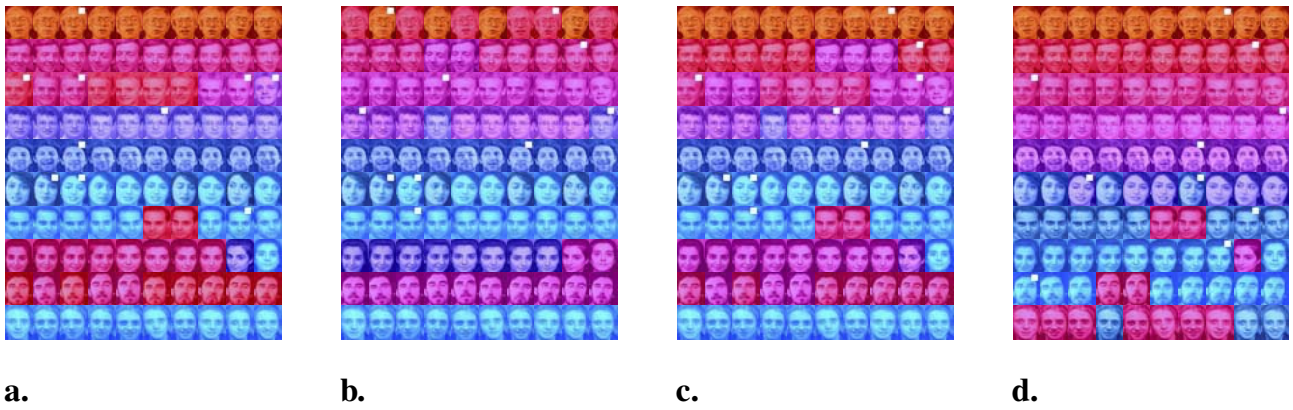

FIGURE 13
Graphical display of clustering results on Olivetti face dataset. **(a)** DPC (p = 3%). **(b)** KNN-DPC (k = 3). **(c)** DPCSA. **(d)** WMKNNDPC (k = 4).

accurately find the ideal 10 density peaks. Our WMKNNDPC performed similarly to FKNN-DPC in density peak detection, efficiently identifying 10 density peaks. Therefore, WMKNNDPC is better than DPC and DPCSA in locating cluster centers in the Olivetti face dataset.

We evaluated the clustering performance of WMKNNDPC on 10 different types of faces. We chose 10 density peaks of DPC, FKNN-DPC, DPCSA and WMKNNDPC as the clustering centers. The final clustering results can be seen in Figure 13, where white box dots indicate the clustering centers. It is evident that DPC is only able to accurately identify six cluster centers. However, in groups 3 and 6, multiple cluster centers appear due to its local density calculation method. FKNN-DPC and DPCSA can identify seven cluster centers, but there may still be multiple peaks problem. This is mainly because they use a fixed K-value for local density without considering the local distribution of the samples. However, our algorithm can efficiently identify 9 cluster centers. As shown in the clustering indicators results in Figure 14, WMKNNDPC is far superior to the other three algorithms in ARI, AMI, and FMI evaluation indicators. This further confirms the good performance of the proposed algorithm in cluster center recognition and remaining points assignment, mainly due to the adaptive ability of weighted mutual K-nearest neighbors.

4.6 Analyze different parameters of WMKNNDPC

This section will discuss the impact of the unique parameter K of WMKNNDPC on clustering performance. Four typical mainfold datasets and datasets with uneven density distribution (Jain,

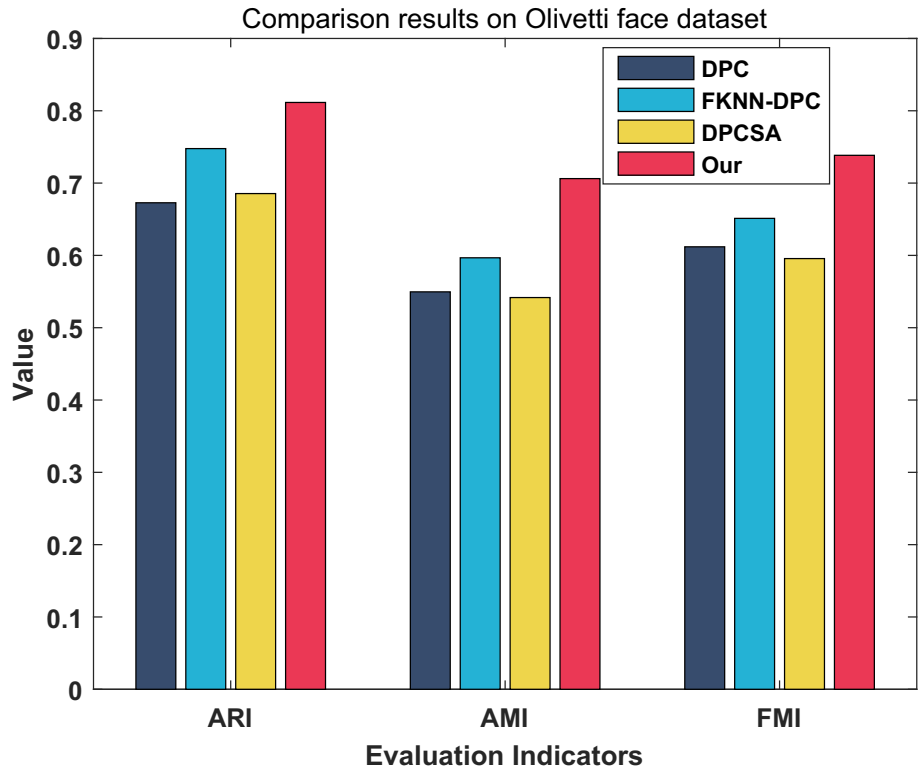

FIGURE 14
Comparison results of evaluation indicators on Olivetti face dataset.

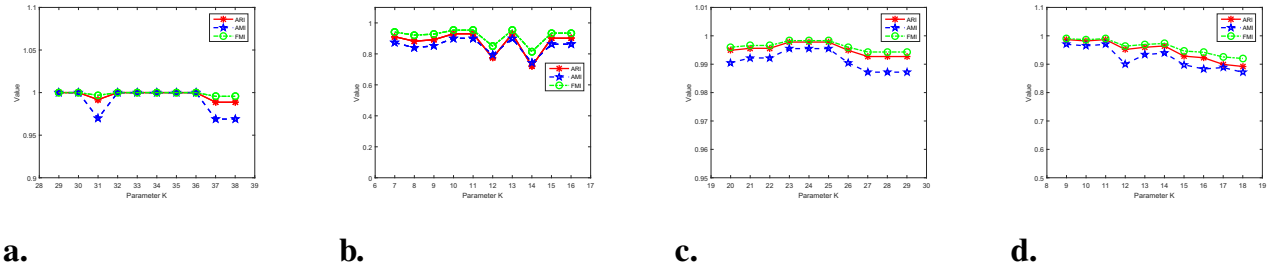

FIGURE 15
Results on four synthetic datasets with different parameter. **(a)** Jain dataset. **(b)** Pathbased dataset. **(c)** Aggregation dataset. **(d)** Compound dataset.

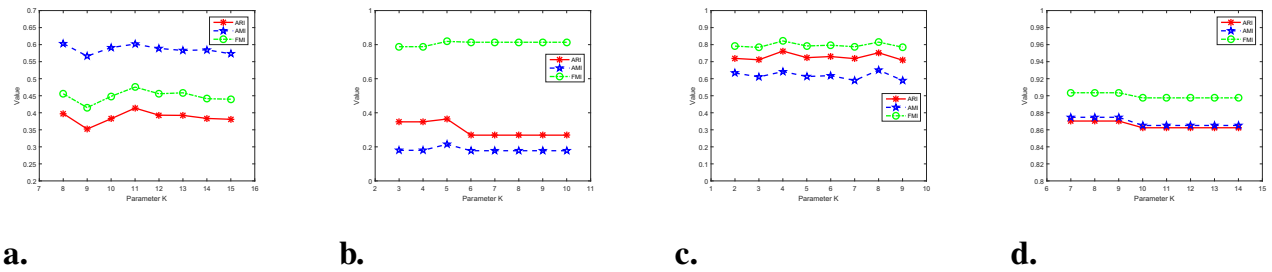

FIGURE 16
Results on four real datasets with different parameter. **(a)** Libras dataset. **(b)** Parkinsons dataset. **(c)** SCADI dataset. **(d)** Dermatology dataset.

TABLE 5 Comparison results of running time.

Dataset	Records	FKNN-DPC	DPCSA	LF-DPC	WMKNNDPC
Jain	373	0.1954	0.1287	0.2077	0.3061
Flame	240	0.1975	0.1204	0.2719	0.3105
Pathbased	300	0.2094	0.1235	0.2202	0.2523
Aggregation	788	0.6702	0.1668	0.7621	0.8369
Spiral	312	0.2169	0.1297	0.2243	0.2722
Compound	399	0.2016	0.1438	0.2077	0.3072
Smile	266	0.2940	0.1312	0.2844	0.3614
D31	3100	5.8327	1.0205	6.2774	7.8623
Seeds	210	0.2255	0.1255	0.1759	0.2423
Libras	360	0.2108	0.1445	0.3529	0.3716
WDBC	569	0.4724	0.1510	0.8468	0.4890
Parkinsons	195	0.2303	0.1182	0.2123	0.2893
Glass	214	0.1786	0.1291	0.1939	0.2504
Wine	178	0.1603	0.1247	0.1734	0.2323
SCADI	70	0.1408	0.1252	0.1511	0.1488
Ecoli	336	0.2078	0.1388	0.2747	0.3945
Dermatology	366	0.2349	0.1320	0.2465	0.3177
Banknote	1372	4.1260	0.3005	4.6558	5.7752
Olivetti face	400	0.2326	0.1864	0.2456	0.3056

Pathbased, Aggregation and Compound), and four real datasets from UCI (Libras, Parkinsons, SCADI, and Dermatology) were used for experimental analysis.

Figure 15 displays the experimental results of parameters on the synthetic datasets. Differently colored lines represent various clustering indicators. The parameter K-value of the Jain dataset in Figure 15a ranges from 29 to 38. It is evident that WMKNNDPC performs clustering exceptionally well, except when K-values are 31, 37, and 38. Even in the worst case scenario, when $K = 37$, the Adjusted Rand Index is 0.9887. From Figure 15b, it is evident that the overall performance of WMKNNDPC is generally stable, except for K-values of 12 and 15. In the Aggregation parameter experiment, it was observed that WMKNNDPC is not significantly affected by the parameter K, and it exhibits a very good clustering effect. Upon examining Figure 15d, it is evident that the performance of WMKNNDPC experiences a slight decrease with an increase in the K-value, but overall performance remains relatively stable. However, in some extreme cases, such as in the Jain dataset, when the K-value is relatively small ($K = 6$), our algorithm obtains an ARI value of 0.5222. This is mainly because the K-value is small and there are fewer mutual K-nearest neighbors, making it difficult to capture the local distribution of data points. In the Aggregation dataset (with a size of 788), when the K-value is relatively large ($K = 50, 60, 70$), our algorithm obtains ARI values of 0.9095, 0.9072, and 0.9095, respectively. We can observe that when the K-value is relatively large, the clustering performance of WMKNNDPC on the Aggregation dataset remains relatively stable.

In Figure 16, the experimental results of the parameters on real datasets are presented. It was found that the optimal parameter value for the Libras dataset is $K = 11$. There are slight fluctuations in the performance of WMKNNDPC on both sides of the optimal parameter. In Figure 16b, our algorithm achieved the best clustering performance when $K = 5$. The clustering performance of WMKNNDPC is very stable when $K > 5$. Similarly, in the testing of SCADI and Dermatology datasets, it is observed that WMKNNDPC is basically not affected by parameter and has good clustering performance.

According to parameter testing, the parameter of the WMKNNDPC algorithm is within a reasonable range, and its clustering performance is basically not affected by synthetic and real datasets. The main reason is that mutual K-nearest neighbors have adaptability and can effectively discover the true cluster centers. Moreover, our proposed remaining points assignment method based on weighted mutual K-nearest neighbors can effectively improve clustering accuracy.

4.7 Running time analysis

This section mainly analyzes the running time of the proposed algorithm and the comparative algorithm. DPC is a basic algorithm with significantly lower running time; DPC-KNN has recently improved local density and its running time is relatively low; The clustering principles of K-means and DBSCAN are inconsistent

with the density peak clustering principle and lack comparability. Therefore, the reason for choosing FKNN-DPC, DPCSA, and LF-DPC as comparison algorithms is that these three algorithms simultaneously improve local density and optimize the remaining point allocation mechanism, which has comparative analysis value.

The running time of WMKNNDPC algorithm and comparative algorithms (FKNN-DPC, DPCSA, and LF-DPC) is shown in Table 5. The running time is the average of three runs of each algorithm, rounded to four decimal places, in seconds. We can see that DPCSA has the lowest running time because the algorithm uses a fixed K-value to calculate local density and allocate remaining points, without preset parameters. The running time of FKNN-DPC and LF-DPC is at the same level because the clustering principles of these two algorithms are very similar. Although WMKNNDPC has higher running time on most datasets compared to other algorithms, this is because our algorithm requires calculating weighted mutual K-nearest neighbors, which increases the time overhead in calculating local density and allocating remaining points. However, the clustering results are relatively good.

5 Conclusion

This paper introduces a novel density peak clustering algorithm called WMKNNDPC, which is based on weighted mutual K-nearest neighbors. It includes a local density calculation method for mutual K-nearest neighbors to address DPC's difficulty in finding cluster centers in unevenly distributed clusters. Additionally, a remaining points assignment method based on weighted mutual K-nearest neighbors is designed, which is more adaptive than FKNN-DPC and LF-DPC. The initial assignment is carried out using mutual K-nearest neighbors and breadth-first search, and the remaining points are further assigned using the membership assignment algorithm of weighted mutual K-nearest neighbors. Extensive experimental testing shows that WMKNNDPC performs better than the original DPC and DPC-derived algorithms on most datasets, and its clustering results also surpass those of classical K-means and DBSCAN. However, it's worth noting that, while the algorithm has certain advantages, the selection of cluster centers still requires manual intervention. Future research will focus on automating the selection of cluster centers, especially in clusters with uneven distribution.

Data availability statement

The datasets presented in this study can be found in online repositories. The names of the repository/repositories and accession number(s) can be found at: http://archive.ics.uci.edu/. The relevant data and source code of this article can be found in the Supplementary material folder.

Author contributions

CR: Writing – review & editing, Writing – original draft. CL: Writing – review & editing, Supervision. YY: Formal analysis, Supervision, Writing – review & editing. WY: Supervision, Writing – review & editing. RG: Writing – review & editing, Methodology.

Funding

The author(s) declare that financial support was received for the research and/or publication of this article. This work was supported by the High-Level Departure Project of Yibin University (Grant No. 2023QH02) and Science and Technology Project of Sichuan Province (Grant Nos. 2024ZYD0089 and 2024YFHZ0022).

Acknowledgments

The authors express their gratitude to the researchers who provided the source codes of the comparative algorithms and the experimental data for this paper.

Conflict of interest

The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

Generative AI statement

The author(s) declare that no Gen AI was used in the creation of this manuscript.

Publisher's note

Supplementary material

The Supplementary Material for this article can be found online at: https://www.frontiersin.org/articles/10.3389/fams.2025.1598165/full#supplementary-material

References

1. Frey BJ, Dueck D. Clustering by passing messages between data points. *Science*. (2007) 315:972–6. doi: 10.1126/science.1136800

2. Borlea ID, Precup RE, Borlea AB, Iercan D. A unified form of fuzzy C-means and K-means algorithms and its partitional implementation. *Knowl-Based Syst*. (2021) 214:106731. doi: 10.1016/j.knosys.2020.106731

3. Murtagh F, Contreras P. Algorithms for hierarchical clustering: an overview, II. *Wiley Interdiscip Rev: Data Min Knowl Discov*. (2017) 7:e1219. doi: 10.1002/widm.1219

4. Bai L, Cheng X, Liang J, Shen H, Guo Y. Fast density clustering strategies based on the k-means algorithm. *Pattern Recognit*. (2017) 71:375–86. doi: 10.1016/j.patcog.2017.06.023

5. Zhao J, Tang J, Fan T, Li C, Xu L. Density peaks clustering based on circular partition and grid similarity. *Concurr Comput Pract Exp*. (2019) 32:e5567. doi: 10.1002/cpe.5567

6. Yang MS, Chang-Chien SJ, Nataliani Y. Unsupervised fuzzy model-based Gaussian clustering. *Inf Sci*. (2019) 481:1–23. doi: 10.1016/j.ins.2018.12.059

7. Yin H, Benson AR, Leskovec J, Gleich DF. Local higher-order graph clustering. In: *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. New York, NY: ACM (2017). p. 555–64. doi: 10.1145/3097983.3098069

8. Li Y, Chu X, Tian D, Feng J, Mu W. Customer segmentation using K-means clustering and the adaptive particle swarm optimization algorithm. *Appl Soft Comput*. (2021) 113:107924. doi: 10.1016/j.asoc.2021.107924

9. Huang L, Ruan S, Denœux T. Application of belief functions to medical image segmentation: a review. *Inf Fusion*. (2023) 91:737–56. doi: 10.1016/j.inffus.2022.11.008

10. Lei T, Liu P, Jia X, Zhang X, Meng H, Nandi AK. Automatic fuzzy clustering framework for image segmentation. *IEEE Trans Fuzzy Syst*. (2019) 28:2078–92. doi: 10.1109/TFUZZ.2019.2930030

11. Tu B, Zhang X, Kang X, Wang J, Benediktsson JA. Spatial density peak clustering for hyperspectral image classification with noisy labels. *IEEE Trans Geosci Remote Sens*. (2019) 57:5085–97. doi: 10.1109/TGRS.2019.2896471

12. Kolhe L, Jetawat AK, Khairnar V. Robust product recommendation system using modified grey wolf optimizer and quantum inspired possibilistic fuzzy C-means. *Cluster Comput*. (2021) 24:953–68. doi: 10.1007/s10586-020-03171-6

13. Cai Q, Gong M, Ma L, Ruan S, Yuan F, Jiao L. Greedy discrete particle swarm optimization for large-scale social network clustering. *Inf Sci*. (2015) 316:503–16. doi: 10.1016/j.ins.2014.09.041

14. Qiu T, Li YJ. Fast LDP-MST: an efficient density-peak-based clustering method for large-size datasets. *IEEE Trans Knowl Data Eng*. (2023) 35:4767–80. doi: 10.1109/TKDE.2022.3150403

15. Lv Z, Di L, Chen C, Zhang B, Li N. A fast density peak clustering method for power data security detection based on local outlier factors. *Processes*. (2023) 11:2036. doi: 10.3390/pr11072036

16. Nie F, Li Z, Wang R, Li X. An effective and efficient algorithm for K-means clustering with new formulation. *IEEE Trans Knowl Data Eng*. (2022) 35:3433–43. doi: 10.1109/TKDE.2022.3155450

17. Ester M, Kriegel HP, Sander J, Xu X. A density-based algorithm for discovering clusters in large spatial databases with noise. In: *Proceedings of the Second International Conference on Knowledge Discovery and Data Mining (KDD'96)*. Portland, OR: AAAI Press (1996). p. 226–231.

18. Schubert E, Sander J, Ester M, Kriegel HP, Xu X. DBSCAN revisited, revisited: why and how you should (still) use DBSCAN. *ACM Trans Database Syst*. (2017) 42:1–21. doi: 10.1145/3068335

19. Shi D, Wang J, Cheng D, Gao J. A global-local affinity matrix model via EigenGap for graph-based subspace clustering. *Pattern Recognit Lett*. (2017) 89:67–72. doi: 10.1016/j.patrec.2016.12.023

20. Asheri H, Hosseini R, Araabi BN. A new EM algorithm for flexibly tied GMMs with large number of components. *Pattern Recognit*. (2021) 114:107836. doi: 10.1016/j.patcog.2021.107836

21. Rodriguez A, Laio A. Clustering by fast search and find of density peaks. *Science*. (2014) 344:1492–6. doi: 10.1126/science.1242072

22. Peterson LE. K-nearest neighbor. *Scholarpedia*. (2009) 4:1883. doi: 10.4249/scholarpedia.1883

23. Xie J, Gao H, Xie W, Liu X, Grant PW. Robust clustering by detecting density peaks and assigning points based on fuzzy weighted K-nearest neighbors. *Inf Sci*. (2016) 354:19–40. doi: 10.1016/j.ins.2016.03.011

24. Liu R, Wang H, Yu X. Shared-nearest-neighbor-based clustering by fast search and find of density peaks. *Inf Sci*. (2018) 450:200–26. doi: 10.1016/j.ins.2018.03.031

25. Ren C, Sun L, Yu Y, Wu Q. Effective density peaks clustering algorithm based on the layered k-nearest neighbors and subcluster merging. *IEEE Access*. (2020) 8:123449–68. doi: 10.1109/ACCESS.2020.3006069

26. Wu C, Lee J, Isokawa T, Yao J, Xia Y. Efficient clustering method based on density peaks with symmetric neighborhood relationship. *IEEE Access*. (2019) 7:60684–96. doi: 10.1109/ACCESS.2019.2912332

27. Du M, Ding S, Jia H. Study on density peaks clustering based on k-nearest neighbors and principal component analysis. *Knowl-Based Syst*. (2016) 99:135–45. doi: 10.1016/j.knosys.2016.02.001

28. Cheng D, Zhu Q, Huang J, Yang L. Natural neighbor-based clustering algorithm with density peeks. In: *2016 International Joint Conference on Neural Networks (IJCNN)*. Vancouver, BC: IEEE (2016). p. 92–8. doi: 10.1109/IJCNN.2016.7727185

29. Yaohui L, Zhengming M, Fang Y. Adaptive density peak clustering based on K-nearest neighbors with aggregating strategy. *Knowl-Based Syst*. (2017) 133:208–20. doi: 10.1016/j.knosys.2017.07.010

30. Li Z, Tang Y. Comparative density peaks clustering. *Expert Syst Appl*. (2018) 95:236–47. doi: 10.1016/j.eswa.2017.11.020

31. Xu X, Ding S, Xu H, Liao H, Xue Y. A feasible density peaks clustering algorithm with a merging strategy. *Soft Comput*. (2019) 23:5171–83. doi: 10.1007/s00500-018-3183-0

32. Parmar M, Wang D, Zhang X, Tan AH, Miao C, Jiang J, et al. REDPC: a residual error-based density peak clustering algorithm. *Neurocomputing*. (2019) 348:82–96. doi: 10.1016/j.neucom.2018.06.087

33. Wang Y, Wang D, Pang W, Miao C, Tan AH, Zhou Y. A systematic density-based clustering method using anchor points. *Neurocomputing*. (2020) 400:352–70. doi: 10.1016/j.neucom.2020.02.119

34. Wang Y, Yang Y. Relative density-based clustering algorithm for identifying diverse density clusters effectively. *Neural Comput Appl*. (2021) 33:10141–57. doi: 10.1007/s00521-021-05777-2

35. Xu T, Jiang J. A graph adaptive density peaks clustering algorithm for automatic centroid selection and effective aggregation. *Expert Syst Appl*. (2022) 195:116539. doi: 10.1016/j.eswa.2022.116539

36. Zhao J, Wang G, Pan JS, Fan T, Lee I. Density peaks clustering algorithm based on fuzzy and weighted shared neighbor for uneven density datasets. *Pattern Recognit*. (2023) 139:109406. doi: 10.1016/j.patcog.2023.109406

37. Xie J, Liu X, Wang M. SFKNN-DPC: standard deviation weighted distance based density peak clustering algorithm. *Inf Sci*. (2024) 653:119788. doi: 10.1016/j.ins.2023.119788

38. Yan H, Wang M, Xie J. ANN-DPC: density peak clustering by finding the adaptive nearest neighbors. *Knowl-Based Syst*. (2024) 294:111748. doi: 10.1016/j.knosys.2024.111748

39. Fan JC, Jia PL, Ge L. Mk-NNG-DPC: density peaks clustering based on improved mutual K-nearest-neighbor graph. *Int J Mach Learn Cybern*. (2019) 11:1179–95. doi: 10.1007/s13042-019-01031-3

40. Li C, Ding S, Xu X, Hou H, Ding L. Fast density peaks clustering algorithm based on improved mutual K-nearest-neighbor and sub-cluster merging. *Inf Sci*. (2023) 647:19. doi: 10.1016/j.ins.2023.119470

41. Yu D, Liu G, Guo M, Liu X, Yao S. Density peaks clustering based on weighted local density sequence and nearest neighbor assignment. *IEEE Access*. (2019) 7:34301–17. doi: 10.1109/ACCESS.2019.2904254

42. Ren C, Sun L, Gao Y, Yu Y. Density peaks clustering based on local fair density and fuzzy k-nearest neighbors membership allocation strategy. *J Intell Fuzzy Syst*. (2022) 43:21–34. doi: 10.3233/JIFS-202449

43. Wang Y, Qian J, Hassan M, Zhang X, Zhang T, Yang C, et al. Density peak clustering algorithms: a review on the decade 2014-2023. *Expert Syst Appl*. (2024) 238:121860. doi: 10.1016/j.eswa.2023.121860

44. Dua D, Graff C. *UCI Machine Learning Repository*. Irvine, CA: University of California, School of Information and Computer Sciences (2017). Available online at: http://archive.ics.uci.edu/ml (Accessed September 19, 2024).

45. Fränti P, Rezaei M, Zhao Q. Centroid index: cluster level similarity measure. *Pattern Recognit*. (2014) 47:3034–45. doi: 10.1016/j.patcog.2014.03.017

46. Vinh NX, Epps J, Bailey J. Information theoretic measures for clusterings comparison: variants, properties, normalization and correction for chance. *J Mach Learn Res*. (2010) 11:2837–54.

47. Boudane F, Berrichi A. Gabriel graph-based connectivity and density for internal validity of clustering. *Prog Artif Intell*. (2020) 9:221–38. doi: 10.1007/s13748-020-00209-z

48. Cambridge AL. *The Database of Faces*. AT&T Laboratories Cambridge (1992). Original dataset release with 400 images of 40 subjects. Available online at: https://www.cl.cam.ac.uk/research/dtg/attarchive/facedatabase.html (Accessed September 19, 2024).

49. Ding S, Du W, Xu X, Shi T, Wang Y, Li C. An improved density peaks clustering algorithm based on natural neighbor with a merging strategy. *Inf Sci*. (2023) 624:252–76. doi: 10.1016/j.ins.2022.12.078