F-DPC: Fuzzy Neighborhood-Based Density Peak Algorithm

YANG LI, WENJU ZHOU, (Member, IEEE), AND HAIKUAN WANG
School of Mechanical and Electrical Engineering and Automation, Shanghai University, Shanghai 200444, China

Corresponding author: Wenju Zhou (zhouwenju@shu.edu.cn)

This work was supported in part by the NSFC, China, under Project 6177065, and in part by the 111 Project under Grant D18003.

ABSTRACT Clustering is a concept in data mining, which divides a data set into different classes or clusters according to a specific standard, making the similarity of data objects in the same cluster as large as possible. Clustering by fast search and find of density peaks (DPC) is a novel clustering algorithm based on density. It is simple and novel, only requiring fewer parameters to achieve better clustering effect, without the requirement for iterative solution. And it has expandability and can detect the clustering of any shape. However, DPC algorithm still has some defects, such as it employs the clear neighborhood relations to calculate local density, so it cannot identify the neighborhood membership of different values of points from the distance of points and It is impossible to accurately cluster the data of the multi-density peak. The fuzzy neighborhood density peak clustering algorithm is proposed for this shortcoming (F-DPC): novel local density is defined by the fuzzy neighborhood relationship. The fuzzy set theory can be used to make the fuzzy neighborhood function of local density more sensitive, so that the clustering for data set of various shapes and densities is more robust. Experiments show that the algorithm has high accuracy and robustness.

INDEX TERMS Clustering, degree of membership, fuzzy neighborhood, local density.

I. INTRODUCTION Clustering is an unsupervised learning method [1] called pattern recognition, it divides data samples into several class clusters, so that the similarity of samples in the same class cluster is high, and the similarity of samples in different class clusters is low. Cluster analysis can find useful information from data, explain the hidden relationships and laws between data, and have extensive applications [2], [3] in engineering systems, computer science, life and medical science, social science, and economic fields. In essence, Clustering can be considered as a process of unsupervised learning [4]–[6], since clustering is classified into the same cluster by measuring the similarity and dissimilarity of the objects in the cluster. According to the different principles given, clustering algorithms can be roughly divided into partition-based clustering algorithms, hierarchical-based clustering algorithms, density-based clustering algorithms, grid-based clustering algorithms, model-based clustering algorithms and graphics-based clustering algorithms. The K-means algorithm [7] is the most widely used partition-based clustering algorithm, this algorithm is fast and easy to be realized, but the K-means algorithm is essentially a greedy descent algorithm for non-convex cost function optimization, which can only obtain the local optimal solution. In addition, the K needs to be given in advance, and the initial clustering center has a great impact on the clustering results. In view of the above shortcomings, K-Mean SCAN and other improved algorithm [8] appeared. CHAMELEON algorithm [9] is a hierarchical-based clustering algorithm, which uses dynamic modeling to determine the similarity between a pair of clusters, while considering the interconnectedness and proximity of each cluster. However, the thresholds of similarity function and the value of K in K- nearest neighbor graph of CHAMELEON algorithm need to be given manually. The density-based clustering algorithm DBSCAN [10] and its improved algorithm [11] can find class clusters of arbitrary shape, which are insensitive to noise, but the clustering results are heavily dependent on the initial parameters. SNN algorithm [12] and Rock algorithm [13] are two excellent clustering algorithms, while dealing with datasets containing different densities, SNN algorithm is better than other algorithms. However, due to the need to calculate similarity matrix, the algorithm has higher time complexity. The spectral clustering algorithm [14] first calculates the similarity affinity matrix between samples according to the
II. RELATED WORK

Recently, Science has published an original clustering algorithm [16]– clustering by fast search and find of density peaks (DPC) [27]–[29]. The algorithm needs less parameters and can detect clusters of arbitrary shape and dimension, and is not sensitive to noise. The clustering algorithm is divided into two stages: in the first stage, the algorithm uses the input parameters of the user to calculate the local density and distance of the sample, and finds the clustering center which is so-called density peak, then, select appropriate clustering centers from the samples according to decision diagrams; in the second stage, the remaining samples are distributed to the cluster where the closed and higher density samples are located, and the whole clustering process is simple and efficient.

The core idea of density peak clustering algorithm [30],[31] is to find the best clustering center, the ideal clustering center should satisfy the following two conditions: (1) The density of the center cluster itself is very large, and it is surrounded by a neighborhood of no more density than its neighborhood; (2) The relative distance of different cluster centers is farther. Therefore, the DPC algorithm introduces two conditions, the local density \( \rho \) and the distance \( \delta \), which correspond to the above clustering center respectively. For any sample \( i \), the definitions of local density \( \rho_i \) and distance \( \delta_i \) are shown in the following expressions (1) and (2):

\[
\rho_i = \sum_{j \neq i} \chi(d_{ij} - d_c) \quad (1)
\]

\[
\delta_i = \min_{j, \rho_j > \rho_i} (d_{ij}) \quad (2)
\]

Among them, \( d_{ij} \) is the distance between samples \( i \) and \( j \), and \( d_c \) is cut-off distance, which requires users to specify in advance. In the function \( \chi(x) \), when the independent variable \( x < 0, \chi(x) = 1 \). Otherwise, \( \chi(x) = 0 \).

For its local density \( \rho \), the maximum sample \( i \), the distance is \( \delta_i = \max_j (d_{ij}) \). In addition, Rodriguez and Laio gives another way to calculate the local density by using Gauss’s function, as is shown in formula (3):

\[
\rho_i = \sum_{j \neq i} e^{-\left(\frac{d_{ij}}{\epsilon}\right)^2} \quad (3)
\]

In formula (1), \( \rho_i \) represents the number of samples in the dataset and the sample \( i \) with a distance less than \( d_c \). In formula (3), \( \rho_i \) represents the sum of the weighted values of the distance between sample \( i \) and other samples in the dataset, in theory, when calculating the local density using formula (3), the probability of different samples having the same local density is smaller. Therefore, for the dataset with fewer samples, formula (3) still have the more data points whose distance of \( x_i \) is less than \( d_c \), the larger the value of \( \rho_i \) is. In the formula (2), the distance \( \delta \) between the samples indicates the distance between the sample and the nearest neighbor samples with higher local density \( \rho \). In the original algorithm, the local density \( \rho \) and the distance \( \delta \) are used to construct the decision diagram, and the samples which are larger in both \( \rho \) and \( \delta \) are selected as cluster centers, and each cluster center is a class cluster. The remaining samples are then assigned to clusters with higher local density and nearest neighbor samples. As shown in formula (2): when \( x_i \) has the largest local density, \( \delta_i \) represents the distance between the \( x_i \) and the data points with the largest distance from \( x_i \) in a dataset; Otherwise, \( \delta_i \) represents the distance between the \( x_i \) and the data point with the smallest distance from \( x_i \) in all data points with local density greater than \( x_i \).

Although the calculation of \( \delta \) is easier to understand, there may be problems in some special cases. For example, if \( \rho_i = \rho_j = \max_{k \in I_s} (\rho_k) \), and data points \( x_i \) and \( x_j \) happen to belong to the same class and the distance is closer, in accordance with the definition, there will be \( \delta_i = \max_{j \in I_s} (d_{ij}), \delta_j = \max_{k \in I} (d_{jk}) \). If both \( \rho_i \) and \( \rho_j \) are larger, the two data points are selected as cluster centers according to the principle of selecting cluster centers, and a class is dismantled into two classes.

DPC algorithm can identify clusters of arbitrary shape and dimension [32]–[35], but some shortcomings of the algorithm will affect the final clustering results. In addition, the algorithm belongs to the hard clustering algorithm, once a sample is allocated wrongly, the related samples will be misallocated. Figure 1 a and b calculate the local density using the formula (1) and the formula (3) respectively, the sample \( F \) is a class cluster with multiple density peaks. Observation shows that the final clustering result is not ideal because the sample is wrongly assigned to the nearest cluster center \( B \).

In view of the above shortcomings, it is found that DPC algorithm is very effective for convex data clustering. However, when the data points between classes are unevenly distributed, it is not sufficient to determine the membership degree of the central point only through the number of center points within the radius of the neighborhood. As shown in Figure 1, points \( x1 \) and \( x2 \) have the same number of points in the neighborhood, the same neighborhood radius \( \varepsilon \), and the neighborhood radius less than the maximum
In the classical case, there is no difference in membership degree between points within the same neighborhood radius of the core point, that is to say, \( x_1 \) and \( x_2 \) have the same neighborhood membership degree. However, it can be seen from Figure 2 that there is a certain gap between \( x_1 \) and \( x_2 \).

Juanying Xie and other [17], [18] proposed an improved algorithm KNN-DPC algorithm and FKNN-DPC algorithm based on DPC algorithm, the algorithm combines the K nearest neighbor idea and solves the defect of the DPC algorithm when measuring the sample density. Wu C proposes an effective clustering method based on density peaks with symmetric neighborhood relations [36]. A feasible residual-based density peak clustering algorithm with fragment merging strategy proposed by Parmar MD [37]. Yan H proposes to use statistical outlier detection method to automatically identify cluster centroids from decision graphs [38]. Parmar M D propose a residual error-based density peak clustering algorithm named REDPC to better handle datasets comprising various data distribution patterns [39]. Parmar MD propose a novel density peak clustering algorithm based on squared residual error [40]. Zhang and Li [19] solved the problem of not recognizing low density clusters by using CHAMELEON algorithm. Rapid searching of gene local density peaks and merging clustering for expressing microarray data proposed by R Mehmood solved clustering problems of different shapes. However, there are still some shortcomings in the DPC algorithm: (1) In the selection of clustering center, DPC algorithm does not give a clear criterion of membership degree, and the correct membership degree can not be obtained by distance measurement; (2) If there are multiple density peaks in a class cluster of dataset or the class cluster with tight density, the clustering results of this algorithm can not achieve the desired results.

In view of this, a density peak clustering algorithm based on Fuzzy neighborhood is proposed in this paper (F-DPC). The algorithm improves the DPC algorithm from two aspects: (1) The algorithm uses the fuzzy set theory to define a new kind of fuzzy neighborhood relation, which can give a definite membership degree standard and identify the correct membership degree of the points within the same neighborhood radius. (2) The algorithm uses the fuzzy neighborhood strategy to get a new kind of local density, which improves the accuracy of the selection of cluster centers, and is more robust to various shapes and density data, and solves the shortage of clustering effect when there are multiple peaks in a class.

III. THE PROPOSED ALGORITHM F-DPC

According to the above analysis, it is limited to determine the membership degree of the center by simply using the number of center points in the neighborhood radius. Especially for unbalanced data, the limitations are greater. In classical clustering, the boundaries of different clusters are very weak, and each pattern is assigned to a unique class. On the other hand, boundaries between clusters cannot be precisely defined in real life, so that certain patterns may belong
to multiple different cluster members. In this case, fuzzy clustering can achieve better results. Fuzzy set theory is widely used to solve such problems, so this paper proposes a new method to solve the problems in the original algorithm by using rough set theory.

The density peak algorithm is described as follows:

Step 1 Initialization and preprocessing
1.1 Give the parameter \( t \in (0, 1) \) for determining the cut-off distance \( d_c \).
1.2 Calculate the distance \( d_{ij} \), and make \( d_{ij} = d_{ji}, i < j, i, j \in I_s \).
1.3 Determine the phase distance \( d_i \) calculated in the previous step is arranged in ascending order of \( M = \frac{1}{2}N(N - 1) \), let the resulting sequence be \( d_1 \leq d_2 \leq \cdots \leq d_M \), take \( d_i = d_{(M/2)} \), where \( f(Mt) \) represents the integer obtained after rounding \( Mt \).
1.4 According to (1.1) or (1.3) to calculate \( \rho_{i1}^N \), and generate the descending order \( \{q_i\}_{i=1}^N \).
1.5 Calculate \( \{d_i\}_{i=1}^N \) and \( \{n_i\}_{i=1}^N \).

Step 2 Determine the cluster center \( \{m_j\}_{j=1}^N \), and initialize the data point classification attribute tag \( \{c_i\}_{i=1}^N \), specific for

\[
\begin{cases} 
  k, & \text{if } x_i \text{ is the cluster center and belongs to the class } k \\
  -1, & \text{others}
\end{cases}
\]

Step 3 Classified the data points of non clustering center.
Step 4 If \( n_i > 1 \), the data points in each class are further divided into cluster core and cluster halo, initialize the mark \( h_i = 0, i \in I_s \). Generate an average local density upper bound \( \{\rho_i\}_{i=1}^N \) for each class, and identify the cluster halo.

**Definition 1 (The Basic Concept of Fuzzy Joint Point Method):** Fuzzy joint point is based on a level based view to explain vagueness. It represents how many elements should be considered when constructing homogeneous groups. Obviously, when the elements are discussed in more detail, the difference between them is even greater. The more fuzzy the elements are, the more similar they are to each other, in this case, the fuzzy neighborhood can more detailed point out the types of attributes considered. Since all elements differ from each other when the minimum ambiguity is zero, each element can be considered to be divided into the same class in a similar manner. Similar elements will belong to a class, and elements that differ from each other will belong to different classes, so they have different membership degrees.

Let \( F(R^m) \) represent the m-dimensional fuzzy set of \( R^m \), and use \( \mu^A : R^m \rightarrow [0, 1] \) to represent the membership function of the fuzzy set, and set \( A \in F(R^m) \). In the conical fuzzy space shown in Figure 3, the conical fuzzy point \( A = (a, R) \in F(R^m) \), where the space \( R^m \) represents a fuzzy set with membership functions, the definition is as shown in (4):

\[
\mu^A = \begin{cases} 
  1 - \frac{d(a,x)}{R} & \text{if } d(a,x) < R \\
  0 & \text{otherwise}
\end{cases} \tag{4}
\]

where \( a \in R^m \) is the center of fuzzy point \( A \), \( R \) is the support radius of \( A \), the calculation formula is shown in formula (5):

\[
supp A = \{ x \in R^m | \mu^A(x) > 0 \} \tag{5}
\]

The \( \alpha \) level set of the conical fuzzy point \( A = (a, R) \) is calculated as formula (6):

\[
A_\alpha = \{ x \in R^m | \mu^A(x) > \alpha \} = \{ x \in R^m | d(a,x) < R < R \cdot (1 - \alpha) \} \tag{6}
\]

**Definition 2 (Distance Between Fuzzy Points):** Let \( A = (a,R), B = (b,R) \) belong to fuzzy points on the fuzzy set \( X \subset F(E^p) \), and the neighborhood fuzzy relation is expressed as \( T : X \times X \rightarrow [0, 1] \), which is represented in the fuzzy set \( X \) as shown in the formula (7):

\[
T(A, B) = 1 - \frac{d(a,b)}{2R} \tag{7}
\]

where \( d(a,b) \) is the distance between \( a \) and \( b \), \( R \) is the neighborhood radius. When \( a \in E^p, b \in E^p \), the distance between the fuzzy center point \( A \) and \( B \) is shown in formula (8):

\[
d(a,b) = 2R(1 - T(A, B)) \tag{8}
\]

Therefore, the mapping of relation \( T \) satisfies is \( \forall A \in X : T(A, A) = 1 \).

**Definition 3 (\( \alpha \)-Neighborhood Fuzzy Points):** Let fuzzy points \( A \) and \( B \) in fuzzy set \( X \subset F(E^p) \), if \( T(A, B) \geq \alpha \) satisfies \( \alpha \in (0, 1] \), then point \( A \) and point \( B \) are called \( \alpha \)-neighborhood fuzzy points and are represented as \( A \sim A^\alpha \).

The \( \alpha \) correlation coefficient of each point can be obtained by the given \( \alpha \)-neighborhood.

**Lemma 1:** Fuzzy point \( A = (a,R) \) and point \( B = (b,R) \) are \( \alpha \)-neighborhood fuzzy points, if \( d(a,b) \leq 2R(1 - \alpha) \),
d(a, b) represents the distance between fuzzy central point A and point B.

Let the fuzzy point A = (a, R) and point B = (B, R) be the α-neighborhood, formula (9) can be obtained according to the T(a, b) ≥ α and formula (7):

\[
1 - \frac{d(a, b)}{2R} \geq \alpha \Rightarrow d(a, b) \leq 2R(1 - \alpha) \quad (9)
\]

From T(a, b) ≥ α, we can get α ≤ 1 - \frac{d(a, b)}{2R}, so it is proved that the Definition 3 is accurate.

If there are a series of α-neighborhood fuzzy points C1, C2, ..., Ck, k ≥ 0, k ≥ 0, for the fixed α ∈ (0, 1], there is formula (10) between point A and B:

\[
A \sim_{α} C^1, C^1 \sim_{α} C^2, ..., C^{k-1} \sim_{α} C^k, C^k \sim_{α} B \quad (10)
\]

Point A and point B are called α-joint fuzzy points as shown in Figure 4.

![FIGURE 4. Joint blur point.](image)

**Definition 5 (α-Joint Fuzzy Point Distance):**

Let X ⊂ F (E^p) be a point on the fuzzy set, if A and B are fuzzy points on the α-neighborhood, α ∈ (0, 1] and A, B ∈ X, where X is called the α-neighborhood fuzzy set. The d(A_a, B_a) represents the horizontal distance between A_a and B_a on the dataset, it is expressed as follows:

\[
d(A_a, B_a) = \min \{d(x, y)|x \in A_a, y \in B_a\} \quad (11)
\]

**Lemma 2:** The fuzzy point A and the fuzzy point B are the fuzzy points of the α-neighborhood, the A_a ∩ B_a ≠ Ø.

**Proof:** Let the fuzzy point A and the fuzzy point B be the fuzzy points of the α-neighborhood, so T(a, b) ≥ α. First of all, assume that A_a ∩ B_a ≠ Ø is incorrect, that is A_a ∩ B_a ≠ Ø. Then let the joint points a ∈ E^p and b ∈ E^p, and x ∈ E^p, x \notin A_a, x \notin B_a. Thus, formula (11) is obtained:

\[
d(a, x) > R(1 - \alpha) \quad \text{and} \quad d(b, x) > R(1 - \alpha) \quad (12)
\]

According to the inequality given by Lemma 1 and the α-neighborhood of point A and point B, assuming A_a ∩ B_a ≠ Ø, then \exists x: x \in A_a, x \in B_a, therefore, the formula (13) is obtained:

\[
d(a, b) = d(a, x) + d(x, b) > 2R(1 - \alpha) \quad (13)
\]

According to the inequality given by Lemma 1 and the α-neighborhood of point A and point B, assuming A_a ∩ B_a ≠ Ø, then \exists x: x \in A_a, x \in B_a, therefore, the formula (14) is obtained:

\[
d(a, b) \leq d(a, x) + d(x, b) \leq 2R(1 - \alpha) \Rightarrow d(a, b) \leq 2R(1 - \alpha) \quad (14)
\]

**Definition 6 (Details of Distance and Local Density):** Let X = \{x_1, x_2, ..., x_n\} represents n data objects in the dataset, each object x_i, 1 ≤ x ≤ n has m attributes. Therefore, for each i, 1 ≤ i ≤ n, and each j, 1 ≤ j ≤ n, let x_{ij} be the j attribute of x_i. So the distance between point x_i and x_j, x_i, x_j ∈ X is expressed as:

\[
d(x_i, x_j) = (\sum_{k=1}^{m} (x_{ik} - x_{jk})^2)^{1/2} \quad (15)
\]

So in order to form a fuzzy relation \( \mu : X \times X \rightarrow [0, 1] \), a concept of fuzzy membership grade is defined, in which the radius of the neighborhood is the only input parameter determined by the percentage, which is called the truncated distance. The neighborhood radius \( \varepsilon = d[d_{max} \cdot d_c] \), \([\cdot]\) indicates the upper limit of the set, \( d_{max} = d_{\max_{nd}} = \max d(x_i, x_j) \). One of the input parameters is the percentage of \( d_c \). A point \( x_i \in X \) and parameter \( \varepsilon \) on a neighborhood set follows the following relation \( \mathcal{N}(X_i, \varepsilon) = \{X_j \in X | d(X_i, X_j) < \varepsilon \} \). The formula for calculating the local density is obtained by formula (4): \( \rho_i = \sum_j \mu_{X_i}(x_j) \), the truncation distance is calculated as follows:

\[
\delta_i = \begin{cases} 
\min_{j: \rho_i < \rho_j} d(X_i, X_j) & \text{if } \exists j: \rho_i < \rho_j \\
\max_j d(X_i, X_j) & \text{otherwise}
\end{cases} \quad (16)
\]

Through the formula (8), the calculation formula of the fuzzy point is obtained, and according to the above definition to calculate the distance of any fuzzy point in the dataset, and the fuzzy distance formula is obtained as shown in formula (15), then according to the definition of the local density, the formula of fuzzy neighborhood density function is obtained as follows:

\[
\rho_i = \sum_j \mu_{X_i}(d(x_i, x_j) - d_c) \quad (17)
\]

where \( d_c \) represents the truncation distance.

The fuzzy neighborhood density peak algorithm is described as follows:

**Step 1** Input dataset \( X \in R^{n \times m} \) and parameter \( d_c \).

**Step 2** Calculate the distance matrix by the formula (14).

**Step 3** Calculate the fuzzy neighborhood density \( \rho_i \) of point \( X_i \) by the formula \( \rho_i = \sum_j \mu_{X_i}(d(x_i, x_j) - d_c) \).

**Step 4** Calculate the truncation distance \( \delta_i \) of the point \( X_i \) by the formula (15).

**Step 5** Draw out the decision map and select the center of the class.

**Step 6** Assign the remaining points to clusters by using algorithms in related work.

**Step 7** Output the cluster index label vector.
The algorithm flow chart is shown in Figure 5. The F-DPC algorithm requires fewer parameters to achieve better clustering results. The algorithm only contains an optional parameter \( t \). In fact, this parameter is used to determine \( d_c \), so the essential parameter should be the cutoff distance parameter.

**FIGURE 5. Algorithm flowchart.**

### IV. EXPERIMENTAL RESULTS AND ANALYSIS

In order to test the clustering results of the algorithm, this paper uses the classical artificial dataset and the real dataset on UCI to conduct experiments, in order to further compare the F-DPC algorithm and the original algorithm based on clear neighborhood relation, we compare the algorithm with DBSCAN, X-means, AP, FCM, BIRCH, KNN-DPC, FKNN-DPC, and the original algorithm, these algorithms are representative clustering algorithms. All the experimental environments are Win10 64bit operating system, Matlab software, 4G memory, Intel (R) Core (TM) i5-3210M CPU@2.50GHz. In this chapter, 3.1 briefly describes the experimental dataset information and evaluation criteria, 3.2 analyzes the clustering results of artificial datasets, 3.3 gives the specific experimental results.

#### A. EXPERIMENTAL DATA SETS AND EVALUATION CRITERIA

The information of the experimental dataset is shown in Table 1, the first seven are artificial datasets [20]–[24], and the latter fourteen are real datasets on UCI, which are commonly used to test datasets, and have a certain degree of division on the number of attributes and the number of class clusters. In view of the fact that the attributes of some datasets are not in the same order of magnitude, they are normalized.

| Dataset   | Size | Attribute |
|-----------|------|-----------|
| R15       | 600  | 2         |
| Shape     | 1000 | 2         |
| Spiral    | 312  | 2         |
| Simle     | 1000 | 2         |
| Banana    | 4811 | 2         |
| SI        | 5000 | 2         |
| Jain      | 373  | 2         |
| Iris      | 150  | 4         |
| Art       | 300  | 4         |
| Compound  | 399  | 2         |
| Flame     | 64   | 2         |
| Ecoli     | 336  | 7         |
| Seeds     | 210  | 7         |
| Waveform  | 5000 | 21        |
| Image Segmentation | 2310 | 19    |
| Ionosphere| 351  | 34        |
| Libras Movement | 360  | 90      |
| WDBC      | 569  | 30        |
| Dim256    | 1024 | 256       |
| Dim1024   | 1024 | 1024      |
| Wine      | 178  | 13        |

**TABLE 1. Dataset size and attribute.**

The evaluation criteria for the experimental results are accuracy (ACC), recall (Re) and normalized mutual information (NMI), and the accuracy and recall are two metrics widely used in the field of information retrieval and statistics to evaluate the quality of the clustering results. Mutual information is a measure of interdependence between variables. The three evaluation indexes range from 0 to 1, and the larger the value, the better the clustering effect. The following are defined as follows:

\[
ACC = \frac{\sum_{i=1}^{k} a_i}{|U|} \quad (18)
\]

\[
Re = \frac{\sum_{i=1}^{k} \frac{a_i}{a_i + c_i}}{k} \quad (19)
\]

Among them, \( k \) is the number of clusters, and \( a_i \) indicates the number of samples correctly classified to class cluster \( C_i \). \( U \) is a full sample, \( c_i \) indicates the number of clusters belonging to the class cluster \( c_i \) being misclassified into other clusters.

\[
NMI(X, Y) = \frac{I(X, Y)}{\sqrt{H(X)H(Y)}} \quad (20)
\]

Among them, \( X \) and \( Y \) represent random variables, and \( I(X, Y) \) represents mutual information of two variables, \( H(X) \) represents the entropy of \( X \). The following are defined as follows:

\[
I(X, Y) = \sum_{h=1}^{k(a)} \sum_{l=1}^{k(b)} n_{h,l} \log \left( \frac{n \cdot n_{h,l}}{n_{h}^{a} n_{l}^{b}} \right) \quad (21)
\]

\[
H(X) = \sum_{h=1}^{k(a)} n_{h}^{a} \log \left( \frac{n_{h}}{n} \right) \quad (22)
\]

\[
H(Y) = \sum_{l=1}^{k(b)} n_{l}^{b} \log \left( \frac{n_{l}}{n} \right) \quad (23)
\]

#### B. CLUSTERING RESULTS AND ANALYSIS OF ARTIFICIAL DATA SETS

As shown in figure6 (a)-(f), the proposed method performs well in clustering datasets with spherical or elliptical shape.
The dataset is a large dataset with different number of clusters. Experimental results show that the F-DPC algorithm is robust in quantity. As shown in figure (d) (e) (f), the algorithm has good performance in the three datasets with different sizes and shapes. The dataset of graph (a) (b) (c) is a two-dimensional set with different complexity in spatial data distribution, and has different degree of overlap. The performance of the proposed F-DPC algorithm is perfect for datasets with different complexity. Due to these experiments show that the algorithm in this paper is very effective in finding clusters of arbitrary shape, density, distribution and quantity, and the results are very good. The algorithm solves the defects of the original algorithm and can correctly calculate the membership grade of each class and then cluster.

C. EXPERIMENTAL RESULTS AND ANALYSIS ON VARIOUS PERFORMANCE INDEXES

The results of this paper are shown in the following tables, Table 2 is the accuracy rate (ACC) of clustering results for each algorithm, Table 3 is the recall rate (Re) of the clustering results of each algorithm, and Table 4 is the normalized mutual information (NMI) [25], [26] of the clustering results of each algorithm. Table 5 compares the running time of clustering algorithm on UCI dataset. The figures highlighted in bold in the following table show that the corresponding assessment has the best performance. The figures highlighted in bold in the following table show that the corresponding assessment has the best performance.

The results of the three evaluation indicators, we can see that the clustering results of F-DPC algorithm are generally better than other clustering algorithms. For the datasets of Simle, S1, Jain and Flame, the results of algorithm in this paper are the same as that of the density peak clustering algorithm, and their accuracy, recall and mutual information are all higher than those of other algorithms. Except for Iris and Ionosphere datasets, it can be seen from other datasets that the indexes of algorithm in this paper are higher than those of other algorithms. However, from the Iris and Ionosphere datasets, we can see that the accuracy and recall of the algorithm in this paper are higher than those of other algorithms, and the mutual information results are lower than the density peak clustering algorithm, it’s because there are two clusters with the same density, and there are errors in the selection, although it did not get the ideal effect, the other indexes are higher than those of the other algorithms. In the unbalanced data set Art, Compound, Ecoli, Waveform, Image Segmentation, Ionosphere, Libras Movement, it can be seen that the clustering effect of the F-DPC algorithm is better than other algorithms. Experimental results show that the algorithm proposed in this paper achieves ideal clustering results, and overcomes the shortcomings of KNN-DPC and FKNN-DPC algorithms that cannot accurately cluster unbalanced data.

Table 5 takes the UCI dataset as an example to compare the proposed algorithm with the original DPC algorithm, as well as the classic AP, DBSCAN, FCM, X-means, BIRCH clustering algorithms, and the time performance of the

| Dataset | DBSCAN | X-means | AP | FCM | BIRCH | DPC | KNN-DPC | FKNN-DPC | F-DPC |
|---------|--------|---------|----|-----|-------|-----|---------|---------|-------|
| R15     | 90.10  | 93.15   | 85.42 | 79.25 | 88.14 | 95.60 | 97.6   | 98.20   | 99.30 |
| Shape   | 88.30  | 90.72   | 79.63 | 80.00 | 77.23 | 92.80 | 94.23  | 95.64   | 100   |
| Spiral  | 80.40  | 89.53   | 74.00 | 74.67 | 73.33 | 87.70 | 91.25  | 96.32   | 100   |
| Simle   | 98.70  | 100     | 65.63 | 84.21 | 64.13 | 100   | 100    | 100     | 100   |
| Banana  | 75.10  | 79.92   | 79.22 | 79.70 | 60.16 | 80.20 | 90.12  | 92.63   | 100   |
| S1      | 96.80  | 90.41   | 85.60 | 79.32 | 61.32 | 99.42 | 95.80  | 96.30   | 99.42 |
| Jain    | 93.00  | 89.86   | 80.40 | 79.64 | 86.23 | 100   | 100    | 100     | 100   |
| Iris    | 63.30  | 88.73   | 90.67 | 89.33 | 84.00 | 76.70 | 96.40  | 96.40   | 99.70 |
| Art     | 66.70  | 83.24   | 80.67 | 80.00 | 84.00 | 76.70 | 79.60  | 80.20   | 84.70 |
| Compound| 75.70  | 76.65   | 71.43 | 65.66 | 76.69 | 76.40 | 81.46  | 82.51   | 87.20 |
| Flame   | 96.20  | 98.17   | 98.90 | 97.20 | 90.10 | 100   | 100    | 100     | 100   |
| Ecoli   | 57.80  | 70.32   | 75.98 | 53.57 | 76.49 | 72.30 | 74.80  | 75.30   | 77.00 |
| Seeds   | 71.40  | 78.56   | 89.52 | 90.00 | 89.05 | 90.10 | 91.40  | 91.40   | 91.40 |
| Waveform| 40.30  | 50.77   | 53.24 | 49.30 | 53.72 | 55.00 | 65.40  | 64.80   | 68.10 |
| Image Segmentation | 41.04 | 53.85 | 56.57 | 62.38 | 42.34 | 62.77 | 68.40 | 68.40 | 69.55 |
| Ionosphere | 72.08 | 71.29 | 70.94 | 70.94 | 64.67 | 68.10 | 72.90 | 75.20 | 79.73 |
| Libras Movement | 30.00 | 33.51 | 45.56 | 18.33 | 28.61 | 41.94 | 45.30 | 43.60 | 60.67 |
| WDBC    | 63.09  | 80.16   | 92.44 | 92.79 | 67.14 | 82.95 | 94.30  | 94.40   | 95.32 |
| Dim256  | 95.60  | 91.54   | 97.20 | 62.50 | 43.75 | 100   | 100    | 100     | 100   |
| Dim1024 | 94.22  | 95.28   | 96.40 | 68.75 | 100   | 100   | 100    | 100     | 100   |
improved algorithm KNN-DPC and FKNN-DPC. From the algorithm flow, we can see that the time complexity of DPC and the algorithm in this paper is $O(n^2)$, and $n$ is the size of dataset, that is, the number of samples. From the experimental
results, we can see that the algorithm in this paper inherits the time advantage of DPC algorithm, and running time takes less time than other algorithms, so it has some advantages in both running time and memory consumption.

V. CONCLUDING REMARKS

The density peak clustering algorithm is not ideal for neighborhood boundary clustering, to solve this problem, a new fuzzy neighborhood density function is introduced and proposed algorithm F-DPC. Comparison experiments show that our algorithm is very effective in finding clusters of arbitrary shape, density, distribution and quantity. It is observed that the algorithm in this paper is not only more robust than the original algorithm for datasets with various shapes and densities, but also significantly improves the clustering effect compared with other improved algorithms. From the experimental results, we can see that the algorithm in this paper effectively solves the shortcomings of the density peak clustering algorithm, and then get an ideal clustering results. However, the algorithm in this article takes a lot of time when the data set is large. How to improve the efficiency of the FDPC algorithm will be our next focus.

ACKNOWLEDGMENT

The authors would like to thank the editors and anonymous reviewers for their valuable comments and helpful suggestions, which have greatly improved the paper’s quality.
