Hindawi
Security and Communication Networks
Volume 2024, Article ID 9818710, 1 page
https://doi.org/10.1155/2024/9818710

Retraction

Retracted: A Dynamic Density Peak Clustering Algorithm Based on K-Nearest Neighbor

Security and Communication Networks

Received 8 January 2024; Accepted 8 January 2024; Published 9 January 2024

This article has been retracted by Hindawi following an investigation undertaken by the publisher [1]. This investigation has uncovered evidence of one or more of the following indicators of systematic manipulation of the publication process:

(1) Discrepancies in scope
(2) Discrepancies in the description of the research reported
(3) Discrepancies between the availability of data and the research described
(4) Inappropriate citations
(5) Incoherent, meaningless and/or irrelevant content included in the article
(6) Manipulated or compromised peer review

The presence of these indicators undermines our confidence in the integrity of the article's content and we cannot, therefore, vouch for its reliability. Please note that this notice is intended solely to alert readers that the content of this article is unreliable. We have not investigated whether authors were aware of or involved in the systematic manipulation of the publication process.

Wiley and Hindawi regrets that the usual quality checks did not identify these issues before publication and have since put additional measures in place to safeguard research integrity.

We wish to credit our own Research Integrity and Research Publishing teams and anonymous and named external researchers and research integrity experts for contributing to this investigation.

The corresponding author, as the representative of all authors, has been given the opportunity to register their agreement or disagreement to this retraction. We have kept a record of any response received.

References

[1] H. Du, Q. Zhai, Z. Wang, Y. Li, and M. Zhang, "A Dynamic Density Peak Clustering Algorithm Based on K-Nearest Neighbor," *Security and Communication Networks*, vol. 2022, Article ID 7378801, 15 pages, 2022.

Hindawi
Security and Communication Networks
Volume 2022, Article ID 7378801, 15 pages
https://doi.org/10.1155/2022/7378801

Research Article

A Dynamic Density Peak Clustering Algorithm Based on K-Nearest Neighbor

Hui Du, Qiaofeng Zhai, Zhihe Wang, Yongbiao Li, and Manjie Zhang

The School of Computer Science and Engineering, Northwest Normal University, Lanzhou 730070, China

Correspondence should be addressed to Qiaofeng Zhai; 2020222011@nwnu.edu.cn

Received 29 October 2021; Revised 18 March 2022; Accepted 2 June 2022; Published 9 August 2022

Academic Editor: Xingsi Xue

The clustering results of the density peak clustering algorithm (DPC) are greatly affected by the parameter d_c, and the clustering center needs to be selected manually. To solve these problems, this paper proposes a low parameter sensitivity dynamic density peak clustering algorithm based on K-Nearest Neighbor (DDPC), and the clustering label is allocated adaptively by analyzing the distribution of K-Nearest Neighbors around each data. It reduces the parameter sensitivity and eliminates selecting the clustering centers manually from the decision graph. Through the experimental analysis and comparison of the artificial dataset and UCI dataset, the results show that the comprehensive clustering effect of DDPC is better than DPC, DBSCAN, DBC, and other algorithms.

1. Introduction

In recent years, data mining technology has become the main means to process a large amount of data and convert it into useful information. It is also a hot issue in artificial intelligence research [1, 2]. At present, it has been applied in many fields, including retail, recommendation, biological information, market analysis, and so on. Clustering is a common unsupervised learning method in the field of data mining [3]. It is also a research tool in the fields of computer vision and image segmentation. The purpose of the clustering algorithm is to divide the data into different clusters according to a certain feature or law [4]. The data with high similarity will be assigned to the same cluster, and the regions with low similarity will be assigned to different clusters [5]. Clustering algorithm also has many applications in the fields of computer science, mathematics, and the Internet of things [6, 7]. Taking wireless sensor as an example, the node distribution of wireless sensor is usually dense, and there is redundancy in the information transmission between nodes [8]. The clustering algorithm is used to cluster and process the sensor node data of different clusters to reduce the impact of information redundancy.

At present, the widely used basic clustering algorithms include the k-means algorithm [9], hierarchical clustering [10], density algorithm [11], and so on. K-means algorithm is the most classical clustering method. Through the random clustering center, the cluster allocation results and clustering center are iteratively optimized until the clustering center is no longer changed. Although the algorithm has a good effect on convex datasets [12], its limitation is that it is easy to fall into local optimization. The method of hierarchical clustering is to calculate the similarity [13, 14] between each node and other nodes at first and then merge the nodes one by one according to the similarity from high to low until the expected number of clusters is reached. DPC is a new density clustering algorithm. It determines the density of a single node by calculating the number of data in a certain range, selects the clustering center according to the density and data spacing, and assigns each low-density point to the nearest high-density point to realize clustering. DPC can get good clustering results not only on convex datasets but also on nonconvex datasets. However, the disadvantage is that it is greatly affected by the parameters [15]. It is hard to select the appropriate clustering center [16]. DPC cannot achieve good clustering results for clustering regions with discontinuous density [17]. Fast density peak clustering for large-scale data based on KNN [18] greatly reduces the complexity of determining local density peaks.

In recent years, there are many improvements for the DPC algorithm, which are mainly divided into the following aspects: in terms of clustering mode, a novel clustering algorithm based on directional propagation of cluster labels (DBC) [19] was proposed at the International Joint Conference of Neural Networks. DBC is a direction-based clustering method. By introducing the concepts of direction and angle, the clustering process is optimized, and the final clustering effect is better than that of DPC. However, the shortcoming of this algorithm is that it has many parameters and high sensitivity. In terms of formula improvement, an improvement of density peak clustering algorithm based on KNN and gravity [20] puts forward a new density formula, which makes the local density of sample points in dense and sparse areas more separable. In terms of centroid selection, a density peak clustering algorithm [21] based on feasible residual error was proposed, which realized semiautomatic clustering recognition and improved the iterative process of centroid selection of DPC. In 2021, a density peak clustering algorithm based on density decay graph [22] was proposed. The algorithm overcomes the shortcomings of the DPC algorithm, which needs to manually select the cluster center, and is greatly affected by chain reaction. The clustering process is realized by introducing a density decay graph. Although the clustering effect of this algorithm is better than that of DPC and other algorithms, there is no way to adjust the parameters dynamically according to the regional density, which is greatly limited by the parameters, and additional parameters are added based on the parameters of DPC algorithm. Even if the final clustering effect is good, the adjustment cost is high. In terms of algorithm combination, the proposed KNN-HDPC algorithm [23] makes the combination of KNN and DPC possible. In addition, the density peak clustering based on improved mutual K-Nearest Neighbor graph [24] solves the problem of poor clustering effect when different density regions are adjacent in DPC. In terms of noise point treatment, a novel density peak clustering algorithm based on squared residual error proposed by Parmar et al. [25] can help DPC solve the problem of noise point detection.

Through the analysis of clustering-related algorithms in recent years, most density clustering algorithms are based on the improvement of DPC, including accuracy improvement, algorithm combination, noise data processing, and so on. The main defects of the current algorithms are that it is hard to obtain the ideal cluster centers, the clustering process is complex, the requirements for parameter sensitivity are high, and the clustering effect on some real datasets is not ideal. In the future, reducing the parameter sensitivity of the clustering algorithm is a research direction.

The main contribution of this paper is to propose a dynamic density peak clustering algorithm based on K-Nearest Neighbor (DDPC) that can reduce the parameter sensitivity and choose cluster centers automatically. The calculation accuracy of DDPC is higher than that of the DPC algorithm. DDPC calculates the local density through the KNN distribution of each data and then divides each data into high-density data and low-density data according to the local density. For high-density data, the scanning distance is calculated according to the average distance of K-Nearest Neighbors. Using the feature that the scanning distance is self-adaptive with the regional density, the two mutually scanned data are classified into a cluster to reduce the sensitivity of parameters. For low-density data, after clustering high-density data, KNN method is used for clustering. We used NMI, ARI, Homogeneity (Homo), and *F*1 as the evaluation indexes in the experiment. The experimental results show that compared with the DPC algorithm, the performance evaluation index NMI of DDPC is improved by 0.23 on average. ARI increased by 0.24 on average, homogeneity increased by 0.21 on average, and *F*1 score increased by 0.19 on average.

2. Related Works

2.1. DPC. DPC is a density clustering algorithm that can remove noise points. It was presented in Science in 2014. At the same time, the clustering effect of the DPC is stable and will not be affected by randomness like the k-means. The core of the DPC mainly involves the following two points: (1) The density of cluster centers is the largest in clustering; (2) the distance between the highest density points in local areas is often large. Therefore, the DPC needs to first calculate the density value ρ_i of each data point x_i, which is determined by the dataset and truncation distance. Then calculate the distance δ_i between each data and its nearest higher density point according to the density value.

Definition 1. Local density: The local density ρ_i of data point x_i is calculated as follows:

For a given dataset $X = \{x_1, x_2, \ldots, x_n\}$, there are two ways to calculate the local density ρ_i: truncation function and Gaussian kernel function. The specific calculation methods are described below.

The truncation function is used to calculate ρ_i, and the calculation method is shown in the following formulas:

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

$$A(x) = \begin{cases} 0, & x < 0, \\ 1, & x \geqslant 0, \end{cases} \tag{2}$$

where $d_c > 0$ is the truncation distance, and the Euclidean distance between x_i and x_j is expressed as d_{ij}. The recommended truncation distance is $1\% - 2\%$ of the distance between all data points [11]. $A(x)$ is the truncation function. The value of the truncation function is determined by X. The value is 1 when $x < 0$ and 0 when $x \geqslant 0$. Therefore, the local density ρ_i represents the number of other data in the d_c range around data x_i.

Use Gaussian kernel function to calculate ρ_i, see formula asfollows:

$$\rho_i = \sum_j^N e^{-d_{ij}/d_c{}^2}. \tag{3}$$

Among them, d_{ij} and d_c have the same meaning as in Definition 1. The Gaussian kernel function is more suitable for the case of a small amount of data because it only produces a small probability conflict, which is not applicable when the amount of data is large.

Definition 2. Delta: The distance δi from the data point x_i to the high-density point x_j is calculated asfollows:

$$\delta_i = \begin{cases} \max\limits_{j:\, \rho_i > \rho_j} \left(d_{ij}\right), \\ \min\limits_{j:\, \rho_j > \rho_i} \left(d_{ij}\right). \end{cases} \tag{4}$$

According to the above formula, for a data point x_i, if its density is the maximum value, its corresponding δ_i is the farthest distance between it and other data points. Otherwise, δ_i is the distance between the data point x_i and the nearest higher density data point.

Therefore, for data points not in the cluster center, the δ_i will be small, on the contrary, the δ_i in the cluster center will be large. In particular, it should be noted that some data have a large δ_i, but the ρ_i is small, which indicates that there are little data around the data and are far from the cluster center. We identify such data as outliers. In cluster allocation, the cluster labels of noncentral points will be consistent with the cluster labels of the nearest higher density points.

2.2. KNN. K-Nearest Neighbor clustering [26] is a simple clustering algorithm. According to the previously entered parameter K, traverse its K-Nearest Neighbor cluster tag and assign the data to the cluster with the most cluster tags in the K-Nearest Neighbor of the data, and so on until all the data are assigned to the cluster tag.

The algorithm of K-Nearest Neighbor is as Algorithm 1.

2.3. Local Density Peak. To prevent the influence of density discontinuous data, we need to obtain the local density peak [27] in the regions where the data with different densities are located. In this way, even if all the densities in some regions are low, high-density points will still be generated for subsequent clustering [28]. We determine whether each data should be viewed as a high-density point by judging the density relationship between each data and its K adjacent data. Two parameters need to be introduced, one is the parameter K to determine the number of neighbors and the other is the ratio parameter R to determine whether it should be used as a high-density point. The local density peak in this region can be calculated by these two parameters.

Definition 3. KNN density: KNN density ρ_i of data point x_i is calculated as follows:

For a given dataset $X = \{x_1, x_2, \ldots, x_n\}$, where the K-Nearest Neighbor of point x_i is expressed as $N = \{x_1, x_2, \ldots, x_k\}$. When calculating the local density ρ_i of x_i, the average distance between its K-Nearest Neighbor is calculated. The larger the average distance is, the lower the point density is. On the contrary, the higher the point density is. The distance measurement here adopts Euclidean distance, which is more convenient for subsequent understanding. Here, the reciprocal of the calculation result is taken to make the result consistent with the corresponding relationship between the density. For details, see the following formula:

$$\rho_i = \frac{k}{\sum_j^k \left\| x_i - N_j \right\|_2}, \tag{5}$$

where ρ_i represents the local density of x_i, N_j represents the jth neighbor of the x_i, that is, the jth nearest neighbor, and K is a parameter used to represent the number of neighbors for each data search. Generally speaking, averaging the distance between each neighbor and the point can reflect the density of the point relative to the K points around the circumference. Therefore, the smaller the calculated average distance, the higher the density of the point. To make the result proportional to the density, it is expressed by the reciprocal.

By comparing the local density ρ_i of the x_i and its K-Nearest Neighbor, combined with the ratio parameter R, calculate whether x_i is a high-density point.

For a given data point x_i, compare its density with the surrounding neighbors through the ρ_i of the point and the local density $P = \{\rho_1, \rho_2, \ldots, \rho_k\}$ of its K neighbors, count the number of all local densities in the neighbors that are higher than the data, calculate a ratio with parameter K, and compare the ratio with ratio parameter R. If it is higher than ratio parameter R, the data are determined as a high-density point. First, it is necessary to compare the density between the point and each neighbor. For details, see the following formula:

$$l_j = \begin{cases} 0, \rho_i > P_j \\ 1, \rho_i \leq P_j \end{cases}, \tag{6}$$

where l_j represents the density comparison result between the point and its jth neighbor, and P is the density set of K neighbors of the data. See the following equation for the judgment of subsequent high-density points:

$$\begin{aligned} x_i \in C, \frac{\sum_j^k l_j}{k} > R, \\ x_i \in L, \frac{\sum_j^k l_j}{k} \leqslant R, \end{aligned} \tag{7}$$

where C is the high-density point set, L is the non-high-density point set, and R is the ratio parameter. It is not difficult to see from the formula that the relative size of the local density peak is determined by the size of R. If the ratio of the number of neighborhoods below the point density to K is greater than the ratio parameter R, the point is defined as a high-density point; otherwise, it is a low-density point. After all high-density points are distinguished through the above calculation process, the area composed of high-density points is called a high-density area, which is also a local density peak. Figure 1 is a schematic diagram of local density peaks on a hard dataset, in which red data points are high-density areas, and black data points are low-density areas.

```
Input: Dataset $D = \{x_1, x_2, x_3, \ldots, x_n\}$, $K$, Some tagged data $C = \{c_1, c_2, c_3, \ldots, c_t\}$
Output: Clusters $= \{c_1, c_2, \ldots, c_t\}$
//Loop to get the first $k$ nearest neighbors of each data and sort them.
for each data point $x$ in $D$ do
  for each data point $y$ in $D$ do
    Calculate the distance between $x$ and $y$
  end for
  Sort the data according to the distance from small to large: $N_x = \{n_1, n_2, n_3, \ldots, n_k\}$
  target = $\max_{c \in C} |c \cap N_x|$
  for each $c$ in $C$ do
    mark = $c_j \cap N_x$
    if mark == target then
      $x \in c_j$
    end if
  end for
end for
```

Algorithm 1: KNN Algorithm.

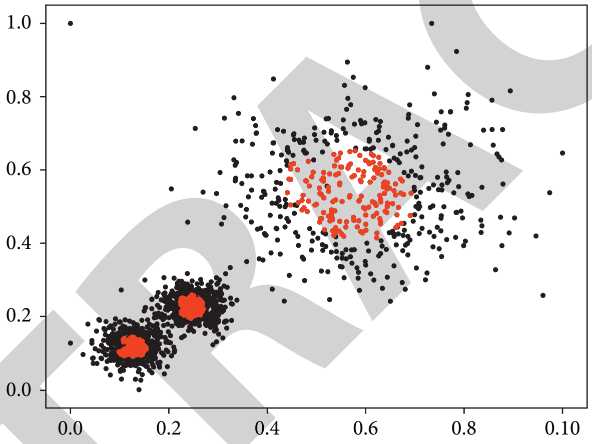


Figure 1: Local density peak on hard dataset.

3. DDPC

DDPC algorithm first obtains the high-density region of the dataset through the local density peak and then clusters the high-density region by dynamically adjusting the scanning distance by judging the density of each high-density region module. After the division of high-density regions is completed, the final division of low-density regions is realized by the KNN algorithm combined with cluster labels. The algorithm has two parameters: proximity parameter K and ratio parameter R. The size of K determines the number of neighbors of a single data point. The larger the K is, the more neighbors of each data, and the density distribution around each data becomes clearer. The clustering effect is more ideal for large-scale datasets, but it will increase the amount of calculation. The value of K should not be greater than the number of data in a cluster, which will cause unnecessary interaction between data in different clusters. The ratio parameter R determines the size of local density peak. The value range of R is [0, 1]. The larger the R is, the smaller the proportion of high-density regions; the distribution of high-density regions will be more discrete, and the number of clusters will be more. The smaller the R is, the larger the proportion of high-density regions is; the high-density regions tend to be a whole, and the number of clusters will be less.

First, we need to obtain the high-density region through the local density peak. Because the local density is adopted after obtaining the high-density regions, the average density difference between different high-density regions may be large. By using local density to dynamically adjust the scanning distance, the influence of density difference can be reduced.

The main step of clustering is to calculate the scanning distance. Only high-density points have the scanning distance, and the purpose of calculating the scanning distance is to dynamically adjust the clustering range according to the surrounding density. The specific calculation method is to calculate the average distance between the point and its K neighbors and take the distance as the scanning distance. The scanning distance of high-density points in high-density areas is short, and the scanning distance of high-density points in low-density areas is long.

Definition 4. Each high-density point has its own scanning distance, which is defined as follows:

$$s_i = \frac{\sum_j^k \left\| x_i - N_j \right\|_2}{k}. \tag{8}$$

Similar to formula (5), N_j represents the jth neighbor of the data point x_i. The scanning distance of x_i will change dynamically according to the density distribution of its K neighbors. Through the formula, we know that the scanning distance calculation method is the average Euclidean distance between the x_i and its K-Nearest Neighbors, that is, when the x_i is in the high-density region, the average distance between its K-Nearest Neighbors and the x_i is small, and the scanning distance is short. When the x_i is in the low-density area, the average distance between the K-Nearest Neighbor and the x_i is large, and the scanning distance is long. From Figure 2, we can observe the scanning distance of high-density area and low-density area when K is 14 (Algorithm 2).

After obtaining the scanning distance of each high-density point, carry out density transfer clustering according to the scanning distance of each high-density point. First, randomly select a high-density point without a cluster label, classify other high-density points within the scanning distance of the high-density point into a cluster, and scan the high-density points without a cluster label within the scanning distance of these high-density points. It is also classified as a cluster. All high-density points in the cluster are scanned until no new high-density points without cluster marks are found. Then, a new high-density point without a cluster label is randomly selected as a new cluster, and the above process is repeated until all high-density points have cluster labels.

Because the high-density points are often inside the cluster, and the scanning distance of each high-density point is strictly limited by its surrounding density, it is difficult for the high-density points between different clusters to be scanned through the scanning distance and merged into a cluster. This has the advantage that the clustering range will change dynamically with the internal density of the cluster, which effectively solves the problem of clustering in areas with discontinuous density; at the same time, different clusters will not be merged into one class. Another purpose of dynamic density peak clustering is to find the high-density regions of each cluster and cluster them to prepare for the final K-neighbor clustering.

The main defects of the current algorithm are two aspects: first, the data density distribution has a great impact on the calculation time of the adaptive algorithm. Second, for high-dimensional and large-scale data, the computational efficiency of the algorithm is not high. In the future, based on maintaining the existing accuracy, we will invest more energy to improve the calculation efficiency and reduce the calculation time of high-dimensional and large-scale data. It will take a lot of time, but I am confident.

Since the high-density points have been assigned cluster labels before, the cluster labels of these high-density points are also applied to K-Nearest Neighbor clustering as the clustering basis of low-density points. The clustering target of K-Nearest Neighbor clustering is low-density points. After a sufficient iterative process, all low-density points are also assigned cluster markers. So far, all data are assigned cluster markers. The pseudocode of the algorithm is shown in Figure 3. In the pseudocode, N_x is the sorted neighbor set, S is the average distance set of K-Nearest Neighbors, H is the high-density point set, and C_t is the unlabeled point set.

4. Experiments

Taking the clustering evaluation index as the standard, we test the proposed algorithm on the artificial dataset and UCI dataset, respectively. The comparison algorithms include the k-means algorithm, DBSCAN algorithm, DPC algorithm, and DDPC algorithm. The datasets adopt artificial datasets and real datasets. Artificial datasets include 2d-3c, three-circles, etc.; UCI datasets include vote, WDBC [29], zoo [30], vowel, seeds, ecoli [31], banknote, etc.

In this paper, all experimental parameters are selected by cyclic parameter adjustment, and the best result of NMI performance is retained as the final experimental result. Among the comparison algorithms selected in this paper, only the k-means algorithm is the meta-heuristic method. We have carried out 10 experiments on the same dataset and used the average value of the evaluation index as the experimental results of the K-means algorithm.

The evaluation indexes of clustering are Adjusted Rand Index (ARI), Normalized Mutual Information (NMI), Homogeneity Index (Homo), and F-Scores ($F1$). ARI is an adjusted RI, which has higher discrimination than RI. The value range of ARI is [−1, 1]. The closer the value is to 1, the better the clustering result is, and the closer it is to 0, the worse the clustering result is. The calculation formulas of RI and ARI are as follows:

$$\mathrm{RI} = \frac{a+b}{\begin{pmatrix} n \\ 2 \end{pmatrix}}, \tag{9}$$

$$\mathrm{ARI} = \frac{\mathrm{RI} - E[\mathrm{RI}]}{\max(RI) - E[\mathrm{RI}]}, \tag{10}$$

where C represents the actual classification, and K represents the clustering results. a is defined as the number of instance pairs divided into the same class in C and the same cluster in K. b is defined as the number of instance pairs divided into different categories in C and different clusters in K. For formula (9), n represents the total number of clusters, $\begin{pmatrix} n \\ 2 \end{pmatrix} = C_n^2 = n(n-1)/2$. Obviously, the value range of RI is [0, 1]. The larger the value, the better the clustering effect. For equation (10), max represents the maximum value and E represents the expectation.

NMI is an external indicator that measures the clustering effect by comparing the clustering results with "real" class labels; the value range of NMI is [0, 1]. The larger the value, the better the clustering effect.

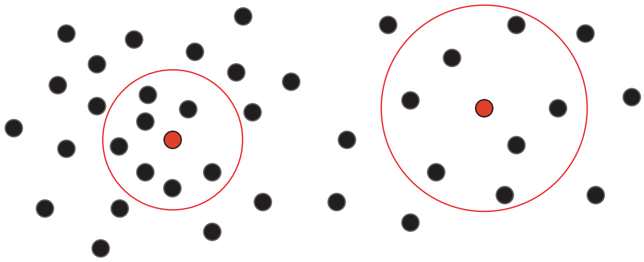


Figure 2: Scanning distance map.

```
Input: Dataset $D = \{x_1, x_2, x_3, \ldots, x_n\}$, $K, R$
Output: Clusters $= \{c_1, c_2, \ldots, c_t\}$
//Calculate the local density of each data.
for each data point $x$ in $D$do
  for each data point $y$ in $D$do
    Calculate the distance between $x$ and $y$
  end for
  Sort the first $K$ data according to the distance from small to large: $N_x = \{n_1, n_2, n_3, \ldots, n_k\}$
  Calculate the average distance $l_x$ from each neighbor $N_x$
end for
The average distance matrix of $K$ neighbors of each node (scanning distance) is obtained: $S = \{s_1, s_2, \ldots, s_n\}$
//The adaptive adjustment range is determined according to parameters $K$//and $R$.
for each $x$ in $D$do
  Calculate the number of neighbors whose average distance is smaller than the node: $m$
  if$m > (1 - R) * K$then
    $x$ is a high-density point: $x \in H$
  end if
end for
Int $t = 1$
//Adaptive clustering
for each $x$ in $H$do
  If$x$ has no cluster label then
    $x \in c_t$
  end if
  for each $y$ in $H$do
    for each $z$ in $c_t$do
      if the distance between $y$ and $z$ is less than $s_y$ or $s_z$then
        If$y$ has cluster label then
          Change all $y$'s cluster labels to $c_t$
        else
          $y \in c_t$
        end if
        break
      end if
    end for
  end for
  $t$++
end for
For the points without cluster label, KNN algorithm is used for clustering
```

Algorithm 2: DDPC Algorithm.

$$\mathrm{NMI} = \frac{\sum_{i=1}^{k(C)} \sum_{j=1}^{k(T)} n_{i,j} \log\left(n * n_{i,j} / n_i * n_j\right)}{\sqrt{\left(\sum_{i=1}^{k(C)} n_i \log n_i / n\right)\left(\sum_{j=1}^{k(T)} n_j \log n_j / n\right)}}, \tag{11}$$

where $K(C)$ is the number of clusters in the clustering result, $K(T)$ is the number of clusters in the real clustering result, n_i is the number of samples in cluster i, n_j is the number of samples in cluster j, $n_{i,j}$ is the number of samples between the samples belonging to cluster i in the clustering result C and the samples belonging to cluster j in the real clustering result T, and n is the total number of samples in the dataset.

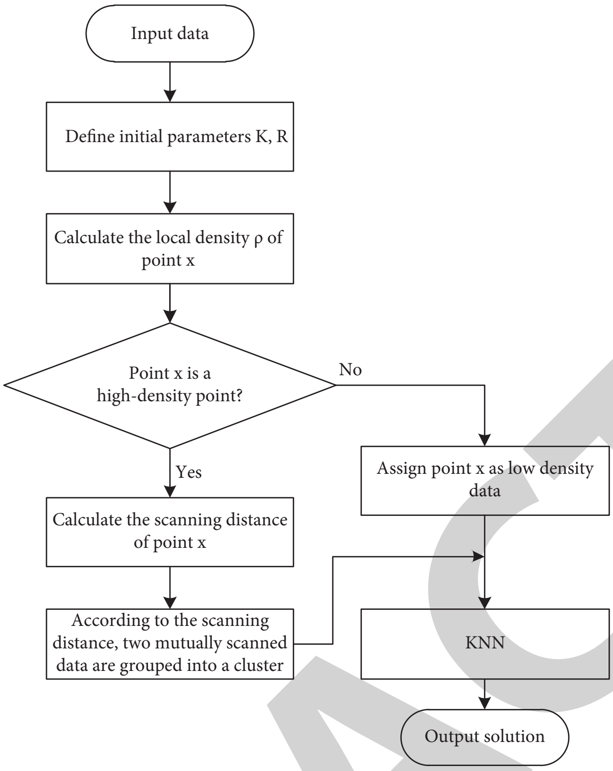

Input data
Define initial parameters K, R
Calculate the local density ρ of point x
Point x is a high-density point?
No
Yes
Assign point x as low density data
Calculate the scanning distance of point x
According to the scanning distance, two mutually scanned data are grouped into a cluster
KNN
Output solution

Figure 3: Flow chart of DDPC.

The value of homogeneity depends on the degree to which each cluster contains only members of a single class; the value range of homogeneity is [0, 1]. The larger the value, the better the clustering effect. Its calculation formula is asfollows:

$$\text{Homogeneity} = 1 - \frac{H(C|K)}{H(C)}, \tag{12}$$

$$H(C|K) = -\sum_{c=1}^{|C|}\sum_{k=1}^{|K|}\frac{n_{c,k}}{n} * \log\left(\frac{n_{c,k}}{n}\right), \tag{13}$$

$$H(C) = -\sum_{c=1}^{|C|}\frac{n_c}{n} * \log\left(\frac{n_c}{n}\right), \tag{14}$$

where n is the total number of samples, n_c and n_k are the number of samples belonging to class C and class K, respectively, and $n_{c,k}$ is the number of samples divided from class C to class K.

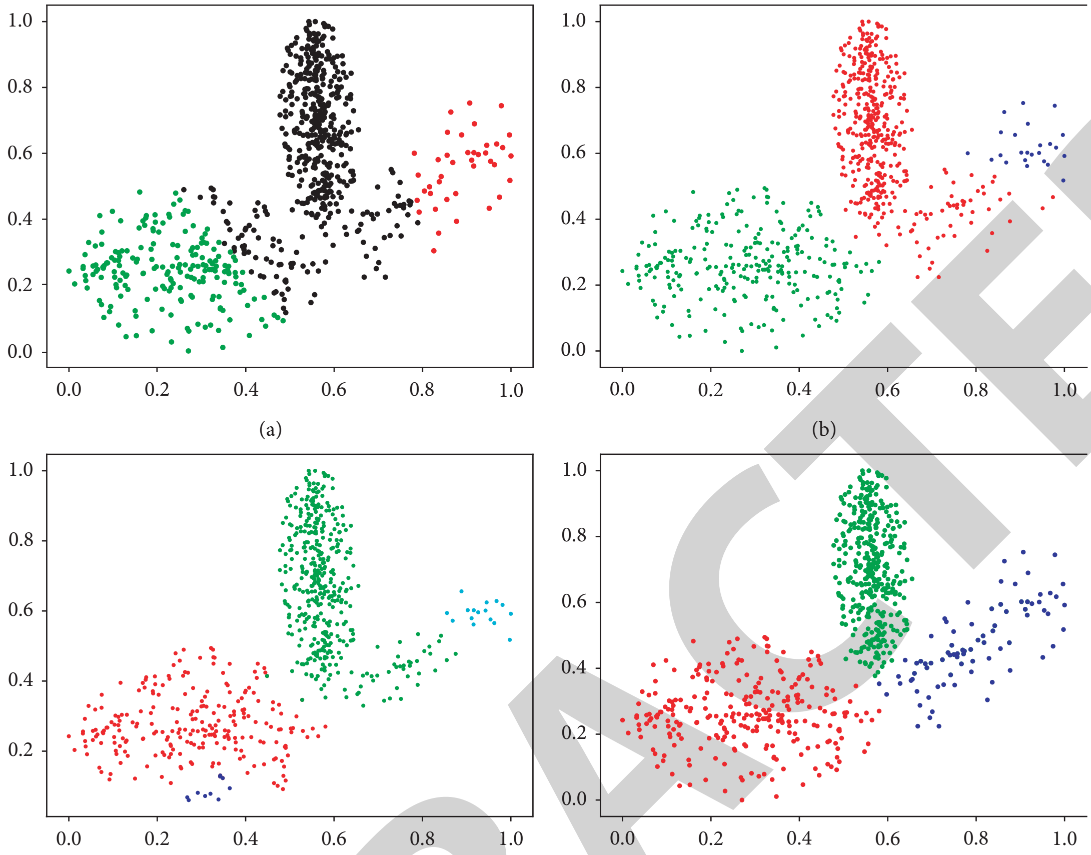
As a comprehensive index, F-scores are to balance the impact of accuracy, recall, and comprehensively evaluate a classifier; the value range of homogeneity is [0, 1]. The larger the value, the better the clustering effect. Its formula is as follows:

$$F1 = \frac{2\text{TP}}{2\text{TP} + \text{FP} + \text{FN}}. \tag{15}$$

TP refers to the data that determine the attribution, and the actual attribution is exactly the same; FP refers to the data that determine the attribution and does not belong, and FN refers to the data that determine the nonattribution but does belong.

4.1. Artificial Dataset. We use k-means, DPC, and DBSCAN algorithms as comparison objects, respectively. Figures 4–7 show the clustering effect of each algorithm on 2d-3c dataset, grid.orig dataset, Jain dataset, and threecircles dataset, respectively. Due to space constraints, the corresponding evaluation indicators of the other six datasets are shown in Table 1. Experiments show that DDPC algorithm performs well on all datasets and is better than DPC algorithm. The details of the dataset are shown in Table 2.

Experimental results show that the DDPC algorithm proposed in this paper can achieve good clustering results on various difficult datasets in different density regions. At the same time, the DDPC algorithm can also achieve good clustering results for some nonconvex datasets. It can be seen from Figures 4 and 5 that due to the limitation of parameters in other algorithms, a single parameter cannot solve the clustering problem of different density regions, resulting in a poor clustering effect. In Figure 6, the DBSCAN algorithm falls into local optimization and cannot cluster accurately. In Figure 7, because the density relationship of the dataset does not increase significantly, the DPC algorithm cannot cluster correctly due to the limitation of the density increasing condition. K-means algorithm cannot achieve a good clustering effect on nonconvex datasets. Therefore, it can be seen that the DDPC algorithm can achieve satisfactory clustering



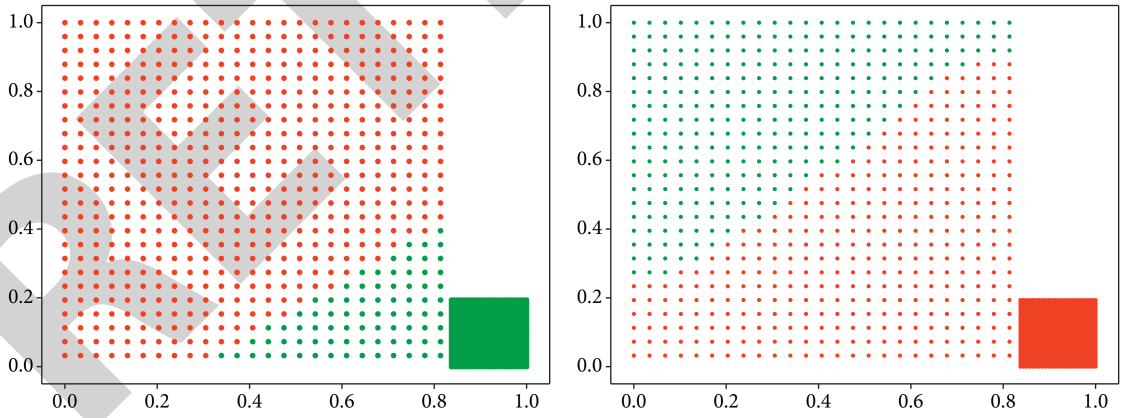
(a)

(b)

(c)

(d)

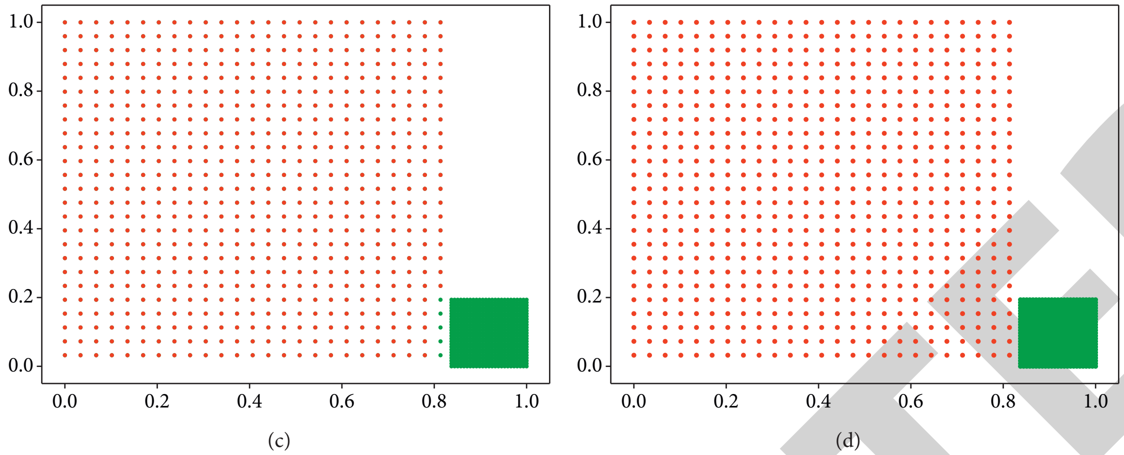
Figure 4: Clustering results on dataset 2d-3c. (a) k-means. (b) DPC. (c) DBSCAN. (d) DDPC.



(a)

(b)

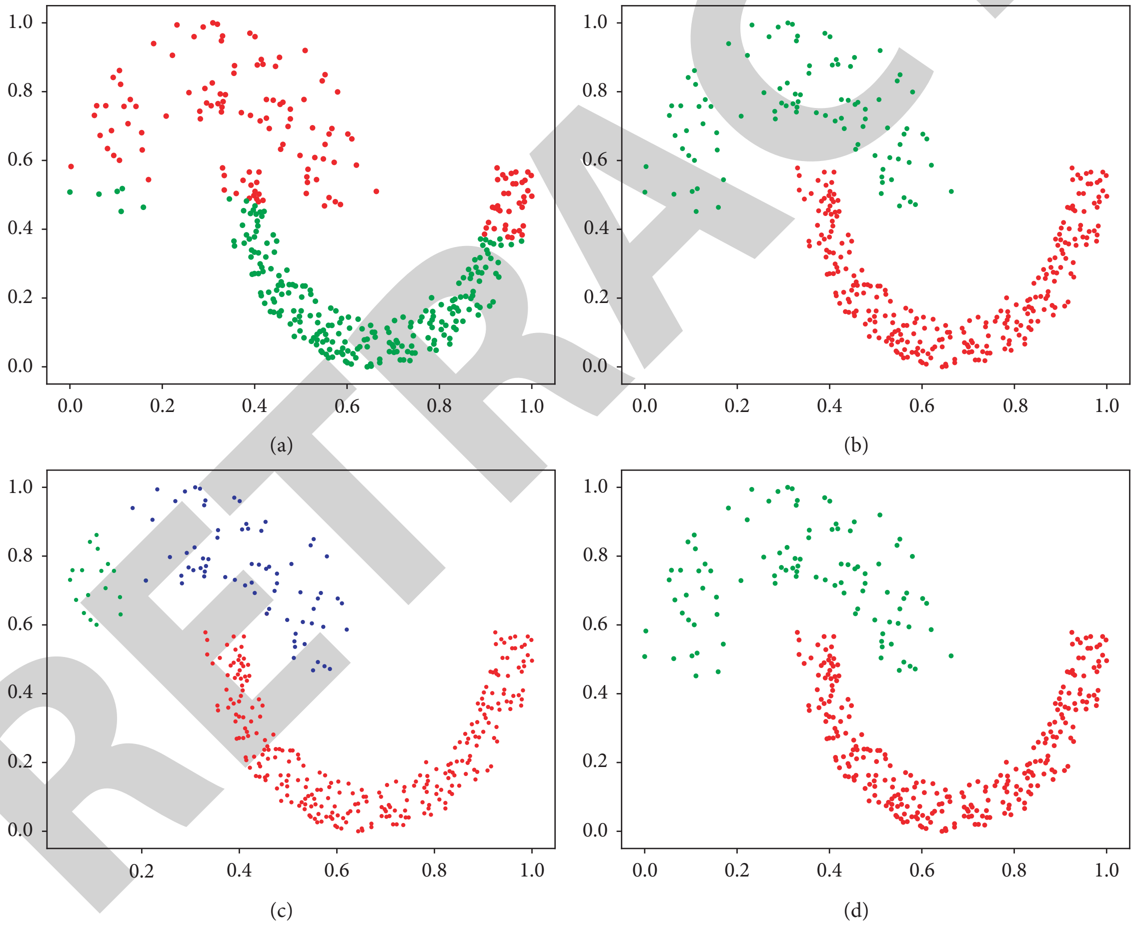
Figure 5: Continued.



(c)

(d)

FIGURE 5: Clustering results on dataset grid.orig. (a) k-means. (b) DPC. (c) DBSCAN. (d) DDPC.



(a)

(b)

(c)

(d)

FIGURE 6: Clustering results on dataset Jain. (a) k-means. (b) DPC. (c) DBSCAN. (d) DDPC.

results whether it is a dataset with uneven density distribution or a nonconvex dataset, which cannot be done by other comparison algorithms.

In terms of parameter sensitivity, DPC and DDPC are tested on the flame dataset. To accurately test the sensitivity of each parameter, based on the ARI evaluation index, we set one of the parameters as the ideal value and analyze the sensitivity of the parameter by observing the impact of the changes of other parameters on the clustering effect. The experimental results are shown in Figure 8 and Tables 3–6. The observation results show that DDPC is superior to DPC in parameter sensitivity.

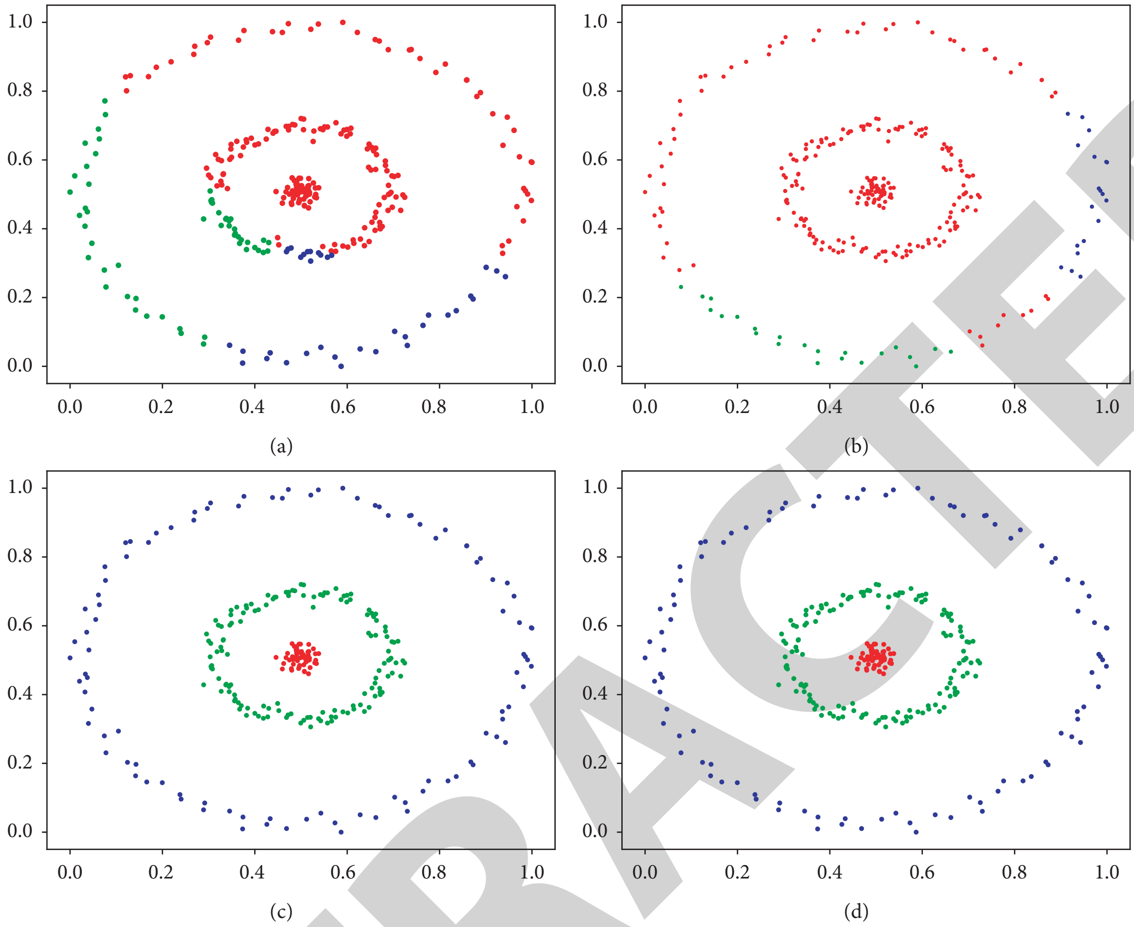


Figure 7: Clustering results on dataset threecircles. (a) k-means. (b) DPC. (c) DBSCAN. (d) DDPC.

Table 1: Evaluation index of four algorithms on artificial datasets.

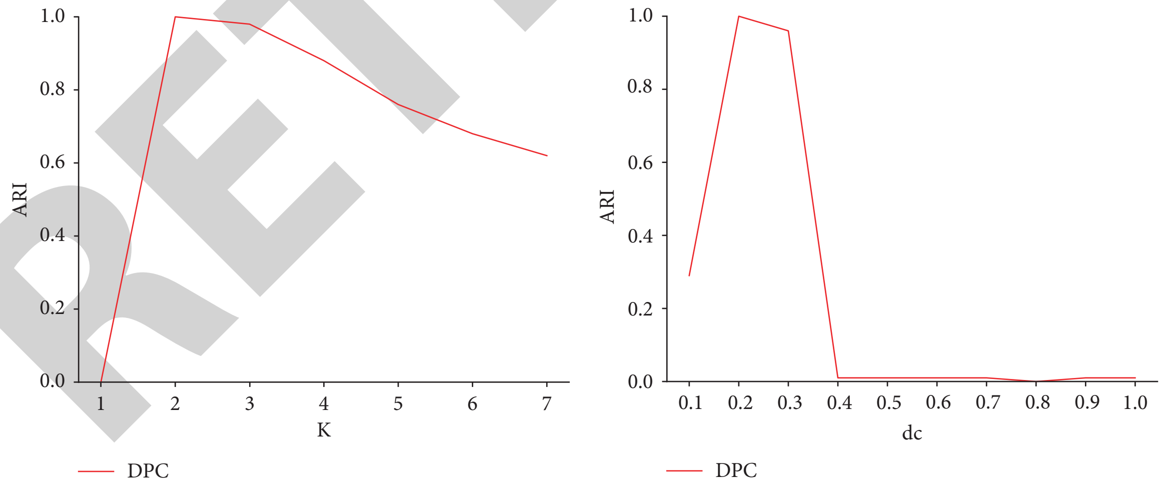
Dataset	Algorithm	ARI	NMI	Homo	*F*1
2d-3c	k-means	0.5898 ± 0.13	0.5733 ± 0.12	0.6056 ± 0.09	0.6619 ± 0.13
	DPC	0.8309	0.8178	**0.9562**	0.9217
	DBSCAN	0.8147	0.7753	0.7410	0.8328
	DDPC	**0.8820**	**0.8578**	0.9007	**0.9236**
grid.orig	k-means	0.7601 ± 0.06	0.7164 ± 0.04	0.7121 ± 0.03	0.7367 ± 0.08
	DPC	0.1988	0.3059	0.3526	0.3626
	DBSCAN	0.9840	0.9663	0.9663	0.9915
	DDPC	**1.0**	**1.0**	**1.0**	**1.0**
Jain	k-means	0.4527 ± 0.09	0.3900 ± 0.07	0.4223 ± 0.08	0.4618 ± 0.11
	DPC	1.0	1.0	1.0	1.0
	DBSCAN	0.9262	0.8241	0.9690	0.9329
	DDPC	**1.0**	**1.0**	**1.0**	**1.0**
Threecircles	k-means	0.0261 ± 0.02	0.1214 ± 0.01	0.1082 ± 0.03	0.1339 ± 0.02
	DPC	0.0964	0.2273	0.3541	0.2846
	DBSCAN	1.0	1.0	1.0	1.0
	DDPC	**1.0**	**1.0**	**1.0**	**1.0**
Flame	k-means	0.7323 ± 0.03	0.7154 ± 0.06	0.6987 ± 0.04	0.7432 ± 0.05
	DPC	1.0	1.0	1.0	1.0
	DBSCAN	0.9865	0.9865	0.9999	0.9873
	DDPC	**1.0**	**1.0**	**1.0**	**1.0**
Spiral	k-means	0.0164 ± 0.01	0.0321 ± 0.01	0.0289 ± 0.02	0.0236 ± 0.01
	DPC	1.0	1.0	1.0	1.0
	DBSCAN	1.0	1.0	1.0	1.0
	DDPC	**1.0**	**1.0**	**1.0**	**1.0**

Table 1: Continued.

Dataset	Algorithm	ARI	NMI	Homo	*F*1
Aggregation	k-means	0.7615 ± 0.03	0.7344 ± 0.02	0.7618 ± 0.06	0.7742 ± 0.05
	DPC	**1.0**	**1.0**	**1.0**	**1.0**
	DBSCAN	0.7221	0.7368	0.7639	0.8234
	DDPC	0.9948	0.9915	0.9923	0.9974
Lineblobs	k-means	0.5293 ± 0.08	0.6135 ± 0.11	0.6177 ± 0.07	0.6210 ± 0.12
	DPC	0.6449	0.6933	0.6860	0.7128
	DBSCAN	1.0	1.0	1.0	1.0
	DDPC	**1.0**	**1.0**	**1.0**	**1.0**
R15	k-means	0.8556 ± 0.03	0.8259 ± 0.03	0.8814 ± 0.01	0.8765 ± 0.02
	DPC	0.9715	0.9833	0.9782	0.9867
	DBSCAN	0.9364	0.9269	0.9516	0.9398
	DDPC	**0.9891**	**0.9913**	**0.9913**	**0.9950**
Sspiral	k-means	0.0237 ± 0.02	0.0231 ± 0.01	0.0216 ± 0.01	0.0253 ± 0.01
	DPC	1.0	1.0	1.0	1.0
	DBSCAN	1.0	1.0	1.0	1.0
	DDPC	**1.0**	**1.0**	**1.0**	**1.0**

Table 2: Artificial dataset.

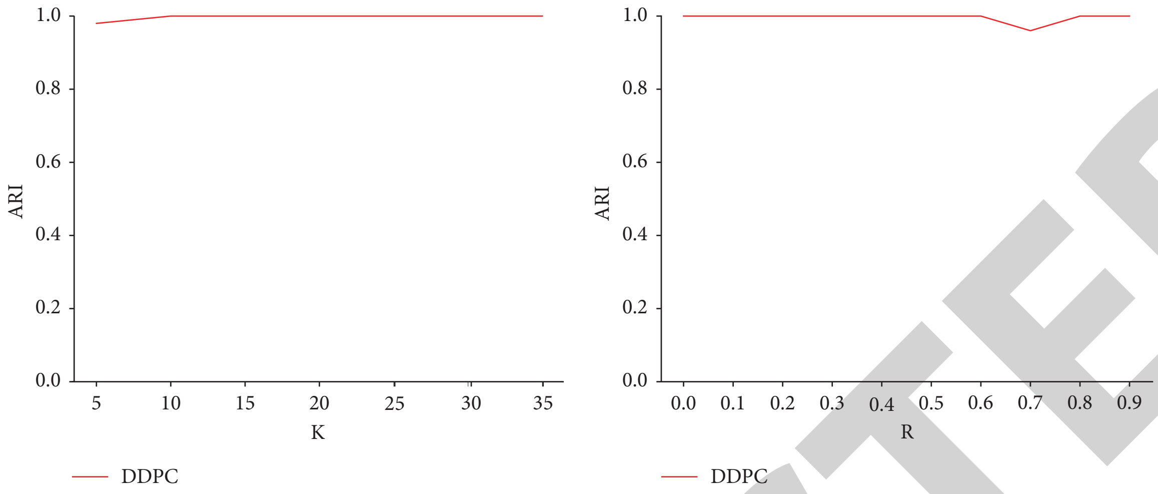
Dataset	Instance	Dimension	Cluster
2d-3c	715	2	3
Grid	1250	2	2
Jain	373	2	2
Threecircles	299	2	3
Flame	219	2	2
Spiral	312	2	3
Aggregation	788	2	7
Lineblobs	266	2	3
R15	600	2	15
Sspiral	944	2	2



(a)

(b)

Figure 8: Continued.



(c)

(d)

Figure 8: Experimental results of parameter sensitivity on flame dataset. (a) DPC, (b) DPC, (c) DDPC, (d) DDPC.

Table 3: Experimental results of DPC on flame dataset when K is optimal.

Dc	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1.0
ARI	0.29	1.0	0.96	0.01	0.01	0.01	0.01	0.0	0.01	0.01

Table 4: Experimental results of DPC on flame dataset when dc is optimal.

K	1	2	3	4	5	6	7
ARI	0	1.0	0.98	0.88	0.76	0.68	0.62

Table 5: Experimental results of DDPC on flame dataset when K is optimal.

R	0.0	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9
ARI	1.0	1.0	1.0	1.0	1.0	1.0	1.0	0.96	1.0	1.0

Table 6: Experimental results of DDPC on flame dataset when R is optimal.

K	5	10	15	20	25	30	35
ARI	0.98	1.0	1.0	1.0	1.0	1.0	1.0

4.2. UCI Dataset. DDPC algorithm shows better clustering results on artificial datasets. To further verify its clustering performance, it also needs to be verified on the real datasets. Considering that the k-means algorithm has been proposed for a long time, this paper uses DBC (a novel clustering algorithm based on directional propagation of cluster labels) algorithm instead of K-means algorithm to compare on UCI dataset. After comparison, the comprehensive experimental results on a variety of different UCI datasets are better than DBC and other algorithms. UCI datasets are shown in Table 7.

In the UCI datasets, because it is difficult to visualize a high-dimensional dataset; the clustering evaluation indexes ARI, NMI, and homogeneity are compared. Table 8 shows the evaluation indexes of each clustering algorithm. Although in the vowel dataset, the ARI of the DPC algorithm is slightly higher than that of the DDPC algorithm, and in the banknote dataset, the NMI of the DPC algorithm is slightly higher than that of the DDPC algorithm. However, in general, DDPC performs significantly better than other clustering algorithms on UCI datasets, and the clustering effect is the best. The second is the DBC algorithm and DPC algorithm. The clustering effect of the DBSCAN algorithm on the UCI dataset is the least ideal.

For DDPC, each data determine that the time complexity of the surrounding K neighbors is $O(n^2)$, the time complexity of calculating the local density and scanning distance is $O(n)$, the time complexity of adaptive clustering is $O(n * k)$, and the overall time complexity of synthesizing the above information is $O(n^2)$. The time complexity of other algorithms compared in the experimental part is shown in Table 9.

Table 7: UCI datasets.

Dataset	Instance	Dimension	Cluster
Vote	435	16	2
WDBC	569	30	2
Vowel	871	3	6
Zoo	101	16	7
Seeds	210	7	3
Ecoli	336	7	8
Banknote	1372	4	2
Dermatology	358	34	6
Segment	2310	18	7
Pendigits	10992	16	10

Table 8: Evaluation index of four algorithms on UCI datasets.

Dataset	Algorithm	ARI	NMI	Homo	*F*1
Vote	DBSCAN	0.4480	0.3977	0.5034	0.6106
	DPC	0.0036	0.0031	0.0270	0.0107
	DBC	0.5709	0.4942	0.5030	0.5236
	DDPC	**0.5850**	**0.5210**	**0.5116**	**0.8827**
WDBC	DBSCAN	0.4661	0.3790	0.3868	0.5131
	DPC	0.4869	0.5028	0.5112	0.6628
	DBC	0.5883	0.5258	0.6656	0.5669
	DDPC	**0.7668**	**0.6822**	**0.6997**	**0.9384**
Vowel	DBSCAN	0.4170	0.5317	0.4902	0.4913
	DPC	**0.5231**	0.5977	0.6285	**0.6199**
	DBC	0.4221	0.5576	0.5242	0.4219
	DDPC	0.5102	**0.6156**	**0.6537**	0.4948
Zoo	DBSCAN	0.1457	0.6161	0.6285	0.5266
	DPC	0.3056	0.4222	0.3264	0.6953
	DBC	0.8801	0.8545	0.7696	0.8752
	DDPC	**0.8918**	**0.8621**	**0.9275**	**0.8811**
Seeds	DBSCAN	0.0009	0.3426	0.4239	0.6137
	DPC	0.7636	0.7219	0.7169	0.9127
	DBC	0.7869	0.7398	0.7397	0.8623
	DDPC	**0.7916**	**0.7784**	**0.7819**	**0.9238**
Ecoli	DBSCAN	0.1459	0.6161	0.6285	0.4523
	DPC	0.4516	0.5402	0.7298	0.6404
	DBC	0.7165	0.6571	0.5580	0.7653
	DDPC	**0.7398**	**0.7067**	**0.7705**	**0.8065**
Banknote	DBSCAN	0.0752	0.2958	0.0056	0.1134
	DPC	0.8935	**0.9316**	0.8761	0.9516
	DBC	0.8298	0.7467	0.7427	0.8819
	DDPC	**0.9538**	0.9099	**0.9091**	**0.9883**
Dermatology	DBSCAN	0.4151	0.6205	0.6484	0.5761
	DPC	0.4269	0.3304	0.4105	0.4429
	DBC	0.7871	0.8486	0.7879	**0.8626**
	DDPC	**0.8459**	**0.9082**	**0.9703**	0.6675
Segment	DBSCAN	0.2446	0.5525	0.4235	0.4695
	DPC	0.2402	0.5262	0.4010	0.4968
	DBC	0.3626	0.6109	0.6260	0.6371
	DDPC	**0.4965**	**0.6576**	**0.8144**	**0.7032**
Pendigits	DBSCAN	0.4151	0.6205	0.6484	0.5753
	DPC	0.5542	0.7243	0.6938	0.7289
	DBC	0.5818	0.6617	0.6704	0.7143
	DDPC	**0.6328**	**0.7775**	**0.9618**	**0.7695**

Table 9: Time complexity of each algorithm.

Algorithm	Time complexity
k-means	$O(n * k * t)$
DPC	$O(n^2)$
DBSCAN	$O(n^2)$
DBC	$O(n^2)$
DDPC	$O(n^2)$

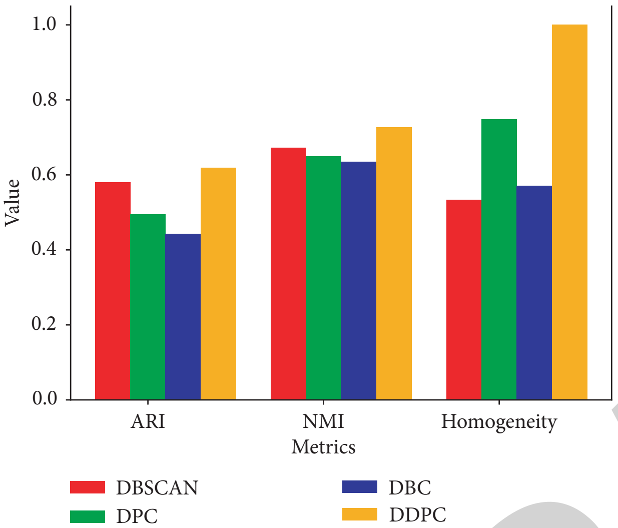


Figure 9: Clustering results on dataset d1p07m.

4.3. Application. Wireless sensors are widely used in the Internet of things. The three functions of data acquisition, processing, and transmission are realized through a sensor network. Due to the large number and complex distribution of nodes in sensor networks, clustering can reduce the cost of information transmission between nodes. At the same time, some clustering algorithms can also eliminate the influence of noise data and improve experimental accuracy. Figure 9 shows the difference in clustering accuracy between the DDPC algorithm and other clustering algorithms in the wireless sensor network dataset. The higher the clustering accuracy, the smaller the difference from the actual situation and the better the effect.

5. Conclusion

A dynamic density peak clustering algorithm is proposed, which effectively solves the problem that the same parameter cannot adapt to different density regions in the process of density clustering. However, due to the limitations of adaptive processing, the main defects of the algorithm are two aspects: first, the adaptive algorithm is greatly affected by the dataset, resulting in the actual operation time being difficult to estimate, and the operation time of the dataset with a small amount of data may be longer than that of the dataset with a large amount of data. Second, for high-dimensional and large-scale data, the calculation efficiency of this algorithm is not high and may take a long time, but the calculation accuracy is greatly improved. In addition, we will try our best to further reduce the number of parameters in the future, but this needs to be realized by continuously optimizing the adaptive algorithm. In the experimental process, we found that the algorithm also has good performance on some datasets that are not suitable for density clustering, and the artificial datasets are completely consistent with the clustering labels. In some UCI datasets, although the performance of a single evaluation index is low, it is usually higher than other related algorithms. We also apply the algorithm to wireless sensor networks. The relative evaluation index of the application result is higher than that of the comparison algorithm, and the expected effect is achieved.

In the future, on the basis of maintaining the existing accuracy, we will spend more energy to improve the computing efficiency and reduce the computing time of high-dimensional and large-scale data. Obviously, it takes a lot of time, but I am confident.

Data Availability

The data used in the report can be obtained from [url = "http://archive.ics.uci.edu/ml"], and these data are referenced at relevant positions in the body.

Conflicts of Interest

The authors declare that they have no conflict of interest.

Acknowledgments

The financial support for this project is provided by the National Natural Science Foundation of China [61962054].

References

[1] Z. H. Zhou and J. Yuan, "Nec4.5: Neural ensemble based c4.5," *IEEE Transactions on Knowledge and Data Engineering*, vol. 16, no. 6, pp. 770–773, 2004.

[2] T. Poggio, "A theory of networks for approximation and learning," *A.i.memo*, vol. 1140, no. 9, pp. 1481–1497, 1989.

[3] A. K. Jain and R. C. Dubes, *Algorithms for Clustering Data*, Prentice-Hall, Hoboken, New Jersey, 1988.

[4] R. Krishnapuram and M. J. Keller and, "A possibilistic approach to clustering," *IEEE Transactions on Fuzzy Systems*, vol. 1, 1993.

[5] L. Kaufman and P. J. Rousseeuw, *Finding Groups in Data: An Introduction to Cluster Analysis*, Wiley, Hoboken, New Jersey, 2005.

[6] M. R. Anderberg, *Cluster Analysis for Applications*, pp. 347–353, Academic Press, New York, NY, USA, 1973.

[7] H. Kopetz, "Internet of things," *Real-Time Systems*, 2011.

[8] C. Lntanagonwiwat, "Directed Diffusion for Wireless Sensor Networking," *IEEE/ACM Transactions on Networking*, ieee/acm trans netw, vol. 11, , 2003.

[9] J. B. Macqueen, "Some methods for classification and analysis of multivariate observations," in *Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability*, California, June 1967.

[10] G. Roy and D'Andrade, "Hierarchical clustering," *Psychometrika*, vol. 43, no. 1, pp. 59–67, 2011.

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

[12] S. I. Gass and C. M. Harris, "Convex Set," *Encyclopedia of Operations Research & Management Science*, Springer, Berlin/Heidelberg, Germany, 2001.

[13] A. Physica, P. A. Boris, and S. C. Jia, "Features of Similarity," *Psychological Review*, vol. 84, 1977.

[14] G. M. Mimmack, S. J. Galpin, and J. S. Galpin, "Choice of distance matrices in cluster analysis: defining regions," *Journal of Climate*, vol. 14, no. 12, pp. 2790–2797, 2001.

[15] P. Mehrmohammadi, M. Hatami, and P. Moradi, "A density peaks clustering method based on mutual knn graph and shortest path," in *Proceedings of the 2020 28th Iranian Conference on Electrical Engineering (ICEE)*, Tabriz, Iran, August 2020.

[16] D. Wang, "Redpc: A residual error-based density peak clustering algorithm," *Neurocomputing*, vol. 348, pp. 82–96, 2019.

[17] G. S. Kolli and K. Kolli, "Fuzzy k-means clustering with fast density peak clustering on multivariate kernel estimator with evolutionary multimodal optimization clusters on a large dataset," *Multimedia Tools and Applications*, vol. 80, no. 3, pp. 4769–4787, 2021.

[18] Y. Chen, X. Hu, W. Fan, L. Shen, and H. Li, "Fast density peak clustering for large scale data based on KNN," *Knowledge-Based Systems*, 2019.

[19] N. Xiao, K. Li, X. Zhou, and K. Li, "A novel clustering algorithm based on directional propagation of cluster labels," in *Proceedings of the 2019 International Joint Conference on Neural Networks (IJCNN)*, Budapest, Hungary, July, 2019.

[20] J. Sun and G. Liu, "An improvement of density peaks clustering algorithm based on knn and gravitation," in *Proceedings of the 2021 4th International Conference on Intelligent Autonomous Systems (ICoIAS)*, pp. 234–239, Wuhan, China, May 2021.

[21] M. D. Parmar, W. Pang, D. Hao et al., "Fredpc: a feasible residual error-based density peak clustering algorithm with the fragment merging strategy," *IEEE Access*, vol. 7, Article ID 89789, 2019.

[22] A. Zz, A. Qz, and Z. B. Fan, "Density Decay Graph–Based Density Peak Clustering," *Knowledge-Based Systems*, vol. 224, 2021.

[23] L. Wang, W. Zhou, H. Wang, M. Han, and X. Han, "A novel density peaks clustering halo node assignment method based on k-nearest neighbor theory," *IEEE Access*, vol. 7, Article ID 174380, 2019.

[24] J. C. Fan, P. L. Ge, and L. Ge, "Mk-nng-dpc: density peaks clustering based on improved mutual k-nearest-neighbor graph," *International Journal of Machine Learning and Cybernetics*, vol. 11, no. 6, pp. 1179–1195, 2019.

[25] M. Parmar, W. Di, A. H. Tan, C. Miao, and Z. You, "A novel density peak clustering algorithm based on squared residual error," in *Proceedings of the 2017 International Conference on Security, Pattern Analysis, and Cybernetics (SPAC)*, Shenzhen, China, December 2017.

[26] L. E. Peterson, "K-nearest neighbor," *Scholarpedia*, vol. 4, no. 2, p. 1883, 2009.

[27] Z. Guo, T. Huang, Z. Cai, and W. Zhu, *A New Local Density for Density Peak Clustering*, Springer, Berlin/Heidelberg, Germany, 2018.

[28] M. Pascual, M. Roy, F. Flierl, and G. Flierl, "Cluster size distributions: signatures of self-organization in spatial ecologies," *Philosophical Transactions of the Royal Society of London Series B Biological Sciences*, vol. 357, no. 1421, pp. 657–666, 2002.

[29] O. L. Mangasarian and W. H. Wolberg, "Cancer Diagnosis via Linear Programming," *Operations Research*, vol. 43, 1990.

[30] K. Bache and M. Lichman, "UCI Machine Learning Repository," 2013, https://archive.ics.uci.edu/ml/index.php.

[31] P. Horton, "A probabilistic classification system for predicting the cellular localization sites of proteins," *Proceedings International Conference on Intelligent Systems for Molecular Biology*, vol. 4, 1996.