ConDPC: Data Connectivity-Based Density Peak Clustering

Yujuan Zou 1,2 and Zhijian Wang 1,*

1 College of Computer and Information, Hohai University, Nanjing 211100, China
2 School of Information Engineering, Jiangsu Maritime Institute, Nanjing 211199, China
* Correspondence: zhjwang@hhu.edu.cn

Abstract: As a relatively novel density-based clustering algorithm, Density peak clustering (DPC) has been widely studied in recent years. DPC sorts all points in descending order of local density and finds neighbors for each point in turn to assign all points to the appropriate clusters. The algorithm is simple and effective but has some limitations in applicable scenarios. If the density difference between clusters is large or the data distribution is in a nested structure, the clustering effect of this algorithm is poor. This study incorporates the idea of connectivity into the original algorithm and proposes an improved density peak clustering algorithm ConDPC. ConDPC modifies the strategy of obtaining clustering center points and assigning neighbors and improves the clustering accuracy of the original density peak clustering algorithm. In this study, clustering comparison experiments were conducted on synthetic data sets and real-world data sets. The compared algorithms include original DPC, DBSCAN, K-means and two improved algorithms over DPC. The comparison results prove the effectiveness of ConDPC.

Keywords: clustering; connectivity; Euclidean distance; neighbor distance; density difference

1. Introduction

Clustering aims at partitioning a collection of objects into different subgroups [1]. Objects are grouped according to their similarity, so the objects in the same subgroup are highly similar [2]. Clustering has been applied in wide areas such as pattern recognition, community detection, image segmentation, trajectory analysis, fault diagnosis, and so on [3–7].

The density-based clustering algorithm is a kind of classical clustering algorithm [8–11]. Density-based clustering methods can discover clusters of various shapes and sizes [12]. DBSCAN [8] is the most typical representative algorithm in this kind of method. It proposes concepts such as density-reachable to construct densely connected clusters. However, when clusters are distributed in markedly various densities, DBSCAN may obtain bad clustering results. DPC [13] is a relatively novel density-based clustering algorithm, which can obtain the cluster centers semi-automatically with the help of decision graphs. However, when the DPC algorithm deals with data with large density differences or manifold structures, it is easy to produce wrong cluster centers or wrong neighbor assignments.

Based on the DPC algorithm, many scholars have proposed improved algorithms [14–20]. Du et al. [14] proposed two KNN-based clustering algorithms and incorporated PCA into them. Xie et al. [15] modified the calculation method of local density and the neighbor assignment mechanism. Abdulrahman et al. [16] constructed cluster backbones to better distinguish different clusters. The above algorithms use KNN to calculate the local density, but the clustering error problem of large density difference and nested structure has not been solved. Du et al. [17] proposed a density-adaptive distance calculation method to solve the problem of poor clustering in manifold structure, but the time complexity of the algorithm is high. Wang et al. [18] proposed multi-center clustering based on a hierarchical approach to solve the problem of poor multi-center structure identification. However, the algorithm has four input parameters which make the algorithm difficult to control.
This study integrates connectivity into the original DPC algorithm to improve the clustering effect. The primary contributions of this study are as follows.

1. Constructing connected groups based on the idea of connectivity. Points in the same group are connectable to each other, whereas points in different groups are not.
2. Adjusting the calculation method of neighbor distance based on the idea of connectivity to improve the accuracy of determining the cluster center.
3. Adjusting the neighbor allocation strategy based on the idea of connectivity to improve the accuracy of clustering.

The rest of this paper is organized as follows. We provide an overview of the DPC algorithm in Section 2. Section 3 explains the detailed steps of the proposed algorithm ConDPC. In Section 4, experiments are conducted on synthetic and real-world datasets. Conclusions are presented last.

2. Original Algorithm

DPC algorithm was proposed by Rodriguez et al. in 2014 [13]. The algorithm is based on the assumption that the local densities of cluster centers are high, and different centers are far apart.

Rodriguez et al. [13] obtained two intermediate results for each point in the data set, which were used for the subsequent judgment of density clustering centers and neighbor assignment. Taking point \( i \) as an example, the two results are: (1) Local density \( \rho \) of point \( i \); (2) The minimum distance between point \( i \) and the points of greater local density than it (For the convenience of description, this distance is called the neighbor distance in the following text).

The authors provided a local density definition based on a Gaussian kernel which is used in our study. The definition is as follows:

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

where \( d_{ij} \) is the Euclidean distance between points \( i \) and \( j \) and \( d_c \) is a given cut-off distance.

The set of points with a density greater than this point can be defined as follows:

\[
N_i = \{x_j \in X | \rho_i < \rho_j\}
\]  

For the convenience of subsequent descriptions, we call the points in \( N_i \) the big neighbors of point \( i \).

For point \( i \), the nearest point with a local density greater than it is its neighbor. Then, the neighbor distance is defined as follows [21]:

\[
\delta_i = \begin{cases} 
\min(d_{i(N_i)}), & N_i \text{ is not null} \\
\max(\delta), & \text{otherwise}
\end{cases}
\]  

With the help of local density and neighbor distance, the paper leads to the definition of \( \gamma \) as follows:

\[
\gamma_i = \rho_i * \delta_i
\]  

The larger the \( \gamma \) value, the more likely the corresponding point is to be the cluster center point. That is to say, the point with a larger local density and neighbor distance value is more likely to be selected as the cluster center point.

Taking \( \rho \) as the x-axis and \( \delta \) as the y-axis to draw the decision graph, the points with a higher \( \gamma \) value can be intuitively found, thus providing strong support for the selection of clustering centers.

It can be concluded from Formula (1) that the calculation method of local density is the same for each point, that is, the case of large density difference is not considered. This can lead to incorrect cluster center selection. As shown in Figure 1, when the density
difference is large, the DPC algorithm may find multiple center points in the group with higher density, while no center points are found in the group with lower density. The two larger dots (marked in red and turquoise) in Figure 1a,b represent the clustering centers selected by the DPC algorithm. In all of the following clustering result figures, the cluster centers are labeled with star shapes. The different clusters produced by the clustering algorithms are distinguished by different colors.

![Figure 1. Best clustering result of DPC on Jain (percent = 0.3) (a) decision graph with $\rho$ as x-axis and $\delta$ as y-axis; (b) distribution of $\gamma$ value with points count as x-axis and $\gamma$ as y-axis; (c) clustering result.](image1)

From Formula (3), it can be concluded that the calculation of delta is based on the Euclidean distance. This calculation does not take into account data with manifold structures. This may lead to poor accuracy in neighbor selection. Figure 2 shows the neighbor selection of point $p$ in the external cluster on the Halfkernal dataset. As shown in Figure 2a, the blue point $p$ has several big neighbors (the points marked in black with greater densities than $p$). As shown in Figure 2b, the blue point $p$ selects the nearest point $q$ among all the big neighbors as its neighbor. This kind of neighbor selection error leads to the inaccurate assignment of points to the cluster to which they belong. The clustering result is shown in Figure 2c.

![Figure 2. Best clustering result of DPC on Halfkernal (percent = 5.6) (a) point $p$ and its big neighbors; (b) point $p$ and its neighbor point $q$; (c) clustering result.](image2)

To sum up, the DPC algorithm does not consider the density difference and the distance calculation problem caused by the manifold structure, which may lead to the wrong selection of clustering centers or neighbors. In this study, data connectivity is integrated into the original algorithm to solve the above problems.
3. Proposed Algorithm

Drawing on the idea of connectivity, we propose the algorithm ConDPC to improve the accuracy of the DPC algorithm in specific scenarios, such as data distribution with large differences in density or manifold structure.

3.1. Basic Definitions

This section defines the relevant concepts of connectivity.

**Definition 1. (directly connectable points)**
Points i and j are directly connectable to each other if \(d_{ij} \leq d_c\), where \(d_c\) is a given cut-off distance.

The set of points directly connectable to point i is defined as follows:

\[D\text{Neis}(i) = \{x_j \in X | d_{ij} \leq d_c\}\] (5)

**Definition 2. (connectable points)**
Points i and j are connectable to each other if there is a chain of points \(p_1, \ldots, p_n, p_1 = i, p_n = j\) where \(p_{k+1}\) is directly connectable to \(p_k\). This can be marked as \(r_{ij} = 1\).

Then, the set of points connectable to point i is defined as follows:

\[\text{LinkNeis}(i) = \{x_j \in X | r_{ij} = 1\}\] (6)

The set of points connectable to point i with a greater density can be defined as follows:

\[N - \text{link}_i = \{x_j \in X | \rho_i < \rho_j \text{ and } x_j \in \text{LinkNeis}(i)\}\] (7)

**Definition 3. (connected group)**
A group is connected when any two points in the group are connectable to each other.

3.2. Steps of the Proposed Algorithm

The algorithm can be divided into three steps: (1) determine the connectivity relationship between points; (2) calculate \(\delta\) value according to the connectivity relationship and assign neighbors; (3) determine cluster centers and assign labels.

**Step 1: Determine the connectivity relationship between points.**

In the first step, connected groups are constructed to determine the connectivity between points. The algorithm idea is summarized, as shown in Algorithm 1.

**Algorithm 1 Calculate connected groups**

**Input:** input dataset; cutoff distance

**Output:** connected groups

1. Mark all points as ungrouped;
2. For each point i that has not been grouped
   3. Create a new group G, mark point i as grouped in G, and add its directly connectable point set \(D\text{Neis}(i)\) to G;
   4. For each point j in G that has not been grouped
      5. Mark point j as grouped in G, and add its directly connectable point set \(D\text{Neis}(j)\) to G;
     6. End For
3. End For
4. Return connected groups;

**Step 2: Calculate \(\delta\) and neighbor assignment based on connectivity.**

The original algorithm does not consider the connectivity relationship between two points, directly calculates the distance between point i and its big neighbors, selects the minimum
value as the $\delta$ value of $i$, and takes point $j$ with the minimum distance as the neighbor of point $i$.

According to the connectivity relation, the proposed algorithm considers whether two points are connectable to each other when choosing neighbors and calculating neighbor distance.

If there is a point with a higher density than point $i$ in the connected group, that is, $N - \text{link}_i$ is not empty, then the nearest point in $N - \text{link}_i$ to point $i$ is taken as its neighbor, and the distance between the two points is the neighbor distance of $i$.

If the density of point $i$ in the connected group is the largest, then select the nearest point in $N_i$ to point $i$ as its neighbor, and the distance is updated to the maximum value of the distance between any two points.

The formula for calculating $\delta$ is adjusted as follows:

$$
\delta_i = \begin{cases} 
\min(d_{ij}(N - \text{link}_i)), & \text{if } N - \text{link}_i \text{ is not empty} \\
\max d, & \text{otherwise}
\end{cases}
$$

(8)

where $\max d$ is the maximum value of the distance matrix.

---

**Algorithm 2 Calculate $\delta$ and neighbor assignment**

**Input:** local density matrix; distance matrix; connected groups

**Output:** $\delta$ and neighbor assignment matrix

1. Sort all points in descending order of density;
2. For each point $i$ that has no neighbors assigned
3. In the set $N_i$ of all points with a density larger than point $i$, if there is a subset $N - \text{link}_i$ connectable to point $i$ (that is, there are big neighbors of point $i$ in the same connected group as $i$), select the point $j$ closest to point $i$ in $N - \text{link}_i$ as the neighbor of point $i$, and update $\delta_i$ to $d_{ij}$;
4. In the set $N_i$, if there is no point connectable to point $i$ (that is, there is no big neighbor in the same connected group as point $i$), select the point $j$ closest to point $i$ in $N_i$ as the neighbor of point $i$, and update $\delta_i$ to the maximum value;
5. End For
6. Return $\delta$ and neighbor assignment matrix;

---

**Algorithm 3 Identifying cluster centers and Label assignment**

**Input:** input dataset

**Output:** clustering result

1. Normalize the dataset;
2. Calculate distance matrix $D_{n \times n} = \{d_{ij}\}$;
3. Calculate the local density of each point according to Formula (1);
4. Call Algorithm 1 to obtain connected groups;
5. Call Algorithm 2 to obtain $\delta$ and neighbor assignment matrix;
6. Calculate decision value $\gamma$ according to Formula (4);
7. Draw the decision diagram and select the appropriate cluster centers according to the decision value;
8. Assign other points to corresponding categories according to the neighbor assignment results in Algorithm 2;

This section describes three algorithms, among which Algorithm 3 calls Algorithm 1 and Algorithm 2. Algorithm 3 is the overall flow of ConDPC algorithm. Compared with the
original DPC algorithm, ConDPC algorithm mainly has one more step to call Algorithm 1. The worst time complexity of Algorithm 1 is \( O(n^2) \), so the overall time complexity of ConDPC is still \( O(n^2) \) level.

4. Experiments and Results

In this section, experiments are designed and implemented to test the feasibility of the proposed algorithm, focusing on the effectiveness of the algorithm in clustering data with unbalanced density distribution and manifold distribution. We used eight artificial datasets and six real-world datasets in the experiments. All datasets are normalized by the ‘min–max’ method, to unify the range of each attribute domain to [0, 1].

ConDPC is an improvement on the original DPC algorithm, which is also a density-based clustering algorithm. Therefore, the original DPC and DBSCAN are selected as comparison algorithms in this paper. The widely known K-means [22] is also chosen as the comparison algorithm. We also choose two improved DPC algorithms, DPC-DBFN [16] and McDPC [18], for comparison. The performances of the algorithms were measured according to F-measure [23], FMI [24], and ARI [25].

We select the best clustering results from the appropriate combination of parameter ranges. The main input parameters of the algorithms are shown in Table 1. It should be noted that the parameter setting of McDPC algorithm is relatively dependent on the nature of the dataset [18]. Learning from the experimental settings of the original paper [18] and other works [26,27], we set the search range of the parameters in this paper, as shown in Table 1. All the experiments were conducted in Matlab 2017B.

### Table 1. Information of the input parameters.

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Parameter(s) and the Search Range</th>
</tr>
</thead>
<tbody>
<tr>
<td>ConDPC</td>
<td>percent ( \in {0.1, 0.2, \ldots, 10} )</td>
</tr>
<tr>
<td>DPC</td>
<td>percent ( \in {0.1, 0.2, \ldots, 10} )</td>
</tr>
<tr>
<td>DBSCAN</td>
<td>( eps \in {0.01, 0.02, \ldots, 0.09}, MinPts \in {2, 3, \ldots, 40} )</td>
</tr>
<tr>
<td>K-means</td>
<td>( n: \text{number of clusters} )</td>
</tr>
<tr>
<td>DPC-DBFN</td>
<td>( k \in {3, 4, \ldots, 40} )</td>
</tr>
<tr>
<td>McDPC</td>
<td>( \theta \in {0.01, 0.02, \ldots, 0.50}, \gamma \in {0.01, 0.02, \ldots, 0.50} ), ( \lambda \in {0.01, 0.02, \ldots, 1.00} ), ( pct \in {0.3, 0.4, \ldots, 6.0} )</td>
</tr>
</tbody>
</table>

4.1. Experiments on Synthetic Datasets

4.1.1. Synthetic Datasets

The synthetic datasets used in the experiments have different sizes and distributions. The details of the datasets are shown in Table 2.

### Table 2. Information on the synthetic datasets.

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Source</th>
<th>Instances</th>
<th>Features</th>
<th>Clusters</th>
</tr>
</thead>
<tbody>
<tr>
<td>Aggregation</td>
<td>[28]</td>
<td>788</td>
<td>2</td>
<td>7</td>
</tr>
<tr>
<td>Flame</td>
<td>[29]</td>
<td>240</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>R15</td>
<td>[30]</td>
<td>600</td>
<td>2</td>
<td>15</td>
</tr>
<tr>
<td>Jain</td>
<td>[31]</td>
<td>373</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>Twocircle</td>
<td>[32]</td>
<td>314</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>Halfkernal</td>
<td>[17]</td>
<td>1000</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>Halfcircle</td>
<td>[33]</td>
<td>238</td>
<td>2</td>
<td>3</td>
</tr>
<tr>
<td>Threecircle</td>
<td>[33]</td>
<td>299</td>
<td>2</td>
<td>3</td>
</tr>
</tbody>
</table>

The synthetic datasets can be generalized into three types: (1) The density of different categories is similar (e.g., the Aggregation, Flame, and R15 datasets) (2) The density of different categories is quite different (e.g., the Jain and Twocircle datasets) (3) The data distribution is enclosed/semi-enclosed structure (e.g., the Halfkernal, Halfcircle, and
Threecircle datasets). These three types of datasets may be intersecting, such as Twocircle and Threecircle datasets with uneven density and also nested structure of distribution. During the experiments, the influence of different characteristics of the datasets on the clustering results will be discussed.

The first type of datasets is used to verify whether the proposed algorithm retains the good clustering effect of the original DPC algorithm on such data sets. The latter two types of datasets are used to verify whether the proposed algorithm can solve the problem that the original DPC algorithm has a poor clustering effect on such datasets.

4.1.2. Results on Synthetic Datasets

Firstly, the clustering results on the first kind of dataset are introduced. DPC performs well on this type of dataset, and this group of experiments mainly verifies whether the proposed algorithm can retain the good results of the original DPC algorithm.

The Aggregation dataset has seven clusters. There is little difference in density among the seven clusters. The clustering effect of DPC algorithm on Aggregation is good. The similar density and uniform data distribution make the DPC algorithm applicable to this dataset. ConDPC also achieves a good clustering effect on Aggregation. The clustering result of K-means algorithm is poor. DBSCAN misjudges some points as noise points, and the clustering effect is mediocre. The clustering results are shown in Figure 3, and the star shaped points are the cluster centers [34].

Figure 3. Results on Aggregation. (a) ConDPC, (b) DPC, (c) K-means, (d) DBSCAN, (e) DPC-DBFN, (f) McDPC.

The Flame dataset has two clusters. The shape of the two clusters is different and the density distribution is uniform. As can be seen from Figure 4, ConDPC, DPC, and McDPC have the best clustering effects.
R15 has a total of 15 clusters, as shown in Figure 5. These 15 clusters have similar densities and similar distributions. The DPC algorithm has achieved a good clustering effect on this dataset. The other five algorithms also have good clustering results.
The clustering effect on the latter two datasets is described below. DPC does not perform well on this kind of data set. This group of experiments mainly verifies the good clustering effect of ConDPC on these datasets.

The Jain dataset consists of two lunar data clusters. The density of these two clusters is quite different, so Jain is often used to check the accuracy of the clustering algorithms. The density calculation of DPC does not consider the density difference between different clusters. In the Jain dataset, the density of the bottom cluster is significantly higher than that of the upper cluster. As shown in Figure 6b, DPC finds two center points in the bottom cluster.

![Figure 6. Results on Jain. (a) ConDPC, (b) DPC, (c) K-means, (d) DBSCAN, (e) DPC-DBFN, (f) McDPC.](image)

After ConDPC finds the point with the highest density in a connected group, it will update $\delta$ value of this point to the maximum value. In this way, the corresponding $\gamma$ value is corrected. As a result, although the density of the upper cluster is sparse, for the point with the largest density in the cluster, its $\delta$ value is the maximum value, so its $\gamma$ value increases accordingly, and finally, this point is selected as the clustering center. In this way, ConDPC finds the cluster centers in both clusters and correctly assigns the remaining points to the corresponding clusters. DBSCAN fails to distinguish multiple clusters with large density differences, divides the upper cluster into two, and misjudges a point as a noise point. Only DPC and McDPC implement clustering perfectly, as shown in Figure 6.

The Twocircle dataset consists of two nested circular clusters. The density of the internal circle cluster is higher than that of the external one. DPC does not consider the difference of different cluster densities and selects two points that are farther apart in the internal circle cluster with a higher density as the cluster centers but does not select the cluster center in the external circle cluster.

For the ConDPC algorithm, although the density of the external circle cluster is relatively small, for the point with the largest density in the cluster, its $\delta$ value is updated to the maximum value, so its $\gamma$ value increases accordingly, and finally, this point is selected as the clustering center.
As can be seen from Figure 7, the clustering results of ConDPC, DBSCAN, and McDPC are perfect. However, DPC, K-means, and DPC-DBFN have poor clustering results on the dataset.

![Figure 7. Results on Twocircle. (a) ConDPC, (b) DPC, (c) K-means, (d) DBSCAN, (e) DPC-DBFN, (f) McDPC.](image)

The Halfkernal dataset consists of two surrounding moon-shaped data clusters. The density of the two clusters is similar. When the DPC algorithm selects neighbors, it only considers the Euclidean distance between two points and does not consider the connectivity between the two points. As shown in Figure 2, DPC can find the clustering centers in the two clusters, respectively. However, in the process of neighbor selection, point $p$ of the external cluster chooses point $q$ of the internal cluster as its neighbor, resulting in many points of the external cluster being incorrectly assigned to the internal cluster.

ConDPC preferentially searches for big neighbors in the connected group where the point resides. Therefore, even if there are big neighbors closer to $p$ in the internal cluster, they are not considered because they are not in the same connected group as $p$.

As shown in Figure 8, ConDPC selects the cluster centers in the two clusters, respectively, and assigns the remaining points to the corresponding clusters. DBSCAN also achieves a good clustering result. The other algorithms have poor clustering effects.

The Halfcirlce dataset consists of one moon-shaped cluster and two rectangle-shaped clusters, which is similar to the semi-enclosed structure as a whole. The three clusters have similar densities. As can be seen from Figure 9, DPC can find the appropriate clustering centers in each cluster. However, when assigning the points to the clusters, some points of the moon-shaped cluster are wrongly assigned to the rectangle-shaped cluster on the right. This is still because the original algorithm directly selects the nearest big neighbor to a given point without considering the connectivity between the two points when searching for the neighbor.

ConDPC preferentially searches for big neighbors in the connected group where the point resides. Therefore, even if there are big neighbors in the rectangle-shaped cluster that are closer to a point in the moon-shaped cluster, they are not considered because they are not in the same connected group as the point.
Figure 8. Results on HalfKernel. (a) ConDPC, (b) DPC, (c) K-means, (d) DBSCAN, (e) DPC-DBFN, (f) McDPC.

Figure 9. Results on HalfCircle. (a) ConDPC, (b) DPC, (c) K-means, (d) DBSCAN, (e) DPC-DBFN, (f) McDPC.

As shown in Figure 9, ConDPC selects the cluster center points in the three clusters, respectively, and assigns the remaining points to the corresponding clusters. DBSCAN also
achieves a good clustering result. K-means and the last two improved algorithms produce similar poor results.

The ThreeCircle dataset is made up of three nested circles. The inner circle is denser than the outer circle. It can be seen from Figure 10 that: (1) DPC algorithm fails to correctly select the clustering center; (2) DPC algorithm fails to correctly assign the remaining points to the cluster. The DPC algorithm selects one cluster center in the inner circle with the largest density, two cluster centers in the middle circle, and no cluster center in the outer circle with the sparsest density. When the other points are assigned to the clusters, some points in the middle circular cluster are incorrectly assigned to the inner circular cluster.

Figure 10. Results on ThreeCircle. (a) ConDPC, (b) DPC, (c) K-means, (d) DBSCAN, (e) DPC-DBFN, (f) McDPC.

ConDPC can find the appropriate cluster center in each cluster because it modifies the calculation method of neighbor distance. In addition, ConDPC preferentially searches for big neighbors in the same connected group, so the neighbors are searched correctly and assigned to the corresponding cluster.

As can be seen from Figure 10, the clustering results of ConDPC, DBSCAN, and McDPC are perfect. Both DPC and K-means get bad clustering results.

Table 3 shows the clustering comparison results on the synthetic datasets. From Table 3, we can see that the traditional DPC algorithm achieves good clustering results on datasets Aggregation, Flame, and R15. On the Jain and TwoCircle datasets with obvious density differences, DPC finds multiple cluster centers in the dense cluster, but fails to find cluster centers in the sparse cluster, so the clustering effect is poor. DPC can find the clustering center point on the semi-enclosed datasets HalfKernel and HalfCircle. However, when assigning neighbors to other points, the points will be assigned to the wrong neighbor because the relative location structure of the data set is not considered, which leads to the wrong cluster allocation.

The proposed algorithm ConDPC has achieved good clustering results on the first three datasets. At the same time, on datasets with obvious density differences, ConDPC can find the corresponding center points in different clusters. On the datasets with enclosed/semi-enclosed distributions, ConDPC takes into account the data connectivity and preferentially
searches for neighbors in the same connected group so that points can be correctly assigned to the corresponding clusters.

Table 3. Performances on synthetic datasets.

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Aggregation</th>
<th>F-measure</th>
<th>Flame</th>
<th>R15</th>
<th>Jain</th>
<th>Twocircle</th>
<th>Halfkernal</th>
<th>Halfcircle</th>
<th>Threecircle</th>
</tr>
</thead>
<tbody>
<tr>
<td>ConDPC</td>
<td></td>
<td>0.998</td>
<td>1</td>
<td>0.997</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td></td>
<td>0.997</td>
<td>1</td>
<td>0.993</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td></td>
<td>0.996</td>
<td>1</td>
<td>0.993</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>Parameter</td>
<td></td>
<td>3.1</td>
<td>2.7</td>
<td>3.4</td>
<td>3</td>
<td>2.7</td>
<td>1.5</td>
<td>3.8</td>
<td>7.2</td>
</tr>
<tr>
<td>DPC</td>
<td></td>
<td>0.998</td>
<td>1</td>
<td>0.997</td>
<td>0.910</td>
<td>0.610</td>
<td>0.819</td>
<td>0.943</td>
<td>0.586</td>
</tr>
<tr>
<td></td>
<td></td>
<td>0.997</td>
<td>1</td>
<td>0.993</td>
<td>0.882</td>
<td>0.534</td>
<td>0.729</td>
<td>0.884</td>
<td>0.492</td>
</tr>
<tr>
<td></td>
<td></td>
<td>0.996</td>
<td>1</td>
<td>0.993</td>
<td>0.715</td>
<td>0.050</td>
<td>0.422</td>
<td>0.824</td>
<td>0.214</td>
</tr>
<tr>
<td>Parameter</td>
<td></td>
<td>3.1</td>
<td>2.7</td>
<td>3.4</td>
<td>0.3</td>
<td>0.2</td>
<td>0.1</td>
<td>4.2</td>
<td>0.2</td>
</tr>
<tr>
<td>DBSCAN</td>
<td></td>
<td>0.996</td>
<td>0.979</td>
<td>0.994</td>
<td>0.976</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td></td>
<td>0.994</td>
<td>0.974</td>
<td>0.988</td>
<td>0.990</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td></td>
<td>0.992</td>
<td>0.944</td>
<td>0.987</td>
<td>0.976</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>Parameter</td>
<td></td>
<td>0.08/21</td>
<td>0.1/10</td>
<td>0.05/30</td>
<td>0.08/2</td>
<td>0.07/4</td>
<td>0.08/5</td>
<td>0.08/5</td>
<td>0.09/5</td>
</tr>
<tr>
<td>K-Means</td>
<td></td>
<td>0.834</td>
<td>0.833</td>
<td>0.997</td>
<td>0.864</td>
<td>0.510</td>
<td>0.522</td>
<td>0.798</td>
<td>0.451</td>
</tr>
<tr>
<td></td>
<td></td>
<td>0.788</td>
<td>0.736</td>
<td>0.993</td>
<td>0.820</td>
<td>0.497</td>
<td>0.506</td>
<td>0.665</td>
<td>0.403</td>
</tr>
<tr>
<td></td>
<td></td>
<td>0.730</td>
<td>0.453</td>
<td>0.993</td>
<td>0.577</td>
<td>−0.003</td>
<td>0.002</td>
<td>0.488</td>
<td>0.053</td>
</tr>
<tr>
<td>Parameter</td>
<td></td>
<td>7</td>
<td>2</td>
<td>15</td>
<td>2</td>
<td>2</td>
<td>3</td>
<td>3</td>
<td>3</td>
</tr>
<tr>
<td>DPC-DBFN</td>
<td></td>
<td>0.995</td>
<td>0.991</td>
<td>0.998</td>
<td>0.925</td>
<td>0.503</td>
<td>0.743</td>
<td>0.818</td>
<td>0.712</td>
</tr>
<tr>
<td></td>
<td></td>
<td>0.994</td>
<td>0.985</td>
<td>0.997</td>
<td>0.906</td>
<td>0.497</td>
<td>0.674</td>
<td>0.687</td>
<td>0.715</td>
</tr>
<tr>
<td></td>
<td></td>
<td>0.993</td>
<td>0.967</td>
<td>0.996</td>
<td>0.768</td>
<td>−0.003</td>
<td>0.267</td>
<td>0.522</td>
<td>0.476</td>
</tr>
<tr>
<td>Parameter</td>
<td></td>
<td>30</td>
<td>6</td>
<td>39</td>
<td>14</td>
<td>4</td>
<td>7</td>
<td>3</td>
<td>17</td>
</tr>
<tr>
<td>McDPC</td>
<td></td>
<td>0.998</td>
<td>1</td>
<td>0.997</td>
<td>1</td>
<td>1</td>
<td>0.819</td>
<td>0.798</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td></td>
<td>0.997</td>
<td>1</td>
<td>0.993</td>
<td>1</td>
<td>1</td>
<td>0.729</td>
<td>0.665</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td></td>
<td>0.996</td>
<td>1</td>
<td>0.993</td>
<td>1</td>
<td>1</td>
<td>0.422</td>
<td>0.488</td>
<td>1</td>
</tr>
<tr>
<td>Parameter</td>
<td></td>
<td>0.21/1</td>
<td>0.4/5</td>
<td>0.05/2</td>
<td>0.23/5</td>
<td>0.08/2</td>
<td>0.21/5</td>
<td>0.3/2</td>
<td>0.08/2</td>
</tr>
</tbody>
</table>

The visualization results from Figures 3–10 and the index values in Table 3 show that the ConDPC algorithm has the highest accuracy among all comparison algorithms on the datasets.

4.2. Experiments on Real-World Datasets

4.2.1. Real-World Datasets

To verify the feasibility of the proposed algorithm, we also use real-world datasets for validation. The real datasets used in this experiment are from UCI Machine Learning Repository [35]. The detailed information of the data sets is shown in Table 4.

Table 4. Information of the real-world datasets.

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Source</th>
<th>Instances</th>
<th>Features</th>
<th>Clusters</th>
</tr>
</thead>
<tbody>
<tr>
<td>Iris</td>
<td>[35]</td>
<td>150</td>
<td>4</td>
<td>3</td>
</tr>
<tr>
<td>Wine</td>
<td>[35]</td>
<td>178</td>
<td>13</td>
<td>3</td>
</tr>
<tr>
<td>Wine</td>
<td>[35]</td>
<td>198</td>
<td>32</td>
<td>2</td>
</tr>
<tr>
<td>Seeds</td>
<td>[35]</td>
<td>210</td>
<td>7</td>
<td>7</td>
</tr>
<tr>
<td>Ecoli</td>
<td>[35]</td>
<td>336</td>
<td>7</td>
<td>8</td>
</tr>
<tr>
<td>Pageblocks</td>
<td>[35]</td>
<td>5473</td>
<td>10</td>
<td>5</td>
</tr>
</tbody>
</table>

4.2.2. Results on Real-World Datasets

Table 5 shows the best clustering index values of the four algorithms on the six datasets. The last column is the average value of the clustering index of each algorithm on the datasets. Figure 11a shows the best F-measure result of each algorithm on each dataset, and Figure 11b shows the average index values of the algorithms on the six datasets.
## Table 5. Performances on real-world datasets.

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Iris</th>
<th>Wine</th>
<th>Wpbc</th>
<th>Seeds</th>
<th>Ecoli</th>
<th>Pageblocks</th>
<th>Average</th>
</tr>
</thead>
<tbody>
<tr>
<td>ConDPC</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>F-measure</td>
<td>0.967</td>
<td>0.879</td>
<td>0.673</td>
<td>0.913</td>
<td>0.444</td>
<td>0.322</td>
<td>0.700</td>
</tr>
<tr>
<td>FMI</td>
<td>0.935</td>
<td>0.775</td>
<td>0.762</td>
<td>0.844</td>
<td>0.777</td>
<td>0.926</td>
<td>0.837</td>
</tr>
<tr>
<td>ARI</td>
<td>0.904</td>
<td>0.660</td>
<td>0.244</td>
<td>0.767</td>
<td>0.692</td>
<td>0.498</td>
<td>0.628</td>
</tr>
<tr>
<td>Parameter</td>
<td>2.5</td>
<td>0.8</td>
<td>0.4</td>
<td>0.7</td>
<td>0.6</td>
<td>2.7</td>
<td></td>
</tr>
<tr>
<td>DPC</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>F-measure</td>
<td>0.960</td>
<td>0.885</td>
<td>0.661</td>
<td>0.913</td>
<td>0.259</td>
<td>0.287</td>
<td>0.661</td>
</tr>
<tr>
<td>FMI</td>
<td>0.923</td>
<td>0.783</td>
<td>0.760</td>
<td>0.844</td>
<td>0.506</td>
<td>0.908</td>
<td>0.787</td>
</tr>
<tr>
<td>ARI</td>
<td>0.886</td>
<td>0.672</td>
<td>0.227</td>
<td>0.767</td>
<td>0.349</td>
<td>0.387</td>
<td>0.548</td>
</tr>
<tr>
<td>Parameter</td>
<td>0.2</td>
<td>2.0</td>
<td>7.4</td>
<td>0.7</td>
<td>0.4</td>
<td>8</td>
<td></td>
</tr>
<tr>
<td>DBSCAN</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>F-measure</td>
<td>0.587</td>
<td>0.526</td>
<td>0.439</td>
<td>0.543</td>
<td>0.298</td>
<td>0.293</td>
<td>0.448</td>
</tr>
<tr>
<td>FMI</td>
<td>0.753</td>
<td>0.718</td>
<td>0.563</td>
<td>0.694</td>
<td>0.763</td>
<td>0.848</td>
<td>0.723</td>
</tr>
<tr>
<td>ARI</td>
<td>0.625</td>
<td>0.538</td>
<td>0.034</td>
<td>0.524</td>
<td>0.669</td>
<td>0.392</td>
<td>0.464</td>
</tr>
<tr>
<td>Parameter</td>
<td>0.13/9</td>
<td>0.51/23</td>
<td>0.64/4</td>
<td>0.3/35</td>
<td>0.23/30</td>
<td>0.08/33</td>
<td></td>
</tr>
<tr>
<td>K-Means</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>F-measure</td>
<td>0.885</td>
<td>0.949</td>
<td>0.555</td>
<td>0.891</td>
<td>0.543</td>
<td>0.263</td>
<td>0.681</td>
</tr>
<tr>
<td>FMI</td>
<td>0.811</td>
<td>0.903</td>
<td>0.579</td>
<td>0.803</td>
<td>0.559</td>
<td>0.583</td>
<td>0.706</td>
</tr>
<tr>
<td>ARI</td>
<td>0.716</td>
<td>0.854</td>
<td>0.032</td>
<td>0.705</td>
<td>0.425</td>
<td>0.108</td>
<td>0.473</td>
</tr>
<tr>
<td>Parameter</td>
<td>3</td>
<td>3</td>
<td>2</td>
<td>3</td>
<td>8</td>
<td>5</td>
<td></td>
</tr>
<tr>
<td>DPC-DBFN</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>F-measure</td>
<td>0.967</td>
<td>0.945</td>
<td>0.489</td>
<td>0.896</td>
<td>0.530</td>
<td>0.358</td>
<td>0.698</td>
</tr>
<tr>
<td>FMI</td>
<td>0.935</td>
<td>0.888</td>
<td>0.721</td>
<td>0.809</td>
<td>0.755</td>
<td>0.914</td>
<td>0.837</td>
</tr>
<tr>
<td>ARI</td>
<td>0.904</td>
<td>0.832</td>
<td>0.004</td>
<td>0.715</td>
<td>0.658</td>
<td>0.254</td>
<td>0.561</td>
</tr>
<tr>
<td>Parameter</td>
<td>5</td>
<td>3</td>
<td>19</td>
<td>4</td>
<td>3</td>
<td>39</td>
<td></td>
</tr>
<tr>
<td>McDPC</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>F-measure</td>
<td>0.883</td>
<td>0.625</td>
<td>0.452</td>
<td>0.903</td>
<td>0.480</td>
<td>0.440</td>
<td>0.631</td>
</tr>
<tr>
<td>FMI</td>
<td>0.816</td>
<td>0.680</td>
<td>0.787</td>
<td>0.826</td>
<td>0.822</td>
<td>0.928</td>
<td>0.810</td>
</tr>
<tr>
<td>ARI</td>
<td>0.720</td>
<td>0.439</td>
<td>0.008</td>
<td>0.742</td>
<td>0.739</td>
<td>0.524</td>
<td>0.529</td>
</tr>
<tr>
<td>Parameter</td>
<td>0.1/0.1/</td>
<td>0.2/0.3/</td>
<td>0.2/0.52/</td>
<td>0.02/0.02/</td>
<td>0.02/0.01/</td>
<td>0.01/0.01/</td>
<td>0.33/1</td>
</tr>
</tbody>
</table>

Figure 11. (a) F-measure of each algorithm on each dataset; (b) The average index values of each algorithm on the six datasets.

It can be seen from Table 5 and Figure 11 that the average values of the three indexes corresponding to the ConDPC algorithm are the highest.

### 4.3. Experimental Discussion

ConDPC follows the input parameter settings of the original DPC. In the actual experiments, we set the range and step size of the input parameters of the two algorithms to be the same. We set the range of the input parameter `percent` to [0.1, 9] and the step size to 0.1, and select the best result from the corresponding results.

In the experiments of artificial datasets in Section 4.1, the value of each clustering index of the DPC algorithm is the highest or equal to the highest. DPC has achieved good...
clustering results on the Aggregation, Flame, and R15 datasets. The parameter settings of ConDPC on these datasets are completely the same as those of DPC. In fact, the clustering centers, neighbor assignment, and final results obtained by ConDPC are also the same as those of DPC. ConDPC retains the advantage of good clustering effects of DPC on such datasets. DPC performs poorly on the other five datasets while ConDPC still achieves good clustering results.

In the experiments of real-world datasets in Section 4.2, it can be seen from the clustering index results that ConDPC obtains better clustering results than DPC does on most datasets. The average values of the three indexes of DPC are the highest, corresponding to the best comprehensive evaluation of the clustering results.

5. Conclusions

In this study, we proposed an improved density peak clustering algorithm ConDPC based on data connectivity. In this algorithm, the judgment of data connectivity is added to the original algorithm, and then the calculation of neighbor distance and the neighbor allocation rule are modified.

Experiments showed that ConDPC improves the accuracy of the original algorithm and expands the application scenarios. When the density of different clusters is different or the data distribution is nested, ConDPC can still achieve a good clustering effect.

The algorithm improves the accuracy of clustering, but it also takes a certain amount of time. The subsequent work will further consider the improvement of algorithm performance and apply the algorithm to specific businesses, such as AIS trajectory clustering.

Author Contributions: Conceptualization and methodology, Y.Z. and Z.W.; software, validation, visualization, investigation, resources, data curation, writing—original draft preparation, Y.Z.; writing—review and editing, Y.Z.; funding acquisition, Y.Z. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by China Postdoctoral Science Foundation (No. 2019M651844), the Natural Science Foundation of the Jiangsu Higher Education Institutions of China (No. 21KJB580007), and the funding of the innovation science and technology project of Jiangsu Maritime Institute (No. 016102).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The datasets underlying this article are cited and mentioned in the article.

Acknowledgments: We thank Juan Zhang, Jie Zhu and Taizhi Lv for their reviewing and editing of the manuscript and their funding acquisition work.

Conflicts of Interest: The authors declare no conflict of interest.

References

16. Lotfi, A.; Moradi, P.; Beigy, H. Density peaks clustering based on density backbone and fuzzy neighborhood. Pattern Recognit. 2020, 107, 107449. [CrossRef]
20. Tao, X.; Guo, W.; Ren, C.; Li, Q.; He, Q.; Liu, R.; Zou, J. Density peak clustering using global and local consistency adjustable manifold distance. Inf. Sci. 2021, 577, 769–804. [CrossRef]