Mathematical and Computational Applications

MDPI

Article

Density Peak Clustering Based on Relative Density under Progressive Allocation Strategy

Yongli Liu *, Congcong Zhao and Hao Chao

School of Computer Science and Technology, Henan Polytechnic University, Jiaozuo 454003, China
* Correspondence: yongli.buaa@gmail.com

Abstract: In traditional density peak clustering, when the density distribution of samples in a dataset is uneven, the density peak points are often concentrated in the region with dense sample distribution, which is easy to affect clustering accuracy. Under the progressive allocation strategy, a density peak clustering algorithm based on relative density is proposed in this paper. This algorithm uses the K-nearest neighbor method to calculate the local density of sample points. In addition, in order to avoid the domino effect during sample allocation, a new similarity calculation method is defined, and a progressive allocation strategy from near to far is used for the allocation of the remaining points. In order to evaluate the effectiveness of this algorithm, comparative experiments with five algorithms were carried out on classical artificial datasets and real datasets. Experimental results show that the proposed algorithm can achieve higher clustering accuracy on datasets with uneven density distribution.

Keywords: density peak clustering; progressive allocation strategy; relative density

Citation: Liu, Y.; Zhao, C.; Chao, H. Density Peak Clustering Based on Relative Density under Progressive Allocation Strategy. *Math. Comput. Appl.* **2022**, *27*, 84. https://doi.org/10.3390/mca27050084

Academic Editor: Leonardo Trujillo

Received: 24 August 2022
Accepted: 27 September 2022
Published: 6 October 2022

1. Introduction

Clustering is an unsupervised machine learning [1–3] technique that aims to group objects according to the similarity relationship so that objects with high similarity are assigned to the same group and objects with high dissimilarity are isolated to different groups. Because clustering can discover the inherent structure information of objects, it has been widely used in image processing [4–6], fraud detection [7,8], information security [9,10], and medical applications [11,12].

In 2014, Rodriguez and Laio [13] proposed the density peaks clustering (DPC) algorithm. This algorithm classifies objects in two steps: (1) assuming that the cluster centers have a high local density and are relatively far away from each other, a decision graph is generated to select cluster centers that meet the assumptions; (2) noncentral points are assigned to the nearest neighbor with higher density. Based on the above steps, DPC cannot only effectively select cluster centers from the decision graph, but also effectively allocate the remaining noncentral points. Benefiting from the simple and efficient clustering logic, DPC can achieve better clustering results on datasets with an arbitrary shape. However, DPC is not impeccable, and it still faces some problems to be improved. For example, this algorithm uses Euclidean distance to calculate the density and search for the peak values of density, which is not suitable for a manifold structure [14], and the results are not satisfactory when processing some datasets with an uneven density. Further, the cluster allocation strategy of DPC may produce a domino effect, that is, the wrong allocation of one point may lead to the wrong allocation of all subsequent points. In order to overcome these problems, many researchers have improved and extended the original DPC algorithm. Du et al. [15] proposed the DPC-KNN algorithm based on the K-nearest neighbor (KNN) [16] concept. This algorithm changed the method of calculating local density in DPC, combined density peak clustering with KNN, and considered the surrounding environment of objects. At the same time, this algorithm used the principal component analysis to improve the

performance of high-dimensional data. Xie et al. [17] proposed the FKNN-DPC algorithm, which calculates local density and performs object allocation based on a fuzzy weighted K-nearest neighbor technique. This algorithm can identify clusters with different shapes and is superior to DPC in many aspects. Liu et al. [18] proposed an SNN-DPC algorithm based on the SNN (Shared Nearest Neighbor) concept. This algorithm adopts a new local density measurement and proposes a relative distance based on shared neighbors, which can more objectively adapt to the surrounding environment and improve the accuracy and robustness of uneven data sets. Hou et al. [19] analyzed the impact of the kernel density estimation method in DPC, redefined the local density using KNN, and designed a new clustering algorithm by using the distance normalization principle. Xu et al. [20] introduced the merging micro cluster strategy, and Zhao et al. [21] proposed the DPC-MND algorithm, which uses KNN to calculate the local density of the samples and find the density peak. The mutual proximity of unallocated points is used to measure the sample proximity, which alleviates the joint and several errors of DPC allocation. Although the clustering results obtained by these methods are more ideal or efficient than those of the DPC algorithm, there are still some problems, such as complex models and increased time consumption.

Inspired by the above algorithms, in this paper, we propose a density peak clustering algorithm based on relative density under the progressive allocation strategy named DPC-RD-PAS. This algorithm redefines the local density of objects by using the idea of K-nearest neighbor and enlarges the influence of the surrounding environment in the calculation of local density. In order to avoid the domino effect caused by the distribution of the remaining noncentral points, the strategy of progressive distribution is adopted.

The rest of this paper is arranged as follows: The second section introduces the traditional DPC algorithm. Section 3 describes the definitions and steps related to the DPC-RD-PAS algorithm in detail. The fourth section describes the experiments of our work, including the experimental preparation and analysis of the experimental results. In the last section, we conclude our work.

2. Density Peak Clustering

DPC is a new clustering algorithm based on density and distance. This algorithm assumes that (1) each cluster center is surrounded by neighbors with low local density, and (2) the distance between the cluster center and any point with high local density is relatively large. In DPC, each data point, i, is described by two important indicators: the local density, ρ_i, and the distance, δ_i, between data point i and the nearest point with a higher density.

For the local density value of data point i, the DPC algorithm provides two calculation methods: the cutoff distance method and the kernel distance method, which are respectively defined as follows:

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

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

where d_{ij} is the Euclidean distance between data points i and j, and d_c is the cutoff distance and is the neighborhood radius set by the user. Therefore, the local density, ρ_i, is related to the number of points whose distance from data point i is less than the cutoff distance, d_c. The local density obtained by Equation (1) is a discrete value, and that obtained by Equation (2) is a continuous value.

The relative distance is defined as follows:

$$\delta_i = \min_{j:\rho_j > \rho_i} (d(x_i, x_j)) \tag{3}$$

As shown in Equation (3), the relative distance of sample point i is the minimum distance d_{ij} to point j, where the condition of sample point j is that its local density is greater than that of sample point i. For sample point i with the highest density, its relative distance is defined as follows:

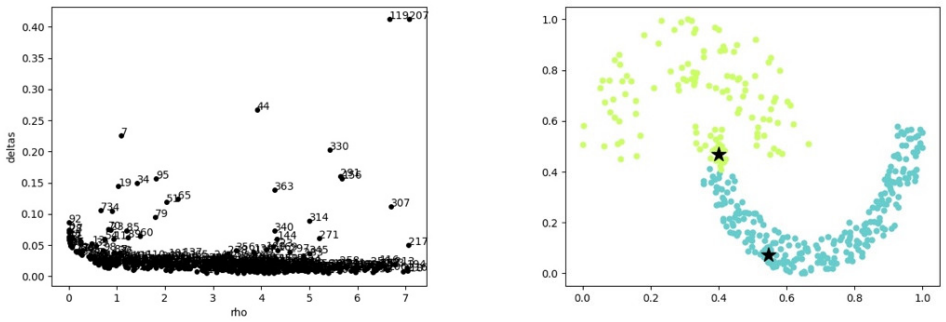
$$\delta_i = \max_j(d(x_i, x_j)) \tag{4}$$

The cluster center points are located at the top right of the decision graph, that is, the cluster centers have a high density and large relative distance at the same time. To facilitate the selection of appropriate cluster center points in the decision diagram, the following formula is defined:

$$\gamma_i = \rho_{\mathrm{i}} * \delta_{\mathrm{i}} \tag{5}$$

DPC algorithm clustering mainly includes two steps. The first step tries to find the density peak. Based on the above analysis, we can find the appropriate cluster centers at the upper right side of the decision graph, where the x-axis of the decision graph is composed of the local density calculated by Equations (1) and (2), and the y-axis of the decision graph is the relative density, which is calculated by Equations (3) and (4). In the second step, the remaining sample points are allocated to the cluster to which the nearest neighbor with a higher density belongs. The nearest neighbor has been obtained when calculating the relative distance, and, therefore, the DPC algorithm has high allocation efficiency.

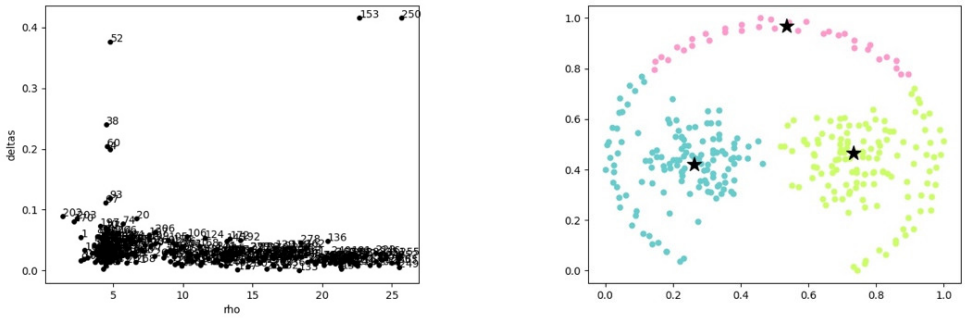
Although the experimental results show that DPC performs well in many cases, the allocation strategy on some non-uniform [22] density datasets has some shortcomings. Figure 1b describes the clustering results of DPC on Jain, a classic data set with an uneven density. In Figure 1, black solid pentagrams represent the cluster centers, and different colors represent different clusters. It can be seen that the density of the upper part of this dataset is significantly lower than that of the lower part when we use the local density calculation method of DPC. After selecting the points with a high local density and relative distance as cluster centers through the decision graph, we can see that the two cluster centers are both wrongly selected in the lower half of the dataset. Moreover, due to the wrong selection of cluster centers, a series of wrong assignments occur in the subsequent points.



(**a**) Decision chart (**b**) Clustering results

Figure 1. Decision graph and clustering results of DPC on the Jain dataset.

As shown in Figure 2, on the Pathbased dataset, the DPC algorithm can select the correct cluster centers from the decision graph. However, as the remaining objects are allocated from high to low density, they are allocated to the cluster where the assigned points with higher density and the smallest relative distance are located. It can be seen from Figure 2b that the blue points are distributed first because of their high density, and, thus, form a blue cluster. The points on the left ring should have been assigned to the pink cluster, but because the density is significantly lower than the blue cluster, they are incorrectly assigned to the blue cluster.



(**a**) Decision chart (**b**) Clustering results

Figure 2. Decision graph and clustering results of DPC on the Pathbased dataset.

3. DPC-RD-PAS

DPC is easily affected by the cutoff distance, d_c, when calculating the density of the sample points. This is because the value of d_c is determined based on the global distribution of objects, ignoring the local information between objects, which is easy to cause the cluster centers to be concentrated in the area with dense objects (as shown in Figure 1). In view of this, our DPC-RD-PAS algorithm uses the K-nearest neighbor idea to define the local density calculation method and then calculates the local density of the sample points.

3.1. Relative K-Nearest Neighbor Local Density

Definition 1. ***Relative K-nearest neighbor local density****. The local density calculated from the relative K-nearest neighbor around the sample point is called the relative K-nearest neighbor local density, which can be calculated as follows:*

$$\rho_i = \frac{\sum\limits_{j \in \Gamma(i)} \exp(-d_{ij}^2)}{\sum\limits_{j \in \overline{\Gamma}(i)} \exp(-d_{ij}^2)} \tag{6}$$

where $\Gamma(i)$*represents the set of the K-nearest neighbors of sample point* i*, and* $\overline{\Gamma}(i)$ *is the total set composed of K-nearest neighbors of all objects in the set* $\Gamma(i)$*.*

By using Equation (6) to calculate the local density of the sample points, the possibility that the cluster centers are located in a relatively sparse region can be improved through the relative concept, so as to avoid the cluster centers being concentrated in the high-density region. This method is helpful to improve the correctness of the cluster centers' selection, especially for datasets with an uneven density distribution.

In addition, our DPC-RD-PAS algorithm optimizes the allocation mode of DPC and adopts the strategy of multi-step progressive allocation.

3.2. Progressive Allocation

To introduce the multi-step progressive allocation strategy in detail, the following two definitions are given.

Definition 2. ***Nearest neighbors among unassigned points.*** *In the KNN range of the allocated point* P*, find all the unassigned points in the KNN range. Among the nearest neighbors of all the unassigned points, the nearest point will be regarded as the unassigned point.*

For example, in Figure 3, take point P_2 as the center to calculate the K-nearest neighbors, which can be divided into two groups, assigned points and unallocated points. The blue points (P_1, P_2, and P_3) are assigned points, and the grey points (Q_1, Q_2, and Q_3) are unallocated points. Find the nearest neighbor from the unallocated points to the

assigned points, taking the shortest distance as the benchmark, and take the corresponding unallocated points as the points to be assigned. From Figure 3, we can obtain that $d(Q_1, P_1) < d(Q_2, P_2) < d(Q_3, P_3)$. Therefore, point Q_1 will be the nearest neighbor among the unassigned points.

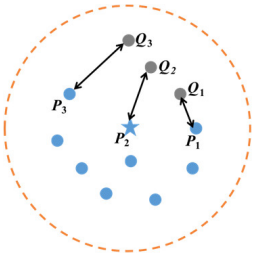


Figure 3. Nearest neighbor among unassigned points.

Definition 3. ***Relation degree.*** *The K-nearest neighbors of point P and point Q are calculated respectively and sorted to obtain the set* $\Gamma(P)$ and $\Gamma(Q)$. *The ranking position of point P in the* $\Gamma(Q)$ *set is* $\overline{P_Q}$, *and the ranking position of point Q in the* $\Gamma(P)$ *set is* $\overline{Q_P}$. *Then, the relation degree between point P and point Q is:*

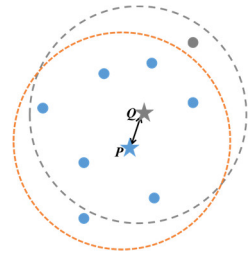
$$Rel = \frac{\overline{P_Q} + \overline{Q_P}}{K} \tag{7}$$

The smaller the value *Rel*, the higher the relation degree between the sample points. Assuming that point P has been assigned the corresponding cluster label, and point Q is one point to be assigned, we need to judge whether Q should be assigned the same cluster label as P by calculating the relation degree between point P and point Q.

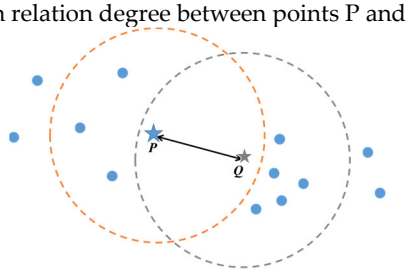
If $0 < Rel < 0.5$, the relation degree between point P and point Q is very high. We think these two points are very similar and, therefore, assign point Q the same cluster label as point P. If $0.5 < Rel < 1$, the relation degree between point P and point Q is relatively high. We think these two points are similar but we cannot assign a cluster label to point Q for the time being. If $Rel > 1$, the relation degree between point P and point Q is so low that point P cannot determine the cluster label of point Q. Figure 4 shows the different correlations between point P and point Q. Suppose $K = 9$, point P has been assigned a cluster label, and point Q is waiting to be assigned a cluster label.

As shown in Figure 4a, point Q is one of the K-nearest neighbors of point P, and $\overline{Q_P} = 2$, and point P is also one of the K-nearest neighbors of point Q, and $\overline{P_Q} = 2$. According to Equation (7), the *Rel* value can be calculated to be $4/9$, indicating that the similarity between these two points is very high, and, therefore, a subordinate label of point P is assigned to point Q. In Figure 4b, point Q is one of the K-nearest neighbors of point P, and $\overline{Q_P} = 5$, and point P is also one of the K-nearest neighbors of point Q, and $\overline{P_Q} = 7$. According to Equation (7), the *Rel* value can be calculated to be $12/9$. This value is greater than 1, so we cannot assign the dependent label of point P to point Q. It can be seen from Figure 4c that point Q is one of the K-nearest neighbors of point P, and $\overline{Q_P} = 4$, and point P is also one of the K-nearest neighbors of point Q, and $\overline{P_Q} = 4$. The *Rel* value calculated using Equation (7) is $8/9$, which indicates that the relation degree between these two points is in an ambiguous area, and we cannot assign a cluster label to point Q temporarily.

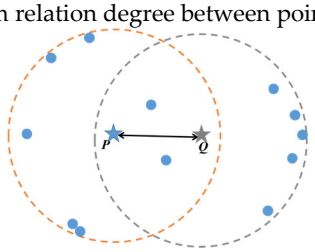
An example of the P and Q ranking calculation is as follows: Calculate the ranking of P and Q, as shown in Figure 4d, where $d(P,P) < d(Q,P) < d(m_3,P) < d(m_4,P) < d(m_5,P) < d(m_6,P) < d(m_7,P) < d(m_8,P) < d(m_9,P)$. In the neighborhood of $K = 9$ centered on P, it can be seen that point Q is in the second place centered on point P, which means that the ranking position is 2.



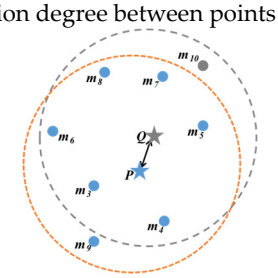
(**a**) High relation degree between points P and Q.



(**b**) Medium relation degree between points P and Q.



(**c**) Low relation degree between points P and Q.



(**d**) Ranking relationship between P and Q.

Figure 4. Relation degree between points P and Q when K = 9.

3.3. Steps of DPC-RD-PAS

After introducing the above concepts, the steps of our DPC-RD-PAS algorithm are designed as follows:

Input: the value of K.
Output: the clustering results.
Step 1: Pre-process and the normalize dataset.
Step 2: Calculate the relative K-nearest neighbor local density, ρ_i, and relative distance, δ_i, using Equations (6) and (3), respectively.
Step 3: Select the cluster centers according to the decision diagram.
Step4: Allocate the K-nearest neighbor points around the cluster centers to their corresponding class cluster.
Step 5: Find the nearest neighbors among the unassigned points of all assigned points according to Definition 2 and calculate the relation degree between the assigned points and the unassigned points according to Definition 3.
Step 6: Assign all the unassigned points with the value of a relation degree between 0 and 0.5 to the cluster where the corresponding assigned point is located; update the sets of assigned points and unassigned points and recalculate the relation degree.
Step 7: If there are still unassigned points with a value of relation degree between 0 and 0.5, go to Step 6.

Step 8: Assign all the unassigned points with a value of relation degree between 0.5 and 1 to the cluster where the corresponding assigned point is located; update the sets of assigned points and unassigned points and recalculate the relation degree.
Step 9: If there are still unassigned points with a value of relation degree between 0.5 and 1, go to Step 8.
Step 10: If there are unassigned sample points, they will be allocated to the cluster where the nearest allocated sample points with a higher density are located, and the clustering process is complete.

4. Discussion

4.1. Experimental Preparation

In order to verify the effectiveness of our DPC-RD-PAS algorithm, comparative experiments with the DPC, DPC-KNN, K-Means [23], DBSCAN (Density-Based Spatial Clustering of Applications with Noise) [24], and DPCSA (DPC based on weighted local density Sequence and nearest neighbor Assignment) [25] algorithms were carried out. The experimental datasets include classic synthetic datasets and UCI datasets. The details of these datasets are listed in Tables 1 and 2.

Table 1. Synthetic datasets.

Dataset	Source	#Samples	#Attributes	#Classes
Jain	[26]	373	2	2
R15	[27]	600	2	15
Flame	[28]	240	2	2
Aggregation	[29]	788	2	7
Pathbased	[30]	300	2	3
Spiral	[30]	312	2	3

Table 2. Real-world datasets.

Dataset	#Samples	#Attributes	#Classes
Iris	150	4	3
Seeds	210	7	3
WDBC	569	30	2
Libras	360	90	15
Wine	178	13	3
Ecoli	336	8	6

In order to quantify the quality of the clustering results, we selected three evaluation indicators to measure the accuracy of the clustering results, namely AMI (Adjusted Mutual Information), the ARI (Adjusted Rand Index), and the FMI (Fowles Mallows Index). The maximum value of these three indicators is 1. In the process of clustering, when the clustering results are better, the values of these three indicators are closer to 1.

In order to ensure that the experimental results were more accurate and objective, in our experiments, we optimized the parameters of all the algorithms and referred to the optimal parameters provided by the SNN-DPC algorithm.

4.2. Results on Synthetic Datasets

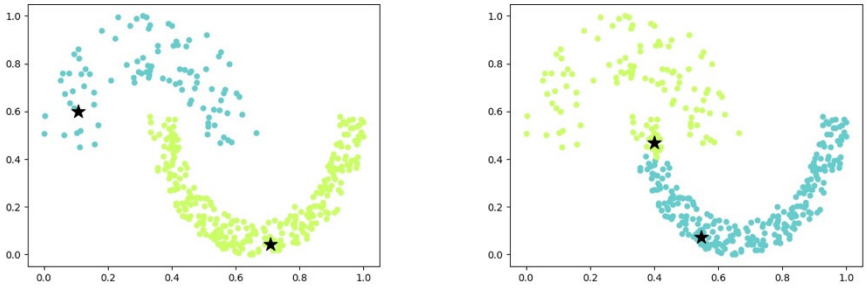
In order to more clearly illustrate the clustering performance of our DPC-RD-PAS algorithm on the uneven density datasets, we graphically displayed the results on datasets Jain, Pathbased, and Spiral, as shown in Figures 5–7, respectively. In these three figures, different colors represent different clusters, the black pentagram represents the center of one cluster, and the grey "×" represents the unallocated sample points.

As shown in Figure 5, the Jain dataset is composed of two crescent moons, of which the sample points in the lower half are evenly distributed and the density is relatively

high, so it is easy to concentrate the cluster centers in the lower half when calculating the local density according to DPC. The DPC-KNN algorithm has been improved to solve this problem. Although the measurement method of local density has been unified, this problem has not been completely solved (as Figure 5c), resulting in the selection of cluster centers still being difficult to be satisfied. In the results of the DBSCAN algorithm, it can be clearly seen from Figure 5e that the upper part is wrongly divided into two clusters, and some sample points at the right corner are treated as noise points, so the clustering results are not particularly satisfactory. Because the K-Means algorithm has some disadvantages for non-spherical datasets, it is still not successful on the Jain dataset. Our DPC-RD-PAS algorithm adopts the concept of relative density, which is different from the concept of the K-nearest neighbor proposed by the DPC-KNN algorithm. It cannot only consider the K-nearest neighbors of each sample point and shrink the calculation range from the global point to the nearest neighbor points but also considers the K-nearest neighbors near the K-nearest neighbor sample points. This strategy enlarges the role of the surrounding points and can better find the cluster centers for uneven sample sets. The experimental results of our DPC-RD-PAS algorithm on the uneven data set Jain to confirm the correctness of the design idea.

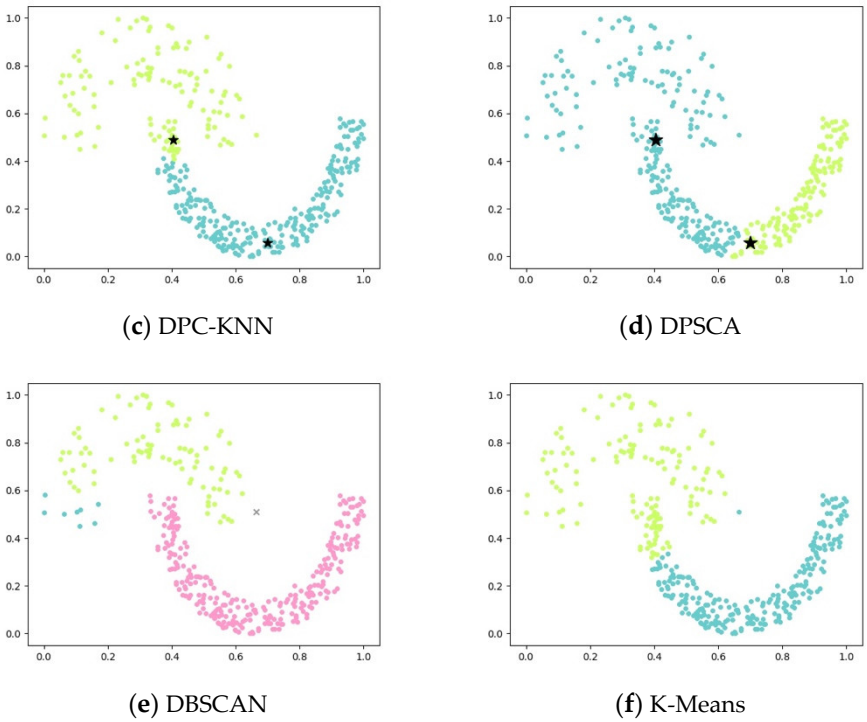
The dataset, Pathbased, is composed of three classes, as shown in Figure 6. There are two dense classes in the middle, and the sparse ring sample points around them form the third class. On this dataset, the DPC, DPC-KNN, and DPCSA algorithms can all find the correct cluster centers, but there are joint errors in the allocation of the remaining sample points. Both DPC and DPC-KNN adopt the principle of ascending the arrangement according to the density and nearest neighbor allocation, which leads to allocation errors. The DBSCAN algorithm (as Figure 6e) mistakenly treats the data of the surrounding ring sample points as noise (the grey "×" sample points in Figure 6e). The K-Means algorithm still fails to allocate the Pathbased dataset correctly. The DPC-RD-PAS algorithm proposed in this paper improves the allocation strategy of the remaining sample points and achieves the optimal clustering results on this dataset.

As shown in Figure 7, the Spiral data set is composed of three spirals. On this dataset, except for the K-Means algorithm, the other algorithms can all obtain the correct clustering centers. Our DPC-RD-PAS algorithm uses the relative density method to calculate the cluster center, which cannot only find the correct cluster centers on this data set, but also correctly allocate the remaining points as other density-based algorithms. This group of results shows that this algorithm cannot only have good clustering performance on uneven data sets, but also obtain satisfactory clustering results on some spiral datasets, such as Spiral.



(**a**) DPC-RD-PAS (**b**) DPC

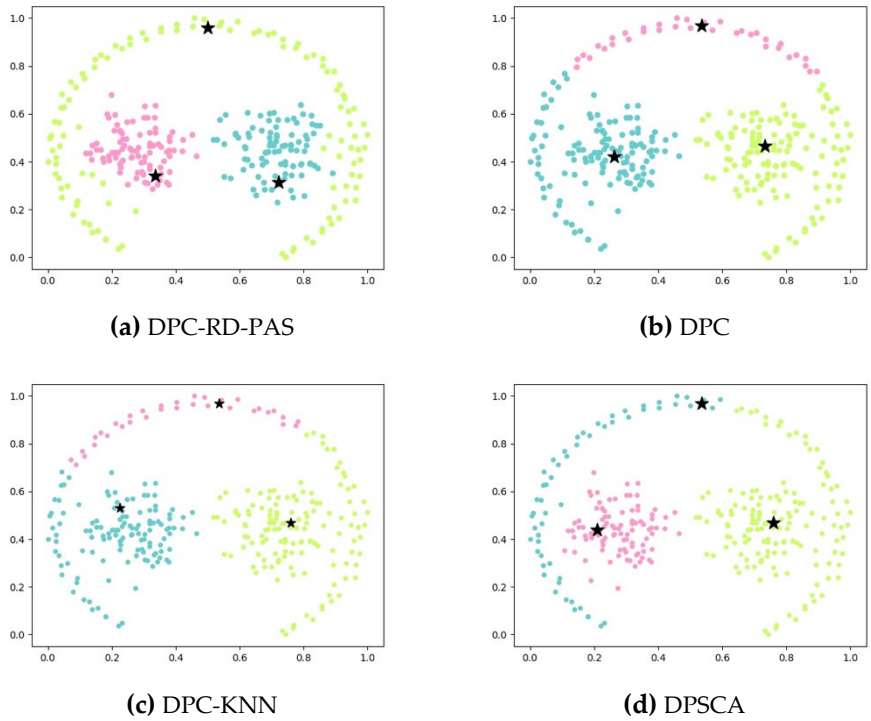
Figure 5. *Cont.*



(**c**) DPC-KNN (**d**) DPSCA

(**e**) DBSCAN (**f**) K-Means

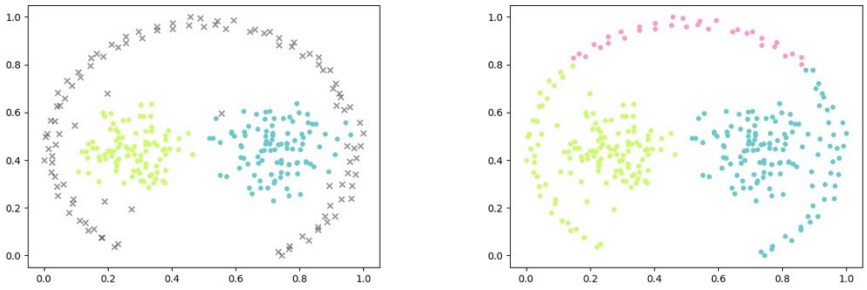
Figure 5. Clustering results on the Jain dataset.



(a) DPC-RD-PAS **(b)** DPC

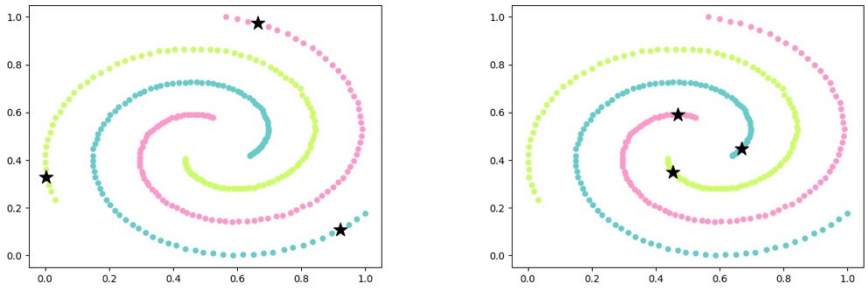
(c) DPC-KNN **(d)** DPSCA

Figure 6. *Cont.*

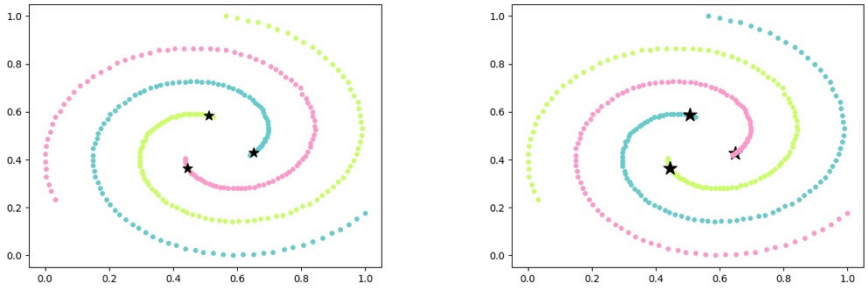


(e) DBSCAN **(f)** K-Means

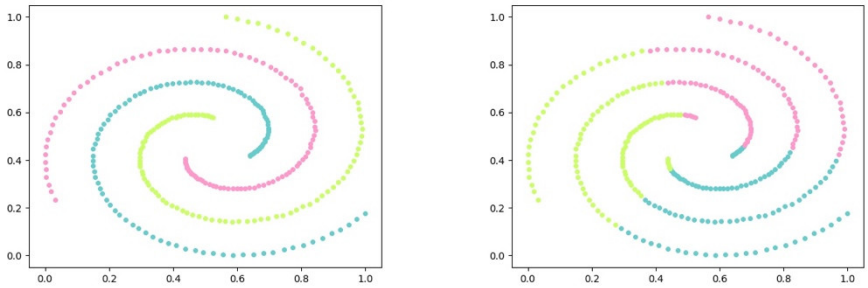
Figure 6. Clustering results on the Pathbased dataset.



(a) DPC-RD-PAS **(b)** DPC



(c) DPC-KNN **(d)** DPSCA



(e) DBSCAN **(f)** K-Means

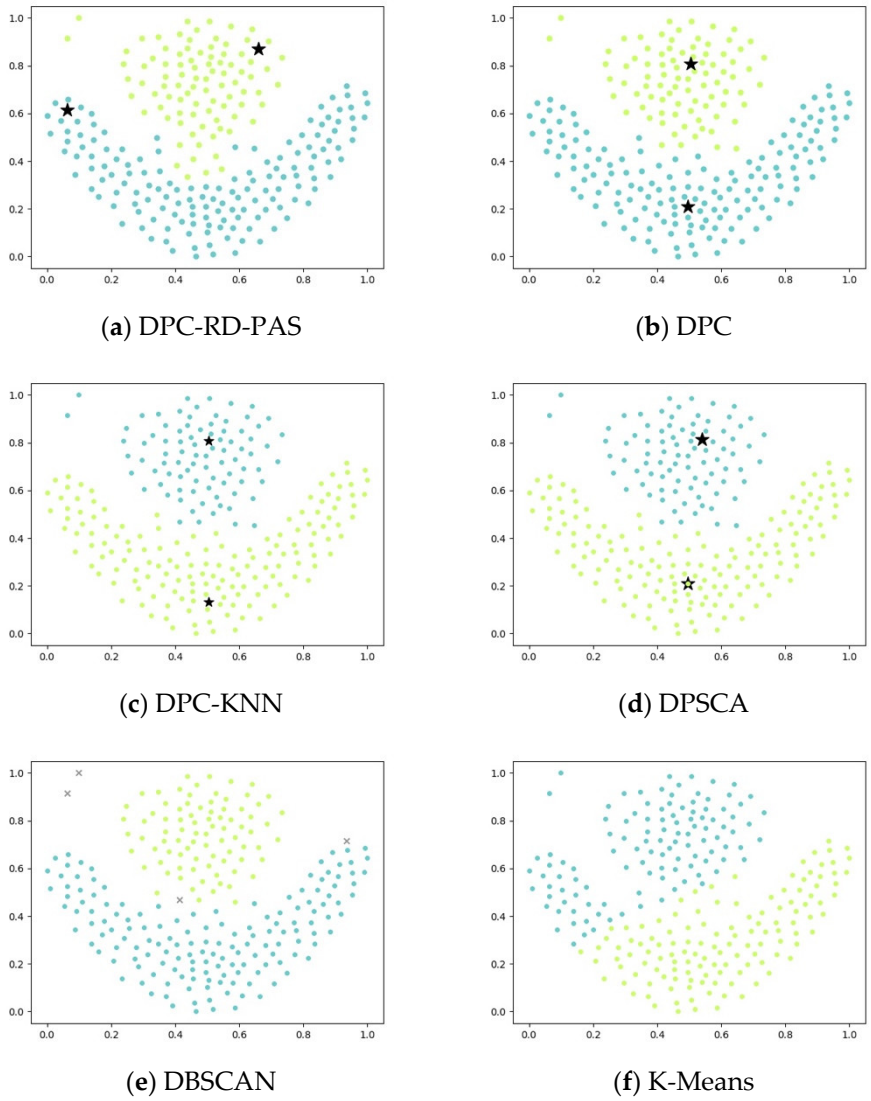
Figure 7. Clustering results on the Spiral dataset.

As shown in Figure 8, the Flame dataset is composed of two types of clusters. It can be seen from this figure that each algorithm can obtain correct clustering results except for K-Means. The clustering performance of the DPC-RD-PAS algorithm is slightly inferior to that of DPC. The main disadvantage is the adjacent position of the two clusters.

Figure 9 illustrates the results on the Aggregation dataset, which consists of seven clusters. The clustering results of the DPC-RD-PAS algorithm are also inferior to that of DPC.

Like the Flame dataset, the reason is still mainly concentrated at the intersection points, which leads to some sample points at the boundary of the orange area being incorrectly allocated to the blue area. Based on our analysis, in the progressive allocation strategy of the DPC-RD-PAS algorithm, the unallocated points with high similarity are allocated first, then the ones with medium similarity, and finally the ones with low similarity. The allocated set is updated every time the unallocated points are allocated according to the similarity until all the points are allocated. For some specific data sets, such as Aggregation, the performance may be poor at the junction, but the progressive strategy allocation method can better avoid the domino effect of DPC in the allocation of the remaining points.

As shown in Figure 10, each algorithm can obtain ideal clustering results on the R15 dataset.

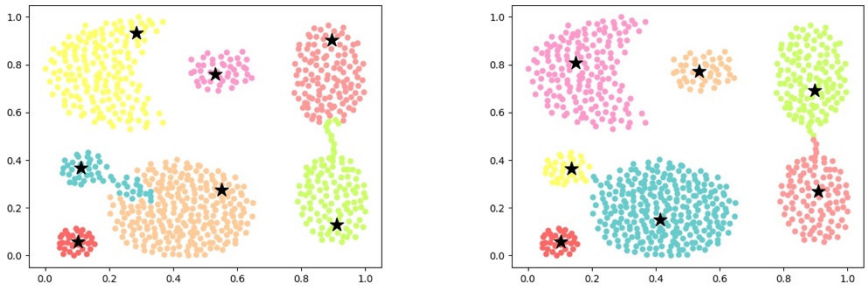


(**a**) DPC-RD-PAS (**b**) DPC

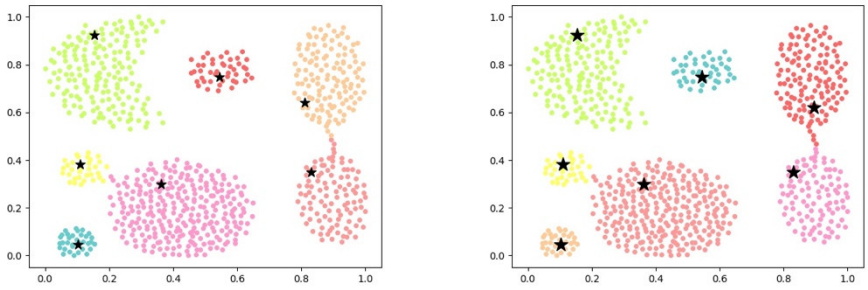
(**c**) DPC-KNN (**d**) DPSCA

(**e**) DBSCAN (**f**) K-Means

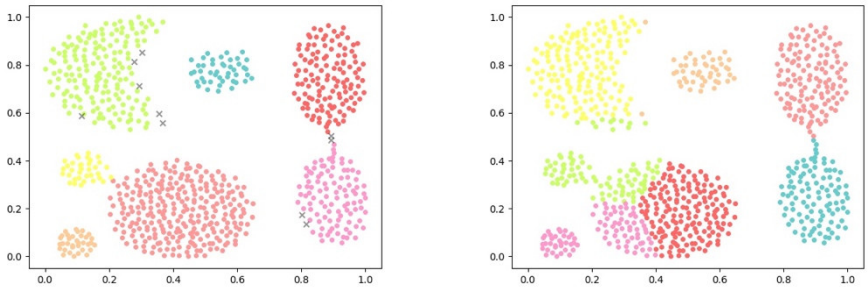
Figure 8. Clustering results on the Flame dataset.



(**a**) DPC-RD-PAS (**b**) DPC

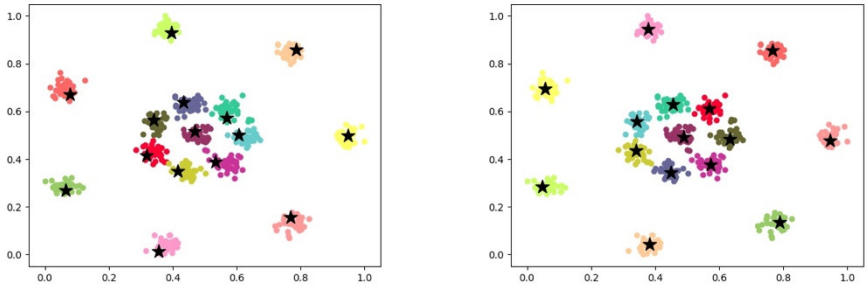


(**c**) DPC-KNN (**d**) DPSCA



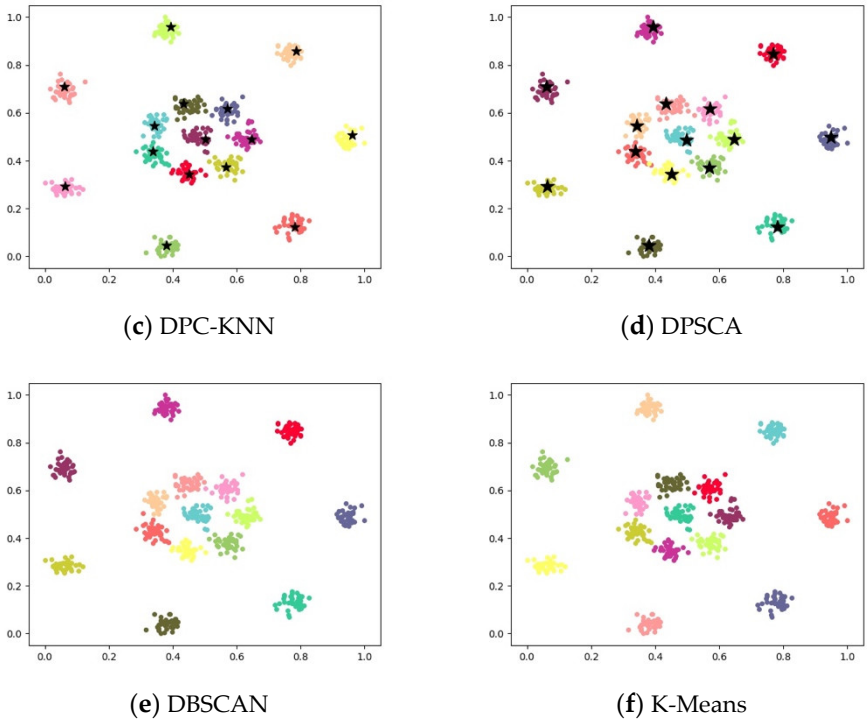
(**e**) DBSCAN (**f**) K-Means

Figure 9. Clustering results on the Aggregation dataset.



(**a**) DPC-RD-PAS (**b**) DPC

Figure 10. *Cont.*



(**c**) DPC-KNN (**d**) DPSCA

(**e**) DBSCAN (**f**) K-Means

Figure 10. Clustering results on the R15 dataset.

The specific clustering results on each dataset are shown in Table 3. The data in Table 3 not only includes the AMI, ARI, and FMI index values of the clustering results, but also gives the corresponding optimal parameters of each algorithm (the column represented by Arg-). The optimal values in the tables of this paper are shown in bold. It can be seen from this table that on the three datasets, Jain, Pathbased, and Spiral, with an uneven density distribution, the AMI, ARI, and FMI index values of our DPC-RD-PAS algorithm are the best. On the R15 and Aggregation datasets, the AMI, ARI, and FMI index values of the DPC-RD-PAS algorithm are close to the optimal DPC and DPC-KNN algorithms. The performance of the DPC-RD-PAS algorithm on the Flame dataset is relatively inferior, which is closely related to the data distribution characteristics of this dataset itself.

Table 3. Performance on synthetic datasets.

Algorithm	**Jain**				**R15**				**Flame**			
	AMI	ARI	FMI	Arg-	AMI	ARI	FMI	Arg-	AMI	ARI	FMI	Arg-
DPC-RD-PAS	**1**	**1**	**1**	18	0.9783	0.9714	0.9733	8	0.7858	0.8701	0.9392	8
DPC	0.6514	0.7146	0.8819	0.9	**0.9938**	**0.9928**	**0.9932**	2	**1**	**1**	**1**	3
DPC-KNN	0.6514	0.7146	0.8819	7	**0.9938**	**0.9928**	**0.9932**	7	**1**	**1**	**1**	5
DPSCA	0.2313	0.0442	0.5924	-	0.9885	0.9857	0.9866	-	**1**	**1**	**1**	-
DBSCAN	0.9276	0.9758	0.9906	0.08/2	0.9850	0.9819	0.9831	0.04/12	0.8656	0.9388	0.9712	0.09/8
K-Means	0.5264	0.5767	0.8200	2	**0.9938**	**0.9928**	**0.9932**	15	0.4045	0.4647	0.7420	2
Algorithm	**Aggregation**				**Pathbased**				**Spiral**			
	AMI	ARI	FMI	Arg-	AMI	ARI	FMI	Arg-	AMI	ARI	FMI	Arg-
DPC-RD-PAS	0.9152	0.8821	0.9077	32	**0.9387**	**0.9494**	**0.9663**	25	**1**	**1**	**1**	7
DPC	**0.9923**	**0.9956**	**0.9966**	4	0.5513	0.4717	0.6664	4	**1**	**1**	**1**	1.8
DPC-KNN	**0.9923**	**0.9956**	**0.9966**	7	0.5489	0.4679	0.6654	5	**1**	**1**	**1**	5
DPSCA	0.9570	0.9581	0.9673	-	0.7290	0.6133	0.7511	-	**1**	**1**	**1**	-
DBSCAN	0.9706	0.9808	0.9850	0.04/6	0.8713	0.9011	0.9340	0.08/10	**1**	**1**	**1**	0.04/2
K-Means	0.8276	0.7151	0.7765	7	0.5428	0.4613	0.6617	3	−0.0055	−0.0060	0.3274	3

4.3. Results on UCI Datasets

Table 4 lists the clustering results of each algorithm on the six UCI datasets. On the Iris dataset, the index of our DPC-RD-PAS algorithm is slightly lower than that of the DPC-KNN algorithm and the DPSCA algorithm. The decline of AMI, ARI, and FMI are 2.7%, 2.0%, and 1.3%, respectively. On the Seeds dataset, DPC-RD-PAS has the best clustering results. Compared with DPC, DPC-KNN, DPSCA, DBSCAN, and K-Means, the AMI index increased by 4.25%, 4.25%, 14.82%, 30.16%, and 10.23% respectively. On the WDBC, Libras, Wine, and Ecoli datasets, DPC-RD-PAS achieved relatively good clustering results. Especially on the WDBC dataset, the values of AMI, ARI, and FMI are 11.1%, 19.9%, and 1.3% higher, respectively, than the K-Means algorithm, which performed the second best in this dataset. On the Wine dataset, our DPC-RD-PAS algorithm also performed well, and its AMI, ARI, and FMI indexes were improved by 7.93%, 14.47%, and 8.14%, respectively, compared with the DPC algorithm. In addition, compared with other algorithms, it also achieved the best clustering results on the Libras dataset with relatively high dimensions.

Table 4. Performance on UCI datasets.

Algorithm	**Iris**				**Seeds**				**WDBC**			
	AMI	ARI	FMI	Arg-	AMI	ARI	FMI	Arg-	AMI	ARI	FMI	Arg-
DPC-RD-PAS	0.8605	0.8857	0.9233	14	**0.7630**	**0.7954**	**0.8631**	34	**0.5157**	**0.5890**	**0.8017**	24
DPC	0.8625	0.8857	0.9233	2	0.7319	0.7670	0.8444	0.7	−0.0003	−0.0050	0.7160	1.3
DPC-KNN	**0.8836**	**0.9038**	**0.9355**	5	0.7319	0.7664	0.8439	6	0.4496	0.4552	0.7813	7
DPSCA	**0.8836**	**0.9038**	**0.9355**	-	0.6645	0.6873	0.7918	-	0.3891	0.3771	0.7595	-
DBSCAN	0.6341	0.6120	0.7291	0.12/5	0.5862	0.5291	0.6711	0.24/16	0.3593	0.4786	0.7570	0.46/38
K-Means	0.7551	0.7302	0.8208	3	0.6922	0.7166	0.8106	3	0.4640	0.4914	0.7915	2
Algorithm	**Libras**				**Wine**				**Ecoli**			
	AMI	ARI	FMI	Arg-	AMI	ARI	FMI	Arg-	AMI	ARI	FMI	Arg-
DPC-RD-PAS	**0.5826**	**0.3256**	**0.3893**	11	**0.7635**	**0.7697**	**0.8473**	33	0.5622	**0.6450**	**0.7355**	15
DPC	0.5531	0.3193	0.3717	0.3	0.7074	0.6724	0.7835	2	0.5139	0.3486	0.5059	0.4
DPC-KNN	0.5278	0.2721	0.3504	7	0.7233	0.6990	0.8006	7	**0.6101**	0.4990	0.6272	7
DPSCA	0.5514	0.2824	0.3617	-	0.7501	0.7414	0.8283	-	0.5079	0.4884	0.6788	-
DBSCAN	0.4544	0.1965	0.2570	0.9/2	0.5858	0.5292	0.7121	0.5/21	0.4934	0.5255	0.6623	0.2/22
K-Means	0.5511	0.3199	0.3720	15	0.4227	0.3711	0.5835	3	0.6000	0.4163	0.5521	8

4.4. Running Time

The time complexity of the DPC algorithm is mainly composed of the complexity of calculating the distance matrix between the samples, the complexity of calculating the local density of the samples, and the complexity of calculating the relative distance of the samples. The time complexity of each part is $O(n^2)$, so the total time complexity is $O(n^2)$. The time complexity of the DPC-RD-PAS algorithm is mainly composed of the following five parts: (1) the complexity, $O(n^2)$, of calculating the distance matrix between the samples; (2) calculate the complexity, O (n^2), of the relative local density of each sample; (3) calculate the complexity, $O(n^2)$, of the sample relative distance; (4) the first step allocates the time complexity of the k-neighboring points around the cluster center as O(n); (5) the second step is to calculate the similarity of the unallocated points. Assuming that the number of remaining unallocated points is m, $m < n$, and that the time complexity is $O(m^2) < O(n^2)$, the time complexity of the DPC-RD-PAS algorithm is $O(n^2)$. Since it takes a relatively long time to find and judge whether it meets the requirements of merging when calculating the similarity between the unallocated points and allocated points, it will lead to a high running time on the datasets.

In this part, we ran the experiment on a computer with a 1.4 GHz quad core Intel i5 CPU and 8.0 GB of RAM. The operating environment was Python 3.9 (the DPC, DPC-KNN, and DPC-RD-PAS algorithms) and MATLAB 2018 (for the other algorithms). In order to reduce the impact of the running environment, the algorithm under the MATLAB 2018 environment was ignored in the time comparison. At the same time, in order to reduce the unexpected scenarios generated during the running of the program, for each data set, during the running of the different algorithms, we used the best parameters, provided

in Tables 3 and 4, to execute the same process ten times. The running time values shown in Table 5 are all the average running times. It can be seen that the time consumption of the multi-step allocation strategy of the DPC-RD-PAS algorithm is larger than that of the one-step sample allocation strategy of the DPC algorithm. Although the time complexity of the DPC-RD-PAS and DPC algorithms is on the order of $O(n^2)$, the time consumption for processing the actual data sets is different. The actual time consumption of the DPC-RD-PAS algorithm in this paper should be greater than that of the original DPC algorithm, but the running time is not as high as expected.

Table 5. Running time of three density peak clustering algorithms (Unit: second).

Name	DPC-RD-PAS	DPC-KNN	DPC	Name	DPC-RD-PAS	DPC-KNN	DPC
Jain	0.6152	0.6851	0.6165	Iris	0.1206	0.1001	0.1199
R15	1.3807	1.4992	1.6148	Seeds	0.2807	0.2268	0.2144
Flame	0.2354	0.2533	0.2787	WDBC	1.6576	1.3596	1.3798
Aggregation	2.8101	2.3595	2.7673	Libras	0.5884	0.6247	0.6017
Pathbased	0.4703	0.4054	0.3937	Wine	0.2088	0.1367	0.1544
Spiral	0.3575	0.4346	0.4707	Ecoli	0.5116	0.5582	0.5269

5. Conclusions

In order to improve the clustering performance of the DPC algorithm in processing datasets with an uneven density, we propose a density peak clustering algorithm based on relative density under a recursive allocation strategy named DPC-RD-PAS. This algorithm inherits the advantages of the DPC algorithm and can quickly find the density peak points. At the same time, using the idea of K-nearest neighbor for reference, the concept of the relative K-nearest neighbor local density has been introduced to improve the calculation method of the local density and improve the ability of cluster center selection on non-uniform density datasets. After obtaining the correct cluster centers, a recursive allocation strategy was designed for avoiding joint errors in the allocation of the remaining points. In order to evaluate the clustering performance of our DPC-RD-PAS algorithm, comparative experiments were carried out on six artificial datasets and six real datasets. The experimental results show that our DPC-RD-PAS algorithm can achieve satisfactory clustering results on datasets with an uneven density distribution. How to determine automatically the optimal parameter k of the algorithm will be the focus of the next step.

Author Contributions: Conceptualization, Y.L. and C.Z.; methodology, Y.L. and C.Z.; software, Y.L.; supervision, Y.L.; writing—original draft preparation, C.Z. and Y.L.; writing—review and editing, Y.L. and H.C. All authors have read and agreed to the published version of the manuscript.

Funding: The authors would like to thank the support of the National Science Fund's subsidized project under Grant [61872126].

Data Availability Statement: The UCI datasets used in this paper were derived from the UCI (University of California Irvine) Machine Learning Repository. Please visit: https://archive.ics.uci.edu/ml/datasets.php (accessed on 1 September 2022).

Acknowledgments: The authors would like to thank the members of the IR&DM Research Group from Henan Polytechnic University for their invaluable advice that allowed this paper to be successfully completed.

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

References

1. Dold, D.; Fahrion, K. Evaluating the feasibility of interpretable machine learning for globular cluster detection. *arXiv* **2022**, arXiv:2204.00017. [CrossRef]
2. Almeida, F.; Romão, E.L.; Gomes, G.F.; de Freitas Gomes, J.H.; de Paiva, A.P.; Miranda Filho, J.; Balestrassi, P.P. Combining machine learning techniques with Kappa–Kendall indexes for robust hard-cluster assessment in substation pattern recognition. *Electr. Power Syst. Res.* **2022**, *206*, 107778. [CrossRef]

3. Srivastava, P.R.; Eachempati, P.; Kumar, A.; Jha, A.K.; Dhamotharan, L. Best strategy to win a match: An analytical approach using hybrid machine learning-clustering-association rule framework. *Ann. Oper. Res.* **2022**, 1–43. [CrossRef]
4. Bindhu, V.; Ranganathan, G. Hyperspectral Image Processing in Internet of Things model using Clustering Algorithm. *J. ISMAC* **2021**, *3*, 163–175.
5. Oskouei, A.G.; Hashemzadeh, M. CGFFCM: A color image segmentation method based on cluster-weight and feature-weight learning. *Softw. Impacts* **2022**, *11*, 100228. [CrossRef]
6. Yan, M.; Chen, Y.; Chen, Y.; Zeng, G.; Hu, X.; Du, J. A Lightweight Weakly Supervised Learning Segmentation Algorithm for Imbalanced Image Based on Rotation Density Peaks. *Knowl. Based Syst.* **2022**, *244*, 108513. [CrossRef]
7. Al-Hashedi, K.G.; Magalingam, P. Financial fraud detection applying data mining techniques: A comprehensive review from 2009 to 2019. *Compu. Sci. Rev.* **2021**, *40*, 100402. [CrossRef]
8. Li, T.; Kou, G.; Peng, Y.; Philip, S.Y. An integrated cluster detection, optimization, and interpretation approach for financial data. *IEEE Trans. Cybern.* **2021**. [CrossRef]
9. Zhang, E.; Li, H.; Huang, Y. Practical multi-party private collaborative k-means clustering. *Neurocomputing* **2022**, *467*, 256–265. [CrossRef]
10. Bozdemir, B.; Canard, S.; Ermis, O.; Möllering, H.; Önen, M.; Schneider, T. Privacy-preserving density-based clustering. In Proceedings of the 2021 ACM Asia Conference on Computer and Communications Security, Hong Kong, 7–11 June 2021; pp. 658–671.
11. Baragilly, M.; Gabr, H.; Willis, B.H. Clustering functional data using forward search based on functional spatial ranks with medical applications. *Stat. Methods Med. Res.* **2022**, *31*, 47–61. [CrossRef]
12. Sridhar, B.; Sridhar, S.; Nanchariah, V.; Gayatri, K. Cluster Medical Image Segmentation using Morphological Adaptive Bilateral Filter based BSA Algorithm. In Proceedings of the 2021 5th International Conference on Trends in Electronics and Informatics (ICOEI), Tirunelveli, India, 3–5 June 2021.
13. Rodriguez, A.; Laio, A. Clustering by fast search and find of density peaks. *Science* **2014**, *344*, 1492–1496. [CrossRef] [PubMed]
14. Zhou, Z.; Feng, B.; Yang, P.; Wen, X. Research and Implementation of KNN classification algorithm for streaming data based on Storm. *Comput. Eng. Appl.* **2017**, *53*, 71–75.
15. Du, M.; Ding, S.; Jia, H. Study on density peaks clustering based on k-nearest neighbors and principal component analysis. *Knowl.-Based Syst.* **2016**, *99*, 1351–1451. [CrossRef]
16. Wu, X.; Wang, S.; Zhang, Y. Survey on theory and application of k-Nearest-Neighbors algorithm. *Comput. Eng. Appl.* **2017**, *53*, 1–7.
17. Xie, J.; Gao, H.; Xie, W.; Liu, X.; Grant, P.W. Robust clustering by detecting density peaks and assigning points based on fuzzy weighted K-nearest neighbors. *Inf. Sci.* **2016**, *354*, 19–40. [CrossRef]
18. Liu, R.; Wang, H.; Yu, X. Shared-nearest-neighbor-based clustering by fast search and find of density peaks. *Inf. Sci.* **2018**, *450*, 200–226. [CrossRef]
19. Hou, J.; Cui, H. Density Normalization in Density Peak Based Clustering. *Graph-Based Represent. Pattern Recognit.* **2017**, *10310*, 187–196.
20. Xu, L.; Zhao, J.; Yao, Z.; Shi, A.; Chen, Z. Density Peak Clustering Based on Cumulative Nearest Neighbors Degree and Micro Cluster Merging. *J. Signal Process. Syst.* **2019**, *91*, 1219–1236. [CrossRef]
21. Zhao, J.; Yao, Z.F.; Lü, L. Density peaks clustering based on mutual neighbor degree. *Control. Decis. Mak.* **2021**, *36*, 543–552.
22. He, H.; Garcia, E.A. Learning from imbalanced data. *IEEE Trans. Knowl. Data Eng.* **2009**, *21*, 1263–1284.
23. Macqueen, J. Some methods for classification and analysis of multivariate observations. *Berkeley Symp. Math. Stat. Probab.* **1967**, *5*, 281–297.
24. Ester, M.; Kriegel, H.P.; Sander, J.; Xu, X. *A Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise*; AAAI Press: Palo Alto, CA, USA, 1996.
25. Yu, D.; Liu, G.; Guo, M.; Liu, X.; Yao, S. Density Peaks Clustering Based on Weighted Local Density Sequence and Nearest Neighbor Assignment. *IEEE Access* **2019**, *7*, 34301–34317. [CrossRef]
26. Jain, A.K.; Law, M.H. Data clustering: A user's dilemma. *PReMI* **2005**, *3776*, 1–10.
27. Veenman, C.J.; Reinders, M.J.T.; Backer, E. A maximum variance cluster algorithm. *IEEE Trans. Pattern Anal. Mach. Intell.* **2002**, *24*, 1273–1280. [CrossRef]
28. Fu, L.; Medico, E. Flame, a novel fuzzy clustering method for the analysis of dna microarray data. *BMC Bioinform.* **2007**, *8*, 3. [CrossRef]
29. Gionis, A.; Mannila, H.; Tsaparas, P. Clustering aggregation. *ACM Trans. Knowl. Discov. Data* **2007**, *1*, 4. [CrossRef]
30. Chang, H.; Yeung, D.-Y. Robust path-based spectral clustering. *Pattern Recognit.* **2008**, *41*, 191–203. [CrossRef]