Hindawi
Mobile Information Systems
Volume 2022, Article ID 4176101, 13 pages
https://doi.org/10.1155/2022/4176101

Research Article

A Fast Density Peak Clustering Method with Autoselect Cluster Centers

Zhihe Wang, Yongbiao Li, Hui Du, and Xiaofen Wei

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

Correspondence should be addressed to Yongbiao Li; 2020211943@nwnu.edu.cn

Received 30 September 2021; Revised 6 December 2021; Accepted 18 December 2021; Published 11 January 2022

Academic Editor: Pei-Wei Tsai

Aiming at density peaks clustering needs to manually select cluster centers, this paper proposes a fast new clustering method with auto-select cluster centers. Firstly, our method groups the data and marks each group as core or boundary groups according to its density. Secondly, it determines clusters by iteratively merging two core groups whose distance is less than the threshold and selects the cluster centers at the densest position in each cluster. Finally, it assigns boundary groups to the cluster corresponding to the nearest cluster center. Our method eliminates the need for the manual selection of cluster centers and improves clustering efficiency with the experimental results.

1. Introduction

Clustering [1–4] is an unsupervised or semisupervised learning method. This method aims at dividing the samples into different clusters according to the similarity between samples so that the samples in the same cluster are as similar as possible and the samples in different clusters are as dissimilar as possible. Clustering has a wide range of applications, such as image analysis [5], pattern recognition [6], data analysis [7], and wireless sensor networks [8]. Under the wide applications, many clustering methods have emerged, such as K-means [9] and fuzzy c-means (FCM) [10] clustering methods, which are only effective for spherical data and have an inferior effect on nonspherical data. But density-based clustering methods [11] such as density peaks clustering (DPC) [12] did not have this problem. However, DPC still has some drawbacks, so improving the density-based clustering method has great significance.

Aiming at the problem that DPC needs manual participation in selecting cluster centers, Flores et al. [13] proposed a density peaks clustering with a gap-based automatic center detection method. This method calculates a threshold to distinguish between cluster center samples and noncenter samples. Lv et al. [14] proposed a density peak clustering algorithm based on shared nearest neighbor and adaptive cluster center. This method selects the cluster center by narrowing the search range. However, the cluster centers obtained by the above methods are the same as that of DPC. In other words, if DPC cannot achieve good results on some data, the above methods cannot achieve good results.

Aiming at the high time complexity of DPC, Lu et al. [15] proposed a fast distributed density peaks clustering method based on the Z-value index. This method effectively reduces the time complex from $O(n^2)$ to $O(n \cdot \log n)$, but the clustering effect is significantly reduced because the data need to be reduced from multidimensional to one dimension. Xu et al. [16] proposed a fast density peaks clustering method based on spare search. This method ensures the clustering effect and effectively reduces the time complexity below $O(n^2)$, but it still needs to select the cluster centers manually. Although the above methods improve the efficiency of DPC, they also have shortcomings.

Recently, there have been many ways to improve the precision of DPC [17–20]. These methods have more advantages for some data with considerable noise and complex structures. In [17], dense cores are introduced as a representative of the original data to reduce runtimes. The density threshold is used to eliminate the interference from noise samples. However, it has difficulties processing high-

dimensional data. In [18], local density peaks are used to construct a minimum spanning tree to avoid noises and reduce runtimes. However, it cannot automatically determine the number of clusters. In [19], local cores representation dataset is introduced, which avoids noise interference and reduces runtimes by only calculating the graph distance between local cores. Nevertheless, it needs to build a decision diagram. In [20], introducing natural neighbor to find local representations, calculating the adaptive distance between local representations effectively reduces the runtimes. Nevertheless, it cannot automatically determine the cluster centers. Du et al. [21] proposed a k-nearest neighbor DPC method based on principal component analysis. It uses k-nearest neighbor to calculate the sample density and principal component analysis to process high-dimensional data. Nevertheless, it is not effective in dealing with manifold data.

This paper presents a new density peak clustering method (GDPC) that can fast and autodetermine the cluster centers by grouping. Our method aims at grouping the datasets to ensure that sample categories of each group are the same and try to minimize the number of groups. In this way, we no longer need to calculate the similarity between all samples but only between groups. At the same time, we find that the cluster centers usually appear in the densest place of each cluster. We can divide each cluster into core and boundary regions according to the feature that the density from cluster center to cluster edge decreases gradually. The distance between the two core regions is considerable so that we can find the cluster centers.

The rest of this paper is as follows: in Section 2, we review the DPC. Section 3 introduces our proposed method. In Section 4, we do experiments on the effectiveness and efficiency of synthetic datasets and real datasets. Finally, we summarize this article and put forward further challenges.

2. DPC

2.1. Quantities. DPC needs to use the density of the sample (ρ) and the distance between the sample and its nearest high-density sample (δ) when finding the cluster centers and determining the sample labels.

The density of the sample i is the number of other samples within its cutoff distance (d_c) range. The density of the sample i is defined as

$$\rho_i = \sum_{j \in D} \chi\left(d_{ij} - d_c\right), \tag{1}$$

where d_{ij} is the distance between sample i and sample j, and $\chi(x) = 1$ if $x < 0$; otherwise, $\chi(x) = 0$.

The distance between sample i and its nearest high-density sample is defined as

$$\delta_i = \min_{j:\, \rho_j > \rho_i} \left(d_{ij}\right). \tag{2}$$

For the sample i with the highest density, $\delta_i = \max_{j \in D}(d_{ij})$.

2.2. Similarity Matrix. When calculating ρ and δ and determining the sample label, the distance between samples will be calculated many times. To improve efficiency, DPC only calculates the distance between samples once and saves it. It is no longer necessary to recalculate it when it is used again. The similarity matrix is defined as

$$S = \begin{bmatrix} d_{11} & \cdots & d_{1n} \\ \vdots & \ddots & \vdots \\ d_{n1} & \cdots & d_{nn} \end{bmatrix}, \tag{3}$$

where d_{ij} is the distance between sample i and sample j. n is the number of samples.

2.3. Process. After calculating ρ and δ with the above method, we use ρ as abscissa and δ as ordinate to draw a decision diagram, as shown in Figure 1. Figure 1(a) displays the distribution of samples. Figure 1(b) is the corresponding decision diagram. We can find that the upper right corner of the decision diagram indicates the cluster centers.

After selecting the cluster centers, the labels of the remaining samples are the same as that of the nearest high-density sample.

2.4. Time Complexity. The time complexity of DPC has three main parts: (a) calculating ρ requires $O(n^2)$ time complexity; (b) calculating δ needs $O(n^2)$ complexity; and (c) determining labels of noncenter samples demands $O(n)$ time complexity, where n is the size of the dataset. Based on the above three parts, the time complexity of DPC is $O(n^2)$.

3. The Proposed Method

To solve the manual intervention in selecting cluster centers of DPC and the high time complexity of DPC, we proposed GDPC. In GDPC, we isolate the core regions of different clusters to find the cluster centers and use the grouping method to reduce the amount of calculation.

3.1. Grouping. Under the condition of ensuring that the sample labels in each group are the same, the fewer the number of groups, the more efficient our method can be.

We define the distance between the ungrouped sample x_i and the first sample in group g as $||x_i - g||$. We divide sample i into group g if $||x_i - g|| < d_c$. If no group g makes $||x_i - g|| < d_c$, we create a new group. Sample i is the first sample in this group. Algorithm 1 explains the grouping method in detail.

These groups have the following characteristics:

(1) $||x_i - g|| < d_c\,(x_i \in g)$: the distance between the first sample and other samples in the group is less than d_c.

(2) $g_i \cap g_j = \varnothing\,(i \neq j)$: the same sample does not exist in two different groups.

(3) $x_i \in C_j\,(x_i \in g)$, if $g^1 \in C_j$, where C_j is cluster j and g^1 is first sample in group g. All samples in the same group belong to the same cluster.

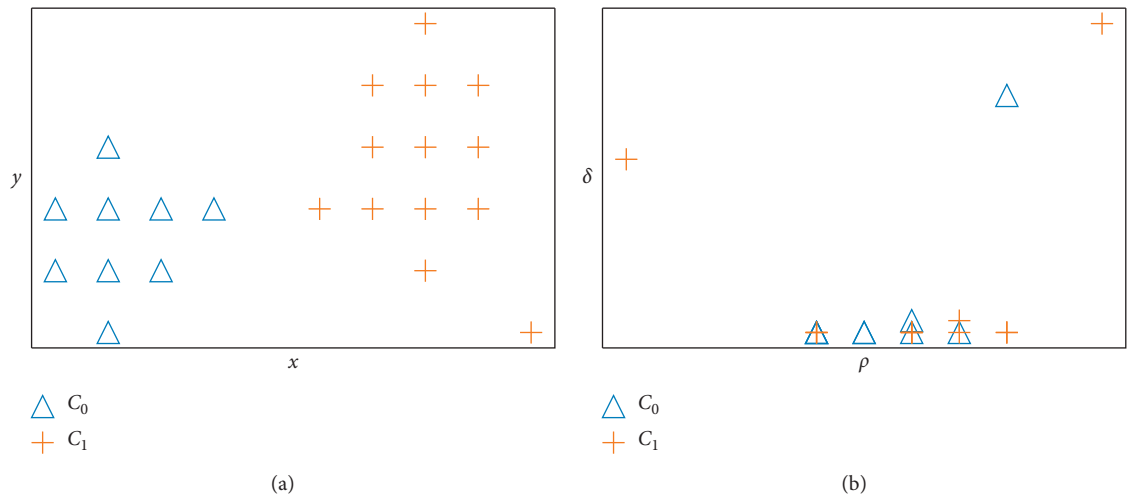

Figure 1: Density peak clustering method. (a) Samples distribution. (b) Decision diagram.

To distinguish whether group i is a core group or a boundary group, we need to use the group density, which is defined as

$$\rho_i = |g_i| - 1. \tag{4}$$

Definition 1. (zero density group). The group with only one sample. If group i is zero density group, $\rho_i = 0$.

Through experience, we first remove groups with zero density and then arrange the remaining group density in descending order. Finally, if the density of the group i is greater than or equal to the density threshold (ρ_t), group g_i is defined as the core group; otherwise, it is defined as the boundary group. The density threshold is defined as

$$\rho_t = f(\lceil 0.7 \times n \rceil - 1), \tag{5}$$

where n is the number of nonzero density group and $f(i)$ is the density of the i-th group after descending sorting.

As shown in Table 1, there are 7 nonzero density groups. The density threshold ρ_t calculated by equation (5) is $f(\lceil 0.7 \times 7 \rceil - 1)$, $f(4) = 2$. Because the density of groups 0, 1, 2, 3, 4, 5 is greater than or equal to ρ_t, they are core groups. The density of groups 6, 7 is less than ρ_t, so it is a boundary group. The density of group 7 is 0, so it is a zero density group.

3.2. Select Cluster Centers and Determine Core Group Sample Labels. When we determine the core group set, we need to find the core region of each cluster through its spatial relationship. Then, we can select the cluster center of each cluster where the density is the densest.

We propose the following definitions.

Definition 2. (key sample). The first sample in each group.

Definition 3. (transitivity). If the distance between any sample in group p and the key sample in group q is less than $2d_c$, p can reach q through transitivity.

The two groups p and q with transitivity satisfy the following equation:

$$||p - q|| \le 2d_c, \tag{6}$$

where $||p - q||$ is the distance between any sample in group p and the key sample in group q. We can combine core groups (join the same cluster) through equation (6).

Figure 2, because $||g_1 - g_2|| \le 2d_c$, so, group 1 can reach group 2 through transitivity. Therefore, there is also transitivity between groups 2 and 3. By transitivity, we can connect group 1, group 2, and group 3, so they belong to the same cluster.

Assume that the density of group 3 is the highest in groups 1, 2, and 3, and group 3 is already in cluster c, whereas groups 2 and 3 are not in any cluster. Because there is transitivity between groups 2 and 3 ($||g_2 - g_3|| \le 2d_c$), and group 3 is in cluster c. So it can be regarded as transitivity between group 2 and cluster c ($||g_2 - c|| \le 2d_c$). After group 2 is added to cluster c, group 1 and cluster c also have transitivity, so group 1 can be added to cluster c.

Algorithm 2 shows the specific steps, where $||o - c||$ is the distance between sample o in core set and any sample in cluster c. g_o is the set of all samples in the group whose first sample is o. o_1 is the first sample in the current core set. Firstly, we select the first sample in the core set as the initial cluster center. Secondly, connect its corresponding group with other core groups through transitivity. The following unconnected sample in the core set is selected as the cluster center when more core groups cannot be connected. Repeat the above steps until all the core groups have been assigned clusters.

3.3. Determine Boundary Group Sample Labels. When selecting the cluster center, we mark two transitive core groups as the same cluster. In this way, we can determine the labels of the core group sample simultaneously when selecting the cluster centers.

(1) **Input:** data set $D = \{x_1, x_2, \ldots, x_n\}$, d_c
(2) **Output:** group set $G = \{g_1, g_2, \ldots, g_m\}$
(3) Create set $G = \varnothing$
(4) **For** each sample x in D **do**
(5) **If** exist $||x - g|| < d_c$ **then**
(6) $g \leftarrow g \cup \{x\}$
(7) **Else**
(8) Create new group (x)
(9) **End if**
(10) **End for**

Algorithm 1: Grouping.

Table 1: Examples of core and boundary groups.

i	0	1	2	3	4	5	6	7
ρ	4	3	3	2	2	2	1	0

(1) **Input:** Core set $O = \{o_1, o_2, \ldots, o_m\}$, Group set $G = \{g_1, g_2, \ldots, g_m\}$, d_c
(2) **Output**: Center set $T = \{t_1, t_2, \ldots, t_k\}$, Cluster set $C = \{c_1, c_2, \ldots, c_k\}$
(3) Create set $C = \varnothing$
(4) **While** exist o in O **do**
(5) / * Merge all core groups * /
(6) Flag = 0
(7) **For** each sample o in O **do**
(8) / * When a core group is added to a cluster or a cluster is created, the transitivity between the remaining core groups and the cluster is judged * /
(9) **If** exist $||o - c|| \leq 2d_c$ **then**
(10) / * transitivity judgment * /
(11) $c \leftarrow c \cup g_o$
(12) $O \backslash o$
(13) Flag = 1
(14) **Break**
(15) **End if**
(16) **End for**
(17) **If** Flag == 0 **then**
(18) / * group o_1 is not transitive to any existing label group * /
(19) Create new cluster (o_1)
(20) $T \leftarrow T \cup o_1$
(21) $O \backslash o_1$
(22) Flag = 1
(23) **End if**
(24) **End while**

Algorithm 2: Select cluster centers and determine core group sample labels.

After finding the cluster centers, we calculate the distance between the boundary group and the core group sample. If the distance between group i and core group sample j is less than $2d_c$, all samples in group i have the same label as sample j, where the distance between group i and sample j is the distance between the key sample in group i and sample j. After that, we need to determine the label of the remaining boundary group sample. We calculate the distance between the key samples in each nonzero density boundary group and each cluster center. We divide the nonzero density boundary group into the cluster corresponding to the nearest cluster center and mark the zero density boundary group samples as noise samples.

3.4. Process. Our method (Algorithm 3) consists of the following five main steps: (1) group the datasets using d_c; (2) select the cluster centers and determine the core group sample labels through transitivity; (3) calculate the distance between the boundary group and the core group sample and determine the boundary group labels with distances less than $2d_c$; (4) calculate the distance between the remaining

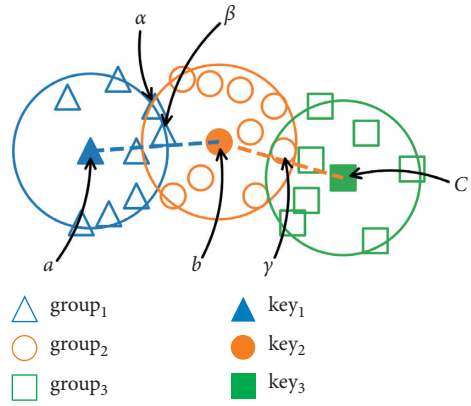

Figure 2: Transitivity.

nonzero density boundary groups and the cluster centers, and determine the labels of the remaining nonzero density boundary group; and (5) mark the remaining zero density group samples as noise samples.

3.5. Example. In this subsection, we introduce the process of our method through an example, as shown in Figure 3 and Table 2.

(1) Group the data: calculate the distance between sample x_1 and the existing group. Since there is no group yet, create a new group g_1, and add sample x_1 to group g_1. Calculate the distance between sample x_2 and the existing group (g_1). Because $||x_2 - g_1|| > d_c$, create a new group g_2 and add sample x_2 to group g_2. Calculate the distance between sample x_3 and the existing group (g_1, g_2). Because $||x_3 - g_1|| > d_c$ and $||x_3 - g_2|| \le d_c$, so add sample x_3 to group g_2. Repeat the above steps until all samples are added to the group. Calculate the density of each group by equation (4). The grouping results and density are shown in Table 2.

(2) Distinguish core or boundary groups: firstly, we sort the groups in descending order of density, and the result is $g_3, g_1, g_2, g_4, g_6, g_9, g_5, g_7, g_8$. Secondly, we remove the zero density group and calculate the density threshold is 1 from equation (5) $(\lceil 0.7 \times 6 \rceil - 1 = 4)$, where 4th group is g_6, and its density is 1. So groups $g_3, g_1, g_2, g_4, g_6, g_9$ are the core groups, and groups g_5, g_7, g_8 are the boundary groups.

(3) Determine core group labels: select the key sample x_4 of group g_3 as the first cluster center, and create a new cluster c_1 to add group g_3 to cluster c_1. Since $\|g_1 - c_1\| \le 2d_c$, add group g_1 to cluster c_1. Because the distances between groups g_2, g_4, g_6, g_9 and cluster c_1 are greater than $2d_c$, take the key sample x_2 of group g_2 as the second cluster center, create a new cluster c_2, and add group g_2 to cluster c_2. Since $\|g_4 - c_1\| > 2d_c$ and $\|g_4 - c_2\| \le 2d_c$, add group g_4 to cluster c_2. Since $\|g_6 - c_1\| > 2d_c$ and $\|g_6 - c_2\| \le 2d_c$, add group g_6 to cluster c_2. Since $\|g_9 - c_1\| > 2d_c$ and $\|g_9 - c_2\| \le 2d_c$, add group g_9 to cluster c_2.

(4) Determine boundary group labels: because $\|g_5 - c_1\| > 2d_c$ and $\|g_5 - c_2\| > 2d_c$, group g_5 is not processed temporarily. Because $\|g_7 - c_1\| > 2d_c$ and $\|g_7 - c_2\| > 2d_c$, group g_7 is not processed temporarily. Since $\|g_8 - c_1\| \le 2d_c$, add group g_8 to cluster c_1. Next, we divide the nonzero boundary groups into a cluster corresponding to the nearest cluster center. In this example, there is no nonzero density boundary group, so this step ends here.

(5) Determine noise samples: in the previous step, the zero density boundary groups not divided into cluster are considered noise samples. In this example, samples x_8 and x_{14} in groups g_5 and g_7 are marked as noise samples.

After completing the above steps, the clustering process is over. The clustering results are shown in Table 3. This example generates two clusters, including 10 samples in cluster c_1, 10 samples in cluster c_2, and 2 noise samples.

3.6. Time Complexity. The time complexity of our method has four main parts:

It is assumed that the dataset with n samples is divided into a core groups and b boundary groups.

(a) Grouping requires $O((a + b) \cdot n)$ time complexity

(b) Sorting core group density needs $O(a \cdot \log a)$ time complexity

(c) Finding the cluster centers and determining the core group sample labels need $O(a \cdot n)$ time complexity

(d) Determining labels of noncenter samples demands $O(b \cdot n + b)$ time complexity

(1) **Input**: data set $D = \{x_1, x_2, \ldots, x_n\}$, d_c
(2) **Output**: center set $T = \{t_1, t_2, \ldots, t_k\}$, cluster set $C = \{c_1, c_2, \ldots, c_k\}$
(3) Grouping data using Algorithm 1
(4) Select the cluster centers and determine the core group sample labels using Algorithm 2
(5) Calculate the distance between the boundary group and the core group sample and determine boundary group labels with distances less than $2d_c$
(6) Determine the label of the remaining nonzero density boundary group sample
(7) Mark the remaining zero density group samples as noise samples

Algorithm 3: GDPC.

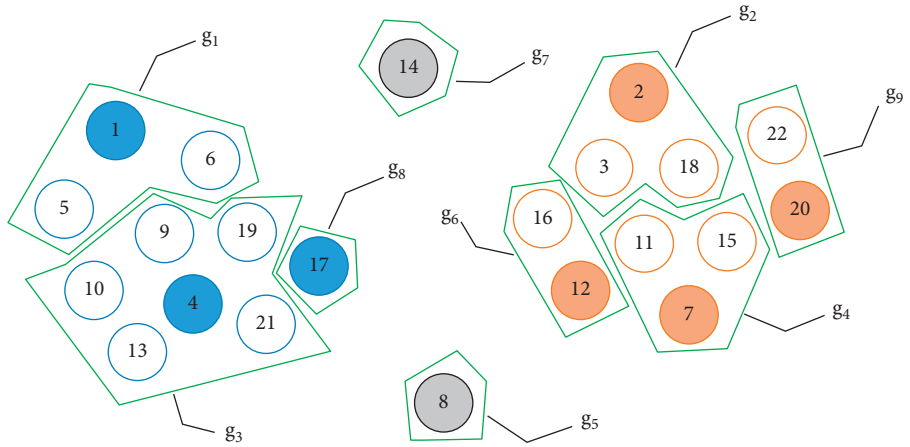

Figure 3: Clustering example.

Table 2: The grouping results.

Group	Sample (s)	Density
g_1	x_1, x_5, x_6	2
g_2	x_2, x_3, x_{18}	2
g_3	$x_4, x_9, x_{10}, x_{13}, x_{19}, x_{21}$	5
g_4	x_7, x_{11}, x_{15}	2
g_5	x_8	0
g_6	x_{12}, x_{16}	1
g_7	x_{14}	0
g_8	x_{17}	0
g_9	x_{20}, x_{22}	1

Table 3: Clustering results of example.

Clusters	Samples	Instances
c_1	$x_4, x_9, x_{10}, x_{13}, x_{19}, x_{21}, x_1, x_5, x_6, x_{17}$	10
c_2	$x_2, x_3, x_{18}, x_7, x_{11}, x_{25}, x_{12}, x_{16}, x_{20}, x_{22}$	10
Noises	x_8, x_{14}	2

Table 4: Properties of synthetic datasets.

Dataset	Instances	Dimensions	Clusters
ThreeCircles	299	2	3
Lineblobs	266	2	3
Spiral	312	2	3
Compound	399	2	6

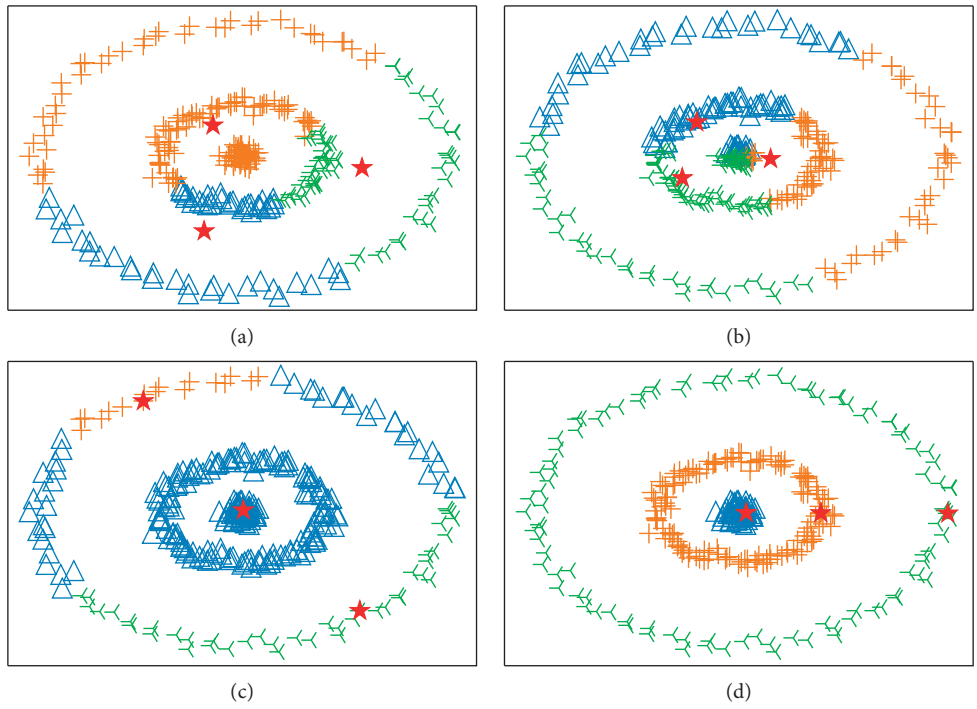

Figure 4: Clusters obtained by K-means, FCM, DPC, and GDPC over ThreeCircles dataset. (a) K-means; (b) FCM; (c) DPC; (d) GDPC $d_c = 0.05$.

Based on the above four parts, the time complexity of our method is $O((a+b)\cdot n)$, where $(a+b) < n$. Therefore, the time complexity of our method is less than $O(n^2)$.

4. Experimental Evaluation

This section has carried out precision and efficiency experiments and has exposed our method source code and synthetic data (download URL=[https://github.com/yongbiaoLi/GDPC.git]), respectively.

4.1. Precision Experiment

4.1.1. Synthetic Datasets. In this section, we used K-means, FCM, DPC, and GDPC to experiment in ThreeCircles, Lineblobs, Spiral, and Compound [22] synthetic datasets. Table 4 shows specific information about the datasets.

From Figures 4, 5, 6, and 7, the cluster centers obtained by K-means and FCM are not necessarily samples in the data, but those obtained by DPC and GDPC are samples. For nonspherical data such as Figures 4 and 6, K-means and FCM cannot obtain the correct cluster centers and result. For data such as Figures 5 and 7, which combine spherical and nonspherical data, K-means and FCM can only obtain partially correct cluster centers and results. DPC can obtain some clustering centers and results of data such as Figures 5 and 7 because DPC is effective not only for nonspherical data but also for spherical data. However, from Figures 4, 5, 6, and 7, because DPC is stringent in distributing nonspherical data, it is impossible to obtain correct cluster centers and results. Our method is not affected by these distributions.

4.1.2. Real-World Datasets. In this section, we use Zoo [23], Thyroid [23], Ecoli [24], Machine [23], Hayes-Roth [23], Sobar-72 [25], Segment [23], and Pendigits [23] real-world data (details are shown in Table 5) to experiment with K-means, FCM, DPC, and our algorithm.

From Table 6, we can see that ARI [26, 27], NMI [28], and homogeneity [29] of GDPC are all higher than those of K-means, FCM, and DPC on Zoo and Thyroid datasets. On machine, Sobar-72 and Sobar-72 datasets, GDPC has two higher metrics than K-means, FCM, and DPC. On Ecoli, Segment, and Pendigits data, only one metric of GDPC is the best. In parentheses is the value of the parameter d_c.

The experimental results show that although GDPC is inferior to K-means, FCM, and DPC in some datasets and metrics, its performance is better than that in general.

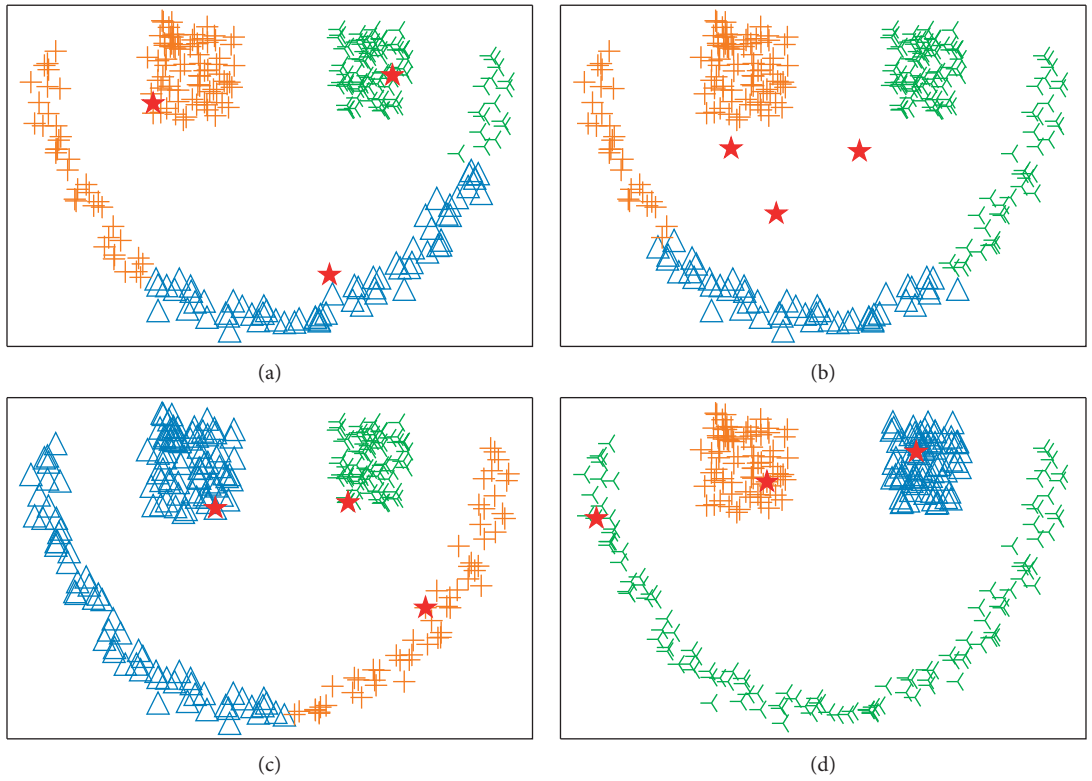

Figure 5: Clusters obtained by K-means, FCM, DPC, and GDPC over Lineblobs dataset. (a) K-means; (b) FCM; (c) DPC; (d) GDPC $d_c = 0.089$.

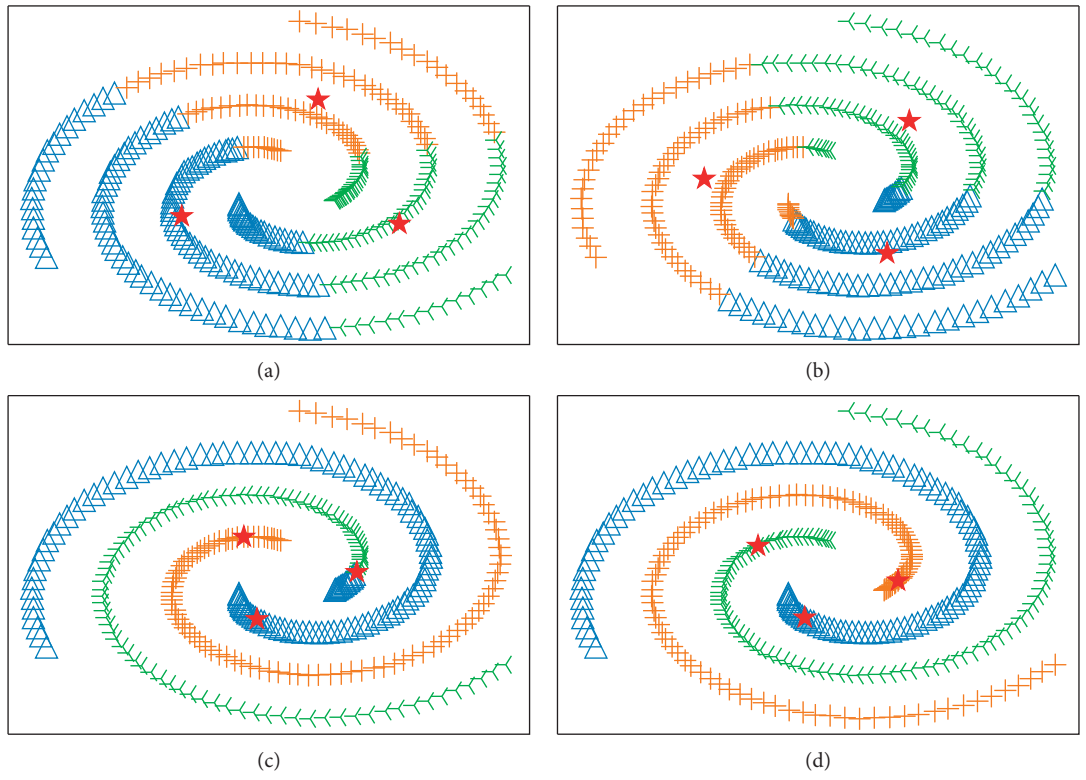

Figure 6: Clusters obtained by K-means, FCM, DPC, and GDPC over Spiral dataset. (a) K-means; (b) FCM; (c) DPC; (d) GDPC $d_c = 0.058$.

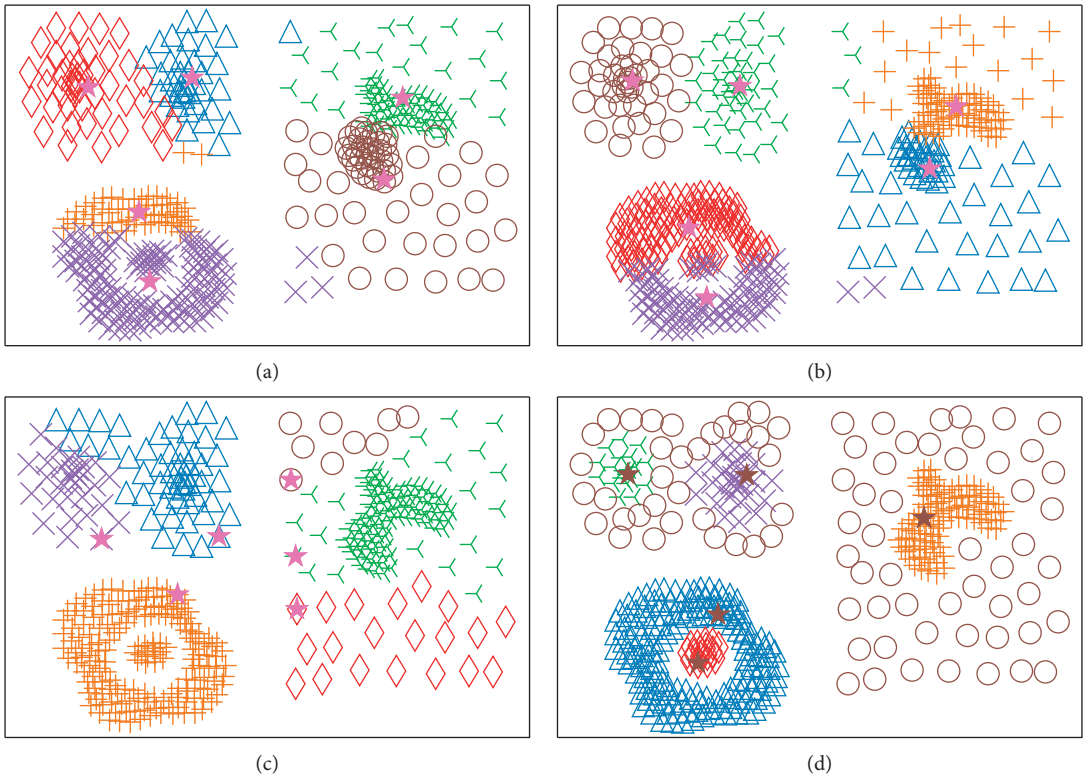

Figure 7: Clusters obtained by K-means, FCM, DPC, and GDPC over Compound dataset. (a) K-means; (b) FCM; (c) DPC; (d) GDPC $d_c = 0.026$.

Table 5: Properties of real-world datasets.

Dataset	Instances	Dimensions	Clusters
Zoo	101	16	7
Thyroid	215	5	3
Ecoli	336	8	8
Machine	209	7	8
Hayes-Roth	132	4	3
Sobar-72	72	19	2
Segment	2310	18	7
Pendigits	20000	16	26

Table 6: Comparison of the evaluation metrics of the four algorithms on the real-world datasets.

Dataset	Algorithm	ARI	NMI	Homogeneity
Zoo	K-means	0.42069	0.64396	0.63305
	FCM	0.36469	0.62433	0.66487
	DPC	0.3056	0.42223	0.32649
	GDPC (0.718)	**0.80158**	**0.84129**	**0.84208**
Thyroid	K-means	0.54707	0.39504	0.40321
	FCM	0.4413	0.34344	0.37
	DPC	0.52063	0.47818	0.40368
	GDPC (0.045)	**0.68386**	**0.55497**	**0.57709**
Ecoli	K-means	0.43884	**0.61493**	**0.70525**
	FCM	0.34935	054486	0.6399
	DPC	0.45087	0.60743	0.63245
	GDPC (0.093)	**0.63227**	0.60598	0.53881
Machine	K-means	0.06195	0.28402	0.32626
	FCM	0.40954	0.44839	**0.54375**
	DPC	0.38929	0.43189	0.38985
	GDPC (0.106)	**0.50151**	**0.53098**	0.42842

Table 6: Continued.

Dataset	Algorithm	ARI	NMI	Homogeneity
Hayes-Roth	K-means	**0.03731**	0.05575	0.05575
	FCM	-0.01491	0.0	0.0
	DPC	0.00591	0.13145	0.08757
	GDPC (0.166)	0.03223	**0.25694**	**0.50898**
Sobar-72	K-means	**0.26671**	0.2106	0.22378
	FCM	**0.26671**	0.2106	0.22378
	DPC	-0.07399	0.06523	0.05014
	GDPC (0.629)	0.18569	**0.31803**	**0.50194**
Segment	K-means	0.36336	0.46123	0.44182
	FCM	**0.50632**	**0.61017**	0.60989
	DPC	0.24025	0.52623	0.40106
	GDPC (0.089)	0.43038	0.58043	**0.75747**
Pendigits	K-means	0.35528	0.53178	0.52341
	FCM	0.43647	0.57868	0.53033
	DPC	**0.55422**	**0.72433**	0.69383
	GDPC (0.15)	0.50352	0.60201	**0.69527**

The meaning of the bold value is to emphasize that the value is the best result of the four algorithms on the same dataset and metric.

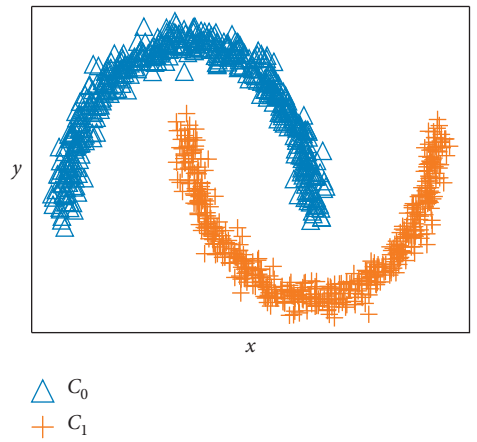

Figure 8: Moons.

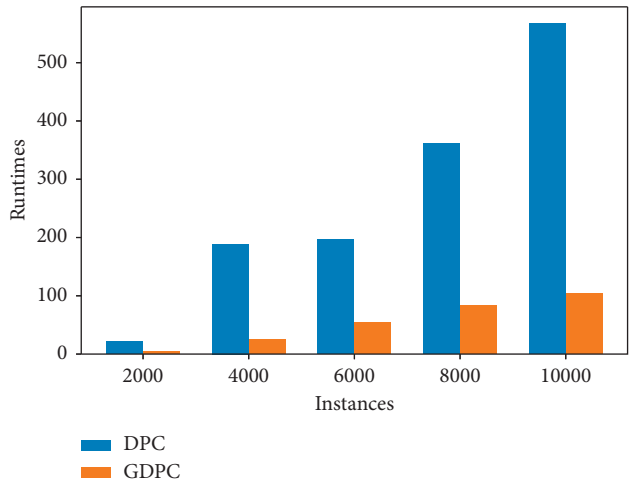

Figure 9: Experiments with different instances.

Table 7: Runtimes (*s*) of real-world datasets.

Dataset	DPC	GDPC
Zoo	0.374	**0.093**
Thyroid	0.401	**0.234**
Ecoli	0.875	**0.462**
Machine	0.398	**0.265**
Hayes-Roth	0.375	**0.234**
Sobar-72	0.37	**0.031**
Segment	1576.384	**1046.281**
Pendigits	5294.01	**3595.974**

The meaning of the bold value is to emphasize the best result of the two algorithms on the same dataset.

Table 8: Results of different parameters on the moon datasets.

Parameter d_c	Runtimes (s)	ARI	NMI	Homogeneity
0.05	277.5	0.99799	0.99429	0.99429
0.1	49.718	1.0	1.0	1.0
0.15	12.125	1.0	1.0	1.0
0.2	2.3125	0.0	0.0	0.0
0.25	1.4531	0.0	0.0	0.0
0.3	1.31375	0.0	0.0	0.0

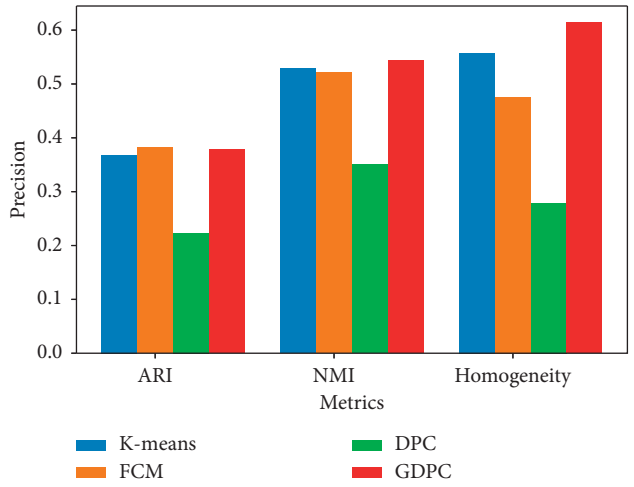

Figure 10: Performance comparison of different methods.

4.2. Efficiency Experiment

4.2.1. Synthetic Datasets. In this section, we use the moons dataset to test the efficiency of DPC and our methods,. The distribution of the moons dataset is shown in Figure 8. From Figure 9, we can see that, with the increasing amount of data, the time required for DPC increases significantly, and our method is better than DPC, and it becomes more and more apparent.

4.2.2. Real-World Datasets. In this section, we use the real-world datasets to test the efficiency of DPC and our methods. As shown in Table 7, our method has always been more efficient than DPC. The larger the d_c, the faster our method is. The more significant the amount of data, the greater the gap between our method and DPC.

4.3. Influence of Parameter $\mathbf{d_c}$. In this section, we use the moons dataset to test the influence of parameter d_c on the clustering results. As shown in Table 8, d_c = 0.05, 0.2, 0.25, and 0.3 cannot make the clustering results completely correct. As d_c increases from 0.05 to 0.3, the running time decreases from 277.5 seconds to 1.34375 seconds.

The influence of d_c on clustering results mainly has two aspects. (a) Efficiency: the larger the d_c, the fewer the number of groups after grouping, and the faster the clustering speed. (b) Precision: too small d_c will lead to grouping errors, which will reduce the precision. Too small d_c will lead to too many noise samples. Combining the above two aspects, we select the largest d_c to ensure that all samples in each group belong to the same cluster.

According to the synthetic data experiment, d_c is generally half of the distance between the core regions of the two nearest clusters. Because high-dimensional real data cannot

be observed, it is challenging to select d_c for real data. How to obtain the value of d_c in real data is also a limitation of this method.

5. Conclusion

This paper presents a new method to improve DPC. We divide the data into minor groups while ensuring that the sample labels in each group are the same, so we only need to calculate the similarity between groups to reduce the amount of calculation. At the same time, we isolate the core regions of different clusters, which makes it easy for us to find the cluster center at the densest location. Among many methods to improve DPC efficiency, the automatic selection of cluster centers is not solved. Our method not only improves DPC efficiency but also solves the problem of automatic selection of cluster centers.

GDPC has a good effect on some spherical and nonspherical data, but it does not perform well in some complex data with noise samples. Because GDPC only uses one d_c in one data, it is difficult to achieve a good grouping result in some complex data, resulting in unsatisfactory clustering results and efficiency. In addition, how to obtain the value of d_c in real data is also a limitation of this method. These are also where we need to improve.

The advantages of the density-based clustering method in nonspherical data are difficult to replace. In the future, we still need to consider how to improve the efficiency under reducing parameters.

6. Application

Clustering can be applied to wireless sensor data annotation. We apply the clustering method to the activity data [30] of the elderly to identify the four states of the elderly: (a) sit on the bed; (b) sit on a chair; (c) lying; and (d) ambulating. The data were obtained by Torres et al. through a battery-free wearable sensor. As shown in Figure 10, the experimental results show that our proposed method has certain advantages in wireless sensor data annotation.

Data Availability

Previously reported data were used to support this study and are available at http://archive.ics.uci.edu/ml. These prior studies are cited at relevant places within the text as references.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

The National Natural Foundation Project of China: Research on Retina Image Segmentation and Aided Diagnosis Technology Based on Deep Learning (61962054) supported this study.

References

[1] R. Xu and D. WunschII, "Survey of clustering algorithms," *IEEE Transactions on Neural Networks*, vol. 16, no. 3, pp. 645–678, 2005.

[2] D. Xu and Y. Tian, "A comprehensive survey of clustering algorithms," *Annals of Data Science*, vol. 2, no. 2, pp. 165–193, 2015.

[3] P. Berkhin, "A survey of clustering data mining techniques," in *Grouping Multidimensional Data*, pp. 25–71, Springer, 2006.

[4] P. Rai and S. Singh, "A survey of clustering techniques," *International Journal of Computer Application*, vol. 7, no. 12, pp. 1–5, 2010.

[5] Li Chen, K. Frank, and J. Zhang, "A review of clustering methods in microorganism image analysis," *Information Technology in Biomedicine*, vol. 13-25, 2021.

[6] N. M and T. R. Prajwala, "A comprehensive overview of clustering algorithms in pattern recognition," *IOSR Journal of Computer Engineering*, vol. 4, no. 6, pp. 23–30, 2012.

[7] Z. Cui, X. Jing, P. Zhao, W. Zhang, and J. Chen, "A new subspace clustering strategy for ai-based data analysis in iot system," *IEEE Internet of Things Journal*, vol. 8, no. 16, pp. 99–2021.

[8] S. Amin, A. Taherkordi, Ø. Haugen, and E. Frank, "Clustering objectives in wireless sensor networks: a survey and research direction analysis," *Computer Networks*, vol. 180, Article ID 107376, 2020.

[9] J. MacQueen, "Some methods for classification and analysis of multivariate observations," in *Proceedings of the fifth Berkeley symposium on mathematical statistics and probability*, pp. 281–297, Oakland, CA, USA, December 1967.

[10] J. C. Bezdek, R. Ehrlich, and W. Full, "Fcm: The Fuzzy C-Means Clustering Algorithm," *Computers and geosciences*, vol. 10, no. 2-3, pp. 191–203, 1984.

[11] P. Bhattacharjee and P. Mitra, "A Survey of Density Based Clustering Algorithms," *Frontiers of Computer Science*, vol. 15, no. 1, pp. 1–27, 2021.

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

[13] K. G. Flores and S. E. Garza, "Density peaks clustering with gap-based automatic center detection," *Knowledge-Based Systems*, vol. 206, 2020.

[14] Yi Lv, M. Liu, and X. Yue, "Fast searching density peak clustering algorithm based on shared nearest neighbor and adaptive clustering center," *Symmetry*, vol. 12, no. 12, 2014.

[15] J. Lu, Y. Zhao, K.-L. Tan, and Z. Wang, "Distributed density peaks clustering revisited," *IEEE Transactions on Knowledge and Data Engineering*, vol. 1-1, 2020.

[16] X. Xu, S. Ding, Y. Wang, L. Wang, and W. Jia, "A fast density peaks clustering algorithm with sparse search," *Information Sciences*, vol. 554, pp. 61–83, 2021.

[17] D. Cheng, J. Huang, S. Zhang, X. Zhang, and X. Luo, "A novel approximate spectral clustering algorithm with dense cores and density peaks," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 40, pp. 1–13, 2021.

[18] D. Cheng, Q. Zhu, J. Huang, Q. Wu, and L. Yang, "Clustering with local density peaks-based minimum spanning tree," *IEEE Transactions on Knowledge and Data Engineering*, vol. 33, no. 2, pp. 374–387, 2019.

[19] D. Cheng, S. Zhang, and J. Huang, "Dense members of local cores-based density peaks clustering algorithm," *Knowledge-Based Systems*, vol. 193, Article ID 105454, 2020.

[20] D. Cheng, Q. Zhu, J. Huang, L. Yang, and Q. Wu, "Natural neighbor-based clustering algorithm with local representatives," *Knowledge-Based Systems*, vol. 123, Article ID 238253, 2017.

[21] M. Du, S. Ding, and H. Jia, "Study on density peaks clustering based on k-nearest neighbors and principal component analysis," *Knowledge-Based Systems*, vol. 99, pp. 135–145, 2016.

[22] C. T. Zahn, "Graph-theoretical methods for detecting and describing gestalt clusters," *IEEE Transactions on Computers*, vol. C-20, no. 1, pp. 68–86, 1971.

[23] D. Dua and C. Graff, *UCI Machine Learning Repository*, https://archive.ics.uci.edu/ml [Online]. Available:, 2017.

[24] P. Horton and K. Nakai, "A probabilistic classification system for predicting the cellular localization sites of proteins," *Proceedings International Conference on Intelligent Systems for Molecular Biology*, vol. 4, pp. 109–115, 1996.

[25] R. Machmud and A. Wijaya, "Behavior determinant based cervical cancer early detection with machine learning algorithm," *Advanced Science Letters*, vol. 22, no. 10, pp. 3120–3123, 2016.

[26] S. Douglas, "Properties of the hubert-arable adjusted rand index," *Psychological Methods*, vol. 9, no. 3, Article ID 386, 2004.

[27] J. M. Santos and M. Embrechts, "On the use of the adjusted rand index as a metric for evaluating supervised classification," in *Proceedings of the International Conference on Artificial Neural Networks*, pp. 175–184, Limassol, Cyprus, September 2009.

[28] A. F. McDaid, D. Greene, and N. Hurley, "Normalized mutual information to evaluate overlapping community finding algorithms," 2013, https://arxiv.org/abs/1110.2515.

[29] M. Sato-Ilic, "On evaluation of clustering using homogeneity analysis," in *Proceedings of the Smc 2000 conference proceedings. 2000 ieee international conference on systems, man and cybernetics.'cybernetics evolving to systems, humans, organizations, and their complex interactions, (cat. no. 0)*, pp. 3588–3593, Nashville, TN, USA, October 2000.

[30] R. L. S. Torres, D. C. Ranas- inghe, Q. Shi, and A. P. Sample, "Sensor enabled wearable rfid technology for mitigating the risk of falls near beds," in *Proceedings of the 2013 IEEE International Conference on RFID (RFID)*, pp. 191–198, Orlando, FL, USA, April 2013.