IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received May 20, 2020, accepted June 26, 2020, date of publication June 30, 2020, date of current version July 20, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.3006069

Effective Density Peaks Clustering Algorithm Based on the Layered K-Nearest Neighbors and Subcluster Merging

CHUNHUA REN[1,2], LINFU SUN[1], YANG YU[1], AND QISHI WU[1,3], (Member, IEEE)
[1]School of Information Science and Technology, Southwest Jiaotong University, Chengdu 611756, China
[2]School of Computer and Information Engineering, Yibin University, Yibin 644000, China
[3]Big Data Center, New Jersey Institute of Technology, Newark, NJ 07101, USA

Corresponding author: Chunhua Ren (418327014@qq.com)

This work was supported in part by the National Key Research and Development Program of China under Grant 2017YFB1400303.

ABSTRACT Density peaks clustering (DPC) algorithm is a novel density-based clustering algorithm, which is simple and efficient, is not necessary to specify the number of clusters in advance, and can find any nonspherical class clusters. However, DPC relies heavily on the calculation methods of the cutoff distance threshold and local density and cannot analyze complex manifold data, especially datasets with uneven density distribution and multiple peaks in the same cluster. To solve these problems, we propose an improved density peaks clustering algorithm based on the layered k-nearest neighbors and subcluster merging (LKSM_DPC). First, we redefine the local density calculation method using the layered k-nearest neighbors. To adapt to datasets with different densities, the k-nearest neighbors are divided into multiple layers. Second, for the multiple peaks in the same cluster problem, we design a new mechanism to calculate the similarity of subclusters based on the idea of shared neighbors and Newton's law of gravitation, and a subcluster merging strategy is proposed. To prove the effectiveness of our algorithm, we compare the LKSM_DPC with K-means, DBSCAN, DPC, and DPC derivatives for 24 datasets. A large number of experiments demonstrate that our algorithm can often outperform other algorithms.

INDEX TERMS Density peaks clustering, uneven density distribution, multiple peaks, k-nearest neighbors, shared neighbors, the law of gravitation, subcluster merging.

I. INTRODUCTION

Clustering is one of the most important techniques in data mining. This technique gathers data with similar characteristics into a cluster, and there are significant differences among different clusters [1], [2]. It is widely used in machine learning, information security, data mining, and other research fields [3], [4]. Clustering algorithms are usually structured in several categories: partition-based clustering, grid-based clustering, hierarchical clustering, model-based clustering, and density-based clustering algorithms [5].

The most famous partition-based algorithm is K-means [6], which is required to specify the number of clusters and form clusters through iterative objective functions. Density clustering is a generic clustering algorithm, which can find datasets of arbitrary shapes, does not need to specify the number of clusters in advance, and is not sensitive to noise data. DBSCAN is a case in the density-based clustering algorithm [7], which clusters datasets of arbitrary shapes and can detect outliers, but it is highly limited by the setting of two parameters. In 2014, Rodriguez and Laio proposed a novel fast search and density peaks clustering algorithm (DPC) in science [8], which only uses one parameter. First, DPC is based on two assumptions: (1) the density of the cluster center is higher than that of the neighboring points, and (2) the distance between the cluster center and the higher density point is relatively large. Second, the local density ρ and distance δ are calculated using the cutoff distance. Then, a 2-D decision graph of the clustering center is constructed. Finally, the remaining points are allocated according to the clustering center.

Although DPC is simple and fast, it has several main disadvantages: (1) the clustering performance of DPC is mainly affected by the local density ρ and distance δ; and (2) DPC

The associate editor coordinating the review of this manuscript and approving it for publication was Feng Xia.

have difficulties processing data with complex structures, such as datasets with uneven density distribution and complex manifold datasets.

To overcome the above defects, numerous DPC derivatives have been proposed by many scholars.

Numerous researchers have studied the influence of the local density ρ and distance δ measurements on clustering. Du *et al.* [9] proposed a density peaks clustering algorithm based on the k-nearest neighbors (DPC-KNN). The local density of the DPC-KNN is in the form of a Gaussian kernel, which is available from the average distance of the k-nearest neighbors. Xie *et al.* [10] proposed a DPC based on the fuzzy weighted k-nearest neighbors (FKNN-DPC). She redesigned the local density calculation method, which was calculated using the sum of the distances among the k-nearest neighbors, and adopted a new allocation strategy for the remaining points. Liu *et al.* [11] put forward the shared nearest neighbors-based density peaks clustering algorithm (SNN-DPC), which redefined the local density using the concepts of shared neighbor similarity and the distance from the nearest larger density point. Li and Tang [12] presented a comparative density peaks clustering algorithm (CDP). He redefined the local density calculation method for the mutual k-nearest neighbors and defined a novel distance measure using the geodesic distance. Cheng *et al.* [13] proposed a natural neighbor based density peaks clustering algorithm (NaNDP), which reassessed the local density of each sample point by using the maximum number of natural neighbors as the k-nearest neighbors. Yaohui *et al.* [14] designed a new adaptive density peak clustering method based on the K-nearest neighbors (ADPC-KNN) and redefined the local density calculation method. The ADPC-KNN used the distribution information of the k-nearest neighbors and the parameter *dc* to calculate local density ρ. Du *et al.* [15] proposed a novel density peaks clustering method. First, he provided a new option based on using the sensitivity of the local density for the local density and then redefined δ based on a new density-adaptive metric. Wu *et al.* [16] proposed a density peaks clustering method with a symmetric neighborhood relationship (DPC-SNR), where the local densities of each point are calculated using the reverse k-nearest neighbors and similar clusters are aggregated using the symmetric neighborhood graph. Sun *et al.* [17] considers that the computational methods of the local density and the distance measure are simple. He described an adaptive density peaks clustering method with Fisher's linear discriminant (ADPC-FLD) and designed the local density using Pearson's correlation coefficient. Jiang *et al.* [18] proposed a density peaks clustering based on the k-nearest neighbors (DPC-KNN), which integrated the idea of the k-nearest neighbors into the formula for equation δ. Mehmood *et al.* [19] proposed a clustering using fast search and finding the density peaks via the heat diffusion (CFSFDP-HD) algorithm. This algorithm conducted kernel density estimation based on the heat diffusion in an infinite domain. Parmar *et al.* [20] proposed a residual error-based density peak clustering (REDPC) algorithm, where the local densities are calculated using the residual error.

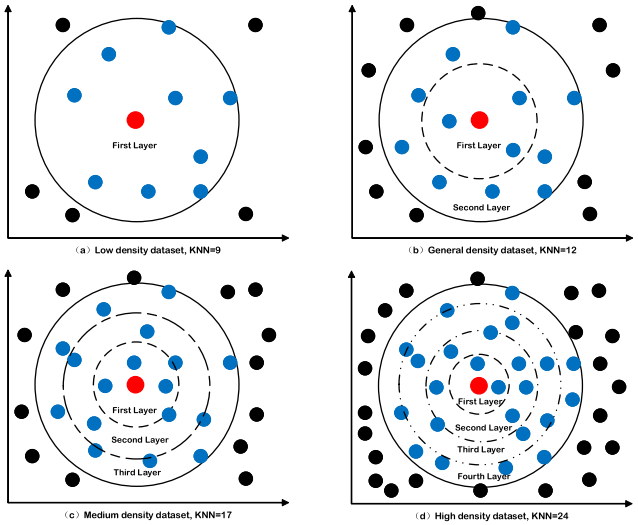


FIGURE 1. Local density diagrams of the layered k-nearest neighbors. (a) The k-nearest neighbors of a low-density dataset. (b) The k-nearest neighbors of a general density dataset. (c) The k-nearest neighbors of a medium-density dataset. (d) The k-nearest neighbors of a high-density dataset.

Many researchers have addressed the problem that the DPC has difficulties dealing with complex structured datasets. Zhuo *et al.* [21] confronted the uneven distribution within local clusters and proposed a density peaks clustering algorithm employing a hierarchical strategy (HCFS). The HCFS used a new mechanism to measure the similarity and connectivity of subclusters, which combined highly similar and interconnected subclusters into a cluster. Wang and Zhu [22] proposed a density peaks clustering method based on the local minimal spanning tree (DPC-LMST), which utilized the subcluster merging factor (SCMF) to aggregate similar subclusters. Xu *et al.* [23] addressed the problem that there are multiple density peaks in one cluster and proposed a density peaks clustering algorithm with a merging strategy. First, he utilized the support vectors to calculate the feedback values and then recursively merged clusters according to the feedback values. Parmar *et al.* [24] proposed a feasible residual error-based density peak clustering algorithm (FREDPC). He not only redefined the local density through the residual error computation but also designed a fragment merging strategy based on residual fragments. Cheng *et al.* [25] addressed the problem that DPC cannot process manifold datasets. He proposed an improved density peaks clustering algorithm based on shared-neighbors between local cores (LORE-DP) and redefined natural neighbor-based density and the newly defined graph-based distance. Qiao *et al.* [26] studied the problem that DPC is not highly effective for the division of unevenly distributed data. He proposed boundary detection-based density peaks clustering (BDDPC) and introduced a new indicator named the asymmetry measure that

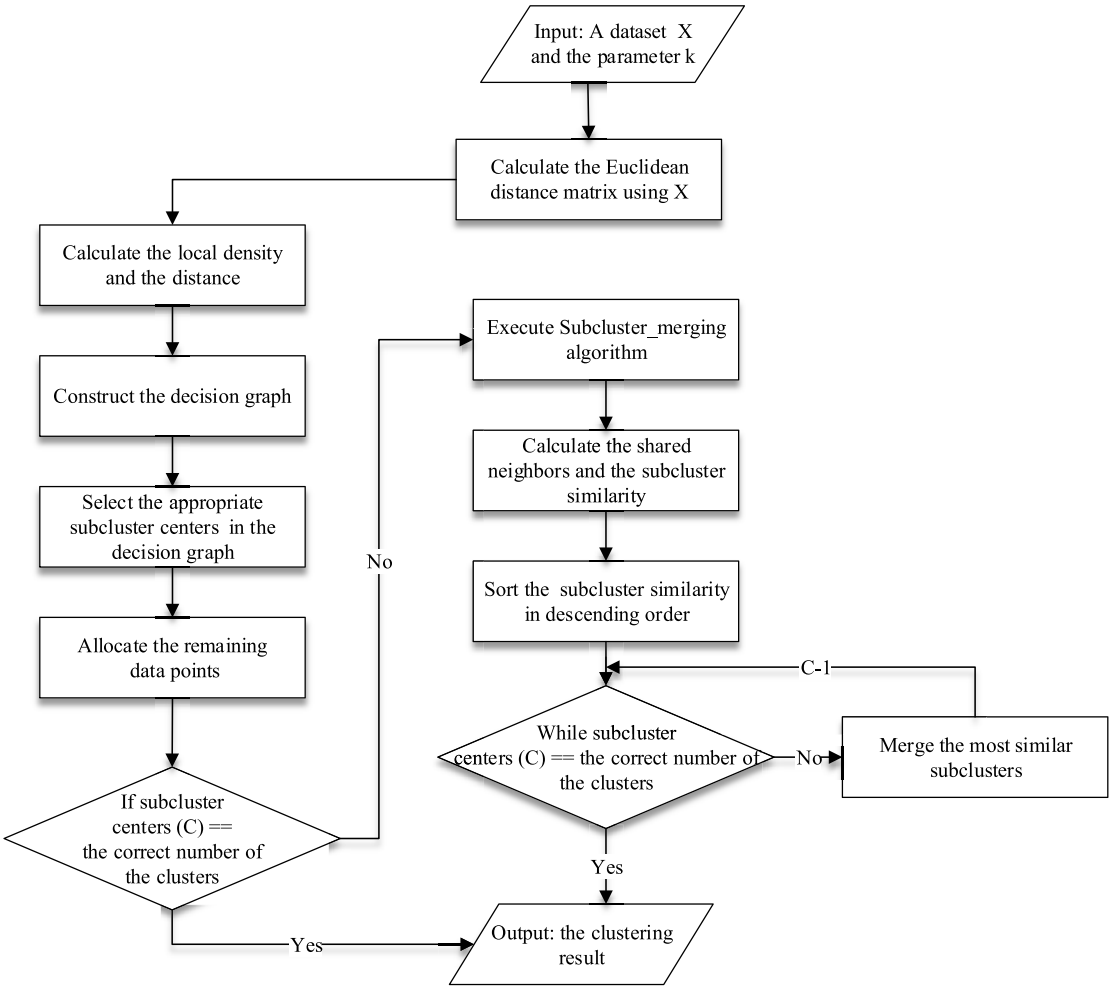


FIGURE 2. A detailed flowchart of the LKSM_DPC algorithm.

enhanced the ability to find boundary points. For complex datasets with irregular shapes, Jiang *et al.* [27] proposed a density fragment clustering without peaks algorithm (DFC), implemented density fragment generation, and density fragment aggregation.

Although the literature has made contributions to the improvement of DPC, there are still several problems: (1) most scholars used the idea of the k-nearest neighbors to calculate the local density, but few people considered the distribution of these k points, especially when the data density distribution is uneven; and (2) in a 2-D decision graph, it is difficult to determine the real cluster center, especially when there are multiple peaks in a cluster.

To solve the above problems, in this paper, we proposed a novel density peaks clustering algorithm based on the layered k-nearest neighbors and subcluster merging (LKSM_DPC). First, to adapt to different density datasets, we divided the k-nearest neighbors into several layers. Second, to address the complex multiple peaks in one cluster problem, we designed a new subcluster similarity measurement mechanism, and a new subcluster merging strategy is proposed.

The rest of this paper is organized as follows. Section II briefly describes the literature related to DPC and three other DPC derivatives. Section III introduces our proposed algorithm in detail. The comparison and analysis of the experiments are provided in Section IV. Finally, a summary and conclusion are given in Section V.

II. RELATED WORKS

To better describe our algorithm, several related DPC algorithms (DPC, DPC-KNN, FKNN-DPC, and DPCSA) are briefly described as follows, and the sources of 4 algorithms are shown in Table 1.

TABLE 1. Sources of the four algorithms.

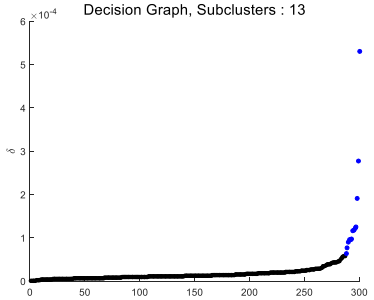
Algorithm	Source	Author	Year
DPC [8]	Science	A. Rodriguez, et al.	2014
DPC-KNN [9]	Knowledge-Based Systems	M. Du, et al.	2016
FKNN-DPC [10]	Information Sciences	J. Xie, et al.	2016
DPCSA [28]	IEEE Access	D. Yu, et al.	2019

$$\begin{bmatrix} 0 & 0.0163 & 0.0561 & 0.0569 & 0.0968 & \cdots \\ 0.0163 & 0 & 0.0551 & 0.0552 & 0.0981 & \cdots \\ 0.0561 & 0.0551 & 0 & 0.0025 & 0.0433 & \cdots \\ 0.0569 & 0.0552 & 0.0025 & 0 & 0.0438 & \cdots \\ 0.0968 & 0.0981 & 0.0433 & 0.0438 & 0 & \cdots \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \end{bmatrix}$$

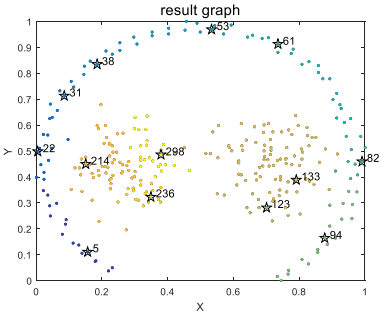
(a)

$$\begin{matrix} \rho & \delta \end{matrix}$$
$$\begin{bmatrix} 7.1598E-04 & 0.0163 \\ 7.2593E-04 & 0.0551 \\ 7.6570E-04 & 0.0433 \\ 7.6570E-04 & 0.0025 \\ 7.6570E-04 & 0.2669 \\ \cdots & \cdots \end{bmatrix}$$

(b)



(c)

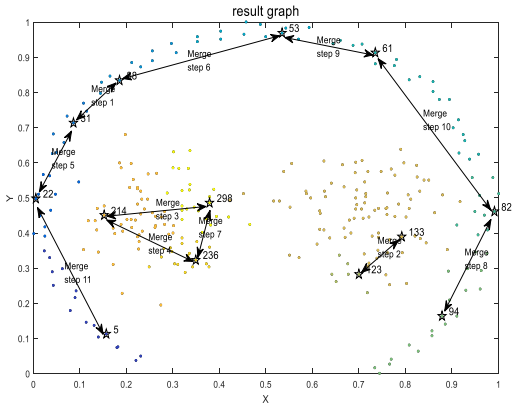


(d)

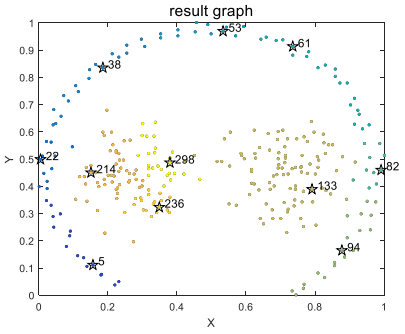
Merge process	Center 1	Center 2	Similarity
Merge step 1	31	38	2.67E-06
Merge step 2	123	133	2.38E-06
Merge step 3	214	298	2.38E-06
Merge step 4	214	236	2.28E-06
Merge step 5	22	31	1.09E-06
Merge step 6	38	53	1.00E-06
Merge step 7	236	298	9.70E-07
Merge step 8	82	94	9.62E-07
Merge step 9	53	61	8.57E-07
Merge step 10	61	82	6.31E-07
Merge step 11	5	22	6.06E-07
Merge step 12	133	298	6.03E-07
Merge step 13	5	214	5.29E-07
.....			

(e)

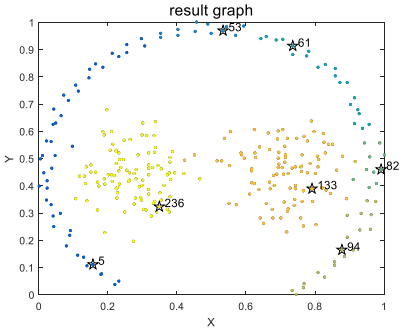
FIGURE 3. A detailed example of the LKSM_DPC algorithm on the Pathbased dataset. (a) The Euclidean distance matrix. (b) The matrix of local density and distance. (c) The decision graph. (d) Initial cluster diagram of 13 candidate cluster centers. (e) The table of subcluster similarity ranking.



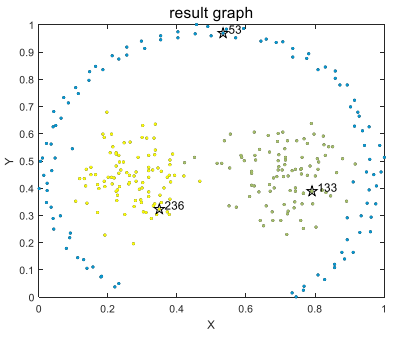
(f)



(g)



(h)



(i)

FIGURE 3. *(Continued.)* A detailed example of the LKSM_DPC algorithm on the Pathbased dataset. (f) Subcluster merging roadmap. (g) The subcluster merge results after performing merge step 2. (h) The subcluster merge results after performing merge step 7. (i) The subcluster merge results after performing merge step 11.

A. DENSITY PEAKS CLUSTERING ALGORITHM

Research on clustering using fast search and finding the density peaks (DPC) has been published, even though it is a new density-based clustering algorithm. DPC has two key variables: the local density ρ and the distance δ. It has been proposed based on the following assumption: the local density of a cluster center is higher than that of other points in the same group and has a relatively large distance from any points with a higher local density.

For each data point i, the local density ρ_i can be calculated using the following equation:

$$\begin{cases} \rho_i = \sum\limits_j \chi\left(d_{ij} - d_c\right) \\ \chi\left(x\right) = \begin{cases} 0, & x \geq 0 \\ 1, & x < 0 \end{cases} \end{cases} \tag{1}$$

where d_{ij} denotes the Euclidean distance between data point i and j; and d_c is the parameter of the cutoff distance and is the unique input parameter, which usually takes 1 to 2 percent of all the data points.

Rodriguez also provided another method for small datasets. The Gaussian kernel of the local density is given by the following equation:

$$\rho_i = \sum_j exp\left(-\left(\frac{d_{ij}}{d_c}\right)^2\right) \tag{2}$$

For each data point i, the distance δ_i can be defined using the following equation:

$$\delta_i = \min_{j:\rho_j > \rho_i}\left(d_{ij}\right) \tag{3}$$

If data point i has the largest local density, then $\delta_i = max\left(d_{ij}\right)$.

Rodriguez considered when ρ_i and δ_i are both large, the data point can be a candidate clustering center. In the decision graph, the potential cluster centers γ_i are defined as follows:

$$\gamma_i = \rho_i \times \delta_i \tag{4}$$

When the cluster center is selected, the remaining points are then allocated to the corresponding cluster. The allocation strategy of the DPC is to allocate the remaining points to the nearest cluster center. Therefore, the original DPC algorithm is as follows.

Through an extensive literature review, it was found that the improvement of DPC mainly focuses on calculating the local density ρ, the distance ρ, and the allocation strategy. That is step2, step3, and step6 in the original algorithm. Thus, step2 was improved by Du *et al.* [9]. Xie *et al.* [10] improved step2 and optimized step6, and Yu *et al.* [28] also redesigned step2 and optimized step6.

Of course, other steps of DPC have also been optimized and improved [29]–[31].

As can be seen from the above algorithm, DPC only needs one input parameter. However, the DPC does not consider the uneven local distribution of the density and does not notice when multiple density peaks occur in a cluster. Furthermore, it is difficult to determine a suitable cutoff distance d_c, and DPC needs to be tested to find a good parameter. Fortunately, some improved algorithms have been proposed to solve the problems, and we will describe these. Then, we will describe in detail the innovations of the following algorithms.

Algorithm 1 DPC

Input: A dataset $x \epsilon R_{N\times M}$ ($R_{N\times M}$ is the data matrix, where N denotes the total number of datasets, and M represents the dimensions of the datasets), the cutoff distance parameter d_C

Output: A label vector of the cluster index: $y \epsilon R_{N\times 1}$

1. Calculate the Euclidean distance matrix using $R_{N\times M}$;
2. Calculate ρ_i for each data point using equation (1) or (2);
3. Calculate δ_i for each data point using equation (3);
4. Construct the decision graph using equation (4);
5. Select the appropriate cluster centers from the decision graph;
6. Allocate the remaining data point (except for the cluster center) to the nearest point with the higher density;
7. Return y and end.

B. DENSITY PEAKS CLUSTERING BASED ON THE K-NEAREST NEIGHBORS

Because it is difficult to find a proper cutoff distance in DPC, Du *et al.* [9] proposed density peaks clustering based on the k-nearest neighbors (DPC-KNN). His main contribution was to improve the calculation of the local density. To better compute the local density, the idea of the k-nearest neighbors is adopted by his algorithm [32].

The idea of the k-nearest neighbors (KNN) is defined as follows:

$$KNN\left(x_i\right) = \left\{j \in X | d(x_i, x_j) \leq d(x_i, NN_k(x_i))\right\} \tag{5}$$

where X denotes the datasets, $d(x_i, x_j)$ represents the Euclidean distance between data points x_i and x_j.$NN_k(x_i)$ is the kth nearest data point to x_i according to the distance.

Then, the new local density is calculated as follows:

$$\rho_i = exp\left(-\left(\frac{1}{k}\sum_{x_j \in KNN(x_i)} d\left(x_i, x_j\right)^2\right)\right) \tag{6}$$

where k is $p \times N$, and p is a percentage of the total data points N.

From the algorithm description, the DPC-KNN only improved the local density calculation method of the original DPC, which is step 2. The other steps of the DPC-KNN are the same as DPC. The experimental results demonstrate that the DPC-KNN has improved the clustering performance on some datasets compared to DPC, Spectral Clustering (SC), and K-means. However, the DPC-KNN only used the k-nearest neighbors' method and does not take into

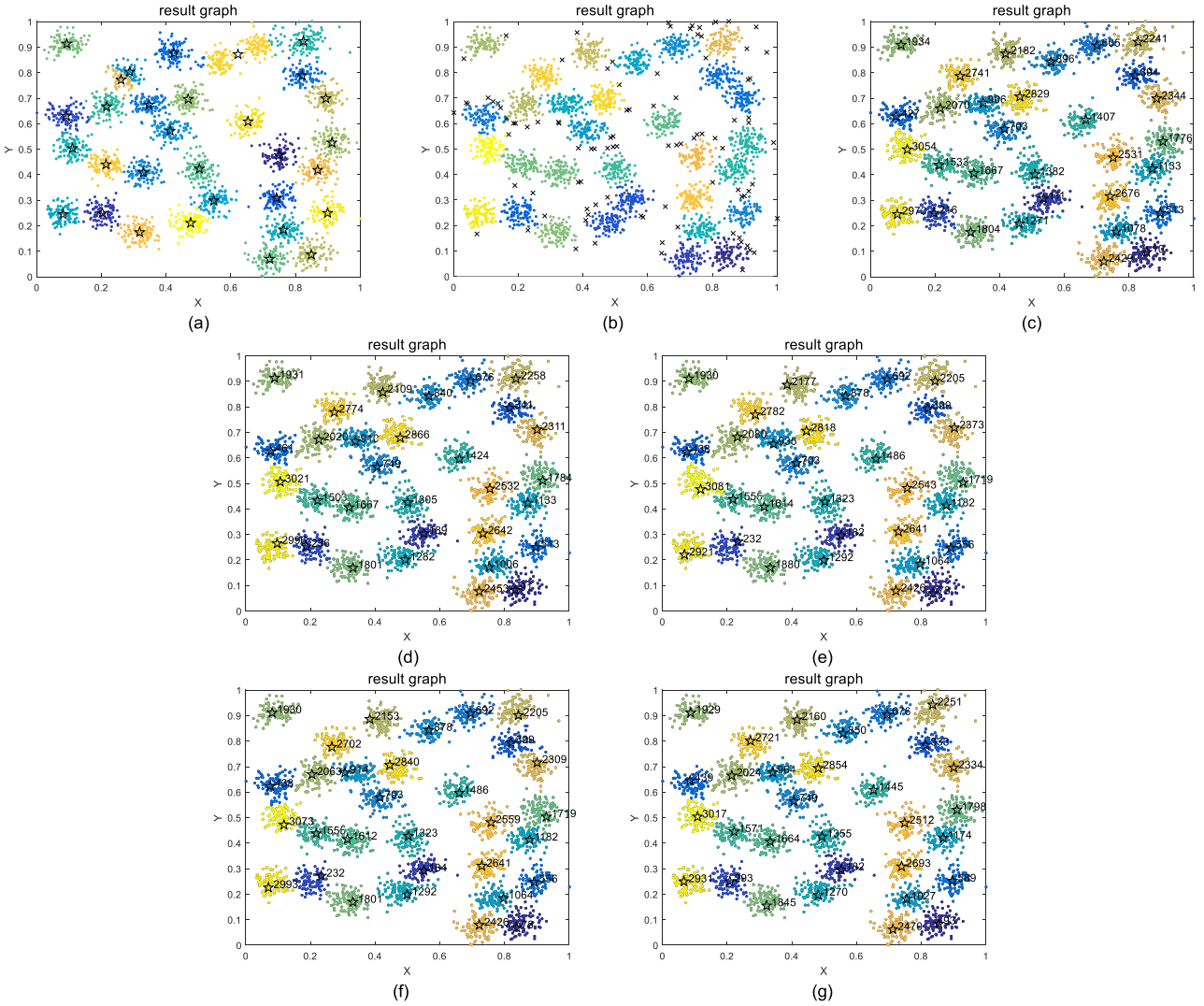


FIGURE 4. Experimental results for the D31 dataset. (a) K-means. (b) DBSCAN. (c) DPC. (d) DPC-KNN. (e) FKNN-DPC. (f) DPCSA. (g) LKSM_DPC.

consideration the internal structure and distribution of the data. Therefore, the DPC-KNN performs poorly on manifold datasets.

C. FUZZY WEIGHTED K-NEAREST NEIGHBORS DENSITY PEAKS CLUSTERING

The local density for DPC is affected by the cutoff distance and the ''Domino Effect'' that is easily caused by the allocation strategy. To address this effect, Xie *et al.* [10] proposed a fuzzy weighted k-nearest neighbors density peak clustering (FKNN-DPC). This algorithm also used k-nearest neighbors to calculate the local density and redesigned the allocation strategy of the remaining points.

The new local density calculated by the following:

$$\rho_i = \sum_{j \in KNN_i} \exp\left(-d_{ij}\right) \tag{7}$$

where d_{ij} represents the Euclidean distance from data point i to j.

To calculate the probability p_i^c, the similarity w_{ij} is defined first.

$$w_{ij} = \frac{1}{1 + d_{ij}} \tag{8}$$

The value of p_i^c is determined by the following equation:

$$p_i^c = \sum_{j \in KNN_i, y_j = c} \gamma_{ij} * w_{ij} \tag{9}$$

where $\gamma_{ij} = {w_{ij}}\Big/{\sum_{l \in KNN_j} w_{lj}}$ and $y_j = c$ represents the cluster label c of data point j.

As seen from the above description, the FKNN-DPC not only improved the calculation method of the local density but also implemented a new redesigned remaining point allocation strategy. From the experimental results, the FKNN-DPC can not only find the cluster centers and identify the clusters in a large number of datasets, but it is also significantly better than the original DPC. However, the allocation strategy of this

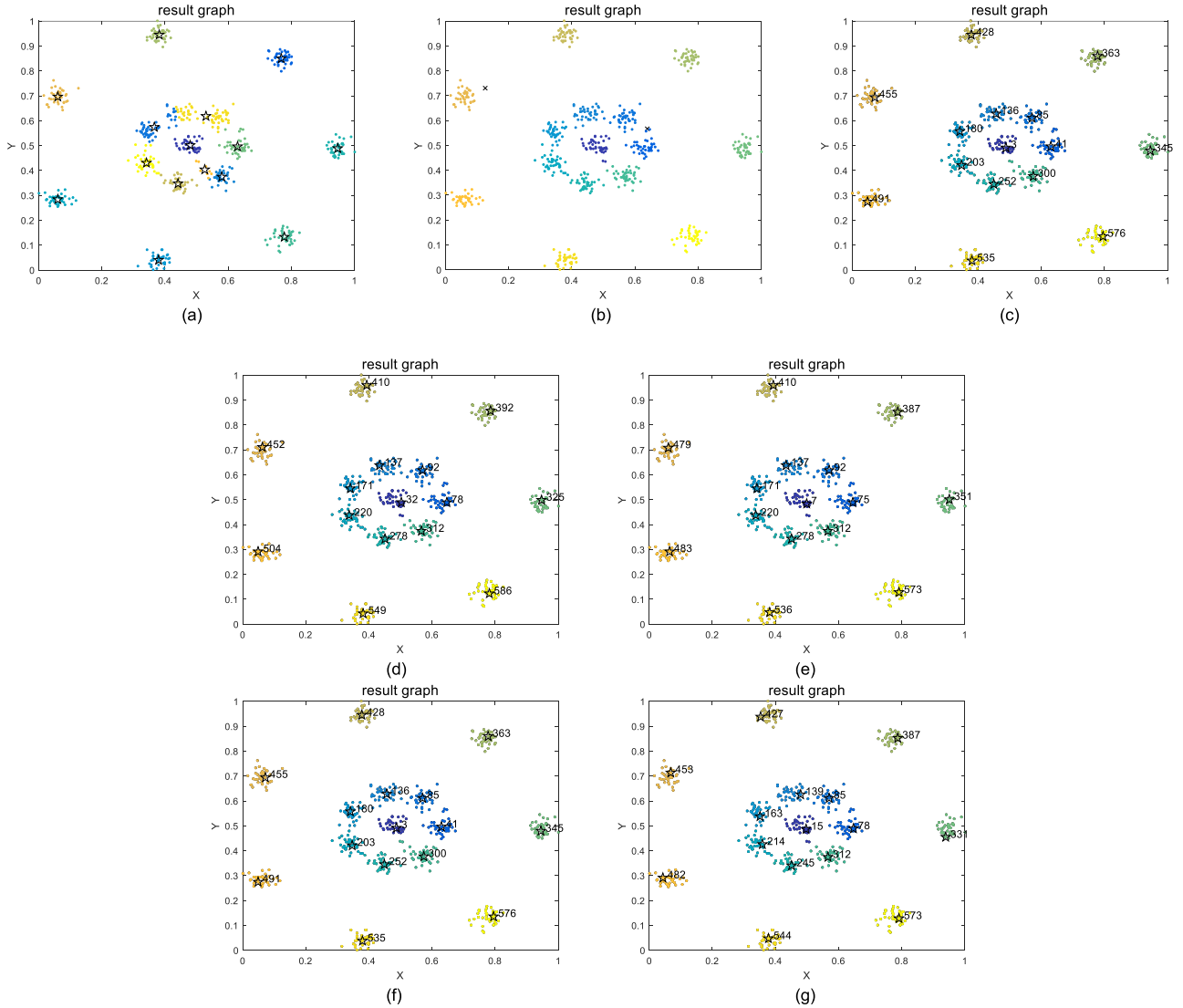


FIGURE 5. Experimental results for the R15 dataset. (a) K-means. (b) DBSCAN. (c) DPC. (d) DPC-KNN. (e) FKNN-DPC. (f) DPCSA. (g) LKSM_DPC.

algorithm is very time-consuming and the input parameter K needs to be set manually.

D. DENSITY PEAKS CLUSTERING ALGORITHM

To solve the problem of finding appropriate input parameters, Yu *et al.* [28] proposed an improved DPC using the weighted local density sequence and the remaining point assignment strategies (DPCSA). The major improvements of the method are that it does not need to input parameters, and it uses the fixed k-nearest neighbors ($K = 5$) to calculate the local density. Moreover, the DPCSA implemented a new redesigned nearest neighbor assignment strategy.

The input parameter (d_c or k) requires manual input and prior knowledge. To solve this problem, and an improved local density clustering method using the weighted sequence and fixed KNN was proposed.

$$\rho_i = \sum_{j=1}^{K} exp\left(-d'_{ij}\right) + \sum_{j=K+1}^{n-1} \frac{exp\left(-d'_{ij}\right)}{(j-K)^2} \tag{10}$$

where $K = 5$ and $d'_{ij}(j = 1, \ldots, n-1)$ is the increasing order of Euclidean distance d_{ij}.

Besides, the DPCSA proposed two-stage assignment strategies, which defined a boundary condition. The data points are used in the first assignment strategy when meeting the conditions are $\delta_i = {}^1\!/_n \sum_{l=1}^{n} \delta_l$, and the first assignment method is the same as the DPC. The remaining points are subjected to the second strategy, and the strategy constructed the nearest neighbor dynamic table to allocate the remaining points. Due to limited space, we note that the second assignment strategy is detailed in the literature [28].

Although the DPCSA does not require input parameters and improved the assignment strategies, the experimental results demonstrate that the performance of the DPCSA is only slightly better than those of other algorithms for partial datasets. Moreover, the DPCSA takes more time than other algorithms to calculate the local density and assign the remaining points.

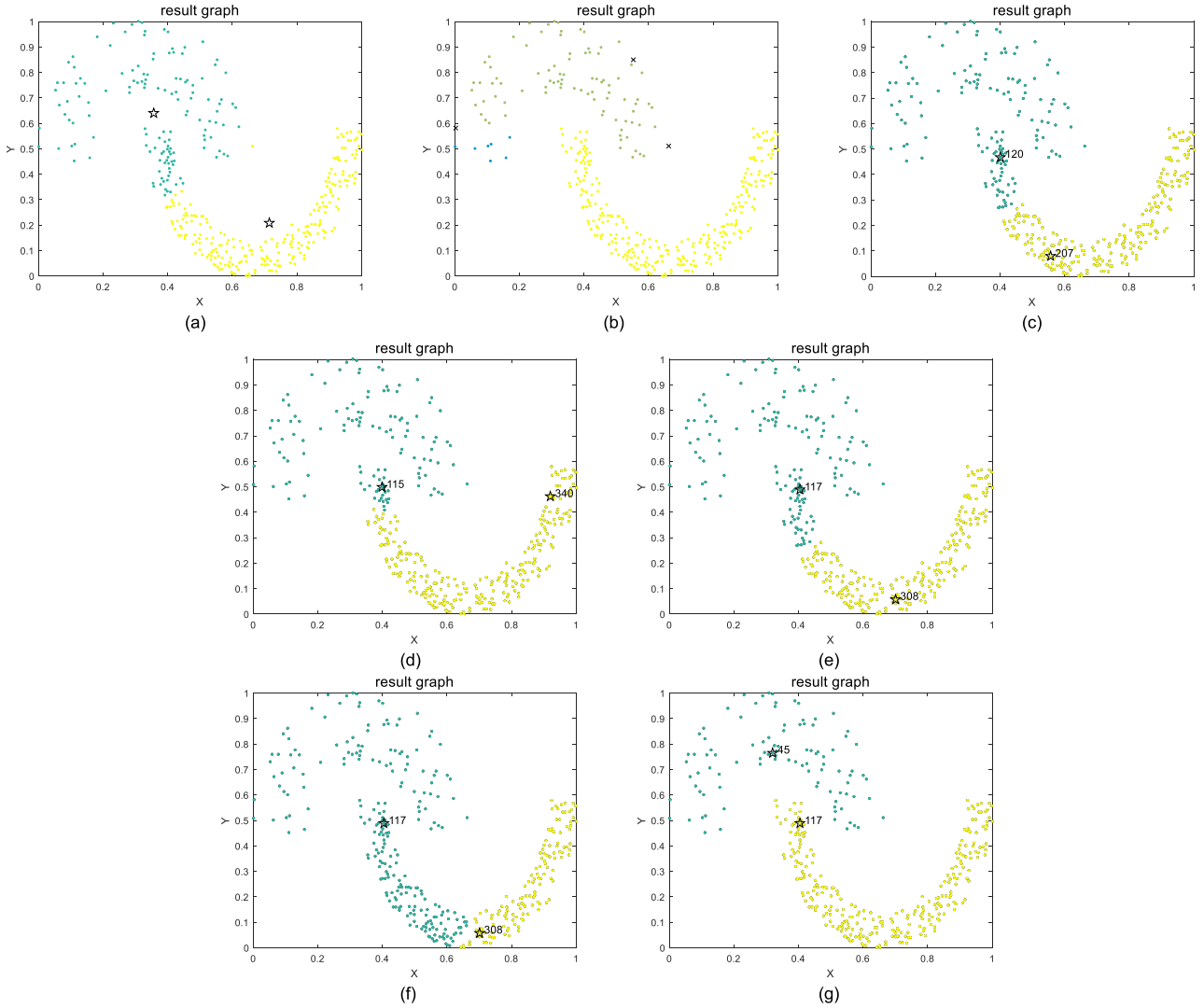


FIGURE 6. Experimental results for the Jain dataset. (a) K-means. (b) DBSCAN. (c) DPC. (d) DPC-KNN. (e) FKNN-DPC. (f) DPCSA. (g) LKSM_DPC.

III. OUR ALGORITHM: LKSM_DPC

In this section, we will discuss the proposed method, namely, a density peaks clustering algorithm based on the layered k-nearest neighbors and subcluster merging (LKSM_DPC). Two important improvements will be presented: (1) the local density of the layered k-nearest neighbors, and (2) the multiple-peak values subcluster merging strategy.

A. LOCAL DENSITY OF THE LAYERED K-NEAREST NEIGHBORS

For the DPC algorithm, the estimation of the density of each point not only prevents the selection of the cluster center but also directly influences the quality of the cluster. We know that by definition of the distance δ, the value of δ is also closely related to the density ρ. Therefore, ρ is highly important for the datasets with uneven density distribution. DPC determines the local density using the cutoff distance d_c, but the method has difficulties selecting an appropriate d_c, which will affect the local density and initial clustering center. Furthermore, it is usually easier to determine the value of K than the cutoff distance d_c, and the FKNN-DPC uses the K value to determine local density.

This paper seeks to determine the local density using the k-nearest neighbors. The k-nearest neighbors are divided into multiple layers according to the density of the dataset itself. That is the degree of unevenness in the density distribution. After numerous experiments, the following layering strategy was adopted. The k-nearest neighbors of a high-density dataset are evenly divided into four layers. The k-nearest neighbors in the first layer are the closest to the cluster center, the k-nearest neighbors in the second layer are closer to the cluster center, the k-nearest neighbors in the third layer are far from the cluster center, and the k-nearest neighbors in the fourth layer are the farthest from the cluster center. The k-nearest neighbors of each layer have different

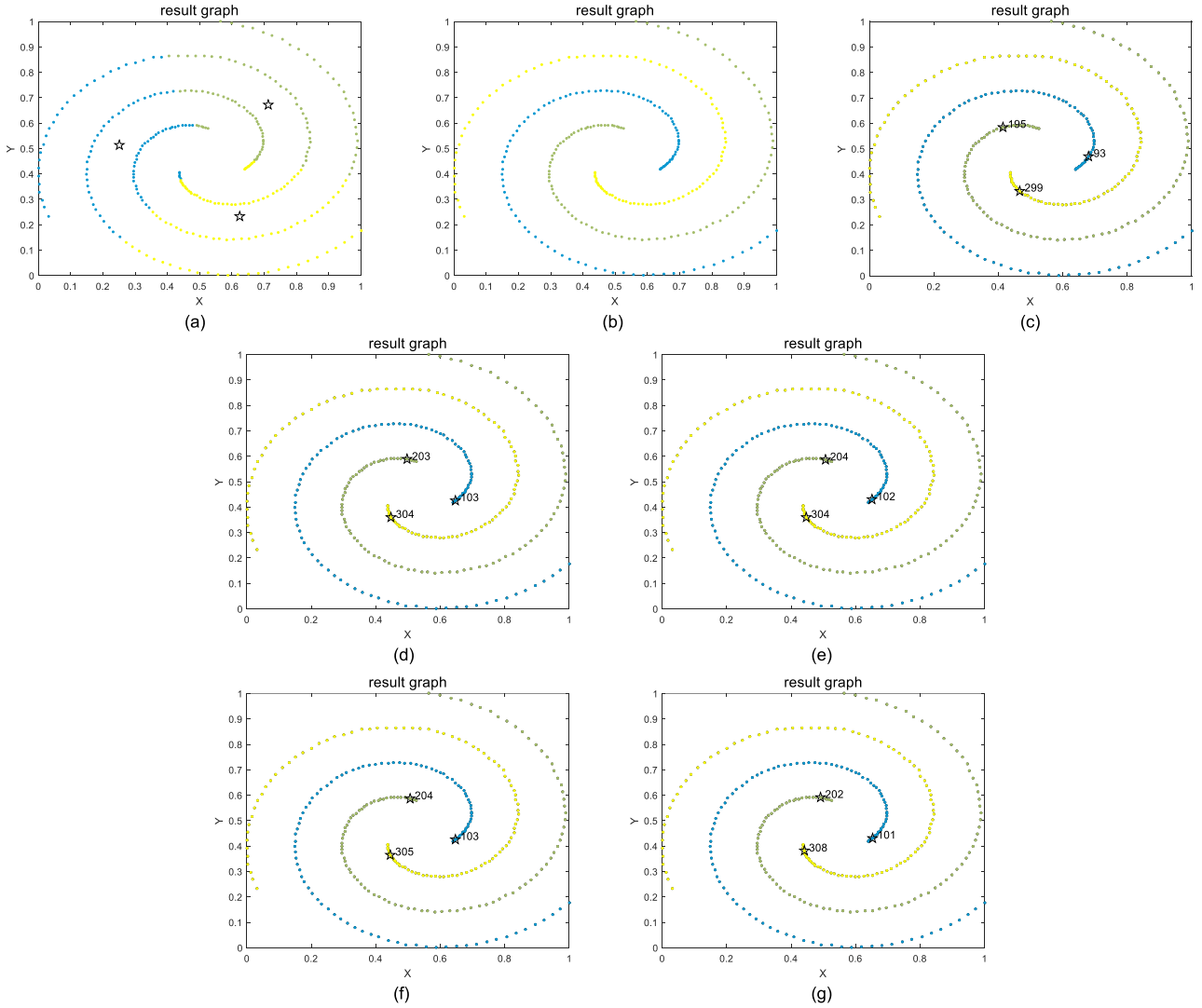


FIGURE 7. Experimental results for the Spiral dataset. (a) K-means. (b) DBSCAN. (c) DPC. (d) DPC-KNN. (e) FKNN-DPC. (f) DPCSA. (g) LKSM_DPC.

local density contributions, and so each layer should be set with different weight. Similarly, the k-nearest neighbors of a medium-density dataset are divided into three layers, those of a general density dataset are divided into two layers, and those of a low-density dataset are divided into one layer. Schematic diagrams of the k-nearest neighbors layering for different local density datasets are shown in Figure 1.

The new layered k-nearest neighbor local density calculation formula is as follows:

$$\begin{cases} \rho_i = \sum_{l=1}^{L} \sum_{j \in lknn} \frac{1}{c+d_{ij}} \cdot w, & 1 \le L \le 4 \\ w = \alpha\,(1-l)+1, & 1 \le l \le 4 \end{cases} \tag{11}$$

where L is the number of layers; l represents the layer of the current k-nearest neighbors; $j \in lknn$ indicates that point j in the k-nearest neighbors belong to layer l; c is the sum of the distance all data points; d_{ij} represents the distance; w represents the local density weight contribution of the k-nearest neighbors of each layer; and α is a parameter, which is usually 0.2.

B. SIMILARITY AND SUBCLUSTER MERGING

In this section, to address the problem of multiple peaks in a cluster in complex manifold data, a new subcluster similarity calculation method and a merging strategy are proposed. First, we calculate the similarity of a subcluster using the idea of shared neighbors and Newton's gravitation. Second, subclusters are merged according to their orders of similarity.

To describe a new method for calculating the similarity of subclusters, we first briefly describe the idea of shared neighbors and Newton's gravitation. The idea of shared neighbors is as follows [11]: for arbitrary data points i and j, we can have the k-nearest neighbors set $D(i)$ of i and the k-nearest neighbors set $D(j)$ of j, and the shared neighbors of point i and j are their common neighbor set $SNN(i,j)$.

$$SNN(i,j) = D(i) \cap D(j) \tag{12}$$

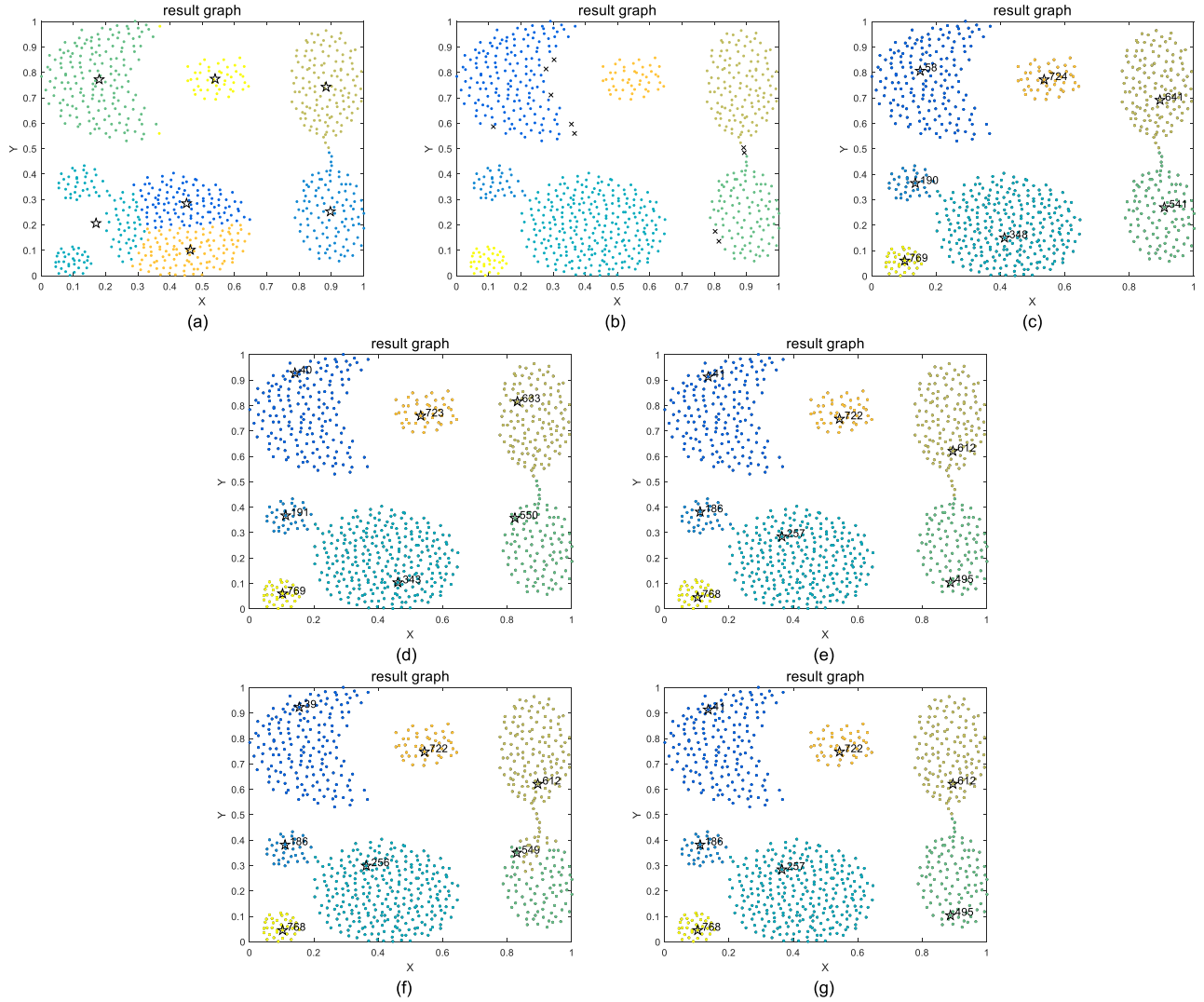


FIGURE 8. Experimental results for the Aggregation dataset. (a) K-means. (b) DBSCAN. (c) DPC. (d) DPC-KNN. (e) FKNN-DPC. (f) DPCSA. (g) LKSM_DPC.

The gravity calculation formula provided by Newton is as follows [31]:

$$F = G\frac{M_1 M_2}{R^2} \tag{13}$$

where F is universal gravitation, and M_1 and M_2 respectively represent the masses of two objects, and R is the distance between two objects.

Next, we consider the problem of the similarity of subclusters. First, if two subclusters are similar, they should have shared neighbors. That is the number of shared neighbors is not zero. Second, if two subclusters are similar, there is a potential attraction between them, which is similar to the gravitation. Therefore, the closer the two subcluster are, the more neighbors they share, the greater the attraction, and the higher the similarity.

In summary, a new similarity calculation method is proposed:

$$SIM\ (C1, C2) = SNN\ (c1, c2) \cdot \frac{\rho_{c1}\rho_{c2}}{d^2_{c1c2}} \tag{14}$$

where $C1$ and $C2$ are two subclusters; $c1$ and $c2$ represent the clustering centers of two subcluster; $SNN\ (c1, c2)$ denotes the shared neighbors of the two subclusters; ρ_{c1} is the local density of $c1$; and d_{c1c2} is the Euclidean distance of the two clustering centers.

For multiple density peaks in a cluster, we can perform subcluster merging by similarity. For multiple candidate cluster centers, we establish a subcluster similarity matrix and sort the similarity matrix in descending order. Subcluster merging is performed in turn until the number of subclusters is equal to the true number of clusters.

C. ALGORITHM FLOW

We address the shortcomings of the DPC and DPC variants in processing manifold datasets that are due to the following: (1) datasets with uneven density distribution, and (2) multiple cluster peaks in a cluster. In this paper, we proposed a DPC algorithm based on the layered k-nearest neighbors and subcluster merging (LKSM_DPC). The main contribution

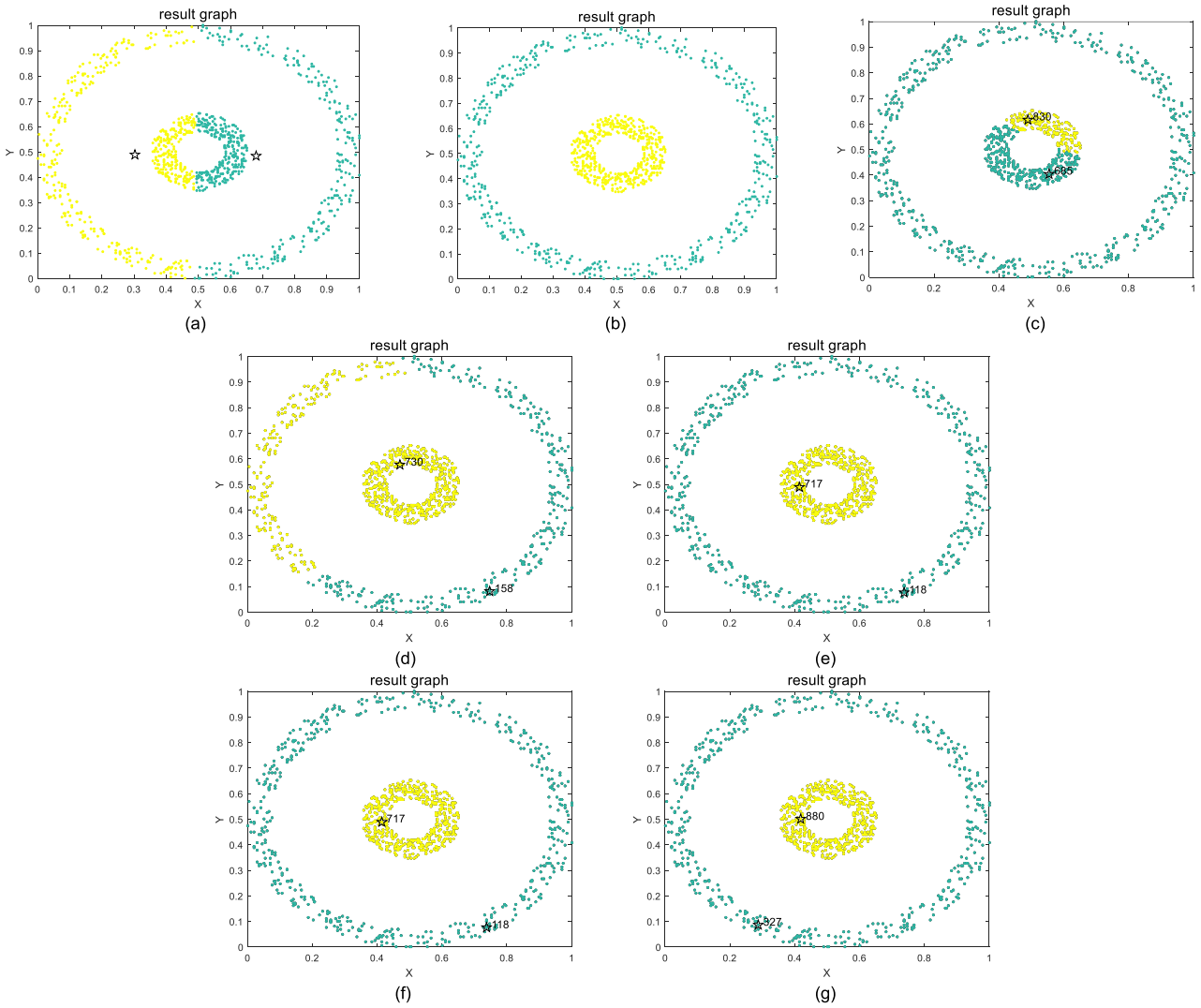


FIGURE 9. Experimental results for the Ring dataset. (a) K-means. (b) DBSCAN. (c) DPC. (d) DPC-KNN. (e) FKNN-DPC. (f) DPCSA. (g) LKSM_DPC.

of our algorithm is to divide the k-nearest neighbors into multiple layers to adapt to datasets with different densities and solve the problem of multiple peaks in a cluster through the subcluster merging strategy.

To demonstrate each step of our algorithm, we had a detailed flowchart in Figure 2.

The specific steps of the LKSM_DPC are described by algorithm 2 and algorithm 3.

D. EXAMPLE OF THE LKSM_DPC ALGORITHM

To show the LKSM_DPC proposed in this paper in detail, we will take the complex manifold dataset Pathbased as an example. The actual number of cluster centers in this dataset is 3. First, the Euclidean distance matrix of the dataset is shown in (a). Second, we calculate the local density ρ and the distance δ respectively, the calculation result is shown in (b). Third, we select 13 candidate cluster centers from the decision graph (c) for subcluster merging. The initial clustering result is shown in (d), and the candidate cluster centers are represented by pentagrams. (e) is the subcluster similarity table, and the steps are arranged in descending order of their similarity. (f) is the subcluster merging roadmap, detailing the merge steps. (g) is the clustering graph of the subcluster merge results after performing merge step 2. (h) is the clustering graph of the subcluster merge results after performing merge step 7. (i) is the clustering graph of the subcluster merge result after executing merge step 11. At this time, the number of subclusters is equal to the number of true clusters, indicating the end of the entire merge process.

E. TIME COMPLEXITY ANALYSIS

The time complexity of the LKSM_DPC is mainly composed of the following steps: (1) calculate the Euclidean distance between two points ($O(n^2)$); (2) calculate the local density ρ of the layered KNN ($O(n^2)$); (3) evaluate the distance δ of each point ($O(n^2)$); (4) allocate the remaining points to

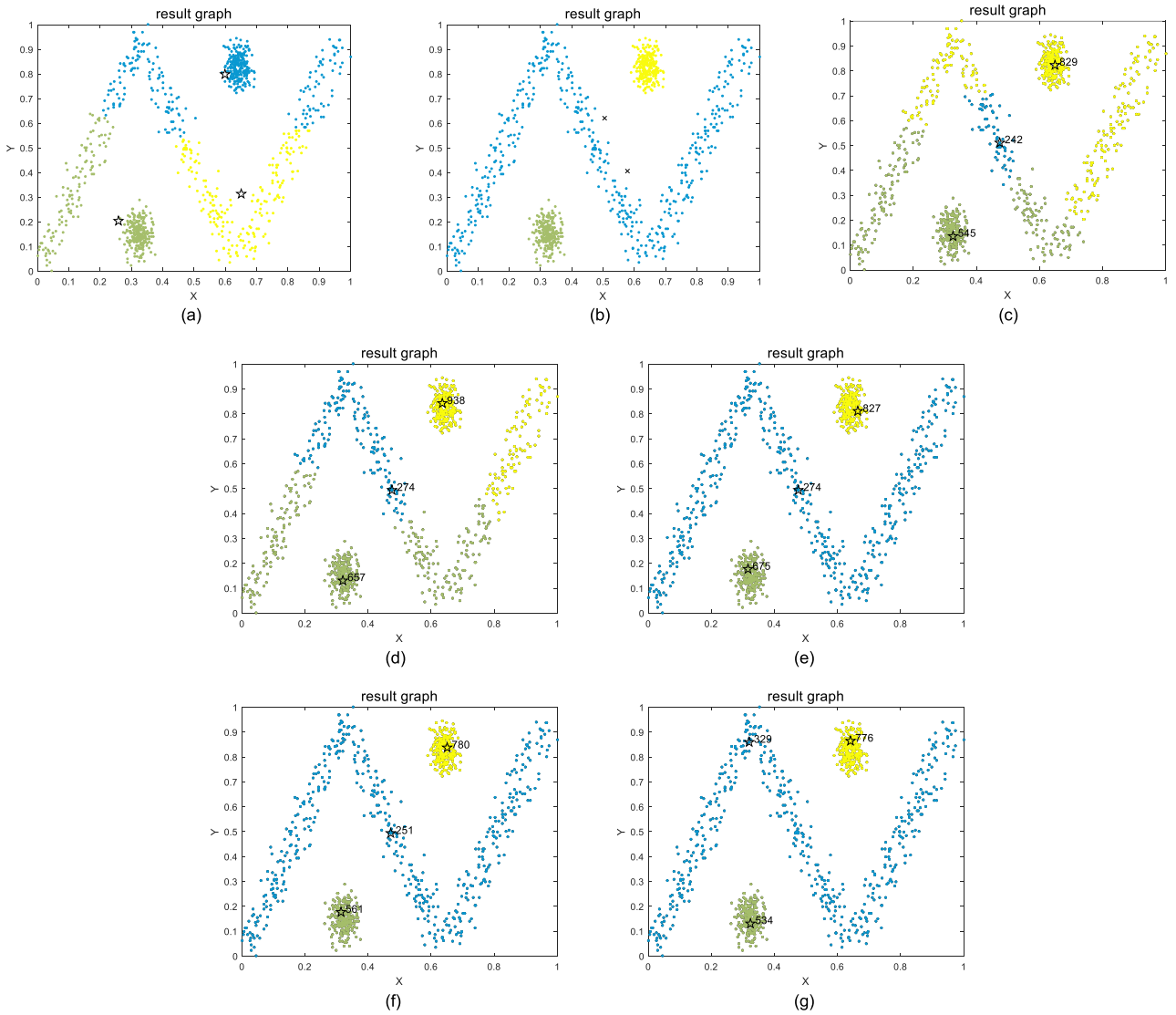


FIGURE 10. Experimental results for the Zigzag dataset. (a) K-means. (b) DBSCAN. (c) DPC. (d) DPC-KNN. (e) FKNN-DPC. (f) DPCSA. (g) LKSM_DPC.

cluster ($O(n^2)$); and (5) perform subcluster merging, including (a) search the shared neighbors ($O(n^2)$), (b) calculate the similarity of subcluster ($O(n^2)$), and (c) process the subcluster merging ($O(n)$). To sum up, the time complexity of our algorithm is $O(n^2)$.

IV. EXPERIMENTS

In this section, we compare the performance of the LKSM_DPC with the K-means, DBSCAN, DPC, DPC-KNN, FKNN-DPC, and DPCSA algorithms. The codes of the DPC, FKNN-DPC, and DPCSA were provided by the authors; we program the code of the DBSCAN and KNN-DPC according to the authors' articles; the code of the K-means was provided by the Matlab built-in functions. Our experimental environment is a PC with an Intel (R) Core (TM) i7-9700 CPU @ 3.00 GHz and 16G RAM. All experiments are implemented with MATLAB R2015b.

Our experimental datasets include 10 synthetic datasets and 14 UCI datasets, which are shown in Tables 2 and 3. The parameter settings of the contrast algorithms are taken from the original articles. Cluster evaluation indexes include the ACC [33], ARI [28] and AMI [34], and the respective

TABLE 2. Synthetic datasets.

Dataset	Size	Dimension	Class
D31	3100	2	31
R15	600	2	15
Jain	373	2	2
Spiral	312	2	3
Aggregation	788	2	7
Ring	1000	2	2
Zigzag	1002	2	3
Flame	240	2	2
Pathbased	300	2	3
S2	5000	2	15

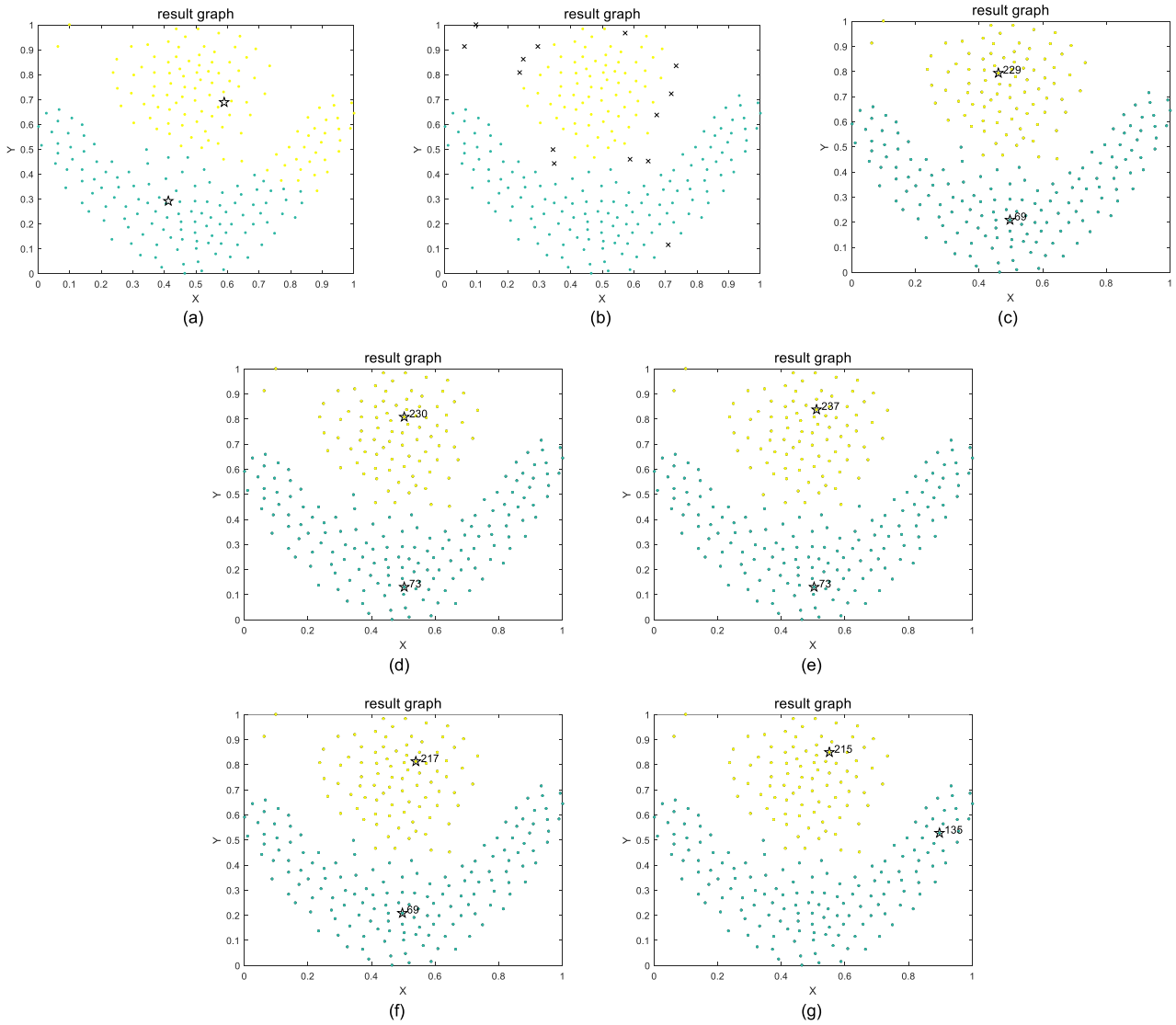


FIGURE 11. Experimental results for the Flame dataset. (a) K-means. (b) DBSCAN. (c) DPC. (d) DPC-KNN. (e) FKNN-DPC. (f) DPCSA. (g) LKSM_DPC.

TABLE 3. UCI datasets.

Dataset	Size	Dimension	Class
Wine	178	13	3
Iris	150	4	3
Seeds	210	7	3
Ionosphere	351	34	2
Wdbc	569	30	2
Segmentation	2310	19	7
Glass	214	9	6
Libras Movement	360	91	15
Dermatology	366	34	6
Waveform	5000	21	3
Parkinsons	195	23	2
Pima	768	8	2
SCADI	70	206	7
Letter	20000	16	26

calculation formulas are shown below:

$$ACC = \frac{1}{n}\sum_{i=1}^{n}\delta\left(u_i, v_i\right) \tag{15}$$

$$ARI = \frac{RI - E\left[RI\right]}{max\left(RI\right) - E\left[RI\right]} \tag{16}$$

$$AMI = \frac{I\left(U, V\right) - E\left\{I\left(U, V\right)\right\}}{\sqrt{H\left(U\right)H\left(V\right)} - E\left\{I\left(U, V\right)\right\}} \tag{17}$$

Before the experiment, to avoid fluctuations caused by inconsistent attribute values, all datasets were standardized using formula (18).

$$x'_{ij} = \frac{x_{ij} - min\left(x_j\right)}{max\left(x_j\right) - min\left(x_j\right)} \tag{18}$$

where x'_{ij} is the result of normalization, x_{ij} is the data point of row i and column j in a dataset, $\max\left(x_j\right)$ is the maximum value of column j in the dataset, and $\min\left(x_j\right)$ is the minimum value of column j in the dataset.

A. EXPERIMENTS USING SYNTHETIC DATASETS

First, our proposed LKSM_DPC algorithm is compared with the K-means, DBSCAN, DPC, KNN-DPC, FKNN-DPC, and

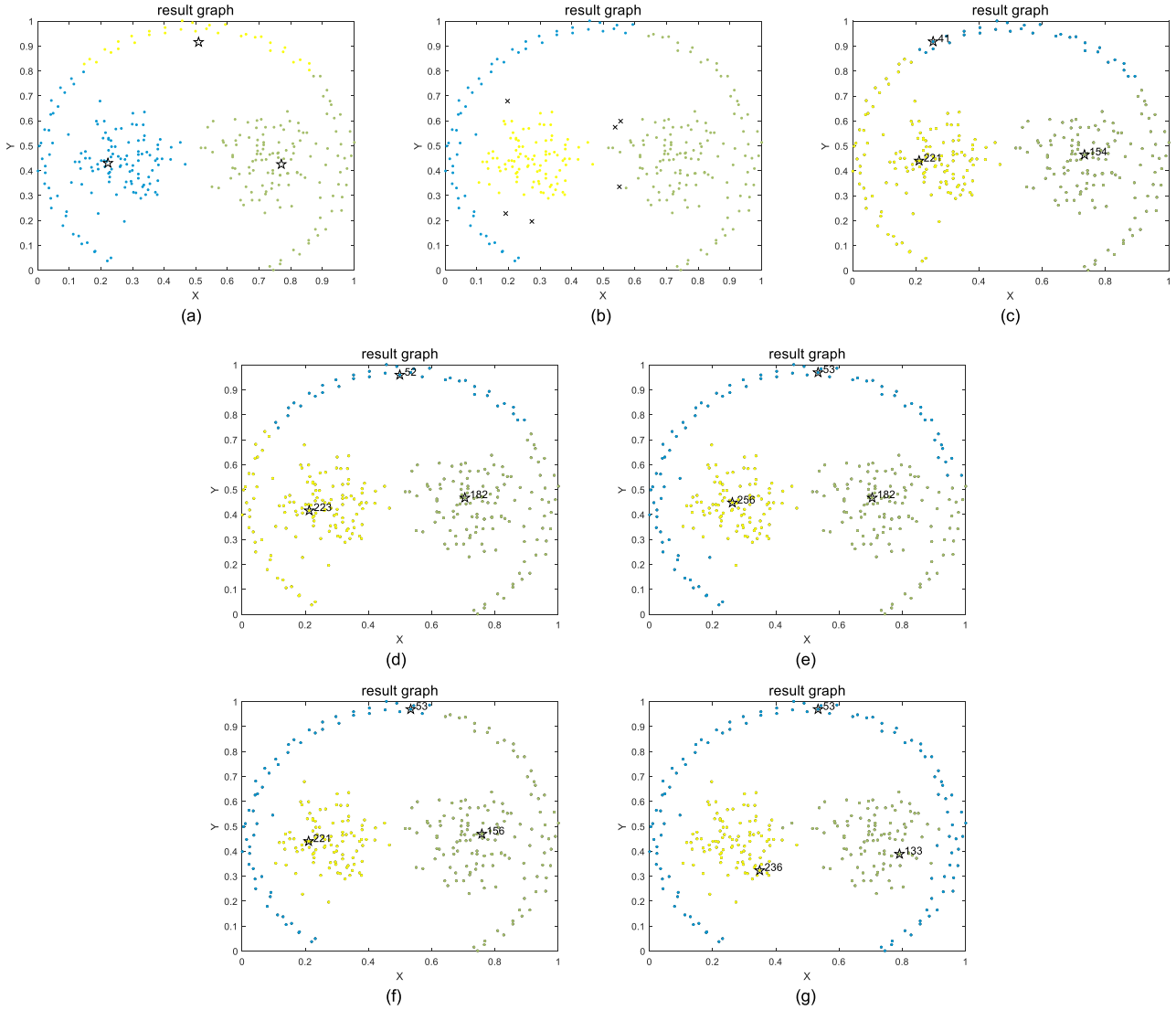


FIGURE 12. Experimental results for the Pathbased dataset. (a) K-means. (b) DBSCAN. (c) DPC. (d) DPC-KNN. (e) FKNN-DPC. (f) DPCSA. (g) LKSM_DPC.

DPCSA on 10 synthetic datasets. The detailed descriptions of the datasets are given in Table 2. The experimental results are shown in Figures 4-13.

Figure 4 shows the clustering results obtained by 7 algorithms on dataset D31. All algorithms can correctly find clusters and reasonably allocate the remaining points. The clustering effect of K-means is similar to that of DBSCAN, but DBSCAN marks many noise points. For the other 5 algorithms, only approximately 3% of the points were misallocated. Among them, the DPC-KNN achieved better clustering performance than the other algorithms, and the clustering effect of our LKSM_DPC algorithm is preferable to that of the DPCSA, K-means, and DBSCAN.

The clustering results for the R15 dataset obtained by 7 algorithms are given in Figure 5. The R15 dataset contains 15 clusters. Outermost 7 clusters are far apart, and it was found that all algorithms can correctly cluster these 7 clusters. Innermost clusters in the dataset can easily cause allocation errors because clusters are adjacent. Nevertheless, LKSM_DPC gets the best clustering performance. The ACC of our algorithm is 0.9967, which is the same as those of the DPC-KNN and FKNN-DPC.

The clustering results for the Jain dataset are shown in Figure 6. Jain is a case in the dataset with uneven density distribution, including two clusters. Especially, the cluster with a compact density distribution is prone to multiple peaks in one cluster. As we can see from Figure 6, the DPC, DPC-KNN, FKNN-DPC, and DPCSA cannot find the correct cluster centers, and the cluster centers all appear in the same cluster. The DBSCAN cannot correctly distinguish the two clusters. However, the LKSM_DPC can correctly identify the two clusters through subcluster merging.

Figure 7 shows the clustering results obtained by 7 algorithms for the Spiral dataset. The spiral is a typical

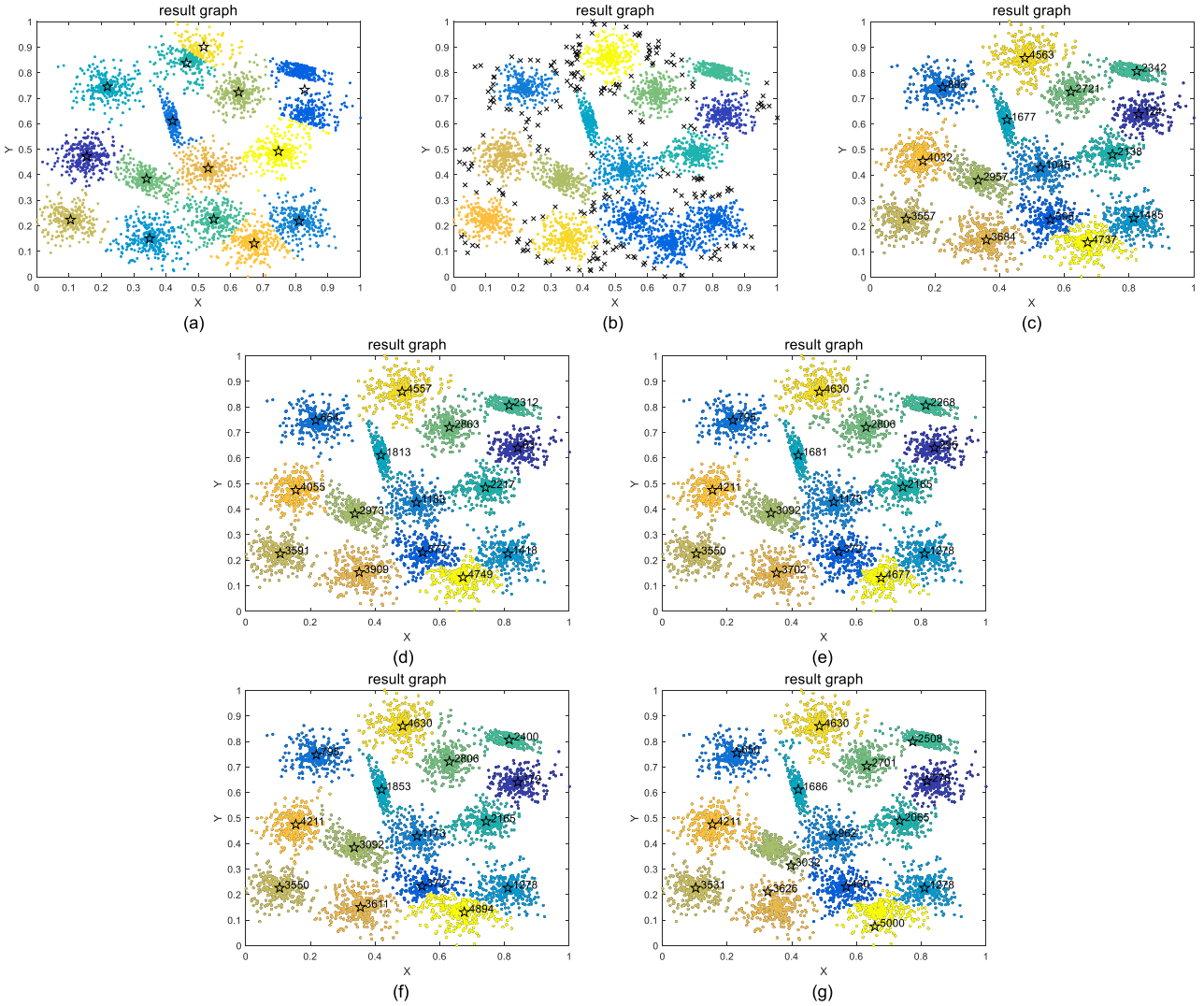


FIGURE 13. Experimental results for the S2 dataset. (a) K-means. (b) DBSCAN. (c) DPC. (d) DPC-KNN. (e) FKNN-DPC. (f) DPCSA. (g) LKSM_DPC.

manifold dataset consisting of 3 spiral clusters. K-means cannot efficiently handle the three clusters. Except for the K-means, all algorithms can identify clustering correctly, and they can allocate the remaining points completely and correctly.

We can see the clustering results for the Aggregation dataset in Figure 8. Aggregation is a complex manifold dataset with 7 irregularly shaped clusters. In terms of the K-means algorithm, it cannot find the clustering center correctly. For the DBSCAN, although it can recognize clustering, there are always a few noise points. For the other five algorithms can find the cluster center of each cluster, but the DPCSA has an obvious allocation error in the two clusters on the far right, in which clustering performance is weak. The LKSM_DPC and DPC have the best clustering performances, and their ARI can reach 0.9956.

Figure 9 shows the clustering results obtained by 7 algorithms for the Ring. The Ring is also a dataset with an uneven density distribution, mainly composed of 2 circular clusters. In Figure 9, the K-means, DPC-KNN, and DPC cannot find the cluster centers correctly. Although the DPC-KNN can find the cluster centers, there is an allocation error. The LKSM_DPC has the same clustering performance as the DBSCAN, FKNN-DPC, and DPCSA through the subcluster merging strategy.

The clustering results for the Zigzag dataset obtained by 7 algorithms are shown in Figure 10. Zigzag is also a manifold dataset consisting of 3 clusters. For the K-means, it cannot address three clusters efficiently. In terms of the DBSCAN, it was able to identify three clusters, but there were two noise points. Although the DPC and the DPC-KNN can find the cluster centers correctly, there are obvious remaining point allocation errors. The FKNN-DPC and DPCSA can completely identify these 3 clusters by improving the allocation of the remaining points. Although the DPC allocation strategy is adopted by the LKSM_DPC, the Zigzag dataset

TABLE 4. ACC, ARI, and AMI of 7 algorithms on the synthetic datasets.

Algorithm	ACC	ARI	AMI	Par	ACC	ARI	AMI	Par
		D31					***R15***	
K-means	0.8390	0.8223	0.9143	31	0.9100	0.8829	0.9320	15
DBSCAN	0.8281	0.8078	0.8895	0.04/38	0.9900	0.9819	0.9825	0.04/12
DPC	0.9687	0.9372	0.9564	2%	0.9917	0.9821	0.9854	2%
DPC-KNN	**0.9710**	**0.9415**	**0.9585**	2%	**0.9967**	**0.9928**	**0.9938**	2%
FKNN-DPC	0.9690	0.9375	0.9566	9	**0.9967**	**0.9928**	**0.9938**	9
DPCSA	0.9677	0.9353	0.9552	-	0.9933	0.9857	0.9885	-
LKSM_DPC	0.9684	0.9364	0.9552	124	**0.9967**	**0.9928**	**0.9938**	24
		Jain					***Spiral***	
K-means	0.8820	0.5767	0.4916	2	0.3462	-0.0057	-0.0052	3
DBSCAN	0.9732	0.9731	0.8470	0.05/8	**1**	**1**	**1**	0.04/2
DPC	0.8606	0.5146	0.4667	2%	**1**	**1**	**1**	4%
DPC-KNN	0.9249	0.7146	0.6183	2%	**1**	**1**	**1**	4%
FKNN-DPC	0.8606	0.5146	0.4667	7	**1**	**1**	**1**	7
DPCSA	0.6247	0.0442	0.2167	-	**1**	**1**	**1**	-
LKSM_DPC	**1**	**1**	**1**	30	**1**	**1**	**1**	13
		Aggregation					***Ring***	
K-means	0.7525	0.6963	0.7947	7	0.5350	0.0039	0.0028	2
DBSCAN	0.9835	0.9779	0.9529	0.04/6	**1**	**1**	**1**	0.05/4
DPC	**0.9975**	**0.9956**	**0.9922**	2%	0.6770	0.1248	0.2041	2%
DPC-KNN	0.9962	0.9935	0.9892	1%	0.8000	0.3595	0.3954	2%
FKNN-DPC	**0.9975**	0.9949	0.9907	8	**1**	**1**	**1**	8
DPCSA	0.9734	0.9581	0.9537	-	**1**	**1**	**1**	-
LKSM_DPC	**0.9975**	**0.9956**	**0.9922**	8	**1**	**1**	**1**	40
		Zigzag					***Flame***	
K-means	0.6996	0.3278	0.4896	3	0.8583	0.5117	0.4693	2
DBSCAN	0.9980	0.9957	0.9876	0.04/3	0.9417	0.9081	0.7570	0.065/4
DPC	0.5649	0.2413	0.3785	2%	**1**	**1**	**1**	5%
DPC-KNN	0.6607	0.3075	0.4719	2%	**1**	**1**	**1**	2%
FKNN-DPC	**1**	**1**	**1**	10	0.9917	0.9666	0.9267	5
DPCSA	**1**	**1**	**1**	-	**1**	**1**	**1**	-
LKSM_DPC	**1**	**1**	**1**	101	**1**	**1**	**1**	60
		Pathbased					***S2***	
K-means	0.7433	0.4613	0.5098	3	0.8866	0.8595	0.9100	15
DBSCAN	0.8033	0.5890	0.6884	0.065/4	0.8210	0.7485	0.8511	0.04/30
DPC	0.7400	0.4572	0.5054	2%	**0.9696**	**0.9370**	**0.9446**	2%
DPC-KNN	0.7600	0.4797	0.5294	5%	0.9678	0.9335	0.9429	2%
FKNN-DPC	0.8967	0.7323	0.7744	8	0.9588	0.9157	0.9341	8
DPCSA	0.8233	0.6133	0.7073	-	0.9580	0.9152	0.9333	-
LKSM_DPC	**0.9833**	**0.9507**	**0.9282**	20	0.9658	0.9292	0.9411	12

can also be completely correctly identified through subcluster merging.

Figure 11 shows the clustering results obtained by 7 algorithms for the Flame dataset. The Flame is a case in the manifold dataset. The K-means making an obvious allocation error in clustering. For the DBSCAN, many boundary points are identified as noise points. The DPC, DPC-KNN, and DPCSA can completely find the cluster centers and correctly assign the remaining points. Although the FKNN-DPC can correctly find the cluster centers, there are 2 points where there is an allocation error. Our algorithm correctly identifies 2 clusters through a subcluster merging strategy, and its clustering performance is the same as those of the DPC, DPC-KNN, and DPCSA.

Figure 12 shows the clustering results obtained by 7 algorithms for the Pathbased dataset. Pathbased is a complicated manifold dataset consisting of 3 clusters. Due to the contact between clusters, remaining point allocation errors can easily occur. For the K-means and DBSCAN, they can not cluster the three clusters correctly and there was an obvious misallocation. Although the DPC and DPC-KNN can find the cluster centers, a large number of data points have been misallocated. The FKNN-DPC and DPCSA are preferable to DPC and the DPC-KNN, but there are still a few points with allocation errors. The subcluster merging strategy is adopted by the LKSM_DPC. The result of LKSM_DPC is perfect, for the specific merging process, see III.D.

Figure 13 shows the clustering results obtained by 7 algorithms for the S2 dataset. S2 is a dataset with many data points, consisting of 15 irregular clusters. Due to the contact between the clusters, it is difficult to assign all the remaining points exactly. For the K-means, it cannot correctly find the cluster center of the three clusters in the top and the top right corner. The DBSCAN cannot correctly distinguish the three clusters in the lower right corner. The ACC of the other five algorithms can reach approximately 96%, and the DPC has better clustering performance than the other algorithms. Our LKSM_DPC performs better than the K-means, DBSCAN, FKNN-DPC, and DPCSA.

Table 4 summarizes the ACC, ARI, and AMI of 7 algorithms on the synthetic datasets, containing the parameter settings. As can be observed in the table, for the 10 synthetic

TABLE 5. ACC, ARI, and AMI of 7 algorithms on the UCI datasets.

Algorithm	ACC	ARI	AMI	Par	ACC	ARI	AMI	Par
		Wine				***Iris***		
K-means	0.9494	0.8471	0.8301	3	0.8867	0.7163	0.7331	3
DBSCAN	0.8146	0.5292	0.5484	0.5/21	0.7400	0.6120	0.5692	0.12/5
DPC	0.8315	0.5716	0.6461	2%	0.8867	0.7196	0.7668	2%
DPC-KNN	0.8933	0.6990	0.7228	8%	0.9600	0.8857	0.8605	2%
FKNN-DPC	0.9551	0.8708	0.8550	7	**0.9733**	**0.9222**	**0.9124**	7
DPCSA	0.9101	0.7414	0.7480	-	0.9667	0.9038	0.8831	-
LKSM_DPC	**0.9607**	**0.8804**	**0.8565**	45	0.9667	0.9038	0.8831	29
		Seeds				***Ionosphere***		
K-means	0.8905	0.7049	0.6705	3	0.7123	0.1776	0.1294	2
DBSCAN	0.6905	0.5291	0.5302	0.24/16	**0.9145**	**0.6835**	**0.5520**	0.78/9
DPC	0.9000	0.7341	0.7172	2%	0.6752	0.1191	0.0764	2%
DPC-KNN	0.9143	0.7664	0.7303	2%	0.7379	0.2183	0.1355	1%
FKNN-DPC	**0.9286**	**0.8024**	**0.7757**	8	0.7520	0.2840	0.3550	8
DPCSA	0.8810	0.6873	0.6609	-	0.7350	0.2135	0.1335	-
LKSM_DPC	0.9048	0.7419	0.7063	42	0.8291	0.4189	0.3077	15
		Wdbc				***Segmentation***		
K-means	0.9279	0.7302	0.6110	2	0.6095	0.5013	0.5927	7
DBSCAN	0.8471	0.4786	0.3581	0.46/38	0.4143	0.2129	0.3693	0.5/4
DPC	0.8260	0.4106	0.3641	2%	**0.7714**	**0.6253**	**0.6952**	2%
DPC-KNN	0.9121	0.6760	0.5576	1%	0.6381	0.5412	0.6320	8%
FKNN-DPC	0.8401	0.4502	0.3974	7	0.5762	0.4128	0.5456	7
DPCSA	0.8137	0.3771	0.3361	-	0.6286	0.4211	0.5136	-
LKSM_DPC	**0.9508**	**0.8114**	**0.7043**	114	**0.7714**	**0.6235**	**0.6952**	26
		Glass				***Libras Movement***		
K-means	0.4439	0.1669	0.2951	6	0.4528	0.3261	0.5416	15
DBSCAN	0.3785	0.0528	0.1819	0.1/4	0.3444	0.1948	0.4217	0.9/2
DPC	0.4159	0.1118	0.1667	2%	0.4306	0.3128	0.5326	0.3%
DPC-KNN	0.4626	0.1348	0.2347	1%	0.4361	0.2694	0.4778	1%
FKNN-DPC	0.4065	0.1074	0.2239	7	0.4111	0.3270	0.5302	9
DPCSA	0.4486	0.1754	0.1585	-	**0.4667**	**0.3553**	**0.5709**	-
LKSM_DPC	**0.4860**	**0.2083**	**0.2462**	11	**0.4667**	0.3113	0.5207	36
		Dermatology				***Waveform***		
K-means	0.4553	0.3542	0.6049	6	0.5006	0.2535	0.3641	3
DBSCAN	0.5894	0.4106	0.5779	0.99/3	0.3538	0.0022	0.0065	0.38/5
DPC	0.7374	0.6554	0.7999	2%	0.5866	0.2794	0.3507	2%
DPC-KNN	0.7402	0.6349	0.7731	2%	0.6608	0.2829	0.3211	0.2%
FKNN-DPC	0.7737	0.7299	**0.8645**	7	0.6712	**0.3406**	**0.3590**	5
DPCSA	0.8017	0.6772	0.7758	-	0.6350	0.2381	0.2603	-
LKSM_DPC	**0.9330**	**0.8661**	0.8631	29	**0.6796**	0.3036	0.2989	150
		Parkinsons				**Pima**		
K-means	0.6308	0.0520	0.2129	2	**0.6680**	**0.1024**	**0.0503**	2
DBSCAN	0.5949	0.0252	0.0071	0.5/17	0.6549	0.0118	0.0029	0.4/4
DPC	0.7385	0.1989	0.0994	2%	**0.6680**	0.0647	0.0213	2%
DPC-KNN	0.8205	0.2686	0.1772	2%	0.6523	0.0023	0.0008	2%
FKNN-DPC	0.8205	0.2686	0.1772	5	0.6549	0.0088	0.0028	6
DPCSA	0.8205	0.2686	0.1772	-	0.6523	0.0023	0.0008	-
LKSM_DPC	**0.8513**	**0.3910**	**0.2728**	20	**0.6680**	0.0681	0.0226	14
		SCADI				***Letter***		
K-means	0.6286	0.4682	0.5135	7	0.2583	0.1327	0.3545	26
DBSCAN	-	-	-	-	-	-	-	-
DPC	0.6143	0.5044	0.4543	2%	0.1773	0.0632	0.2281	2%
DPC-KNN	0.6571	0.6588	0.5509	2%	0.2316	0.0739	0.2875	2%
FKNN-DPC	0.7286	0.6191	0.5319	6	0.1174	0.004	0.2204	400
DPCSA	0.6857	0.5939	0.4988	-	0.2276	0.0354	0.4118	-
LKSM_DPC	**0.7857**	**0.7183**	**0.6251**	4	**0.3124**	**0.1551**	**0.4164**	400

datasets, our algorithm has better clustering performance for 8 datasets than the other algorithms, and it only slightly lags behind other algorithms on the remaining 2 datasets. Therefore, it shows that the algorithm proposed in this paper is effective on synthetic datasets.

B. EXPERIMENTS USING UCI DATASETS

To further verify the effectiveness of our proposed algorithm. We compared it with the other 6 algorithms on 14 UCI datasets. The clustering results of 7 algorithms on the UCI datasets are presented in Table 5. Each clustering index of the LKSM_DPC algorithm is significantly better than those of the other algorithms on the Wine, Ionosphere, Wdbc, Glass, Parkinsons, SCADI, and Letter datasets. Among these datasets, the subcluster merging strategy used in our algorithm has an obvious effect on the clustering quality. The FKNN-DPC performs best on the Iris dataset. The LKSM_DPC is only slightly behind the FKNN-DPC in

Algorithm 2 LKSM_DPC

Input: A dataset $x \epsilon R_{N\times M}$ ($R_{N\times M}$ is the data matrix, where N denotes the total number of datasets, and M represents the dimensions of the datasets), the parameter K
Output: A label vector of the cluster index: $y \epsilon R_{N\times 1}$

1. Calculate the Euclidean distance matrix using $R_{N\times M}$;
2. **for** each $x_i \in R_{N\times M}$ **do**
3. Calculate ρ_i and δ_i respectively using equation (11)and (3);
4. **end for**
5. Construct the decision graph using equation (4), and sort γ_i in descending order;
6. Select the appropriate subcluster centers C ($C \geq$ the correct number of the clusters) in the decision graph;
7. Allocate the remaining data points (except for the cluster center) to the nearest point of higher density;
8. Get the initial cluster $y \epsilon R_{N\times 1}$;
9. **if** $C ==$ the correct number of the clusters **then**
10. **return** $\boldsymbol{y}$;
11. **else**
12. $y \leftarrow$ Subcluster_merging(x, y, C);
13. **end if**
14. **return** $\boldsymbol{y}$;

Algorithm 3 Subcluster_merging

Input: data set x, y, and C
Output: y

1. **for** each $c_i \in C$ **do**
2. **for** each $c_j \in C$, $i \neq j$ **do**
3. $SNN\left(c_i, c_j\right) \leftarrow c_i \cap c_j$ (using equation (12));
4. **if** $SNN\left(c_i, c_j\right) \neq 0$ **then**
5. Calculate the subcluster similarity SIM using equation (14);
6. **end if**
7. **end for**
8. **end for**
9. Sort the SIM in descending order;
10. **while** $C \neq$ the correct number of the clusters **do**
11. Merge the most similar subclusters according to SIM;
12. $C \leftarrow C - 1$;
13. **end while**
14. Get the clustering result $y \epsilon R_{N\times 1}$;
15. **return** $\boldsymbol{y}$;

ACC, which is the same as the DPCSA's clustering performance. The FKNN-DPC also achieved the best clustering performance on the Seeds dataset, our LKSM_DPC's clustering performance outperforms those of K-means, DBSCAN, DPC, and the DPCSA. On the Ionosphere dataset, the result of DBSCAN is better than other algorithms, but our algorithm is second. The LKSM_DPC and DPC obtain the same clustering performance on the Segmentation dataset, and they are also considerably better than other algorithms. On the Libras dataset, the ACC index of the LKSM_DPC is the same as that of the DPCSA, but its ARI and AMI are slightly behind those of the DPCSA. On the Dermatology dataset, the LKSM_DPC is dramatically better than six other algorithms in terms of its ACC and ARI and only lags behind the FKNN-DPC in AMI. On the Waveform dataset, our algorithm only surpasses other algorithms on the ACC indicator, and it's ARI and AMI lag behind those of the FKNN-DPC. On the Pima dataset, the result of k-means is best. Nevertheless, the ACC of LKSM_DPC is the same as that of the K-means and DPC.

In summary, the LKSM_DPC is markedly better than the other 6 algorithms on most UCI datasets, and it is slightly behind the FKNN-DPC on the Seeds and Iris datasets. On the Ionosphere dataset, our algorithm lags behind the DBSCAN. On the other datasets, the LKSM_DPC can obtain the highest ACC, and only its ARI and AMI indicators are behind those of other algorithms. Therefore, the proposed algorithm is effective on most UCI datasets.

C. TIME COMPLEXITY ANALYSIS

Tables 6 and 7 show the run time of the K-means, DBSCAN, DPC, DPC-KNN, FKNN-DPC, DPCSA, and LKSM_DPC algorithms on the synthetic datasets and UCI datasets, respectively.

As can be seen from Table 6, on the synthetic datasets, the run time of the FKNN-DPC is significantly longer than those of other algorithms, mainly because the remaining point allocation of this algorithm is highly time-consuming, especially on large datasets, such as S2. Although the algorithm in

TABLE 6. Run time of 7 algorithms on the synthetic datasets.

Datasets	K_means	DBSCAN	DPC	DPC-KNN	FKNN-DPC	DPCSA	LKSM_DPC
D31	1.9318	0.3155	1.1638	0.7843	9.8460	2.0055	1.2381
R15	0.0163	0.0649	0.1872	0.1889	0.5082	0.2195	0.2112
Jain	0.0099	0.0523	0.1702	0.1775	0.5657	0.1708	0.2372
Spiral	0.0081	0.0058	0.1691	0.1456	0.2901	0.1859	0.1578
Aggregation	0.0121	0.0187	0.2316	0.1842	0.9830	0.2379	0.2587
Ring	0.0097	0.0310	0.1507	0.2167	3.1910	0.1970	0.2017
Zigzag	0.0111	0.0294	0.2574	0.2029	1.6083	0.3020	0.3292
Flame	0.0084	0.0611	0.1424	0.1480	0.2725	0.1641	0.1791
Pathbased	0.0059	0.0235	0.1520	0.1534	0.2922	0.1612	0.3922
S2	0.0282	0.8068	2.5557	1.9633	49.1123	5.9797	1.9407

TABLE 7. Run time of 7 algorithms on the UCI datasets.

Datasets	K_means	DBSCAN	DPC	DPC-KNN	FKNN-DPC	DPCSA	LKSM_DPC
Wine	0.0082	0.0071	0.1614	0.1577	0.2229	0.1644	0.1582
Iris	0.0059	0.0070	0.1498	0.1463	0.2158	0.1571	0.1558
Seeds	0.0068	0.0082	0.1092	0.1552	0.2216	0.1112	0.1311
Ionosphere	0.0141	0.0116	0.1693	0.1626	0.5284	0.1873	0.1996
Wdbc	0.0065	0.0240	0.1916	0.1962	0.4890	0.2083	0.1758
Segmentation	0.0071	0.0084	0.1504	0.1818	0.2535	0.1603	0.1997
Glass	0.0083	0.0108	0.1553	0.1701	0.2321	0.1649	0.1905
Libras	0.0264	0.0293	0.2082	0.2291	0.3310	0.1911	0.2287
Dermatology	0.0068	0.0145	0.1584	0.1793	0.4320	0.1700	0.2196
Waveform	0.0208	0.8091	2.6703	1.9680	72.806	6.3695	2.2835
Parkin.	0.0110	0.0285	0.1433	0.1406	0.3389	0.1429	0.1765
Pima	0.0059	0.0268	0.1890	0.1796	1.0639	0.2629	0.2123
SCADI	0.0102	-	0.3207	0.1576	0.5073	0.1722	0.2014
Letter	1.2755	-	38.4269	41.2109	731.3221	187.1137	33.0661

this paper calculated the subcluster similarity and proposed a subcluster merging strategy, it does not significantly increase the experimental time. The run time of LKSM_DPC is at the same level as those of DPC and the DPC-KNN.

As shown by the run time on UCI datasets in Table 7, the most time-consuming algorithm is still the FKNN-DPC, followed by the DPCSA. The foremost reason is that the FKNN-DPC and DPCSA proposed a new remaining point allocation strategy. The run time of our LKSM_DPC algorithm remains at the same level as DPC and the DPC-KNN.

V. CONCLUSIONS

This paper focuses on the problems of uneven density distribution and multiple peaks in the same cluster, and we proposed an improved LKSM_DPC algorithm to address these problems. The algorithm first designed a new local density calculation method based on the idea of the k-nearest neighbors and divided the k-nearest neighbors into multiple layers to accommodate datasets with uneven density distribution. Second, based on the ideas of shared neighbors and gravitation, a new method for calculating the similarity of subclusters is proposed, and a strategy for subcluster merging is also proposed. The LKSM_DPC is tested on a large number of datasets. The experimental results show that our algorithm's clustering performance is often better than those of the other 6 algorithms.

Although this algorithm can effectively deal with uneven density distribution and multiple peak datasets, there are still several shortcomings. First, the layering k-nearest neighbors strategy requires many experiments to determine the specific layers. Second, how many candidate cluster centers are selected for subcluster merging also needs many experiments to determine. Therefore, the following research will focus on solving the above problems, such as proposing a strategy for automatically selecting candidate cluster centers.

REFERENCES

[1] R. Xu and D. Wunsch, II, "Survey of clustering algorithms," *IEEE Trans. Neural Netw.*, vol. 16, no. 3, pp. 645–678, May 2005.

[2] D. Cheng, Q. Zhu, J. Huang, Q. Wu, and L. Yang, "A novel cluster validity index based on local cores," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 30, no. 4, pp. 1–15, Apr. 2018.

[3] A. K. Jain, M. N. Murty, and P. J. Flynn, "Data clustering: A review," *ACM Comput. Surv.*, vol. 31, no. 3, pp. 264–323, Sep. 1999.

[4] J. Huang, Q. Zhu, L. Yang, D. Cheng, and Q. Wu, "QCC: A novel clustering algorithm based on quasi-cluster centers," *Mach. Learn.*, vol. 106, no. 3, pp. 337–357, Mar. 2017.

[5] J. Han, M. Kamber, and J. Pei, *Data Mining: Concepts and Techniques*, Waltham, MA, USA: Morgan Kaufman, 2011, pp. 443–450.

[6] A. K. Jain, "Data clustering: 50 years beyond K-means," *Pattern Recognit. Lett.*, vol. 31, no. 8, pp. 651–666, Jun. 2010.

[7] M. Ester, H. Kriegel, J. Sander, and X. Xu, "A density-based algorithm for discovering clusters in large spatial databases with noise," in *Proc. 2nd Int. Conf. Knowl. Discovery Data Mining*, Portland, Oregon, Aug. 1996, pp. 226–231.

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

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

[10] J. Xie, H. Gao, W. Xie, X. Liu, and P. W. Grant, "Robust clustering by detecting density peaks and assigning points based on fuzzy weighted K-nearest neighbors," *Inf. Sci.*, vol. 354, pp. 19–40, Aug. 2016.

[11] R. Liu, H. Wang, and X. Yu, "Shared-nearest-neighbor-based clustering by fast search and find of density peaks," *Inf. Sci.*, vol. 450, pp. 200–226, Jun. 2018.

[12] Z. Li and Y. Tang, "Comparative density peaks clustering," *Expert Syst. Appl.*, vol. 95, pp. 236–247, Apr. 2018.

[13] D. Cheng, Q. Zhu, J. Huang, and L. Yang, "Natural neighbor-based clustering algorithm with density peeks," in *Proc. Int. Joint Conf. Neural Netw. (IJCNN)*, Jul. 2016, pp. 92–98.

[14] L. Yaohui, M. Zhengming, and Y. Fang, "Adaptive density peak clustering based on K-nearest neighbors with aggregating strategy," *Knowl.-Based Syst.*, vol. 133, pp. 208–220, Oct. 2017.

[15] M. Du, S. Ding, Y. Xue, and Z. Shi, "A novel density peaks clustering with sensitivity of local density and density-adaptive metric," *Knowl. Inf. Syst.*, vol. 59, no. 2, pp. 285–309, May 2019.

[16] C. Wu, J. Lee, T. Isokawa, J. Yao, and Y. Xia, "Efficient clustering method based on density peaks with symmetric neighborhood relationship," *IEEE Access*, vol. 7, pp. 60684–60696, 2019.

[17] L. Sun, R. Liu, J. Xu, and S. Zhang, "An adaptive density peaks clustering method with Fisher linear discriminant," *IEEE Access*, vol. 7, pp. 72936–72955, 2019.

[18] J. Jiang, Y. Chen, X. Meng, L. Wang, and K. Li, "A novel density peaks clustering algorithm based on k nearest neighbors for improving assignment process," *Phys. A, Stat. Mech. Appl.*, vol. 523, pp. 702–713, Jun. 2019.

[19] R. Mehmood, G. Zhang, R. Bie, H. Dawood, and H. Ahmad, "Clustering by fast search and find of density peaks via heat diffusion," *Neurocomputing*, vol. 208, pp. 210–217, Oct. 2016.

[20] M. Parmar, D. Wang, X. Zhang, A.-H. Tan, C. Miao, J. Jiang, and Y. Zhou, "REDPC: A residual error-based density peak clustering algorithm," *Neurocomputing*, vol. 348, pp. 82–96, Jul. 2019.
[21] L. Zhuo, K. Li, B. Liao, H. Li, X. Wei, and K. Li, "HCFS: A density peak based clustering algorithm employing a hierarchical strategy," *IEEE Access*, vol. 7, pp. 74612–74624, 2019.
[22] R. Wang and Q. Zhu, "Density peaks clustering based on local minimal spanning tree," *IEEE Access*, vol. 7, pp. 108438–108446, 2019.
[23] X. Xu, S. Ding, H. Xu, H. Liao, and Y. Xue, "A feasible density peaks clustering algorithm with a merging strategy," *Soft Comput.*, vol. 23, no. 13, pp. 5171–5183, Jul. 2019.
[24] M. D. Parmar, W. Pang, D. Hao, J. Jiang, W. Liupu, L. Wang, and Y. Zhou, "FREDPC: A feasible residual error-based density peak clustering algorithm with the fragment merging strategy," *IEEE Access*, vol. 7, pp. 89789–89804, 2019.
[25] D. Cheng, J. Huang, S. Zhang, and H. Liu, "Improved density peaks clustering based on shared-neighbors of local cores for manifold data sets," *IEEE Access*, vol. 7, pp. 151339–151349, 2019.
[26] D. Qiao, Y. Liang, and L. Jiao, "Boundary detection-based density peaks clustering," *IEEE Access*, vol. 7, pp. 152755–152765, 2019.
[27] J. Jiang, X. Tao, and K. Li, "DFC: Density fragment clustering without peaks," *J. Intell. Fuzzy Syst.*, vol. 34, no. 1, pp. 525–536, Jan. 2018.
[28] D. Yu, G. Liu, M. Guo, X. Liu, and S. Yao, "Density peaks clustering based on weighted local density sequence and nearest neighbor assignment," *IEEE Access*, vol. 7, pp. 34301–34317, 2019.
[29] L. Bai, X. Cheng, J. Liang, H. Shen, and Y. Guo, "Fast density clustering strategies based on the k-means algorithm," *Pattern Recognit.*, vol. 71, pp. 375–386, Nov. 2017.
[30] J. Jiang, D. Hao, Y. Chen, M. Parmar, and K. Li, "GDPC: Gravitation-based density peaks clustering algorithm," *Phys. A, Stat. Mech. Appl.*, vol. 502, pp. 345–355, Jul. 2018.
[31] J. Jiang, Y. Chen, D. Hao, and K. Li, "DPC-LG: Density peaks clustering based on logistic distribution and gravitation," *Phys. A, Stat. Mech. Appl.*, vol. 514, pp. 25–35, Jan. 2019.
[32] T. Cover and P. Hart, "Nearest neighbor pattern classification," *IEEE Trans. Inf. Theory*, vol. IT-13, no. 1, pp. 21–27, Jan. 1967.
[33] L. Sun, S. Bao, S. Ci, X. Zheng, L. Guo, and Y. Luo, "Differential privacy-preserving density peaks clustering based on shared near neighbors similarity," *IEEE Access*, vol. 7, pp. 89427–89440, 2019.
[34] N. X. Vinh, J. Epps, and J. Bailey, "Information theoretic measures for clusterings comparison: Variants, properties, normalization and correction for chance," *J. Mach. Learn. Res.*, vol. 11, pp. 2837–2854, Jan. 2010.

CHUNHUA REN is currently pursuing the Ph.D. degree with the School of Information Science and Technology, Southwest Jiaotong University, Chengdu, China. He has published three journal articles and conference papers at the CIMS and ISKE. His main research interests include machine learning and data mining.

LINFU SUN received the Ph.D. degree from Southwest Jiaotong University, in 1993. He is currently a Chief Professor of Southwest Jiaotong University and is the chief scientist of modern services. He has published more than 100 journal articles and conference papers. His main research interests include cloud platform technology, manufacturing industry chain collaboration technology, and manufacturing industry data mining.

YANG YU received the B.S. degree from the University of Electronic Science and Technology, Chengdu, China, in 2011. He is currently pursuing the Ph.D. degree with the School of Information Science and Technology, Southwest Jiaotong University, China. His research interests are in the areas of cloud platform technology, data intelligence, and data mining.

QISHI WU (Member, IEEE) received the B.S. degree from Zhejiang University, China, in 1995, the M.S. degree in geomatics from Purdue University, in 2000, and the Ph.D. degree in computer science from Louisiana State University. He was a Research Fellow of the Division of Computer Science and Mathematics, Oak Ridge National Laboratory, from 2003 to 2006, and an Assistant Professor and an Associate Professor with the Department of Computer Science, The University of Memphis, from 2006 to 2015. He is currently a Professor with the Department of Computer Science, New Jersey Institute of Technology, and Southwest Jiaotong University. He has published more than 300 articles. His research interests include big data, distributed computing, computer networks, and scientific visualization.