Density Peaks Clustering Algorithm with Connected Local Density and Punished Relative Distance

Jingwen Xiong
Shandong Normal University
Wenke Zang (wink@sdnu.edu.cn)
Shandong Normal University
Yuzhen Zhao
Shandong Normal University
Xiyu Liu
Shandong Normal University

Research Article

Keywords: Density peaks clustering method, Flexible connectivity distance, Connected k-nearest neighbor, Punished relative distance

Posted Date: May 25th, 2023

DOI: https://doi.org/10.21203/rs.3.rs-2965154/v1

Additional Declarations: No competing interests reported.

Density Peaks Clustering Algorithm with Connected Local Density and Punished Relative Distance

Jingwen Xiong [1] · Wenke Zang [1] * · Yuzhen Zhao [1] · Xiyu Liu [1]

Abstract

Density peaks clustering (DPC) algorithm has been widely applied in many fields due to its innovation and efficiency. However, the original DPC algorithm and many of its variants choose Euclidean distance as local density and relative distance estimations, which affects the clustering performance on some specific shaped datasets, such as manifold datasets. To address the above-mentioned issue, we propose a density peak clustering algorithm with connected local density and punished relative distance (DPC-CLD-PRD). Specifically, the proposed approach computes the distance matrix between data pairs using the flexible connectivity distance metric. Then, it calculates the connected local density of each data point via combining the flexible connectivity distance measure and k-nearest neighbor method. Finally, the punished relative distance of each data point is obtained by introducing a connectivity estimation strategy into the distance optimization process. Experiments on synthetic, real-world, and image datasets have demonstrated the effectiveness of the algorithm in this paper.

Keywords Density peaks clustering method · Flexible connectivity distance · Connected k-nearest neighbor · Punished relative distance

1 Introduction

With the development of Artificial Intelligence, Big Data, the Internet of Things, and other Internet technologies, a large amount of data is produced and further collected from various walks of life [1]. Taking this into account, data mining is further developed and improved to identify and seek valuable, novel, and valid information from various types of data [2, 3].

Clustering analysis, one of the most active unsupervised learning approaches in the field of data mining, aims at classifying data points in one cluster into several sub clusters so that similar data points are divided into the same group while dissimilar data points are divided into different groups [4-7]. Over the past decades, clustering analysis has been serving many applications in the field of machine learning [8-10], pattern segmentation [11-14], and recommendation [15-17]. Up to now, different kinds of clustering schemes have been developed, density-based clustering methods [18, 19], partition-based clustering methods [20-22], hierarchical-based clustering methods [23, 24], grid-based clustering methods [25, 26], model-based clustering methods [27-29], graph-based clustering methods [30, 31]and others.

K-means, the most classic and well-known partition-based clustering algorithm, has emerged as a flexible and efficient approach for its simplicity in procedure and efficiency in clustering [32, 33]. K-means clustering algorithm first finds specific numbers of initial cluster centers and then minimizes the sum of squared distances between data points and their nearest centers [34]. However, K-means algorithm suffers two obvious shortcomings that its clustering performance is easily affected by the original cluster centers and it fails to identify arbitrary shapes of clusters [35]. Hierarchical-based clustering algorithm, an important way to construct embedded classification schemes, has been widely applied since its publication and has been generalized in a variety of forms [36, 37]. Hierarchical-based methods perform clustering by producing a nested hierarchy of clusters. Sequence steps of them can be implemented as either a bottom-up (agglomerative) approach or a top-down (divisive) method [38]. Classic CHAMELEON [39] and CURE [40] are agglomerative and divisive hierarchical clustering algorithms respectively. Grid-based clustering algorithms enjoy extremely high attention due to their low complexity in computation and high efficiency in processing spatial datasets [41, 42]. STING [43], one of the most classic grid-based clustering algorithms, generates statistical information of several individual grid units by dividing the original spatial data instead of scanning all individual points to reduce the time complexity further. Despite its simplicity in computation, current grid-based clustering algorithms still suffer some problems that it is incapable to handle high-dimensional and arbitrary shapes of datasets [44]. Model-based clustering algorithm, popular for its probabilistic foundations and flexibility in implementation, is gaining global significance and has shown promising performance during past decades [45]. Classic model-based EM [46] approach maximizes the conditional expectation

of the complete log-likelihood iteratively to estimate parameters. With the development of clustering analysis, a fast model-based Newton EM [47] algorithm is proposed and is combined with the coordinate descent EM method to reintroduce GMMs pattern recognition community. However, there still exist several obvious deficiencies like the uncertainty of parameters, models, and distribution in grid-based clustering algorithms [48]. Spectral clustering algorithm [49] and multi-view subspace clustering [50] algorithms enjoy the highest attention among a variety of graph-based clustering algorithms in past decades. Graph-based clustering algorithms learn a common affinity matrix firstly based on the original data and further apply the procedure of k-means method to perform the clustering process [51]. Many existing graph-based clustering algorithms utilize the two-step strategy we discussed above and are able to obtain satisfactory clustering results on some specific datasets such as manifold datasets [52]. However, existing graph-based clustering algorithms still have some deficiencies such as inadequate mining of potential information in multi-view data [53]. Density-based clustering algorithms have enjoyed a high profile and have been widely used in various fields, able to recognize non-spherical and irregularly shaped clusters [54]. A density-based algorithm for discovering clusters in large spatial databases with noise (DBSCAN) is the earliest and the most classic density-based algorithm, but it suffers high parameter sensitivity and complexity [18].

Clustering by fast search and find of density peaks (DPC) algorithm, another classic density-based method, was published by Alex Rodriguez and Alessandro Laio in Science in 2014 [19]. In contrast to DBSCAN, DPC applies only one parameter to calculations of local density and relative distance of each data point, which leads to lower parameter sensitivity. Compared with K-means, DPC, according to Euclidean distance, assigns remaining points to their nearest cluster centers without iteration, which leads to a simpler process [55]. In addition, DPC can get satisfactory performance on non-spherical datasets, overcoming the weakness of some partition-based clustering methods. However, there still exists great room for improvement in calculations of local density and relative distance, the allocation strategy, and the selection of cluster centers automatically.

First of all, a variety of DPC variants are proposed to innovate estimations of local density and relative distance of each data point. Zhang et al. [56] improved the original DPC using balance density and connectivity and further proposed the BC-DPC algorithm. BC-DPC creates balance density to eliminate the density difference of different clusters to identify the cluster centers precisely. Zhao et al. [48] innovated the local density estimation based on the nearest neighbor fuzzy kernel function and further proposed the density peaks clustering algorithm based on fuzzy and weighted shared neighbor for uneven density datasets (DPC-FWSN). DPC-FWSN defines the density weights of data points in dense and sparse regions to better adapt to uneven-density datasets. Ding et al. [57] redefined the relative distance estimation based on a sampling method and further proposed a novel sampling-based density peaks clustering algorithm for large-scale data (SDPC). SDPC reduces the time complexity of relative distance calculation and performs more satisfactorily than the original DPC algorithm. Rasool et al. [58] created a novel data-dependent similarity measure according to Probability Mass (MP-Similarity) and further proposed MP-DPC, a data-dependent variant of the original DPC, by applying MP-Similarity to the DPC algorithm. MP-DPC utilizes MP-Similarity as a similarity metric, able to get more satisfactory performance than using the Euclidean distance.

Secondly, the creation of the allocation strategy of DPC has been gaining greater significance in the past decades. Ding et al. [59] proposed an improved density peaks clustering algorithm based on natural neighbor with a merging strategy (IDPC-NNMS). IDPC-NNMS recognizes as many centers as possible to form the initial sub-clusters, and then merge the sub-clusters based on an innovative allocation strategy to complete the clustering process. Lin et al. [60] improved the original DPC by automatic peak selection and single linkage methods and proposed "improving density peak clustering by automatic peak selection and single linkage clustering". It firstly identifies potential cluster centers automatically based on the radius of the neighborhood and then eliminated the domino effect in the original DPC algorithm by introducing a single-linkage approach.

Thirdly, in the area of selecting cluster centers, researchers have been devoted to identifying cluster centers more accurately and simply. Guan et al. [61] used a novel center assumption idea and further proposed clustering by fast detection of main density peaks within a peak digraph (MDPC+). MDPC+ considers clustering as a graph cut problem, able to identify the true centers of multi-peak clusters easily. Li et al. [62] introduced a relative semantic distance that concerns the distance between fuzzy semantic cells and further proposed an approach of fuzzy semantic cells to density peaks clustering (DPC-FSC). DPC-FSC applies the relative semantic distance, able to recognize the cluster centers in the decision graph in an informative manner. Tong et al. [63] proposed a density-peak-based clustering algorithm of automatically determining the number of clusters by introducing an automatic approach to determine the

true number of clusters. Although domestic and foreign scholars have made innovations to the original DPC from a variety of perspectives, it still does not work well in clustering some specific shapes of datasets, such as manifold datasets.

In this paper, a novel density peaks clustering algorithm with connected local density and punished relative distance (DPC-CLD-PRD) is introduced to address the above deficiencies. DPC-CLD-PRD first calculates the flexible connectivity distance between data pairs. Next, the flexible connectivity distance is selected as a similarity measure to calculate the connected k-nearest neighbor of data points and the connected density of data points is further calculated. At last, a connectivity estimation strategy is applied to improve the relative distance estimation. The key contributions of this paper are summarized as follows:

1. We use the flexible connectivity distance metric instead of Euclidean distance to calculate the distance matrix of data pairs.
2. We calculate the connected local density of each data point by combining flexible connectivity distance measure and k-nearest neighbor method.
3. We utilize a connectivity estimation strategy to perform distance punishment and further calculate the punished relative distance.
4. Extensive experiments on synthetic datasets, real-world datasets from UCI repository and Olivetti Face image dataset demonstrate that DPC-CLD-PRD outperforms the original DPC and its variants.

The remainder of this paper is organized as follows. In section 2, the idea and the flow of the original DPC are discussed in detail. In section 3, details of the proposed DPC-CLD-PRD algorithm are provided. In section 4, extensive experiments are carried out to verify the effectiveness and feasibility of the proposed algorithm. In section 5, we summarize this paper overall.

2 Related works

In this section, some relevant contents about the original DPC algorithm and the flexible connectivity distance are introduced in detail.

2.1 DPC algorithm

As stated in the introduction section, the original DPC algorithm has enjoyed a great profile during the past decade. The high efficiency and the feasibility of the original DPC approach are supported by two basic assumptions: a) cluster centers hold a comparatively larger local density; b) different cluster centers are located relatively far away from each other.

The basic workflow of the original DPC method is both efficient and understandable. From a holistic view, there are two variables to be calculated for each data point, including local density ρ and relative distance δ.

In the first place, the original DPC applies either a cut-off kernel or a Gaussian kernel for local density estimation. The local density ρ using cut-off kernel of data point i is defined by the following formula:

$$\rho_i = \sum_j \chi(d_{ij} - d_c), \tag{1}$$

where d_{ij} indicates the Euclidean distance between data point i and data point j in a dataset and d_c means the cut-off distance predefined. Note that $\chi(x)$ represents the indicator function defined by the following formula:

$$\chi(x) = \begin{cases} 1, & if \ d_{ij} < d_c \\ 0, & otherwise \end{cases}. \tag{2}$$

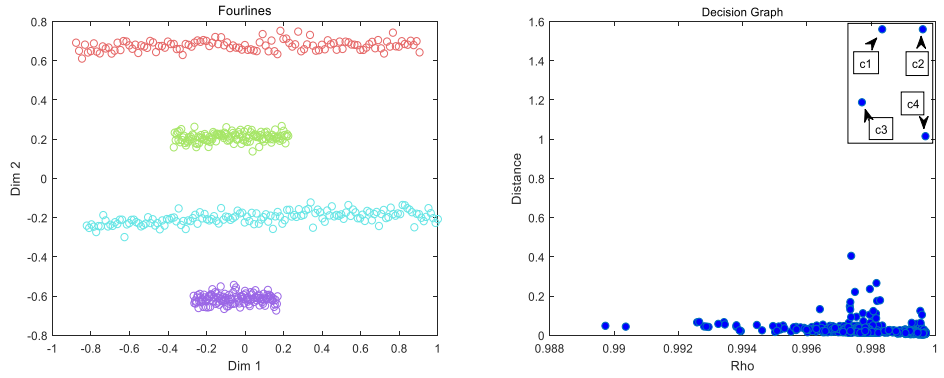
The local density can also be calculated by the Gaussian kernel, which is demonstrated by the following formula:

$$\rho_i = \sum_j \exp\left(-\left(\frac{d_{ij}}{d_c} \right)^2 \right). \tag{3}$$

With the local density of each data point obtained, the original DPC algorithm further gives the definition of the formula for calculating another variable, the relative distance, for each data point. The estimation of the relative distance δ is provided by the following formula:

$$\delta_i = \begin{cases} \max_j (d_{ij}), & if \ \rho_i = \rho_{\max} \\ \min_{j:\rho_j > \rho_i} (d_{ij}), & otherwise \end{cases} . \tag{4}$$

Then, a two-dimensional decision graph is further produced with local density ρ of all points in a dataset as horizontal coordinate and relative distance δ as vertical coordinate. After generating the decision graph, points with both the most prominent local density and relative distance, lying in the top-right position of the decision diagram, are identified as cluster centers. For instance, the ground-truth of Four-lines dataset and its decision graph with recognized cluster centers $c_1 - c_4$ in the rectangular box are displayed in Fig. 1(a) and Fig. 1(b) respectively.



(a) (b)

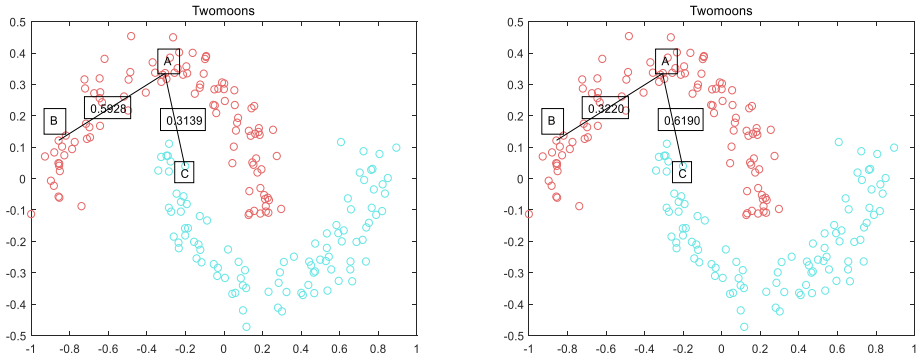
Fig. 1 (a) The ground-truth of Four-lines dataset (b) The decision graph with 4 selected cluster centers of Four-lines dataset

Finally, after cluster centers have been obtained, the original DPC algorithm performs a single-step assignment strategy. All remaining points are allocated to the same clusters as their nearest neighbors with higher density.

2.2 Flexible connectivity distance

Recall that the original DPC algorithm applies the Euclidean distance to calculate the local density and the relative distance, which is likely to bring various problems such as chain reaction. To overcome the above deficiencies, a novel graph-based flexible distance measure with connectivity information is proposed currently [64]. Considering the connectivity information of the dataset, the flexible distance connectivity distance achieves more satisfactory clustering performance than the Euclidean distance on certain specially shaped datasets, such as manifold datasets. By adding the connectivity information into the distance measure, the relative distance between data points in the same cluster is decreased while the distance between data points in different clusters is enlarged, thus better satisfying the global consistency of distance.

Take Two-moons dataset as an example, as is shown in Fig. 2(a), data points A and B in the same cluster should have higher similarity than points A and C in two different clusters. After calculation, the Euclidean distance between A and B, C is 0.5928 and 0.3139 respectively, indicating a higher similarity between A and C. In contrast, the flexible connectivity distance between A and B is 0.3220, smaller than that between A and C. As we can conclude, the Euclidean distance in some datasets like Jain can ignore global consistency between data points and result in allocation errors. On the contrary, the flexible connectivity distance enjoys higher efficiency and feasibility due to its global consistency.



(a) (b)

Fig. 2 (a) Euclidean distance among A, B and C (b) Flexible connectivity distance among A, B and C

3 DPC-CLD-PRD: density peaks clustering algorithm with connected local density and punished relative distance

In this section, we describe the workflow of the proposed density peaks clustering algorithm with connected local density and punished relative distance (DPC-CLD-PRD) in detail. In the first place, the distance matrix is obtained by applying the flexible connectivity distance metric instead of the Euclidean distance. Secondly, we define a novel density estimation method, combining flexible connectivity distance and *k*-nearest neighbor, for the connected local density calculation. Next, we introduce a special connectivity estimation strategy to optimize the flexible connectivity distance. At last, cluster centers are identified in the decision graph and connectivity distance is also used to allocate remaining data points in the same way as the original DPC.

3.1 Calculation of distance matrix

Recall that DPC fails to obtain satisfactory clustering performance on some specific-shaped datasets when using the Euclidean distance measure. Therefore, a flexible distance measure [32] which contains more connectivity information of data pairs, named flexible connectivity distance in this paper, is utilized to produce the distance matrix between data pairs. It is calculated by the following definition.

Definition 1 *(Flexible Connectivity Distance (FCD))* Define P_{ij} as the set that connects data points i and j, $|p|$ denotes the length of the whole path, and $d(p_k, p_{k+1})$ represents the Euclidean distance between two adjacent data points p_k and p_{k+1}. Then the flexible connectivity distance of data points i and j is defined as follows:

$$FCD_{i,j} = \frac{1}{\lambda}\ln(1+\min_{p_{ij}\in P_{ij}}\sum_{k=1}^{|p|}(e^{\lambda d(p_k,p_{k+1})}-1)), \tag{5}$$

where λ controls the scaling ratio of the distance between data pairs in the same cluster to the distance between data pairs in different groups. In this way, an improved distance matrix can be generated based on the above definition.

3.2 Estimation of local density

Inspired by the *k*-nearest neighbor approach and the flexible connectivity distance metric, we apply the flexible connectivity distance to the *k*-nearest neighbor approach and design a novel connected k-nearest neighbor approach. Besides, we further propose a new connected density estimation method. Firstly, the definition of *k*-nearest neighbor is provided by the following definition.

Definition 2 *(K-Nearest Neighbor (KNN))* Given a dataset X, the *k*-nearest neighbor of data point i is defined as follows:

$$KNN(i) = \{\, j \in X \mid d(i,j) \le d(i,k) \,\}, \tag{6}$$

where $d(i, j)$ represents the Euclidean distance between i and j in X, $d(i,k)$ represents the Euclidean distance between point i and the kth closest point to it.

It is obvious that the k-nearest neighbor method takes the Euclidean distance as a similarity metric to calculate nearest neighbors of data points. In the proposed DPC-CLD-PRD method, we apply the above flexible connectivity distance metric to the k-nearest neighbor method and further design the connected k-nearest neighbor approach. The flexible connectivity distance instead of the Euclidean distance of data pairs is selected as the similarity measure between them.

Definition 3 *(Connected K-Nearest Neighbor (CKNN))* Given a dataset X, the connected k-nearest neighbor of data point i is defined as follows:

$$CKNN(i) = \{ j \in X \mid FCD_{i,j} \le FCD_{i,k} \}, \tag{7}$$

where $FCD_{i,j}$ is the flexible connectivity distance between i and j in X, and $FCD_{i,k}$ is the flexible connectivity distance between i and the kth closest point to it.

Then, we improve the local density estimation by updating the connected k-nearest neighbor of data pairs. The enhanced connected local density of data point i is demonstrated as follows:

$$CLD_i = \exp\left(-(\frac{1}{k} \sum\nolimits_{j \in CKNN(i)} FCD_{i,j}{}^2) \right) , \tag{8}$$

where $CKNN(i)$ represents the connected k-nearest neighbor of data point i.

In this way, the optimized local density of each data point can better adapt to the structure of the dataset and help to obtain more satisfactory clustering results.

3.3 Calculations of the punished relative distance

The problem in the original DPC is that the Euclidean distance only considers the local information of data points but ignores the global consistency of the whole. Consequently, a special graph-based connectivity estimation strategy (CES) [64] is applied to punish the flexible connectivity distance and further produced the punished relative distance. Firstly, we provide explanations of some definitions involved in CES.

Definition 4 *(Connected Points)* Define data points i and j are recognized as connected only if the maximum distance of any two adjacent points on the path connecting points i and j is smaller than a given threshold T_{ij}.

Definition 5 *(Found Points)* For the path connecting points i and j, the next point with the max but less than T_{ij} distance to the previous point are recognized as found points.

Suppose n is the number of data points in a dataset, the detailed procedure of CES is described here. In the beginning, a threshold $T_{ij} = T_r * d_{ij}$ is defined for later connectivity estimation. Then, define the basic item of distance punishment dis_{basic} as the maximum value of the average distance of all found points. Take found point i as an example, the dis_{basic} of it is demonstrated by the following formula:

$$dis_{basic} = \max(mean(\sum\nolimits_{k=1}^{n} d_{ik} * \chi(d_{ik} - T_{ij}))) . \tag{9}$$

Then, record the number of all found points on a path as Num_{ij}, and add the second item $dis_{adaptive}$, defined by the following formula, into distance punishment to take a better application of the connectivity and spatial distribution between two points.

$$dis_{adaptive} = T_{ij} * (1 + \left(Num_{ij} - (\frac{1}{T_r} + 1) \right) * P_r) . \tag{10}$$

Finally, the punished relative distance of two points is further calculated by the following formula in the last step of CES:

$$PRD_i = dis^i{}_{basic} + dis^i{}_{adaptive} . \tag{11}$$

In Eq. (6) - (8), T_r and P_r two hyper parameters and values of them are fixed at 0.25 and 0.3 respectively on the basis of extensive experiments. In conclusion, the punished relative distance PRD_i of each data point is obtained after the punishment of the flexible connectivity distance. The specific procedure of CES is described in Algorithm 1.

Algorithm 1 process of CES

Input: Distance matrix, T_r, P_r, X (data set)

Output: PRD_i

Step 1: record the number of all found points on a path $Num_{i,j}$;

Step 2: calculate the basic item of distance punishment using formula (9);

Step 3: calculate the second item using formula (10);

Step 4: obtain the punished relative distance using formula (11);

Return: $Num_{i,j}$, dis_{basic}, $dis_{adaptive}$, PRD_i

3.4 Cluster centers and allocation of remaining points

With the punished relative distance and connected local density both obtained, we use the same decision graph method as the original DPC to identify cluster centers. Data points in the up-right corner of the decision graph are recognized as cluster centers.

Then in terms of the allocation of remaining points, we perform the single-step allocation strategy by replacing the Euclidean distance with the flexible connectivity distance. After the cluster centers have been found, the remaining data points are assigned to the same cluster as their nearest neighbor with higher connected local density according to the flexible connectivity distance. The detailed workflow of the proposed DPC-CLD-PRD algorithm is elaborated in Algorithm 2.

Algorithm 2 process of DPC-CLD-PRD

Input: distance matrix, data set X ,parameter λ , the number of nearest neighbors k

Output: the clustering results

Step 1: calculate the distance matrix using formula (5);

Step 2: calculate the connected local density of each data point using formula (8);

Step 3: calculate the punished relative distance of each data point using formula (9)-(11);

Step 4: choose cluster centers and allocate remain points;

Return: the clustering results

3.5 Complexity analysis of DPC-CLD-PRD

The whole time complexity of the proposed DPC-CLD-PRD algorithm is mainly composed of three following components: (1) calculating the punished relative distance of each data point $O(n^2)$; (2) calculating the updated connected local density of each data point $O(n^2)$; (3) assigning the remaining data points on the basis of the flexible connectivity distance $O(n)$. Consequently, the overall time complexity of DPC-CLD-PRD is $O(n^2)$.

4 Experiments

4.1 Experimental settings

In this section, nine two-dimensional synthetic datasets and eight real-world datasets from UCI repository are selected to verify the feasibility and efficiency of the proposed DPC-CLD-PRD method. Five algorithms related to DPC are chosen to compare with the proposed DPC-CLD-PRD algorithm on the basis of the same experiments. In addition, based on different algorithms described above, three common evaluation metrics are calculated over the same datasets to illustrate the effectiveness of the proposed method.

4.1.1 Datasets

In this section, nine two-dimensional synthetic datasets include Jain, Flame, Smile, Three-circles, Spiral, Compound, Aggregation, and R15 and eight real-world datasets include Vowel, Zoo, Blood, Ecoli, Wine, Seeds, Cancer, and Glass are selected in this paper to illustrate the feasibility and efficiency of the proposed method. Besides, we apply the Olivetti Face image dataset to the experiments. Detailed information about the dimensions, number of samples, and number of clusters of the datasets are provided in Tables 1 and 2 respectively.

Table 1 Two-dimensional synthetic datasets

Datasets	#samples	#cluster	#dimensions
Jain	373	2	2
Flame	240	2	2
Smile	266	3	2
Three-circles	299	3	2
Spiral	312	3	2
Compound	399	5	2
Aggregation	788	7	2
R15	600	15	2

4.1.2 Algorithms for comparison

In this section, we introduce the original DPC algorithm and four variants of DPC including MDPC, DPC-CE, DPC-MST, and DPC-LDP as comparative algorithms. Experiments on each individual DPC-related algorithm are conducted based on the same synthetic and real-world datasets in order to compare the clustering performance of the proposed DPC-CLD-PRD algorithm. In terms of Olivetti Face image dataset, we conduct a comparative experiment on DPC and DPC-CLD-PRD algorithm proposed in this paper.

Table 2 Real-world datasets from UCI repository

Datasets	#samples	#cluster	#dimensions
Vowel	871	6	3
Zoo	101	7	16
Blood	748	2	4
Ecoli	336	8	7
Wine	178	3	13
Seeds	210	3	7
Cancer	683	2	9
Glass	214	6	9

4.1.3 Parameter setting

In this section, the detailed parameter settings of the original DPC algorithm, four variants of DPC, and the proposed DPC-CLD-PRD algorithm on both 9 synthetic datasets and 8 real-world datasets from UCI repository are given in Table 3 and Table 4 as follows. Note that DPC-MST and FHC-DPC require the true number of clusters of datasets as their input.

Table 3 Parameter settings of six algorithms

Algorithms	Parameter settings	References
DPC	$d_c = 1\% \sim 2\%$	[19]
MDPC	$d_c = 2\%$, $K = 7$, $\theta = 5$	[32]
DPC-CE	$d_c = 2\%$	[64]
DPC-MST	$NC = \#cluster$	[7]
FHC-LDP	$k = 9$, $C = \#cluster$	[65]
DPC-CLD-PRD	$k = 10$, $\lambda = 1 \sim 100$	/

4.1.4 Evaluation metrics

In this section, clustering performance of five DPC-related algorithms and the proposed DPC-CLD-PRD are illustrated according to three common evaluation metrics including clustering accuracy (ACC), normalized mutual information (NMI), and rand index (RI). By evaluating labels obtained from the clustering results and original true labels, the clustering performance of different algorithms can be compared. Note that the larger values of evaluation indexes represent the more satisfactory clustering results of algorithms. Besides, the values of evaluation metrics are ranged between the interval [0, 1].

Suppose n represents the number of instances in a dataset, t_i and r_i are the true label and the obtained clustering label respectively. In this case, the detailed definition of ACC is given by the following formula:

$$ACC = \frac{\sum_{i=1}^{n} f(t_i, map(r_i))}{n}, \tag{12}$$

where $f(x, y) = \begin{cases} 1, & if\ x = y \\ 0, & otherwise \end{cases}$ is a discriminate function.

Suppose that n indicates the number of samples in a dataset. n_i and n_j denotes the number of samples in cluster i and cluster j respectively. In addition, n_{ij} represents samples in both group i and j. In this way, the specific definition of NMI is given by the following formula:

$$NMI = \frac{\sum_{i=1}^{k}\sum_{j=1}^{k} n_{ij} \log(\frac{n \cdot n_{ij}}{n_i \cdot n_j})}{\sqrt{(\sum_i n_i \log \frac{n_i}{n})(\sum_j n_j \log \frac{n_j}{n})}}. \tag{13}$$

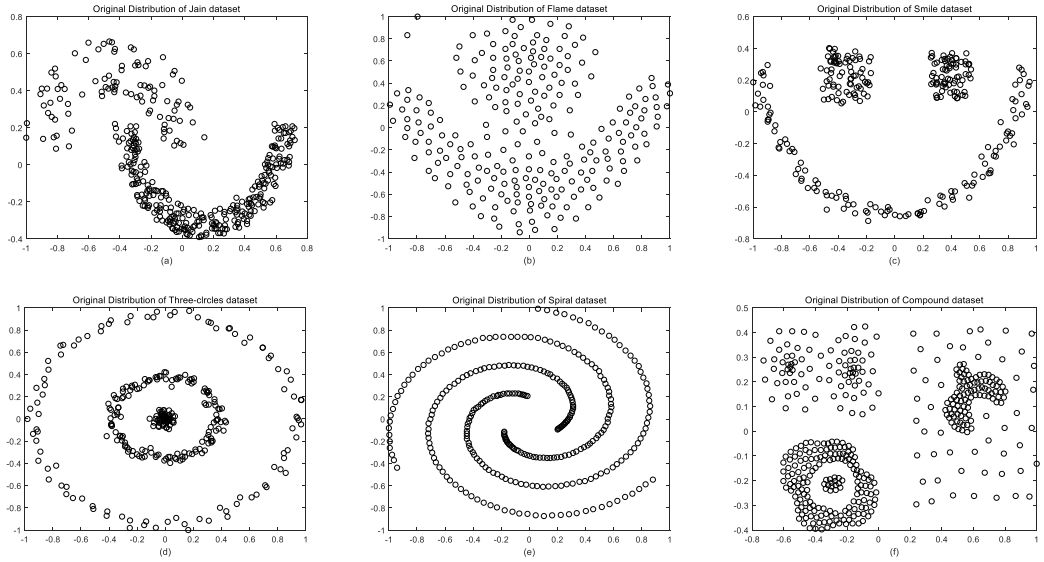
Here, we suppose a represents the number of instances of the same label and b represents the number of instances of different labels in t_i and r_i respectively. Therefore, the detailed definition of RI is given by the following formula:

$$RI = \frac{a+b}{C_n^2}, \tag{14}$$

where C represents the real category information of a dataset.

4.2 Experiments on synthetic datasets

In this section, experiments of the original DPC algorithm, four DPC-related methods, and the proposed DPC-CLD-PRD approach are provided on nine two-dimensional synthetic datasets. We display the original distribution of two-dimensional artificial datasets mentioned above in Fig. 3 to demonstrate the efficiency of the proposed algorithm in a more intuitive way. In addition, Fig. 4-Fig. 12 demonstrate the clustering results of six comparative algorithms above on nine two-dimensional synthetic datasets in detail. Note that figures (a)-(f) in Fig. 4-Fig. 12 represent the clustering performance of DPC, MDPC, DPC-CE, LDP-MST, FHC-DPC, and DPC-CLD-PRD respectively.



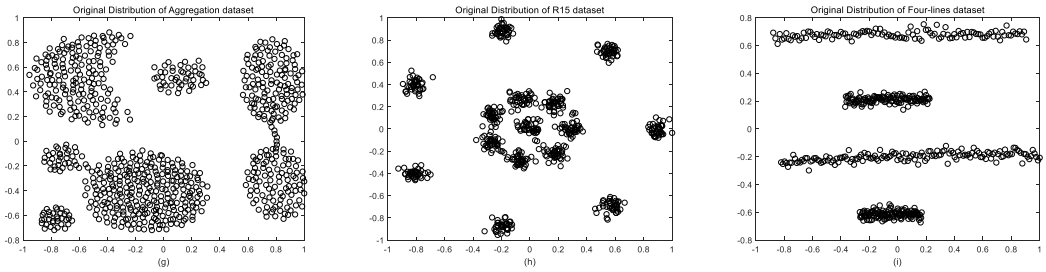


Fig. 3 The original distribution of 9 two-dimensional synthetic datasets

The clustering performance of six comparative algorithms on Jain, Flame, Smile, and Four-lines datasets are illustrated in Fig. 4-Fig. 7 as follows. As we can observe, the original DPC algorithm is not good at identifying the accurate distribution of clusters on Jain, Flame, Smile, and Four-lines datasets. On the contrary, the real distribution can be more accurately recognized by the rest five algorithms, including MDPC, DPC-CE, LDP-MST, FHC-LDP, and the proposed DPC-CLD-PRD.

Fig. 8 demonstrates the clustering results of six comparative algorithms on Three-circles data set. Three-circles, composed of a round group and two circles of data points, is a classic manifold dataset. As is exhibited in Fig. 8, the original DPC and DPC-CE algorithms fail to achieve rewarding clustering performance while MDPC, LDP-MST, FHC-LDP, and DPC-CLD-PRD methods are capable of obtaining satisfactory clustering results on the dataset.

The clustering performance on Spiral dataset, shown in Fig. 9, demonstrates that six comparative algorithms are all able to obtain the most optimal clustering performance. Note that the original DPC method can get the most accurate clustering result when using Cut-off kernel and $d_c = 2\%$.

As is shown in Fig. 10, the clustering accuracy of the proposed DPC-CLD-PRD method in this paper is the same as MDPC, DPC-CE, and FHC-LDP algorithms on Compound dataset. They are able to discover the correct distribution of data points. However, the original DPC and LDP-MST algorithms are incapable of obtaining great clustering performance.

Fig. 11 and Fig. 12 present the clustering results of six comparative algorithms on Aggregation and R15 datasets respectively. Except FHC-DPC, the clustering accuracy of the rest five algorithms is equally high overall. In addition, it can be seen from Fig. 12 that all six comparative algorithms perform pretty well on R15 dataset.

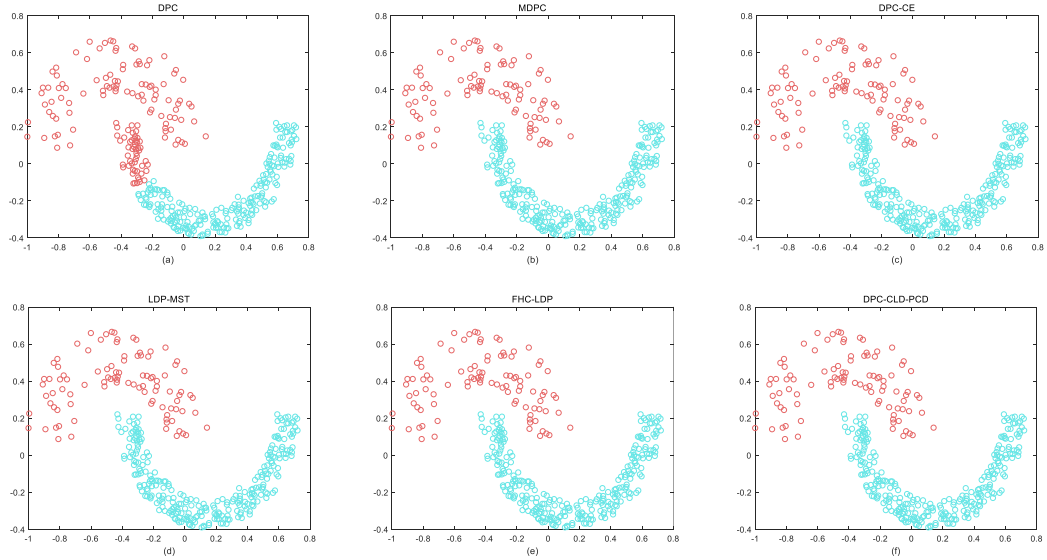


Fig. 4 Clustering results on Jain dataset

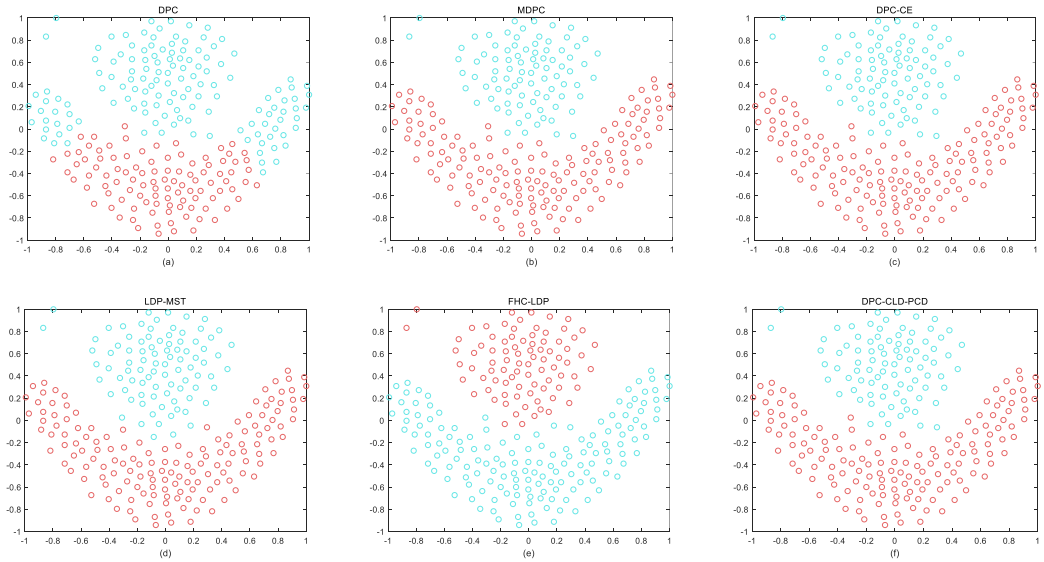


Fig. 5 Clustering results on Flame dataset

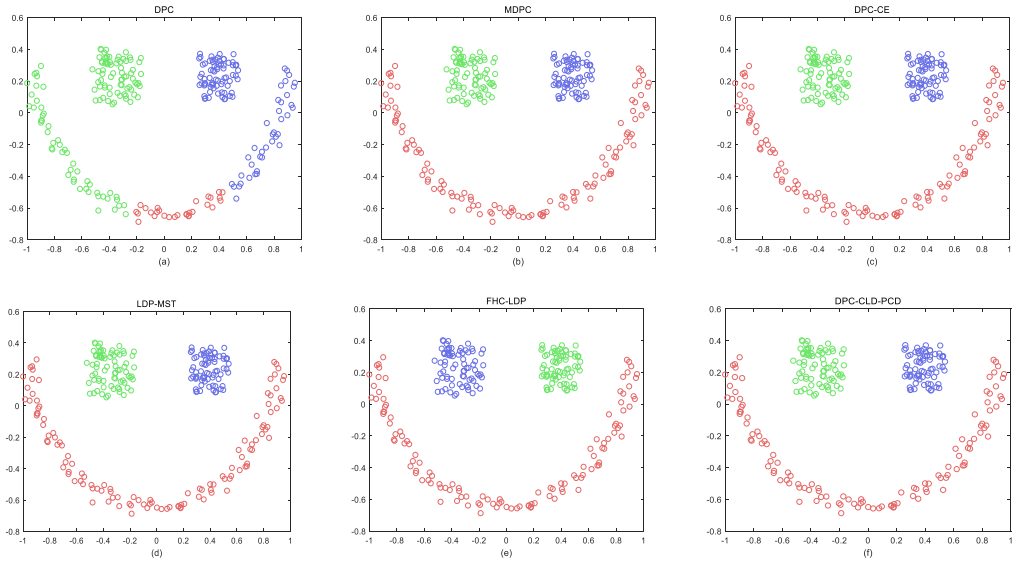


Fig. 6 Clustering results on Smile dataset

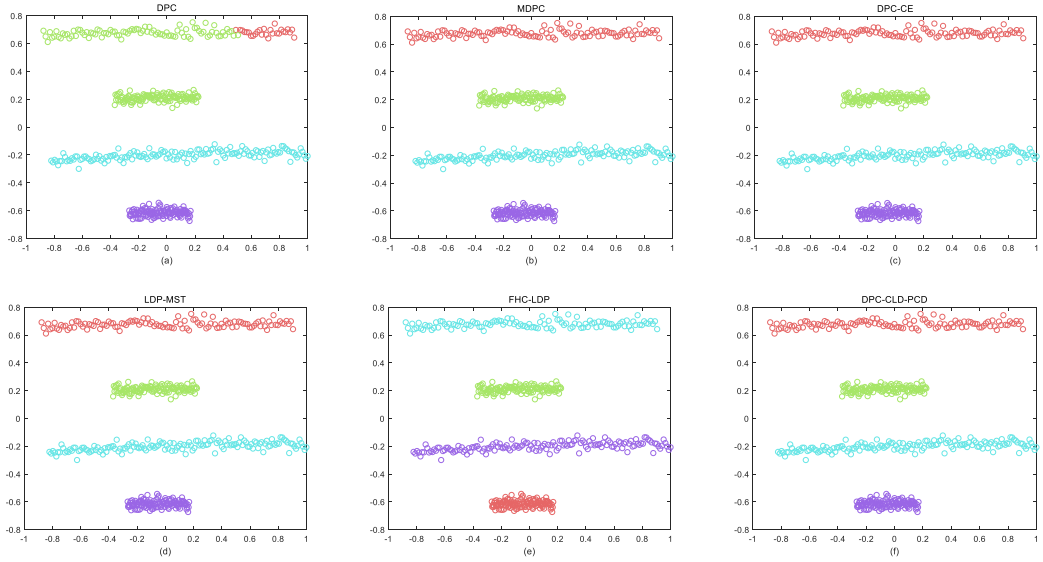


Fig. 7 Clustering results on Four-lines dataset

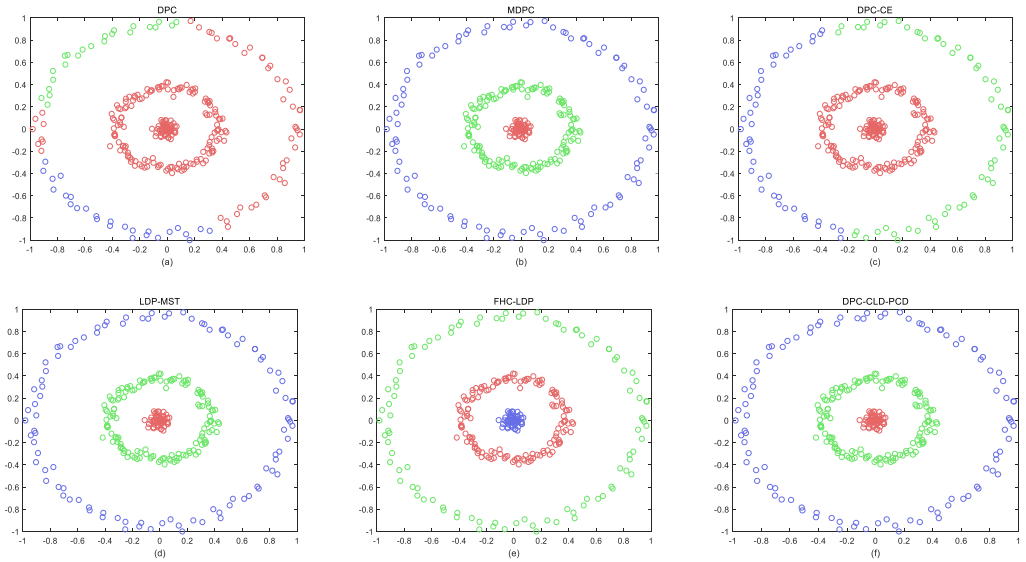


Fig. 8 Clustering results on Three-circles dataset

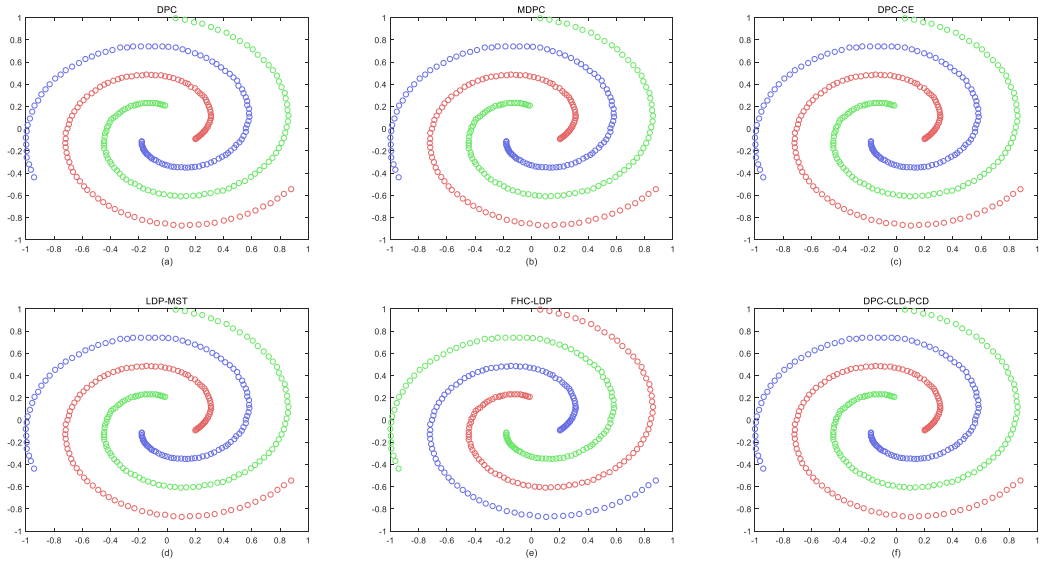


Fig. 9 Clustering results on Spiral dataset

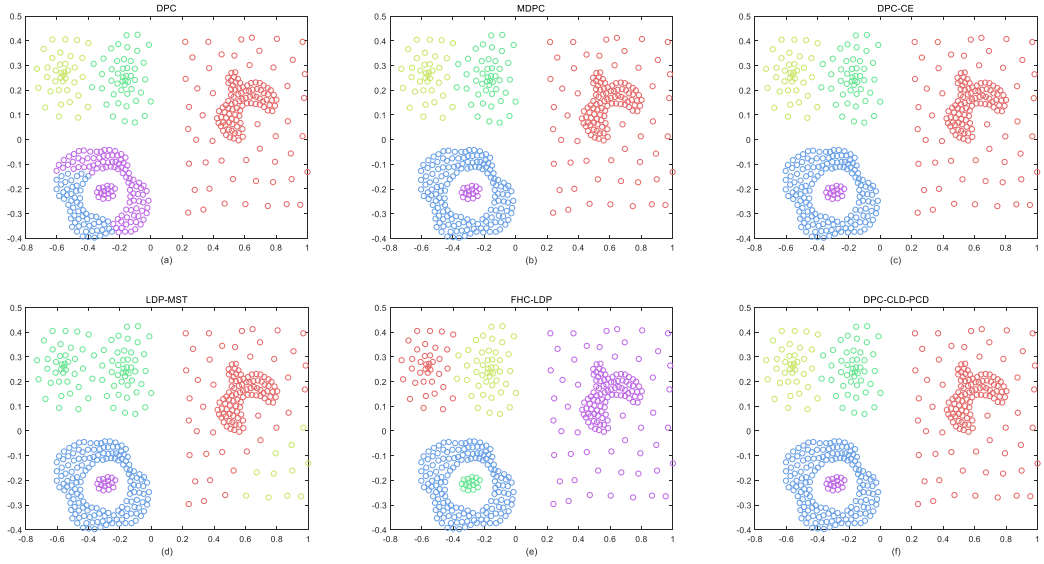


Fig. 10 Clustering results on Compound dataset

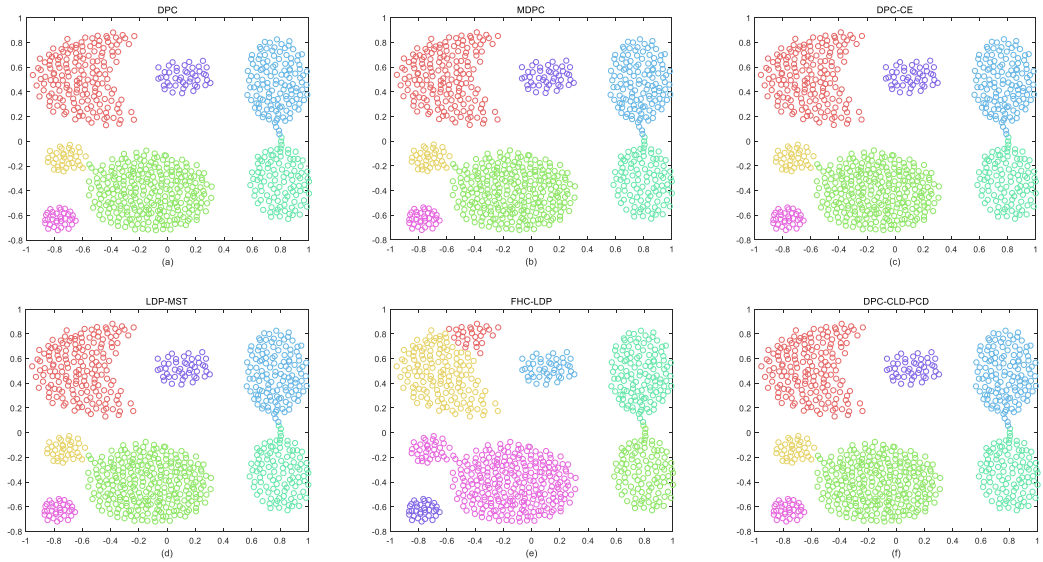


Fig. 11 Clustering results on Aggregation dataset

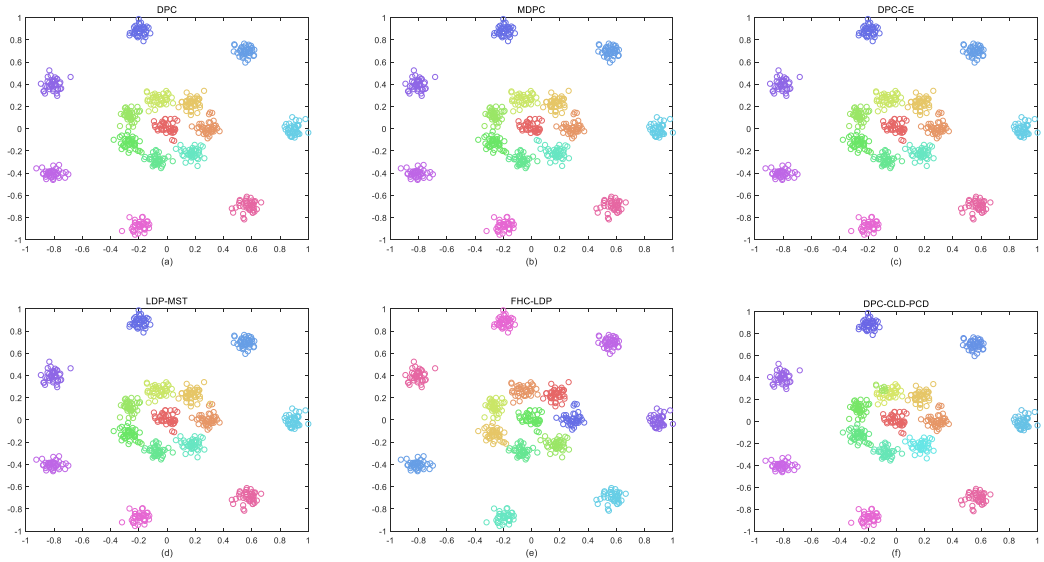


Fig. 12 Clustering results on R15 dataset

As we can see, Table 5-Table 7 demonstrate ACC, NMI, and RI values respectively of six comparative algorithms on nine two-dimensional synthetic datasets we discussed above. It can be seen that the original DPC works best only on Spiral dataset. For MDPC, except Aggregation and R15 datasets, the values of three evaluation indicators on the remaining seven datasets are the highest. In terms of DPC-CE, it gets the maximum values of three evaluation metrics on six datasets, including Jain, Flame, Smile, Spiral, Aggregation, and Four-lines. In addition, LDP-MST achieves the most satisfactory performance on Jain, Smile, Three-circles, Spiral and Four-lines datasets. In addition, FHC-LDP is good at processing six datasets, including Jain, Smile, Three-circles, Spiral, Compound, and Four-lines. Finally, the proposed DPC-CLD-PRD method obtains the biggest values of three estimation indicators, which means the best performance on all selected synthetic datasets.

4.3 Experiments on real-world datasets

In this section, a number of experiments are demonstrated on 8 real-world datasets from UCI repository to further illustrate the feasibility and efficiency of the proposed DPC-CLD-PRD method in this paper. The specific information of eight real-world datasets is provided in Table 2 in detail. Table 6-Table 8 display values of three evaluation indicators of the original DPC algorithm, four variants of DPC, and the proposed DPC-CLD-PRD method.

In terms of ACC and RI values displayed in Table 6 and Table 8 respectively, MDPC has the best

clustering performance on Ecoli dataset while DPC-CE gets the most satisfactory result on Cancer dataset. Besides, DPC-CLD-PRD shows the most effective results on six remaining datasets, including Vowel, Zoo, Blood, Wine, Seeds, and Glass. Besides, for NMI value shown in Table 7, DPC-CE achieves the most optimal clustering results on Vowel, Ecoli, and Cancer datasets while DPC-CLD-PRD gets the best performance on the rest Zoo, Blood, Wine, Seeds, and Glass datasets. Table 8 shows RI values of six algorithms on eight real-world datasets.

Table 3 ACC values of six algorithms on nine two-dimensional synthetic datasets

	DPC	MDPC	DPC-CE	LDP-MST	FHC-LDP	DPC-CLD-PRD
Jain	0.8606	**1**	**1**	**1**	**1**	**1**
Flame	0.7875	**1**	**1**	0.9833	0.9917	**1**
Smile	0.6654	**1**	**1**	**1**	**1**	**1**
Three-circles	0.6957	**1**	0.6589	**1**	**1**	**1**
Spiral	**1**	**1**	**1**	**1**	**1**	**1**
Compound	0.6767	**0.8722**	0.9971	0.8070	**0.8722**	**0.8722**
Aggregation	0.9901	0.8236	**0.9987**	0.9975	0.9201	**0.9987**
R15	0.9226	0.9023	0.9104	0.9021	0.9011	**0.9345**
Four-lines	0.8242	**1**	**1**	**1**	**1**	**1**

Table 4 NMI values of six algorithms on nine two-dimensional synthetic datasets

	DPC	MDPC	DPC-CE	LDP-MST	FHC-LDP	DPC-CLD-PRD
Jain	0.5068	**1**	**1**	**1**	**1**	**1**
Flame	0.4132	**1**	**1**	0.8752	0.9355	**1**
Smile	0.4768	**1**	**1**	**1**	**1**	**1**
Three-circles	0.6781	**1**	0.6695	**1**	**1**	**1**
Spiral	**1**	**1**	**1**	**1**	**1**	**1**
Compound	0.7920	**0.9151**	0.9912	0.8605	**0.9151**	**0.9151**
Aggregation	0.9916	0.7644	**0.9957**	0.9924	0.9237	**0.9957**
R15	0.9942	0.9893	0.9892	0.9762	0.9808	**0.9963**
Four-lines	0.8453	**1**	**1**	**1**	**1**	**1**

Table 5 RI values of six algorithms on nine two-dimensional synthetic datasets

	DPC	MDPC	DPC-CE	LDP-MST	FHC-LDP	DPC-CLD-PRD
Jain	0.7594	**1**	**1**	**1**	**1**	**1**
Flame	0.6639	**1**	**1**	0.9671	0.9834	**1**
Smile	0.6843	**1**	**1**	**1**	**1**	**1**
Three-circles	0.7632	**1**	0.7563	**1**	**1**	**1**
Spiral	**1**	**1**	**1**	**1**	**1**	**1**
Compound	0.8467	**0.9410**	0.9986	0.9279	**0.9410**	**0.9410**
Aggregation	0.9963	0.9257	**0.9993**	0.9985	0.9561	**0.9993**
R15	0.9927	0.9214	0.9354	0.9162	0.9150	**0.9968**
Four-lines	0.8968	**1**	**1**	**1**	**1**	**1**

Table 6 ACC values of six algorithms on eight real-world datasets

	DPC	MDPC	DPC-CE	LDP-MST	FHC-LDP	DPC-CLD-PRD
Vowel	0.3786	0.4225	0.4237	0.4328	0.2595	**0.4340**
Zoo	0.1584	0.1623	0.1683	0.2283	0.4158	**0.8119**
Blood	0.6912	0.6925	0.7674	0.6925	0.7634	**0.7847**
Ecoli	0.6458	**0.6905**	0.6458	0.5774	0.5298	0.4494
Wine	0.5281	0.6517	0.5281	0.5674	0.5393	**0.6904**
Seeds	0.6476	0.5619	0.6190	0.6333	0.8143	**0.8857**
Cancer	0.6691	0.8634	**0.8653**	0.6258	0.7452	0.6515
Glass	0.3458	0.3505	0.3411	0.3084	0.2897	**0.3551**

Table 7 NMI values of six algorithms on eight real-world datasets

	DPC	MDPC	DPC-CE	LDP-MST	FHC-LDP	DPC-CLD-PRD
Vowel	0.4035	0.2462	**0.4727**	0.2882	0.1415	0.4122
Zoo	0.7443	0.8012	0.8034	0.3221	0.2140	**0.9097**
Blood	0.0463	0.0001	0.0350	0.0001	0.0257	**0.1195**
Ecoli	0.5616	0.5527	**0.5626**	0.5133	0.4997	0.2304
Wine	0.3998	0.4010	0.3998	0.3041	0.4147	**0.4341**
Seeds	0.5866	0.4295	0.5482	0.5423	0.6252	**0.6983**
Cancer	0.0823	0.4325	**0.4679**	0.3576	0.1897	0.0258
Glass	0.0525	0.0170	0.0433	0.0486	0.0349	**0.0606**

Table 8 RI values of six algorithms on eight real-world datasets

	DPC	MDPC	DPC-CE	LDP-MST	FHC-LDP	DPC-CLD-PRD
Vowel	0.4989	0.5778	0.7874	0.6571	0.3763	**0.7876**
Zoo	0.8347	0.8531	0.8695	0.6277	0.3335	**0.8850**
Blood	0.5869	0.5736	0.6225	0.5736	0.6382	**0.6383**
Ecoli	0.7055	**0.7957**	0.7086	0.7880	0.7869	0.6332
Wine	0.6105	0.6440	0.6105	0.6016	0.6123	**0.6783**
Seeds	0.7286	0.5882	0.7133	0.7325	0.8035	**0.8673**
Cancer	0.5565	0.7434	**0.7665**	0.6367	0.6197	0.5462
Glass	0.4107	0.4769	0.4186	0.6038	0.6062	**0.6165**

4.4 Experiments on Olivetti Face dataset

In this section, we demonstrate the clustering results of the original DPC algorithm and the proposed DPC-CLD-PRD method in this paper on Olivetti Face dataset. The Olivetti Face image dataset contains more than 400 face images and about 100 face images are selected for experiments in this paper. Fig. 13 and Fig. 14 display the clustering performance of DPC and DPC-CLD-PRD respectively.

As we can recognize in the first and last images, DPC-CLD-PRD achieves a more satisfactory clustering performance than DPC. Note that Gaussian kernel is used for density estimation in DPC and the value of d_c is set to 2%. In addition, values of three evaluation indicators including ACC, NMI, and RI of DPC are 0.6300, 0.7801, and 0.4110 respectively. As we can see, DPC-CLD-PRD obtains larger values of three evaluation indicators, which are 0.6300, 0.8356, and 0.7000.

Fig. 13 The clustering result of DPC on Olivetti Face dataset

Fig. 13 The clustering result of DPC-CLD-PRD on Olivetti Face dataset

4.5 Analysis of Experimental Results

As we can see from the above experiments, DPC-CLD-PRD overcomes the poor performance of the original DPC on some specific datasets such as manifold datasets. As is shown in experiments on synthetic datasets in section 4.3, DPC-CLD-PRD can achieve satisfactory clustering performance on manifold datasets. Also, DPC-CLD-PRD obtains the highest values of three evaluation indicators on most real-world datasets as is demonstrated in experiments on real-world datasets in section 4.4. In addition, the proposed DPC-CLD-PRD algorithm performs better than the original DPC algorithm on Olivetti Face image dataset as shown in section 4.5. In conclusion, DPC-CLD-PRD has more successful clustering results than DPC and its four variants based on the experiments on synthetic datasets, real-world datasets from UCI repository, and Olivetti Face image dataset.

5 Conclusion

The original DPC algorithm uses the Euclidean distance as the similarity measure and ignores the spatial consistency of the data. In this paper, a novel density peaks clustering algorithm with connected local density and punished relative distance (DPC-CLD-PRD) is proposed to address the problem. A flexible connectivity distance is applied and combined with a special connectivity estimation strategy. In addition, a novel density estimation approach is designed for local density calculation. Experiments on both synthetic datasets and real-world datasets demonstrate the feasibility and effectiveness of the proposed method.

In the future, we are expecting to reduce the time complexity of the proposed algorithm and improve its performance in handling high-dimensional or multi-feature datasets. In addition, we are dedicated to enhancing the simplicity and robustness of the proposed algorithm by reducing the sensitivity of the parameters in this paper.

Acknowledgements This work was supported by National Natural Science Foundation of China (No.61806114, 61876101), and China Postdoctoral Science Foundation (No.2018M642695, 2019T120607).

Declarations

Ethical Approval Not applicable.

Competing interests There is no competing interests.

Authors' contributions Jingwen Xiong: Conceptualization, Methodology, Writing – original draft. Wenke Zang: Conceptualization Ideas, Supervision, Writing - review & editing. Yuzhen Zhao: Validation. Xiyu Liu: Reviewing, Funding acquisition.

Funding This work was supported by National Natural Science Foundation of China (No.61806114, 61876101) and China Postdoctoral Science Foundation (No.2018M642695, 2019T120607).

Availability of data and materials Data and materials will be made available on reasonable request.

References

1. K. G. Flores and S. E. Garza, "Density peaks clustering with gap-based automatic center detection," (in English), *Knowl-Based Syst,* vol. 206, Oct 28 2020, doi: ARTN 10635010.1016/j.knosys.2020.106350.
2. A. K. Pujari, K. Rajesh, and D. S. Reddy, "Clustering techniques in data mining - A survey," (in English), *Iete J Res,* vol. 47, no. 1-2, pp. 19-28, Jan-Apr 2001, doi: Doi 10.1080/03772063.2001.11416199.
3. E. Pastuchova and S. Vaclavikova, "Cluster Analysis - Data Mining Technique for Discovering

Natural Groupings in the Data," (in English), *J Electr Eng-Slovak,* vol. 64, no. 2, pp. 128-131, Mar-Apr 2013, doi: 10.2478/jee-2013-0019.

4. K. Gao, H. A. Khan, and W. W. Qu, "Clustering with Missing Features: A Density-Based Approach," (in English), *Symmetry-Basel,* vol. 14, no. 1, Jan 2022, doi: ARTN 6010.3390/sym14010060.
5. H. F. Liu, J. Li, Y. Wu, and Y. Fu, "Clustering With Outlier Removal," (in English), *Ieee T Knowl Data En,* vol. 33, no. 6, pp. 2369-2379, Jun 1 2021, doi: 10.1109/Tkde.2019.2954317.
6. X. Xu, S. F. Ding, Y. R. Wang, L. J. Wang, and W. K. Jia, "A fast density peaks clustering algorithm with sparse search," (in English), *Inform Sciences,* vol. 554, pp. 61-83, Apr 2021, doi: 10.1016/j.ins.2020.11.050.
7. D. D. Cheng, Q. S. Zhu, J. L. Huang, Q. W. Wu, and L. J. Yang, "Clustering with Local Density Peaks-Based Minimum Spanning Tree," (in English), *Ieee T Knowl Data En,* vol. 33, no. 2, pp. 374-387, Feb 1 2021, doi: 10.1109/Tkde.2019.2930056.
8. M. Raissi, P. Perdikaris, and G. E. Karniadakis, "Machine learning of linear differential equations using Gaussian processes," (in English), *J Comput Phys,* vol. 348, pp. 683-693, Nov 1 2017, doi: 10.1016/j.jcp.2017.07.050.
9. C. F. Tsai, W. C. Lin, Y. H. Hu, and G. T. Yao, "Under-sampling class imbalanced datasets by combining clustering analysis and instance selection," (in English), *Inform Sciences,* vol. 477, pp. 47-54, Mar 2019, doi: 10.1016/j.ins.2018.10.029.
10. A. Fahad *et al.*, "A Survey of Clustering Algorithms for Big Data: Taxonomy and Empirical Analysis," (in English), *Ieee T Emerg Top Com,* vol. 2, no. 3, pp. 267-279, Jul-Sep 2014, doi: 10.1109/Tetc.2014.2330519.
11. Q. H. Zhao, X. L. Li, Y. Li, and X. M. Zhao, "A fuzzy clustering image segmentation algorithm based on Hidden Markov Random Field models and Voronoi Tessellation," (in English), *Pattern Recogn Lett,* vol. 85, pp. 49-55, Jan 1 2017, doi: 10.1016/j.patrec.2016.11.019.
12. S. K. Choy, S. Y. Lam, K. W. Yu, W. Y. Lee, and K. T. Leung, "Fuzzy model-based clustering and its application in image segmentation," (in English), *Pattern Recogn,* vol. 68, pp. 141-157, Aug 2017, doi: 10.1016/j.patcog.2017.03.009.
13. J. Hou, W. X. Liu, X. E, and H. X. Cui, "Towards parameter-independent data clustering and image segmentation," (in English), *Pattern Recogn,* vol. 60, pp. 25-36, Dec 2016, doi: 10.1016/j.patcog.2016.04.015.
14. H. Wang *et al.*, "Pattern recognition and classification of two cancer cell lines by diffraction imaging at multiple pixel distances," (in English), *Pattern Recogn,* vol. 61, pp. 234-244, Jan 2017, doi: 10.1016/j.patcog.2016.07.035.
15. M. Nilashi, K. Bagherifard, M. Rahmani, and V. Rafe, "A recommender system for tourism industry using cluster ensemble and prediction machine learning techniques," (in English), *Comput Ind Eng,* vol. 109, pp. 357-368, Jul 2017, doi: 10.1016/j.cie.2017.05.016.
16. G. B. Guo, J. Zhang, and N. Yorke-Smith, "Leveraging multiviews of trust and similarity to enhance clustering-based recommender systems," (in English), *Knowl-Based Syst,* vol. 74, pp. 14-27, Jan 2015, doi: 10.1016/j.knosys.2014.10.016.
17. S. Zahra, M. A. Ghazanfar, A. Khalid, M. A. Azam, U. Naeem, and A. Prugel-Bennett, "Novel centroid selection approaches for KMeans-clustering based recommender systems," (in English), *Inform Sciences,* vol. 320, pp. 156-189, Nov 1 2015, doi: 10.1016/j.ins.2015.03.062.
18. M. Ester, H. P. Kriegel, J. Sander, and X. Xu, "A Density-Based Algorithm for Discovering

Clusters in Large Spatial Databases with Noise," *AAAI Press,* 1996.

19. A. Rodriguez and A. Laio, "Clustering by fast search and find of density peaks," (in English), *Science,* vol. 344, no. 6191, pp. 1492-1496, Jun 27 2014, doi: 10.1126/science.1242072.
20. J. Macqueen, "Some methods for classification and analysis of multivariate observations," *Proc. Symp. Math. Statist. and Probability, 5th,* vol. 1, 1967.
21. G. M. Mazzeo, E. Masciari, and C. Zaniolo, "A fast and accurate algorithm for unsupervised clustering around centroids," (in English), *Inform Sciences,* vol. 400, pp. 63-90, Aug 2017, doi: 10.1016/j.ins.2017.03.002.
22. T. Lei, X. H. Jia, Y. N. Zhang, L. F. He, H. Y. Meng, and A. K. Nandi, "Significantly Fast and Robust Fuzzy C-Means Clustering Algorithm Based on Morphological Reconstruction and Membership Filtering," (in English), *Ieee T Fuzzy Syst,* vol. 26, no. 5, pp. 3027-3041, Oct 2018, doi: 10.1109/Tfuzz.2018.2796074.
23. S. Johnson, "Hierarchical clustering schemes," *Psychometrika*.
24. Z. Tian, R. Ramakrishnan, and M. Livny, "BIRCH: An Efficient Data Clustering Method for Very Large," *acm sigmod record,* vol. 25, no. 2, pp. 103-114, 1996.
25. J. Zhao, J. J. Tang, T. H. Fan, C. M. Li, and L. Z. Xu, "Density peaks clustering based on circular partition and grid similarity," (in English), *Concurr Comp-Pract E,* vol. 32, no. 7, Apr 10 2020, doi: ARTN e556710.1002/cpe.5567.
26. S. H. Yue, J. S. Wang, T. Wu, and H. X. Wang, "A new separation measure for improving the effectiveness of validity indices," (in English), *Inform Sciences,* vol. 180, no. 5, pp. 748-764, Mar 1 2010, doi: 10.1016/j.ins.2009.11.005.
27. T. Chen, N. L. Zhang, T. F. Liu, K. M. Poon, and Y. Wang, "Model-based multidimensional clustering of categorical data," (in English), *Artif Intell,* vol. 176, no. 1, pp. 2246-2269, Jan 2012, doi: 10.1016/j.artint.2011.09.003.
28. M. S. Yang, S. J. Chang-Chien, and Y. Nataliani, "Unsupervised fuzzy model-based Gaussian clustering," (in English), *Inform Sciences,* vol. 481, pp. 1-23, May 2019, doi: 10.1016/j.ins.2018.12.059.
29. C. Selvi and E. Sivasankar, "A novel optimization algorithm for recommender system using modified fuzzy c-means clustering approach," (in English), *Soft Comput,* vol. 23, no. 6, pp. 1901-1916, Mar 2019, doi: 10.1007/s00500-017-2899-6.
30. Y. Peng, Q. Zhu, and B. Huang, "Spectral clustering with density sensitive similarity function," *Knowl-Based Syst,* vol. 24, no. 5, pp. p.621-628, 2011.
31. Y. Wang, Y. Jiang, Y. Wu, and Z. H. Zhou, "Spectral Clustering on Multiple Manifolds," (in English), *Ieee T Neural Networ,* vol. 22, no. 7, pp. 1149-1161, Jul 2011, doi: 10.1109/Tnn.2011.2147798.
32. X. M. Tao *et al.*, "Density peak clustering using global and local consistency adjustable manifold distance," (in English), *Inform Sciences,* vol. 577, pp. 769-804, Oct 2021, doi: 10.1016/j.ins.2021.08.036.
33. A. M. Ikotun, A. E. Ezugwu, L. Abualigah, B. Abuhaija, and J. Heming, "K-means clustering algorithms: A comprehensive review, variants analysis, and advances in the era of big data," (in English), *Inform Sciences,* vol. 622, pp. 178-210, Apr 2023, doi: 10.1016/j.ins.2022.11.139.
34. F. P. Nie, Z. H. Li, R. Wang, and X. L. Li, "An Effective and Efficient Algorithm for K-Means Clustering With New Formulation," (in English), *Ieee T Knowl Data En,* vol. 35, no. 4, pp. 3433-3443, Apr 1 2023, doi: 10.1109/Tkde.2022.3155450.

35. D. D. Cheng, J. L. Huang, S. L. Zhang, S. Y. Xia, G. Y. Wang, and J. Xie, "K-Means Clustering With Natural Density Peaks for Discovering Arbitrary-Shaped Clusters," (in English), *Ieee T Neur Net Lear,* Feb 28 2023, doi: 10.1109/Tnnls.2023.3248064.
36. F. Murtagh and P. Legendre, "Ward's Hierarchical Agglomerative Clustering Method: Which Algorithms Implement Ward's Criterion?," *Springer US,* no. 3, 2014.
37. F. Murtagh and P. Contreras, "Algorithms for hierarchical clustering: an overview," (in English), *Wires Data Min Knowl,* vol. 2, no. 1, pp. 86-97, Jan-Feb 2012, doi: 10.1002/widm.53.
38. P. K. Kimes, Y. F. Liu, D. N. Hayes, and J. S. Marron, "Statistical Significance for Hierarchical Clustering," (in English), *Biometrics,* vol. 73, no. 3, pp. 811-821, Sep 2017, doi: 10.1111/biom.12647.
39. G. Karypis, E. H. Han, and V. Kumar, "Chameleon: Hierarchical clustering using dynamic modeling," (in English), *Computer,* vol. 32, no. 8, pp. 68-+, Aug 1999, doi: Doi 10.1109/2.781637.
40. S. Guha, R. Rastogi, and K. Shim, "Cure: An efficient clustering algorithm for large databases," (in English), *Inform Syst,* vol. 26, no. 1, pp. 35-58, Mar 2001, doi: Doi 10.1016/S0306-4379(01)00008-4.
41. M. J. Du and F. Y. Wu, "Grid-Based Clustering Using Boundary Detection," (in English), *Entropy-Switz,* vol. 24, no. 11, Nov 2022, doi: ARTN 160610.3390/e24111606.
42. A. Starczewski, M. M. Scherer, W. Ksiazek, M. Debski, and L. P. Wang, "A Novel Grid-Based Clustering Algorithm," (in English), *J Artif Intell Soft,* vol. 11, no. 4, pp. 319-330, Oct 2021, doi: 10.2478/jaiscr-2021-0019.
43. W. Wang, J. Yang, and R. Muntz, "STING: A statistical information grid approach to spatial data mining," in *VLDB'97, Proceedings of 23rd International Conference on Very Large Data Bases, August 25-29, 1997, Athens, Greece*, 1997.
44. M. Tareq, E. A. Sundararajan, A. Harwood, and A. Abu Bakar, "A Systematic Review of Density Grid-Based Clustering for Data Streams," (in English), *Ieee Access,* vol. 10, pp. 579-596, 2022, doi: 10.1109/Access.2021.3134704.
45. C. Bouveyron and C. Brunet-Saumard, "Model-based clustering of high-dimensional data: A review," (in English), *Comput Stat Data An,* vol. 71, pp. 52-78, Mar 2014, doi: 10.1016/j.csda.2012.12.008.
46. Z. Ghahramani and G. E. Hinton, "The EM Algorithm for Mixtures of Factor Analyzers," 1997.
47. H. Asheri, R. Hosseini, and B. N. Araabi, "A new EM algorithm for flexibly tied GMMs with large number of components," (in English), *Pattern Recogn,* vol. 114, Jun 2021, doi: ARTN 10783610.1016/j.patcog.2021.107836.
48. J. Zhao, G. Wang, J. S. Pan, T. H. Fan, and I. V. Lee, "Density peaks clustering algorithm based on fuzzy and weighted shared neighbor for uneven density datasets," (in English), *Pattern Recogn,* vol. 139, Jul 2023, doi: ARTN 10940610.1016/j.patcog.2023.109406.
49. A. Y. Ng, M. I. Jordan, and Y. Weiss, "On Spectral Clustering: Analysis and an algorithm," *proc nips,* 2002.
50. P. Zhang *et al.*, "Consensus One-Step Multi-View Subspace Clustering," (in English), *Ieee T Knowl Data En,* vol. 34, no. 10, pp. 4676-4689, Oct 1 2022, doi: 10.1109/Tkde.2020.3045770.
51. X. Zhu, S. Zhang, W. He, R. Hu, C. Lei, and P. Zhu, "One-Step Multi-View Spectral Clustering," *IEEE Transactions on Knowledge & Data Engineering,* vol. 31, no. 10, pp. 2022-2034, 2019.
52. H. Yin, W. Hu, F. Li, and J. Lou, "One-step multi-view spectral clustering by learning common and specific nonnegative embeddings," *Int J Mach Learn Cyb,* no. 6, 2021.

53. Q. H. Zheng, J. H. Zhu, Z. Y. Li, S. M. Pang, J. Wang, and Y. C. Li, "Feature concatenation multi-view subspace clustering," (in English), *Neurocomputing,* vol. 379, pp. 89-102, Feb 28 2020, doi: 10.1016/j.neucom.2019.10.074.
54. J. Schneider and M. Vlachos, "Scalable density-based clustering with quality guarantees using random projections," (in English), *Data Min Knowl Disc,* vol. 31, no. 4, pp. 972-1005, Jul 2017, doi: 10.1007/s10618-017-0498-x.
55. M. Ester, "Density-based Clustering," *Springer US,* 2009.
56. Q. H. Zhang, Y. Y. Dai, and G. Y. Wang, "Density peaks clustering based on balance density and connectivity," (in English), *Pattern Recogn,* vol. 134, Feb 2023, doi: ARTN 10905210.1016/j.patcog.2022.109052.
57. S. F. Ding *et al.*, "A Sampling-Based Density Peaks Clustering Algorithm for Large-Scale Data," (in English), *Pattern Recogn,* vol. 136, Apr 2023, doi: ARTN 10923810.1016/j.patcog.2022.109238.
58. Z. Rasool, S. Aryal, M. R. Bouadjenek, and R. Dazeley, "Overcoming weaknesses of density peak clustering using a data-dependent similarity measure," (in English), *Pattern Recogn,* vol. 137, May 2023, doi: ARTN 10928710.1016/j.patcog.2022.109287.
59. S. F. Ding, W. Du, X. Xu, T. H. Shi, Y. R. Wang, and C. Li, "An improved density peaks clustering algorithm based on natural neighbor with a merging strategy," (in English), *Inform Sciences,* vol. 624, pp. 252-276, May 2023, doi: 10.1016/j.ins.2022.12.078.
60. J. L. Lin, J. C. Kuo, and H. W. Chuang, "Improving Density Peak Clustering by Automatic Peak Selection and Single Linkage Clustering," (in English), *Symmetry-Basel,* vol. 12, no. 7, Jul 2020, doi: ARTN 116810.3390/sym12071168.
61. J. Y. Guan, S. Li, X. X. He, and J. J. Chen, "Clustering by fast detection of main density peaks within a peak digraph," (in English), *Inform Sciences,* vol. 628, pp. 504-521, May 2023, doi: 10.1016/j.ins.2023.01.144.
62. Y. Li, L. Y. Sun, and Y. C. Tang, "DPC-FSC: An approach of fuzzy semantic cells to density peaks clustering," (in English), *Inform Sciences,* vol. 616, pp. 88-107, Nov 2022, doi: 10.1016/j.ins.2022.10.041.
63. W. N. Tong, S. Liu, and X. Z. Gao, "A density-peak-based clustering algorithm of automatically determining the number of clusters," (in English), *Neurocomputing,* vol. 458, pp. 655-666, Oct 11 2021, doi: 10.1016/j.neucom.2020.03.125.
64. W. J. Guo, W. H. Wang, S. P. Zhao, Y. L. Niu, Z. Y. Zhang, and X. G. Liu, "Density Peak Clustering with connectivity estimation," (in English), *Knowl-Based Syst,* vol. 243, May 11 2022, doi: ARTN 10850110.1016/j.knosys.2022.108501.
65. J. Y. Guan, S. Li, X. X. He, J. H. Zhu, and J. J. Chen, "Fast hierarchical clustering of local density peaks via an association degree transfer method," (in English), *Neurocomputing,* vol. 455, pp. 401-418, Sep 30 2021, doi: 10.1016/j.neucom.2021.05.071.

Authors and Affiliations

Jingwen Xiong [1] · Wenke Zang [1] * · Yuzhen Zhao [1] · Xiyu Liu [1]

[1] School of Business, Shandong Normal University, Jinan, Shandong 250014, China

Corresponding author: Wenke Zang, School of Business, Shandong Normal University, Jinan, Shandong 250014, China.

Tel.: +86-531-86180509, **Fax**: +86-531-86180509, **E-mail:** wink@sdnu.edu.cn