Knowl Inf Syst (2019) 59:285–309
https://doi.org/10.1007/s10115-018-1189-7

REGULAR PAPER

A novel density peaks clustering with sensitivity of local density and density-adaptive metric

Mingjing Du[1] · **Shifei Ding**[1,2] · **Yu Xue**[3] · **Zhongzhi Shi**[2]

Received: 11 July 2016 / Accepted: 8 April 2018 / Published online: 16 April 2018

Abstract The density peaks (DP) clustering approach is a novel density-based clustering algorithm. On the basis of the prior assumption of consistency for semi-supervised learning problems, we further make the assumptions of consistency for density-based clustering. The first one is the assumption of the local consistency, which means nearby points are likely to have the similar local density; the second one is the assumption of the global consistency, which means points on the same high-density area (or the same structure, i.e., the same cluster) are likely to have the same label. According to the first assumption, we provide a new option based on the sensitivity of the local density for the local density. In addition, we redefine δ and redesign the assignation strategy based on a new density-adaptive metric according to the second assumption. We compare the performance of our algorithm with traditional clustering schemes, including DP, K-means, fuzzy C-means, Gaussian mixture model, and self-organizing maps. Experiments on different benchmark data sets demonstrate the effectiveness of the proposed algorithm.

Keywords Clustering analysis · Density peaks clustering · Sensitivity of local density · Density-adaptive distance

1 Introduction

Clustering perhaps is the most important and widely used method of unsupervised learning. It can group together similar objects according to some similarity metrics [15]. As a result,

✉ Shifei Ding
dingsf@cumt.edu.cn

1 School of Computer Science and Technology, China University of Mining and Technology, Xuzhou 221116, China

2 Key Laboratory of Intelligent Information Processing, Institute of Computing Technology, Chinese Academy of Sciences, Beijing 100090, China

3 School of Computer and Software, Nanjing University of Information Science and Technology, Nanjing 210044, China

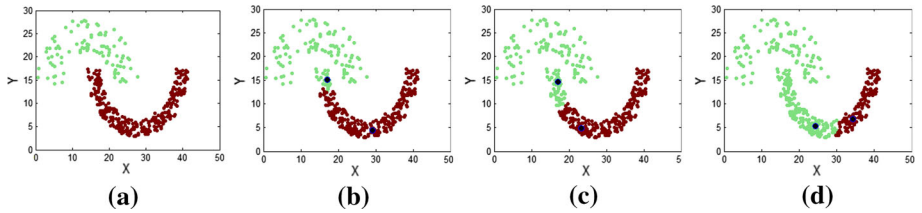


Fig. 1 DP on the Jain data set. **a** Ground truth, **b** $d_c = 0.5\%$, **c** $d_c = 2\%$, and **d** $d_c = 6\%$

similar objects are grouped together, while dissimilar objects belong to different clusters. Clustering has many applications in various domains, including biology, economics, and medicine. Its applications include data mining, document retrieval, image segmentation, and pattern classification.

A new algorithm, density peaks (DP) clustering [25], is proposed by Rodriguez and Laio. Unlike traditional density-based clustering methods, such as DBSCAN [8] and OPTICS [1], this algorithm starts by determining the center points of clusters. If a data point is found that its nearest point (higher density than this data point) belongs to a certain cluster, then allocates the data point to that cluster. This algorithm is able to detect non-spherical clusters without specifying the number of clusters. Moreover, the DP clustering does not need to iterate. Currently, density peaks clustering algorithm is used in outlier detection [3], image processing [5, 18], and document processing [27, 35]. For example, inspired by this, Ma et al. [19] propose the LED algorithm, which is based on Structural Clustering, which converts structural similarity between vertices to weights of network. However, there are still some shortcomings. For example, the DP clustering only has taken the global structure of data into account, which leads to missing many clusters. In order to overcome this problem, Du et al. [7] propose a density peaks clustering based on k nearest neighbors (DPC-KNN) which introduces the idea of k nearest neighbors (KNN) into the DP clustering and has another option for the local density computation. The DP clustering performs not well when the DP clustering finds some pseudocluster centers. In order to overcome this difficulty, Zhang and Li [34] propose an extension of CFSFDP (E_CFSFDP). This algorithm introduces the idea of an agglomerative hierarchical clustering algorithm, CHAMELEON, into the original DP clustering. However, the DP clustering algorithm cannot find the correct number of clusters automatically. In order to overcome a similar problem, Liang and Chen [17] propose the 3DC clustering based on the divide-and-conquer strategy and the density-reachable concept. And then they integrate this scheme with the idea of divisive hierarchical clustering algorithms. To improve the running speed of DP algorithm, Xu et al. [30] propose a novel approach based on grid, called density peaks clustering algorithm based on grid (DPCG).

The DP clustering algorithm performs excellent clustering results on some data sets, but there still exist some other situations when it obtains very poor results. Specifically, Fig. 1 shows that the DP clustering does not perform well on the Jain data set [13]. Data points have two clusters of different densities. Each black point represents a cluster center obtained by DP, as shown in Fig. 1. Obviously, two center points obtained by DP are in the same cluster which has higher density compared with the other cluster. When clusters have different densities on the data set, the main reason for a poor result obtained by DP obtains is that the definition of the local density and the choice of the cluster centers are both based on the assumption that all data points in different clusters have the similar density. In the real world, however, there exist many situations that the real-world data have clusters of different densities.

In addition, the assignation strategy of the original DP clustering algorithm is applicable only to some data sets with simple structure. The original DP method is not able to find

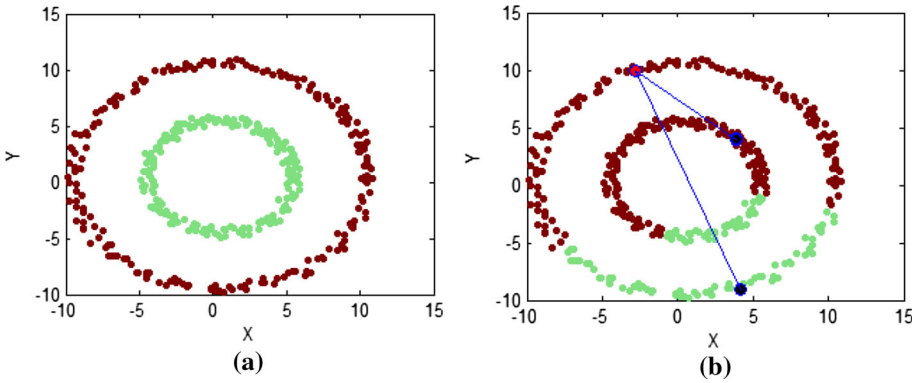


Fig. 2 Clustering results of the two-circles data set. **a** Ground truth and **b** mis-assignation

clusters of twisted or curved structure. For example, the two clusters in the Two-circles data set cannot be found correctly. As shown in Fig. 2b, the black points represent the cluster centers obtained by the DP method, and the red point has the third-highest local density. It can be easily seen that the red point should belong to the outer circle, but this data point is found that its nearest point, with higher local density compared with itself, belongs to the inner circle rather than the outer circle. Obviously, a poor clustering result is mainly due to its assignation strategy. In addition, another hidden reason is the computation of δ.

In order to overcome the first problem that DP cannot generate the optimal structure of clusters on some data sets that have clusters of different densities, we define the sensitivity of the local density and then propose another option for the local density based on the concept of the sensitivity of the local density. In addition, we introduce the density-adaptive metric into the computation of δ and the assignation strategy.

The rest of this paper is organized as follows. In Sect. 2, we describe the principle of the DP clustering method. In Sect. 3, we make a detailed description of DP-DA. In Sect. 4, we present experimental results on synthetic data sets and real-world data sets, and then we analyze the performance of the proposed algorithm. Finally, some conclusions and the intending work are given in the last section.

2 Density peaks clustering

2.1 Notations

Let $\mathbf{X} = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_N\}$ denote a data set of N data objects, where for each i, $1 \leq i \leq N$, $\mathbf{x}_i$ with M attributes. Therefore, for each i, $1 \leq i \leq N$, and for j, $1 \leq j \leq M$ let $x_{i,j}$ be the jth attribute of $\mathbf{x}_i$. Let $d\left(\mathbf{x}_i, \mathbf{x}_j\right)$ denote the Euclidean distance between the object $\mathbf{x}_i$ and the object $\mathbf{x}_j$. The mathematical expression is as follows:

$$d\left(\mathbf{x}_i, \mathbf{x}_j\right) = \left\|\mathbf{x}_i - \mathbf{x}_j\right\|_2. \tag{1}$$

2.2 The original algorithm

The DP clustering proposed by Rodriguez and Laio is a new density-based clustering algorithm. Unlike DBSCAN, the DP clustering finds the cluster centers before data points are

assigned. Determining the cluster centers is of vital importance to guarantee good clustering results. Because it determines the number of the clusters and affects the assignation indirectly, obviously, the key to this algorithm is how to determine the cluster centers. Rodriguez and Laio propose the decision graph of the ρ and δ axis. ρ_i denotes the local density ρ_i of each point $\mathbf{x}_i$, and δ_i denotes its distance from points with higher density. The authors suggest that the cluster centers have two important properties. One is that the cluster centers are surrounded by neighbors with a lower local density. The other is that they have relatively larger distance to the points with higher density. In the following, we will describe the calculation of ρ_i and δ_i in much more detail.

DP represents data objects as points in a space and adopts a distance metric, such as formula (1), as a dissimilarity between objects. In the DP method, d_c is the only input parameter, which is a cutoff distance.

The local density ρ_i of a point $\mathbf{x}_i$ is defined as

$$\rho_i = \sum_j \chi\left(d\left(\mathbf{x}_i, \mathbf{x}_j\right) - d_c\right)$$
$$\chi(x) = \begin{cases} 1, & x < 0 \\ 0, & x \geq 0 \end{cases}, \tag{2}$$

ρ_i is simply defined as the number of points that are closer than d_c to point $\mathbf{x}_i$. Besides, there is an alternative definition of the local density in the code presented by Rodriguez and Laio. If the former is called a hard threshold, the latter will be called a soft threshold. Authors adopt a Gaussian kernel function to estimate the local density ρ_i, as follows:

$$\rho_i = \sum_j \exp\left(-\frac{d\left(\mathbf{x}_i, \mathbf{x}_j\right)^2}{d_c^2}\right), \tag{3}$$

If formula (2) is the definition of the hard threshold, formula (3) is the definition of the soft threshold.

The calculation of the delta value, again, is quite simple. The minimum distance between the point of $\mathbf{x}_i$ and any other points with higher density, denoted by δ_i, is defined as

$$\delta_i = \begin{cases} \min\limits_{j:\rho_i<\rho_j} \left(d\left(\mathbf{x}_i, \mathbf{x}_j\right)\right), & \text{if } \exists\, j \text{ s.t. } \rho_i < \rho_j \\ \max\limits_j \left(d\left(\mathbf{x}_i, \mathbf{x}_j\right)\right), & \text{otherwise} \end{cases}. \tag{4}$$

When the local density and the delta value for each point have been calculated, this method identifies the cluster centers by searching anomalously large parameters ρ_i and δ_i. On the basis of this idea, cluster centers always appear on the upper-right corner of the decision graph. After cluster centers have been found, the DP clustering assigns remaining points to the same cluster as its nearest neighbors with higher density.

3 The proposed algorithm

On the basis of the prior assumption of consistency for semi-supervised learning problems [37], we make the assumptions of consistency for density-based clustering. The first one is the assumption of the local consistency, which means nearby points are likely to have the similar local density; the second one is the assumption of the global consistency, which means points on the same high-density area (or the same structure, i.e., the same cluster) are

likely to have the same label. We utilize some prior assumptions to improve the clustering performance of the original DP clustering algorithm.

3.1 New local density

When different clusters have different densities, the DP clustering does a poor job of finding the clusters. This situation is particularly acute in the case that the clusters have irregularly distributed data. Due to the original local density computation, the DP clustering can find more than one density peak for one cluster that has higher density than the others. In order to overcome this problem, we redefine the local density using an idea of the sensitivity of the local density.

We introduce a new local density computation based on the sensitivity of the local density in further detail as below.

Firstly, ρ_i' of point $\mathbf{x}_i$ is defined as:

$$\rho_i' = \exp\left(-\left(\frac{1}{k}\sum_{\mathbf{x}_j \in \mathrm{kNN}(\mathbf{x}_i)} d\left(\mathbf{x_i}, \mathbf{x_j}\right)^2\right)\right), \tag{5}$$

where $\mathrm{kNN}(\mathbf{x}_i)$ denotes k nearest points to $\mathbf{x}_i$ according to formula (1). The most important variable in formula (5), k is $\lceil p \cdot N \rceil$, where p is a percentage, and N denotes the number of data points in the data sets. $\lceil \cdot \rceil$ is a ceiling function. This means that p is one of the most important parameters of the proposed algorithm.

We define the concept of the sensitivity of the local density, $\mathrm{LS}_{i,j}$, that indicates that the absolute value of the difference between ρ_i' of point $\mathbf{x}_i$ and ρ_j' of point $\mathbf{x}_j$ divided by ρ_i', where $\mathbf{x}_j \in \mathrm{kNN}(\mathbf{x}_i)$. Let ε denotes the threshold for the sensitivity of the local density. When the sensitivity of the local density $\mathrm{LS}_{i,j}$ of $\mathbf{x}_i$ and $\mathbf{x}_j$ ($\mathbf{x}_j \in \mathrm{kNN}(\mathbf{x}_i)$) is less than the threshold, we refer to point $\mathbf{x}_j$ as one of the "real" k nearest neighbors to point $\mathbf{x}_i$. More formally, $\Delta_{i,j}$ can be expressed as:

$$\begin{aligned} \Delta_{i,j} = \mathrm{LS}_{i,j} - \varepsilon = \left|\rho_i' - \rho_j'\right| / \rho_i' - \varepsilon, \\ \text{where } \mathbf{x}_j \in \mathrm{kNN}(\mathbf{x}_i). \end{aligned} \tag{6}$$

The "real" k nearest neighbors of $\mathbf{x}_i$, denoted by $\widehat{\mathrm{kNN}}(\mathbf{x}_i)$, is defined as

$$\widehat{\mathrm{kNN}}(\mathbf{x}_i) = \left\{\mathbf{x}_j \in \mathrm{kNN}(\mathbf{x}_i) \,|\, \Delta_{i,j} \leq 0\right\}. \tag{7}$$

The local density of a point $\mathbf{x}_i$, denoted by ρ_i, is defined as

$$\begin{aligned} \rho_i &= \sum_{\mathbf{x}_j \in \widehat{\mathrm{kNN}}(\mathbf{x}_i)} \psi\left(\Delta_{i,j}\right) \\ \psi(x) &= \begin{cases} 1, & x \leq 0 \\ 0, & x > 0 \end{cases}. \end{aligned} \tag{8}$$

3.2 New density-adaptive metric

The selection of the cluster centers depends not only on the local density ρ_i of each point $\mathbf{x}_i$ but also on its distance δ_i from points with higher density. However, the original calculation of δ uses the Euclidean distance which only reflects the local structure of data. As shown in Fig. 3, $d(a, b) > d(a, c)$. However, it is obvious that points a and b belong to the same cluster, and points a and c belong to different clusters. By the definition of the clustering,

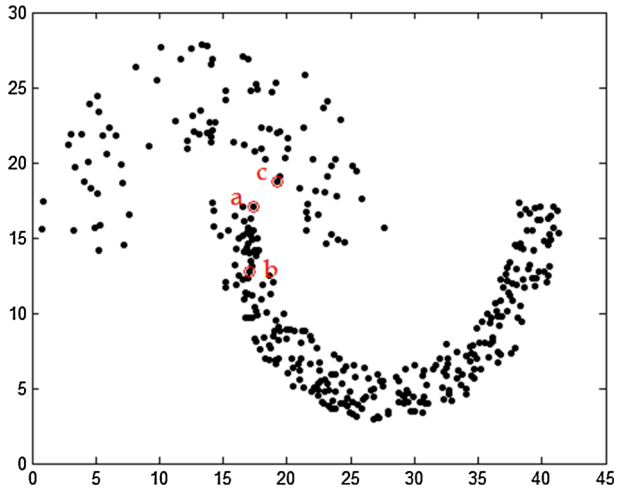


Fig. 3 Jain data set

points a and b are more similar than points a and c. To be specific, in terms of the DP clustering algorithm, the traditional distance metrics, such as the Euclidean metric, share some similar disadvantages. Assuming that point a has the highest local density, point b and point c tie for the second highest local density. According to the Euclidean metric, $\delta_c < \delta_b$, so that points a and b are selected as the cluster centers. But in fact, they belong to the same class. It follows that the traditional metrics that only take the local structure of data into account may lead to producing incorrect cluster centers. In addition, they may also lead to a misclassification. Assuming that points b and c are the center points of the two classes, and ρ_a is only slightly less than ρ_b and ρ_c. By the assignation strategy of the original algorithm, point a is assigned to the cluster that point c belongs to. In [28], in order to calculate the similarity matrix of Spectral Clustering (SC), Wang et al. first propose the concept of the density-adaptive distance. In order to overcome these difficulties, we propose another option for the computation of δ and a new assignation using a new density-adaptive metric redesigned by us.

Intuitively, the data of within-cluster tend to be distributed in a relatively high-density area, while the density of between-cluster is (externally) lower. Specifically, we want that the distance between points a and c is longer than the distance between points a and b. In other words, the path length in a low-density area is magnified, or the path length in a high-density area is shortened, or both. In fact, we expect to be able to design a distance metric: If a path connecting two points exists in a high-density area, this path will be given a relatively short distance; or if a path connecting two points passes through a low-density area, this path will be given an externally long distance [14, 32].

Similar to the path-based distance, the proposed density-adaptive distance reflects the idea that no matter how far the physical distance between two points is, they can be considered as in one cluster if they are connected by a set of successive points in dense regions.

Firstly, the data points, $\mathbf{X} = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_N\}$, can be regarded as the vertices in an undirected weighted graph, $G = (X, E)$, where E denotes an edge set. To be consistent with the aforementioned notations, we still use $\mathbf{x}_i$ to represent a vertex on the graph G in the following formulas. Then, the density-adjustable line segment length is defined as:

$$L\left(\mathbf{x}_i, \mathbf{x}_j\right) = \exp\left(\alpha \cdot d\left(\mathbf{x}_i, \mathbf{x}_j\right)\right), \tag{9}$$

where $\alpha > 0$ is called the scalability factor. The length of density-adjustable line segment can be scaled by adjusting the scalability factor.

Based on the length of the density-adjustable line segment, we will introduce the definition of the density-adaptive metric. P_{ij} denotes the set of all paths connecting vertices $\mathbf{x}_{p_i}$ and $\mathbf{x}_{p_j}$ on graph G, and τ is one of all paths P_{ij}. $|\tau|$ is the length of the path τ. More formally, P_{ij} can be expressed as:

$$\tau = \left\{\mathbf{x}_{p_i}, \ldots, \mathbf{x}_{p_{k-1}}, \mathbf{x}_{p_k}, \mathbf{x}_{p_{k+1}}, \ldots, \mathbf{x}_{p_j}\right\} \in P_{ij}, \tag{10}$$

where $\left(\mathbf{x}_{p_k}, \mathbf{x}_{p_{k+1}}\right) \in E\ (1 \leq k \leq |\tau| - 1)$.

Finally, a new density-adaptive distance is defined as:

$$d_{\mathrm{ds}}\left(\mathbf{x}_i, \mathbf{x}_j\right) = \min_{\tau \in P_{ij}} \sum_{k=1}^{k=|\tau|-1} L\left(\mathbf{x}_{p_k}, \mathbf{x}_{p_{k+1}}\right). \tag{11}$$

Note that $L\left(\mathbf{x}_{p_k}, \mathbf{x}_{p_{k+1}}\right)$ is the density-adjustable line segment length of two adjacent points along path $\tau \in P_{ij}$. In order to calculate density-adaptive distance, Floyd's algorithm is used. Firstly, k nearest points for each point are calculated according to formula (9). Then, we construct the k nearest-neighbors graph by linking all neighbor points. And each edge is weighted with the density-adjustable line segment length between the corresponding linked points. The density-adaptive distance between two points is calculated by the sum of the density-adjustable line segment lengths along the shortest path linking both points. The shortest path is computed by Floyd's algorithm.

Similar to the Euclidean distance metric, the proposed density-adaptive distance metric is symmetric, non-negative, and reflexivity. Moreover, the proposed distance has the following properties:

(1) $\forall \mathbf{x}_i, \mathbf{x}_j, \mathbf{x}_k \in \mathbf{X}$, we have $d_{\mathrm{ds}}\left(\mathbf{x}_i, \mathbf{x}_j\right) \leq d_{\mathrm{ds}}\left(\mathbf{x}_i, \mathbf{x}_k\right) + d_{\mathrm{ds}}\left(\mathbf{x}_j, \mathbf{x}_k\right)$, i.e., the proposed distance metric is triangle inequality.
(2) $\mathbf{x}_i, \mathbf{x}_j$ are in the same dense region while $\mathbf{x}_l, \mathbf{x}_k$ are in different dense regions, then with very high probability we have $d_{\mathrm{ds}}\left(\mathbf{x}_i, \mathbf{x}_j\right) < d_{\mathrm{ds}}\left(\mathbf{x}_l, \mathbf{x}_k\right)$.

Use the previous example in Fig. 3 again. The points a and b exist in the same high-density area, but the points a and c are separated by a low-density area. In Fig. 4, the values in the bracket indicate the Euclidean distance, and those outside the bracket indicates the density-adaptive distance. The Euclidean distance between a and b is 4.31, and the Euclidean distance between a and c is 2.52. The former is about 1.7 times as long as the latter. When $\alpha = 1$, the density-adaptive distance between a and b is 11.99, and the density-adaptive distance between a and c is 12.38. It is obvious that the density-adaptive distance between a and b is shorter than that between a and c. Supposing $\left\{a, \mathbf{x}_{p_2}, \mathbf{x}_{p_3}, \ldots, \mathbf{x}_{p_k}, b\right\}$ is the shortest path linking a and b, we have $d_{\mathrm{ds}}(a, b) = L\left(a, \mathbf{x}_{p_2}\right) + L\left(\mathbf{x}_{p_2}, \mathbf{x}_{p_3}\right) + \cdots + L\left(\mathbf{x}_{p_k}, b\right)$ in terms of the proposed distance metric. Note that $d_{\mathrm{ds}}(a, b) = 11.9913 < L(a, b) = \exp(4.3073) = 74.2398$. That is to say, the density-adaptive distance between points a and b is shorter than the density-adjustable line segment length between them. Nevertheless, $d_{\mathrm{ds}}(a, c) = 12.3845 \approx L(a, c) = \exp(2.5164) = 12.3839$ (Since approximate value is used in the exp() operation, $d_{\mathrm{ds}}(a, c)$ is approximately equal to $L(a, c)$). As you can see from the above example, our proposed distance metric has an effect of squeezing the distance in high-density regions, particularly in the complex manifolds. The proposed distance metric relies primarily on the short path. And the short path depends on data distribution and is adjustable with the data density. Thus, the proposed distance metric is density adaptive.

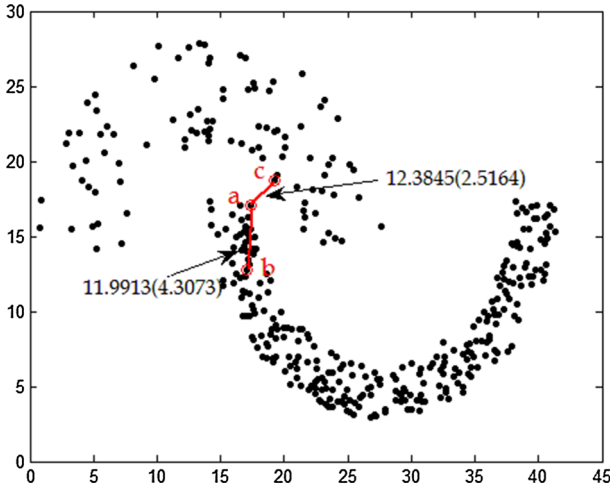


Fig. 4 New density-adaptive distance and Euclidean distance

δ_i is also redefined, as follows:

$$\delta_i = \begin{cases} \min\limits_{j:\rho_i<\rho_j} \left(d_{\mathrm{ds}}\left(\mathbf{x}_i, \mathbf{x}_j\right)\right), \text{ if } \exists j \text{ s.t. } \rho_i < \rho_j \\ \max\limits_{j} \left(d_{rmds}\left(\mathbf{x}_i, \mathbf{x}_j\right)\right), \text{ otherwise} \end{cases}, \tag{12}$$

where ρ_i, ρ_j $(1 \leq i, j \leq N)$ are obtained according to formula (8).

3.3 The description of the proposed algorithm

In this sub-section, we introduce the ideas of Sects. 3.1 and 3.2 into the DP clustering algorithm. The following algorithm is a summary of the proposed algorithm.

Algorithm. The proposed algorithm

Inputs:

The samples $\boldsymbol{X} \in \mathbb{R}_{N\times M}$

The parameter p, ε, α

Outputs:

The label vector of cluster index: $\boldsymbol{y} \in \mathbb{R}_{N\times 1}$

Method:

Step 1: Calculate distance matrix according to formula (1)

Step 2: Calculate ρ_i for point $\mathbf{x}_i$ according to formula (8)

Step 3: Calculate density-adaptive distance according to formula (11)

Step 4: Calculate δ_i for point $\mathbf{x}_i$ according to formula (12)

Step 5: Plot decision graph and select cluster centers

Step 6: Assign each remaining point to the cluster, which has its nearest neighbor with higher local density according to formula (11)

Step 7: **Return y**

3.4 Performance analysis

The computational complexity is an important indicator of the algorithm. If the complexity is too high, it will limit the application of the algorithm in complex scenes [24]. Now, we give the time complexity of the proposed algorithm. To be consistent with the aforementioned notations, we assume that N is the number of objects in the data set; M is the number of attributes. The computational complexity of the similarity matrix is $O\left(N^2\right)$. $O\left(N^2\right)$ is needed to compute the new local density. We calculate the density-adaptive distance by using Floyd's algorithm. The time complexity of Floyd's algorithm is $O\left(N^3\right)$. In addition, the cost of the sorting process with quick sort is $O\left(N \log N\right)$. We take no account of the time of determining the cluster centers. Considering the complexity in the assignment procedure is $O\left(N\right)$, the total time cost of the proposed algorithm is $O\left(N^2\right) + O\left(N^2\right) + O\left(N^3\right) + O\left(N \log N\right) + O\left(N\right) \sim O\left(N^3\right)$.

4 Experiments and results

In this section, experimental results show the clustering performance of our algorithm. To achieve this, we use two kinds of data (1) some synthetic data sets and (2) 7 real-world data sets, the Wisconsin Prognostic Breast Cancer (WPBC) data set [29], the Ionosphere data set [26], the Dermatology data set [10], the Wisconsin Diagnostic Breast Cancer (WDBC) data set [21], the Titanic data set, the Waveform Database, and Magic data set. And all of the real-world data sets are obtained from the UCI repository.

We compare the proposed algorithm with the original DP clustering algorithm and the classical clustering algorithms, such as, K-means [20], fuzzy C-means (FCM) [2], Gaussian mixture model (GMM) [6], and self-organizing maps (SOM) [16], in some evaluations of cluster quality. These evaluations include clustering accuracy (ACC), normalized mutual information (NMI), adjusted Rand index (ARI), and F_1 Score [12, 31].

We conduct experiments on a desktop computer with a core i7 DMI2-Intel 3.6 GHz processor and 16 GB RAM running MATLAB 2013A. In the DP clustering algorithm, the range of value of the parameter d_c is restricted to a set [0.1% 0.5% 1% 2% 4% 6%]. In the FCM method, the range of value of the parameter m is [1.2 1.5 2.0 2.5 3.0]. In the GMM algorithm, in order to ensure that the estimates are positive-definite, a non-negative regularization value, λ, is added to the diagonal of each covariance matrix. The range of value of the parameter λ is [0 0.01 0.1]. In the proposed algorithm, the range of value of the parameter p is [0.1% 0.5% 1% 2% 4% 6%], and we select the parameter ε from [0.01 0.05 0.1 0.2 0.5] or [1% 5% 10% 20% 50%]. The parameter p and the parameter ε are selected based on the clustering performance. In almost all our experiments (both on synthetic data sets and on real-world data sets), we set a default value of $\alpha = 1$. In order to avoid trapping into local optimum, K-means, FCM, GMM, and SOM are all repeated 10 times in each corresponding parameter value. On synthetic data sets, for demonstration purposes, we only present the best results in terms of clustering accuracy. On real-world data sets, we present the mean value and the standard deviation in the four evaluations.

In order to avoid clustering bias due to different scales in the attributes, a data normalization method is performed during the data preprocessing step. The min–max normalization is the

simplest method that is rescaling the range of attributes to scale the range in [0, 1]. The min–max normalization function is given by [11, 22, 36]

$$x' = \frac{x - \min_A}{\max_A - \min_A}, \tag{13}$$

where x' and x are the original and the normalized values, respectively; $\max_A$ and $\min_A$ are the maximal and the minimal values of an attribute A, respectively.

4.1 Experiments on synthetic data sets

We test the performance of the classical clustering algorithms and our algorithm on synthetic data sets. The main reason why we use synthetic data sets to validate clustering algorithms is that the structures of synthetic data sets can be controlled. In the following sub-sections, we introduce two kinds of synthetic data sets to validate our algorithm. The first category, the data sets with clusters of different densities, validates that the new local density helps to detect correct cluster centers. The second category, the data sets with clusters of twisted, folded, or curved data distribution, validates that the new δ and the new assignation strategy help to detect clusters.

4.1.1 Synthetic data sets

Some classical algorithms and the proposed algorithm are tested by ten data sets whose geometric shapes are shown in Fig. 5. The first data set, the Jain data set, is a kind of two moons data set with 2 clusters of 373 points. The second data set is the DD data set which has 3 clusters of different densities. The third data, the multi-scale data set [33], consists of 3 clusters that are of 238 points. The fourth data set, the Half-kernel data set, contains 2 clusters of 1000 points. The fifth data set, the Two-circles data set [23], contains 2 clusters of 400 points. The sixth data set, the A1 data set [33], contains 3 clusters of 299 points. Except for internal cluster, A1 looks just the same as three circles. The seventh data set, the Four-lines data set [33], contains 4 clusters of 512 points. The eighth data set, the A3 data set [33], contains 3 clusters of 266 points. The last data set, Crescent-full-moon data set, contains 2 clusters of 1000 points. The first four data sets are all of clusters of different densities. Other data sets are of clusters of twisted, folded, or curved data distribution. We demonstrate the power of our algorithm on these test cases. The details of these data sets are listed in Table 1.

4.1.2 The performance of clustering results on synthetic data sets

Two moons data set is widely used in some manifold learning algorithms and clustering algorithms [4]. Jain data set is a kind of two moons data set. Figure 6 shows comparison against the classical clustering algorithms on the Jain data set. On the Jain data set, the DP clustering is not able to find clusters whatever the value of the parameter d_c is. When the parameter d_c is fixed at 0.5%, the DP clustering obtains the best result on this data set, as shown in Fig. 6b. Other methods are repeated ten times in each corresponding parameter value. We select the best one in all clustering results. Similarly, they also fail to find clusters, as shown in Fig. 6c–f. Unlike DP, our method can effectively find two cluster centers and further obtains the correct clustering result.

The DD data set is of 3 clusters that have different densities. Figure 7 shows the clustering results on this data set. The DP method produces two cluster centers in the highest density

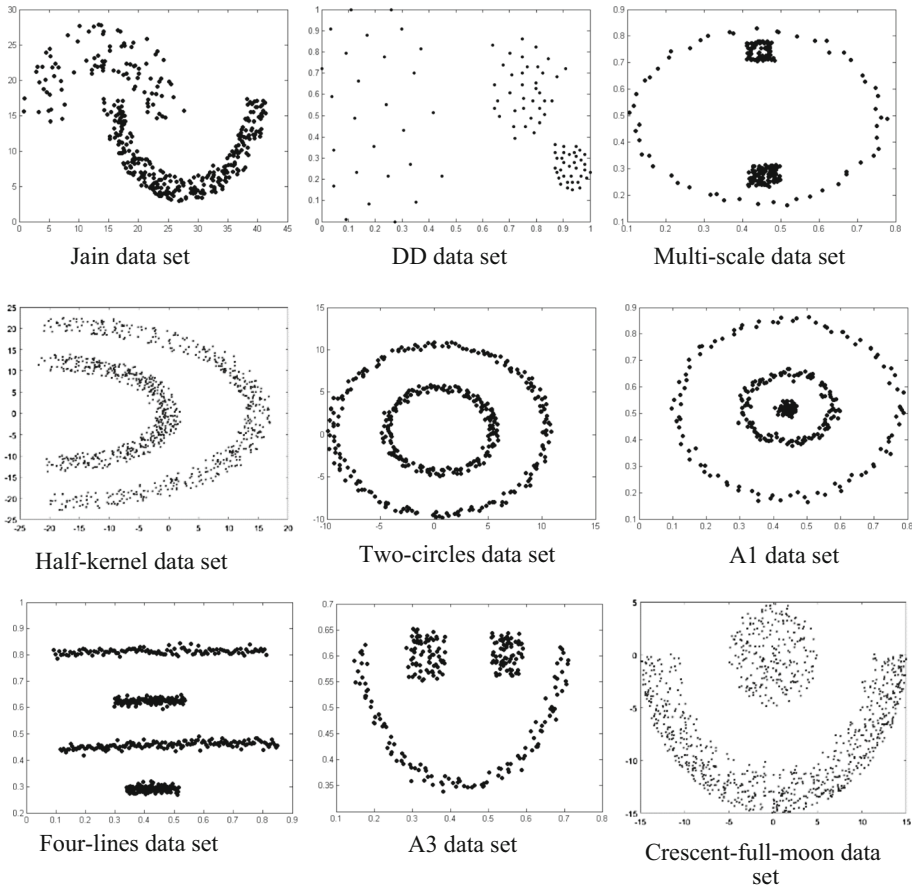


Fig. 5 Visualization of two-dimensional data sets

Table 1 Details of synthetic data sets

Data sets	Cluster	Dimension	Number
Jain	2	2	373
DD	3	2	99
Multi-scale	3	2	238
Half-kernel	2	2	1000
Two-circles	2	2	400
A1	3	2	299
Four-lines	4	2	512
A3	3	2	266
Crescent-full-moon	2	2	1000

cluster, as shown in Fig. 7b. The rest of the comparative methods are able to find clusters except FCM, as shown in Fig. 7c–f. That is, because these algorithms are adept at detecting spherical clusters. By comparison with the DP clustering algorithm, the proposed algorithm is able to detect these clusters in all the cases.

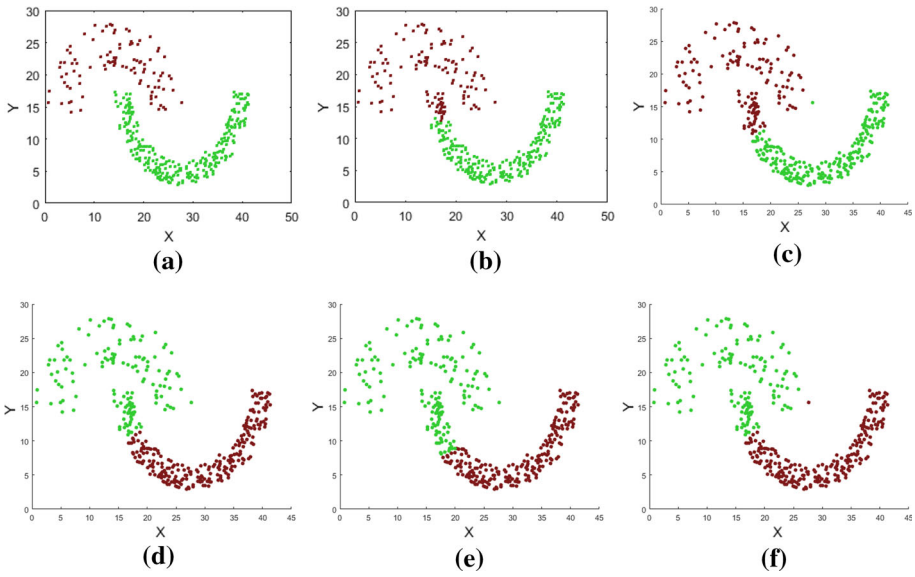


Fig. 6 Clustering results on the Jain data set. **a** Our method $p = 1\%$, $\varepsilon = 0.01$, **b** DP $d_c = 0.5\%\%$, **c** K-means, **d** FCM $m = 2.0$, **e** GMM $\lambda = 0.01$, and **f** SOM

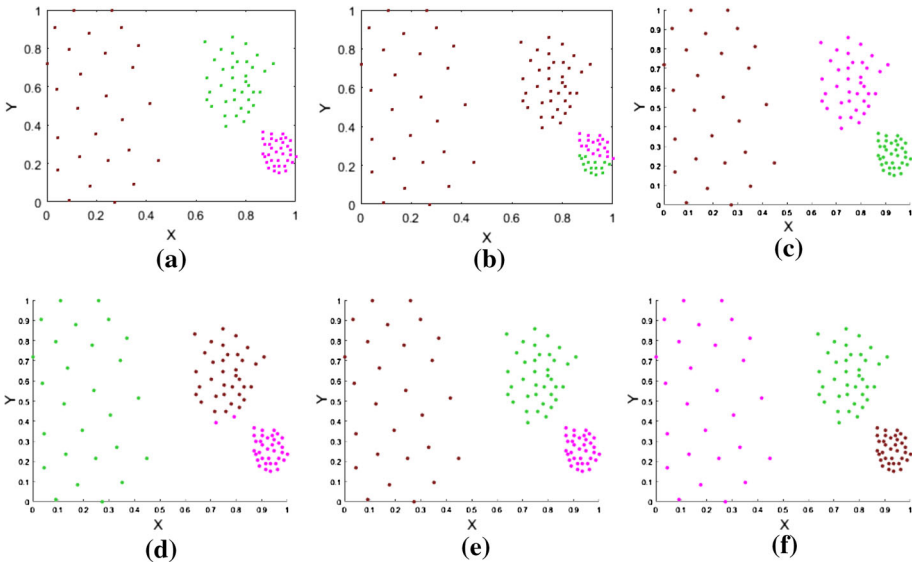


Fig. 7 Clustering results on the DD data set. **a** Our method $p = 2\%$, $\varepsilon = 0.01$, **b** DP $d_c = 0.5\%$, **c** K-means, **d** FCM $m = 2.0$, **e** GMM $\lambda = 0$, and **f** SOM

The multi-scale data set has two quadrilateral clusters and one circular cluster. Figure 8 shows the clustering results on this data set. The densities of the two quadrilateral clusters are much higher than that of the circular cluster. As shown in Fig. 8b, the DP clustering method cannot even find correct cluster number and can only generate two clusters on this data set. Other traditional methods also perform poorly, as shown in Fig. 8c–f. Owing to the

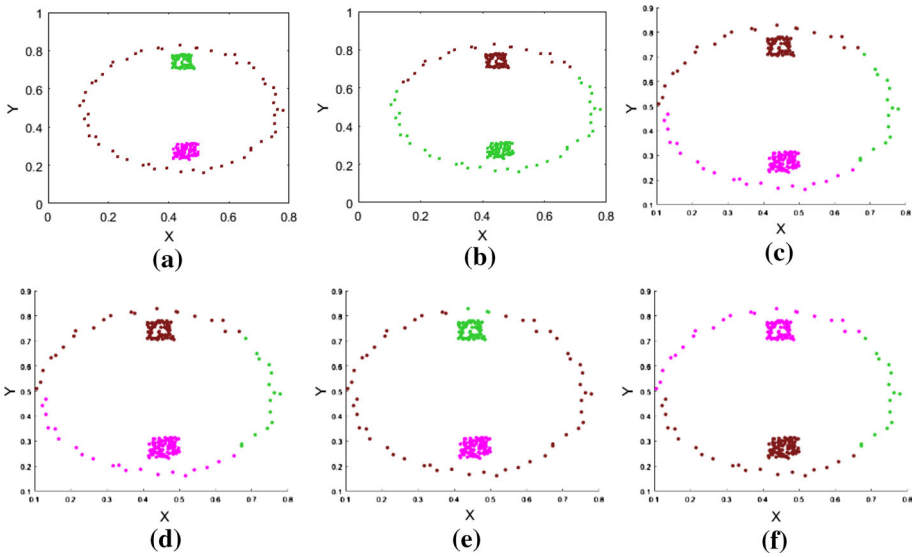


Fig. 8 Clustering results on the multi-scale data set. **a** Our method $p = 2\%, \varepsilon = 0.01$, **b** DP $d_c = 1\%$, **c** K-means, **d** FCM $m = 2.0$, **e** GMM $\lambda = 0$, and **f** SOM

new local density and the new assignation strategy, the proposed method does an excellent job in clustering this data set.

The Half-kernel data set has two clusters of different densities. Figure 9 shows the clustering results on this data set. The DP clustering cannot find the optimal structure of clusters, as shown in Fig. 9b. Despite other methods fail to obtain good results, our algorithm does an excellent job in clustering this data set.

As shown in Figs. 6, 7, 8, and 9, the proposed algorithm gets better clustering performance compared with these classical clustering algorithms in almost all case. Especially, the proposed algorithm gets extraordinarily favorable performance compared with the DP clustering algorithm for the above data sets. This is because the concept of the sensitivity of the local density is introduced into the calculation of the local density. The original algorithm is improved in finding the cluster centers for some data sets of different densities. The experimental results illustrate our algorithm is very effective in finding clusters of arbitrary density.

The Two-circles data set contains two circular clusters, where the density of the inner cluster is slightly higher than that of the outer cluster. It is easy to see why the DP method does a poor job. Firstly, we introduce the new local density into the original DP method, and it can find two center points in different clusters. However, it still cannot generate the optimal structure of clusters, as shown in Fig. 2b. Thus, our algorithm further improves the calculation of δ and the assignation strategy according to a new density-adaptive distance. Figure 10 shows that the proposed algorithm gets extraordinarily favorable performance compared with other algorithms.

Analogous to the Two-circles data set, the A1 data set adds a spherical cluster on the basis of that data set. Figure 11 shows the clustering results on this data set. The proposed algorithm clearly outperforms these traditional clustering methods on this data set.

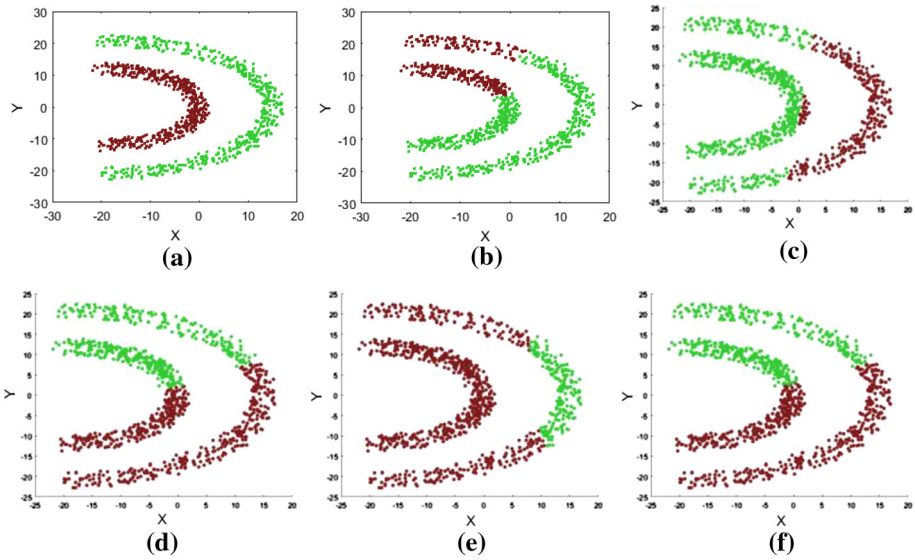


Fig. 9 Clustering results on the Half-kernel data set. **a** Our method $p = 2\%, \varepsilon = 0.01$, **b** DP $d_c = 2\%$, **c** K-means, **d** FCM $m = 2.5$, **e** GMM $\lambda = 0$, and **f** SOM

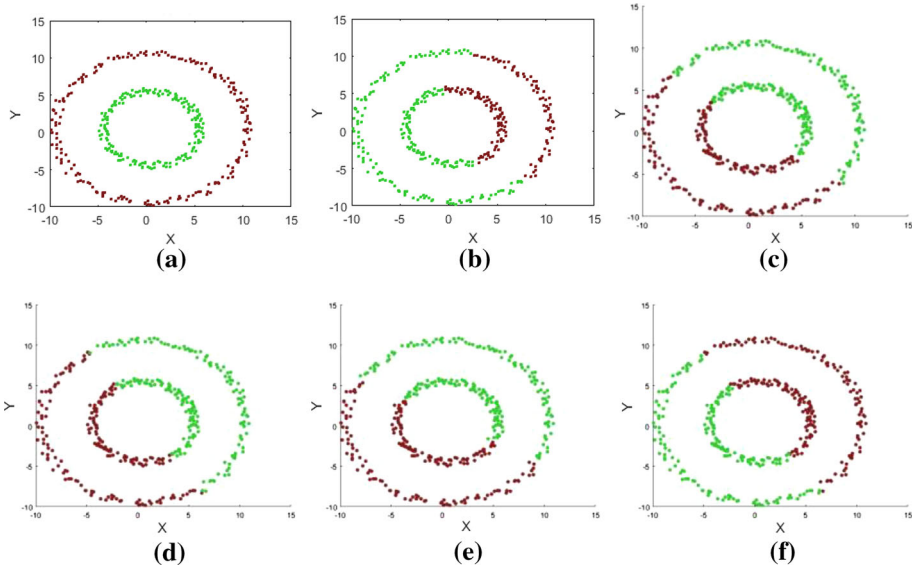


Fig. 10 Clustering results on the two-circles data set. **a** Our method $p = 6\%, \varepsilon = 0.1$, **b** DP $d_c = 2\%$, **c** K-means, **d** FCM $m = 2.0$, **e** GMM $\lambda = 0$, and **f** SOM

The Four-lines data set contains four linear clusters. GMM and our algorithm perform slightly better, as shown in Fig. 12.

The A3 data set is a challenging data set. The proposed method does an excellent job in clustering this data, as shown in Fig. 13.

Figure 14 shows that our method can effectively find clusters.

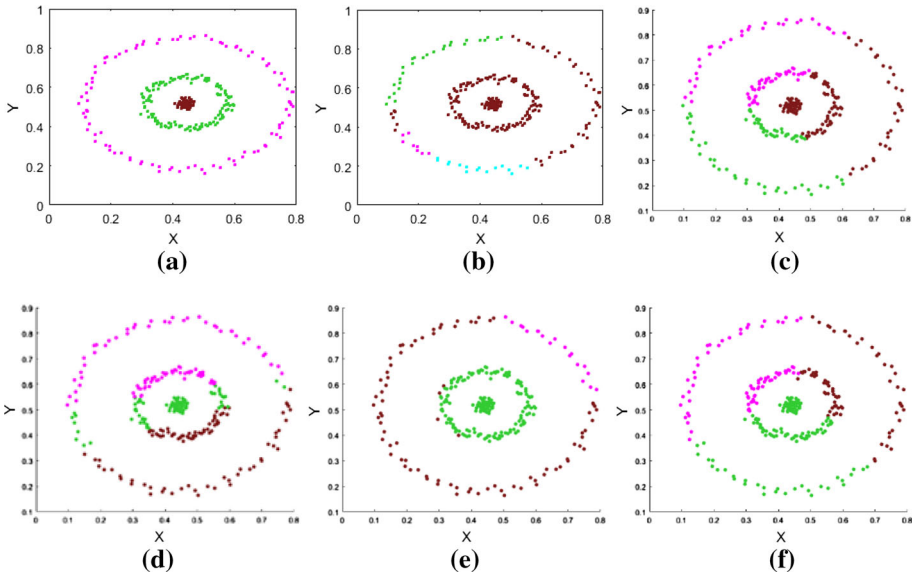


Fig. 11 Clustering results on the A1 data set. **a** Our method $p = 4\%$, $\varepsilon = 0.01$, **b** DP $d_c = 4\%$, **c** K-means, **d** FCM $m = 3.0$, **e** GMM $\lambda = 0$, and **f** SOM

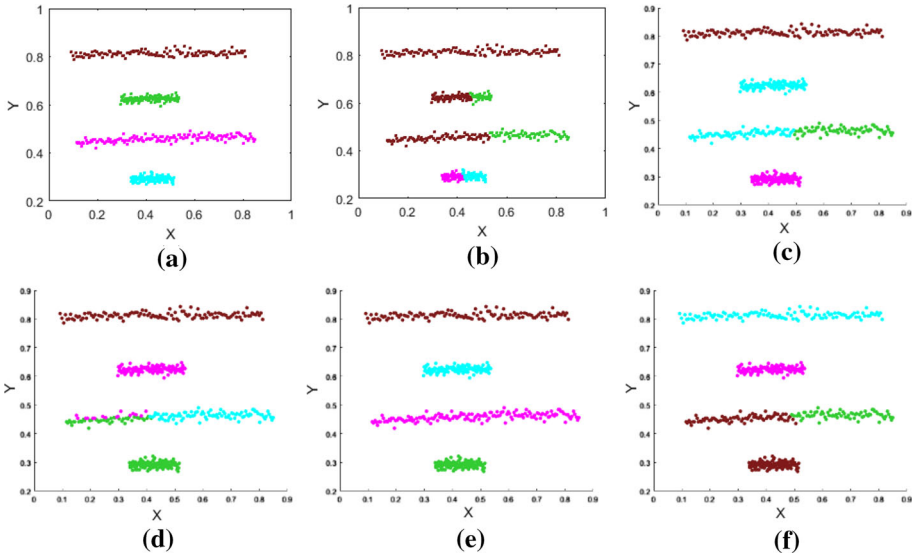


Fig. 12 Clustering results on the Four-lines data set. **a** Our method $p = 6\%$, $\varepsilon = 0.01$, **b** DP $d_c = 0.5\%$, **c** K-means, **d** FCM $m = 3.0$, **e** GMM $\lambda = 0$, and **f** SOM

The original DP clustering method is not able to detect clusters of twisted, folded, or curved data distribution. Based on a new local density, we also introduce a new density-adaptive metric into the calculation of δ and the assignation strategy. As shown in Figs. 10, 11, 12, 13, and 14, on these data sets with different manifolds, this metric squeezes the distance between data points in the same cluster (i.e., high-density area). From a different perspective, the

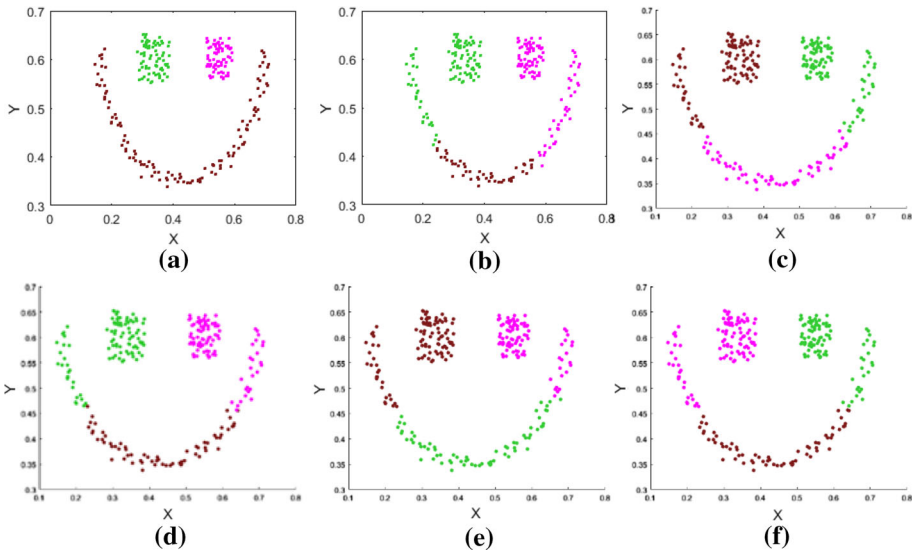


Fig. 13 Clustering results on the A3 data set. **a** Our method $p = 4\%, \varepsilon = 0.01$, **b** DP $d_c = 6\%$, **c** K-means, **d** FCM $m = 3.0$, **e** GMM $\lambda = 0$, and **f** SOM

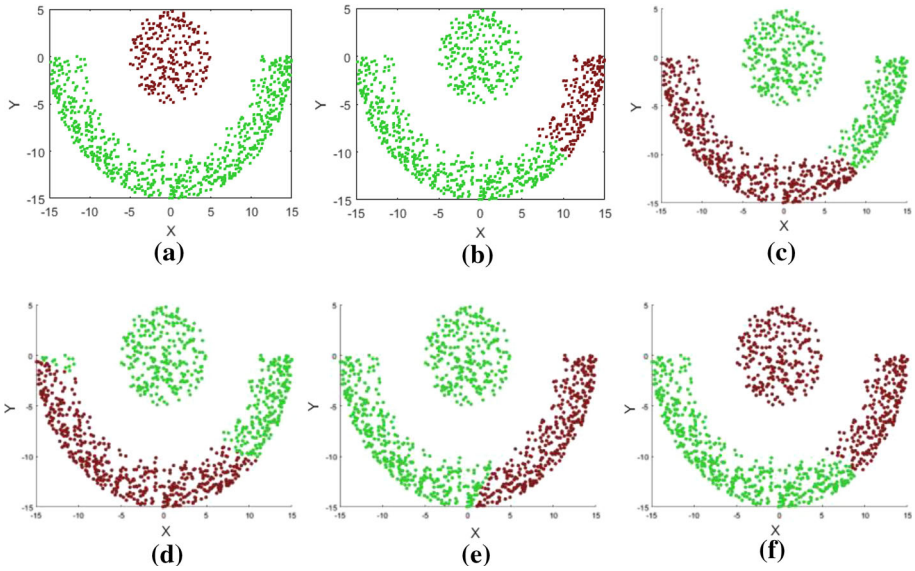


Fig. 14 Clustering results on the Crescent-full-moon data set. **a** Our method $p = 1\%, \varepsilon = 0.01$, **b** DP $d_c = 1\%$, **c** K-means, **d** FCM $m = 2.0$, **e** GMM $\lambda = 0$, and **f** SOM

distance between cluster centers is comparatively magnified in disguise. In other words, the values of δ of the center candidates become larger contributing to more prone selecting the "real" centers. This is desirable for high-quality clustering. The proposed algorithm clearly outperforms these classical clustering algorithms through most of the cases, as shown in Figs. 10, 11, 12, 13, and 14.

Table 2 Details of UCI data sets

Data sets	Cluster	Dimension	Number
WPBC	2	33	198
Ionosphere	2	34	351
Dermatology	6	34	366
WDBC	2	30	569
Titanic	2	3	2201
Waveform	3	21	5000
Magic	2	10	19,020

4.2 Experiments on real-world data sets

The performance of the proposed algorithm is compared with some classical methods, such as DP, *K*-means, FCM, GMM, and SOM on 7 real-world data sets.

4.2.1 Real-world data sets

The real-world data sets used in the experiments are all from the UCI Machine Learning Repository. On the Wisconsin Prognostic Breast Cancer (WPBC) data set, all objects are divided into two classes, "recur" (class #1), including 151 objects, and "non-recur" (class #2), including 47 objects. According to its class distribution, this data set is a typical imbalanced data set. Its imbalance ratio [9] value is higher than 3. The Ionosphere data set is classified into two classes: good (class #1), including 225 objects, and bad (class #2), including 126 objects. Its imbalance ratio value is about 1.9. The Dermatology data set is classified into six classes: psoriasis (class #1), including 112 objects; seborrheic dermatitis (class #2), including 61 objects; lichen planus (class #3), including 72 objects; pityriasis rosea (class #4), including 49 objects; chronic dermatitis (class #5), including 52 objects; and pityriasis rubra pilaris (class #6), including 20 objects. The Wisconsin Diagnostic Breast Cancer (WDBC) data set is classified into two classes: benign (class #1), including 357 objects and malignant (class #2), including 212 objects. The Titanic data set is classified into two classes: survivor (class #1), including 711 objects and victim (class #2), including 1490 objects. Its imbalance ratio value is higher than 2. It is to be observed that the Waveform data set is a sample of 5000 objects obtained from the original data generator. Thus, the Waveform data set consists of 5000 objects with 21 attributes and with continuous values between 0 and 6. Each instance is generated added noise (mean 0, variance 1) in each attribute, and each class is generated from a combination of 2 of 3 "base" waves. It is classified into three classes: class #1, including 1657 objects; class #2, including 1647 objects; and class #3, including 1696 objects. The Magic data set is classified into two classes: gamma (class #1), including 12,332 objects and hadron (class #2), including 6688 objects. Its imbalance ratio value is about 1.8. The details of these data sets are listed in Table 2.

4.2.2 The performance of clustering results on real-world data sets

In Table 3, we list the performance of our proposed algorithm, DP, *K*-means, FCM, GMM, and SOM on the WPBC data set. In Table 3, the symbol means that the algorithm cannot work in the data set (the reasons are explained later). In the following tables, the numbers highlighted in bold indicate that the corresponding algorithm has the best performance in

Table 3 Performance comparison of the proposed algorithm on the WPBC data set

Algorithms	ACC	NMI	ARI	F_1
Our method $p = 6\%$, $\varepsilon = 0.01$	**0.7525**	**0.0552**	**0.1593**	**0.7200**
DP	–	–	–	–
K-means $k = 2$	0.6010 ± 0	0.0241 ± 0	0.0351 ± 0	0.5763 ± 0
FCM $m = 1.2$	0.6010 ± 0	0.0209 ± 0	0.0340 ± 0	0.5771 ± 0
GMM $\lambda = 0$	0.5843 ± 0.0451	0.0101 ± 0.0068	0.0023 ± 0.0259	0.5911 ± 0.0296
SOM $k = 2$	0.6010 ± 0	0.0241 ± 0	0.0351 ± 0	0.5763 ± 0

Table 4 Performance comparison of the proposed algorithm on the Ionosphere data set

Algorithms	ACC	NMI	ARI	F_1
Our method $p = 1\%$, $\varepsilon = 0.05$	**0.8205**	**0.2894**	**0.3953**	**0.7515**
DP	–	–	–	–
K-means $k = 2$	0.7123 ± 0	0.1312 ± 0	0.1776 ± 0	0.6049 ± 0
FCM $m = 1.2$	0.7094 ± 0	0.1264 ± 0	0.1727 ± 0	0.6028 ± 0
GMM $\lambda = 0.1$	0.6590 ± 0.0799	0.0802 ± 0.0544	0.0802 ± 0.0684	0.6659 ± 0.0666
SOM $k = 2$	0.7117 ± 0.0012	0.1303 ± 0.0020	0.1766 ± 0.0021	0.6045 ± 0.0009

terms of its corresponding evaluation (i.e., the corresponding column). As shown in Table 3, the values of the quality measures of the clusters formed by K-means, FCM, and SOM are close, within a difference of 0.001. However, GMM performs slightly worse. Table 3 shows the results of our algorithm are significantly better than those obtained by other methods. It is worth mentioning that the proposed algorithm is at least 15% more accurate than other methods. The ARI value of our method is one order of magnitude higher than those of other methods.

Table 4 shows comparison against the classical clustering algorithms on the Ionosphere data set. Similar to the results on the WPBC data set, the values of the quality measures of the clusters formed by K-means, FCM, and SOM are close, within a difference of 0.01 and GMM performs slightly worse. As we can see, the clustering results obtained by our method are superior to those obtained by other methods. The ACC value and the F_1 value of the proposed algorithm are about 10% higher than those of other methods. The NMI value and the ARI value of the proposed algorithm are twice more than those of other methods.

In Table 5, we list the performance of our proposed algorithm, DP, K-means, FCM, GMM, and SOM on the Dermatology data set. As can be seen in Table 5, our algorithm outperforms against the competitors in terms of the ACC. Both the ARI and the F_1 of the proposed method

Table 5 Performance comparison of the proposed algorithm on the Dermatology data set

Algorithms	ACC	NMI	ARI	F_1
Our method $p = 6\%$, $\varepsilon = 0.5$	**0.8579**	0.8022	**0.7455**	**0.7921**
DP	–	–	–	
K-means $k = 6$	0.6809 ± 0.1041	0.8215 ± 0.0863	0.6582 ± 0.1341	0.7303 ± 0.1007
FCM $m = 1.5$	0.7448 ± 0.0052	0.7733 ± 0.0039	0.6955 ± 0.0151	0.7511 ± 0.0134
GMM $\lambda = 0.01$	0.6929 ± 0.0594	0.7830 ± 0.0510	0.6552 ± 0.0882	0.7240 ± 0.0713
SOM $k = 6$	0.7361 ± 0.0039	$\mathbf{0.8765 \pm 0.0005}$	0.7152 ± 0.0012	0.7717 ± 0.0010

Table 6 Performance comparison of the proposed algorithm on the WDBC data set

Algorithms	ACC	NMI	ARI	F_1
Our method $p = 4\%, \varepsilon = 0.01$	**0.9508**	**0.7029**	**0.8116**	**0.9125**
DP	–	–	–	–
K-means $k = 2$	0.9279 ± 0	0.6115 ± 0	0.7302 ± 0	0.8702 ± 0
FCM $m = 3.0$	0.9367 ± 0	0.6434 ± 0	0.7614 ± 0	0.8889 ± 0
GMM $\lambda = 0$	0.9186 ± 0.0314	0.5925 ± 0.0874	0.7033 ± 0.0998	0.8592 ± 0.0493
SOM $k = 2$	0.9281 ± 0.0006	0.6123 ± 0.0025	0.7308 ± 0.0019	0.8771 ± 0.0009

are slightly better than those of other solutions. However, on the Dermatology data set, SOM performs the best job in terms of the NMI.

Table 6 lists the clustering results on the WDBC data set. All methods (except for DP) generate great results. By comparison with these classic methods (DP, K-means, FCM, GMM, and SOM), the proposed algorithm shows a small advantage in terms of the quality measures (ACC, NMI, ARI, and F_1) on this data set.

In Table 7, we list the performance of our proposed algorithm, DP, K-means, FCM, GMM, and SOM on the Titanic data set. K-means, SOM, and GMM have similar performance. FCM and our method have the same performance, and they perform better than other methods on the Titanic data set.

Table 8 lists the clustering results on the Waveform data set. Due to including class noise and attribute noise, results from Table 8 illustrate that all methods cannot generate an excellent structure on this data set. Compared with other methods, our method shows a small advantage in terms of the most quality measures (ACC, NMI and ARI). While, it is slightly inferior to GMM in terms of F_1.

Table 7 Performance comparison of the proposed algorithm on the Titanic data set

Algorithms	ACC	NMI	ARI	F_1
Our method $p = 4\%, \varepsilon = 0.05$	**0.7760**	**0.1569**	**0.2747**	**0.7164**
DP	–	–	–	–
K-means $k = 2$	0.7343 ± 0.0393	0.1006 ± 0.0652	0.1813 ± 0.1058	0.6908 ± 0.0192
FCM $m = 3.0$	$\mathbf{0.7760 \pm 0}$	$\mathbf{0.1569 \pm 0}$	$\mathbf{0.2747 \pm 0}$	$\mathbf{0.7164 \pm 0}$
GMM $\lambda = 0.01$	0.7448 ± 0.0458	0.1114 ± 0.0734	0.1981 ± 0.1232	0.7071 ± 0.0047
SOM $k = 2$	0.7506 ± 0	0.1341 ± 0	0.2348 ± 0	0.6731 ± 0

Table 8 Performance comparison of the proposed algorithm on the Waveform data set

Algorithms	ACC	NMI	ARI	F_1
Our method $p = 0.5\%, \varepsilon = 0.01$	**0.6612**	**0.3715**	**0.2934**	0.5306
DP $d_c = 4\%$	0.5446	0.2535	0.2657	0.5223
K-means $k = 3$	0.5010 ± 0.0004	0.3632 ± 0	0.2535 ± 0	0.5037 ± 0
FCM $m = 2.5$	0.4930 ± 0	0.3291 ± 0	0.2436 ± 0	0.4974 ± 0
GMM $\lambda = 0$	0.6037 ± 0.0096	0.2539 ± 0.0112	0.2676 ± 0.0220	$\mathbf{0.5624 \pm 0.0139}$
SOM $k = 3$	0.5014 ± 0.0010	0.3632 ± 0	0.2536 ± 0	0.5037 ± 0

Table 9 shows comparison against the classical clustering algorithms on the Magic data set. As we can see, the clustering results obtained by our method are superior to those obtained by other methods.

In Tables 3, 4, 5, 6, and 7, we can see that DP cannot work on the corresponding data set. The main reason is that we cannot find the optimal cluster number based on DP's decision graph on these data sets. In the case that DP deals with data in the Titanic set, only one cluster center is found by DP with different d_c on decision graph, as shown in Fig. 15. It is unacceptable that we are incapable of making the right choices.

In most of the cases, our algorithm obtains better clustering performance compared with these classical clustering algorithms, especially to the DP clustering algorithm. We conjecture that the main reason is that most of experimental data sets are imbalanced and our method is suitable for processing some specific data sets with clusters of different densities. We assume that a data set is imbalanced when its imbalance ratio value is higher than 1.8. Therefore, in this paper, the imbalanced data sets include WPBC, Ionosphere, Titanic, and Magic. These types of sets suppose a new challenging problem for these traditional methods (DP, K-means,

Table 9 Performance comparison of the proposed algorithm on the Magic data set

Algorithms	ACC	NMI	ARI	F_1
Our method $p = 1\%, \varepsilon = 0.01$	**0.6612**	**0.0323**	**0.0831**	**0.6613**
DP $d_c = 0.5\%$	0.5173	0.0251	-0.0133	0.5561
K-means $k = 3$	0.5917 ± 0.0005	0.0159 ± 0	0.0317 ± 0	0.5420 ± 0
FCM $m = 1.2$	0.6073 ± 0	0.0228 ± 0	0.0438 ± 0	0.5488 ± 0
GMM $\lambda = 0$	0.6292 ± 0	0.0269 ± 0	0.0597 ± 0	0.5705 ± 0
SOM $k = 3$	0.5915 ± 0.0010	0.0159 ± 0.0003	0.0316 ± 0.0007	0.5418 ± 0.0005

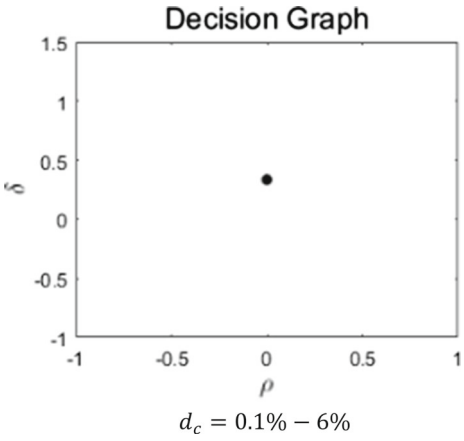


Fig. 15 DPC on Titanic set with different values of d_c

FCM, GMM, and SOM). For these four data sets, we try to calculate the data density in every class. We adopt the simplest method to estimate the density in a class C_i:

$$\text{den}\,(C_i) = \frac{|C_i|}{V_i}, \tag{14}$$

where $|C_i|$ is the number of objects in the class C_i and $V_i = \prod_j \text{DOM}_j^i$ is the volume of the attribute domains in the class C_i. $\text{DOM}_j^i = \max_{A_j}^i \left(x'_{(i)}\right) - \min_{A_j}^i \left(x'_{(i)}\right)$ is the size of the range (of the normalized values) of the attribute A_j in the class C_i. Thus, V_i is the volume of the hypercube in the M-dimensional space, and den (C_i) is the number of objects of the class C_i in the unit hypercube, i.e., density. We can easily get the fact that the two classes have the same "volume" (0.282) on the WPBC data set. The same situation applies to the other data sets (Ionosphere:0.0434; Titanic:1; Magic:0.0856). The density of the class with more objects is higher than that of another class. And our method is designed to handle such data sets (including clusters of different densities). The new local density computation can reduce the local density gap between low-density clusters and high-density clusters. It helps detect correct cluster centers. On the WPBC, Ionosphere, and Titanic data sets, DP cannot

even find correct cluster number, but our method can perform a good job. The experimental results illustrate the superior performance of our algorithm compared with other clustering approaches. Note that these four data sets are all special cases. Actually, not all the imbalanced data sets have this property: Different classes have the same "volume."

5 Conclusions

Referring to the ideas of the prior assumption of consistency for semi-supervised learning problems, we make the assumptions of consistency for density-based clustering. In order to tackle the problem that DP does a poor job of find clusters with different densities, we provide a new option based on the sensitivity of the local density for the local density. This satisfies the assumption of the local consistency. In addition, the original assignation strategy does not take into account of the assumption of the global consistency. Therefore, we propose a new density-adaptive metric which can magnify the path length in a low-density area and shorten the path length in a high-density area. Furthermore, we redefine δ and redesign the assignation strategy based on this metric. The experimental results support our claim that the proposed algorithm is an efficient algorithm.

Unlike DP, our method requires two parameters: p and ε. As a rule of thumb, one can choose $\varepsilon = 0.01$, and p is around 1–6%. But, indeed, it is not simple to estimate the optimal parameters if the ground truth is unknown. Future works will develop a method with nonparametric estimation calculates the local density. Moreover, there are also some other approaches to addressing the issue that the clusters have greatly varied densities. For example, a solution for the weakness is finding different parameter d_{c_i} for different data point $\mathbf{x}_i$. However, how to select all the parameters automatically is one of the interesting challenges. In future works, we will focus on finding the solution.

In addition, similar to DP, our method cannot select the cluster centers automatically. Thus, we want to develop an automatic cluster centroid selection method. As described in Sect. 3.3, our method becomes computationally expensive, since the density-adaptive distance requires minimizing a function for each pair of points. We will try to introduce the idea of the grid into our method. The cost is only associated with the number of cells. And the number of cells K is far less than the number of objects N.

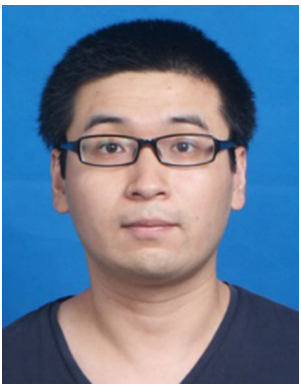
Acknowledgements This work is supported by the National Natural Science Foundation of China (Nos. 61672522 and 61379101), and the National Key Basic Research Program of China (No. 2013CB329502).

References

1. Ankerst M, Breunig MM, Kriegel HP et al (1999) OPTICS: ordering points to identify the clustering structure. In: Proceedings of the ACM international conference on management of data, pp 49–60
2. Backer E, Jain AK (1981) A clustering performance measure based on fuzzy set decomposition. IEEE Trans Pattern Anal Mach Intell 3(1):66–75
3. Chen G, Zhang X, Wang ZJ et al (2015) Robust support vector data description for outlier detection with noise or uncertain data. Knowl-Based Syst 90:129–137
4. Chen WJ, Shao YH, Hong N (2014) Laplacian smooth twin support vector machine for semi-supervised classification. Int J Mach Learn Cybern 5(3):459–468
5. Chen Z, Qi Z, Meng F et al (2015) Image segmentation via improving clustering algorithms with density and distance. Proc Comput Sci 55:1015–1022
6. Dempster AP, Laird NM, Rubin DB (1977) Maximum likelihood from incomplete data via the EM algorithm. J R Stat Soc Ser B (Methodological) 39(1):1–38

7. Du M, Ding S, Jia H (2016) Study on density peaks clustering based on k-nearest neighbors and principal component analysis. Knowl-Based Syst 99:135–145
8. Ester M, Kriegel HP, Sander J et al (1996) A density-based algorithm for discovering clusters in large spatial databases with noise. In: Proceedings of second international conference on knowledge discovery and data mining, pp 226–231
9. Fernández A, García S, del Jesus MJ et al (2008) A study of the behaviour of linguistic fuzzy rule based classification systems in the framework of imbalanced data-sets. Fuzzy Sets Syst 159(18):2378–2398
10. Güvenir HA, Demiröz G, Ilter N (1998) Learning differential diagnosis of erythemato-squamous diseases using voting feature intervals. Artif Intell Med 13(3):147–165
11. He Q, Jin X, Du C et al (2014) Clustering in extreme learning machine feature space. Neurocomputing 128:88–95
12. Iam-On N, Boongoen T, Kongkotchawan N (2014) A new link-based method to ensemble clustering and cancer microarray data analysis. Int J Collab Intell 1(1):45–67
13. Jain AK, Law MC (2005) Data clustering: a user's Dilemma. In: Proceedings of first international conference of the pattern recognition and machine intelligence, pp 20–22
14. Jia H, Ding S, Meng L et al (2014) A density-adaptive affinity propagation clustering algorithm based on spectral dimension reduction. Neural Comput Appl 25(7–8):1557–1567
15. Jiang X, Zhang W (2016) Structure learning for weighted networks based on Bayesian nonparametric models. Int J Mach Learn Cybern 7(3):479–489
16. Kohonen T (1982) Self-organized formation of topologically correct feature maps. Biol Cybern 43(1):59–69
17. Liang Z, Chen P (2016) Delta-density based clustering with a divide-and-conquer strategy: 3DC clustering. Pattern Recogn Lett 73:52–59
18. Lu K, Xia S, Xia C (2015) Clustering based road detection method. In: Proceedings of the 34th Chinese control conference, pp 3874–3879
19. Ma T, Wang Y, Tang M et al (2016) LED: a fast overlapping communities detection algorithm based on structural clustering. Neurocomputing 207:488–500
20. MacQueen J (1967) Some methods for classification and analysis of multivariate observations. In: Proceedings of the fifth Berkeley symposium on mathematical statistics and probability, pp 281–297
21. Mangasarian OL, Street WN, Wolberg WH (1995) Breast cancer diagnosis and prognosis via linear programming. Oper Res 43(4):570–577
22. Mohamad IB, Usman D (2013) Standardization and its effects on k-means clustering algorithm. Res J Appl Sci Eng Technol 6(17):3299–3303
23. Ng AY, Jordan MI, Weiss Y (2002) On spectral clustering: analysis and an algorithm. In: Proceedings of advances in neural information processing systems, pp 849–856
24. Pan Z, Lei J, Zhang Y et al (2016) Fast motion estimation based on content property for low-complexity H.265/HEVC encoder. IEEE Trans Broadcast 62(3):675–684
25. Rodriguez A, Laio A (2014) Clustering by fast search and find of density peaks. Science 344(6191):1492–1496
26. Sigillito VG, Wing SP, Hutton LV et al (1989) Classification of radar returns from the ionosphere using neural networks. Johns Hopkins APL Tech Dig 10(3):262–266
27. Wang B, Zhang J, Liu Y et al (2017) Density peaks clustering based integrate framework for multi-document summarization. CAAI Trans Intell Technol 2(1):26–30
28. Wang L, Bo LF, Jiao LC (2007) Density-sensitive spectral clustering. Acta Electron Sin 35(8):1577–1581
29. Wolberg WH, Street WN, Heisey DM et al (1995) Computerized breast cancer diagnosis and prognosis from fine-needle aspirates. Arch Surg 130(5):511–516
30. Xu X, Ding S, Du M et al (2016) DPCG: an efficient density peaks clustering algorithm based on grid. Int J Mach Learn Cybern. https://doi.org/10.1007/s13042-016-0603-2
31. Xu X, Law R, Chen W et al (2016) Forecasting tourism demand by extracting fuzzy Takagi–Sugeno rules from trained SVMs. CAAI Trans Intell Technol 1(1):30–42
32. Yang P, Zhu Q, Huang B (2011) Spectral clustering with density sensitive similarity function. Knowl-Based Syst 24(5):621–628
33. Zelnik-Manor L, Perona P (2004) Self-tuning spectral clustering. In: Proceedings of advances in neural information processing systems, pp 1601–1608
34. Zhang W, Li J (2015) Extended fast search clustering algorithm: widely density clusters, no density peaks. https://doi.org/10.5121/csit.2015.50701. arXiv preprint arXiv:1505.05610
35. Zhang Y, Xia Y, Liu Y et al (2015) Clustering sentences with density peaks for multi-document summarization. In: Proceedings of human language technologies: the 2015 annual conference of the North American Chapter of the ACL, pp 1262–1267

36. Zhong Q, Chen F (2016) Trajectory planning for biped robot walking on uneven terrain–Taking stepping as an example. CAAI Trans Intell Technol 1(3):197–209
37. Zhou D, Bousquet O, Lal TN et al (2004) Learning with local and global consistency. In: Proceedings of advances in neural information processing systems, pp 321–328



Mingjing Du born in Jiangsu, China, in 1989. He is working toward the Ph.D. degree in computer science and technology at China University of Mining and Technology. His research interests are in density peaks clustering, machine learning, and data mining.

Shifei Ding born in Qingdao, China, in 1963, received his Ph.D. degree from Shandong University of Science and Technology in 2004. He received postdoctoral degree from Key Laboratory of Intelligent Information Processing (IIP), Institute of Computing Technology (ICT), Chinese Academy of Sciences (CAS). He is a professor and Ph.D. supervisor at China University of Mining and Technology. His research interests include intelligent information processing, pattern recognition, machine learning, data mining, and granular computing. He has published five books and more than about 180 papers in international conferences and journals.

Yu Xue born in Shandong, China, in 1990, received his Ph.D. degree from Nanjing University of Aeronautics and Astronautics. He is an associate professor at Nanjing University of Information Science and Technology. His research interests include pattern recognition, machine learning, and data mining.

Zhongzhi Shi born in Jiangsu, China, in 1941, graduated in computer science from the Graduate School of University of Science and Technology of China in 1968 and graduated in computer science from the University of Science and Technology of China in 1964. Professor Shi's research and teaching interests are in the areas on intelligence science, distributed intelligence, machine learning, neural computing, and data mining. He has published 11 monographs, 12 books, and more than 400 research papers in journals and conferences.