Pattern Recognition 152 (2024) 110458

Contents lists available at ScienceDirect

Pattern Recognition

journal homepage: www.elsevier.com/locate/pr

Fast main density peak clustering within relevant regions via a robust decision graph☆

Junyi Guan [a], Sheng Li [a,*], Jinhui Zhu [b], Xiongxiong He [a], Jiajia Chen [a]

[a] College of Information Engineering, Zhejiang University of Technology, Hangzhou 310023, China
[b] Second Affiliated Hospital, Zhejiang University School of Medicine, Hangzhou 310009, China

ARTICLE INFO

MSC:
00-01
99-00

Keywords:
Clustering
Density peak
Decision graph
kNN

ABSTRACT

Although Density Peak Clustering (DPC) can easily locate cluster centers by detecting density peaks in its decision graph, its allocation strategy may unadvisedly associate irrelevant points, its decision graph may mislead the cluster center selection, and its high computational complexity $O(n^2)$ shies itself away from large-scale data. Herein, a Fast Main Density Peak Clustering Within Relevant Regions Via A Robust Decision Graph (R-MDPC) is proposed. R-MDPC assigns points within the relevant regions to avoid the association of irrelevant points. With the removal of regional differences and the attenuation of satellite peaks, a robust decision graph is obtained. Moreover, based on the kNN distance of data points, R-MDPC is believed to be suitable for large-scale data. Experimental results demonstrated the high robustness of R-MDPC's decision graph in identifying cluster centers, and its outstanding performance and fast running speed in recognizing complex-shaped clusters.

1. Introduction

Clustering as a problem without a unique solution is critical for data analysis in pattern recognition, machine learning, image processing [1,2], etc. The aim of clustering is to automatically divide similar points into clusters. By modeling data based on clusters, the exploration of complex data information becomes simpler [3]. Nonetheless, the identification of clusters still faces challenges due to the internal complexity of data.

K-means [4] is a well-known clustering algorithm that minimizes the distance between points and cluster centers to perform clustering. It has been widely applied due to its simplicity and effectiveness. Nevertheless, it heavily depends on the initialization of cluster centers and may fail to identify clusters of nonconvex and nested shapes, because points are always assigned to the nearest centers [5].

Density-based methods can effectively remedy the deficiencies of K-means. As a classic density-based algorithm, Density-based Spatial Clustering of Applications With Noise (DBSCAN) [6], can reconstruct clusters of arbitrary shapes according to a specific density-connectivity criterion. However, DBSCAN with a wide density-connectivity threshold may merge the overlapping clusters when dealing with high overlapping clusters, while with a rigor threshold, it may lose its cluster reconstruction ability [7].

DPC proposed by Rodriguez and Laio [8] can easily separate highly overlapping clusters by searching for density peaks as cluster centers in its decision graph. After labeling the selected cluster centers, each remaining point inherits the label of its nearest point of a higher density to complete clustering. Such an easy and efficient implementation of locating cluster centers makes DPC one of the top-performing clustering methods [1,9,10]. Still, DPC's embedded limitations may draw itself back from the complex-shaped cluster reconstruction, real cluster center identification, and large-scale dataset clustering:

1. DPC's allocation strategy may unadvisedly associate irrelevant points, triggering some irreversible "domino effect" [11].
2. DPC's decision graph is not robust in identifying real cluster centers, because it indiscriminately exhibits main density peaks (i.e., density peaks that are real cluster centers, hereinafter, main peaks) and satellite peaks (i.e., density peaks that are not real centers), which shall interfere with the selection of main peaks.
3. DPC is a time-consuming algorithm with computational complexity $O(n^2)$ [12].

Herein, a Fast Main Density Peak Clustering Within Relevant Regions Via A Robust Decision Graph (R-MDPC) is proposed to easily determine real cluster centers and accurately reconstruct complex-shaped clusters. The main contributions of R-MDPC are as follows:

☆ The source code of this paper is available at https://github.com/Guanjunyi/R-MDPC.

* Corresponding author.
E-mail addresses: guanjy@zjut.edu.cn (J. Guan), shengli@zjut.edu.cn (S. Li), 2512016@zju.edu.cn (J. Zhu), hxx@zjut.edu.cn (X. He), kecnu715@gmail.com (J. Chen).

https://doi.org/10.1016/j.patcog.2024.110458
Received 7 May 2021; Received in revised form 11 January 2024; Accepted 26 March 2024
Available online 28 March 2024

1. A relevance-based allocation strategy that only associates points within relevant regions (i.e., a region composed of relevant points) is proposed, which can effectively avoid irrelevant points being grouped together.
2. A robust decision graph that easily identifies real cluster centers by highlighting main peaks and reducing the interference of satellite peaks is obtained.
3. R-MDPC is mainly based on the kNN distance of data points, and it can run fast by applying fast kNN search technology.

The rest paper is composed as follows: Section 2 gives a brief introduction to DPC and introduces some improved works. Section 3 mainly focuses on the proposed method. Section 4 is the experiment and discussion. Section 5 ends the paper with a conclusion.

2. DPC and its improved works

2.1. The DPC algorithm

Given a dataset of n points $X = \{x_1, x_2, \dots, x_n \mid x_i \in \mathbb{R}^m\}, X \in \mathbb{R}^{m\times n}$, for each point x_i, DPC first estimates its local density ρ_{x_i} as in Eq. (1), where $d_{x_i x_j}$ is the Euclidean distance between points x_i and x_j, and "cutoff distance" d_c is a user-specified parameter. Subsequently, for point x_i (except for x_i with the highest density), DPC calculates its distance δ_{x_i} from the nearest higher density point as in Eq. (2). Then, for the highest density point x_i, DPC gives $\delta_{x_i} = \max_{x_i \neq x_j}\left(d_{x_i x_j}\right)$.

$$\rho_{x_i} = \sum_{x_j \in X} \chi(d_{x_i x_j} - d_c), \quad \chi(\Delta) = \begin{cases} 1 & \Delta < 0 \\ 0 & \Delta \geqslant 0 \end{cases} \tag{1}$$

$$\delta_{x_i} = \min_{x_j : \rho_{x_j} > \rho_{x_i}}\left(d_{x_i x_j}\right) \tag{2}$$

According to the assumption that cluster centers are density peaks that are characterized by a higher density ρ than their surrounding neighbors and by a relatively large distance δ from points with higher densities [8], cluster centers are easily located by selecting density peaks (namely points with large ρ-δ) in a decision graph (i.e., a plot of δ_{x_i} as a function of ρ_{x_i} for each point x_i). After cluster centers are selected and given unique labels, each of the remaining points is allowed to inherit the label of its nearest higher density point. Once each point obtains its label, clustering completes.

DPC's contribution is remarkable due to its capacity to locate cluster centers without prior knowledge, however, in addition to high computational complexity, its drawbacks are also obvious.

2.1.1. Unreliability of allocation strategy

Consider $O_r(x_i)$ as a m-dimensional sphere with a radius of r and a center of point x_i, as in Eq. (3). For point x_i, we introduce the concept of "domain", (denoted as D_{x_i}), namely, the maximum sphere $O_r(i)$ that makes point x_i be the density maximum within it, as in Eq. (4), where $\hat{r}$ is the domain radius of point x_i.

$$O_r(x_i) = \left\{ x_z \in \mathbb{R}^m \;\middle|\; d^2_{x_i x_z} \leqslant r^2 \right\} \tag{3}$$

$$D_{x_i} = O_{\hat{r}}(x_i), \ \hat{r} = \underset{r:\max_{x_j \in O_r(x_i)}(\rho_{x_j}) = \rho_{x_i}}{\arg\max}(r) \tag{4}$$

According to DPC's allocation strategy, for point x_i, its directly associated point x_j (the nearest higher density point) should be outside its domain D_{x_i}, since $\rho_{x_j} > \rho_{x_i}$. Note that, the size of D_{x_i} is solely determined by the unknown density distribution surrounding x_i. In other words, the size of D_{x_i} is uncertain. Therefore, there may be some large D_{x_i} that is far beyond the local area of x_i and even cover some points in other clusters. In such case, some nearest higher density points outside the D_{x_i} may fall in other clusters. So point x_i may be associated with the nearest higher density point of another cluster, thereby, leading to a misclassification.

Fig. 1(a) illustrates the limitation of DPC's allocation strategy in dealing with the classic *Jain* dataset [13], with two crescent-shaped clusters. As shown, point x_a within the upper-side cluster has a large domain D_{x_a} that extends far beyond its local area and even covers some points in the under-side cluster. According to DPC's allocation strategy, x_a must be associated with the nearest higher-density point x_b within the under-side cluster, leading to the misclassification of point x_a. Besides, points in the upper-side cluster that inherit the label of x_a are also misclassified. Therefore, DPC's allocation strategy may unadvisedly associate irrelevant points, leading to some unpleasant chain errors.

2.1.2. Unreliability of decision graph

DPC's decision graph can be considered as a detector for density peaks with the top largest γ values in global, where $\gamma = \rho \times \delta$. This is unreliable, because a cluster is usually a local area distributed on the dataset, so it can only guarantee its cluster center has the largest γ value within its local area, rather than in global. It is unreasonable for the decision graph to detect cluster centers by globally comparing γ values, since there may be some cluster centers with local maximum γ values but are not conspicuous in global.

Fig. 1(b) presents DPC's decision graph on the *Jain* dataset. As shown, cluster center (x_c) of the upper-side sparse cluster all fall in the left-side of the decision graph due to the low density. In contrast, point x_e with a large γ value falls in the intuitive cluster center area. Consequently, point x_d and point x_e with the largest global γ are selected as cluster centers, while the upper-side cluster has no cluster center, leading to a poor clustering result, see "result i" in Fig. 1(c). However, even if the cluster center is selected correctly, DPC will still cause incorrect allocation due to the limitation of its allocation strategy, see "result ii" Fig. 1(d).

2.2. Improved works

Numerous methods have been proposed to improve DPC. To improve the allocation strategy, Xie, et al. [11] designed a kNN-based allocation strategy that assigns non-center points within their k nearest neighbors; Liu, et al. [14] proposed a shared-nearest-neighbor-based allocation strategy that analyzes the association between points via counting their shared neighbors; Pizzagalli, et al. [7] proposed a shortest-path-based allocation strategy that associates points according to the properties of paths between them; Guo, et al. [15] designed a graph-based allocation strategy for local centers with estimating the connectivity information between local centers; Ding, et al. [16] employed the variance between points to improve the allocation strategy. These improved allocation strategies are more powerful in reconstructing complex-shaped clusters, but they are still time-consuming.

To improve the decision graph, Du, et al. [17] replaced density with kNN-based density. This reduces the density difference between density peaks within sparse and dense clusters, so density peaks of sparse clusters can stand out in the decision graph. Liu, et al. [14] proposed a new definition of distance δ by considering the information of the shared-nearest neighbors, which gives density peaks within sparser clusters larger δ values to make them conspicuous in the decision graph. However, similar to DPC's decision graph, density peaks are indiscriminately shown, that is to say, main peaks and satellite peaks matter equally in these decision graphs, hindering the correct selection of main peaks.

Some other works were developed to speed up DPC to improve its suitability for large-scale data. In [18], a density-grid-based clustering is proposed to reduce computational complexity by gridding; Bai et al. [19] improved the speed of DPC by combining K-means; Xu et al. [20] proposed a fast density peaks clustering algorithm based on pre-screening; Chen et al. [12] applied the cover tree algorithm [21] to fast calculate density and distance with a lower complexity $O(n\log(n))$. However, the clustering accuracy of these speed-up methods is similarly unsatisfying as DPC.

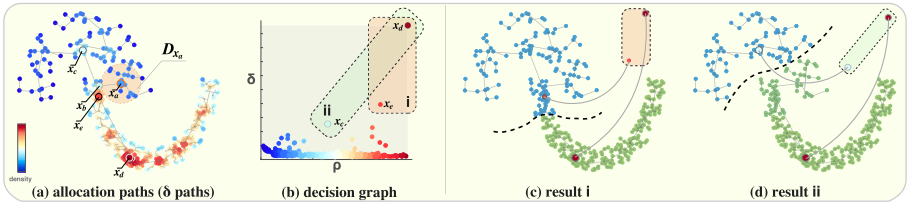

Fig. 1. The limitations of DPC on the *Jain* dataset.

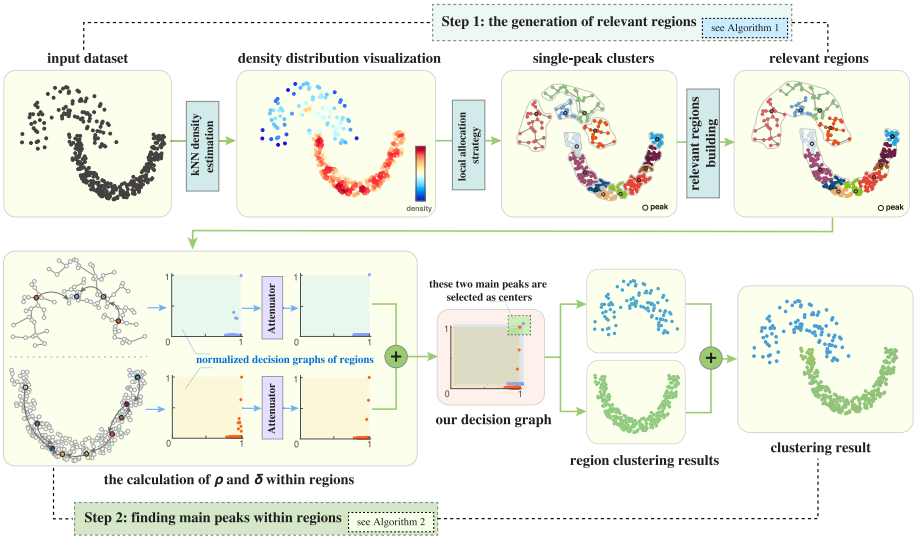

Fig. 2. The clustering process of the proposed R-MDPC algorithm.

Some methods also aim to abandon the decision graph by automatically merging sub-clusters into clusters according to a special merge threshold [22–24] or by setting the number of cluster centers in advance [25]. Here we mainly focus on the improvement of decision-graph-based DPC.

3. The R-MDPC algorithm

This section provides a detailed introduction to R-MDPC. Fig. 2 presents its flow chart: (1) the generation of relevant regions; (2) the finding of main peaks within relevant regions.

3.1. The generation of relevant regions

To avoid the association of irrelevant points like DPC's allocation strategy, we introduce a concept of "relevant region" as a constraint condition. A relevant region is a density-connected area with multiple (or a single) single-peak clusters (i.e., a cluster with only one density peak [26]). Relevant regions can be obtained by: first, estimating the local density of data points and identifying the single-peak clusters; then, merging density-connected single-peak clusters into relevant regions.

3.1.1. The identification of single-peak clusters

A single-peak cluster is a density area led by one density peak [27]. So, to generate single-peak clusters, we need to detect density peaks, and then, associate non-peak points to density peaks. We give clear definitions of density peak and single-peak cluster as in Definitions 1 and 2.

Definition 1. A point x_i is a density peak, denoted as $p \in P$, if it possesses the highest density within its neighborhood $N_k(x_i)$, i.e., $\rho_{x_i} > \max_{x_j \in N_k(x_i)}(\rho_{x_j})$. P is a set of all density peaks of X.

Definition 2. A single-peak cluster, denoted as S, consists of one density peak, and all non-peak points associated with that density peak. Herein, $S(p)$ represents a single-peak cluster leading by density peak p.

To reduce computational complexity, we employ a kNN-based density estimation method to calculate the local density of the data, as in Eq. (5), where parameter λ is employed to control the density distribution.

$$\rho_{x_i} = \left(\sum_{x_j \in N_k(x_i)} d^{\lambda}_{x_i x_j} \right)^{-1} \tag{5}$$

As mentioned in Section 2.1.1, DPC may directly associate two irrelevant points that are not in the same local area. To avoid such incident, we employ the local density peak clustering [26]—*each point x_i is only allowed to be associated with the nearest higher density neighbor within its neighborhood $N_k(x_i)$*, as our local allocation strategy. The local allocation strategy links all non-peak points with density peaks, while density peaks remain unassociated with each other due to the absence of higher-density neighbors within their neighborhoods. Then, single-peak clusters are generated.

3.1.2. Merging of single-peak clusters

A relevant region (hereinafter, a region, denoted as Ω) is a density-connected area composed of connected single-peak clusters. The "connectivity" between single-peak clusters is defined as:

Definition 3. For single-peak clusters $S(p_a)$ and $S(p_b)$, if $\exists x_j \in S(p_b), x_i \in N_{k_b}(x_j), x_j \in N_{k_b}(x_i)$, then, $S(p_a)$ and $S(p_b)$ are directly connected, denoted as $S(p_a) \leftrightarrow S(p_b)$. This is a chain process, e.g., if $S(p_a) \leftrightarrow S(p_b)$, $S(p_c) \leftrightarrow S(p_b)$, then, $S(p_a)$ and $S(p_c)$ are indirectly connected, denoted as $S(p_a) \nleftrightarrow S(p_c)$.

where a small-value $k_b = \lceil \frac{k}{2} \rceil$ is introduced to effectively detect the mutual proximity between intersecting single-peak clusters, and $\lceil \cdot \rceil$ is a round upper function. Therefore, a region Ω is defined as in:

Definition 4. A relevant region Ω is composed of n_Ω connected single-peak clusters, i.e., $\Omega = \left\{ S_1, S_2, \dots, S_{n_\Omega} \right\}, \forall S_i, S_j \in \Omega, S_i \leftrightarrow S_j$ (or $S_i \leftrightsquigarrow S_j$).

Thus, by merging connected single-peak clusters, the relevant regions are generated.

3.2. Finding main peaks within relevant regions

After obtaining relevant regions, clustering is performed by finding main peaks within relevant regions. Because a cluster should not be across different relevant regions (i.e., to own irrelevant points). Thus, the cluster center detection turns into the detection of density peaks within relevant regions.

3.2.1. Regional density normalization

As discussed in Section 2.1.2, DPC's decision graph consistently directs the selection of density peaks in high-density regions as cluster centers, inadvertently leaving out some potential cluster centers in low-density regions. This limitation arises from the density differences between regions. So, to thoroughly eliminate the differences of points among different regions, we normalize ρ values of each relevant region, called the regional density normalization:

$$\rho_{x_i}^{(1)} = \frac{\rho_{x_i}}{\max_{x_j \in \Omega}(\rho_{x_j})}, x_i \in \Omega \tag{6}$$

By this, cluster centers in low-density relevant regions can be more intuitively seen in the decision graph.

3.2.2. The δ-calculation of density peaks

According to the center assumption of DPC [8], the "density peak" attribute is a necessary but not sufficient condition for a cluster center. That is to say, not all density peaks can be selected as cluster centers, while all non-peak points will never be cluster centers. To better distinguish density peaks that can represent cluster centers from those that cannot, we define "main peak" and "satellite peak" [27], as in:

Definition 5. Given a cluster Cl, if density peak $p \in Cl$ own the highest density, then, p is the main peak of Cl, i.e., $\rho_p = \max_{p' \in Cl}(\rho_{p'})$, otherwise, p is a satellite peak.

To detect density peaks within relevant regions, for each density peak p_i (except for p_i with the highest density in its region), we define its distance δ_{p_i} from the nearest higher density point within its own region, as in Eq. (7).

$$\delta_{p_i} = \min_{p_i, p_i \in \Omega,\ \rho_{p_j} > \rho_{p_i}} \left(d_{p_i p_j} \right) \tag{7}$$

Then, for every highest density peak p_i within its regions, we give $\delta_{p_i} = \frac{3}{2} \times \max_{p: \rho_p \neq \varnothing}(\delta_p)$. Conversely, for non-peak points that are not selected as centers, δ values are set to 0 (i.e., $\delta_{x_i} = 0$ if $x_i \notin P$), which effectively eliminates interference from non-peak centers during cluster center selection.

3.2.3. The satellite peak attenuator

To further reduce the interference from satellite peaks and highlight the significance of main peaks in the decision graph, we introduce a satellite peak attenuator, according to:

Assumption 1. Unlike main-peak clusters (composed of a single-peak cluster led by a main peak), satellite-peak clusters (composed of a single-peak cluster led by a satellite peak) are more prone to sharing high-density borders with surrounding higher-density single-peak clusters.

Discussion. Consider a cluster composed of one main-peak cluster and several satellite-peak clusters. Within the cluster, the main peak has no higher single-peak clusters surrounding it, while a satellite-peak cluster may have one or more. Therefore, compared to the main-peak cluster, a satellite-peak cluster is more likely to share high-density border points with surrounding higher single-peak clusters.

For a single-peak cluster $S(p_i)$, its surrounding higher-density single-peak clusters (denoted as $H(S(p_i))$ is defined in Eq. (8).

$$H(S(p_i)) = \left\{ S(p_j) \middle| S(p_j) \leftrightarrow S(p_i), \rho_{p_j} > \rho_{p_i}, p_i, p_j \in P \right\} \tag{8}$$

Then, the ascent-border points of $S(p_i)$ (denoted as $\hat{B}S(p_i)$), i.e., the border points between $S(p_i)$ and its surrounding higher-density single-peak clusters $HS(p_i)$, is defined as Eq. (9).

$$\hat{B}(S(p_i)) = N_{k_b}(S(p_i)) \cap N_{k_b}(H(S(p_i))) \tag{9}$$

where, $N_{k_b}(S(p_i))$ indicates the set of k_b nearest neighbors of all points in cluster $S(p_i)$, i.e., $N_{k_b}(S(p_i)) = \bigcup_{x \in S(p_i)} N_{k_b}(x)$.

On this basis, we introduce "the highest ascent-border peak ratio", i.e., the ratio of a density peak p_i's highest ascent-border density to its density, denoted as ϕ_{p_i}, as in Eq. (10). While, for $\hat{B}(S(p_i)) = \varnothing$, we give $\phi_{p_i} = 0$.

$$\phi_{p_i} = \frac{1}{\rho_{p_i}} \times \max_{x: x \in \hat{B}(p_i), x \notin S(p_i)} (\rho_x), \quad \exists \hat{B}(S(p_i)) \neq \varnothing \tag{10}$$

Assumption 1 implies that satellite peaks are usually easier to obtain larger ascent-border peak ratio ϕ than main peaks. As shown in Fig. 3, two clusters are composed of three density peaks (main peak G, satellite peak L, and main peak Q), where main peak G shares its ascent-border point J with the higher-density satellite peak L, satellite peak L shares its ascent-border point O with the higher-density main peak Q, while main peak Q with density maximum has no other higher-density peak to share its border point. According to Eq. (10), we have: $\phi_L = \frac{d_{OP}}{d_{LM}} > \phi_G = \frac{d_{JI}}{d_{GH}} > \phi_Q = 0$. This example verifies that a satellite peak can usually obtain a relatively larger ascent-border peak ratio ϕ value than a main peak, since it is usually close to higher-density peaks. While a main peak has to find one beyond its cluster, which may lead to a relatively low-density border to obtain a small ascent-border peak ratio.

Based on this, for each density peak p_i, we design a satellite peak attenuator (as in Eq. (11)), where, function "$\max(\cdot)$" is used to ensure final $\delta_{p_i}^{(1)} \geqslant 0$.

$$\delta_{p_i}^{(1)} = \delta_{p_i} \times \max(0, 1 - \phi_{p_i}) \tag{11}$$

The satellite peak attenuator can reduce the distance values of satellite peaks with greater probability while affecting main peaks less, thereby, reducing the interference of satellite peaks and highlighting main peaks in the decision graph. For non-peak point x, we give $\delta_x^{(1)} = 0$ (since, $\delta_x = 0$ if $x \notin P$).

Notably, Eq. (11) implies that the larger difference of the ϕ values between satellite peaks and main peaks, the stronger the attenuation of satellite peaks. This helps in obtaining a more robust decision graph that strongly highlights main peaks while reducing the interference of satellite peaks powerfully.

Since the calculation of ϕ is based on the difference between border and peak density, the density distribution can directly impact ϕ value. An over-gentle density distribution will make little difference between border and peak density, so all ϕ values tend are to 1; while an over-steep density distribution will lay an opposite impact, so all ϕ values are tend to 0. In such cases, an over-small ϕ difference will make the decision graph have low performance in distinguishing between main peaks and satellite peaks. Therefore, a suitable density distribution (a large ϕ value difference between main peaks and satellite peaks) is necessary for a robust decision graph.

To obtain a suitable density distribution in dealing with various datasets, we define the gentleness degree as the Coefficient of Variation

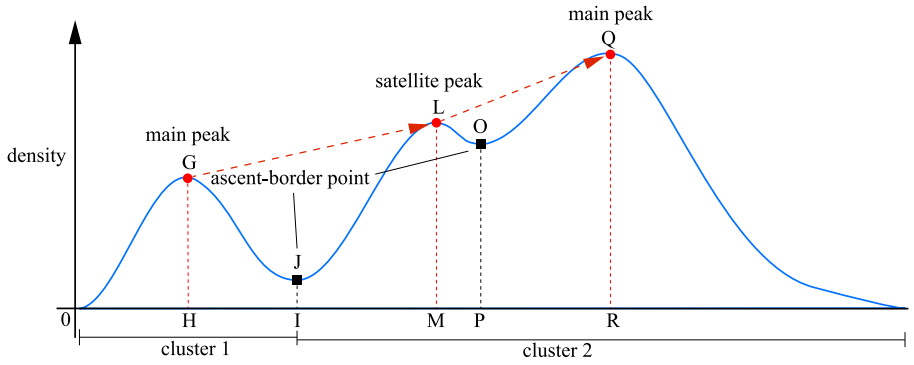

Fig. 3. An explanation for Assumption 1.

(the ratio of standard deviation to mean) [28] of the density values, as in Eq. (12), where $\bar{\rho}$ is the average density.

$$CV = \frac{1}{\bar{\rho}}\sqrt{\frac{1}{n}\times\sum_{i=1}^{n}(\rho_{x_i}-\bar{\rho})^2} \tag{12}$$

By inputting parameter cv (a CV index) and k, the density estimation (Eq. (5)) can be obtained by automatically tuning parameter $\lambda(\lambda > 0)$ until the CV is close to cv as:

$$\lambda = \underset{\lambda:\lambda>0}{\arg\min}(|CV - cv|) \tag{13}$$

As a result, R-MDPC can controllably obtain a robust decision graph by adaptively estimating density values to meet a suitable gentleness degree.

3.3. Time complexity

Algorithm 1 and 2 combine to form the pseudocode of R-MDPC, where the total number of points is set to n, the number of neighbors to k (default is $\lceil\sqrt{n}\rceil$), the CV index cv (default is 1), and the total number of single-peak clusters (density peaks) to n_p.

On the basis of the kNN distance matrix calculated by fast kNN search technology [21,29] with time complexity $O(n\log(n))$, R-MDPC performs clustering in two steps:

Step 1: the generation of relevant regions (Algorithm 1) that consists of the density calculation with $O(nt)$ (Line 1~8), the generation of single-peak clusters with $O(n\tilde{k})$ (Line 9~18), and the generation of relevant regions with $O(nk_b) = O(\frac{1}{2}nk)$ (Line 19~24). The parameter $\tilde{k}$ (an average concept) signifies that the $\tilde{k}$th neighbor is the nearest higher density point of a point. Notably, the majority of data points can pinpoint a considerably closer higher density point, i.e., $\tilde{k} \ll k$. Thus, the overall time complexity of 1 is $O(nt + n\tilde{k} + \frac{1}{2}nk) = O(nt + nk)$.

Step 2: the clustering within relevant regions (Algorithm 2) that consists of the calculation of $\rho^{(1)}$ and $\delta^{(1)}$ within relevant regions with $O(nn_p)$ (Line 1~13) and the selection of cluster centers and allocation of non-center points with $O(n)$ (Line 14~16). So, the overall time complexity of 2 $O(nn_p + n) = O(nn_p)$.

Therefore, the overall time complexity of R-MDPC is $O(n\log(n)+nt+nk+nn_p)$, where for $t =$, k, and n_p are all far less than n.

4. Experiments

4.1. Experimental set up

Comparison Algorithms: DPC [8] and six state-of-the-art DPC variant algorithms (SNN-DPC [14], FastDPeak [12], SSSP-DPC [7], FKNN-DPC [11], DPC-CE [15], DPCV [16]). In terms of the parameter setting, we choose the most effective settings from a complete spectrum of possible configurations. To ensure a fair evaluation, we adopt the top-γ method to select centers in the decision graph, i.e., to select NC density peaks with the highest γ values as centers (NC represents the true number of clusters).

Algorithm 1 R-MDPC: the generation of relevant regions

Input: dataset X, k nearest neighbor information N_k, and CV index cv.
Output: relevant regions Ω, single-peak clusters S, density peaks P, and density ρ.

1: $\lambda \leftarrow 0$
2: $t \leftarrow 0$ // t is used to record the number of iterations of density calculation.
3: **while** $CV \geqslant cv$ **do**
4: $\lambda \leftarrow \lambda + 0.1$
5: $t \leftarrow t + 1$
6: **for** each point $x \in X$ **do**
7: calculate ρ_x // Eq. (5)
8: **end for**
9: calculate CV // Eq. (12)
10: **end while**
11: obtain $\rho = \{\rho_1, \rho_2, ..., \rho_n\}$
12: **for** each point $x \in X$ **do**
13: **for** $u = 1$ up to k **do**
14: **if** $\rho_x < \rho_{x^u}$ // x^u represents point x's uth nearest neighbor. **then**
15: points x and x^u are in the same single-peak cluster.
16: **break**
17: **end if**
18: **end for**
19: **end for**
20: obtain $S = \{S_1, S_2, ..., S_{n_p}\}$ and $P = \{p_1, p_2, ..., p_{n_p}\}$ // n_p: the numbers of density peaks.
21: **for** each pair of single-peak clusters $S_a, S_b \in S$ **do**
22: **if** S_a and S_b is connected according to Definition 3 **then**
23: S_a and S_b are in the same relevant region
24: **end if**
25: **end for**
26: **return** $\Omega = \{\Omega_1, \Omega_2, ..., \Omega_{n_\Omega}\}$ // n_Ω means the number of relevant regions generated

Algorithm 2 R-MDPC: finding main peaks within relevant regions.

Input: dataset X, k nearest neighbor information N_k, relevant region set Ω, single-peak cluster set S, density peak set P, and density set ρ
Output: clustering result

1: **for** each point $x \in X$ **do**
2: $\delta_x = 0$
3: **end for**
4: **for** each density peak $p \in P$ **do**
5: calculate δ_p // Eq. (7)
6: **end for**
7: obtain $\delta = \{\delta_{x_1}, \delta_{x_2}, ..., \delta_{x_n}\}$
8: **for** each point $x \in X$ **do**
9: calculate $\rho_{x_i}^{(1)}$ // Eq. (6)
10: **end for**
11: **for** each density peak $p \in P$ **do**
12: calculate $\delta_p^{(1)}$ // Eq. (11)
13: **end for**
14: select cluster centers in the decision graph.
15: associate each non-center point with its nearest higher density point within its own relevant region.
16: **return** clustering result

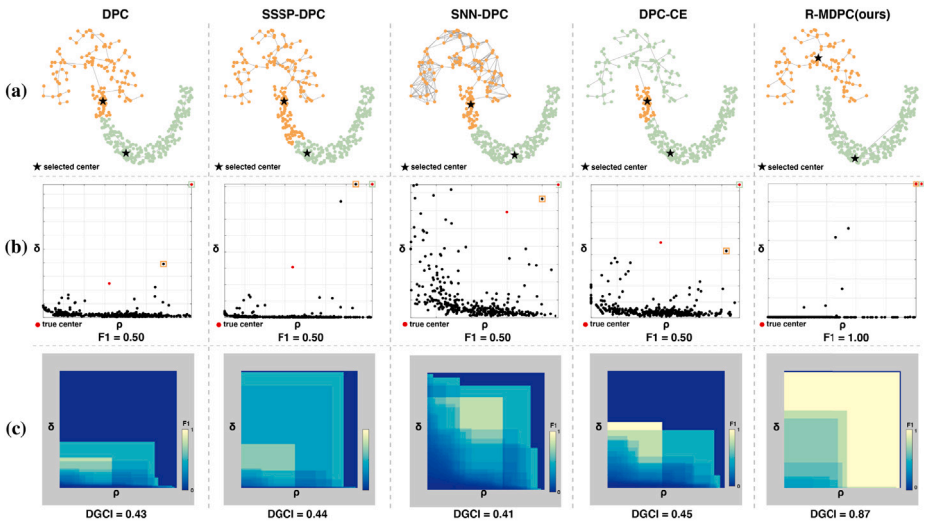

Fig. 4. The clustering results on the *Jain* dataset by different DPC-based algorithm.

Table 1
Datasets.

Dataset	Instances	Attributes	Clusters	Source
Agg	788	2	7	[13]
Jain	373	2	2	[13]
Lineblobs	266	2	3	[30]
R15	600	2	15	[13]
Flame	240	2	3	[13]
S3	5000	2	15	[13]
AggFlame	1028	2	9	[–]
Atom	800	3	2	[31]
Chainlink	1000	3	2	[31]
Iris	150	4	3	[32]
Wine	178	13	3	[32]
Seeds	210	7	3	[32]
Parkinsons	195	22	2	[32]
Ecoli	336	7	8	[32]
Movementlibras	360	90	15	[32]
Segmentation	2310	19	7	[32]
Drivepoints	606	16	4	[32]
YTF	10 000	10	41	[33]
REUTERS	10 000	10	4	[34]
MNIST	10 000	500	10	[35]
USPS	11 000	10	10	[36]
OlivettiFaces	400	92 × 112	40	[37]
Immunecells	8681	2	20	[7]

Datasets: nine different types of synthetic datasets (seven 2-dimension and two 3-dimension datasets) are selected to evaluate the performance of R-MDPC in recognizing various complex-shaped clusters and the robustness of R-MDPC's decision graph in detecting the real cluster centers. Besides, fourteen real-world datasets are selected to further evaluate the performance of R-MDPC on high-dimensional and large real-world datasets. More detailed information is listed in Table 1.

Data preprocessing: the min–max normalization method [38] is applied to normalize datasets to reduce the influence of different metrics in different dimensions.

Machine configuration: experiments are conducted by using Matlab (r2017b) on MacBook Pro with 2.9 GHz Intel Core i5, 8 G RAM.

Evaluation metric: the popular Adjusted Rand Index (ARI) [39], Adjusted Mutual Information (AMI) [39], Fowlkes–Mallows Index (FMI) [40], and Accuracy (ACC) are used to evaluate the clustering performance. Besides, the F1-score [41] is used to evaluate the center detection performance, and the F1-score-based Decision Graph Clarity Index ($DGCI$) [42] is applied to evaluate the clarity of decision graphs.

4.2. Experiments on synthetic datasets

Quantitative comparisons are conducted on nine different synthetic datasets. Figs. 4 to 12 present a comprehensive comparison of clustering results and decision graph clarity.

Fig. 4(a) illustrates that DPC, SSSP-DPC, SNN-DPC, and DPC-CE all incorrectly identified two cluster centers within the dense under-side cluster. The distribution of the upper-side cluster is excessively sparse compared to the under-side, so its low-density center failed to be included in the two points with the top highest γ values by the top-γ-method, as shown in Fig. 4(b) (the decision graphs). As a result, with no cluster center, the upper-side cluster can only be assigned to the under-side cluster, thereby, leading to a poor result.

In contrast, R-MDPC successfully identified cluster centers. It treated clusters as different regions, and by applying the regional density normalization, the density difference between regions was removed. So, the low-density center of the upper-side cluster stands out in the decision graph and is successfully selected as a cluster center.

The gray lines in Fig. 4(a) depict the data associations generated during the assignment. R-MDPC, by distinguishing regions in advance, ensures that no association between data points exists in different regions. This allows R-MDPC to effectively avoid erroneously associating unrelated data.

In Fig. 4(c), the decision graph performance is compared. As shown, R-MDPC exhibits the decision graph with the highest $DGCI$ score, indicating that identifying and selecting the correct cluster center in R-MDPC's decision graph is easier compared to other decision graphs.

Fig. 5 illustrates the clustering results on the *Agg* dataset of seven clusters. In Fig. 5(a), the top-γ method applied to DPC accurately identified six cluster centers, yet it inaccurately selected an additional center, resulting in suboptimal clustering. Although other algorithms successfully pinpointed all seven cluster centers and efficiently partitioned the dataset (SNN-DPC made a minor flaw in cluster reconstruction), R-MDPC's decision graph stands out as the clearest. It excels in guiding the selection of seven centers without prior knowledge, as depicted in Fig. 5(b). This superiority stems from R-MDPC's ability to not only eliminate non-peak points but also attenuate interference from satellite peaks. In Fig. 5(c), R-MDPC's decision graph achieved the highest *DGCI* score.

Fig. 6 presents the results on the *Lineblobs* dataset that consists of one semi-arc-shaped cluster and two square-shaped clusters. Fig. 6(b) shows that all algorithms identified the three cluster centers accurately, but only DPC made errors when reconstructing the semi-arc-shaped cluster, as shown in Fig. 6(a). Because the DPC allocation strategy does

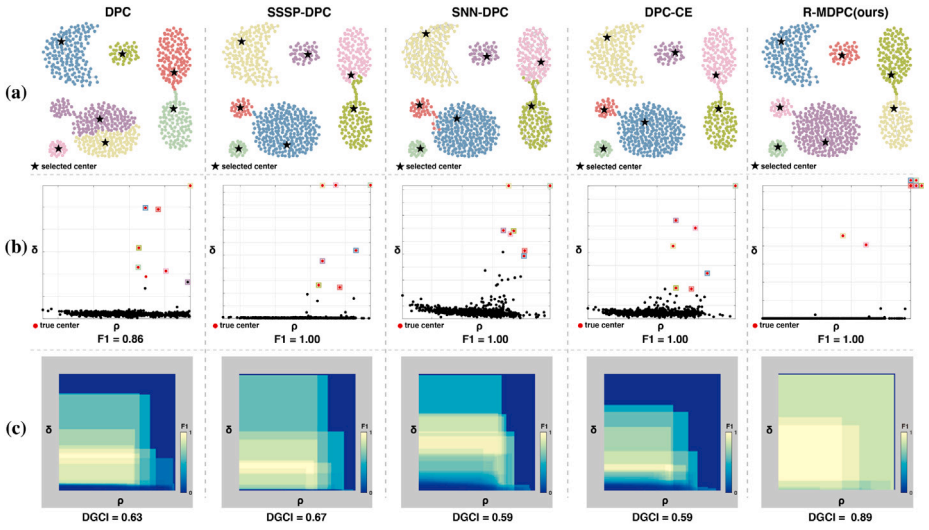

Fig. 5. The clustering results on the *Agg* dataset by different DPC-based algorithms.

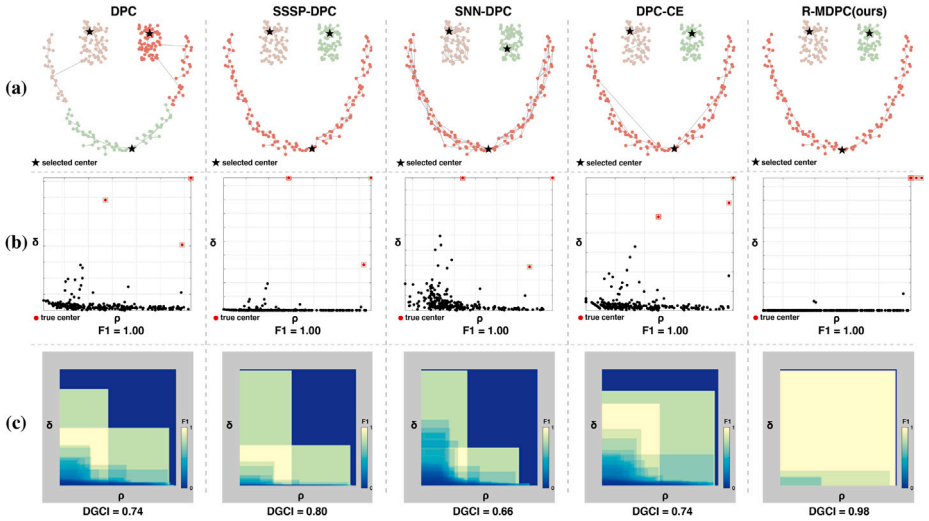

Fig. 6. The clustering results on the *Lineblobs* dataset by different DPC-based algorithm.

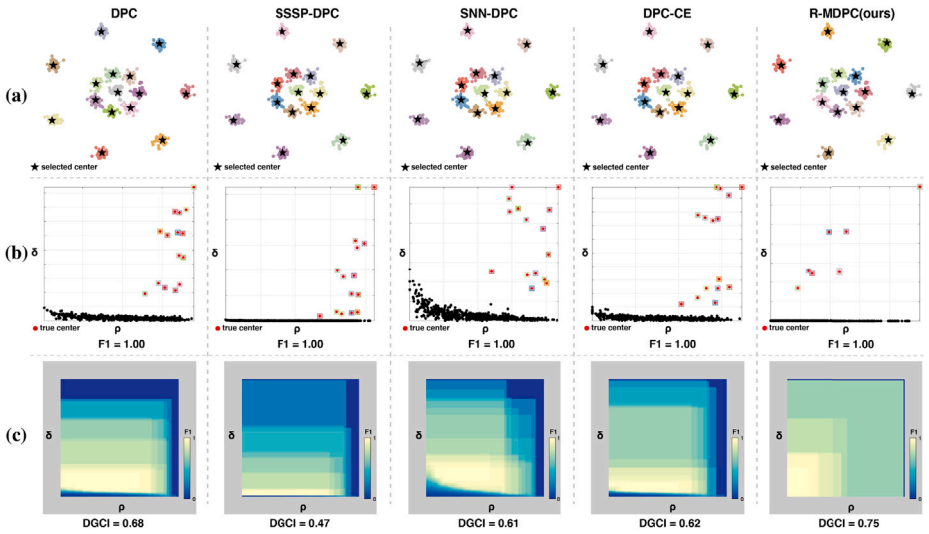

Fig. 7. The clustering results on the *R15* dataset by different DPC-based algorithm.

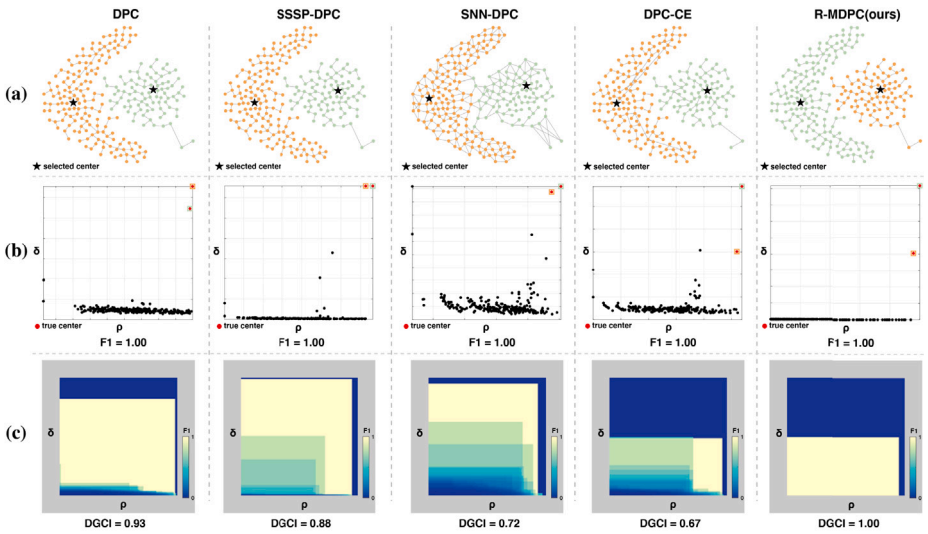

Fig. 8. The clustering results on the *Flame* dataset by different DPC-based algorithm.

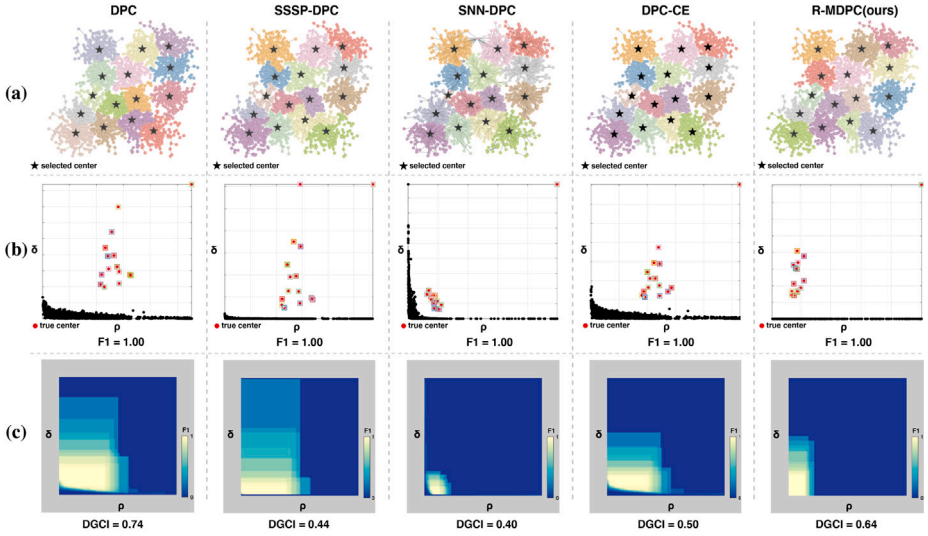

Fig. 9. The clustering results on the *S3* dataset by different DPC-based algorithm.

not constrain the association between data points, resulting in cross-cluster association phenomena. To be noted, R-MDPC's decision graph is also the clearest with the highest *DGCI* score, as in Fig. 6(c).

Fig. 7 displays the results on the *R15* dataset. The distribution of fifteen spherical clusters renders it the most straightforward dataset for all algorithms. Despite minor imperfections, each algorithm was adept at recognizing both clusters and centers, as in Fig. 7(a). Note that R-MDPC's decision graph is still the clearest and owns the highest *DGCI* score, as shown in Fig. 7(c).

Fig. 8(a) shows that all algorithms successfully partitioned the *Flame* dataset. However, compared with DPC and R-MDPC, the clarity of the decision graphs of SSSP-DPC, SNN-DPC, and DPC-CE is lower, as shown in Fig. 8(b) and (c). Note that, only the two real centers that satisfied the density peak characteristics are displayed in R-MDPC's decision graph, which makes R-MDPC's decision graph own the highest $DGCI = 1.00$. While, the decision graph of SNN-DPC, DPC-CE, and SSSP-DPC are not clear enough due to the interference of non-peak points.

Fig. 9 presents the results on the *S3* dataset composed of fifteen overlapping spherical clusters. As depicted in Fig. 9(a), all algorithms recognized both clusters and centers, despite some minor flaws. The clarity of the decision graphs is comparable, and it is easy to identify the fifteen true cluster centers, as shown in Fig. 9(b) and (c). It is worth noting that if all data points are located within one relevant region, R-MDPC and DPC adopt similar allocation strategies, as seen in datasets such as *S3* and *Flame*.

Fig. 10 presents the results on the *AggFlame* dataset. The *AggFlame*, composed by aggregating the *Agg* and *Flame*, is used to test the performance of DPC-based algorithms in identifying cluster centers when dealing with data of different distributions. As shown (see Fig. 5 and Fig. 8), all algorithms accurately identified cluster centers and effectively reconstructed cluster shapes when handling Agg and Flame datasets, respectively, with the exception of DPC misidentifying cluster centers in the *Agg* dataset.

However, when the *Agg* and *Flame* datasets were combined to form *AggFlame* dataset, only R-MDPC succeeded in identifying the true cluster centers, as shown in Fig. 10(a) and (b). Because DPC, SSSP-DPC, SNN-DPC, and DPC-CE detect cluster centers globally, however, the distribution difference between *Agg* and *Flame* datasets, makes the top-γ method fail to accurately capture the nine true cluster centers. R-MDPC addressed the distribution difference through regional density normalization and effectively diminished the impact of satellite peaks by using the satellite peak attenuator. This adjustment enables the top-γ method to easily identify the true centers.

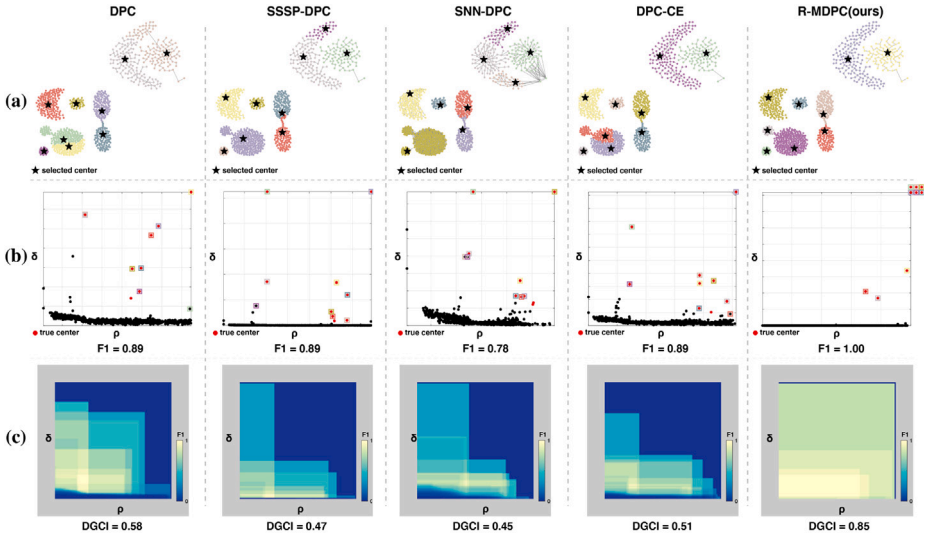

Fig. 10. The clustering results on the *AggFlame* dataset by different DPC-based algorithm.

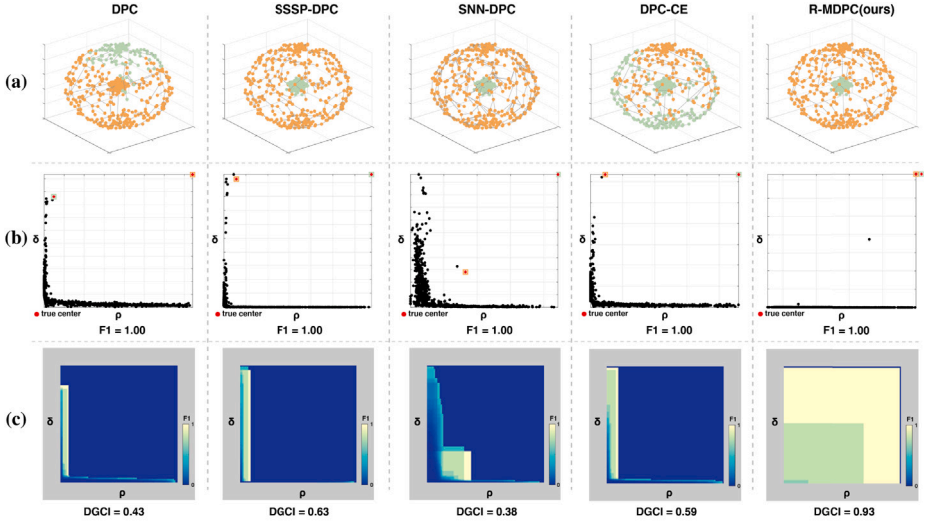

Fig. 11. The clustering results on the *Atom* dataset by different DPC-based algorithm.

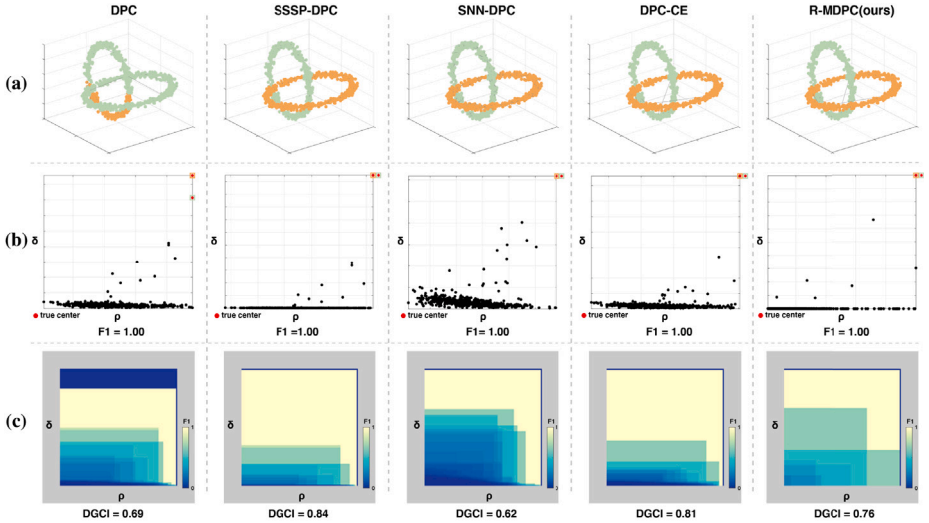

Fig. 12. The clustering results on the *Chainlink* dataset by different DPC-based algorithm.

Table 2
The comparison of AMI, ARI, FMI, and ACC on synthetic datasets.

Dataset	Metric	R-MDPC	DPC	SNN-DPC	FKNN-DPC	FastDPeak	SSSP-DPC	DPC-CE	DPCV
Agg	AMI ARI	**1.00 1.00**	0.86 0.75	0.95 0.96	0.98 0.99	0.99 0.99	0.97 0.97	0.99 0.99	0.99 0.99
	FMI ACC	**1.00 1.00**	0.81 0.95	0.97 0.98	0.99 0.99	0.99 0.99	0.98 0.98	0.99 0.99	0.99 0.99
Jian	ARI AMI	**1.00 1.00**	0.54 0.62	0.47 0.51	0.05 −0.07	0.61 0.71	0.35 0.32	0.06 −0.09	0.62 0.71
	FMI ACC	**1.00 1.00**	0.84 0.90	0.79 0.86	0.62 0.74	0.88 0.92	0.70 0.79	0.67 0.74	0.88 0.92
Lineblobs	ARI AMI	**1.00 1.00**	0.58 0.49	**1.00 1.00**	0.71 0.56	0.78 0.72	**1.00 1.00**	**1.00 1.00**	**1.00 1.00**
	FMI ACC	**1.00 1.00**	0.67 0.80	**1.00 1.00**	0.72 0.82	0.81 0.90	**1.00 1.00**	**1.00 1.00**	**1.00 1.00**
R15	ARI AMI	**0.99 0.99**	**0.99 0.99**	**0.99 0.99**	**0.99 0.99**	**0.99 0.99**	**0.99 0.99**	**0.99 0.99**	**0.99** 0.98
	FMI ACC	**0.99 0.99**	**0.99 0.99**	**0.99 0.99**	**0.99 0.99**	**0.99 0.99**	**0.99 0.99**	**0.99 0.99**	0.98 **0.99**
Falme	ARI AMI	**1.00 1.00**	**1.00 1.00**	0.91 0.95	**1.00 1.00**	**1.00 1.00**	**1.00 1.00**	**1.00 1.00**	**1.00 1.00**
	FMI ACC	**1.00 1.00**	**1.00 1.00**	0.98 0.99	**1.00 1.00**	**1.00 1.00**	**1.00 1.00**	**1.00 1.00**	**1.00 1.00**
S3	ARI AMI	**0.96 0.95**	0.94 0.93	0.88 0.83	0.80 0.67	0.94 0.92	0.88 0.83	**0.96 0.95**	**0.96 0.95**
	FMI ACC	**0.96 0.98**	0.93 0.96	0.84 0.91	0.71 0.79	0.93 0.96	0.84 0.91	0.95 **0.98**	**0.96 0.98**
AggFlame	ARI AMI	**1.00 1.00**	0.87 0.76	0.86 0.79	0.75 0.69	0.94 0.90	0.93 0.89	0.91 0.84	0.94 0.88
	FMI ACC	**1.00 1.00**	0.80 0.93	0.83 0.92	0.78 0.76	0.92 0.96	0.91 0.96	0.87 0.96	0.90 0.97
Atom	ARI AMI	**1.00 1.00**	0.16 0.09	**1.00 1.00**	0.24 0.17	0.25 0.18	**1.00 1.00**	0.32 0.26	**1.00 1.00**
	FMI ACC	**1.00 1.00**	0.65 0.65	**1.00 1.00**	0.65 0.70	0.65 0.71	**1.00 1.00**	0.67 0.75	**1.00 1.00**
Chainlink	ARI AMI	**1.00 1.00**	0.27 0.20	**1.00 1.00**	0.25 0.18	0.29 0.22	**1.00 1.00**	**1.00 1.00**	**1.00 1.00**
	FMI ACC	**1.00 1.00**	0.66 0.72	**1.00 1.00**	0.65 0.71	0.66 0.74	**1.00 1.00**	**1.00 1.00**	**1.00 1.00**

Table 3
The comparison of DGCI and F1 on synthetic datasets.

Dataset	Metric	R-MDPC	DPC	SNN-DPC	FKNN-DPC	FastDPeak	SSSP-DPC	DPC-CE	DPCV
Agg	DGCI F1	**0.89 1.00**	0.63 0.86	0.59 **1.00**	0.65 0.86	0.63 **1.00**	0.67 **1.00**	0.59 **1.00**	0.61 **1.00**
Jian	DGCI F1	**0.87 1.00**	0.43 0.50	0.41 0.50	0.46 0.50	0.42 0.50	0.44 0.50	0.45 0.50	0.40 0.50
Lineblobs	DGCI F1	**0.98 1.00**	0.74 **1.00**	0.66 **1.00**	0.76 **1.00**	0.70 **1.00**	0.80 **1.00**	0.74 **1.00**	0.73 **1.00**
R15	DGCI F1	**0.75 1.00**	0.68 **1.00**	0.61 **1.00**	0.69 **1.00**	0.61 **1.00**	0.47 **1.00**	0.62 **1.00**	0.58 **1.00**
Falme	DGCI F1	**1.00 1.00**	0.93 **1.00**	0.72 **1.00**	0.74 **1.00**	0.89 **1.00**	0.88 **1.00**	0.67 **1.00**	0.89 **1.00**
S3	DGCI F1	**0.64 1.00**	0.52 **1.00**	0.40 **1.00**	0.70 **1.00**	0.65 **1.00**	0.44 **1.00**	0.50 **1.00**	0.53 **1.00**
AggFlame	DGCI F1	**0.85 1.00**	0.58 0.89	0.45 0.78	0.53 0.89	0.57 0.89	0.47 0.89	0.51 0.89	0.60 0.89
Atom	DGCI F1	**0.93 1.00**	0.43 **1.00**	0.38 **1.00**	0.53 0.50	0.56 **1.00**	0.63 **1.00**	0.59 **1.00**	0.47 **1.00**
Chainlink	DGCI F1	0.76 **1.00**	0.69 **1.00**	0.62 **1.00**	0.64 **1.00**	0.55 **1.00**	**0.84 1.00**	0.81 **1.00**	0.75 **1.00**

R-MDPC's decision graph stands out as the clearest with the highest $DGCI$ score, as in Fig. 10(b) and (c). Such clarity not only eases the detection of true centers but also effectively guides the correct center selection. While the other decision graphs appear too ambiguous to guide accurate center selection.

Fig. 11 presents the results on the three-dimensional *Atom* dataset, consisting of a hollow spherical cluster enveloping a solid spherical cluster. As depicted in Fig. 11(a), a significant gap exists between the two clusters. Despite correctly identifying the cluster centers, both DPC and DPC-CE encountered challenges in reconstructing the hollow spherical cluster. SSSP-DPC, SNN-DPC, and R-MDPC successfully partitioned the dataset.

The decision graphs of DPC, SSSP-DPC, SNN-DPC, and DPC-CE lack the clarity required for center selection, even though the top-γ method can detect the true centers, as shown in Fig. 11(b) and (c). In comparison, R-MDPC's decision graph is clear (with the highest $DGCI$ score), facilitating the selection of true centers without the need for prior information.

Fig. 12 displays the results on the three-dimensional *Chainlink* dataset, composed of two interlocked ring-shaped clusters. As observed, only DPC struggled to reconstruct the ring shapes. Furthermore, all decision graphs are clear for center selection, and the top-γ method successfully detected the true centers for all algorithms, as in Fig. 12(b) and (c).

The excellent performance of R-MDPC has been demonstrated through the above comparison experiments. The AMI, ARI, FMI, and ACC scores in Table 2 verify R-MDPC's outstanding in recognizing datasets with clusters of various shapes. The $DGCI$ and $F1$ scores in Table 3 verify that R-MDPC's decision graph is the clearest, thus its center detection capability is outstanding.

4.3. Experiments on real-world datasets

The fourteen tested real-world datasets are composed of eight UCI datasets (the *Iris*, *Wine*, *Parinsons*, *Ecoli*, *Movementlibras*, *Segmentation*, and *Drivepoints* datasets), four large-scale datasets (the *REUTERS*, *YTF*, *USPS*, and *MNIST* datasets), the classic *OlivettiFaces* face image dataset, the *Immunecells* dataset from [7].

Table 4 shows the AMI, ARI, FMI, and ACC scores of all algorithms on these real-world datasets, and Table 5 displays the $DGCI$ and $F1$ scores of all algorithms. The AMI, ARI, FMI, and ACC scores in Table 4 verify R-MDPC's exhibits competitiveness in handling real-world datasets. The $DGCI$ and $F1$ scores in Table 5 verify that R-MDPC's decision graph clarity and center detection capability is outstanding.

4.3.1. Evaluation on UCI datasets

The *Iris*, *Wine*, *Seeds*, *Parinsons*, *Ecoli*, *Movementlibras*, *Segmentation*, and *Drivepoints* are common UCI datasets [32] that used to evaluate the clustering performance for their high dimension and structural complexity. As Table 4 shows, the overall clustering performance of R-MDPC bears favorable comparison with SNN-DPC, but the latter adopts a quite time-consuming allocation strategy.

Fig. 13 presents the t-SNE visualization results, including true labels and clustering labels assigned by R-MDPC, for eight UCI datasets. As shown, R-MDPC exhibits a tendency to capture local density-connected regions as clusters. When clusters in a dataset correspond to relatively independent density-connected regions, R-MDPC effectively identifies them, as observed in the *Iris*, *Wine*, and *Seeds* datasets.

Conversely, R-MDPC struggles to capture real clusters when they are not well represented as density-connected regions, as seen in the

Table 4
The comparison of AMI, ARI, FMI, and ACC on real-world datasets.

Dataset	Metric	R-MDPC	DPC	SNN-DPC	FKNN-DPC	FastDPeak	SSSP-DPC	DPC-CE	DPCV
Iris	ARI AMI	0.88 0.90	0.57 0.45	**0.91 0.92**	0.88 0.90	0.86 0.89	0.88 0.90	0.86 0.89	0.86 0.89
	FMI ACC	0.93 **0.97**	0.69 0.67	**0.950.97**	0.94 0.97	0.92 0.96	0.94 **0.97**	0.92 0.96	0.92 0.96
Wine	ARI AMI	0.74 0.73	0.71 0.67	**0.87 0.90**	0.80 0.80	0.74 0.73	0.75 0.74	0.51 0.48	0.72 0.70
	FMI ACC	0.82 0.90	0.78 0.88	**0.93 0.97**	0.87 0.93	0.82 0.90	0.83 0.91	0.67 0.79	0.80 0.89
Seeds	ARI AMI	0.67 0.71	0.72 0.73	**0.74 0.78**	0.70 0.74	0.67 0.71	0.66 0.69	0.71 0.74	0.71 0.74
	FMI ACC	0.81 0.89	0.82 0.90	**0.85 0.92**	0.83 0.90	0.81 0.89	0.79 0.88	0.83 0.90	0.83 0.90
Parinsons	ARI AMI	**0.25** 0.13	0.23 0.09	0.15 0.29	0.01 0.03	0.18 0.27	0.18 0.27	0.18 0.27	0.24 0.12
	FMI ACC	0.62 0.75	0.61 0.75	0.80 **0.82**	0.59 0.75	**0.81 0.82**	**0.81 0.82**	**0.81 0.82**	0.62 0.75
Ecoli	ARI AMI	0.52 0.45	0.45 0.33	**0.67 0.75**	0.47 0.54	0.63 0.70	0.49 0.35	0.53 0.42	0.54 0.47
	FMI ACC	0.58 0.81	0.49 0.75	**0.82 0.85**	0.68 0.70	0.78 0.82	0.51 0.77	0.56 0.78	0.59 0.78
Movementlibras	ARI AMI	**0.59 0.40**	0.48 0.26	0.58 0.39	0.46 0.31	0.55 0.35	0.53 0.30	0.54 0.31	0.52 0.30
	FMI ACC	**0.46 0.52**	0.35 0.45	0.45 0.53	0.39 0.41	0.42 0.48	0.37 0.49	0.37 0.48	0.36 0.49
Segmentation	ARI AMI	0.68 0.58	**0.70 0.59**	0.67 0.58	0.52 0.45	0.64 0.54	0.65 0.46	0.67 0.55	0.63 0.53
	FMI ACC	**0.66** 0.68	**0.66 0.75**	0.65 0.72	0.55 0.62	0.63 0.69	0.59 0.68	0.63 0.69	0.61 0.66
Drivepoints	ARI AMI	0.78 0.71	0.71 0.62	0.73 0.73	0.77 **0.79**	**0.79** 0.76	0.78 0.77	0.73 0.74	**0.79** 0.74
	FMI ACC	0.79 **0.85**	0.73 0.82	0.80 0.84	**0.86 0.85**	0.84 **0.85**	0.85 **0.85**	0.82 0.84	0.82 **0.85**
YTF	ARI AMI	**0.79 0.60**	0.73 0.50	0.72 0.46	0.61 0.46	0.78 0.57	0.80 0.58	0.70 0.45	0.75 0.45
	FMI ACC	**0.62 0.76**	0.52 0.62	0.50 0.63	0.48 0.58	0.59 0.69	0.60 0.73	0.48 0.62	0.48 0.68
REUTERS	ARI AMI	**0.39 0.38**	0.24 0.26	**0.39** 0.36	0.05 0.00	0.24 0.25	0.23 0.20	0.29 0.24	0.26 0.28
	FMI ACC	**0.64** 0.64	0.50 0.62	0.57 **0.68**	0.54 0.41	0.53 0.60	0.51 0.58	0.49 0.62	0.57 0.61
MNIST	ARI AMI	**0.85 0.77**	0.43 0.30	0.79 0.63	0.59 0.24	0.84 0.77	0.67 0.31	0.71 0.60	0.50 0.23
	FMI ACC	**0.80 0.84**	0.41 0.45	0.69 0.78	0.45 0.41	0.80 0.83	0.51 0.41	0.66 0.68	0.43 0.34
USPS	ARI AMI	**0.72 0.63**	0.34 0.24	0.62 0.43	0.22 0.01	0.65 0.51	0.55 0.42	0.51 0.42	0.32 0.21
	FMI ACC	**0.68 0.73**	0.37 0.39	0.51 0.63	0.29 0.22	0.57 0.67	0.57 0.47	0.51 0.49	0.34 0.39
OlivettiFaces	ARI AMI	0.82 **0.70**	0.77 0.62	0.81 0.68	0.64 0.47	**0.83 0.70**	0.81 0.68	0.76 0.59	0.79 0.64
	FMI ACC	**0.71 0.80**	0.64 0.74	0.69 **0.80**	0.50 0.62	**0.71 0.80**	0.69 0.79	0.61 0.75	0.66 0.77
Immunecells	ARI AMI	**0.93 0.86**	0.9 0.78	0.88 0.73	0.86 0.63	0.88 0.68	0.91 0.81	**0.93** 0.85	0.89 0.77
	FMI ACC	**0.87 0.97**	0.81 0.94	0.75 0.90	0.66 0.80	0.70 0.85	0.83 0.96	0.86 0.96	0.79 0.94

Table 5
The comparison of DGCI and F1 on real-world datasets.

Dataset	Metric	R-MDPC	DPC	SNN-DPC	FKNN-DPC	FastDPeak	SSSP-DPC	DPC-CE	DPCV
Iris	DGCI F1	**0.81 1.00**	0.77 0.67	0.67 **1.00**	0.67 **1.00**	0.67 0.67	0.74 0.67	0.72 **1.00**	0.69 **1.00**
Wine	DGCI F1	**0.82 1.00**	0.53 **1.00**	0.65 **1.00**	0.60 **1.00**	0.67 **1.00**	0.74 **1.00**	0.64 0.67	0.66 **1.00**
Seeds	DGCI F1	**0.85 1.00**	0.76 **1.00**	0.66 **1.00**	0.74 **1.00**	0.76 **1.00**	0.82 **1.00**	0.78 **1.00**	0.75 **1.00**
Parinsons	DGCI F1	**0.50 0.50**	0.25 **0.50**	0.31 **0.50**	0.29 **0.50**	0.20 **0.50**	0.39 **0.50**	0.40 **0.50**	0.31 **0.50**
Ecoli	DGCI F1	**0.43** 0.38	0.32 0.25	0.29 0.38	0.34 0.38	0.35 0.38	0.38 0.38	0.33 0.38	0.34 **0.50**
Movementlibras	DGCI F1	**0.46 0.64**	0.26 0.53	0.24 0.53	0.26 0.47	0.29 0.53	0.33 0.53	0.35 0.60	0.24 0.47
Segmentation	DGCI F1	**0.64 0.77**	0.49 0.71	0.26 0.86	0.42 0.71	0.39 0.71	0.53 0.71	0.49 0.71	0.48 0.71
Drivepoints	DGCI F1	0.60 0.75	0.64 0.75	0.53 0.75	0.59 0.75	0.55 0.75	**0.69** 0.75	0.67 **1.00**	0.62 0.75
YTF	DGCI F1	**0.38** 0.59	0.23 0.54	0.14 0.51	0.35 **0.63**	0.27 0.61	0.17 **0.63**	0.19 0.54	0.23 0.51
REUTERS	DGCI F1	**0.76 0.75**	0.10 **0.75**	0.25 **0.75**	0.38 **0.75**	0.33 0.50	0.16 0.50	0.14 0.50	0.10 **0.75**
MNIST	DGCI F1	**0.57** 0.80	0.10 0.60	0.13 0.50	0.27 0.60	0.33 **0.90**	0.17 0.50	0.28 0.70	0.11 0.60
USPS	DGCI F1	**0.38 0.70**	0.20 0.50	0.15 0.40	0.26 0.40	0.24 0.40	0.29 0.50	0.27 0.60	0.18 0.50
OlivettiFaces	DGCI F1	**0.91 0.90**	0.33 0.73	0.28 0.78	0.43 0.88	0.36 0.83	0.35 0.85	0.37 0.78	0.35 0.78
Immunecells	DGCI F1	**0.88 0.95**	0.38 0.90	0.45 0.85	0.42 0.80	0.44 0.85	0.28 0.90	0.51 **0.95**	0.39 0.90

Parkinsons, *Ecoli*, and *Movementlibras* datasets. Consequently, R-MDPC demonstrates poor clustering performance on the *Parkinsons*, *Ecoli*, and *Movementlibras* datasets, while achieving relatively good results on the *Iris*, *Wine*, and *Seeds* datasets. The same observations apply to other algorithms, given their shared DPC-type nature, relying on the density concept to identify density-connected regions as clusters.

4.3.2. Evaluation on large-scale real datasets

The *REUTERS* [34] dataset consists of 10,000 samples of English news stories; the *YTF* (YouTube Faces) [33] dataset contains 10,000 samples of faces; the *USPS* [36] dataset consists of 11,000 samples of handwritten digits; and the *MNIST* [35] dataset consists of 10,000 samples of labeled images of handwritten digits. In addition to the high dimension and structural complexity, the large amount of data size also adds an extra burden to clustering.

As illustrated in Table 4, R-MDPC consistently maintains its exceptional performance on large real-world datasets, achieving the highest scores in AMI, ARI, FMI, and ACC on the *YTF*, *REUTERS*, *USPS*, and *MNIST* datasets. The only exception is the second-highest ACC score on the *REUTERS* dataset. Compared with DPC and other DPC variants, the performance of R-MDPC is superior.

Notably, the results of all algorithms on the *REUTERS* dataset are as ideal as the results on the *YTF*, *USPS*, and *MNIST* datasets. To explore the reasons, we obtained the t-SNE visualization results of the four datasets as in Fig. 14. As shown, there is mutual blending among the four clusters in the *REUTERS* dataset, indicating that they are not density-connected in relatively independent spaces. This poses a challenging scenario for density-based DPC-type algorithms, as these techniques attempt to define clusters by reconstructing density-connected local density regions, as discussed in Section 4.3.1. As a result, all DPC-type algorithms produce unreasonable results in the *REUTERS* dataset. In contrast, the clusters in the *YTF*, *USPS*, and *MNIST* datasets exist in relatively independent spaces, which is advantageous for the identification by clustering techniques.

These above experiments on twelve real-world datasets verify the feasibility and wide applicability of R-MDPC in real-world datasets regardless of data size and space dimension.

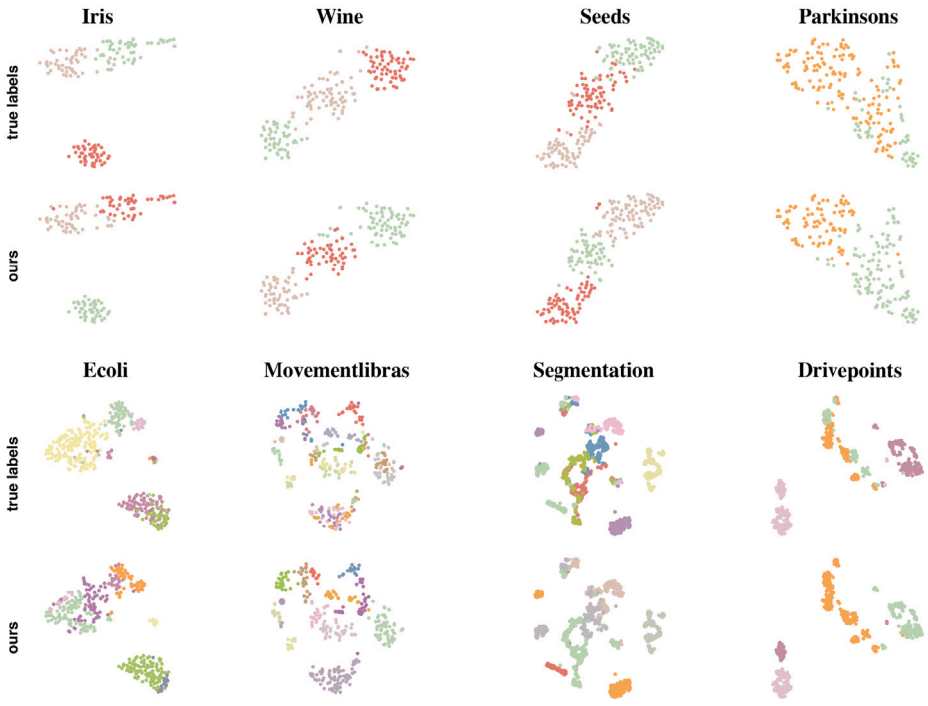

Fig. 13. The t-SNE-based visualization comparison between the true labels and our clustering labels on UCI datasets.

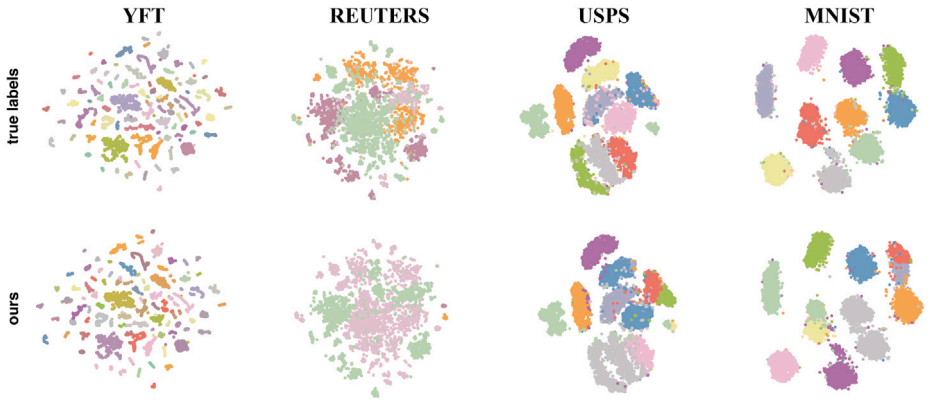

Fig. 14. The t-SNE-based visualization comparison between the true labels and our clustering labels on the *YTF*, *REUTERS*, *USPS*, and *MNIST* datasets.

Fig. 15. The recognition result of MDPC on 101 to 200 face images of the *OlivettiFace* dataset.

4.3.3. Evaluation on the OlivettiFace dataset

The *OlivettiFace* dataset [37] that contains 10 different face-angle images of 40 people is a widespread benchmark for machine learning algorithms. To accurately divide 400 face images of 40 different people is challenging, because the number of clusters (40) is comparable with the total number of elements (400) [8].

As Table 4 shows, R-MDPC obtains the highest scores of AMI, FMI, and ACC in recognizing these 400 faces. Fig. 15 presents the

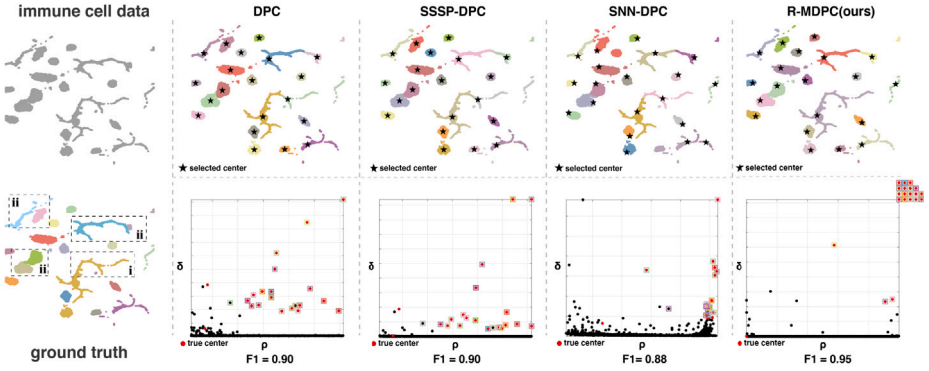

Fig. 16. The segmentation results of immune cells in a background removal microscopy image by different algorithms.

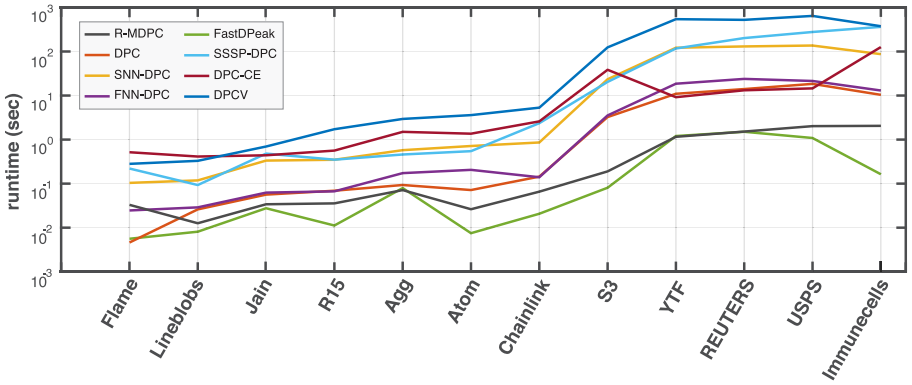

Fig. 17. The runtime of comparison algorithms on different tested datasets.

fine result of R-MDPC ($k = 5, cv = 1$) on selected 101 to 200 face images, where different colors represent different clusters, and white dots indicate the cluster centers, which verifies the superiority of R-MDPC compared with other state-of-the-art approaches.

4.3.4. Evaluation on the immunecells dataset

Microscopy images are challenging spatiotemporal biomedical datasets [43], especially images of immune cells that possibly show high plasticity regardless of the shapes (spherical, non-spherical) in contact [44], and the segmentation of immune cells naturally be regarded as a painstaking clustering task (i.e., heterogeneous shapes in proximity) [7].

Fig. 16 displays the segmentation results of various algorithms for clustering pixels of immune cells in a background removal microscopy image [7]. The immune cells exhibit two different structures: dendritic ("area i") and spherical, with some cells in contact ("area ii"). As illustrated, R-MDPC achieved the most accurate segmentation results by successfully identifying both "area i" and "area ii". In contrast, DPC, SSSP-DPC, and SNN-DPC effectively segment "area ii" but struggled with "area i". Moreover, R-MDPC also presented a relatively clear decision graph, facilitating the easier selection of twenty cluster centers.

4.4. The speed of R-MDPC

When dealing with large datasets, running speed is an important performance index. To demonstrate the fast speed of R-MDPC, speed comparisons were conducted.

In Fig. 17, R-MDPC and FastDPeak are significantly faster than other DPC-based algorithms, because these two algorithms are both based on the kNN distance of data points, rather than the distance between data points. Nevertheless, R-MDPC highlights itself in terms of clustering accuracy. SNN-DPC, SSSP-DPC, and DPCV are particularly slow due to their highly time-consuming allocation strategies, which becomes a drag on their large dataset clustering.

4.5. Evaluation of parameter sensitivity

R-MDPC has two parameters: k, the number of neighbors, and cv, the CV index of the density distribution. The performance of the allocation strategy can be adjusted by changing the parameter k. The robustness of the decision graph can be adjusted by changing the density distribution of the dataset by changing the parameter cv. What follows is a discussion on the sensitivity of parameters k and cv.

4.5.1. Insensitivity of parameter k

The k-sensitivity evaluation experiments are conducted on all the tested synthetic and UCI datasets. Fig. 18 presents several k-AMI plots and an overall k-AMI plot (with $k \in [5, 2\sqrt{n}]$) of R-MDPC ($cv = 1$), SNN-DPC, FKNN-DPC, and FastDPeak. Where k is no less than 5 because the kNN-density evaluation and the relevance definition require a sufficiently large value of k.

As shown, for R-MDPC, once parameter k reaches a certain value, it obtains a stable optimal performance within a wide range around $k = \sqrt{n}$. While for other kNN-based methods, there are still some fluctuations after reaching the optimal performance, e.g., SNN-DPC is not stable for the *Agg* and *Jain* datasets; FKNN-DPC and FastDPeak are not stable for the *Jain* and *Lineblobs* datasets. The overall k-AMI plot on all test datasets shows that R-MDPC can obtain an optimal stable performance at the range around $k = \sqrt{n}$. This demonstrates the

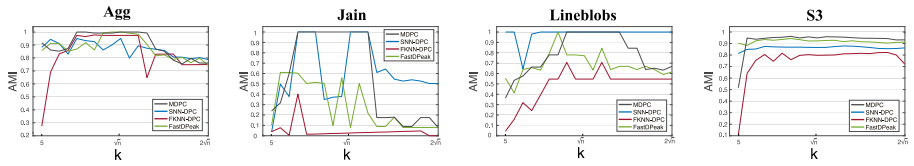

Fig. 18. The k-AMI plots (with $k \in [5, 2\sqrt{n}]$) of R-MDPC ($cv = 1$), SNN-DPC, FKNN-DPC, and FastDPeak on several datasets.

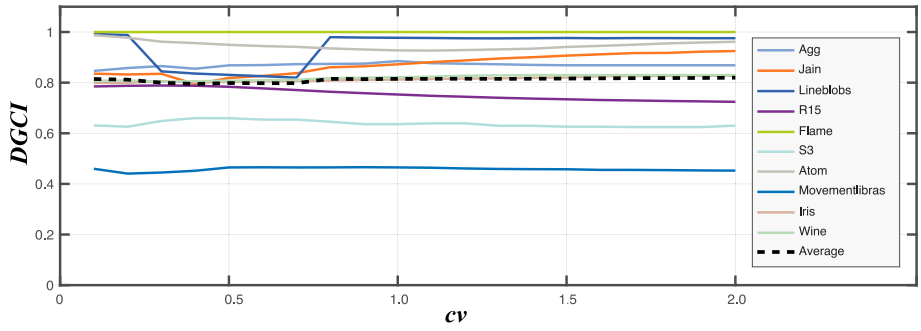

Fig. 19. The cv-$DGCI$ plot of R-MDPC on several datasets.

insensitivity of parameter k and the effectiveness of its setting range, $k = \lceil \sqrt{n} \rceil$.

4.5.2. Insensitivity of parameter cv

As mentioned in Section 3.2.3, parameter cv is used to control the gentleness degree of the density distribution of the dataset, aiming to obtain a robust decision graph. Since $DGCI$ scores is used to evaluate the robustness of decision graph [42], we conduct R-MDPC's cv-sensitivity evaluation experiments on all the synthetic and UCI datasets.

Fig. 19 presents the cv-$DGCI$ plot of R-MDPC on these test datasets. As shown, all $DGCI$ scores are smooth and stable after parameter cv approaches 1. This verifies the insensitivity of CV index cv and the effectiveness of the setting CV index $cv = 1$.

5. Conclusion

Fast Main Density Peak Clustering Within Relevant Regions Via A Robust Decision Graph (R-MDPC) inherits DPC's core idea of finding density peaks as cluster centers. R-MDPC's initial division of relevant regions of the dataset adds relevance constraints to DPC's allocation strategy, which is a simple and effective remedy for DPC's allocation strategy that may unadvisedly allocate irrelevant points together. Furthermore, R-MDPC's removal of regional differences effectively solves the problem of unbalanced density distribution caused by differences among relevant regions, which avoids low-density clusters being easily overlooked in the decision graph.

The satellite peak attenuator and the controllable density estimation method proposed by R-MDPC greatly reduce the interference of satellite peaks on the cluster center selection in the decision graph. Thus, R-MDPC owns a robust decision graph that highlights main peaks (real cluster centers).

Moreover, by applying fast kNN search technology, R-MDPC is much faster than DPC. It hardly requires distance computation except for the kNN search, which enables R-MDPC to deal with large-scale data. Numerous experiments have been launched and verified that R-MDPC can fast recognize complex-shaped clusters regardless of their spatial dimensions and data size.

However, the relevant region generation method employed by R-MDPC is not suitable for datasets with extremely high dimensions, where data points sparsely populate the entire data space. In such cases, the kNN-based region generation method tends to connect the entire dataset into a single relevant region. When a dataset forms a single relevant region, our allocation strategy, which involves finding density peaks within regions, becomes akin to DPC's process of finding density peaks across the entire dataset. Furthermore, in R-MDPC, cluster centers are manually selected. Therefore, for future work, we plan to explore the use of dimensionality reduction techniques (such as t-SNE [45] and PCA [46]) to preprocess high-dimensional datasets and design a new region generation technique that allows our method to effectively distinguish relevant regions. Additionally, we aim to develop a fully automatic and robust method for cluster center selection.

CRediT authorship contribution statement

Junyi Guan: Conceptualization, Data curation, Methodology, Software, Writing – original draft. **Sheng Li:** Conceptualization, Supervision. **Jinhui Zhu:** Supervision. **Xiongxiong He:** Supervision. **Jiajia Chen:** Writing – review & editing.

Declaration of competing interest

None.

Data availability

Data will be made available on request.

Acknowledgments

This work was supported by the "Pioneer" and "Leading Goose" R&D Program of Zhejiang (Grant:2024C04023), and the National Science Foundation of P.R. China (Grant:62233016, 62306282).

References

[1] C. Wiwie, et al., Comparing the performance of biomedical clustering methods, Nat. Methods 12 (11) (2015) 1033, http://dx.doi.org/10.1038/nmeth.3583.
[2] A.K. Jain, et al., Data clustering: a review, ACM Comput. Surv. (CSUR) 31 (3) (1999) 264–323.
[3] P. Berkhin, A survey of clustering data mining techniques, in: Grouping Multidimensional Data, 2006, pp. 25–71, http://dx.doi.org/10.1007/3-540-28349-8_2.
[4] J. MacQueen, Some methods for classification and analysis of multivariate observations, in: Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability, Vol. 1, (14) 1967, pp. 281–297.

[5] A.K. Jain, Data clustering: 50 years beyond K-means, Pattern Recognit. Lett. 31 (8) (2010) 651–666, http://dx.doi.org/10.1016/j.patrec.2009.09.011.
[6] M. Ester, H.-P. Kriegel, et al., A density-based algorithm for discovering clusters in large spatial databases with noise, in: KDD'96: Proceedings of the Second International Conference on Knowledge Discovery and Data Mining, Vol. 96, (34) 1996, 266–231.
[7] D.U. Pizzagalli, S.F. Gonzalez, et al., A trainable clustering algorithm based on shortest paths from density peaks, Sci. Adv. 5 (10) (2019) eaax3770, http://dx.doi.org/10.1126/sciadv.aax3770.
[8] A. Rodriguez, A. Laio, Clustering by fast search and find of density peaks, Science 344 (6191) (2014) 1492–1496, http://dx.doi.org/10.1126/science.1242072.
[9] M. Karaayvaz, S. Cristea, et al., Unravelling subclonal heterogeneity and aggressive disease states in TNBC through single-cell RNA-seq, Nat. Commun. 9 (1) (2018) 1–10, http://dx.doi.org/10.1038/s41467-018-06052-0.
[10] J. Liao, H. Sun, et al., Density cluster based approach for controller placement problem in large-scale software defined networkings, Comput. Netw. 112 (2017) 24–35, http://dx.doi.org/10.1016/j.comnet.2016.10.014.
[11] J. Xie, H. Gao, et al., Robust clustering by detecting density peaks and assigning points based on fuzzy weighted K-nearest neighbors, Inform. Sci. 354 (2016) 19–40, http://dx.doi.org/10.1016/j.ins.2016.03.011.
[12] Y. Chen, X. Hu, et al., Fast density peak clustering for large scale data based on kNN, Knowl.-Based Syst. 187 (2020) 104824, http://dx.doi.org/10.1016/j.knosys.2019.06.032.
[13] P. Fränti, S. Sieranoja, K-means properties on six clustering benchmark datasets, Appl. Intell. 48 (12) (2018) 4743–4759, [Online] available: http://cs.uef.fi/sipu/datasets/.
[14] R. Liu, H. Wang, X. Yu, Shared-nearest-neighbor-based clustering by fast search and find of density peaks, Inform. Sci. 450 (2018) 200–226, http://dx.doi.org/10.1016/j.ins.2018.03.031.
[15] W. Guo, W. Wang, et al., Density Peak Clustering with connectivity estimation, Knowl.-Based Syst. 243 (2022) 108501, http://dx.doi.org/10.1016/j.knosys.2022.108501.
[16] S. Ding, et al., Density peaks clustering algorithm based on improved similarity and allocation strategy, Int. J. Mach. Learn. Cybern. 14 (2023) 1527–1542, http://dx.doi.org/10.1007/s13042-022-01711-7.
[17] M. Du, S. Ding, H. Jia, Study on density peaks clustering based on k-nearest neighbors and principal component analysis, Knowl. Based Syst. 99 (2016) 135–145, http://dx.doi.org/10.1016/j.knosys.2016.02.001.
[18] B. Wu, B.M. Wilamowski, A fast density and grid based clustering method for data with arbitrary shapes and noise, IEEE Trans. Ind. Inform. 13 (4) (2016) 1620–1628, http://dx.doi.org/10.1109/TII.2016.2628747.
[19] L. Bai, X. Cheng, J. Liang, et al., Fast density clustering strategies based on the k-means algorithm, Pattern Recognit. 71 (2017) 375–386, http://dx.doi.org/10.1016/j.patcog.2017.06.023.
[20] X. Xu, S. Ding, T. Sun., A fast density peaks clustering algorithm based on pre-screening, in: 2018 IEEE International Conference on Big Data and Smart Computing, 2018, http://dx.doi.org/10.1109/BigComp.2018.00084.
[21] A. Beygelzimer, S. Kakade, J. Langford, Cover trees for nearest neighbor, in: Proceedings of the 23rd International Conference on Machine Learning, 2006, pp. 97–104, http://dx.doi.org/10.1145/1143844.1143857.
[22] J. Jiang, X. Tao, K. Li, DFC: Density fragment clustering without peaks, J. Intell. Fuzzy Systems 34 (1) (2018) 525–536, http://dx.doi.org/10.3233/JIFS-17678.
[23] M. Parmar, W. Peng, et al., FREDPC: A feasible residual error-based density peak clustering algorithm with the fragment merging strategy, IEEE Access 7 (2019) 89789–89804, http://dx.doi.org/10.1109/ACCESS.2019.2926579.
[24] F. Fang, et al., Adaptive core fusion-based density peak clustering for complex data with arbitrary shapes and densities, Pattern Recognit. 107 (2020) 107452, http://dx.doi.org/10.1016/j.patcog.2020.107452.
[25] G. Wang, Y. Wei, Clustering by defining and merging candidates of cluster centers via independence and affinity, Neurocomputing 315 (2018) 486–495, http://dx.doi.org/10.1016/j.neucom.2018.07.043.
[26] J. Guan, S. Li, X. He, et al., SMMP: A stable-membership-based auto-tuning multi-peak clustering algorithm, IEEE Trans. Pattern Anal. Mach. Intell. 45 (5) (2022) 6307–6319, http://dx.doi.org/10.1109/TPAMI.2022.3213574.
[27] J. Guan, S. Li, X. He, J. Chen, Clustering by fast detection of main density peaks within a peak digraph, Inform. Sci. 628 (2023) 504–521, http://dx.doi.org/10.1016/j.ins.2023.01.144.
[28] H. Abdi, Coefficient of variation, in: Encyclopedia of Research Design, Vol. 1, 2010, pp. 169–171, http://dx.doi.org/10.1002/0471667196.ess0371.
[29] J. Friedman, et al., An algorithm for finding best matches in logarithmic expected time, ACM Trans. Math. Softw. 3 (3) (1977) 209–226, http://dx.doi.org/10.1145/355744.355745.
[30] L. Zelnik-manor, P. Perona, Self-tuning spectral clustering, in: Neural Information Processing Systems, 2004, pp. 1601–1608.
[31] M.C. Thrun, A. Ultsch, Clustering benchmark datasets exploiting the fundamental clustering problems, Data Brief 30 (2020) 105501.
[32] K. Bache, M. Lichman, UCI machine learning repository, 2013, URL: http://archive.ics.uci.edu/ml.
[33] L. Wolf, et al., Face recognition in unconstrained videos with matched background similarity, in: Computer Vision and Pattern Recognition, CVPR, 2011.
[34] D.D. Lewis, et al., Rcv1: A new benchmark collection for text categorization research, J. Mach. Learn. Res. 5 (2004) 361–397, http://dx.doi.org/10.1023/B:JODS.0000024125.05337.9e.
[35] Y. LeCun, C. Cortes, MNIST handwritten digit database, 2010, URL: http://yann.lecun.com/exdb/mnist/.
[36] D. Keysers, et al., Deformation models for image recognition, IEEE Trans. Pattern Anal. Mach. Intell. 29 (8) (2007) 1422–1435.
[37] F. Samaria, A. Harter, Parameterisation of a stochastic model for human face identification, in: Proceedings of 1994 IEEE Workshop on Applications of Computer Vision, IEEE, 1994, pp. 138–142.
[38] P. Franti, O. Virmajoki, Iterative shrinking method for clustering problems, Pattern Recognit. 39 (5) (2006) 761–775.
[39] N.X. Vinh, J. Epps, J. Bailey, Information theoretic measures for clusterings comparison: Variants, properties, normalization and correction for chance, J. Mach. Learn. Res. 11 (2010) 2837–2854.
[40] E.B. Fowlkes, C.L. Mallows, A method for comparing two hierarchical clusterings, J. Am. Stat. Assoc. (1983) 553–569.
[41] A. Patil, D. Huard, C.J. Fonnesbeck, PyMC: Bayesian stochastic modelling in Python, J. Stat. Softw. 35 (4) (2010).
[42] J. Guan, S. Li, X. Chen, et al., DEMOS: Clustering by pruning a density-boosting cluster tree of density mounts, IEEE Trans. Knowl. Data Eng. 35 (10) (2023) 10814–10830, http://dx.doi.org/10.1109/TKDE.2023.3266451.
[43] V. Ulman, et al., An objective comparison of cell tracking algorithms, Nature Methods 14 (12) (2017) 1141–1152, http://dx.doi.org/10.1038/nmeth.4473.
[44] D.U. Pizzagalli, et al., Leukocyte Tracking Database, a collection of immune cell tracks from intravital 2-photon microscopy videos, Sci. Data 5 (1) (2018) 180129, http://dx.doi.org/10.1038/sdata.2018.129.
[45] V. der Maaten, et al., Visualizing data using t-SNE, J. Mach. Learn. Res. 9 (11) (2008) 789–798.
[46] J. Yang, et al., Two-dimensional PCA: a new approach to appearance-based face representation and recognition, IEEE Trans. Pattern Anal. Mach. Intell. 26 (1) (2004) 131–137, http://dx.doi.org/10.1109/TPAMI.2004.1261097.

Junyi Guan received a Ph.D. degree from the Zhejiang University of Technology (ZJUT), Hangzhou, China. He is currently working toward a post-doctoral degree with ZJUT. His current research interests include data mining, pattern recognition, unsupervised learning, and machine learning.

Sheng Li Ph.D. in electronic engineering, University of York, York, U.K. Associate professor of ZJUT. His research interests include signal processing, machine learning, and pattern recognition.

Jinhui Zhu Ph.D. in surgery, Zhejiang Chinese Medical University, Hangzhou, China. Chief Physician of Second Affiliated Hospital, Zhejiang University School of Medicine. His research interests include bioinformatics and pattern recognition.

Xiongxiong He received Ph.D. in Zhejiang University, Hangzhou, China. Professor of ZJUT. His research areas include nonlinear control, signal processing, and pattern recognition.

Jiajia Chen received M.A. in East China Normal University, Shanghai, China. Her current research interests include data mining and pattern recognition.