Information Sciences 628 (2023) 504–521

Contents lists available at ScienceDirect

Information Sciences

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

Clustering by fast detection of main density peaks within a peak digraph ☆

Junyi Guan, Sheng Li *, Xiongxiong He, Jiajia Chen

College of Information Engineering, Zhejiang University of Technology, Hangzhou 310023, China

ARTICLE INFO

Keywords:
Density-based clustering
Density peak
Cluster center detection
Arbitrary shape clustering

ABSTRACT

The well-known Density Peak Clustering algorithm (DPC) proposed a heuristic center detection idea, i.e., to find density peaks as cluster centers. Nevertheless, such a center detection idea cannot work well on multi-peak clusters of complex shapes. Besides, DPC needs the distances between data, making it prohibitively time-consuming. To overcome these problems, a Main Density Peak Clustering algorithm (MDPC +)—clustering by fast detection of main density peaks within a peak digraph—is proposed, where a main density peak is the highest density peak in a cluster. MDPC+ can easily detect the real centers of multi-peak clusters based on its new center assumption. In MDPC+ , the clustering problem is viewed as a graph cut problem and a specific graph structure is designed for non-peak and density peak allocation, respectively, so it can reasonably reconstruct clusters of complex shapes. Meanwhile, a satellite peak attenuation technique is embedded into MDPC+ to give it a high resistance to the interference of satellite peaks (i.e., non-center density peaks). Besides, MDPC+ only needs kNN distances of data as its input, so it is suitable for large datasets. Experimental results on both synthetic and real-world datasets demonstrate the superiority of MDPC+ in center detection, complex shape reconstruction, and running speed.

1. Introduction

Clustering that aims to group similar objects is critical for the extraction of potential and valuable knowledge from data, which has been applied to pattern recognition [1,2], image processing [3–5], machine learning [6], computer vision [7], etc.

Different clustering methods have been developed based on specific assumptions of a “cluster” [8,6]. For the popular K-centers methods [9], [10], a cluster is a group of points with minimum distances to a single center. Different center initializations are used to rerun the K-centers to find a relatively good clustering result, because it requires presetting the number of centers and is usually sensitive to its center initialization. To remedy this, the Affinity Propagation algorithm [11] devised an “affinity propagation” strategy to adaptively find high-quality exemplars as centers. Although K-centers and AP are efficient in partitioning hyper-spherical clusters, they are not applicable to arbitrary-shaped clusters.

Density-based clustering methods can work well for arbitrary shape reconstruction. DBSCAN [12], following its assumption that a cluster is a set of maximum density-connected points, can detect arbitrary-shaped clusters with sufficient density. Its parameters (ϵ and $MinPts$) need to be well-tuned to obtain a reasonable density-connectivity criterion, which is usually a tedious process. Late works, e.g., [13], [14], managed to do automatic parameter tuning. Still, they may merge highly overlapping clusters [15].

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

* Corresponding author.
E-mail addresses: jonnyguan73@163.com (J. Guan), shengli@zjut.edu.cn (S. Li), hxx@zjut.edu.cn (X. He), fl_katrina@163.com (J. Chen).

https://doi.org/10.1016/j.ins.2023.01.144
Received 31 January 2022; Received in revised form 6 January 2023; Accepted 31 January 2023

Available online 4 February 2023

In 2014, *Science* published the clustering by fast search and find of density peaks–Density Peak Clustering (DPC) [16]. Its center assumption—cluster centers are density peaks that are surrounded by low-density neighbors and are far away from points of higher densities—enables it to easily divide highly overlapping clusters by finding appropriate density peaks as centers. But DPC cannot well capture multi-peak clusters of complex shapes [17,18], for its center assumption provides no criterion to distinguish the correct density peaks (that can represent the true cluster centers), misleading the center selection and resulting in a poor clustering result. Although improved works were proposed [17–23], but still following DPC's center assumption. Also, DPC is prohibitively time-consuming [18], and its allocation strategy may rudely associate points without considering density connectivity, leading to wrong allocations [17].

To achieve successful clustering without encountering the above-mentioned issues, a Main Density Peak Clustering algorithm (MDPC+) is proposed, which follows a new center assumption: a cluster center is a main density peak (hereinafter, a main peak) that should have a relatively higher density than surrounding points and have a path with a relatively large density deviation cost towards higher density peaks. So, MDPC+ can achieve the easy detection of real cluster centers of multi-peak clusters, and fast reconstruct complex shapes. The main contributions of MDPC+ are as follows:

1. A cluster center is herein regarded as a main peak, which helps to precisely distinguish the correct cluster centers (main peaks) from non-center density peaks (satellite peaks). Besides, a satellite peak attenuation technique is designed to resist the interference of satellite peaks, easing the detection of main peaks in the decision graph;
2. The clustering problem is regarded as a graph clustering problem, and two graph structures with specific weight functions are designed to allocate non-peaks and density peaks, respectively. Based on this, MDPC+ can well reconstruct complex shape clusters;
3. MDPC+ only needs kNN distances of data as its input, so it is suitable for large datasets.

The rest paper is composed as follows: Section 2 introduces the related works. Section 3 mainly focuses on the proposed method. Section 4 presents the experiments and discussion. Section 5 gives the final conclusion.

2. Related works

2.1. The DPC algorithm

Given a dataset of n points $X = \{x_1, x_2, \ldots, x_n \mid x_i \in \mathbb{R}^m\}, X \in \mathbb{R}^{m\times n}$, for each point x_i, DPC first estimates its local density ρ_i as in Eq. (1), and then calculates its distance δ_i from the nearest higher density point as in Eq. (2). Where d_{ij} is the Euclidean distance between point x_i and x_j, and "cutoff distance" d_c is a user-specified parameter. For the highest density point x_i, DPC gives $\delta_i = \max_{x_i \neq x_j} \left(d_{ij}\right)$.

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

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

According to DPC's cluster center assumption, by observing the decision graph, density peaks with the top largest γ ($\gamma = \rho \cdot \delta$) are manually selected as centers and given unique labels. Subsequently, the remaining points directly inherit the labels of their nearest higher density points. Once each point obtains a label, clustering is done.

From the perspective of graph clustering, let $G(X, \vec{E})$ be complete digraph of dataset X according to density boosting, where $\vec{E} = \left\{\vec{e}_{ij} | \rho_j > \rho_i, x_j, x_i \in X\right\}$.

$$w(\vec{e}_{ij}) = d_{ij} \cdot \rho_i, \vec{e}_{ij} \in \vec{E} \tag{3}$$

DPC gives a weight to each directed edge $\vec{e}_{ij}$ as in Eq. (3) and cuts $G(X, \vec{E})$ into n_c clusters $Cl = \left\{Cl_1, Cl_2, ..., Cl_{n_c}\right\}$ with minimum weight, where a cluster Cl_i is connected subgraph $G_{Cl_i}(Cl_i, \vec{E}_{Cl_i})$ of G, i.e., $Cl_i \subset X$ and $\vec{E}_{Cl_i} \subset \vec{E}$. That is, to solve the following problem:

$$\min_{Cl} \sum_{i=1}^{n_c} \sum_{\vec{e}_{ab} \in \vec{E}_{Cl_i}} w(\vec{e}_{ab}) \quad \text{s.t. } n_c < n \tag{4}$$

To solve Problem (4) needs to first reserve the minimum weight edge (δ path) projected from each point: to get the minimum spanning tree with the highest density point as the root node; and then, to cut off $n_c - 1$ edges of the top largest weight values: to search for density peaks with the top largest γ values as centers.

2.2. DPC's limitations and improvements

The simplicity and efficiency of DPC in capturing non-spherical shapes make it a promising and concerning clustering algorithm [24]. Nevertheless, as mentioned in [25], density peaks with the top largest γ values are not necessarily centers. DPC may select

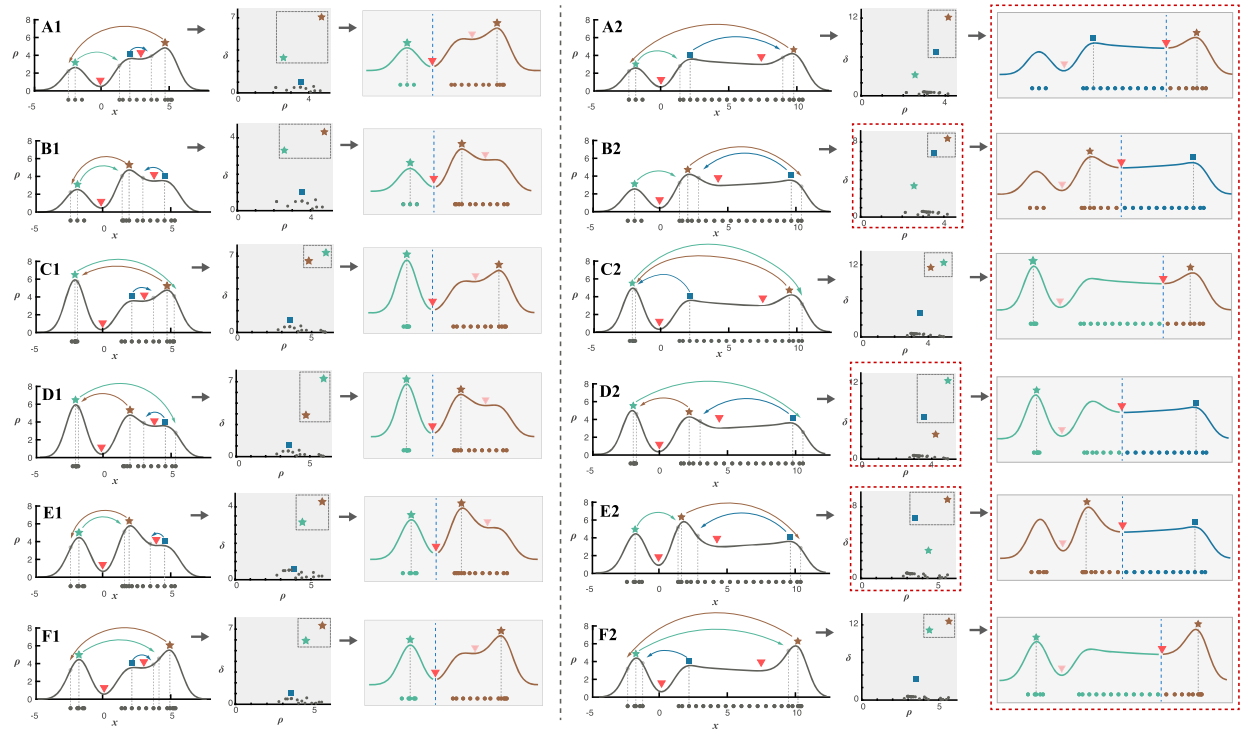

Fig. 1. The performance of DPC on different one-dimensional datasets with multi-peak clusters, where "★" marks a main peak, blue "■" marks a satellite peak, and red "▼" marks a boundary location between two single-peak clusters. Error results are framed in red dashed.

inappropriate density peaks as cluster centers. Besides, DPC may connect non-center density peaks to irrelevant points beyond their local areas, for giving no distance constraint to edge $\vec{e} \in \vec{E}$, resulting in the misallocation of points [17].

To better demonstrate DPC's limitation on the dataset of multi-peak clusters, twelve one-dimensional simple datasets of multi-peak clusters (i.e., datasets of two clusters with three density peaks) are presented in Fig. 1. The permutations of these datasets are $P(3,3) = 6$. Being divided into two categories according to the distance between the two density peaks of the multi-peak cluster, the twelve datasets are obtained with each dataset being given a Gaussian density estimation function. As shown, the two density peaks of the multi-peak cluster in the left-side six datasets (A1, B1,. . . , F1) are relatively close, while in the right-side datasets (A2, B2,. . . , F2) are far apart.

Note that DPC actually cuts all twelve datasets into two clusters at a local minimum point (i.e., a boundary point, marked with a red "▼") on their density estimation functions, which is only workable for the left-side six datasets, as shown in Fig. 1. For the right-side datasets, the cutting boundary points are mischoosen due to the long distance between two density peaks in the multi-peak cluster. So, the non-center density peaks (blue "■") get large δ values, resulting in the wrong cluster center selections in A2, B2, D2, and E2. For the same reason, the non-center density peaks are connected to wrong areas in datasets C2 and F2. As verified, DPC is not robust in detecting cluster centers from density peaks or assigning the remaining density peaks. Also, DPC's assumption pays more attention to the width of the valley between density peaks rather than the depth, which violates the principle of density-based methods that clusters are separated by low-density gaps [26,12].

Although improvement methods were proposed, they all follow DPC's core center assumption without discussing the effective detection of the correct cluster centers among multiple density peaks. For example, Du et al. [19] applied kNN-based density to help discover density peaks within the sparse cluster. [20] designed a fuzzy kNN-based allocation strategy to ensure that each non-center point is assigned within the neighborhood; Liu, et al. [23] designed a shared-nearest-neighbor-based allocation strategy that associates points according to shared nearest neighbor information; Abbas, et al. [21] designed a robust allocation strategy that fully takes mutual nearest neighbor information into account; Du, et al. [22] and Pizzagalli, et al. [17] used geodesic distance instead of Euclidean distance for the allocation of non-center points.

MDPC+ inherits the main idea of DPC to search for cluster centers, but, based on our new assumption, it narrows down the search range of cluster centers by only searching for main peaks. Our method can effectively distinguish true cluster centers from density peaks, and accurately assign non-center density peaks.

3. The proposed MDPC+ algorithm

In this section, a detailed introduction to MDPC+ is given.

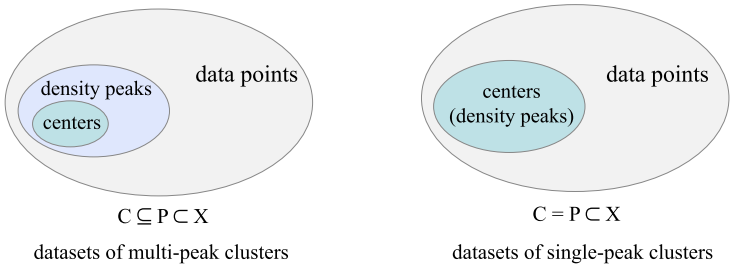

Fig. 2. The Venn diagram of cluster centers, density peaks, and data points.

3.1. The assumption of MDPC+

In density-based clustering methods, clusters are normally defined as high-density areas separated by low-density gaps [12], and the highest density point within a cluster is regarded as the cluster center [27]. Nevertheless, the density peak attribute (i.e., local density maxima) is a necessary but insufficient condition for cluster centers. Because a cluster usually has multiple density peaks, i.e., $C \subseteq P \subset X$, where C is a set of cluster centers, P is a set of density peaks, and X is a set of data points. Only and if only all clusters are single-peak clusters, then $C = P \subset X$.

Fig. 2 demonstrates the Venn diagram of cluster centers, density peaks, and data points in a dataset. As shown, non-peaks (i.e., $\bar{P} = X \setminus P$) shall never be cluster centers, thus the cluster center selection region can be narrowed down to density peaks. To detect density peaks, a clear definition of a density peak is needed. According to the local density maxima characteristic of a density peak, the k-nearest neighbors $N_k(x_i)$ of a point x_i is defined as its local area, as in Definition 1. A kNN-based density estimation method [20] (as in Eq. (5)) is herein applied to reduce computational complexity.

$$\rho_i = \sum_{x_j \in N_k(x_i)} e^{-d_{ij}} \tag{5}$$

To effectively distinguish cluster centers from density peaks, the highest density peaks in clusters are defined as main peaks (i.e., main peaks are cluster centers) and the remaining density peaks as satellite peaks, as in Definition 2.

Definition 1. Point x_i is a density peak, denoted as p, if it has highest density within its k nearest neighbors $N_k(x_i)$, i.e., $\rho_i > \max_{x_j \in N_k(x_i)}(\rho_j)$.

Definition 2. A density peak $p_i \in Cl_a$ is the main peak of cluster Cl_a, if $\rho_{p_i} = \max_{p_j \in Cl_a}(\rho_{p_j})$, otherwise, p_i is a satellite peak.

In a cluster, all satellite peaks can find a higher density peak within the cluster; while the main peak needs to walk through a path with at least one low-density gap between clusters to find higher density peaks. The path will have a deep density drop that should have a large density deviation cost. Inspired by this, a new assumption of a main peak is proposed to help distinguish main peaks from satellite peaks, as in Assumption 1.

Assumption 1. Cluster centers are main peaks that have a relatively high density than surrounding neighbors and have a path with a relatively large density deviation cost towards density peaks of higher densities.

To perform clustering, based on Assumption 1, adjacent points of similar density are grouped into a cluster, ensuring that only the main peak has a path with a relatively large density deviation cost towards a higher density peak, which constructs our clustering idea. In what follows, the corresponding graph clustering problem is proposed.

3.2. The graph clustering problem of MDPC+

MDPC+ aims to assign each point to a denser area in proximity, like Mean-Shift [27]. By classifying dataset X into density peaks $P = \left\{p_1, p_2, ..., p_{n_p}\right\}$ and non-peaks $\bar{P}$, $X = P \cup \bar{P}$, we note that non-peaks can always find the best adjacent higher density point of similar density within local areas, while density peaks cannot. In other words, non-peaks can be assigned within local areas, while density peaks cannot. So, to assign non-peaks, a kNN digraph of dataset X is only required, denoted as $G_{kNN}(X, \vec{E}_K)$, where $\vec{E}_K = \left\{\vec{e}_{ij} \,|\rho_j > \rho_i, x_j \in N_k(x_i), x_i, x_j \in X\right\}$. To assign density peaks, a complete digraph of density peaks should be constructed, called peak digraph, and denoted as $G_P(P, \vec{E}^*)$, where $\vec{E}^* = \left\{\vec{e^*}_{p_i p_j} \,|\rho_{p_j} > \rho_{p_i}, p_i, p_j \in P\right\}$.

Based on our clustering idea, specific edge-weight functions are introduced to quantitatively describe the relationship between data points. So, the clustering problem of MDPC+ can be executed as two graph-cut problems.

First, MDPC+ cuts $G_{kNN}(X,\vec{E}_K)$ into n_p sub-clusters $Cl^s=\left\{Cl_1^s,Cl_2^s,...,Cl_{n_p}^s\right\}$ with minimum weight, where a sub-cluster Cl_i^s is connected subgraph $G_{Cl_i^s}(Cl_i^s,\vec{E}_{Cl_i^s})$ of G_{kNN}, i.e., $Cl_i^s\subset X$ and $\vec{E}_{Cl_i^s}\subset\vec{E}_K$. That is, to solve Problem (6), where $w_{\bar{P}}(\cdot)$ is a edge-weight function.

$$\min_{Cl^s}\sum_{i=1}^{n_p}\sum_{\vec{e}_{ab}\in\vec{E}_{Cl_i^s}} w_{\bar{P}}(\vec{e}_{ab})\quad \text{s.t. } n_p=|P| \tag{6}$$

Then, MDPC+ detects n_p density peaks to build peak digraph $G_P(P,\vec{E}^*)$, and cuts $G_P(P,\vec{E}^*)$ into n_c clusters $Cl^*=\left\{Cl_1^*,Cl_2^*,...,Cl_{n_c}^*\right\}$ with minimum weight, where a cluster Cl_i^* is connected subgraph $G_{Cl_i^*}(Cl_i^*,\vec{E}^*_{Cl_i})$ of G_P, i.e., $Cl_i^*\subset P$ and $\vec{E}^*_{Cl_i}\subset\vec{E}^*$. That is, to solve Problem (7), where $w_P(\cdot)$ is an edge-weight function, and n_c is the number of selected main peaks.

$$\min_{Cl^*}\sum_{i=1}^{n_c}\sum_{\vec{e^*}_{p_ap_b}\in\vec{E}_{Cl_i^*}} w_P(\vec{e^*}_{p_ap_b})\quad \text{s.t. } n_c\leqslant n_p \tag{7}$$

3.3. The clustering of MDPC+

In what follows, a detailed illustration of MDPC+'s two steps: 1) the local allocation of non-peaks in a kNN digraph; 2) the global clustering of density peaks in a peak digraph, will be given.

3.3.1. The allocation of non-peaks in a kNN digraph

Based on $G_{kNN}(X,\vec{E}_K)$, a weight function $w_{\bar{P}}(\cdot)$ is designed to make sure that each non-peak is associated with a reasonable adjacent higher density point of similar density, as in Eq. (8). Where $\phi_d(\vec{e}_{ij})$ and $\phi_\rho(\vec{e}_{ij})$ are the relative influence of distance and density deviation from x_j to x_i, as in Eq. (9) and (10), and μ_{ij} is the density deviation value between x_j and x_i, as in Eq. (11).

$$w_{\bar{P}}(\vec{e}_{ij})=\phi_d(\vec{e}_{ij})+\phi_\rho(\vec{e}_{ij}),\vec{e}_{ij}\in\vec{E}_K \tag{8}$$

$$\phi_d(\vec{e}_{ij})=\frac{d_{ij}-\min\limits_{\vec{e}_{it}\in\vec{E}_K}\left(d_{it}\right)}{\max\limits_{\vec{e}_{it}\in\vec{E}_K}\left(d_{it}\right)-\min\limits_{\vec{e}_{it}\in\vec{E}_K}\left(d_{it}\right)},\vec{e}_{ij}\in\vec{E}_K \tag{9}$$

$$\phi_\rho(\vec{e}_{ij})=\frac{\mu_{ij}-\min\limits_{\vec{e}_{it}\in\vec{E}_K}\left(\mu_{it}\right)}{\max\limits_{\vec{e}_{it}\in\vec{E}_K}\left(\mu_{it}\right)-\min\limits_{\vec{e}_{it}\in\vec{E}_K}\left(\mu_{it}\right)},\vec{e}_{ij}\in\vec{E}_K \tag{10}$$

$$\mu_{ij}=\frac{|\rho_i-\rho_j|}{\max(\rho_i,\rho_j)},\vec{e}_{ij}\in\vec{E}_K \tag{11}$$

To best allocate non-peaks (i.e., to solve Problem (6)), we need to reserve the minimum weight edges projected from all non-peaks in $\bar{P}$, that is to say, to get n_p minimum spanning trees with unique density peaks in P as root nodes in graph $G_{kNN}(X,\vec{E}_K)$.

As a result, a strong association forest $F(X,\vec{E}_F)$ of n_p trees (sub-clusters) is obtained, where all points are connected with adjacent points with similar density through edges in $\vec{E}_F$.

3.3.2. The grouping of density peaks in a peak digraph

In forest $F(X,\vec{E}_F)$, each non-peak $x_i\in\bar{P}$ has only one association path, denoted as $\theta^*_{x_ip(x_i)}$, to its density peak $p(x_i)$ (root node), thus, by adding bridge-edges E_B that cross trees (i.e., sub-clusters) to $F(X,\vec{E}_F)$, as in Definition 3, $F(X,\vec{E}_F)$ is transferred into an association graph $G_A(X,E_A),E_A=E_F\cup E_B$ (E_F is the undirected version of $\vec{E}_F$), where associated density peaks are connected through paths.

Definition 3. A bridge-edge $e_{ij}\in E_B$ connects points x_i and its mutual neighbors x_j of another sub-cluster, and the bridge-edge set E_B is defined in Eq. (12), where $k_b=\min(2\lfloor\ln(n)\rfloor,k)$ (symbol $\lfloor\cdot\rfloor$ is a floor function), and Label(x) returns the label of point x. Note that, k_b is set to $k_b<k$ to better detect the proximal mutual neighbors between intersecting sub-clusters.

$$E_B=\left\{e_{ij}|x_i\in N_{k_b}(x_j)\wedge x_j\in N_{k_b}(x_i),\text{Label}(x_i)\neq\text{Label}(x_j)\right\} \tag{12}$$

According to Assumption 1, the calculation of the density deviation cost between density peaks is needed to find main peaks as centers. Since each non-peak has a path to its density peak, the density deviation cost between them can be calculated during its allocation. Therefore, an adjacent graph matrix $A_P\in\mathbb{R}^{n_p\times n_p}$ about the minimum density deviation cost between density peaks of intersecting sub-clusters can be fast obtained as in Eq. (13). Function $\Gamma(x_i,x_j,\lambda)$ outputs the minimum density deviation cost along the path between points x_i and x_j in association graph $G_A(X,E_A)$, as in Eq. (14), where $\Theta_{x_ix_j}$ is a set of paths (denoted as θ) from x_i to x_j in association graph $G_A(X,E_A)$, and $\lambda\in[1,5]$ is the attenuation coefficient (discussed in Section 3.4).

$$A_P(i,j) = \min_{x_a \in Cl_i^s, x_b \in Cl_j^s, e_{ab} \in E_B} \left(\Gamma(x_a, p_i, \lambda) + \Gamma(x_b, p_j, \lambda) + \mu_{ab}^{\lambda}\right) \tag{13}$$

$$\Gamma(x_i, x_j, \lambda) = \begin{cases} \min\limits_{\theta \in \Theta_{x_i x_j}} \left(\sum\limits_{e_{ab} \in \theta} \mu_{ab}^{\lambda} \right) & \Theta_{x_i x_j} \neq \varnothing \\ +\infty & \Theta_{x_i x_j} = \varnothing \end{cases} \tag{14}$$

Eq. (13) tells that only when density peaks p_i and p_j are in intersected sub-clusters, $A_p(i,j) = \Gamma(p_i, p_j, \lambda)$, otherwise, $A_p(i,j) = +\infty$.

Notably, the shortset path between point x_i and its density peak $p(x_i)$ in $G_A(X, E_A)$ is actually their association path $\theta^*_{x_i p(x_i)}$ in forest $F(X, \vec{E}_F)$. Therefore, during the building of forest $F(X, \vec{E}_F)$, the value of $\Gamma(x_i, p(x_i), \lambda)$ can be simultaneously calculated, as in Eq. (15). This greatly reduces computational complexity, making MDPC+ run fast.

$$\Gamma(x_i, p(x_i), \lambda) = \sum_{e_{ab} \in \theta^*_{x_i p(x_i)}} \mu_{ab}^{\lambda} \tag{15}$$

According to A_P, the minimum density deviation cost between each pair of density peaks in P can be fast calculated by applying the Dijkstra algorithm [28]. Based on Assumption 1, the weight function $w_P(\cdot)$ of a edge $\vec{e}^*_{p_i,p_j}$ in peak digraph $G_P(P, \vec{E}^*)$ is defined as in Eq. (16).

$$w_P(\vec{e}^*_{p_i p_j}) = \rho_{p_i} \cdot \Gamma(p_i, p_j, \lambda) \tag{16}$$

For each density peak $p_i \in P$, the minimum weight δ_{p_i} towards a higher density peak is recorded, as in Eq. (17). Density peaks that have no path to higher density peaks in association graph $G_A(X, E_A)$, i.e., $\delta = +\infty$ are usually defined as main peaks, given $\delta = 1.2 \times \max\limits_{p_i : \delta_{p_i} \neq +\infty} (\delta_{p_i})$. Where constant "1.2" is used to highlight the main peaks in the decision graph. In addition, as discussed in Section 3.1, non-peaks shall never be cluster centers, thus set $\delta_i = 0$ for each non-peak $i \in \bar{P}$.

To solve the grouping problem of density peaks (Problem (7)), n_c main peaks should be selected with the top largest γ values as centers with unique cluster labels; and then, each remaining satellite peak $p_i \in P$ inherits the label of the higher density peak along its δ_{p_i} path.

$$\delta_{p_i} = \min_{p_j : \rho_{p_i} < \rho_{p_j}} \Gamma(p_i, p_j, \lambda) \tag{17}$$

Once each density peak owns a cluster label, all non-peaks of its sub-cluster are given the same cluster label. After each point gets a cluster label, clustering is done.

3.3.3. The learning of confidence

In a sub-cluster, a density peak (sub-cluster center) with local density maxima should have top confidence, while a non-peak with a large density deviation cost should have low confidence. In MDPC+, each point has a short path towards a density peak, thus, for each point x_i, its confidence can be learned as in Eq. (18).

$$\xi_i = \frac{1}{\Gamma(x_i, p(x_i), \lambda)} \tag{18}$$

According to Eq. (18), since a density peak p_i's most associated density peak $p(p_i)$ is itself, $\Gamma(p_i, p(p_i), \lambda) = 0$, therefore, $\xi_{p_i} = +\infty$.

Noise (outliers) that are usually along the borders of clusters tends to have relatively low confidence. Thus, good confidence learning can realize high-efficiency denoising (i.e., cutting out low-confidence data as noise), thereby, greatly improving the clustering precision. It is of great significance in real-world applications (see experiment Section 4.3.1 and 4.3.2).

3.4. The effect of attenuation-coefficient λ

As shown in Eq. (14), $\Gamma(p_i, p_j, \lambda)$ indicates the total density deviation cost of edges along the shortest path between p_i and p_j. According to Eq. (11), density deviation $\mu \in (0, 1)$. Therefore, $\lambda \in [1, 5]$ can be used to attenuate the influence (i.e., the density deviation μ) of edges: when $\lambda > 1$, the larger the influence μ of an edge, the smaller it will be attenuated, and vice versa. It effectively retains the strong-influence (i.e., high density deviation) edge while attenuating the weak-influence edge. In MDPC+, $\lambda = 2$ is set as the default, and λ can be adjusted to change the attenuation strength.

For a main peak, it has to cross at least a low-density gap to find a higher density peak, and such a cross-cluster path usually has strong-influence edges; while, for a satellite peak, it can find a higher density peak within its cluster along a path that consists of similar density points, such an inter-cluster path usually has weak-influence edges. Therefore, the attenuation-coefficient λ is to amplify the difference between main peaks and satellite peaks, thereby highlighting main peaks in the decision graph. In this way, the accurate selection of cluster centers can be achieved (discussed in Section 4.4.1).

3.5. The overall clustering process of MDPC+

In this section, our clustering idea is visualized on one-dimensional datasets, and our real clustering algorithm is demonstrated on a two-dimensional synthetic dataset.

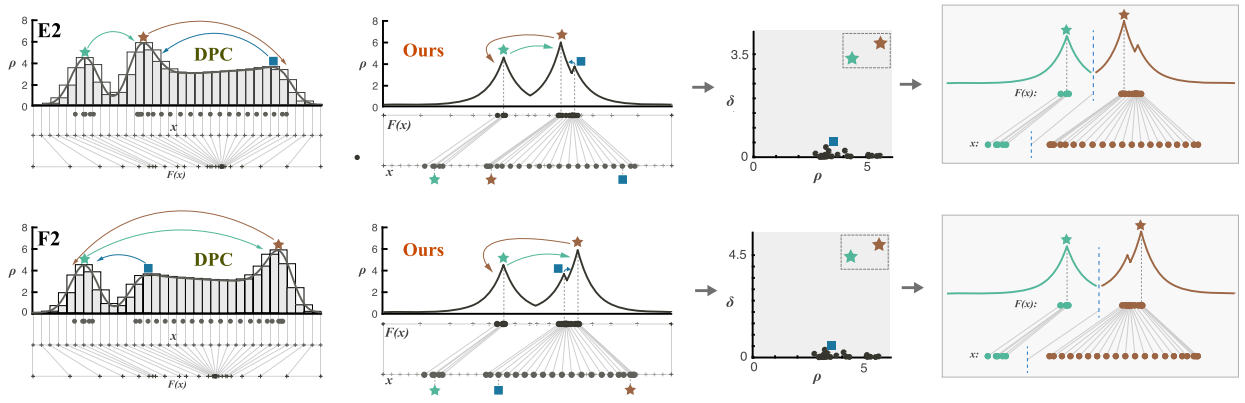

Fig. 3. The core clustering idea of MDPC+ on two different one-dimensional datasets of multi-peak clusters. **F(x)**-space is the transform space of **x**-space based on density deviation rate.

3.5.1. *The one-dimensional display of our clustering idea*

Fig. 3 illustrates the main clustering idea of MDPC+. Based on the density distribution function, MDPC+ theoretically changes the lengths of segments on the x-axis according to their density deviation rates. It elongates segments with a large density deviation rate and squeezes segments with a low density deviation rate. As a result, the distance between a satellite peak and its main peak is greatly squeezed, while the deep valley (i.e., the low-density gap) with a large density drop is elongated. It not only reduces the interference of satellite peaks on the cluster center selection but also ensures a reasonable allocation of satellite peaks.

As in Fig. 3, unlike DPC's assumption that focuses on wide valleys between density peaks, our assumption pays more attention to the depth of the valleys following the principle that clusters are separated by low-density gaps. Under our new assumption, MDPC+ can accurately find main peaks and reasonably allocate the remaining data points even when dealing with datasets of multi-peak clusters.

Fig. 4 shows the performance of MDPC+ in processing the twelve datasets of multi-peak clusters (as in Fig. 1). Obviously, MDPC+ is superior to DPC, since it can easily identify main peaks as centers to achieve perfect clustering in all situations.

3.5.2. *The clustering process on a two-dimensional dataset*

Fig. 5 presents the process of MDPC+ on a two-dimensional synthetic dataset with a spherical multi-peak cluster and a crescent multi-peak cluster.

According to Eq. (8), by inputting a kNN matrix of the dataset with $k = 20$, the dataset is constructed into an association graph of 21 sub-clusters (sub-trees) with 21 density peaks as centers (root nodes); and then, after searching for minimum weight paths between adjacent peaks, MDPC+ obtains an adjacent graph of density peaks. On the basis of the adjacent graph, the peak digraph is fast built by applying the Dijkstra algorithm [28]; subsequently, MDPC+ cuts the peak digraph into a minimum spanning tree by searching δ paths in the peak digraph. With the ρ-δ decision graph, the two real cluster centers (density peak 13 and 21) with main peak characteristics are successfully selected as cluster centers; followed, MDPC+ cuts the edge projected from density peak 13 to cut the graph into two clusters with minimum weight, and assigns the cluster labels to sub-clusters; finally, after the dataset being perfectly separated into two clusters, the clustering is done.

Although the Geodetic distance along path "13–21" is shorter than path "13–4", its density deviation cost is much higher. Because path "13–21" crosses a low-density gap between clusters, it has a relatively higher density deviation cost; while path "13–4" passes within one cluster of similar density, so it has a low density deviation cost. As a result, only main peak 13 with the highest density of the crescent cluster has to cross the low-density gap between clusters to find a higher density peak, so it has a relatively larger δ value.

These examples demonstrate the effectiveness of MDPC+ in identifying cluster centers from density peaks and assigning non-center density peaks.

3.6. *Complexity analysis*

Algorithm 1 shows the pseudocode of the proposed MDPC+ algorithm.

Line 1~4: the kNN-based density estimation ρ of data points, needs $O(n\log(n))$.

Line 5~19: the identification of density peaks P and the generation of sub-clusters Cl^s, needs $O(nk)$.

Line 20~27: the acquisition of adjacent graph matrix A_P, needs $O(nk_b)$.

Line 28~37: the fast calculation of δ by using the Dijkstra algorithm, overall needs $O(n_p\log(n_p) + n_{e_{A_p}})$, where $n_{e_{A_p}}$ indicates the total number of adjacent edges in the adjacency matrix A_P.

Line 38~49: the generation of clusters Cl, needs $O(n)$.

So, the overall time complexity of MDPC+ is $O(n\log(n) + nk + nk_b + n_p\log(n_p) + n_{e_{A_p}} + n) = O(n\log(n) + nk)$, since $k_b < k$, $n_p \ll n$, and $n_{e_{A_p}} \ll n$.

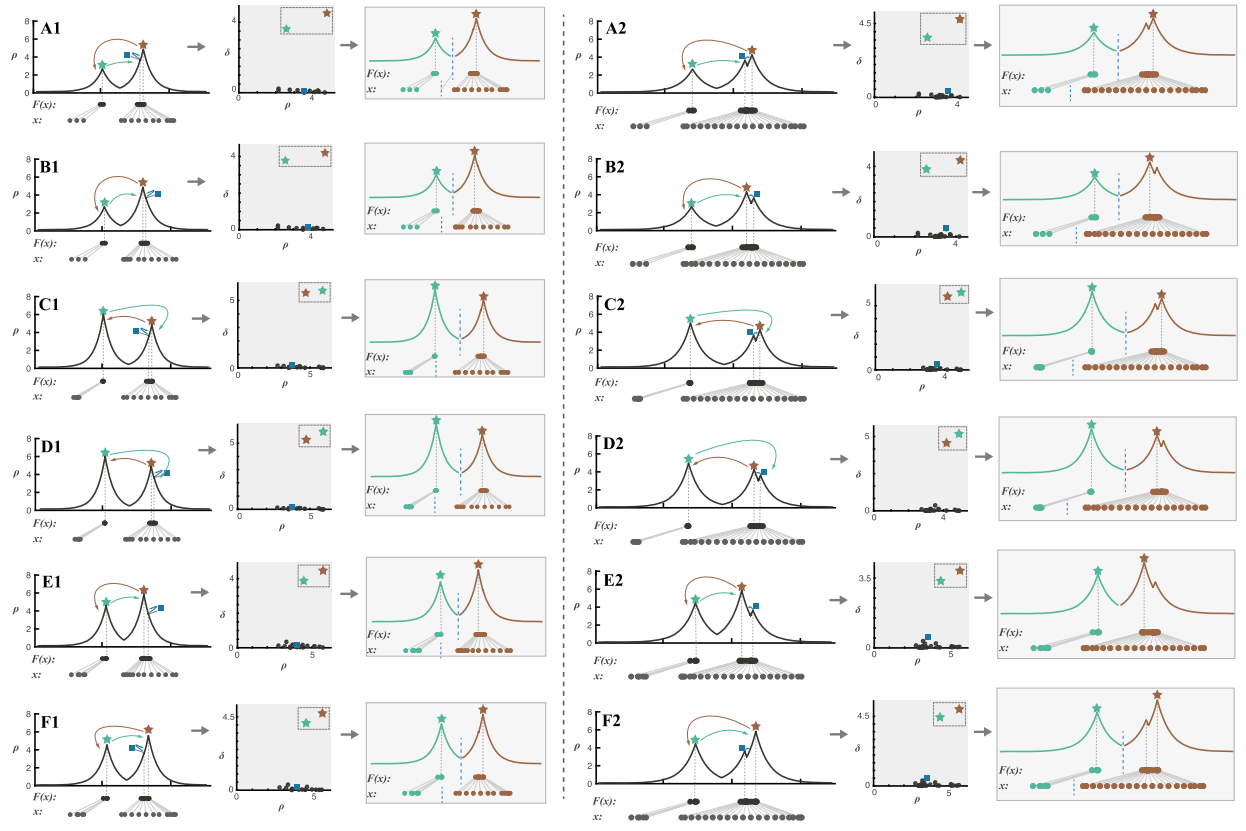

Fig. 4. The performance of MDPC+ on different types of one-dimensional datasets of multi-peak clusters. **F(x)**-space is the transform space of **x**-space based on density deviation rate.

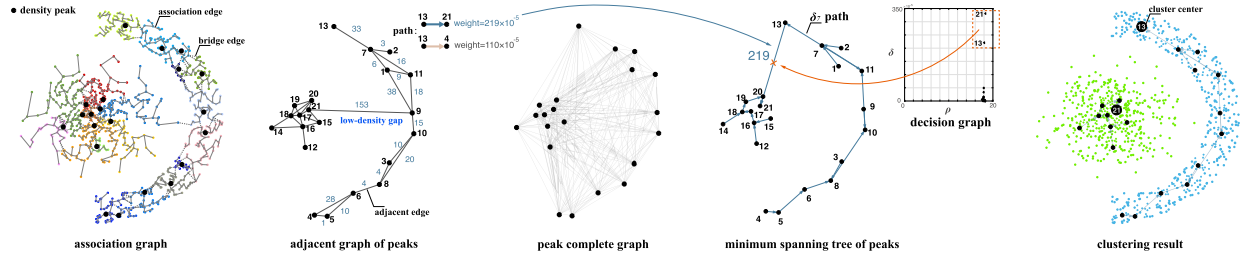

Fig. 5. The overall clustering process of MDPC+ on a synthetic dataset, where black numbers are the density ascending order of density peaks, and blue numbers are weight values.

4. Experiment

4.1. Experimental set up

Datasets: nine popular synthetic datasets of different shapes and eight real-world datasets are selected to benchmark the proposed algorithm. The detailed summarization of these datasets is displayed in Table 1.

Algorithms and settings: K-means [9], a popular K-centers clustering technique; AP [11] a classic partition clustering technique; the Self-tuning Spectral Clustering algorithm (SSC) [41], a widely used Spectral Clustering technique; DBSCAN [12], the classic density-based clustering technique; DPC [16] and three state-of-the-art DPC variations: SSSP-DPC [17], SNN-DPC [23], and PGDPC [18]; the proposed MDPC+.

For K-means and SSC, we report the mean results of 10 runs on each dataset; while for all DPC-based algorithms, we manually select appropriate density peaks as cluster centers via their own decision graphs.

Parameter requirements: K-means (N_c), AP ($dampfact$), SSC (N_c/k), DBSCAN ($\epsilon/MinPts$), DPC(p), SSSP-DPC (p), SNN-DPC (k), PGDPC (k), and MDPC (k), where parameter N_c is the pre-set number of clusters, and p is the cutoff percentile for setting d_c [16]. These required parameters are shown as PAR in the subsequent experimental results tables (Table 2, Table 3).

Algorithm 1 The MDPC+ algorithm.

Input: dataset $X = \{x_1, x_2, \ldots, x_n \mid x_i \in \mathbb{R}^m\}$, and the number of neighbors k, and attenuation-coefficient λ
Output: cluster result $Cl = \{Cl_1, Cl_2, ..., Cl_{n_c}\}$.
fast obtain kNN matrix of data with number k of nearest neighbors
for each point $x_i \in X$ **do**
  $\rho_i = \sum_{x_j \in N_k(x_i)} e^{-d_{ij}}$ // Eq. (5)
end for
for each point $x_i \in X$ **do**
  $\Gamma(x_i, p(x_i), \lambda) = 0$ // initialize the density deviation cost.
end for
order the dataset X as X' in descending order of density ρ
for each point $x_i \in X'$ **do**
  if $\rho_i > \max_{x_j \in N_k(x_i)} (\rho_j)$ **then**
    x_i is a density peak p, $P = P \cup x_i$// Definition 1
    Label(x_i) ← a unique sub-cluster label
  else
    find the higher density neighbor $x_j \in N_k(x_i)$ with minimum weight $w_{\vec{P}}(\vec{e}_{ij})$, according to Eq. (8)
    Label(x_i) ← Label(x_j)
    $\Gamma(x_i, p(x_i), \lambda) = \Gamma(x_j, p(x_j), \lambda) + \mu_{ij}^{\lambda}$ // here $p(x_i) = p(x_j)$.
  end if
end for
points with same sub-cluster label form n_p clusters $Cl^s = \left\{Cl_1^s, Cl_2^s, ..., Cl_{n_p}^s\right\}$.
for each pair of density peaks $p_i, p_j \in P$ **do**
  if p_i, p_j are in intersecting sub-clusters **then**
    $A_P(i, j) = A_P(j, i) = \Gamma(p_i, p_j, \lambda)$ // Eq. (13)
  else
    $A_P(i, j) = A_P(j, i) = +\infty$
  end if
end for
obtain adjacent graph matrix A_P of density peaks
order the density peak set P as P' in descending order of density ρ
for each density peak $p_i \in P$ (from high-ρ to low-ρ) **do**
  apply the Dijkstra algorithm to find density peak p_i's nearest higher density peak p_j in adjacent graph A_P
  if $p_j \neq \varnothing$ **then**
    $\delta_{p_i} = \Gamma(p_i, p_j, \lambda)$
  else
    $\delta_{p_i} = +\infty$
  end if
end for
find all density peaks with $\delta = +\infty$, and set them $\delta = 1.2 \times \max_{p_i : \delta_{p_i} \neq +\infty} (\delta_{p_i})$
select the number n_c of cluster centers C with large γ in decision graph
for each $p_i \in P$ **do**
  if $p_i \in C$ **then**
    density peak p_i is a center
    Label(p_i) ← an unique cluster label
  else
    Label(p_i) ← Label(p_j) // p_j is the higher density peak with minimum density deviation cost path δ_{p_i} from p_i
  end if
end for
all points inherit the cluster label from their density peak.
points with same cluster label form n_c clusters $Cl = \{Cl_1, Cl_2, ..., Cl_{n_c}\}$.
return cluster result $Cl = \{Cl_1, Cl_2, ..., Cl_{n_c}\}$.

Data preprocessing: the min-max normalization [33] is applied to preprocess datasets to avoid the influence of different dimensional metrics.

Machine configuration: Matlab (r2017b) on Mac-Book Pro with 2.9 GHz Intel Core i5, 8G RAM.

Evaluation metric: the popular Adjusted Rand Index (ARI) [42], Adjusted Mutual Information (AMI) [42], and the Fowlkes-Mallows index (FMI) [43] are used to evaluate the clustering performance of comparison algorithms.

4.2. Experiments on synthetic datasets

A quantitative evaluation of MDPC+ is presented on eight common synthetic datasets, consisting of heterogeneous clusters that lie in proximity and are difficult to be detected. Fig. 6 presents the comparison results.

The DPC, SSSP-DPC, SNN-DPC, and MDPC+ were compared in terms of cluster center detection and non-center data allocation. As shown, the proposed MDPC+ almost perfectly identified the real cluster centers and divided the remaining non-center points. SNN-DPC did satisfying jobs on most datasets except for the *Impossible* dataset, but some small flaws existed in its border recognitions of the *Agg*, *Compound*, and *Pathbased* datasets; DPC and SSSP-DPC well recognized the *Agg* and *S3* datasets, but they failed on the *Jain*, *Compound*, *Pathbased*, and *Impossible* datasets due to the inaccurate identification of cluster centers and the wrong allocation of non-center points. By contrast, PGDPC is only more robust on the *Jain* dataset. DBSCAN successfully reconstructed all shapes, but

Table 1
Datasets.

Dataset	Instances	Attributes	Clusters	Source
Agg	788	2	7	[29]
Compound	399	2	6	[30]
Jain	373	2	2	[31]
Pathbased	300	2	3	[32]
S3	5000	2	15	[33]
Impossible	3673	2	7	[26]
R15	600	2	15	[34]
D31	3100	2	31	[34]
Birchrg1	100000	2	100	[35]
Iris	150	4	3	[36]
Wine	178	13	3	[36]
Breastcancer	569	30	2	[36]
Parkin	195	22	2	[36]
YTF	10000	10	41	[37]
USPS	11000	10	10	[38]
OlivettiFaces	400	92×112	40	[39]
MNIST	10000	500	10	[40]

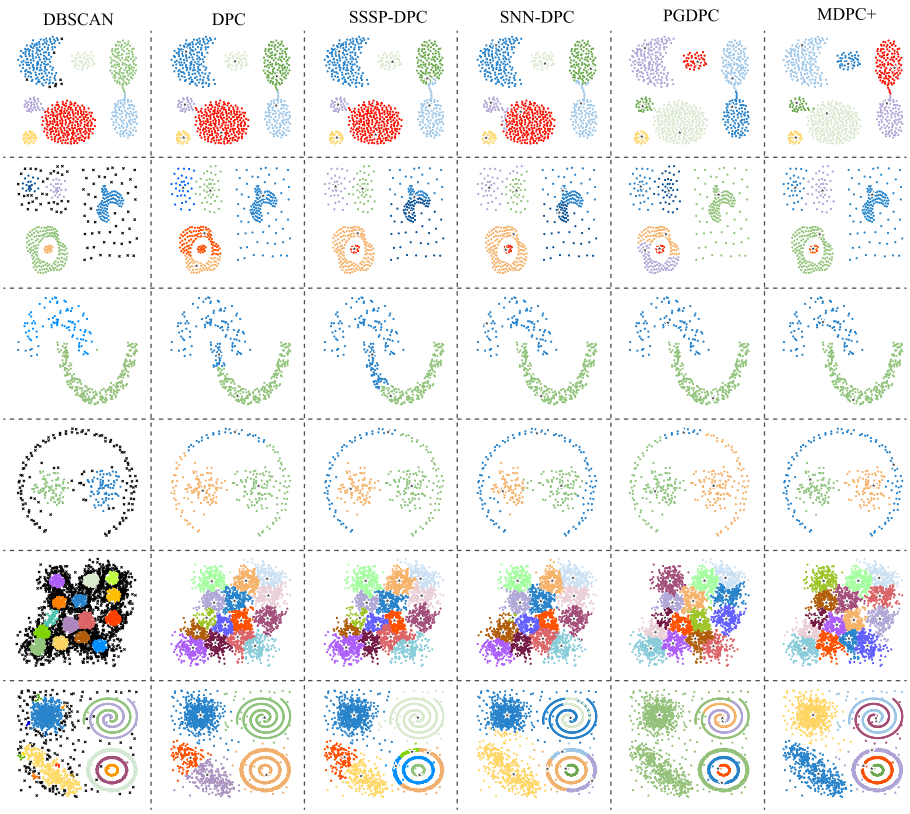

Fig. 6. The clustering results of different algorithms on synthetic datasets, where gray "★" marks the detected cluster centers, and black "×" marks identified noise. The datasets from top to bottom are: *Agg*, *Compound*, *Jain*, *Pathbased*, *S3*, and *Impossible*.

failed to identify the cluster number of the *Pathbased*, *Jain*, and *Impossible* datasets, and misrecognized lots of non-noise points as noise in the *Pathbased* and *S3* datasets.

As shown, only MDPC+ accurately recognized the seven complex-shaped clusters of the *Impossible* dataset [26], verifying the superiority of our cluster assumption.

Table 2 presents the AMI, ARI, and FMI scores with highlighted best results. As shown, MDPC+ shows itself with high scores on almost all datasets; while K-means and SSC seemed inferior in the identification of arbitrarily shaped clusters. As verified, the proposed MDPC+ algorithm has an excellent performance in identifying cluster centers and shape reconstruction.

Table 2
The comparison of AMI, ARI, and FMI on synthetic datasets. PAR represents the parameter setting.

Dataset	Metric	K-means	AP	SSC	DBSCAN	DPC	SSSP-DPC	SNN-DPC	PGDPC	MDPC+
Agg	AMI	0.80 (±0.01)	0.61	0.94 (±0.01)	0.96	**0.99**	0.97	0.93	**0.99**	**0.99**
	ARI	0.72 (±0.03)	0.40	0.95 (±0.01)	0.98	**0.99**	0.97	0.94	**0.99**	**0.99**
	FMI	0.78 (±0.02)	0.54	0.96 (±0.01)	0.98	**0.99**	0.98	0.95	0.99	**0.99**
	PAR	7	0.5	7/20	0.004/7	2	2	18	25	10
Compound	AMI	0.66(±0.06)	0.50	0.71 (±0.01)	**0.86**	0.76	0.84	0.81	0.82	0.84
	ARI	0.58 (±0.14)	0.33	0.49 (±0.00)	**0.90**	0.59	0.83	0.81	0.62	0.85
	FMI	0.68 (±0.11)	0.48	0.61 (±0.00)	**0.92**	0.69	0.87	0.86	0.71	0.90
	PAR	6	0.5	6/10	0.002/4	2	2	6	15	10
Jain	AMI	0.49 (±0)	0.22	0.50 (±0)	0.86	0.54	0.35	**1.00**	**1.00**	**1.00**
	ARI	0.57 (±0)	0.11	0.57 (±0)	0.97	0.62	0.32	**1.00**	**1.00**	**1.00**
	FMI	0.80 (±0)	0.37	0.81 (±0)	0.99	0.84	0.70	**1.00**	**1.00**	**1.00**
	PAR	2	0.5	2/10	0.006/1	2	2	13	10	20
Pathbased	AMI	0.51 (±0)	0.36	0.57 (±0)	0.75	0.50	0.71	0.82	0.44	**0.98**
	ARI	0.46 (±0)	0.22	0.53 (±0)	0.77	0.45	0.61	0.86	0.41	**0.99**
	FMI	0.66 (±0)	0.42	0.59 (±0)	0.85	0.66	0.75	0.91	0.65	**0.99**
	PAR	3	0.5	3/10	0.004/10	2	2	10	15	7
S3	AMI	0.85 (±0.02)	0.47	0.89 (±0)	0.66	0.94	0.88	0.87	**0.96**	**0.96**
	ARI	0.77 (±0.05)	0.32	0.85 (±0)	0.30	0.92	0.83	0.82	**0.95**	**0.95**
	FMI	0.78 (±0.04)	0.44	0.82 (±0)	0.39	0.93	0.84	0.83	0.95	**0.96**
	PAR	15	0.5	15/15	0.001/50	2	2	36	50	100
Impossible	AMI	0.63 (±0.01)	0.18	0.82 (±0.02)	0.90	0.66	0.81	0.67	0.78	**0.95**
	ARI	0.52 (±0.07)	0.05	0.76 (±0.03)	0.93	0.62	0.71	0.53	0.68	**0.97**
	FMI	0.59 (±0.06)	0.16	0.79 (±0.03)	0.94	0.72	0.76	0.63	0.74	**0.98**
	PAR	7	0.5	7/30	0.0005/4	3	2	30	15	10
R15	AMI	0.94 (±0.05)	**0.99**	**0.99** (±0)	0.94	**0.99**	**0.99**	**0.99**	**0.99**	**0.99**
	ARI	0.88 (±0.11)	**0.99**	**0.99** (±0)	0.95	**0.99**	**0.99**	**0.99**	**0.99**	**0.99**
	FMI	0.89 (±0.10)	**0.99**	**0.99** (±0)	0.95	**0.99**	**0.99**	**0.99**	**0.99**	**0.99**
	PAR	15	0.5	15/10	0.001/10	2	2	20	20	40
D31	AMI	0.92 (±0.02)	0.77	**0.97** (±0)	0.86	0.95	0.96	0.96	0.96	0.95
	ARI	0.83 (±0.04)	0.80	**0.95** (±0)	0.71	0.93	0.94	0.94	0.94	0.93
	FMI	0.84 (±0.04)	0.81	**0.95** (±0)	0.72	0.94	0.94	0.94	0.94	0.93
	PAR	31	0.5	31/30	0.001/30	2	2	30	30	30

4.3. Experiments on real-world datasets

Real-world data clustering is difficult due to its high-dimensional and large size characteristics, but it has vital importance in real applications.

To further evaluate the performance of MDPC+, experiments were conducted on eight common real-world datasets: *Iris*, *Wine*, *Breastcancer*, *Parkin*, *YTF*, *USPS*, *OlivettiFaces*, and *MNIST*, where the *OlivettiFaces* data is processed by [23], and the three large datasets: *YTF*, *MNIST* and *USPS*, are processed by [15]. Table 1 lists the details of these datasets, and Table 3 reports the experimental results, where the best results are highlighted.

As shown, the overall performance of MDPC+ is outstanding, verifying that MDPC+ shall be an alternative method for real-world data clustering.

4.3.1. Face recognition on the OlivettiFaces dataset

OlivettiFaces [39] is a well-known face database consisting of 400 face images of 40 persons, and each person has 10 images with different angles. Since the number of clusters (i.e., 40 persons) is considerable to the total number of data points (i.e., 400 faces), it is quite difficult to accurately obtain the 40 real cluster centers [16]. Because some non-center points may seriously interfere with the center selection, thus MDPC+ excludes the interference of non-peaks. Fig. 10 shows the decision graphs of DPC and MDPC+ on *OlivettiFaces*, and obviously, it is much easier to detect cluster centers in the decision graph of MDPC+.

Fig. 7 shows the clustering results of DPC and MDPC+ with selected 44 density peaks of the top largest γ as cluster centers, where the extra 4 centers (10%) are used to increase the recall of cluster centers. Faces with the same color are in the same cluster. Cluster centers are circled in white, while the gray faces (i.e., noise) do not belong to any cluster. As shown, DPC cuts off 25% of faces as noise by applying its specific denoising method [16], thus, for a fair comparison, MDPC+ also cuts off 25% of faces with bottom confidence ξ values as noise.

According to the strong confidence of cluster centers [15], for each person, recognized faces without a cluster center are regarded as misclassified, framed in white. It can be observed that in 75% of the recognized faces, DPC misclassified 82 faces while MDPC+ only misclassified 22 faces, which demonstrates MDPC+ has better clustering performance than DPC. In addition, MDPC+ recalled

Table 3
The comparison of AMI, ARI, and FMI on real-world datasets. PAR represents the parameter setting.

Dataset	Metric	K-means	AP	SSC	DBSCAN	DPC	SSSP-DPC	SNN-DPC	PGDPC	MDPC+
Iris	AMI	0.72 (±0.01)	0.42	0.83 (±0.02)	0.58	0.86	0.88	**0.91**	0.88	0.88
	ARI	0.71 (±0.01)	0.34	0.84 (±0.03)	0.57	0.88	0.90	**0.92**	0.90	0.90
	FMI	0.81 (±0.01)	0.51	0.89 (±0.02)	0.77	0.92	0.93	**0.95**	0.93	0.93
	PAR	3	0.5	3/19	0.12/9	0.2	2	15	20	22
Wine	AMI	0.62 (±0.23)	0.36	**0.89** (±0)	0.53	0.70	0.75	0.87	0.74	0.80
	ARI	0.62 (±0.26)	0.27	**0.91** (±0)	0.45	0.67	0.74	0.90	0.73	0.82
	FMI	0.78 (±0.14)	0.46	**0.94** (±0)	0.68	0.78	0.83	0.93	0.82	0.88
	PAR	3	0.5	3/30	0.44/6	2	2	18	18	26
Breastcancer	AMI	0.61 (±0)	0.15	0.67 (±0.00)	0.26	0.41	0.34	**0.75**	0.63	0.68
	ARI	0.73 (±0)	0.09	0.79 (±0.00)	0.29	0.47	0.38	**0.85**	0.74	0.79
	FMI	0.88 (±0)	0.25	0.90 (±0.00)	0.64	0.79	0.76	**0.93**	0.88	0.90
	PAR	2	0.5	2/30	0.1/6	0.1	1	12	8	12
Parkin	AMI	0.21 (±0.00)	0.09	0.19 (±0)	0.18	0.18	0.18	0.15	0.18	**0.30**
	ARI	0.04 (±0.00)	0.03	0.15 (±0)	0.28	0.27	0.27	0.29	0.27	**0.46**
	FMI	0.59 (±0.00)	0.25	0.63 (±0)	0.81	0.81	0.81	0.80	0.81	**0.84**
	PAR	2	0.5	2/10	0.05/4	2	2	5	12	12
YTF	AMI	0.74 (±0.01)	0.53	0.75 (±0.01)	0.81	0.80	0.80	0.76	0.80	**0.83**
	ARI	0.54 (±0.02)	0.24	0.50 (±0.02)	0.71	0.59	0.58	0.52	0.60	**0.77**
	FMI	0.56 (±0.02)	0.38	0.53 (±0.01)	0.73	0.60	0.60	0.54	0.61	**0.79**
	PAR	41	0.5	41/50	0.04/10	1	2	80	80	15
USPS	AMI	0.58 (±0.03)	0.34	0.72 (±0.00)	0.38	0.52	0.74	0.61	0.75	**0.76**
	ARI	0.46 (±0.05)	0.05	0.60 (±0.00)	0.20	0.30	0.60	0.45	0.64	**0.66**
	FMI	0.52 (±0.05)	0.16	0.64 (±0.00)	0.36	0.45	0.67	0.51	0.67	**0.70**
	PAR	10	0.5	10/50	0.05/30	0.05	0.5	105	20	16
OlivettiFaces	AMI	0.71 (±0.04)	0.69	0.78 (±0.04)	0.73	0.76	0.81	0.81	0.81	**0.82**
	ARI	0.56 (±0.06)	0.62	0.66 (±0.01)	0.56	0.60	0.68	0.68	**0.69**	0.68
	FMI	0.52 (±0.05)	0.64	0.67 (±0.01)	0.57	0.62	0.69	0.69	**0.70**	0.69
	PAR	40	0.5	40/10	0.5/2	0.85	0.3	6	5	5
MNIST	AMI	0.79 (±0.05)	0.35	0.89 (±0.00)	0.56	0.71	0.61	0.77	0.82	**0.92**
	ARI	0.72 (±0.07)	0.05	0.82 (±0.00)	0.24	0.61	0.31	0.66	0.73	**0.93**
	FMI	0.76 (±0.06)	0.16	0.84 (±0.00)	0.39	0.67	0.51	0.70	0.77	**0.93**
	PAR	10	0.5	10/30	4/6	0.1	2	80	20	120

cluster centers for 37 persons, while DPC only recalled 31 persons, demonstrating that MDPC+ has higher center recall than DPC in center detection.

4.3.2. Handwritten digit recognition on the MNIST dataset

MNIST [40] is a widely used handwritten digit image database. Herein, a strong feature representation *MNIST* test set of 10,000 samples of 500 features from [15] is used. As shown in Table 3, MDPC+ did a pleasing job on the *MNIST* dataset by obtaining the highest scores: AMI = 0.92, ARI = 0.93, FMI = 0.93.

To demonstrate the performance of our confidence learning (see Section 3.3.3), digits with bottom confidence (ξ) (under different noise-cutting rates) and digits with the top confidence (15) were selected in the clustering results of MDPC+ on *MNIST*, as presented in Fig. 8. Note that the 15 digits with the top confidence are accurate (AMI = 1.00, ARI = 1.00), particularly neat, and identifiable. By contrast, digits with bottom confidence under zero noise-cutting rate are much more difficult to identify, and the corresponding clustering result (AMI = 0.68, ARI = 0.61) is less satisfying. But if we cut off 25% data points of low confidence as noise, the recognizability of the 15 digits with bottom confidence is greatly improved, as presented in the middle figure.

This verifies the effectiveness of the confidence learning of MDPC+. In real applications, it can actively cut out a part of data with the lowest confidence as noise to obtain a high-precision clustering result. Additionally, it's worth mentioning that non-negative matrix factorization-based clustering [44] and sub-space clustering [45–47] are outstanding techniques for face images clustering and handwritten digits clustering [48], due to their effective dimension-reduction techniques of data representation. So, it is enlightening about applying dimension reduction technology to further explore MDPC+'s practicability.

4.4. Robustness of center detection

The identification of cluster centers in the decision graph is a crucial step for MDPC+. Unlike DPC, MDPC+ aims to find main peaks as cluster centers and is supported by a new satellite peak attenuation technique (see Section 3.4), by which MDPC+ is more robust in center detection.

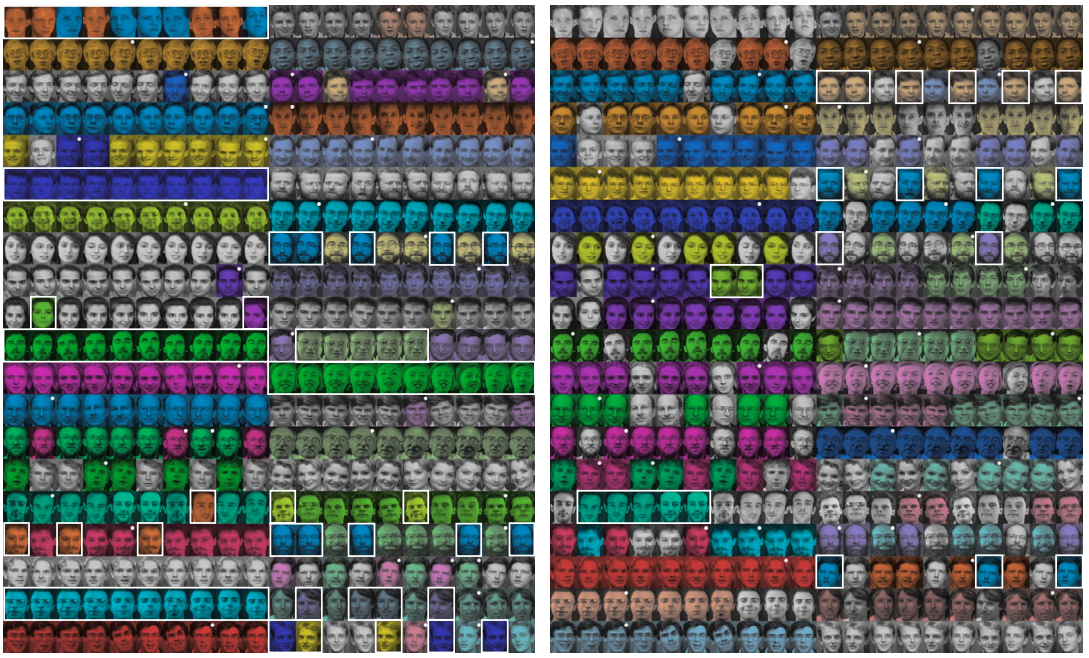

Fig. 7. The clustering results on *OlivettiFaces* by DPC and MDPC+.

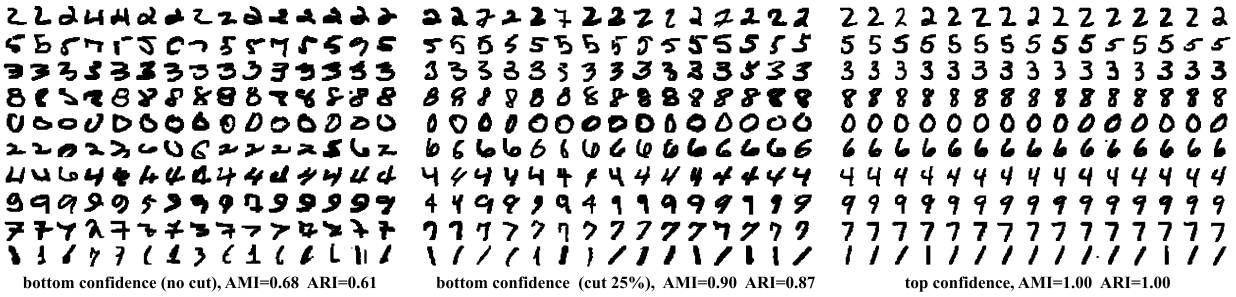

Fig. 8. The bottom 15 confidence digit images (with noise cutting rate: 0% and 25%) and the top 15 confidence digit images.

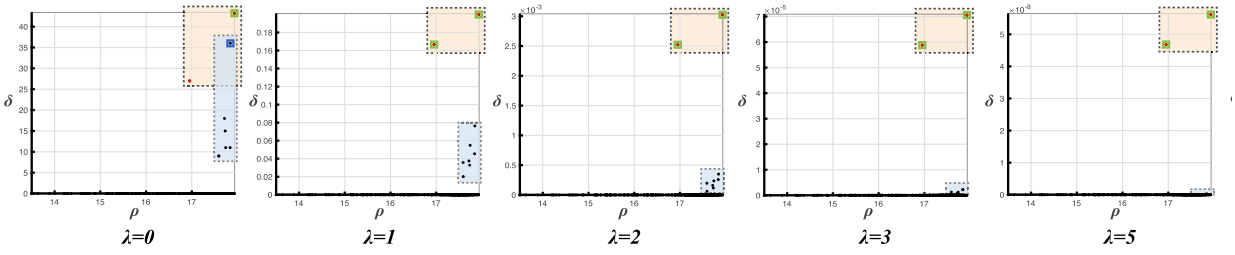

Fig. 9. The decision graphs with different λ of MDPC+ on the *Jain* dataset.

4.4.1. *Anti-satellite peak performance*

To illustrate the anti-satellite peak performance of MDPC+, Fig. 9 presents 5 decision graphs of different attenuation-coefficient λ of MDPC+ in dealing with the *Jain* dataset. As shown, without collecting density deviation information (i.e., set $\lambda = 0$), the interference of satellite peaks seriously affected the center selection. But once the density deviation information was taken into account (i.e., set $\lambda = 1$), the interference of satellite peaks was greatly reduced, standing out the two main peaks (i.e., real cluster centers, labeled by red color). As the value of λ increases, the gap between satellite peaks and the main peaks widens. For example, when $\lambda = 5$, satellite peaks almost disappeared in the decision graph, while the main peak still fell firmly in the upper right corner,

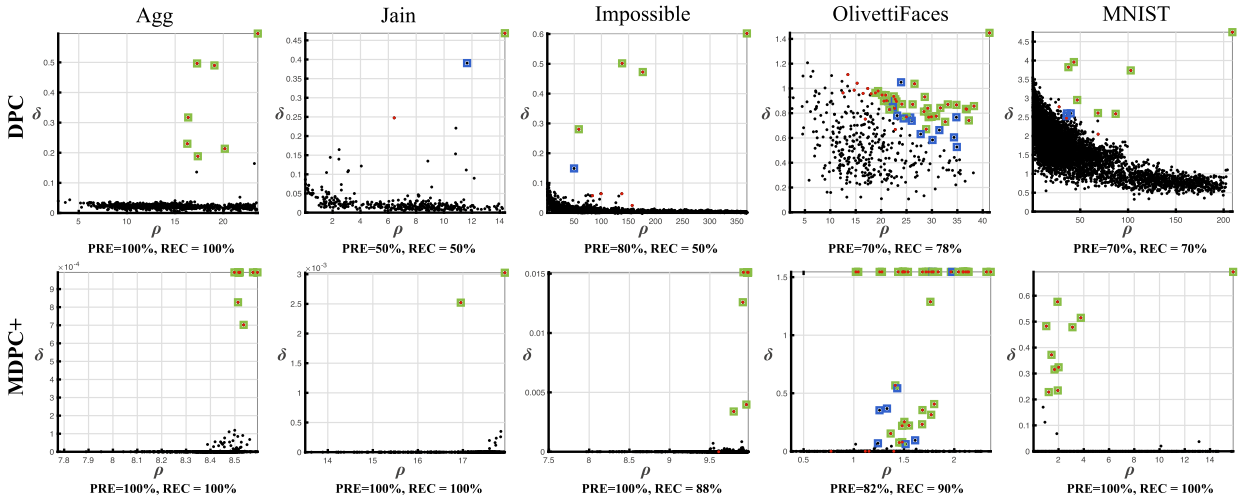

Fig. 10. The decision graphs of DPC and MDPC+ on different datasets. Cluster centers are marked in red. The green box marks the correct cluster center selection, while the blue indicates the wrong selection.

Table 4
The comparison of F1 scores of different DPC-based algorithms on tested datasets.

Algorithm	DPC	SSSP-DPC	SNN-DPC	PGDPC	MDPC+
Metric	F1	F1	F1	F1	F1
Agg	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**
Compound	0.73	0.83	0.73	0.83	**0.91**
Jain	0.50	0.50	**1.00**	**1.00**	**1.00**
Pathbased	**1.00**	**1.00**	0.67	0.67	**1.00**
S3	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**
Impossible	0.62	0.59	0.67	0.80	**0.94**
R15	**1.00**	**1.00**	**0.97**	**1.00**	**1.00**
D31	**0.93**	**1.00**	**1.00**	**1.00**	**1.00**
Iris	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**
Wine	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**
Breastcancer	0.50	0.50	**1.00**	**1.00**	**1.00**
Parkin	**0.50**	**0.50**	**0.50**	**0.50**	**0.50**
YTF	0.51	0.63	**0.68**	0.47	0.64
USPS	0.40	0.50	0.50	0.64	**0.70**
OlivettiFaces	0.74	0.83	0.78	0.81	**0.86**
MNIST	0.70	0.50	0.70	0.90	**1.00**

verifying that the satellite peak attenuation function embedded in MDPC+ can eliminate the satellite peak interference concisely and effectively.

4.4.2. The superiority of decision graph

Based on the new center assumption and anti-satellite peak function, MDPC+'s decision graph can better highlight cluster centers than DPC, indicating that MDPC+ is more robust in cluster center detection. Fig. 10 presents the decision graphs of MDPC+ and DPC on testing datasets. As shown, the decision graphs of MDPC+ are more concise than DPC's, especially for the *OlivettiFaces* and *MNIST* datasets. MDPC+ obtains much higher precision (PRE) and recall (REC) than DPC.

To further verify the superiority of MDPC+, F1-score [50] (as in Eq. (19)) was applied to conduct quantitative experiments on different decision graphs. The corresponding results are displayed in Table 4, in which MDPC+ has the highest F1 scores on all datasets except the *YTF* dataset.

$$F1 = \frac{2 \times TP}{2 \times TP + FP + FN} \tag{19}$$

To further compare the center detection robustness of MDPC+'s decision graph, we designed the γ_c-F1 plot, that is, a plot of F1 score as a function of center threshold $\gamma_c \in [\gamma_c^{min}, \gamma_c^{max}]$, where $[\gamma_c^{min}, \gamma_c^{max}]$ is the valid interval of center threshold γ_c, and points with $\gamma > \gamma_c$ are selected as centers. As known, an ideal decision graph should have a γ_c-F1 plot that owns a relatively large F1 score on a

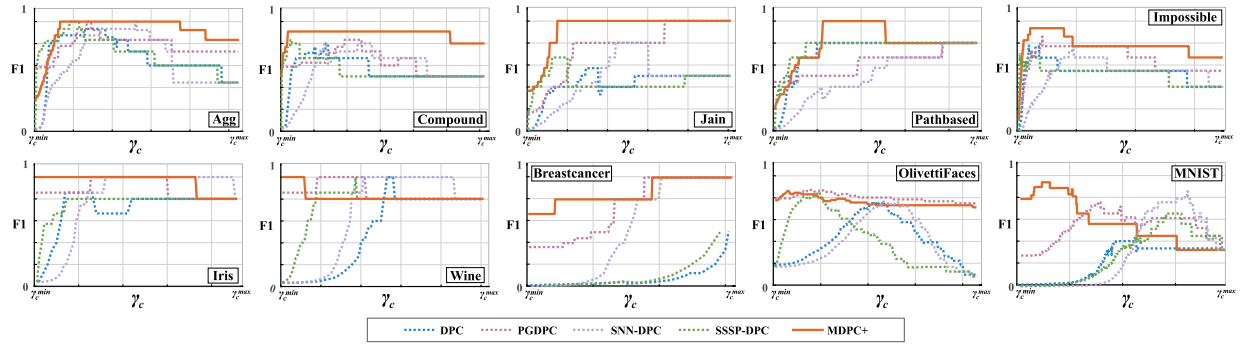

Fig. 11. The γ_c-F1 plots of different DPC-based algorithms on several datasets.

Table 5
The comparison of the runtime of different algorithms (unit: second).

Dataset (Instances)	K-means	AP	SSC	DBSCAN	DPC	SSSP-DPC	SNN-DPC	PGDPC	MDPC+
Agg (788)	0.0085	1.4849	0.5033	0.0558	0.1292	0.3483	0.6848	0.0218	0.0361
Compound (399)	0.0073	0.4476	0.2037	0.0042	0.0393	0.1013	0.2036	0.0034	0.0170
Jain (373)	0.0212	0.2815	0.242	0.0224	0.0534	0.1126	0.2545	0.0091	0.0142
Pathbased (300)	0.0032	0.1792	0.1422	0.0039	0.0399	0.1181	0.1663	0.0063	0.0091
S3 (5000)	0.0581	234.1025	5.1464	0.7526	2.5115	12.9207	26.7429	0.0395	0.1732
Impossible (3673)	0.0315	99.5429	3.7503	0.2413	1.4488	6.8049	12.1977	0.0356	0.1270
R15 (600)	0.0357	1.6074	0.4121	0.0106	0.0497	0.3829	0.5237	0.0147	0.0173
D31 (3100)	0.7494	69.0944	3.3671	0.4509	1.2265	5.8842	10.2211	0.0681	0.1051
Iris (150)	0.0105	0.0479	0.1141	0.0062	0.0077	0.0260	0.2598	0.0414	0.0105
Wine (178)	0.0127	0.0619	0.1121	0.0185	0.0056	0.0471	0.0999	0.0054	0.0080
Breastcancer (569)	0.0141	0.6029	0.4440	0.0131	0.0461	0.1736	0.5643	0.0232	0.0369
Parkin (195)	0.0282	0.0724	0.1170	0.0012	0.0190	0.0590	0.0436	0.0071	0.0095
YTF (10000)	0.2371	1437.9252	26.632	1.4860	11.6643	85.9085	121.6136	0.7101	0.6590
USPS (11000)	0.0992	9048.9395	30.5068	1.9598	15.8159	92.3958	171.5179	0.6996	1.1347
OlivettiFaces (400)	0.0408	0.2887	0.2042	0.0055	0.0840	0.1493	0.1883	0.0132	0.0312
MNIST (10000)	1.2577	565.0760	47.7253	3.1544	27.8584	77.1532	144.1165	0.6075	1.0979
Total time	2.6152	11459.7549	119.6226	8.1864	60.9993	282.5855	489.3985	2.3060	3.4867

Table 6
The time complexity of algorithms. T indicates iteration times.

K-means [9]	$O(nN_cT)$	AP [11]	$O(n^2T)$	SSC [41]	$O(n^2)$
DBSCAN [12]	$O(n\log(n))$	DPC [16]	$O(n^2)$	SSSP-DPC [17]	$O(n^2)$
SNN-DPC [23]	$O(k+N_c)n^2$	PGDPC [18]	$O(n\log(n)+\tilde{k}n)$	MDPC+ (ours)	$O(n\log(n)+kn)$

relatively large continuous γ_c-interval. Fig. 11 presents the normalized γ_c-F1 plots of different decision graphs on several datasets. As shown, the performance of MDPC+ (orange line) is the most robust.

The above experiments verified the higher robustness of MDPC+ in cluster center detection compared with traditional DPC and the state-of-the-art DPC variant algorithms.

4.5. Running speed

As analyzed in Section 3.6, MDPC+ can run fast by applying some fast kNN search techniques [49].

Table 5 demonstrates the runtime of algorithms on different datasets, and Table 6 lists the time complexity. As shown, MDPC+ is much faster than other density-based algorithms except for PGDPC, and it can run a dataset of 10,000 data points in about one second. PGDPC with $O(n\log(n)+\tilde{k}n)$ is a little faster than MDPC+. Because PGDPC is also based on kNN-graph and its allocation strategy with $O(\tilde{k}n)$ is fast than MDPC+'s $O(kn)$, $\tilde{k} \leqslant k$, where $\tilde{k}$ (an average concept) indicates that each non-peak point can find its $\tilde{k}$-th nearest neighbor as the nearest higher density point. However, PGDPC's clustering accuracy is inferior to that of MDPC+. SNN-DPC and SSSP-DPC, as excellent improved methods of DPC, are more time-consuming than DPC. While K-means with $O(nN_cT)$ owns the fastest speed. Because the cluster numbers (N_c) of all tested datasets are small, which allows K-means to achieve convergence in a small number of iterations, i.e., a small T value.

To further verify the fast speed of MDPC+, experiments were launched as in Table 7, in which the runtime of K-means and MDPC+ on the *Birchrg1* [35] dataset of 100,000 points with setting different cluster numbers are presented. As shown, K-means is slower when N_c turns larger, while MDPC+ has a stable speed no matter how N_c changes. Note that, when dealing with the *Birchrg1*

Table 7
The comparison of runtime of K-means and MDPC+ with setting different number of clusters on the *Birchrg1* dataset (unit: second).

Algorithm	Metric	$N_c = 500$	$N_c = 100$	$N_c = 300$	$N_c = 200$	$N_c = 100$
K-means	runtime	30.5909 (± 3.7202)	27.0610 (± 2.8171)	25.5954 (± 2.5675)	16.2591 (± 0.9817)	7.5152 (± 1.7920)
	iterations (T)	124 (± 16)	131 (± 32)	124 (± 34)	(107 ± 44)	(72 ± 29)
MDPC+ ($k = 40$)	runtime	4.7278 (± 0.1719)	4.7479 (± 0.2092)	4.65695 (± 0.1291)	4.67915 (± 0.1788)	4.77135 (± 0.1639)
	iterations (T)	1	1	1	1	1

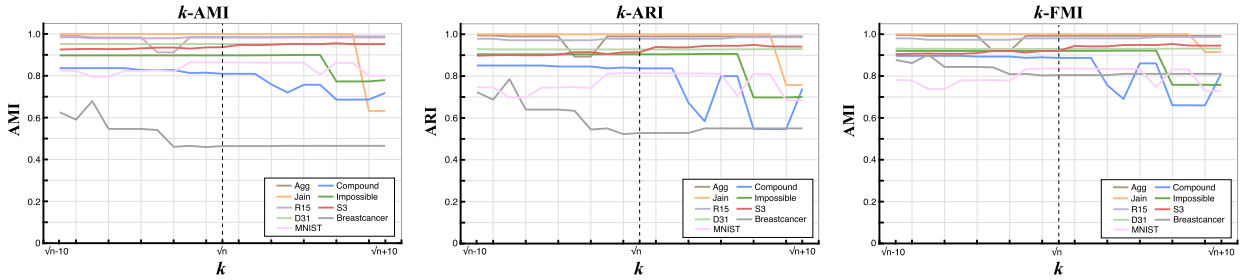

Fig. 12. The k-AMI, k-ARI, and k-FMI plots of MDPC+ on different datasets with different parameter k.

dataset with a large number of clusters (such as setting $N_c = 500$), K-means is more time-consuming than MDPC+. So, MDPC+ is more promising for large-scale clustering.

4.6. Parameter k

In MDPC+, the parameter k, $k = \sqrt{n}$, is used for graph construction, density estimation, density peak detection, and weight evaluation, so the performance of the MDPC+ algorithm is highly dependent on k.

Fig. 12 shows the k-AMI, k-ARI, and k-FMI plot on 10 different datasets with $k \in [\sqrt{n} - 10, \sqrt{n} + 10]$. As shown, the overall performance of MDPC+ is robust to changes in k, especially for large datasets. Meanwhile, MDPC+ works well at $k = \sqrt{n}$.

This verifies the parameter insensitivity and the effectiveness of the parameter setting of the MDPC+ algorithm.

5. Conclusion

In this work, a Main Density Peak Clustering algorithm (MDPC+) is proposed following a new center assumption that views main peaks as cluster centers. Meanwhile, through the exclusion of non-peaks and the attenuation of satellite peaks, the detection of cluster centers is more accurate and easy. Our allocation strategy based on digraph structures can accurately assign non-peaks and satellite peaks. In addition, MDPC+ only requires kNN distances of data points as input, so it can run fast and is suitable for large data clustering. The center detection robustness, the clustering accuracy, and the running speed of MDPC+ are well verified in the conducted comparative experiments on synthetic datasets and real-world datasets, as well as its application to the face recognition of *OlivettiFaces* and the handwritten digital recognition of *MNIST*.

To be noted, the clustering performance of MDPC+ is based on the quality of the allocation of non-peaks, and the latter is highly relied on the design of edge-weight function $w_{\bar{P}}(\cdot)$ (see Eq. (8)). In terms of $w_{\bar{P}}(\cdot)$ design, there is still room for improvement (as a part of our future work). Benefiting from the clear decision graph, MDPC+'s center detection is outstanding, but it still relies on manual operation. However, in many real applications, automatic cluster detection is needed. Therefore, in future work, we intend to improve MDPC+ to realize the automatic detection of cluster centers. Besides, to further expand the applications of MDPC+ on high-dimensional data, we will seek some effective dimension-reduction techniques of data representation from non-negative matrix factorization-based clustering [44] and sub-space clustering [46].

CRediT authorship contribution statement

Junyi Guan: Conceptualization, Methodology, Software. **Sheng Li:** Conceptualization, Validation. **Xiongxiong He:** Supervision. **Jiajia Chen:** Writing – review & editing.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

Data will be made available on request.

Acknowledgement

This work was supported by the National Science Foundation of P.R. China (Grant: 62233016) and Key R&D Program Projects in Zhejiang Province: 2020C03074.

References

[1] M. Gao, G.Y. Shi, Ship-handling behavior pattern recognition using AIS sub-trajectory clustering analysis based on the T-SNE and spectral clustering algorithms, Ocean Eng. 205 (2020) 106919.
[2] M. Paolanti, E. Frontoni, Multidisciplinary pattern recognition applications: a review, Comput. Sci. Rev. 37 (2020) 100276.
[3] G.B. Coleman, H.C. Andrews, Image segmentation by clustering, Proc. IEEE 67 (5) (1979) 773–785.
[4] H. Zhang, H. Li, N. Chen, et al., Novel fuzzy clustering algorithm with variable multi-pixel fitting spatial information for image segmentation, Pattern Recognit. 121 (2022) 108201.
[5] T. Lei, P. Liu, X. Jia, et al., Automatic fuzzy clustering framework for image segmentation, IEEE Trans. Fuzzy Syst. 28 (9) (2019) 2078–2092.
[6] M.I. Jordan, T.M. Mitchell, Machine learning: trends, perspectives, and prospects, Science 349 (6245) (2015) 255–260.
[7] R. Achanta, S. Susstrunk, Superpixels and polygons using simple non-iterative clustering, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2017, pp. 4651–4660.
[8] A.K. Jain, R.C. Dubes, Algorithms for Clustering Data, Prentice-Hall, Inc., 1988.
[9] J. MacQueen, Classification and analysis of multivariate observations, in: 5th Berkeley Symp. Math. Statist. Probability, 1967, pp. 281–297.
[10] H.S. Park, C.H. Jun, A simple and fast algorithm for K-medoids clustering, Expert Syst. Appl. 36 (2) (2009) 3336–3341.
[11] B.J. Frey, D. Dueck, Clustering by passing messages between data points, Science 315 (5814) (2007) 972–976.
[12] M. Ester, H.P. Kriegel, J. Sander, et al., A density-based algorithm for discovering clusters in large spatial databases with noise, KDD 96 (34) (1996) 226–231.
[13] K. Sawant, Adaptive methods for determining DBSCAN parameters, Int. J. Innov. Sci. Eng. Technol. 1 (4) (2014) 329–334.
[14] A. Karami, R. Johansson, Choosing DBSCAN parameters automatically using differential evolution, Int. J. Comput. Appl. 91 (7) (2014) 1–11.
[15] H. Averbuch-Elor, N. Bar, D. Cohen-Or, Border-peeling clustering, IEEE Trans. Pattern Anal. Mach. Intell. 42 (7) (2019) 1791–1797.
[16] A. Rodriguez, A. Laio, Clustering by fast search and find of density peaks, Science 344 (6191) (2014) 1492–1496.
[17] D.U. Pizzagalli, S.F. Gonzalez, R. Krause, A trainable clustering algorithm based on shortest paths from density peaks, Sci. Adv. 5 (10) (2019), eaax3770.
[18] J. Guan, S. Li, X. He, et al., Peak-graph-based fast density peak clustering for image segmentation, IEEE Signal Process. Lett. 28 (2021) 897–901.
[19] 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.
[20] J. Xie, H. Gao, W. Xie, et al., Robust clustering by detecting density peaks and assigning points based on fuzzy weighted K-nearest neighbors, Inf. Sci. 354 (2016) 19–40.
[21] M. Abbas, A. El-Zoghabi, A. Shoukry, DenMune: density peak based clustering using mutual nearest neighbors, Pattern Recognit. 109 (2021) 107589.
[22] M. Du, S. Ding, X. Xu, et al., Density peaks clustering using geodesic distances, Int. J. Mach. Learn. Cybern. 9 (8) (2018) 1335–1349.
[23] R. Liu, H. Wang, X. Yu, Shared-nearest-neighbor-based clustering by fast search and find of density peaks, Inf. Sci. 450 (2018) 200–226.
[24] C. Wiwie, J. Baumbach, R. Röttger, Comparing the performance of biomedical clustering methods, Nat. Methods 12 (11) (2015) 1033–1038.
[25] J. Guan, S. Li, X. He, et al., Fast hierarchical clustering of local density peaks via an association degree transfer method, Neurocomputing 455 (2021) 401–418.
[26] A.K. Jain, Data clustering: 50 years beyond K-means, Pattern Recognit. Lett. 31 (8) (2010) 651–666.
[27] D. Comaniciu, P. Meer, Mean shift: a robust approach toward feature space analysis, IEEE Trans. Pattern Anal. Mach. Intell. 24 (5) (2002) 603–619.
[28] E.W. Dijkstra, A note on two problems in connexion with graphs, in: Edsger Wybe Dijkstra: His Life Work, and Legacy, 2022, pp. 287–290.
[29] A. Gionis, H. Mannila, P. Tsaparas, Clustering aggregation, ACM Trans. Knowl. Discov. Data 1 (1) (2007) 4-es.
[30] C.T. Zahn, Graph-theoretical methods for detecting and describing gestalt clusters, IEEE Trans. Comput. 100 (1) (1971) 68–86.
[31] A.K. Jain, M.H.C. Law, Data Clustering: A User's Dilemma, International Conference on Pattern Recognition and Machine Intelligence, Springer, Berlin, Heidelberg, 2005, pp. 1–10.
[32] H. Chang, D.Y. Yeung, Robust path-based spectral clustering, Pattern Recognit. 41 (1) (2008) 191–203.
[33] P. Fränti, O. Virmajoki, Iterative shrinking method for clustering problems, Pattern Recognit. 39 (5) (2006) 761–775.
[34] C.J. Veenman, M.J.T. Reinders, E. Backer, A maximum variance cluster algorithm, IEEE Trans. Pattern Anal. Mach. Intell. 24 (9) (2002) 1273–1280.
[35] T. Zhang, R. Ramakrishnan, M. Livny, BIRCH: a new data clustering algorithm and its applications, Data Min. Knowl. Discov. 1 (2) (1997) 141–182.
[36] K. Bache, M. Lichma, UCI machine learning repository, [online] available: http://archive.ics.uci.edu/ml, 2013.
[37] L. Wolf, T. Hassner, I. Maoz, Face recognition in unconstrained videos with matched background similarity, in: CVPR 2011, IEEE, 2011, pp. 529–534.
[38] D. Keysers, T. Deselaers, C. Gollan, et al., Deformation models for image recognition, IEEE Trans. Pattern Anal. Mach. Intell. 29 (8) (2007) 1422–1435.
[39] F.S. Samaria, A.C. 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.
[40] L. Deng, The mnist database of handwritten digit images for machine learning research [best of the web], IEEE Signal Process. Mag. 29 (6) (2012) 141–142.
[41] L. Zelnik-Manor, P. Perona, Self-Tuning Spectral Clustering, Advances in Neural Information Processing Systems, 2004.
[42] N.X. Vinh, J. Epps, J. Bailey, Information theoretic measures for clusterings comparison: is a correction for chance necessary?, in: Proceedings of the 26th Annual International Conference on Machine Learning, 2009, pp. 1073–1080.
[43] E.B. Fowlkes, C.L. Mallows, A method for comparing two hierarchical clusterings, J. Am. Stat. Assoc. 78 (383) (1983) 553–569.
[44] C. Peng, Z. Kang, M. Yang, et al., Feature selection embedded subspace clustering, IEEE Signal Process. Lett. 23 (7) (2016) 1018–1022.
[45] C. Peng, Z. Kang, H. Li, et al., Subspace clustering using log-determinant rank approximation, in: Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2015, pp. 925–934.
[46] E. Elhamifar, R. Vidal, Sparse subspace clustering: algorithm, theory, and applications, IEEE Trans. Pattern Anal. Mach. Intell. 35 (11) (2013) 2765–2781.
[47] C. Peng, Z. Kang, Q. Cheng, Subspace clustering via variance regularized ridge regression, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2017, pp. 2931–2940.
[48] C. Peng, Q. Zhang, Z. Kang, et al., Kernel two-dimensional ridge regression for subspace clustering, Pattern Recognit. 113 (2021) 107749.
[49] N. Bhatia, Survey of nearest neighbor techniques, arXiv preprint arXiv:1007.0085, 2010.
[50] A. Patil, D. Huard, C.J. Fonnesbeck, PyMC: Bayesian stochastic modelling in Python, J. Stat. Softw. 35 (4) (2010) 1.

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

Sheng Li received 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.

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.