DEMOS: Clustering by Pruning a Density-Boosting Cluster Tree of Density Mounts

Junyi Guan, Sheng Li, Xiaojun Chen, Xiongxiong He, and Jiajia Chen

***Abstract*—Most existing clustering algorithms require presetting cluster number and often fail to capture complex shapes. Herein, we propose a clustering algorithm by pruning a density-boosting cluster tree of density mounts—DEnsity MOuntains Separation clustering algorithm (DEMOS). A cluster is assumed to be a density-connected area with multiple (or a single) density mounts (i.e., single-peak clusters) and a relatively large dis-connectivity from density-connected areas of higher densities. Based on this assumption, DEMOS can easily detect the number of clusters and robustly reconstruct their complex shapes. It first builds the dataset into a peak graph, where each density peak represents a density mount. A multi-valley-link-based connectivity estimation method is embedded to efficiently estimate the connectivity between density peaks during peak graph building. Then, by applying a new linkage metric designed based on our assumption, DEMOS builds density mounts into a reasonably density-boosting cluster tree. After obtaining a robust center detection in a clarity-enhancing decision graph (i.e., a two-dimensional plot for detecting centers), DEMOS prunes the cluster tree into final clusters to finish clustering. Experimental results on both synthetic and real datasets demonstrated the effectiveness of DEMOS and its applicability to large-scale data clustering.**

***Index Terms*—Clustering, complex shape clustering, cluster number detection, density peak.**

I. Introduction

CLUSTERING that aims to group similar objects is a vital unsupervised learning technique for data mining [1], [2], which has been applied to different areas, such as computer vision [3], [4], pattern recognition [5], [6], image processing [7], [8], machine learning [9].

An ideal clustering can reasonably identify the number of clusters and accurately reconstruct cluster shapes. Different clustering methods have been proposed based on specific assumptions regarding the nature of a "cluster" [9], which can be classified as partitioning, hierarchical, density-based, graph-based, etc [11].

Manuscript received 13 May 2022; revised 3 January 2023; accepted 8 April 2023. Date of publication 12 April 2023; date of current version 15 September 2023. This work was supported in part by the National Science Foundation of P.R. China under Grant 62233016 and in part by Zhejiang Science Foundation under Grant 2020C03074. Recommended for acceptance by X. Chen. *(Corresponding author: Sheng Li.)*

Junyi Guan, Sheng Li, Xiongxiong He, and Jiajia Chen are with the College of Information Engineering, Zhejiang University of Technology, Hangzhou 310023, China (e-mail: jonnyguan73@163.com; shengli@zjut.edu.cn; hxx@zjut.edu.cn; fl_katrina@163.com).

Xiaojun Chen is with the Department of Endocrinology, The First Affiliated Hospital of Wenzhou Medical University, Wenzhou 325000, China (e-mail: chenxj@medmail.com.cn).

Digital Object Identifier 10.1109/TKDE.2023.3266451

K-means [12] is one of the most popular partitioning clustering methods. It aims to partition a dataset into K clusters to minimize the total distance sum of points to cluster centroids (centers). Despite the simplicity and efficiency of K-means in clustering hyperspherical shapes, it is unsuitable for non-spherical shapes [10]. Kernel K-means [13], [14], [15] can capture non-spherical shapes by applying kernel methods to transform non-spherical shapes into linearly separable spherical shapes. Nevertheless, these K-centers techniques require presetting the number K of clusters. Some density-based clustering methods that view a cluster as a set of maximum density-connected points can reconstruct non-spherical shapes of clusters without presetting the number of clusters [16], [17], [18]. But these methods may possibly merge highly overlapping clusters [19].

Hierarchical clustering that aims to build a cluster tree (i.e., a dendrogram) according to a specific (dis)similarity matrix of data is suitable for capturing non-spherical shapes [10]. One of the most classic hierarchical clustering methods is linkage-based clustering, which builds a cluster tree according to a specific linkage metric.

For example, Single-linkage [20] views the minimum member distance between clusters as the linkage metric can detect link shapes but is sensitive to outliers, while Complete-linkage [21] is insensitive to outliers and views the maximum member distance between clusters as the linkage metric. Although these methods effectively identify non-spherical shapes, a given number of clusters is required to prune the cluster tree, similar to the parameter K in K-center techniques.

In 2014, *Science* published a remarkable linkage-based clustering algorithm—the density peak clustering algorithm (DPC) [22]. DPC builds a cluster tree according to its special density-distance-based linkage metric. It assumes that a cluster center is a density peak that should have a higher density ρ than its surrounding points and have a far distance δ away from higher density areas. Based on its assumption, DPC cuts the cluster tree into clusters via a decision graph (i.e., a ρ-δ plot for detecting centers). For its excellent detection of the number of clusters and reconstruction of non-spherical shapes, DPC is recognized as a promising and concerned clustering algorithm [25], [26].

Nevertheless, DPC's linkage metric is unsuitable for complex non-spherical shapes reconstruction [29], and its decision graph may unclearly display the real cluster centers in dealing with multi-peak clusters. In other words, DPC's center detection method is not robust for multi-peak clusters [27], [28]. Although different improved methods [29], [30], [31], [32], [33], [34],

TABLE I
NOTATIONS AND DEFINITIONS USED IN THIS PAPER

Symbol	Definition
$X = \{x_1, x_2, \ldots, x_n \mid x_i \in \mathbb{R}^d\}$	The dataset with x_i as its i-th point.
$P_X = \{p_1, p_2, \ldots, p_{n_p}\}$	The set of density peaks of dataset X.
$\bar{P}_X = X \setminus P$	The set of normal points of dataset X.
$N_k(x_i)$	The k nearest neighbors of point x_i
$C^{(t)} = \{C_1, C_2, \ldots, C_{n-t}\}$	The cluster set of the t-th iteration, where C_1 is a cluster of data points.
$\mathbf{C}^{(t)} = \{\mathbf{C}_1, \mathbf{C}_2, \ldots, \mathbf{C}_{n_p-t}\}$	The cluster set of density peaks of the t-th iteration, where $\mathbf{C}_1$ is a cluster of density peaks.
$ct(C)$	The center (the highest density point) of cluster C, see Eq. (5).
$mt(p)$	The density mount (the single peak cluster) of denisty peak p.
d_{ij}	The Euclidean distance between points x_i and x_j.
$\hat{d}_{p_i p_j}$	The dissimilarity between cluster centers p_i and p_j, see Eq. (17).
$d^*_{p_i p_j}$	The dis-connectivity between density peaks p_i and p_j, see Eq. (15).
$s^*_{p_i p_j}$	The connectivity between density peaks p_i and p_j, see Eq. (22).
D_s	The linkage metric of Single-linkage [20] see Eq. (1).
D_d	The linkage metric DPC [22], see Eq. (6).
D_p	The linkage metric of DEMOS (ours), see Eq. (8).
$G(X, E, w)$	The weighted complete graph of dataset X.
$G_d(X, E_d, w_d)$	The weighted complete digraph of dataset X according to density boosting.
$G_p(P_X, E_p, w_p)$	The weighted complete digraph of density peaks P according to density boosting.
$G_p^S(P_X, E_p^S, w_p^S)$	The peak graph of dataset X.
T_G	The minimum spanning tree (MST) of graph $G(X, E, w)$, see Eq. (2).
T_{G_d}	The MST of digraph $G_d(X, E, w_d)$, see Eq. (7).
T_{G_p}	The MST of digraph $G_p(P_X, E_p, w_p)$, see Eq. (9).
$\Gamma_{p_i p_j}$	The shortest path between density peaks p_i and p_j in peak graph $G_p^S(P_X, E_p^S, w_p^S)$, see see Eq. (16).
$\rho = \{\rho_1, \rho_2, \ldots, \rho_n\}$	The local density of data, see Eq. (3) and (10).
$\delta = \{\delta_1, \delta_2, \ldots, \delta_n\}$	The distance from nearest higher density point, see Eq. (4).
$\delta_p = \{\delta_{p_1}, \delta_{p_2}, \ldots, \delta_{p_{n_p}}\}$	The distance (of peaks) from nearest higher density peaks, see Eq. (27).
$\hat{\delta}_p = \{\hat{\delta}_{p_1}, \hat{\delta}_{p_2}, \ldots, \hat{\delta}_{p_{n_p}}\}$	The clarity-enhancing distance (of peaks) from nearest higher density peaks, see Eq. (33).
$\hat{\rho}_p = \{\hat{\rho}_{p_1}, \hat{\rho}_{p_2}, \ldots, \hat{\rho}_{p_{n_p}}\}$	The clarity-enhancing density values of density peaks, see Eq. (32).
$\theta(x_i, x_j)$	The representativeness value of x_i to represent x_j, see Eq. (19).
θ_i	The peak-representativeness of point $x_i \in mt(p)$, see Definition 4.
v_i	The valley link v_i of point x_i, see Eq. (13).
s_v	The connectivity message of a valley link v, see Eq. (18).
$\mathbf{s}_{p_i p_j} = \{s_{[1]}, s_{[2]}, \ldots, s_{[n_\mathbf{s}]}\}$	The connectivity message vector between density peaks p_i and p_j, see Eq. (21).
$\mathbf{w}_{p_i p_j} = \{w_{[1]}, w_{[2]}, \ldots, w_{[n_\mathbf{s}]}\}$	The weight vector of $\mathbf{s}_{p_i p_j}$, see Eq. (23).

[35], [36], [37] were proposed, based on DPC's cluster center assumption, their cluster center detections were still not robust when dealing with multi-peak clusters. Note that most DPC-based methods are unsuitable for large-scale datasets due to high time-consuming, like most hierarchical clustering algorithms.

Herein, a clustering algorithm by pruning a density-boosting cluster tree of density mounts—DEnsity MOuntains Separation clustering algorithm (DEMOS[1]) is proposed, which executes linkage-based clustering on a sparse density peak graph (hereinafter, a peak graph). DEMOS views a cluster as a density-connected area with multiple (or a single) density mounts and a relatively large dis-connectivity from density-connected areas of higher densities. A density mount refers to a single-peak local cluster, i.e., a local cluster with only one density peak as the cluster center. DEMOS can fast capture complex non-spherical shapes and has robust center detection performance when dealing with multi-peak clusters. The main contributions of DEMOS are as follows:

1) A cluster is assumed to be a density-connected area with multiple (or a single) density mounts and a relatively large dis-connectivity from density-connected areas of higher densities. So, DEMOS can reasonably reconstruct complex shapes to obtain a robust center detection performance by clearly displaying the real cluster centers in our decision graph.
2) A novel linkage metric is designed to build a cluster tree of density peaks, and a multi-valley-link-based connectivity estimation method to achieve a fast connectivity estimation of density peaks with high fidelity.
3) DEMOS obtains robust performance in center detection, because our decision graph can better highlight cluster centers by further expanding the difference between centers and non-centers.
4) DEMOS is suitable for large-scale data clustering with only requiring kNN distances of data as input.

The rest paper is composed as: Section II is the related works; Section III mainly focuses on the proposed DEMOS algorithm; Section IV displays the experiment and discussion; and Section V gives the final conclusion.

II. RELATED WORKS

A. Notations

Table I lists the major symbols and notations used in the following parts.

B. Linkage-Based Clustering

Given a dataset $X = \{x_1, x_2, \ldots, x_n \mid x_i \in \mathbb{R}^d\}$, and let $C^{(t)} = \{C_1, C_2, \ldots, C_{n-t}\}$ denote the cluster set of the t-th iteration, where C_y means y-th cluster in $C^{(t)}$, $\bigcap_{y=1}^{n-t} C_y = \varnothing, \bigcup_{y=1}^{n-t} C_y = X$, and $C^{(0)}$ means the initial cluster set.

[1]The code is available at https://github.com/Guanjunyi/DEMOS

Linkage-based clustering first views each point as a singleton cluster, i.e., $t = 0$, and then iteratively merges the most similar cluster pair into one cluster according to a specific linkage metric, i.e., $C^{(t)} \leftarrow C^{(t+1)}, t \leftarrow t+1$; finally, when all data are merged into a given number n_c of clusters, i.e, $t = n - n_c$, clustering is done. For exmaple, at t-th iteration, the most similar cluster pair C_i and $C_j \in C^{(t)} = \{C_i, C_j, \ldots\}$ are merged into one cluster $C_* = \{C_i, C_j\}$, then have $C^{(t+1)} = (C^{(t)} \setminus \{C_i, C_j\}) \cup C_*$, indicating that $|C^{(t)}| - |C^{(t+1)}| = 1$ (where $|\cdot|$ denotes the number of elements in one set). Subsequently, cluster set $C^{(t)}$ and iteration number t are updated to: $C^{(t)} \leftarrow C^{(t+1)}, t \leftarrow t+1$.

The core of a linkage-based method is its linkage metric, that is, a specific pair-wise dissimilarity measure between two clusters.

1) Single-Linkage Clustering: Single-linkage clustering [20] is one of the most popular linkage-based clustering techniques. It treats the minimum membership distance between clusters as their dissimilarity, that is, the linkage metric of Single-linkage, denoted as D_s, as defined in (1), where $d_{ij} = ||x_i - x_j||_2$ indicates the euclidean distance between points x_i and x_j.

$$D_s(C_y, C_z) = \min_{x_i \in C_y, x_j \in C_z} d_{ij} \tag{1}$$

Let $G(X, E, w)$ be a weighted complete graph of dataset X, where $E = \{e_{ij} | x_i, x_j \in X\}$, and edge weight function $w : E \to \mathbb{R}_+, w(e_{ij}) = d_{ij}$. Single-linkage essentially builds a minimum spanning tree (MST) T_G of graph G, as in (2), and then cuts the tree into a given number of clusters with the minimum sum of weights. Where $\mathscr{T}(G)$ means all spanning trees of graph G, and $E(T)$ means all edges in tree T.

$$T_G = \arg\min_{T \in \mathscr{T}(G)} \sum_{e \in E(T)} w(e) \tag{2}$$

2) Density Peak Clustering: DPC [22] inherits the main idea of Mean-shift [23]—to search local density maxima (density peaks) as cluster centers, but unlike Mean-shift that views all density peaks as centers, DPC selects appropriate centers based on its center assumption.

As a linkage-based clustering method with a specific density-distance-based linkage metric, DPC calculates point x_i's local density ρ_i and distance δ_i from its nearest higher density point, as in (3) and (4), respectively, where the "cutoff distance" d_c is user-specified [22]. For the highest density point x_i, its $\delta_i = \max_{x_j}(d_{ij})$.

$$\rho_i = \sum_{x_j \in X} \chi(d_{ij} - d_c), \quad \chi(z) = \begin{cases} 1 & z < 0 \\ 0 & z \geqslant 0 \end{cases} \tag{3}$$

$$\delta_i = \min_{x_j} (d_{ij}), \text{s.t. } \rho_j > \rho_i \tag{4}$$

Except for the selected n_c cluster centers with top $\gamma (\gamma = \rho \cdot \delta)$ values, DPC associates each non-center point along its δ path (i.e., each non-center point is associated with its nearest higher density point). In fact, DPC's allocation idea–to connect all data points into a single tree–is the same as that of Quick-shift [24] (a fast variant of Mean-shift), but the difference lies in how to prune the tree into final clusters.

For a cluster C_y, DPC considers the minimum distance from its cluster center $ct(C_y)$ (i.e., the highest density point in cluster C_y, as in (5)) to a higher density point of another cluster C_z as their dissimilarity. The minimum distance herein is called "the minimum center-boosting distance". Besides, DPC additionally adds center density ρ as a penalty term to resist the interference of outliers (noise). Thus, the linkage metric of DPC is defined as in (6).

$$ct(C_y) = \arg\max_{x_i \in C_y} (\rho_i) \tag{5}$$

$$D_d(C_y, C_z) = \min_{x_i, x_j} (\rho_i \cdot d_{ij})$$
$$\text{s.t. } x_i = ct(C_y), x_j \in C_z, \rho_j > \rho_i \tag{6}$$

Equation (6) implies that the dissimilarity evaluation between clusters is unidirectional, in other words, DPC can only evaluate the dissimilarity from a cluster to other higher density clusters.

$$T_{G_d} = \arg\min_{T \in \mathscr{T}(G_d)} \sum_{e \in E(T)} w_d(e) \tag{7}$$

Let $G_d(X, E_d, w_d)$ be a weighted complete digraph of dataset X according to density boosting, where $E_d = \{e_{ij} | \rho_i < \rho_j, x_i, x_j \in X\}$, and edge weight function $w_d : E_d \to \mathbb{R}_+, w_d(e_{ij}) = \rho_i \cdot d_{ij}$. Similar to Single-linkage, DPC also builds a MST T_{G_d} of digraph G_d, as in (7), and then cuts the tree into a given number of clusters with the minimum sum of weights. As a result, all edges in T_{G_d} are exactly δ paths. Also, DPC applies a decision graph [22] to assist in the detection of cluster number.

3) Comparison of DPC and Single-Linkage: Although DPC and Single-linkage both aim to generate a MST of the dataset, DPC builds a density-boosting MST, constraining data points to only associate with higher-density areas.

Single-linkage's metric D_s (see (1)) focuses on the minimum gap (connectivity) between clusters, so Single-linkage tends to build link structures and is sensitive to outliers; while DPC's metric D_d (see (6)) adds density-boosting as a constraint, so DPC normally builds density-boosting link structures and is insensitive to outliers. But D_d ignores the gap between clusters, causing DPC to mistakenly merge clusters with a big gap (i.e., without density-connectivity) [27], [29], resulting in poor clustering.

Fig. 1 presents the limitations of Single-linkage and DPC on toy datasets *D1* and *D2*. As shown in *D1* dataset, the gap between cluster B and C is narrower, and cluster (point) O is an outlier. Single-linkage captured the gap differences, i.e., $D_s(O, B) > D_s(C, B)$, and divided them into cluster $\{O\}$ and cluster $\{B, C\}$. Obviously, to classify outlier O as a separate cluster is unreasonable. In contrast, DPC obtained $D_d(O, B) < D_d(C, B)$ and divided *D1* dataset into clusters $\{O, B\}$ and $\{C\}$. With the density-boosting constraint, DPC no longer relies on the minimum gap between clusters to evaluate the dissimilarity, which effectively removes the interference of outliers.

Nevertheless, D_d's ignoring of the minimum gap between clusters may cause DPC to mistakenly merge clusters with a

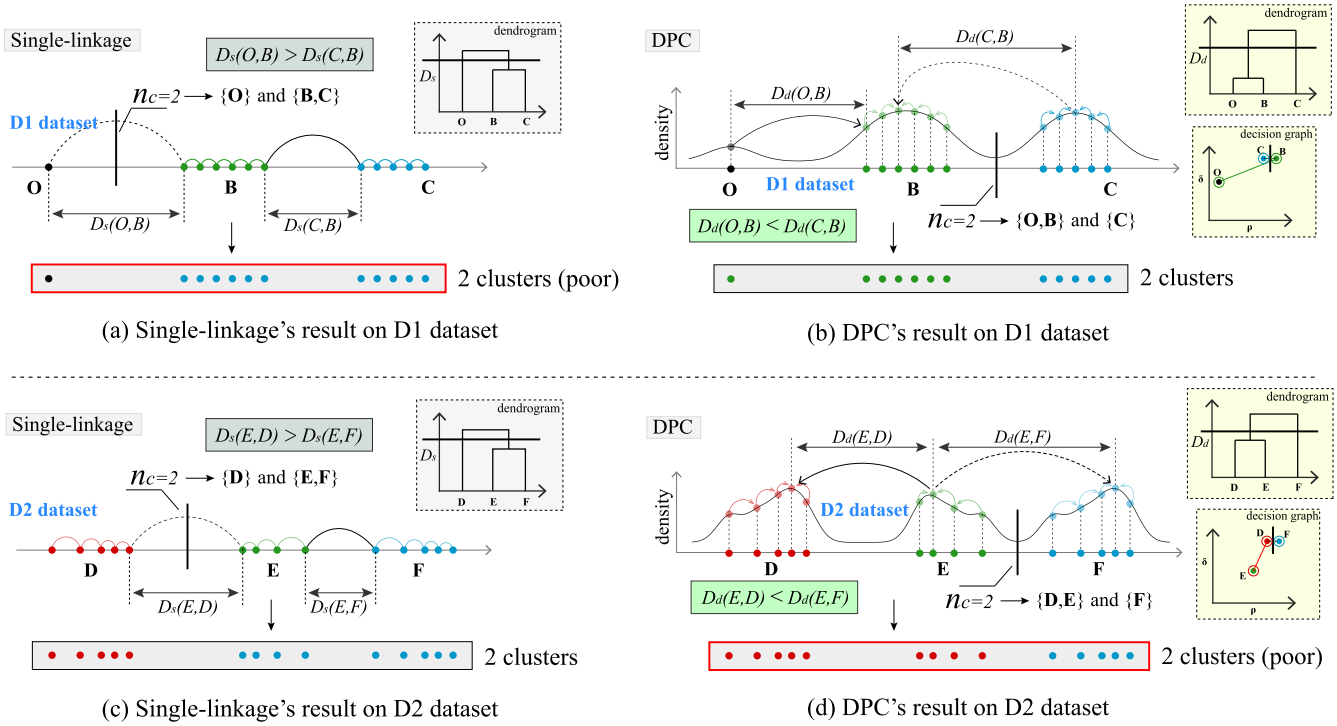


Fig. 1. The limitations of Singe-linkage and DPC. Relatively poor clustering results are highlighted in the red box.

big gap, resulting in poor clustering. For example, D, E, and F are the three normal clusters of *D2* dataset, where the gap between clusters E and F is narrower. Single-linkage successfully captured gap differences among clusters, i.e., $D_s(E,D) > D_s(E,F)$; while DPC failed, i.e., $D_d(E,D) < D_d(E,F)$. If cluster D, E, and F need to be further divided into two categories, the output of Single-linkage is cluster $\{D\}$ and cluster $\{E,F\}$, while DPC's is $\{D,E\}$ and $\{F\}$. Because D_d focuses on the minimum center-boosting distance from a low-density cluster center (e.g., the center of cluster E) to high-density areas (e.g., cluster D and F) and ignores the true gap between clusters.

Herein, a novel linkage metric that follows the density-boosting rule is designed to reasonably merge clusters into high-density areas, which takes the gap (dis-connectivity) information between clusters into account to effectively estimate the cluster dissimilarity.

C. Density-Based Clustering

Density-based clustering can well reconstruct arbitrary shapes. DBSCAN [16], a typical density-based clustering method, considers a cluster as a set of density-connected points. DBSCAN can not work well on clusters of varied densities [39]. Moreover, it needs to well tune parameters (ϵ and $minPts$) to obtain a reasonable density-connectivity criterion, which is usually a tedious process.

OPTICS [38], an improved version of DBSCAN, extends DBSCAN in detecting clusters of varied densities by applying a reachability plot. HDBSCAN [39], a hierarchical version of DBSCAN, constructs a cluster hierarchy of connected components and extracts the stable clusters from the hierarchy. HDBSCAN can detect clusters of varied densities and is more robust to parameter selection. Qian et al. [40] introduced the concept of "local density information" to help detect clusters of varied densities. In DEMOS, inspired by density-based clustering, we ensure points within a cluster are density-connected to achieve the robust reconstruction of complex-shaped clusters.

D. Density Peaks and Peak Graph

A cluster center in most density-based clustering methods is considered as a density maximum point within its local density area or a density peak [22], [23]. Thus, a non-density peak (herein, a normal point) will never be a cluster center and should be associated with at least one density peak. So, after normal points are pre-associated with density peaks, the original clustering of data points is reduced to the clustering of density peaks.

A specific sparse graph structure of density peaks (or a peak graph) has been proposed in our previous work [33], which helps to achieve a fast clustering of density peaks and effectively reconstruct complex shapes. Based on the peak graph, a multi-valley-link-based connectivity estimation method is designed to provide a high-fidelity connectivity estimation of density peaks.

III. The Proposed Method

This section gives a detailed introduction to the proposed DEMOS algorithm. Fig. 2 presents its clustering process: 1) peak graph building (blue); and 2) the clustering of density mounts (green).

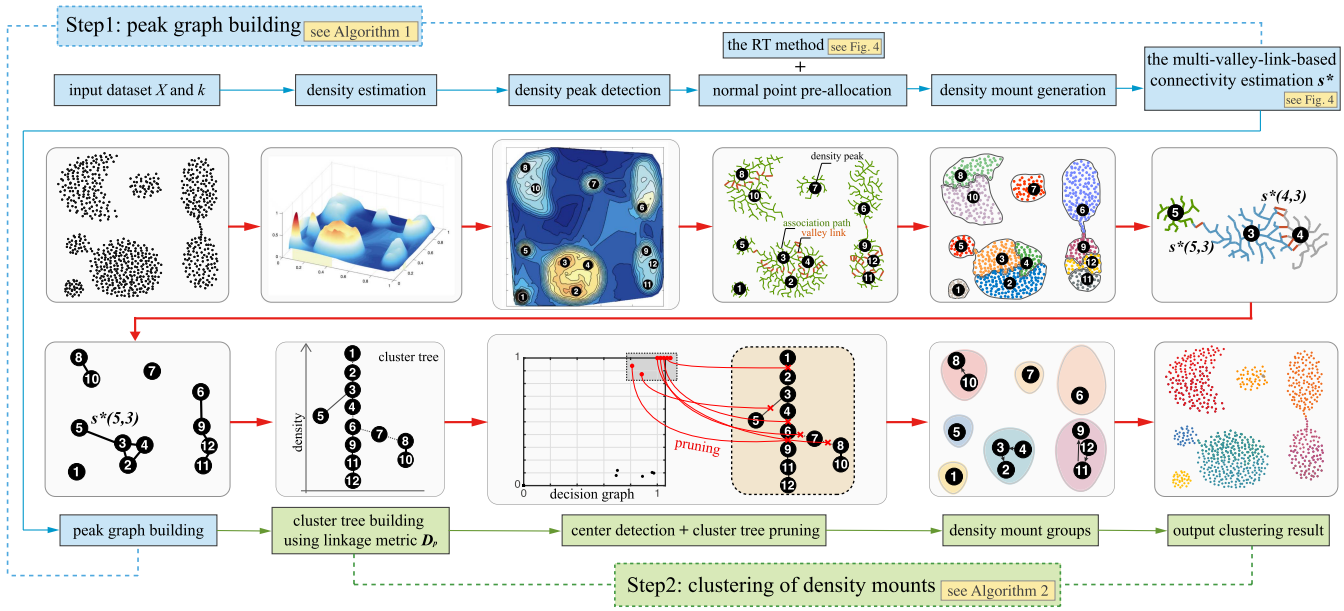


Fig. 2. The clustering process of the proposed DEMOS algorithm.

Step 1: with dataset X and parameter k as inputs, DEMOS performs density estimation of data points (by (10)) to identify density peaks (according to Definition 1); followed, DEMOS pre-assigns normal points (i.e., non-density peak points) to obtain density mounts (see Section III-B2), during which DEMOS completes the representative learning of data by using our RT method (see Section III-C2); subsequently, the multi-valley-link-based connectivity estimation method is used to evaluate the connectivity (i.e., similarity) between density peaks (see Section III-C); finally, according to the obtained similarity information, DEMOS completes the peak graph building;

Step 2: based on the peak graph, DEMOS builds a cluster tree of density mounts by using the linkage metric D_p (see Section III-A); then, DEMOS selects cluster centers using a robust cluster center selection method and prunes the cluster trees into density mount groups (see Section III-D); finally, after points in the same density mount groups being grouped together as clusters, clustering is done.

A. The Linkage Metric of Density Peaks

Consider $P_X = \{p_1, p_2, \ldots, p_{n_p}\}$ as a set of density peaks of dataset X, $P_X \subset X$, then $\bar{P}_X = X \setminus P_X$ is a set of normal points. DEMOS executes linkage-based clustering of density peaks to output cluster set $\mathbf{C}^{(t)} = \{\mathbf{C}_1, \mathbf{C}_2, \ldots, \mathbf{C}_{n_p-t}\}, t \in [0, n_p - 1]$, where $\mathbf{C}_y \in \mathbf{C}^{(t)}$ represents a cluster of density peaks, $\bigcap_{y=1}^{n_p-t} \mathbf{C}_y = \varnothing$, and $\bigcup_{i=1}^{n_p-t} \mathbf{C}_y = P_X$.

As the core of DEMOS, the linkage metric of clusters (of density peaks) is defined in (8), where $ct(\mathbf{C}_y) = \arg\max_{p_i \in \mathbf{C}_y}(\rho_{p_i})$ returns the center of cluster $\mathbf{C}_y$, and $\hat{d}_{p_i p_j}$ represents the dissimilarity between cluster centers p_i and p_j (of $\mathbf{C}_y$ and $\mathbf{C}_z$). Our linkage matric D_p ensures that each cluster is merged into a high-density area, while fully considering the connectivity between clusters. Fig. 3 presents D_p with dissimilarity estimation function $\hat{d}$ (see Section III-B4).

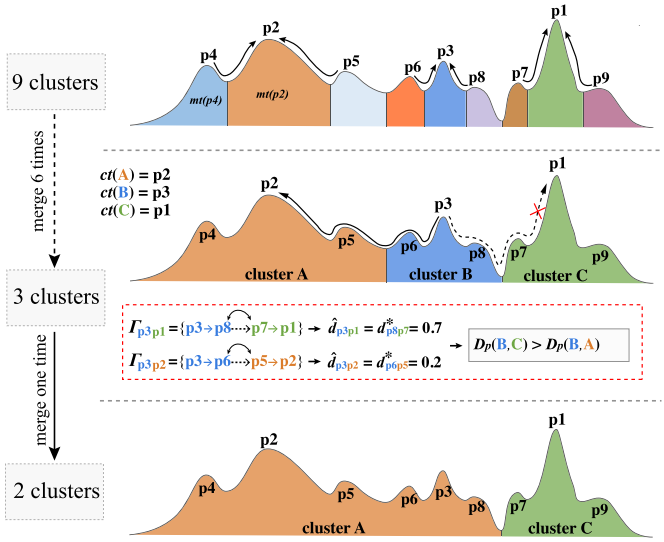


Fig. 3. The core idea of our linkage metric D_p with the dissimilarity estimation function $\hat{d}$. D_p focuses on the member dis-connectivity along the shortest path between cluster centers.

$$D_p(\mathbf{C}_y, \mathbf{C}_z) = \min_{p_i, p_j} \left(\rho_{p_i} \cdot \hat{d}_{p_i p_j}\right)$$
$$\text{s.t. } p_i = ct(\mathbf{C}_y), p_j = ct(\mathbf{C}_z), \rho_{p_j} > \rho_{p_i} \quad (8)$$

Let $G_p(P_X, E_p, w_p)$ be a complete digraph of density peaks, where $E_p = \{e_{p_i p_j} | \rho_{p_i} < \rho_{p_j}\}$, and weight function $w_p : E_p \to \mathbb{R}_+, w_p(e_{p_i p_j}) = \rho_{p_i} \cdot \hat{d}_{p_i p_j}$. Where D_p is used to build

a MST of digraph $G_p(P_X, E_p, w_p)$, as in (9).

$$T_{G_p} = \underset{T \in \mathscr{T}(G_p)}{\arg\min} \sum_{e \in E(T)} w_p(e) \tag{9}$$

In what follows, a robust connectivity-based dissimilarity estimation function $\hat{d}$ is designed based on the peak graph concept [33].

B. Peak Graph Buliding

Before building a peak graph, we need to identify density peaks and pre-assign normal points.

1) The Identification of Density Peaks: For each point $x_i \in X$, we search for its k nearest neighbors (denoted as $N_k(x_i)$) as its surrounding points, and fast estimate its local density ρ_i according to the within-surrounding similarity [41], as in (10), where m is a parameter for density smoothness control (discussed in Section III-E). Note that Laplacian centrality [54] is also an excellent method for local density estimation, but for achieving a faster execution speed, a KNN-based density estimation method is applied in DEMOS.

$$\rho_i = \frac{1}{\sum_{x_j \in N_k(x_i)} (d_{ij})^m} \tag{10}$$

Definition 1. Point x_i is a density peak, denoted as $p \in P_X$, if $\rho_i > \max_{x_j \in N_k(x_i)}(\rho_j)$.

Density peaks with the characteristic of local density maxima are defined in Definition 1. On this basis, a fast identification of density peaks within a kNN-graph of dataset X can be achieved. Parameter k, as the only hyperparameter in DEMOS, is set as $k = \lceil \sqrt{n} \rceil$ in default, where symbol $\lceil \cdot \rceil$ is a ceiling function. Note that DEMOS is insensitive to the default k setting (see discussion in Section IV-F).

2) The Pre-Association Strategy of Normal Points: According to the clustering idea of Mean-shift [23], each normal point x_i should be directly assigned to the nearest dense area within its surrounding area $N_k(i)$, and the nearest dense area is exactly where x_i's nearest higher density neighbor locates [27]. Therefore, normal points are pre-allocated to the same clusters of their nearest higher density points directly. As a result, normal points spontaneously form sub-clusters with unique density peaks as sub-cluster centers. Sub-cluster with only one density peak p as its center is called a *density mount*, denoted as $mt(p)$. Therefore, a cluster $\mathbf{C}$ of density peaks can represent a cluster C of points as in (11).

$$C = \bigcup_{p \in \mathbf{C}} mt(p) \tag{11}$$

Let adjacency matrix $A \in \mathbb{R}^{n \times n}$ express the pairwise association relationship between points, where the (i, j)-th element $a_{ij} = 1$ means that point x_j is associated to its nearest higher density neighbor x_j. Then, let $G_A(X, E_a)$ be the adjacency digraph of adjacency matrix A, where $E_a = \{e_{ij} | a_{ij} = 1, x_i, x_j \in X\}$, thus density mounts are connected components in digraph G_A. In a density mount, the density peak x_i is a sink point without outdegree in G_A, i.e., $deg^+(x_i) = \sum_{x_j \in X} a_{ij} = 0$; oppositely, source point x_i without indegree, i.e., $deg^-(x_i) = \sum_{x_j \in X} a_{ji} = 0$, is considered as an edge point.

Therefore, the original clustering of data points is simplified to the clustering of density peaks, with each normal point being associated with a density peak. To perform the clustering of density peaks, a peak graph is needed.

3) Graph Buliding: A peak graph is a sparse graph structure of density peaks, where edges only exist between density peaks within intersecting density mounts (sub-clusters). For the detection of intersecting density mounts, we introduce the concepts of valley points and links (also known as border points and links).

Valley points are considered to only exist on both sides of the borderline between intersecting density mounts. In other words, a borderline is a line that divides the mutual-proximal valley points of different density mounts. So, a small-value k_v is introduced, as:

$$k_v = \min(k, \lfloor 2 \times \ln(n) \rfloor) \tag{12}$$

to effectively detect the mutual-proximity between valley points of intersecting density mounts, where $\lfloor \cdot \rfloor$ is a floor function. Clearly, $k_v \ll n$, since $k_v \leqslant k \ll n$. The small-value k_v can effectively help to detect the proximal valley points between intersecting density mounts.

On this basis, we define valley points between intersecting density mounts as in Definition 2.

Definition 2. If mutual-proximity points x_i and x_j are in different density mounts, i.e., $x_i \in mt(p_a) \cap N_{k_v}(x_j), x_j \in mt(p_b) \cap N_{k_v}(x_i)$, then points x_i, x_j are valley points, and density mounts $mt(p_a)$, $mt(p_b)$ are intersected. Besides, points x_i, x_j are herein called a cross-mount valley point pair, denoted as $x_i \rightleftharpoons x_j$.

Since each density peak represents its density mount, the term "intersecting density peaks" is used to indicate their intersecting density mounts in the peak graph.

For a valley point x_i, we define its valley link v_i as in (13), where point τ_i is the unlinked nearest cross-mount valley point of x_i, as in (14). To ensure the independence of each valley link, once a valley pair is linked, two valley points at both ends are labeled as"linked", in other words, the connection of valley links is a without put-back operation.

$$v_i = \{x_i, \tau_i\} \tag{13}$$

$$\tau_i = \underset{x_j : x_j \rightleftharpoons x_i}{\arg\min} (d_{ij}), \text{ s.t. } x_i, x_j \text{ are both unlinked} \tag{14}$$

Definition 2 implies that merely a valley link can indicate intersecting density mounts, thus, the peak graph can be described as $G_p^S(P_X, E_p^S, w_p^S)$, where $E_p^S = \{e_{p_i p_j} | \exists v = \{x_y, \tau_y\}, x_y \in mt(p_i), \tau_y \in mt(p_j)\}$. Weight function $w_p^S : E_p^S \to \mathbb{R}_+, w_p^S(e_{p_i p_j}) = d^*_{p_i p_j}$ outputs the weight cost between density peaks p_i and p_j connected by edge $e_{p_i p_j}$, where $d^*_{p_i p_j} \in [0, 1]$ means the dis-connectivity between p_i and p_j, as defined in (15).

$$d^*_{p_i p_j} = 1 - s^*_{p_i p_j} \tag{15}$$

4) Dissimilarity Estimation Function $\hat{d}$: Let $\Gamma_{p_i p_j}$ be the shortest path (obtained by Dijkstra [42]) between density peaks

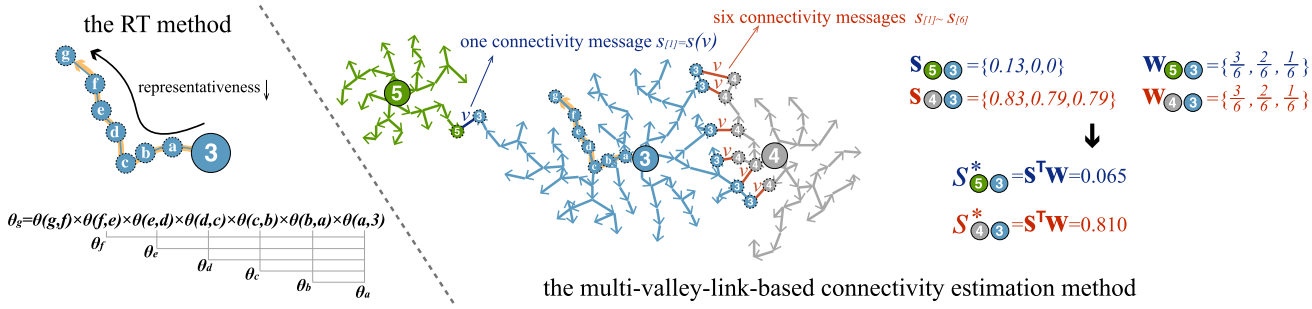


Fig. 4. The core ideas of RT method (left) and the multi-valley-link-based connectivity estimation method (right), where the minimum standard element number $n_{\mathbf{s}}$ of two connectivity message vectors are both 3.

p_i and p_j in peak graph $G_p^S(P_X, E_p^S, w_p^S)$, as in (16), where p_i and p_j are head and tail, respectively, and p'_2, p'_3 refer to the 2nd and the 3 rd density peaks along the path.

$$\Gamma_{p_ip_j} = \left\{p_i, p'_2, p'_3, \ldots, p'_{n_\Gamma-1}, p_j\right\}, n_\Gamma = |\Gamma_{p_ip_j}| \tag{16}$$

The dissimilarity estimation function $\hat{d}_{p_ip_j}$ outputs the member dis-connectivity between cluster $\mathbf{C}_y$ and $\mathbf{C}_z$ along path $\Gamma_{p_ip_j}$, s.t. $p_i = ct(\mathbf{C}_y), p_j = ct(\mathbf{C}_z)$, as in (17), where condition $\Gamma_{p_ip_j} \subseteq (\mathbf{C}_y \cup \mathbf{C}_z)$ ensures that cluster $\mathbf{C}_y$ and $\mathbf{C}_z$ are adjacent and associative. For $\Gamma_{p_ip_j} = \varnothing$, we set $\hat{d}_{p_ip_j} = 1$.

$$\begin{aligned}&\hat{d}_{p_i,p_j} = d^*_{\Gamma_{p_ip_j}(q)\Gamma_{p_ip_j}(q+1)}\\ &\text{s.t. } p_i = ct(\mathbf{C}_y), p_j = ct(\mathbf{C}_z), \Gamma_{p_ip_j} \subseteq (\mathbf{C}_y \cup \mathbf{C}_z),\\ &\Gamma_{p_ip_j}(q) \in \mathbf{C}_y, \text{and } \Gamma_{p_ip_j}(q+1) \in \mathbf{C}_z\end{aligned} \tag{17}$$

Fig. 3 presents the core idea of our linkage metric D_p with dissimilarity estimation function $\hat{d}$. At first, there are nine density peaks (mounts), and then after six times of merging according to our linkage metric D_p, three clusters A, B, and C are generated. As shown, cluster B's center $p3$ is closer to cluster C's center $p1$, but is more connected to cluster A (of $p2$). Our linkage metric D_p detected connectivity differences and successfully merged the cluster A and B.

Since, $d^* = 1 - s^*$ (15), we will detail introduce the measure of connectivity s^*.

C. Multi-Valley-Link-Based Connectivity Estimation

This subsection introduces the multi-valley-link-based connectivity estimation method, in which, the connectivity message based on representativeness is fast calculated via our RT method. The core idea of our connectivity estimation method is presented in Fig. 4.

1) Connectivity Message: Suppose that density peak p_i and p_j are mutual neighbors with the highest connectivity $s^*_{p_ip_j} = 1$ (i.e., absolute similarity) as:

Assumption 1. Mutual-neighboring density peaks are considered to own the highest connectivity, i.e. $s^*_{p_ip_j} = 1$, s.t. $p_i \in N_k(p_j), p_j \in N_k(p_i)$.

Actually, Assumption 1 is meaningless, because density peaks can never be mutual neighbors as in Definition 1.

Definition 2 indicates that at least one (usually multiple) valley link exists between intersecting density peaks. To utilize Assumption 1, we introduce the concept of *representativeness*, i.e., to what extent one point can represent another, as in Definition 3.

Definition 3. If point $x_i, x_j \in mt(p)$, then x_i has an representativeness value to represent x_j, denoted as $\theta(x_i, x_j) \in [0, 1]$.

According to Definition 3, we introduce the concept of *peak-representativeness* as in Definition 4:

Definition 4. The peak-representativeness of point $x_i \in mt(p)$ indicates x_i's representativeness to its density peak p, denoted as $\theta_i \in [0, 1]$.

According to Definition 4, valley link $v = \{x_i, x_j\}(x_j = \tau_i)$, where point x_i and x_j own representativeness θ_i and θ_j to their density peaks, respectively. Thus, a valley link v can provide a *connectivity message* s_v between density peaks, as defined in (18).

$$s_v = \theta_i \times \theta_j, v = \{x_i, x_j\}, x_j = \tau_i \tag{18}$$

To quantify connectivity message s_v, a representativeness transfer method (RT method) is designed to fast calculate the peak-representativeness θ of points.

2) Fast Representativeness Learning via the RT Method: The proposed RT method follows three definitions:

Definition 5. Density peak $x_i \in mt(p)$, i.e., $x_i = p$, owns the largest representativeness $\theta_i = 1$ to itself.

Definition 6. Direct representativeness is mutual and independent, which only exists between directly associated points x_i and x_j in G_A, i.e., *mutuality:* $\theta(x_i, x_j) = \theta(x_j, x_i)$, s.t. $a_{ij} \vee a_{ji} = 1$; and *independence:* $\theta(x_i, x_{i''}) = \theta(x_i, x_{i'}) \times \theta(x_{i'}, x_{i''})$, s.t. $a_{ii'} \wedge a_{i'i''} = 1$.

Definition 7. The representativeness between directly associated points x_i and x_j is equal to their reduced density ratio, as in (19).

$$\theta(x_i, x_j) = \frac{\min(\rho_i, \rho_j)}{\max(\rho_i, \rho_j)}, \text{s.t. } a_{ij} \vee a_{ji} = 1 \tag{19}$$

According to Definitions 5 through 7, peak-representativeness θ_i is defined in (20), where $\Delta_{x_ip} = \{x_i, x'_2, \ldots, x'_{n_\Delta-1}, p\}$ represents all n_Δ ($n_\Delta = |\Delta_{x_ip}|$) corresponding adjacent points on the path from point x_i to its density peak p (see the left

of Fig. 4).

$$\theta_i = \prod_{x_j \in \Delta_{x_i p}} \theta(x_j, x_{j'}),\ \text{s.t. } a_{jj'} = 1 \tag{20}$$

Equation (20) indicates that the peak-representativeness can be transferred to all normal points within a density mount, named the "representativeness transfer method" (the RT method). Note that the RT method can be embedded in the pre-association of normal points without adding computational complexity, so the calculation of s_v is not time-consuming.

In what follows, a multi-valley-link-based connectivity estimation method is proposed for a reasonable connectivity estimation between density peaks based on sufficient connectivity messages.

3) Connectivity Estimation: Inspired by the density-connectivity of DBSCAN [16], we get Assumption 2. Based on it, a multi-valley-link-based connectivity estimation method is proposed.

Assumption 2. Density peaks with high similarities should be well-connected and usually have multiple valley links with large connectivity messages.

For density peaks p_i and p_j, we pick a set of valley links with the top connectivity messages to structure a *connectivity message vector* $\mathbf{s}_{p_i p_j}$, as in (21), where $s_{[1]} \geqslant s_{[2]} \geqslant \ldots \geqslant s_{[n_\mathbf{s}]}$, and $n_\mathbf{s}$ represents the minimum standard number of valley link samples for structuring vector $\mathbf{s}$ (discussed in Section III-C4). If the total number n_s^* of s values (i.e., valley links) between density peaks is less than $n_\mathbf{s}$, i.e., $n_s^* < n_s$, then, we set $s_{[i]} = 0$, s.t. $n_s^* < i \leqslant n_s$.

Based on connectivity message vectors, the multi-valley-link-based connectivity $s^*_{p_i p_j}$ is defined as in (22), where $\mathbf{w}_{p_i p_j}$ is an equally decreasing weight vector for $\mathbf{s}_{p_i p_j}$ (see (23)), i.e., $s_{[i]}$ owns its corresponding weight value $w_{[i]}$. The core idea of our multi-valley-link-based connectivity estimation method is in the right of Fig. 4.

$$\mathbf{s}_{p_i p_j} = \left\{ s_{[1]}, s_{[2]}, \ldots, s_{[n_\mathbf{s}]} \right\} \tag{21}$$

$$s^*_{p_i p_j} = \mathbf{s}^\top_{p_i p_j} \mathbf{w}_{p_i p_j} \tag{22}$$

$$\begin{aligned} \mathbf{w}_{p_i p_j} &= \left\{ w_{[i]} = \frac{2}{n_\mathbf{s}^2 + n_\mathbf{s}} (n_\mathbf{s} + 1 - i) \right\}_{i=1}^{n_\mathbf{s}} \\ &= \frac{2}{n_\mathbf{s}^2 + n_\mathbf{s}} \{ n_\mathbf{s}, n_\mathbf{s} - 1, \ldots, 1 \} \end{aligned} \tag{23}$$

Our connectivity metric s^* roughly follows Assumption 2, that is, a density peak that owns multiple valley links with large connectivity messages will obtain a high similarity (connectivity) s^*, as verified after:

Proof: Since $\mathbf{1}^\top \mathbf{w}_{p_i p_j} = 1$ and $s \in [0, 1]$, therefore, $s^*_{p_i p_j} \in [s_{[n_\mathbf{s}]}, s_{[1]}] \subseteq [0, 1]$. Clearly, if and only if $s_{[1]} = s_{[n_\mathbf{s}]} = 1$, $s^*_{p_i p_j} \equiv 1$ (i.e., the largest connectivity). Where $s_{[n_\mathbf{s}]} = s_{[1]}$ represents uniformity; and $s_{[n_\mathbf{s}]} = 1$ indicates that all messages own the largest connectivity value, i.e., $\forall s \in \mathbf{s}_{p_i p_j}, s = 1$. Therefore, s^* roughly follows Assumption 2.

4) The Self-Acquisition of $n_\mathbf{s}$: The minimum standard element number $n_\mathbf{s}$ of connectivity message vector $\mathbf{s}_{p_i p_j}$ is supposed to be positively related to the edge point number of a density mount, i.e., a density mount with more edge points should use more sufficient connectivity messages to describe its connectivity with other density mounts.

$$n_\mathbf{s} = \lceil \eta \times \min\left(n_\epsilon(p_i), n_\epsilon(p_j)\right) \rceil \tag{24}$$

$$n_\epsilon(p) = \sum_{x_i \in mt(p)} \epsilon(x_i),\ \epsilon(x) = \begin{cases} 1 & deg^-(x) = 0 \\ 0 & \text{others} \end{cases} \tag{25}$$

On this basis, a self-acquisition method is proposed to determine $n_\mathbf{s}$, as in (24), where η is a ratio parameter (default is $\eta = 0.25$). The function $n_\epsilon(p)$ outputs the total number of edge points in density mount $mt(p)$, as in (25), where $\epsilon(\cdot)$ is an edge point judgment function.

D. A Robust Cluster Center Selection Method

According to the obtained connectivity s^* values between density peaks, the dissimilarity matrix of density peaks can be obtained via function $\hat{d}$ according to (17), and then, the density-boosting cluster tree T_{G_p} based on linkage metric D_p is built according to (9).

$$\dot{p} = \arg\min_{p'} (\hat{d}_{pp'}), \text{s.t. } \rho_{p'} > \rho_p \tag{26}$$

In cluster tree T_{G_p}, each density peak p views the most similar density peak of higher density as its unique parent node $\dot{p}$, as in (26). To prune cluster tree T_{G_p} to achieve final clustering, we propose a robust cluster center selection method based on our cluster assumption.

1) Our Cluster Assumption: According to (17), $\hat{d}_{p\dot{p}}$ essentially outputs a member dis-connectivity (gap) between cluster $\mathbf{C}_y$ and $\mathbf{C}_z$ with $p = ct(\mathbf{C}_y), \dot{p} = ct(\mathbf{C}_z)$. Thus, if density peak p is a real cluster center, its corresponding cluster $\mathbf{C}_y$ should have a relatively large member dis-connectivity $\hat{d}_{p\dot{p}}$ with cluster $\mathbf{C}_z$. Inspired by this, we propose our cluster assumption:

Assumption 3. A cluster $\mathbf{C} \subseteq P_X$ should have a high-density cluster center $p = ct(\mathbf{C})$, and have a relatively large dis-connectivity (gap) δ with other higher-density clusters of density peaks.

Based on Assumption 3, δ of density peaks is calculated as in (27). For each normal point $x_i \in \bar{P}_X, \delta_i = 0$, as in (28), because normal points own no center attribute.

$$\delta_p = \min_{p'} \hat{d}_{pp'}, \text{s.t. } \rho_{p'} > \rho_p \tag{27}$$

$$\delta_i = 0, x_i \in \bar{P}_X \tag{28}$$

Once each density peak p owns its candidate center attributes: ρ_p and δ_p, true cluster centers are clearly displayed in our decision graph. Fig. 5 (left) demonstrates the superiority of our decision graph on the *Agg* dataset [44].

After selecting cluster centers, cluster tree T_{G_p} is pruned at the selected cluster centers (as new root nodes) to generate final clusters of density peaks. Then, normal points are allowed to inherit cluster labels from their density peaks. Once each point owned its cluster label, clustering is done.

2) Decision Graph Clarity Metric: A clear decision graph increases the center detection robustness. Herein, F1-score [46]

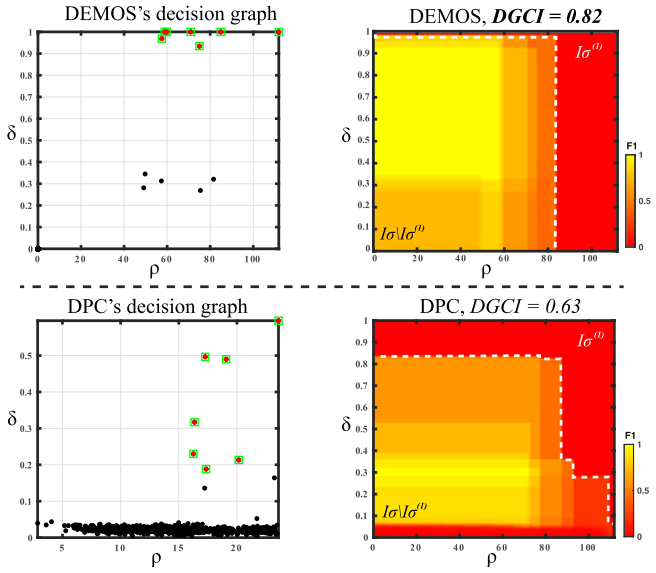


Fig. 5. Our decision graph performance comparison with DPC on dataset *Agg* (7 clusters). Cluster centers are marked by red color.

is used to quantify the center detection robustness, as in (29).

$$F1 = \frac{2 \times TP}{2 \times TP + FP + FN} \tag{29}$$

Let $\sigma = \{\sigma_1, \sigma_2\} \in I_\sigma$ be a threshold point for center detection, i.e., points x_i with $\frac{\rho_i - \rho_{min}}{\rho_{max} - \rho_{min}} \geqslant \sigma_1$ and $\frac{\delta_i - \delta_{min}}{\delta_{max} - \delta_{min}} \geqslant \sigma_2$ are selected as centers, where $\rho_{min} = \min_{x_i \in X}(\rho_i)$, $\rho_{max} = \max_{x_i \in X}(\rho_i)$, $\delta_{min} = \min_{x_i \in X}(\delta_i)$, and $\delta_{max} = \max_{x_i \in X}(\delta_i)$. I_σ indicates the domain of σ as in (30). Therefore, a threshold point $\sigma = \{\sigma_1, \sigma_2\} \in I_\sigma$ can provide an F1 score of center detection as $f_{F1}(\sigma_1, \sigma_2) : I_\sigma \to [0, 1]$.

$$I_\sigma = \left\{\sigma = \{\sigma_1, \sigma_2\} \in \mathbb{R}^2 | 0 \leqslant \sigma_1 \leqslant 1, 0 \leqslant \sigma_2 \leqslant 1\right\} \tag{30}$$

$$DGCI = \frac{\iint_{\left(I_\sigma \setminus I_\sigma^{(1)}\right)} f_{F1}(\sigma_1, \sigma_2) \mathrm{d}\sigma_1 \mathrm{d}\sigma_2}{\iint_{\left(I_\sigma \setminus I_\sigma^{(1)}\right)} \mathrm{d}\sigma_1 \mathrm{d}\sigma_2} \tag{31}$$

A F1-score-based Decision Graph Clarity Index ($DGCI$) is also designed to quantify the clarity of a decision graph, as in (31), where $I_\sigma^{(1)}$ indicates meaningless threshold domain, i.e., $\sigma \in I_\sigma^{(1)}$ can only provide a single cluster. Generally, a clear decision graph owns a high $DGCI \in [0, 1]$. As Fig. 5 shows, DEMOS's decision graph is much clearer than DPC's, for owning a much larger optimal threshold setting area (yellow area) than DPC. As a result, DEMOS gets a higher $DGCI = 0.82$ than DPC ($DGCI = 0.63$).

3) Clarity-Enhancing Methods of Decision Graph: According to (31), if all cluster centers fall in the upper right corner of the decision graph (i.e., the optimal candidate region), and all non-centers fall in the bottom left corner (i.e., the worst candidate region), the decision graph will have a high $DGCI$ score. To obtain a high-$DGCI$ (clear) decision graph, two clarity-enhancing methods are proposed.

Algorithm 1: DEMOS-Step1: Peak Graph Building.

Input: dataset $X = \{x_1, x_2, \ldots, x_n\}$, and parameter k.
Output: peak graph $G_p^S(P_X, E_p^S, w_p^S)$ and density mounts $mt(P_X) = \{mt(p)|p \in P_X\}$.

```
fast obtain the kNN matrix, by applying fast kNN technique [45].
estimate density ρ = {ρ_1, ρ_2, ..., ρ_n}, w.r.t c_v ≈ 0.4, (10).
for each point x_i ∈ X do
    θ_i = 1 // initialization
end for
order the dataset X as X' in descending order of ρ.
for each point x_i ∈ X', from high-ρ to low-ρ do
    for each neighbor x_j ∈ N_k(x_i), from near to far do
        if ρ_j > ρ_i then
            a_ij = 1 // adjacency matrix A.
            θ_i = θ_j × θ(x_i, x_j) // (19) and (20), RT method.
            break
        end if
    end for
end for
for each point x_i ∈ X do
    if deg^+(x_i) = Σ_{x_j ∈ X} a_ij = 0 then
        P_X = P_X ∪ x_i // density peaks
        x_i ← a unique label
    else
        P̄_X = P̄_X ∪ x_i// normal points
        Label(x_i) ← Label(x_j), s.t. a_ij = 1 // Label(x) outputs x's label.
    end if
end for
points with same label form density mounts mt(P_X) = {mt(p)|p ∈ P_X}.
for each pair of density peak p_i, p_j ∈ P_X do
    calculate the connectivity s*_{p_i p_j} // see Section III-C.
    if s*_{p_i p_j} ≠ ∅ then
        E_p^S = E_p^S ∪ e_{p_i p_j}// for peak graph building.
    end if
end for
return G_p^S(P_X, E_p^S, w_p^S) and density mounts mt(P_X) = {mt(p)|p ∈ P_X}.
```

1). *The difference removal of* ρ: cluster centers in low-density areas are always squeezed to the left regions of the decision graph, causing a low $DGCI$ score. To assure that density peaks in different density areas have an equal probability to be selected as cluster centers, we remove density differences among connected components of peak graph $G_p(P_X, E_p)$ (denoted as $G_p^{con}(P_X^{con}, E_p^{con})$, where $P_X^{con} \subseteq P_X, E_p^{con} \subseteq E_p^S$), as in (32).

$$\hat{\rho}_p = \frac{\rho_p}{\max_{p_i \in P_X^{con}}(\rho_{p_i})}, p \in P_X^{con} \tag{32}$$

Algorithm 2: DEMOS-Step2: Clustering of Density Mounts.

Input: dataset X, and peak graph $G_p^S(P_X, E_p^S, w_p^S)$ and density mounts $mt(P_X) = \{mt(p)|p \in P_X\}$.
Output: clustering result $C = \{C_1, C_2, \ldots, C_{n_c}\}$

```
build fully connected peak graph G_p(P_X, E_p, w_p)
  based on peak graph G_p^S(P_X, E_p^S, w_p^S).
build cluster tree T_{G_p} according to (9).
for each density peak p ∈ P_X do
    calculate δ_p according to (27).
end for
for each density peak p ∈ P_X do
    ρ̂_p = ρ_p / max_{p_i ∈ P_X^{con}(ρ_{p_i})}, p ∈ P_X^{con}
    δ̂_p = δ_p^2
end for
for each normal point x_i ∈ P̄_X do
    δ_i = 0
end for
select appropriate cluster centers c={c_1, c_2, ..., c_{n_c}}
  in decision graph of ρ̂ and δ̂.
non-center density peaks inherit cluster labels from
  their parent node in cluster tree T_{G_p}.
obtain clusters of density peaks
  C = {C_1, C_2, ..., C_{n_c}}.
for each C_i ∈ C do
    C_i = ⋃_{p∈C_i} mt(p)
end for
return Clustering result C = {C_1, C_2, ..., C_{n_c}}
```

Therefore, cluster centers of low-density areas fall in the optimal candidate region of the decision graph, obtaining a high $DGCI$ score.

2). *The difference amplification of* δ: Assumption 3 tells that $\delta \in [0, 1]$ values of center density peaks (i.e., real cluster centers) are closer to 1 than non-center density peaks. Herein, we provide a simple method to amplify the difference between δ values of centers and non-centers, as in (33), where λ is called an amplification factor (default is $\lambda = 2$).

$$\hat{\delta}_p = \delta_p^{\lambda}, p \in P_X \tag{33}$$

As a result, δ value differences between center and non-center density peaks are amplified, which presses non-center density peaks to the bottom of the decision graph, obtaining a high $DGCI$ score.

Fig. 6 shows the performance of our decision graph on *Agg* dataset using our clarity-enhancing methods. As shown, our decision graph with $DGCI = 0.93$ is clearer than the original decision graph with $DGCI = 0.82$.

E. Density Smoothness Control Parameter m

According to (19) and (20), a smoother density distribution tends to produce a high peak-representativeness, which may result in a high connectivity value between density peaks, and vice versa. However, the two cases can both lead to a small-variance

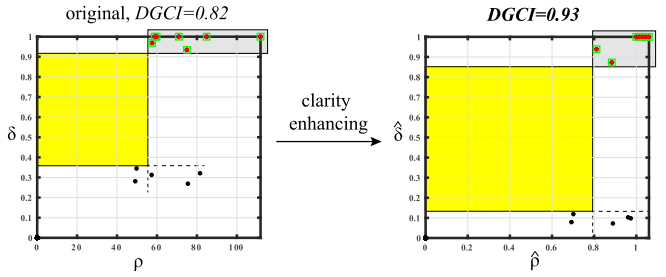


Fig. 6. The demonstration of the clarity-enhancing of decision graph (on *Agg*) via the proposed clarity-enhancing methods. The yellow area indicates the optimal threshold setting area and the gray area indicates the real cluster center area.

connectivity estimation that will yield a small-variance δ estimation (i.e., the δ-distribution tends to be uniform), according to (27). Thus, δ-difference between centers and non-centers may be narrowed, so as to increase the difficulty of cluster center detection.

$$c_v = \frac{\sqrt{\frac{1}{n}\sum_{i=1}^{n}(\rho_i - \bar{\rho})^2}}{\bar{\rho}}, \bar{\rho} = \frac{1}{n}\sum_{x_i \in X}\rho_i \tag{34}$$

To obtain an appropriate connectivity estimation, a controllable local density estimation method with a density smoothness control parameter m (see (10)) is designed. The variation coefficient c_v of density distribution is applied to describe the smoothness of density distribution, as in (34). Let parameter m self-tune to control $c_v \approx c_v^*$, where c_v^* is a smoothness threshold (default is $c_v^* = 0.4$). By using the controllable local density estimation method, DEMOS can always produce a stable and appropriate connectivity estimation for robust center detection, even when dealing with datasets of different distributions.

F. Pseudocode and Complexity

Algorithms 1 and 2 show the pseudocode of DEMOS in two steps: 1) peak graph building; 2) the clustering of density mounts.

The computational complexity of peak graph building is $O(n\log(n) + n\tilde{k} + n)$, where $\tilde{k}$ means that a point's $\tilde{k}$-th neighbor (an average concept) is its nearest higher density point. In fact, most points can find a really close higher density point, i.e., $\tilde{k} \ll k$. And peak-graph-based clustering is $O(n_p\log(n_p) + |E_p^S| + n_p + n)$, where $|E_p^S|$ is the total edges of peak graph G_p^S.

Since $n_p \ll n$, the overall computational complexity of DEMOS is $O(n\log(n) + n\tilde{k} + |E_p^S|)$, where $\tilde{k}$, and $|E_p^S|$ are all far less than n.

IV. Experiments

A. Experimental Set up

Datasets. Ten common synthetic datasets of different shapes and eight popular real-world datasets are selected to benchmark clustering algorithms. The detailed summarization is displayed in Table II.

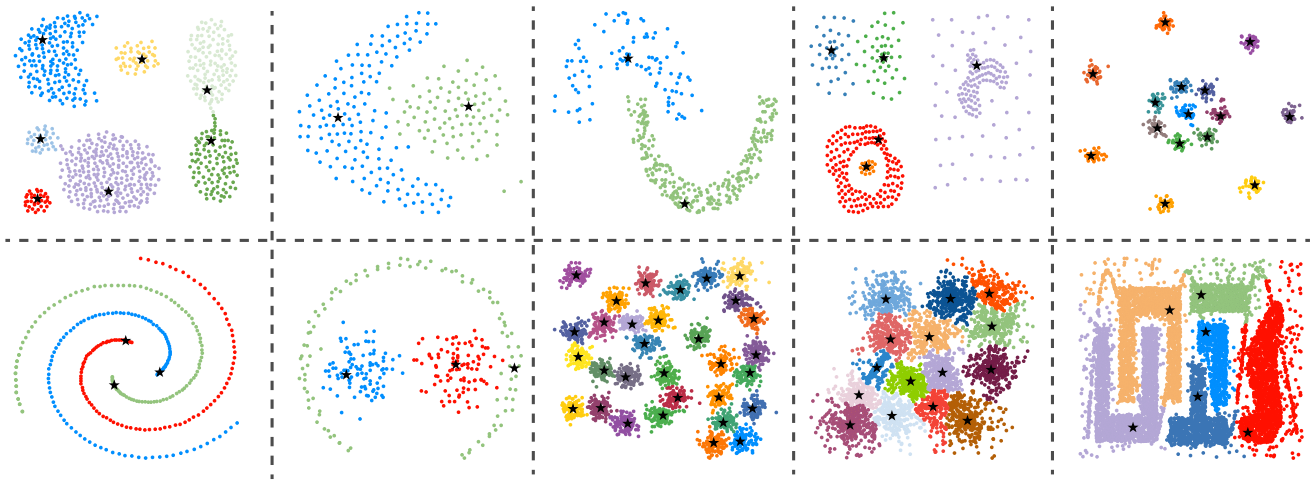


Fig. 7. The clustering results of the proposed DEMOS on 10 tested synthetic datasets (in Table II) of different shapes.

TABLE II
DATASETS

Dataset	Instances	Attributes	Clusters	Source
Agg	788	2	7	[44]
Flame	240	2	3	[44]
Jain	373	2	2	[44]
Comound	399	2	6	[44]
R15	600	2	15	[44]
Spiral	312	2	3	[44]
Pathbased	300	2	3	[44]
D31	3100	2	31	[44]
S3	5000	2	15	[44]
T48K	8000	2	6	[44]
Iris	150	4	3	[48]
Wine	178	13	3	[48]
Segment	2310	19	7	[48]
Drivedata	606	16	4	[48]
Breastcancer	569	30	2	[48]
YTF	10000	10	41	[49]
USPS	11000	10	10	[50]
MNIST	10000	500	10	[51]

Comparison Algorithms and Settings. four classic clustering methods (K-means (KM) [12], Self-tuning Spectral Clustering (SSC) [43], DBSCAN (DB) [16], HDBSCAN (HDB) [39]), seven state-of-the-art DPC-based algorithms (DPC [22], SSSP-DPC [29], SNN-DPC [30], DPC-CE [34], FastDPeak [32], PGDPC [33], and DPC-DBFN [37]), and the proposed DEMOS are the comparison algorithms. We set parameters of different comparison algorithms according to their best performance over a large range of possible configurations. Besides, for K-means and SSC, we use the best results among ten runs; while for all DPC-based algorithms, we manually select appropriate density peaks as cluster centers by observing their decision graphs.

Data Preprocessing. the min-max normalization [47] is used to preprocess datasets to avoid the difference of dimensional metrics.

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

Evaluation Metric. The popular Adjusted Rand Index (ARI) [52], and Adjusted Mutual Information (AMI) [52] are applied to evaluate the clustering performance.

B. Experiments on Synthetic Datasets

Fig. 7 presents the clustering results of DEMOS on ten synthetic datasets of different shapes, where the "★" represents the identified cluster center, and different colors indicate different clusters. As shown, the proposed DEMOS algorithm almost perfectly reconstructs the complex-shaped clusters in all tested datasets.

Table III presents the AMI and ARI scores of DEMOS and other comparison algorithms, where the best results are highlighted. As shown, DEMOS stands out for its high scores on almost all synthetic datasets.

In reality, complex-shaped data may have many density peaks, and the number of density peaks is changeable and difficult to settle. Therefore, to further verify the practicability of DEMOS, we applied the DEMOS to several complex-shaped synthetic datasets with a changeable number of density peaks. For each dataset, its density distribution is obtained by (10) with $k = \lceil \sqrt{n} \rceil$, and then, four peak graphs with different density peak numbers are built for clustering. Fig. 8 shows the clustering results of DEMOS on the *Agg* dataset, where n_p indicates the total number of density peaks, and n_c indicates the selected cluster number. As shown, DEMOS always successfully assigned density peaks to the right clusters, regardless of the changeable number of density peaks. Table IV shows the AMI and ARI scores obtained by DEMOS on six synthetic datasets, which verifies the robustness of DEMOS in handling complex-shaped data with a changeable number of density peaks.

As verified, the proposed DEMOS algorithm has an excellent performance in reconstructing complex shapes.

C. Experiments on Real-World Datasets

To further evaluate the clustering performance of DEMOS in dealing with datasets of high-dimensional and large size, we conducted experiments on five common UCI real-world datasets [48] of high-dimensional (*Iris*, *Wine*, *Segment*, *Drivedata*, and *Breastcancer*), and three common machine learning real-world datasets of large size (the *YTF* dataset that contains

TABLE III
THE COMPARISON OF AMI AND ARI ON SYNTHETIC DATASETS

Dataset	Metric	KM	SSC	DB	HDB	DPC	SSSP-DPC	SNN-DPC	PGDPC	DPC-CE	FastDPeak	DPC-DBFN	DEMOS
Agg	AMI	0.80	0.94	0.96	0.77	0.99	0.97	0.93	0.99	0.99	0.98	0.99	**1.00**
	ARI	0.71	0.94	0.98	0.78	0.99	0.97	0.94	0.99	0.99	0.99	0.99	**1.00**
Flame	AMI	0.44	0.54	0.87	0.62	**1.00**	**1.00**	0.82	**1.00**	**1.00**	0.93	0.90	**1.00**
	ARI	0.46	0.61	0.95	0.80	**1.00**	**1.00**	0.89	**1.00**	**1.00**	0.97	0.95	**1.00**
Jain	AMI	0.49	0.50	0.86	0.72	0.54	0.35	**1.00**	**1.00**	**1.00**	0.50	0.62	**1.00**
	ARI	0.57	0.57	0.97	0.91	0.62	0.32	**1.00**	**1.00**	**1.00**	0.57	0.71	**1.00**
Compound	AMI	0.60	0.70	**0.86**	0.70	0.76	0.84	0.81	0.82	0.83	0.68	0.78	0.84
	ARI	0.44	0.49	**0.90**	0.76	0.59	0.83	0.81	0.62	0.84	0.53	0.76	0.85
R15	AMI	**0.99**	**0.99**	0.94	0.85	**0.99**	**0.99**	**0.99**	**0.99**	**0.99**	**0.99**	**0.99**	**0.99**
	ARI	**0.99**	**0.99**	0.95	0.75	**0.99**	**0.99**	**0.99**	**0.99**	**0.99**	**0.99**	**0.99**	**0.99**
Spiral	AMI	-0.01	**1.00**	**1.00**	0.97	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	0.29	**1.00**
	ARI	-0.01	**1.00**	**1.00**	0.99	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	0.25	**1.00**
Pathbased	AMI	0.51	0.57	0.75	0.42	0.50	0.71	0.82	0.44	0.49	0.44	0.47	**0.96**
	ARI	0.46	0.53	0.77	0.47	0.45	0.61	0.86	0.41	0.47	0.41	0.43	**0.97**
D31	AMI	0.95	**0.97**	0.86	0.81	0.95	0.96	0.96	0.96	0.96	0.95	0.96	0.96
	ARI	0.91	**0.95**	0.71	0.56	0.93	0.94	0.94	0.94	0.94	0.93	0.93	0.94
S3	AMI	0.86	0.89	0.66	0.61	0.94	0.88	0.87	0.96	0.96	0.93	0.87	**0.97**
	ARI	0.80	0.85	0.30	0.25	0.92	0.83	0.82	0.95	0.95	0.91	0.82	**0.97**
T48K	AMI	0.58	0.66	**0.88**	0.61	0.66	0.84	0.59	0.61	0.70	0.55	0.47	0.83
	ARI	0.47	0.53	**0.86**	0.24	0.57	0.81	0.50	0.42	0.59	0.39	0.31	**0.86**

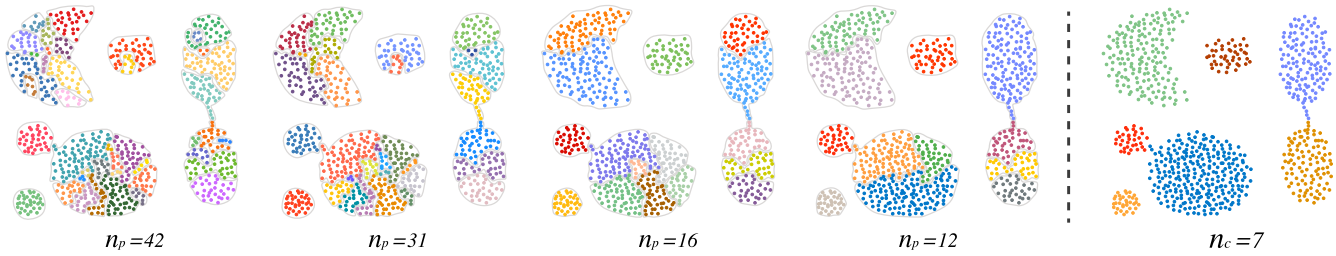


Fig. 8. The clustering result ($n_c = 7$) of DEMOS on the *Agg* dataset with $n_p = 42, 31, 16$, and 12.

TABLE IV
THE AMI AND ARI SCORES OF DEMOS ON DIFFERENT SYNTHETIC DATASETS WITH DIFFERENT NUMBERS OF DENSITY PEAKS

Agg	AMI ARI	1.00 1.00	1.00 1.00	1.00 1.00	1.00 1.00
	n_p/n_c	42/7	31/7	16/7	12/7
Flame	AMI ARI	1.00 1.00	1.00 1.00	1.00 1.00	1.00 1.00
	n_p/n_c	11/2	9/2	7/2	6/2
Jain	AMI ARI	1.00 1.00	1.00 1.00	1.00 1.00	1.00 1.00
	n_p/n_c	21/2	18/2	13/2	10/2
Spiral	AMI ARI	1.00 1.00	1.00 1.00	1.00 1.00	1.00 1.00
	n_p/n_c	30/2	15/2	5/2	4/2
D31	AMI ARI	0.96 0.94	0.96 0.94	0.96 0.94	0.96 0.94
	n_p/n_c	62/31	52/31	36/31	34/31
S3	AMI ARI	0.92 0.89	0.93 0.90	0.96 0.95	0.97 0.97
	n_p/n_c	34/15	28/15	20/15	17/15

10,000 samples of 41 persons' faces, the *USPS* dataset that contains 11,000 samples of handwritten digits, and the *MNIST* that contains 10,000 samples of handwritten digits preprocessed [19]). The detailed information of these eight datasets is in Table II.

Table V reports the experimental results of comparison algorithms, where the best results are highlighted. As shown, the overall performance of DEMOS is outstanding, especially on three large datasets: the *YTF*, *USPS*, and *MNIST* datasets. The above experiments verified that the proposed DEMOS shall be an alternative method for real-world dataset clustering.

Fig. 9(a) presents the two-dimensional data visualizations (marked by ground truth) of *MNIST* and *USPS* acquired by t-SNE [53] (a classic dimensionality reduction algorithm). As shown, t-SNE demonstrated admirable data visualization results. It almost perfectly distinguished the ten classes. For comparison, Fig. 9(b) shows the t-SNE data visualizations of *MNIST* and *USPS* that are marked by DEMOS's clustering labels. As shown, for the *MNIST* dataset, DEMOS obtained a pleasing partitioning result very close to the true labels; and for the *USPS* dataset, DEMOS also obtained a good partitioning result. The above experiment verifies the superiority of DEMOS in dealing with the *MNIST* and *USPS* datasets.

Although t-SNE's excellent dimensionality reduction ability can help visualize high-dimensional data to achieve naked-eye clustering, it is prohibitively time-consuming in dealing with large data. For example, t-SNE took 6740 seconds for *USPS* visualization, while DEMOS only took about 1 second on clustering *USPS*. Therefore, DEMOS is more suitable for large data clustering.

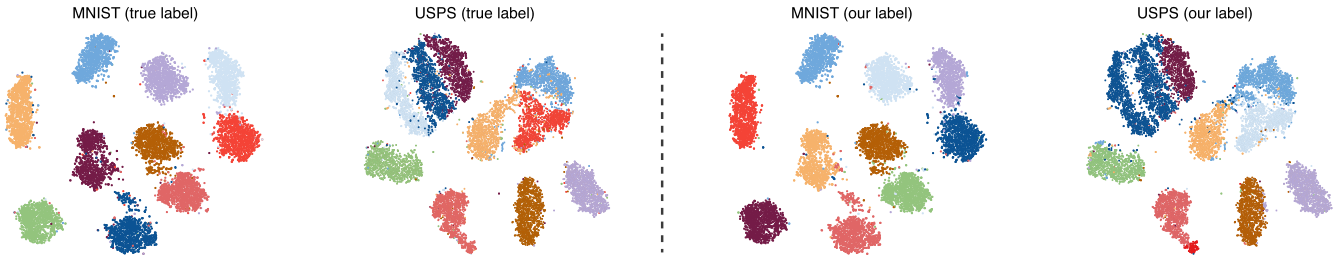
In addition, Fig. 10 presents handwriting digits of different peak-representativeness values. As shown, standard handwriting digits tend to have a large peak-representativeness value, which verifies the effectiveness of our representativeness learning.

D. The Clarity of Decision Graphs

Benefited from the new center assumption and clarity-enhancing methods, DEMOS can own a clear decision graph,

TABLE V
THE COMPARISON OF AMI AND ARI ON REAL-WORLD DATASETS

Dataset	Metric	KM	SSC	DB	HDB	DPC	SSSP-DPC	SNN-DPC	PGDPC	DPC-CE	FastDPeak	DPC-DBFN	DEMOS
Iris	AMI	0.73	0.84	0.58	0.52	0.86	0.88	**0.91**	0.88	0.73	0.77	0.81	0.88
	ARI	0.72	0.87	0.57	0.48	0.88	0.90	**0.92**	0.90	0.66	0.72	0.81	0.90
Wine	AMI	0.87	**0.89**	0.53	0.41	0.70	0.75	0.87	0.74	0.58	0.74	0.80	0.74
	ARI	0.90	**0.91**	0.45	0.34	0.67	0.74	0.90	0.73	0.54	0.73	0.83	0.73
Segment	AMI	0.61	0.65	0.41	0.42	0.63	0.62	0.67	0.72	0.60	0.64	0.00	**0.74**
	ARI	0.50	0.57	0.22	0.23	0.50	0.41	0.58	0.64	0.50	0.54	0.00	**0.66**
Drivedata	AMI	0.75	0.75	0.71	0.59	0.70	**0.78**	0.70	**0.78**	0.70	**0.78**	0.76	**0.78**
	ARI	0.74	0.74	0.72	0.54	0.72	0.77	0.74	**0.79**	0.67	0.77	0.77	**0.79**
Breastcancer	AMI	0.61	0.67	0.26	0.19	0.41	0.34	**0.75**	0.63	0.37	0.45	0.18	0.64
	ARI	0.73	0.79	0.29	0.17	0.47	0.38	**0.85**	0.74	0.43	0.52	0.19	0.75
YTF	AMI	0.74	0.75	0.67	0.61	0.80	0.80	0.76	0.80	0.74	0.75	0.71	**0.81**
	ARI	0.52	0.49	0.38	0.18	0.59	0.58	0.52	0.60	0.52	0.54	0.54	**0.64**
USPS	AMI	0.61	0.72	0.38	0.44	0.52	0.74	0.61	0.71	0.46	0.58	0.26	**0.79**
	ARI	0.51	0.60	0.20	0.08	0.30	0.60	0.45	0.58	0.34	0.41	0.16	**0.72**
MNIST	AMI	0.82	0.89	0.56	0.54	0.71	0.61	0.77	0.82	0.77	0.79	0.32	**0.94**
	ARI	0.77	0.82	0.24	0.16	0.61	0.31	0.66	0.73	0.68	0.69	0.12	**0.94**



(a) t-SNE visualization results on MNIST and USPS with ture labels

(b) t-SNE visualization results on MNIST and USPS with our clusteing labels

Fig. 9. The t-SNE-based visualization comparison between the true labels and our clustering labels on *MNIST* and *USPS*.

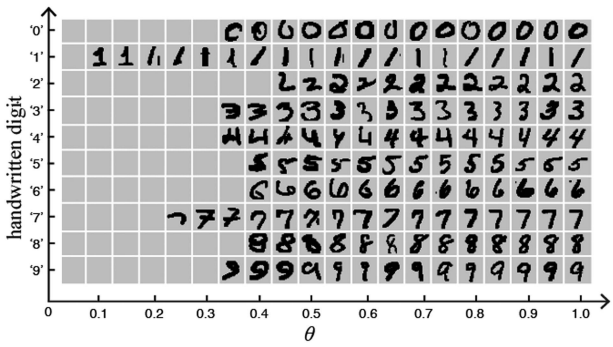


Fig. 10. The MNIST handwriting digits with different learned peak-representativeness θ by our RT method.

which implies that DEMOS is robust in cluster center detection. To verify the superiority of DEMOS to other DPC-based algorithms, we apply F1-score and $DGCI$ index to conduct quantitative comparison experiments on different decision graphs. The corresponding results are presented in Table VI, where DEMOS(-) is a DEMOS version without clarity-enhancing.

As Table VI shows, our DEMOS(-) and DEMOS obtained the highest F1 scores on all datasets except the *YTF* dataset, which verifies the effectiveness of our decision graph in cluster center detection. Besides, DEMOS obtained the highest $DGCI$ scores on most datasets, which verifies that the decision graphs of DEMOS are the clearest, and the second is DEMOS(-), indicating the proposed clarity-enhancing methods are effective.

Fig. 11 presents the decision graph comparisons of DEMOS and four DPC-based algorithms on the *MNIST* dataset. As shown, the decision graphs of DEMOS and PGDPC are more concise than others, because the two only consider searching for cluster centers among density peaks; while the decision graphs of DPC, SNN-DPC, and SSSP-DPC are relatively ambiguous, because these methods search for cluster centers among data points. Note that, benefiting from the clear decision graph, only DEMOS achieved an accurate detection of the ten real cluster centers.

As verified, DEMOS is more robust in cluster center detection than the state-of-the-art DPC-based algorithms.

E. The Speed of DEMOS

In dealing with large-scale data clustering, the running speed is one of the most important factors that need special attention. As analyzed in Section III-F, DEMOS with computational complexity of $O(n\log(n) + n\tilde{k} + |E_p^S|)$ is faster than DPC ($O(n^2)$).

In Fig. 12, while the histogram (left panel) shows the runtime of different DPC-based algorithms on all the eighteen tested datasets in Table II, the line chart (right panel) shows the runtime of DPC, SNN-DPC, and DEMOS on all the tested datasets.

As shown in the histogram, DEMOS is much faster than DPC, DPC-CE, SSSP-DPC, and SNN-DPC, but is slightly slower than FastDPeak and PGDPC with a relatively lower computational complexity of $O(n\log(n))$. Although DEMOS and PGDPC are both peak-graph-based, the former needs more calculations in

TABLE VI
THE COMPARISON OF F1 AND DGCI SCORES OF DIFFERENT DPC-BASED ALGORITHMS ON TESTED DATASETS

Algorithm	DPC	SSSP-DPC	SNN-DPC	PGDPC	DPC-CE	FastDPeak	DPC-DBFN	DEMOS(-)	DEMOS
Metric	F1/DGCI	F1/DGCI	F1/DGCI	F1/DGCI	F1/DGCI	F1/DGCI	F1/DGCI	F1/DGCI	F1/DGCI
Agg	**1.00**/0.63	**1.00**/0.66	**1.00**/0.60	**1.00**/0.76	**1.00**/0.59	**1.00**/0.65	**1.00**/0.48	**1.00**/0.82	**1.00**/**0.93**
Flame	**1.00**/0.81	**1.00**/0.87	**1.00**/0.66	**1.00**/0.80	**1.00**/0.67	**1.00**/0.92	**1.00**/0.82	**1.00**/0.86	**1.00**/**0.93**
Jain	0.50/0.43	0.50/0.44	0.50/0.42	**1.00**/0.73	**1.00**/0.45	0.50/0.38	0.50/0.39	**1.00**/0.51	**1.00**/**0.85**
Compound	0.73/0.55	0.83/0.54	0.73/0.48	0.83/0.62	**0.91**/0.55	0.83/0.54	0.83/0.48	**0.91**/0.70	**0.91**/**0.82**
R15	**1.00**/0.68	**1.00**/0.47	**0.97**/0.63	**1.00**/0.79	**1.00**/0.62	**1.00**/0.67	**1.00**/0.52	**1.00**/0.89	**1.00**/**0.94**
Spiral	**1.00**/0.83	**1.00**/0.85	**1.00**/0.51	**1.00**/0.86	**1.00**/0.88	**1.00**/0.64	**1.00**/0.54	**1.00**/0.82	**1.00**/**0.93**
Pathbased	**1.00**/0.68	**1.00**/**0.74**	0.67/0.47	0.67/0.66	0.67/0.60	0.67/0.52	0.67/0.64	**1.00**/0.61	**1.00**/0.71
D31	**0.93**/0.42	**1.00**/0.22	**1.00**/0.32	**1.00**/0.29	**1.00**/0.46	**1.00**/0.34	**1.00**/0.19	**1.00**/**0.84**	**1.00**/0.79
S3	**1.00**/0.52	**1.00**/0.43	**1.00**/0.32	**1.00**/0.71	**1.00**/0.50	**1.00**/0.70	**1.00**/0.32	**1.00**/**0.86**	**1.00**/0.82
T48K	0.77/0.60	0.86/0.57	0.77/0.48	0.62/0.43	0.77/**0.53**	0.57/0.45	0.77/0.42	**1.00**/0.43	**1.00**/0.52
Iris	**1.00**/0.65	**1.00**/0.79	**1.00**/0.66	**1.00**/0.87	0.67/0.72	**1.00**/0.75	**1.00**/0.61	**1.00**/**0.88**	**1.00**/0.84
Wine	**1.00**/0.55	**1.00**/0.74	**1.00**/0.65	**1.00**/0.85	**1.00**/0.65	**1.00**/0.67	0.67/0.56	**1.00**/0.88	**1.00**/**0.90**
Segment	0.71/0.46	0.71/0.57	0.86/0.25	0.42/0.39	0.77/0.52	0.71/0.39	0.57/-	**0.93**/**0.78**	**0.93**/0.72
Drivedata	0.75/0.63	0.75/0.68	0.51/0.75	**0.86**/0.80	0.75/0.67	0.75/0.58	0.75/0.62	**0.86**/**0.81**	**0.86**/**0.81**
Breastcancer	0.50/0.19	0.50/0.21	**1.00**/0.33	**1.00**/0.73	0.50/0.24	0.50/0.15	0.50/0.15	**1.00**/0.75	**1.00**/**0.83**
YTF	0.51/0.23	0.63/0.16	**0.68**/0.14	0.47/0.21	0.56/0.19	0.61/0.21	0.59/0.31	0.63/0.40	0.63/**0.51**
USPS	0.40/0.22	0.50/0.31	0.50/0.17	0.64/0.33	0.48/0.27	0.40/0.23	0.30/0.24	**0.90**/0.51	**0.90**/**0.58**
MNIST	0.70/0.20	0.50/0.17	0.70/0.15	0.90/0.43	0.84/0.28	0.80/0.28	-/0.17	**1.00**/0.47	**1.00**/**0.59**

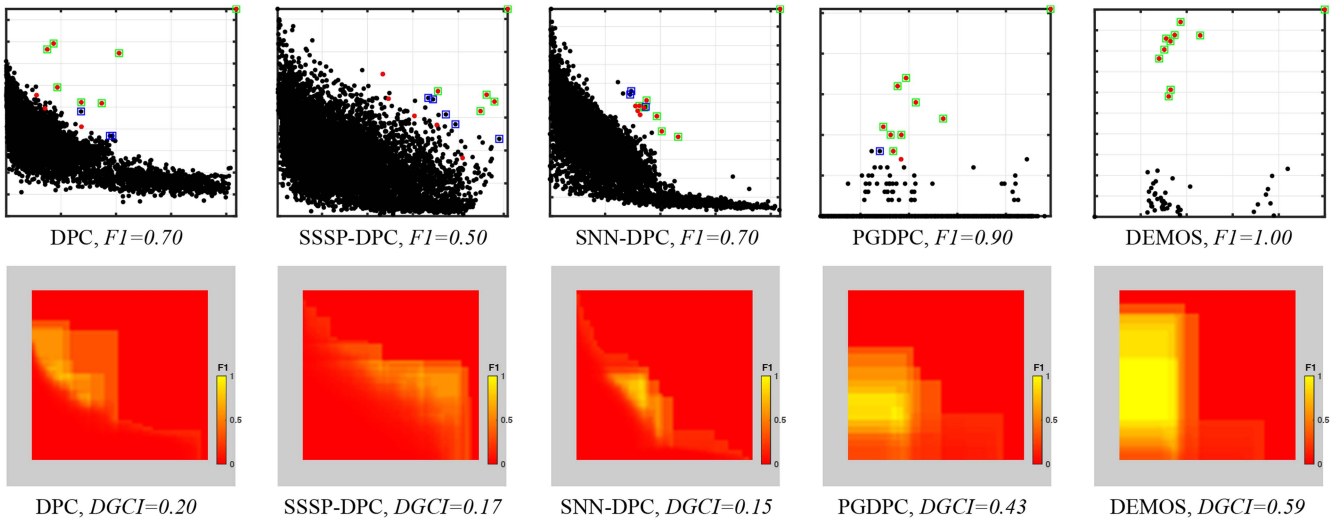


Fig. 11. The clarity comparison of different decision graphs on *MNIST*. Cluster centers are marked by red color. The green box marks the correct cluster center selection, while the blue indicates the wrong selection.

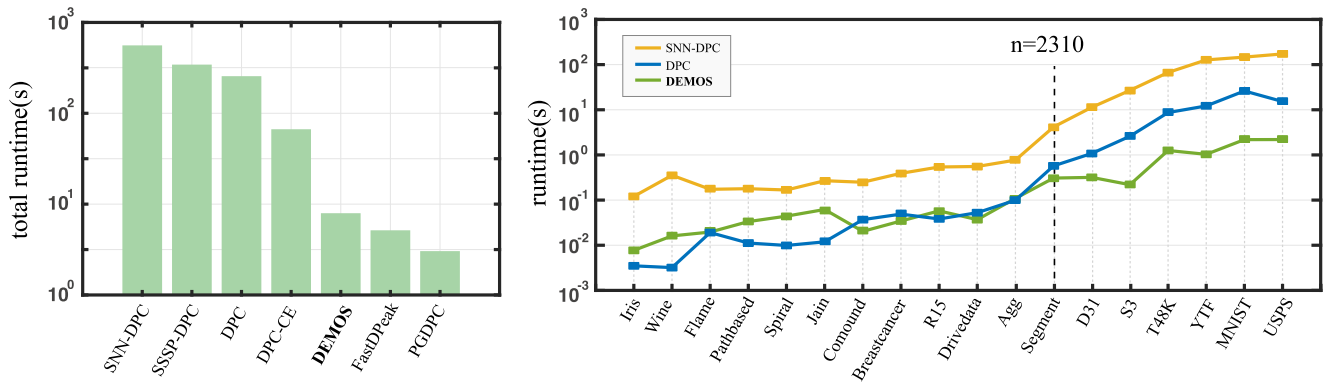
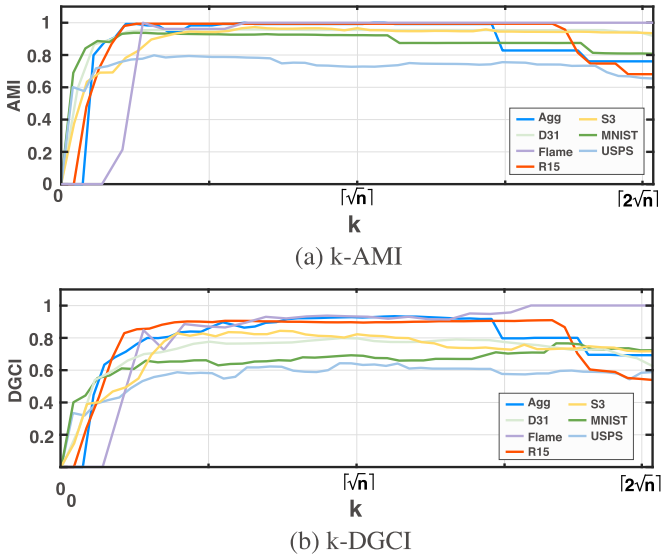


Fig. 12. The runtime of different DPC-based algorithms.



(a) k-AMI

(b) k-DGCI

Fig. 13. The k-AMI (a) and k-DGCI (b) plots of DEMOS on different datasets with $k \in [0, 2\lceil\sqrt{n}\,\rceil]$.

density peak similarity estimation. DEMOS is slower but has higher clustering accuracy. As shown in the line chart, DEMOS only takes about one second to execute a dataset of 10,000 data. DEMOS highlights itself (is much faster) on large datasets (such as *YTF* and *USPS*) compared with DPC.

As verified, the proposed DEMOS algorithm with fast speed is promising for large-scale data clustering.

F. Parameter Insensitivity

Parameter k (i.e., the number of neighbors) is the only key parameter of DEMOS, $k = \lceil\sqrt{n}\,\rceil$ as default. k is needed for density estimation, density peak identification, peak graph building, and decision graph generation. So, the performance of DEMOS is highly dependent on the setting of k. Therefore, the parameter insensitivity of k deserves a discussion.

Fig. 13 presents the k-AMI and k- DGCI plots of several tested datasets, with $k \in [0, 2\lceil\sqrt{n}\,\rceil]$ and a given correct cluster number as inputs. As Fig. 13(a) and (b) show, within the range of $[0, 2\lceil\sqrt{n}\,\rceil]$, DEMOS obtains a stable optimal performance over a wide range around $k = \lceil\sqrt{n}\,\rceil$, which verifies the effectiveness of the $k = \lceil\sqrt{n}\,\rceil$ setting and the insensitivity of DEMOS to parameter k.

V. Conclusion

A linkage-based clustering by pruning a density-boosting cluster tree of density mounts—DEnsity MOuntains Separation clustering algorithm (DEMOS) is proposed following our own cluster assumption: a cluster is a density-connected area with multiple (or a single) density mounts and a relatively large dis-connectivity from density-connected areas of higher densities. The proposed linkage metric helps DEMOS effectively reflect the dis-connectivity between clusters and exclude the interference of outliers. Besides, a multi-valley-link-based connectivity estimation method is designed to achieve a fast (dis)connectivity estimation of density peaks with high fidelity. As a result, DEMOS can reasonably reconstruct clusters with arbitrarily complex shapes and easily find cluster centers in its clear decision graph. In addition, DEMOS can work well on large-scale datasets. The clustering performance of DEMOS is well verified in the conducted comparison experiments on synthetic datasets and real-world datasets, concerning the complex shape reconstruction, the center detection, and the running speed. Moreover, our decision graph clarity metric can quantitatively measure the clarity of the decision graph, demonstrating that our decision graph is much clearer than other DPC-based methods. As verified, our clear decision graph also equips DEMOS with robust center detection performance.

Nevertheless, the selection of cluster centers is manual; although DEMOS is not sensitive to the core parameter k, it is still preset. Therefore, we plan on improving DEMOS to realize the automatic detection of cluster centers and to get self-tuning k.

References

[1] A. K. Jain, "Data clustering: 50 years beyond K-means," *Pattern Recognit. Lett.*, vol. 31, no. 8, pp. 651–666, 2010, doi: 10.1016/j.patrec.2009.09.011.
[2] H. Liu and L. Yu, "Toward integrating feature selection algorithms for classification and clustering," *IEEE Trans. Knowl. Data Eng.*, vol. 17, no. 4, pp. 491–502, Apr. 2005, doi: 10.1109/TKDE.2005.66.
[3] M. Caron, P. Bojanowski, A. Joulin, and M. Douze, "Deep clustering for unsupervised learning of visual features," in *Proc. Eur. Conf. Comput. Vis.*, 2018, pp. 132–149.
[4] R. Achanta and S. Süsstrunk, "Superpixels and polygons using simple non-iterative clustering," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 4651–4660.
[5] M. Gao and 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.*, vol. 205, 2020, Art. no. 106919, doi: 10.1016/j.oceaneng.2020.106919.
[6] M. Paolanti and E. Frontoni, "Multidisciplinary pattern recognition applications: A review," *Comput. Sci. Rev.*, vol. 37, 2020, Art. no. 100276, doi: 10.1016/j.cosrev.2020.100276.
[7] T. Lei, P. Liu, X. Jia, X. Zhang, H. Meng, and A. K. Nandi, "Automatic fuzzy clustering framework for image segmentation," *IEEE Trans. Fuzzy Syst.*, vol. 28, no. 9, pp. 2078–2092, Sep. 2020, doi: 10.1109/TFUZZ.2019.2930030.
[8] Y. Xie et al., "Joint deep multi-view learning for image clustering," *IEEE Trans. Knowl. Data Eng.*, vol. 33, no. 11, pp. 3594–3606, Nov. 2021, doi: 10.1109/TKDE.2020.2973981.
[9] M. I. Jordan and T. M. Mitchell, "Machine learning: Trends, perspectives, and prospects," *Science*, vol. 349, no. 6245, pp. 255–260, 2015, doi: 10.1126/science.aaa8415.
[10] A. K. Jain and R. C. Dubes, *Algorithms for Clustering Data*. Englewood Cliffs, NJ, USA: Prentice-Hall, 1988.
[11] J. Han, J. Pei, and H. Tong, *Data Mining: Concepts and Techniques*. San Mateo, CA, USA: Morgan Kaufmann, 2022.
[12] J. MacQuee, "Some methods for classification and analysis of multivariate observations," in *Proc. 5th Berkeley Symp. Math. Statist. Probability*, 1967, pp. 281–297.
[13] I. S. Dhillon, Y. Guan, and B. Kulis, "Kernel k-means: Spectral clustering and normalized cuts," *Proc. 10th ACM SIGKDD Int. Conf. Knowl. Discov. Data Mining*, 2004, pp. 551–556, doi: 10.1145/1014052.1014118.
[14] F. Camastra and A. Verri, "A novel kernel method for clustering," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 27, no. 5, pp. 801–805, May 2005, doi: 10.1109/TPAMI2005.88.
[15] H. Liu, J. Wu, T. Liu, D. Tao, and Y. Fu, "Spectral ensemble clustering via weighted k-means: Theoretical and practical evidence," *IEEE Trans. Knowl. Data Eng.*, vol. 29, no. 5, pp. 1129–1143, May 2017, doi: 10.1109/TKDE.2017.2650229.
[16] M. Ester, H.-P. Kriegel, J. Sander, and X. Xu, "A density-based algorithm for discovering clusters in large spatial databases with noise," in *Proc. 2nd Int. Conf. Knowl. Discov. Data Mining*, 1996, pp. 266–231.

[17] Y. Chen, S. Tang, N. Bouguila, C. Wang, J. Du, and H. Li, "A fast clustering algorithm based on pruning unnecessary distance computations in DBSCAN for high-dimensional data," *Pattern Recognit.*, vol. 83, pp. 375–387, 2018, doi: 10.1016/j.patcog.2018.05.030.
[18] A. Bechini, F. Marcelloni, and A. Renda, "TSF-DBSCAN: A novel fuzzy density-based approach for clustering unbounded data streams," *IEEE Trans. Fuzzy Syst.*, vol. 30, no. 3, pp. 623–637, Mar. 2022, doi: 10.1109/tfuzz.2020.3042645.
[19] H. Averbuch-Elor, N. Bar, and D. Cohen-Or, "Border-peeling clustering," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 42, no. 7, pp. 1791–1797, Jul. 2020, doi: 10.1109/TPAMI.2019.2924953.
[20] R. Sibson, "SLINK: An optimally efficient algorithm for the single-link cluster method," *Comput. J.*, vol. 16, no. 1, pp. 30–34, 1973, doi: 10.1093/comjnl/16.1.30.
[21] D. Defays, "An efficient algorithm for a complete link method," *Comput. J.*, vol. 20, no. 4, pp. 364–366, 1977, doi: 10.1093/comjnl/20.4.364.
[22] A. Rodriguez and A. Laio, "Clustering by fast search and find of density peaks," *Science*, vol. 344, no. 6191, pp. 1492–1496, 2014, doi: 10.1126/science.1242072.
[23] Y. Cheng, "Mean shift, mode seeking, and clustering," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 17, no. 8, pp. 790–799, Aug. 1995, doi: 10.1109/34.400568.
[24] A. Vedaldi and S. Soatto, "Quick shift and kernel methods for mode seeking," in *Proc. Eur. Conf. Comput. Vis.*, Springer, Berlin, Germany, 2008, pp. 705–718.
[25] M. Karaayvaz et al., "Unravelling subclonal heterogeneity and aggressive disease states in TNBC through single-cell RNA-seq," *Nature Commun.*, vol. 9, no. 1, pp. 1–10, 2018.
[26] J. Liao et al., "Density cluster based approach for controller placement problem in large-scale software defined networkings," *Comput. Netw.*, vol. 112, pp. 24–35, 2017, doi: 10.1016/j.comnet.2016.10.014.
[27] J. Guan, S. Li, X. He, J. Zhu, and J. Chen, "Fast hierarchical clustering of local density peaks via an association degree transfer method," *Neurocomputing*, vol. 455, pp. 401–418, 2021, doi: 10.1016/j.neucom.2021.05.071.
[28] Y. Wang et al., "McDPC: Multi-center density peak clustering," *Neural Comput. Appl.*, vol. 32, no. 17, pp. 13465–13478, 2020.
[29] D. U. Pizzagalli, S. F. Gonzalez, and R. Krause, "A trainable clustering algorithm based on shortest paths from density peaks," *Sci. Adv.*, vol. 5, no. 10, 2019, Art. no. eaax3770, doi: 10.1126/sciadv.aax3770.
[30] R. Liu, H. Wang, and X. Yu, "Shared-nearest-neighbor-based clustering by fast search and find of density peaks," *Inf. Sci.*, vol. 450, pp. 200–226, 2018, doi: 10.1016/j.ins.2018.03.031.
[31] M. Du, S. Ding, and H. Jia, "Study on density peaks clustering based on k-nearest neighbors and principal component analysis," *Knowl.-Based Syst.*, vol. 99, pp. 135–145, 2016, doi: 10.1016/j.knosys.2016.02.001.
[32] Y. Chen et al., "Fast density peak clustering for large scale data based on kNN," *Knowl.-Based Syst.*, vol. 187, pp. 104824, 2020, doi: 10.1016/j.knosys.2019.06.032.
[33] J. Guan, S. Li, X. He, and J. Chen, "Peak-graph-based fast density peak clustering for image segmentation," *IEEE Signal Process. Lett.*, vol. 28, pp. 897–901, Apr. 2021, doi: 10.1109/LSP.2021.3072794.
[34] W. Guo et al., "Density peak clustering with connectivity estimation," *Knowl.-Based Syst.*, vol. 243, 2022, Art. no. 108501, doi: 10.1016/j.knosys.2022.108501.
[35] K. K. Sharma, A. Seal, A. Yazidi, and O. Krejcar, "A new adaptive mixture distance-based improved density peaks clustering for gearbox fault diagnosis," *IEEE Trans. Instrum. Meas.*, vol. 71, 2022, Art. no. 3528716, doi: 10.1109/TIM.2022.3216366.
[36] Y. Yang, J. Cai, H. Yang, and X. Zhao, "Density clustering with divergence distance and automatic center selection," *Inf. Sci.*, vol. 596, pp. 414–438, 2022, doi: 10.1016/j.ins.2022.03.027.
[37] A. Lotfi, P. Moradi, and H. Beigy, "Density peaks clustering based on density backbone and fuzzy neighborhood," *Pattern Recognit.*, vol. 107, pp. 107449, 2020, doi: 10.1016/j.patcog.2020.107449.
[38] M. Ankerst et al., "OPTICS: Ordering points to identify the clustering structure," *ACM SIGMOD Rec.*, vol. 28, no. 2, pp. 49–60, 1999.
[39] L. McInnes, J. Healy, and S. Astels, "HDBSCAN: Hierarchical density based clustering," *J. Open Source Softw.*, vol. 2, no. 11, 2017, Art. no. 205, doi: 10.21105/joss.00205.
[40] L. Qian, C. Plant, and C. Böhm, "Density-based clustering for adaptive D. Variation," in *Proc. IEEE Int. Conf. Data Mining*, 2021, pp. 1282–1287.
[41] K. M. Ting et al., "Overcoming key weaknesses of distance-based neighbourhood methods using a data dependent dissimilarity measure," in *Proc. 22nd ACM SIGKDD Int. Conf. Knowl. Discov. Data Mining*, 2016, pp. 1205–1214, doi: 10.1145/2939672.2939779.
[42] E. W. Dijkstra, "A note on two problems in connexion with graphs," *Numerische Mathematik*, vol. 1, no. 1, pp. 269–271, 1959.
[43] L. Zelnik-Manor and P. Perona, "Self-tuning spectral clustering," in *Proc. Int. Conf. Neural Inf. Process. Syst.*, 2004, pp. 1601–1608.
[44] P. Fränti and S. Sieranoja, "K-means properties on six clustering benchmark datasets," *Appl. Intell.*, vol. 48, no. 12, pp. 4743–4759, 2018. [Online] available: http://cs.uef.fi/sipu/datasets/
[45] N. Bhatia, "Survey of nearest neighbor techniques," 2010, *arXiv:1007.0085*.
[46] A. Patil, D. Huard, and C. J. Fonnesbeck, "PyMC: Bayesian stochastic modelling in Python," *J. Statist. Softw.*, vol. 35, no. 4, 2010, Art. no. 1.
[47] P. Fränti and O. Virmajoki, "Iterative shrinking method for clustering problems," *Pattern Recognit.*, vol. 39, no. 5, pp. 761–775, 2006.
[48] K. Bache and M. Lichma, "UCI machine learning repository," 2013. [Online] available: http://archive.ics.uci.edu/ml
[49] L. Wolf, T. Hassner, and I. Maoz, "Face recognition in unconstrained videos with matched background similarity," in *Proc. IEEE Int. Conf. Comput. Vis. Pattern Recognit.*, 2011, pp. 529–534.
[50] T. Hastie, J. Friedman, and R. Tibshirani, *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. New York, NY, USA: Springer, 2009.
[51] Y. LeCun, C. Cortes, and C. Burges, "MNIST handwritten digit database," 2010. [Online] available: http://http://yann.lecun.com/exdb/mnist/
[52] N. Xuan, V. Julien, S. Wales, and J. Bailey, "Information theoretic measures for clusterings comparison: Variants, properties, normalization and correction for chance," *J. Mach. Learn. Res.*, vol. 11, pp. 2837–2854, 2010.
[53] L. van der Maaten and G. Hinton, "Visualizing data using t-SNE," *J. Mach. Learn. Res.*, vol. 9, no. 11, pp. 2579–2605, 2008.
[54] X. Qi et al., "Laplacian centrality: A new centrality measure for weighted networks," *Inf. Sci.*, vol. 194, pp. 240–253, 2012.

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

Sheng Li received the bachelor's degree from the Zhejiang University of Technology (ZJUT), Hangzhou, China, in 2006, and the MSc degree in communications engineering and the PhD degree in electronic engineering from the University of York, York, U.K., in 2007 and 2010, respectively. From November 2010 to October 2011, he was a postdoctoral researcher with the Ilmenau University of Technology, Ilmenau, Germany. Since April 2012, he has been with ZJUT, where he is currently an associate professor. He received the K. M. Stott Prize for Excellence in Scientific Research in 2010. He received the Best Paper Award from the VTC 2011 spring for the track signal processing for wireless communication. His research interests include signal processing, machine learning, and pattern recognition.

Xiaojun Chen received the PhD degree in internal medicine from Soochow University, Soochow, China. He is an associate chief physician with the First Affiliated Hospital of Wenzhou Medical University. His research interests include bioinformatics and pattern recognition.

Xiongxiong He received the MS degree from Qufu Normal University, Qufu, China, in 1994, and the PhD degree from Zhejiang University, Hangzhou, China, in 1997. He held a post-doctoral position with the Harbin Institute of Technology from 1998 to 2000. He joined the Zhejiang University of Technology Hangzhou, China, in 2001, where he has been a professor with the College of Information Engineering. His research areas include nonlinear control, signal processing, and pattern recognition.

Jiajia Chen received the MA degree from the East China Normal University, Shanghai, China. Her current research interests include data mining and pattern recognition.