SMMP: A Stable-Membership-Based Auto-Tuning Multi-Peak Clustering Algorithm

Junyi Guan, Sheng Li, Xiongxiong He, Jinhui Zhu, Jiajia Chen, and Peng Si

Abstract—Since most existing single-prototype clustering algorithms are unsuitable for complex-shaped clusters, many multi-prototype clustering algorithms have been proposed. Nevertheless, the automatic estimation of the number of clusters and the detection of complex shapes are still challenging, and to solve such problems usually relies on user-specified parameters and may be prohibitively time-consuming. Herein, a stable-membership-based auto-tuning multi-peak clustering algorithm (SMMP) is proposed, which can achieve fast, automatic, and effective multi-prototype clustering without iteration. A dynamic association-transfer method is designed to learn the representativeness of points to sub-cluster centers during the generation of sub-clusters by applying the density peak clustering technique. According to the learned representativeness, a border-link-based connectivity measure is used to achieve high-fidelity similarity evaluation of sub-clusters. Meanwhile, based on the assumption that a reasonable clustering should have a relatively stable membership state upon the change of clustering thresholds, SMMP can automatically identify the number of sub-clusters and clusters, respectively. Also, SMMP is designed for large datasets. Experimental results on both synthetic and real datasets demonstrated the effectiveness of SMMP.

Index Terms—Clustering, density peak, arbitrary shape clustering, auto-tuning

1 Introduction

Data clustering, aiming to automatically group similar objects into clusters, is a critical unsupervised learning technique for extracting potential and valuable knowledge from data [1], [2]. It has been applied to marketing analysis [4], computer vision [5], pattern recognition [6], image processing [7], machine learning [8], etc.

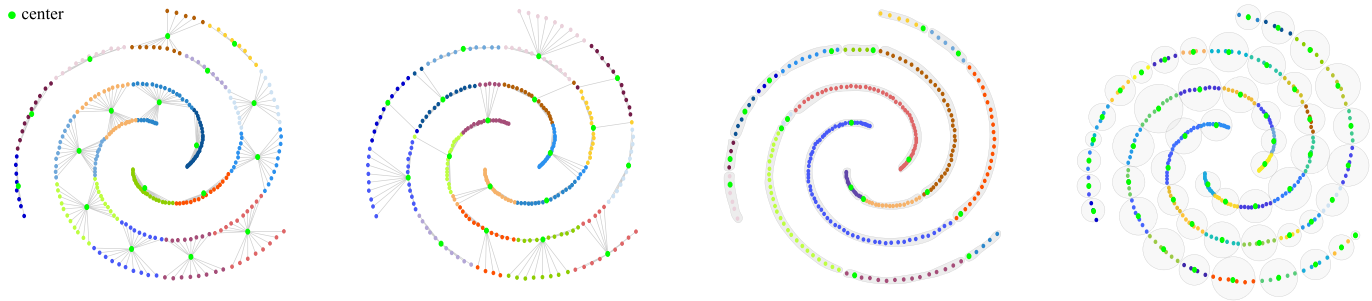
Clustering algorithms are commonly classified as partitional and hierarchical [3]. Partitional clustering aims to obtain a single partition of data, and most partitional clustering algorithms use a single prototype to represent a cluster. The well-known K-centers technique [9], [10] considers a center as the centroid (or medoid) of a cluster and assigns each point to its closest center. All K-centers algorithms require the cluster number as a prior input. While the Affinity Propagation algorithm (AP) [11] can automatically identify the most representative points as the high-quality centers. Although the K-centers and the AP all shine with simpleness and efficiency in partitioning hyper-spherical clusters, they cannot work well on non-spherical clusters. Some density-based partitional methods try to divide the dataset into clusters with maximum density-connected points [12], [13] that can effectively reconstruct non-spherical shapes, however, they often merge high-overlapping clusters [14], [15].

Unlike partitional clustering which directly generates clusters, hierarchical clustering generates a hierarchical structure of clusters, that is, a dendrogram. The popular linkage-based clustering [16], [17], [18] achieves clustering by gradually merging similar data points according to a specific linkage metric. Although linkage-based clustering can effectively identify non-spherical clusters, a given number of clusters is usually required to cut the dendrogram into final clusters. In 2014, the Density Peak Clustering algorithm (DPC) [19] was proposed. DPC can manually find appropriate centers without prior knowledge by a heuristic method (i.e., finding density peaks). In general, hierarchical clustering is more suitable for complex-shaped clusters (or multi-prototype clusters), but it is usually unable to handle large-scale data due to its higher time complexity.

However, in real applications, we usually encounter large-size multi-prototype clusters [3]. In such a dilemma, the extended algorithms of K-centers and AP directly formalize the multi-prototype clustering problem as an objective function, such as in [20], [21]. Another more common and simpler way to achieve fine multi-prototype clustering is to apply a hybrid clustering technique. It often uses a partitional technique to fast divide the dataset into small sub-clusters, and then uses a hierarchical technique to gradually merge similar sub-clusters into a given number of clusters, such as in [22], [23], [24], [25], [26]. Although these methods can achieve satisfying clustering, they often highly rely on the user-specified parameters, mainly the numbers of sub-clusters and clusters; and methods involving objective function optimization techniques may have unstable performances and be prohibitively time-consuming.

- *Junyi Guan, Sheng Li, Xiongxiong He, Jiajia Chen, and Peng Si are with the College of Information Engineering, Zhejiang University of Technology, Hangzhou 310023, China. E-mail: {jonnyguan73, fl_katrina}@163.com, {shengli, hxx, 2111903023}@zjut.edu.cn.*
- *Jinhui Zhu is with Second Affiliated Hospital, School of Medicine, Zhejiang University, Hangzhou 310014, China. E-mail: 2512016@zju.edu.cn.*

Manuscript received 29 November 2021; revised 2 September 2022; accepted 4 October 2022. Date of publication 11 October 2022; date of current version 3 April 2023.
This work was supported in part by the National Science Foundation of P.R. China under Grant 61873239 and in part by the Science Technology Department of Zhejiang Province under Grant 2020C03074.
(Corresponding author: Sheng Li.)
Recommended for acceptance by M. Salzmann.
Digital Object Identifier no. 10.1109/TPAMI.2022.3213574



(a) K-means, 18 sub-clusters (b) AP, 18 sub-clusters (c) DPC, 18 sub-clusters (d) K-means, 50 sub-clusters

Fig. 1. The split results of K-means (a), AP (b), and DPC (c) on the *Spiral* dataset with respect to eighteen sub-clusters, and the split result of K-means with fifty sub-clusters.

To achieve successful clustering without encountering the above-mentioned issues, a stable-membership-based auto-tuning multi-peak clustering algorithm (SMMP) is proposed. SMMP views a cluster as a density-connected area with multiple density peaks to achieve fast and automatic multi-prototype clustering without iterations. The main contributions of SMMP are as follows:

1) A dynamic association-transfer method (DAT) is designed to learn the representativeness of points to sub-cluster centers during the generation of sub-clusters by applying the density peak clustering technique;
2) A border-link-based connectivity measure is proposed to reliably evaluate the cohesion of sub-clusters based on the learned representativeness;
3) An assumption that a reasonable clustering should have a relatively stable membership state upon the change of clustering thresholds is proposed to guide SMMP's adaptive estimation of the number of sub-clusters and clusters, respectively;
4) SMMP only requires kNN distances of data as input and is suitable for large dataset clustering.

The rest paper is composed as: Section 2 introduces the related works; Section 3 mainly focuses on the proposed method; while Section 4 displays the experiments and discussions; and Section 5 gives the final conclusion.

2 Related Works

DPC [19] inherits the main idea of the Mean-shift clustering method (MSC) [28] to search for local density areas as clusters. But unlike Mean-shift which views all local density maximum points as cluster centers, DPC selects appropriate cluster centers according to its assumption—cluster centers are density peaks that have high densities and are far away from points of higher densities.

Given a dataset $X = \{x_1, x_2, \ldots, x_n \,|\, x_i \in \mathbb{R}^d\}$, for each point x_i, DPC first estimates its local density ρ_i as in Eq. (1), where the "cutoff distance" d_c is a user-specified parameter, and d_{ij} indicates the euclidean distance between point x_i and x_j. Subsequently, for point x_i (except for x_i with the highest density), DPC measures δ_i by computing the minimum distance between point x_i and another point with a higher density, as in Eq. (2). Then, for point x_i with the highest density, DPC defines $\delta_i = \max_{x_j}(d_{ij})$.

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

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

Then, based on DPC's center assumption, density peaks with large ρ-δ are manually selected as centers by observing through a decision graph (i.e., a ρ-δ plot). Subsequently, each non-center point is allocated to the same cluster of its nearest higher density point.

DPC can work well on single-peak sub-clusters (i.e., a cluster with only one density peak) [29], [30], [31]. But its heuristic method of finding density peaks may incorrectly select some density peaks as cluster centers in dealing with multi-prototype (or multi-peak) clusters, leading to a poor clustering result; also, DPC's allocation strategy may mistakenly allocate data points of a multi-peak cluster [31], [32], [36].

For multi-prototype clustering, one often first splits the dataset into multiple single-prototype sub-clusters by applying a single-prototype clustering method [22]. K-centers and AP are two commonly used and well-functioning single-prototype clustering methods [20], [21]. Nevertheless, they have to over-divide the dataset to pursue a reasonable multi-prototype clustering, leading to the missing of some important structure information (since they can only work well on spherical shapes). DPC can better preserve the structure information by identifying single-peak sub-clusters regardless of shapes to divide clusters reasonably. Hence, DPC is a quite promising single-prototype clustering method.

Fig. 1 illustrates four split results obtained by K-means, AP, and DPC on the *Spiral* [41] dataset composed of three spiral-shaped multi-peak clusters. As shown in Figs. 1a, 1b, and 1c, while K-means and AP failed to identify the eighteen sub-clusters due to non-spherical shapes, DPC perfectly identified all sub-clusters. In Fig. 1(d), although K-means did a successful split, it over-divided the dataset into fifty sub-clusters. This demonstrates that DPC can be a good single-prototype clustering method.

Herein, we only apply the superior structure recognition technique of DPC to split arbitrary-shaped sub-clusters, and then use our connectivity measure to accurately merge high-connected sub-clusters into complex-shaped clusters. Note that DPC deals with the "global information" of data, which may cause an unreasonable allocation. But we only apply the DPC technique to deal with the "local information" to achieve reliable local clustering, even when dealing with

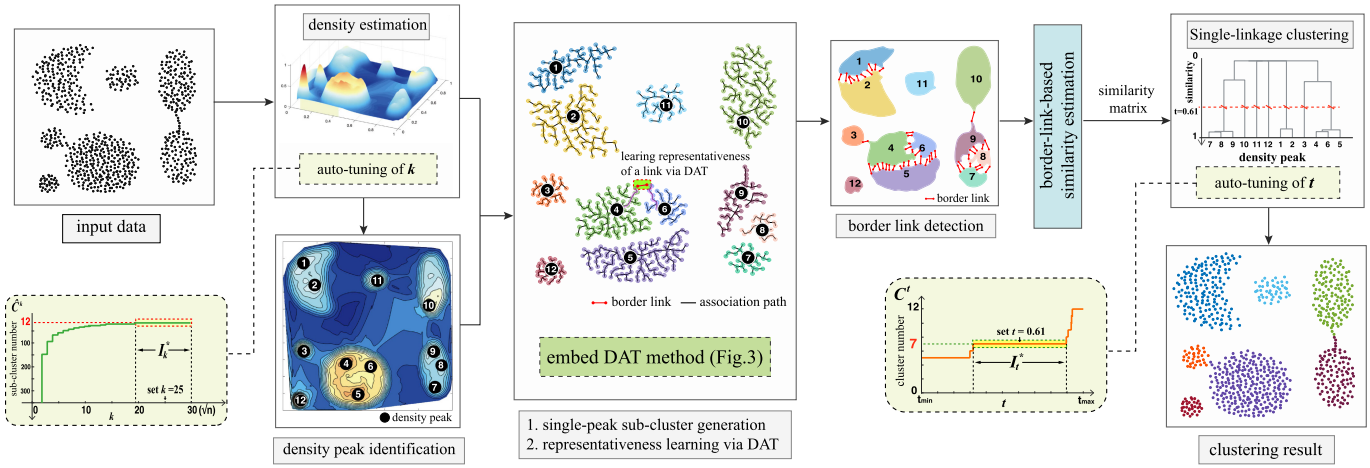


Fig. 2. The clustering process of SMMP on the *Agg* [41] dataset.

datasets of arbitrary shapes, different sizes, variable density, and overlapping clusters [33], [34], [35], [36], [37], [38]. Also, the allocation of each point in the local clustering process results from its local behavior (without expecting any point outside the local area), which significantly reduces the time complexity.

3 The Proposed SMMP Algorithm

In this section, the concept of "stable membership" that can help achieve auto-tuning is introduced, together with a detailed analysis of the proposed SMMP clustering strategy. Fig. 2 demonstrates the clustering process of SMMP.

3.1 Stable Membership

Consider a clustering function $F(X, s)$ that takes dataset X and similarity function s as inputs and returns a result described as a membership logical matrix $M \in \mathbb{R}^{n\times n}$, where the (i, j)th element $m_{ij} = 1$ means points x_i and x_j are members in the same cluster. Consider a clustering threshold $t \in I_t \subseteq [\min(s(x_i, x_j)), \max(s(x_i, x_j))]$ over $F(X, s)$, making: if $s(x_i, x_j) \geqslant t$, then $m_{ij} = 1$, where I_t indicates the interval of threshold t. Then, clustering function $F(X, s)$ can own the *Consistency* property proposed by Kleinberg [39]. Clearly, a reasonable threshold t should be within the range $(\max_{m_{ij}=0}(s(x_i, x_j)), \min_{m_{ij}=1}(s(x_i, x_j))]$. And the ideal range is when $\max_{m_{ij}=0}(s(x_i, x_j)) = \min(s(x_i, x_j))$ and $\min_{m_{ij}=1}(s(x_i, x_j)) = \max(s(x_i, x_j))$. Therefore, a reasonable similarity function s should bring a relatively large range $(\max_{m_{ij}=0}(s(x_i, x_j)), \min_{m_{ij}=1}(s(x_i, x_j))]$. Inspired by this, we propose Assumption 1.

Assumption 1. *A reasonable clustering should have a relatively stable membership upon the change of the clustering threshold according to a reasonable similarity measurement.*

Definition 1. *Notion $(\cdot)^t$ represents a corresponding result at a threshold $t \in I_t$. For example, M^t, Cl^t and C^t represent the membership matrix, the clustering result, and the number of clusters at threshold $t \in I_t$, respectively.*

Let notion $(\cdot)^t$ represent a corresponding result at a threshold $t \in I_t$ as in Definition 1. According to the Assumption 1, we can approximately detect a reasonable threshold interval by solving the following problem:

$$\begin{aligned} &\text{maximize} \quad \text{range}(I'_t) \\ &\text{s.t.} \quad \forall t_1, t_2 \in I'_t \subseteq I_t, C^{t_1} = C^{t_2}. \end{aligned} \tag{3}$$

Where I'_t is a sub-interval within I_t. The constraint $C^{t_1} = C^{t_2}$, as a sufficient condition of $M^{t_1} = M^{t_2}$, roughly represents a stable membership to reduce calculation.

The optimal threshold sub-interval $I^*_t \subseteq I_t$ of Eq. (3) gives the most stable C^{t^*}, and is considered as the reasonable clustering threshold interval, i.e., $(\max_{m_{ij}=0}(s(x_i, x_j)), \min_{m_{ij}=1}(s(x_i, x_j))] \leftarrow I^*_t$. Then, we set clustering threshold as $t = \text{mean}(I^*_t)$.

In what follows, based on the stable membership, an automatic method is proposed to estimate the ideal numbers of sub-clusters and clusters, respectively.

3.2 The Identification of Single-Peak Sub-Clusters

For each point x_i, its k nearest points are defined as its surrounding points, denoted as $N_k(x_i)$. Its local density ρ_i is estimated according to its within-surrounding similarity [40], as in Eq. (4).

$$\rho_i = \frac{1}{\frac{1}{k}\sum_{x_j \in N_k(x_i)} d_{ij}}. \tag{4}$$

Density peaks with the characteristic of local density maxima are defined in Definition 2.

Definition 2. *Point x_i is a density peak, denoted as $p \in P$, if $\rho_i > \max_{x_j \in N_k(x_i)}(\rho_j)$. P is the density peak set of X.*

On this basis, we obtain sub-clusters with only one density peak by applying the DPC technique: to select all density peaks as centers and assign each non-center point to the same cluster of its nearest higher density point within its surrounding points.

The above assignment relationship of data can be expressed as an adjacency graph $A_G \in \mathbb{R}^{n\times n}$, where the (i, j)th element $a_{ij} \in \{0, 1\}$, and $a_{ij} = 1$ means that point x_j is associated with its nearest higher density neighbor x_j. Single-peak sub-clusters are connected components of A_G, where point x_i without out-degree ($deg^+(x_i) = 0$) is a density

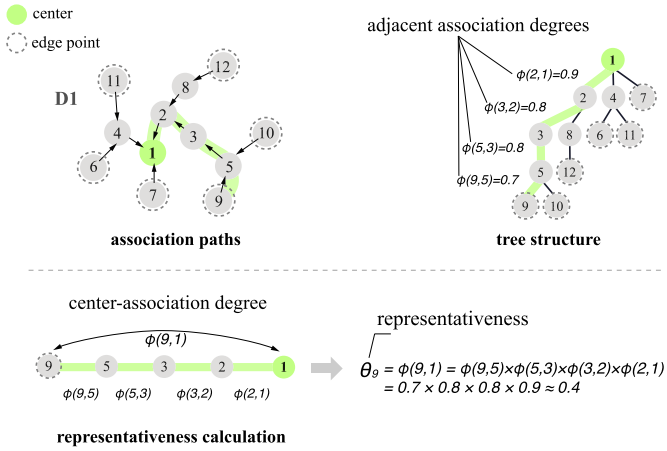


Fig. 3. The idea of the DAT method on a toy dataset *D1*.

peak; while point x_i without in-degree ($deg^-(x_i) = 0$) is an edge point (see Fig. 3).

For the generation of sub-clusters, parameter k is the only dependent variable of the clustering threshold, because the construction of A_G is only subject to parameter k. Clearly, a large k tends to produce a small number of sub-clusters. Let notion $(\cdot)^k$ represent a corresponding result at a $k \in I_k$, just like $(\cdot)^t$ as in Definition 1. Based on Assumption 1, a reasonable threshold interval of k can be detected by solving the following problem:

$$\begin{aligned} &\text{maximize} \quad \text{range}(I'_k) \\ &\text{s.t.} \quad \forall k_1, k_2 \in I'_k \subseteq I_k = [1, \lceil\sqrt{n}\rceil], \hat{C}^{k_1} = \hat{C}^{k_2}, \end{aligned} \tag{5}$$

where $I_k = [1, \lceil\sqrt{n}\rceil]$ is the default interval of k [51], and symbol $\lceil\cdot\rceil$ is a ceiling function. $\hat{C}^k$ indicates the sub-cluster number at $k \in I_k$. Then, the optimal threshold sub-interval $I^*_k \subseteq I_k$ gives the most stable $\hat{C}^{k^*}, k^* \in I^*_k$. So, by auto-tuning $k = \lceil \text{mean}(I^*_k) \rceil$, we can automatically generate sub-clusters as $\hat{Cl} = \{\hat{Cl}_1, \hat{Cl}_2, \ldots, \hat{Cl}_{\hat{C}}\}, \hat{C} = \hat{C}^k$. Then, the original clustering of data points is simplified into the clustering of sub-clusters.

3.3 Border-Link-Based Connectivity Measure

In this subsection, a border-link-based connectivity measure is proposed to reliably evaluate the cohesion of sub-clusters. Besides, a dynamic association-transfer method (DAT) is designed to learn the representativeness of points to sub-cluster centers during the generation of sub-clusters.

3.3.1 Border Links

In Definition 3, border points are defined to be only existed in intersecting sub-clusters, where $k_b = \lfloor \min(\frac{k}{2}, 2\ln(n)) \rfloor$ (symbol $\lfloor\cdot\rfloor$ is a floor function). Note that small-value $k_b \ll k$ can effectively help to detect the proximal border points between intersecting sub-clusters.

Definition 3. *If mutual-proximity points x_i and x_j are in different sub-clusters, i.e., $x_i \in \hat{Cl}_y \cap N_{k_b}(x_j), x_j \in \hat{Cl}_z \cap N_{k_b}(x_i)$, then, points x_i and x_j are cross-cluster border points, denoted as $x_i \rightleftharpoons x_j$, indicating that sub-clusters $\hat{Cl}_y$ and $\hat{Cl}_z$ are intersected.*

Among cross-cluster border points, we link unlinked border point x_i to its nearest unlinked cross-cluster border point τ_i (see Eq. (6)) as a border link $l_i = \{x_i, \tau_i\}$.

$$\tau_i = \underset{x_j : x_j \rightleftharpoons x_i}{\arg\min}(d_{ij}), \ \text{s.t. } x_i, x_j \text{ are both unlinked.} \tag{6}$$

To quantitatively evaluate the connectivity between intersecting sub-clusters, we let each border point have a "representativeness" value (denoted as $\theta \in [0, 1]$) to represent its own sub-cluster, and design the DAT method—an enhanced version of the association-transfer method (AT) of our previous work [32]—to learn the representativeness.

3.3.2 Representativeness Learning of Border Links via DAT

In the DAT method, each point has a transferable association degree $\phi \in [0, 1]$ with its adjacent point in A_G, called adjacent association degree, as in Eq. (7), and the transfer logic is as in Definition 4:

Definition 4. *Point x_y and its adjacent point x_z have an adjacent association degree of $\phi(x_y, x_z)$, point x_z and its adjacent point x_r have $\phi(x_z, x_r)$, then points x_y and x_r have an association degree $\phi(x_y, x_r) = \phi(x_y, x_z) \times \phi(x_z, x_r)$, s.t. $a_{yz} = a_{zr} = 1$.*

$$\phi(x_i, x_j) = \frac{\rho_i}{\rho_j} \quad a_{ij} = 1 \tag{7}$$

$$\theta_i = \phi(x_i, p) = \prod_{x_y \in \Delta_{x_i p}} \phi(x_y, x_z), \ a_{yz} = 1 \tag{8}$$

Based on Definition 4, for each point $x_i \in X$, we define its center-association degree $\phi(x_i, p)$ as its representativeness θ_i, as in Eq. (8), where $\Delta_{x_i p}$ means all corresponding adjacent points on the path from point x_i to center p. If point x_i is a density peak p (a sub-cluster center), $\theta_p = 1$, as in Definition 5:

Definition 5. *A density peak $p \in P$ owns the largest representativeness to represent its sub-cluster, i.e., $\theta_p = 1$.*

Fig. 3 shows the DAT method on a toy dataset *D1*, where the point number indicates the density ranking order. As shown in Fig. 3a, by applying the DPC technique, all points are associated with their nearest higher density neighbors, except for the cluster center (point 1). The corresponding adjacency graph structure (a tree structure) is presented in Fig. 3b. As Fig. 3c shows, point 9 has the path $9 \rightarrow 5 \rightarrow 3 \rightarrow 2 \rightarrow 1$ towards the center point 1, then its representativeness $\theta_9 = \phi(9,1) = \phi(9,5) \times \phi(5,3) \times \phi(3,2) \times \phi(2,1) = 0.7 \times 0.8 \times 0.8 \times 0.9 \approx 0.4$.

Note that along a path, the low-density points are usually far away from the center, causing their small representativeness and speculating the small representativeness of edge points (dashed circles).

3.3.3 Border-Link-Based Similarity Evaluation

After obtaining the representativeness of border points, each border link $l_i = \{x_i, \tau_i\}$ can help to judge the cohesion (ie., the similarity) between sub-clusters. The representativeness g_{l_i} of border link l_i is defined as the average representativeness of its two border points, as in Eq. (9).

$$g_{l_i} = \frac{\theta_i + \theta_{\tau_i}}{2}, l_i = \{x_i, \tau_i\}. \tag{9}$$

Inspired by the idea of density-connectivity [12], we propose Assumption 2:

Assumption 2. *High-similarity sub-clusters are well-connected and often have multiple border links of high representativeness.*

We pick out a set of n_g (i.e., the minimum standard sample number) border link samples with the top largest g values for the similarity evaluation of sub-clusters, denoted as G, as in Eq. (10), where $g_{[1]} \geqslant g_{[2]} \geqslant \ldots \geqslant g_{[n_g]}$. If the total number of border links is less than n_g, we fill the number of samples (with $g = 0$) to n_g.

The estimation of n_g is defined in Eq. (11), where $\eta \in [0, 1]$ is a ratio parameter (default is 0.1), function $n_\epsilon(\hat{Cl})$ means the total number of edge points in sub-cluster $\hat{Cl}$, as in Eq (12), and $\epsilon(\cdot)$ is an edge point judgment function.

$$G_{\hat{Cl}_y\hat{Cl}_z} = \left\{g_{[1]}, g_{[2]}, \ldots, g_{[n_g]}\right\} \tag{10}$$

$$n_g = \lceil \eta \times \min(n_\epsilon(\hat{Cl}_y), n_\epsilon(\hat{Cl}_z)) \rceil \tag{11}$$

$$n_\epsilon(\hat{Cl}) = \sum_{x_i \in \hat{Cl}} \epsilon(x_i),\ \ \epsilon(x) = \begin{cases} 1 & deg^-(x) = 0 \\ 0 & \text{others} \end{cases}. \tag{12}$$

Based on Assumption 2, for two intersecting sub-clusters, if the representativeness values of all border link samples are (or almost) uniformly high, they are high-similar. We calculate the similarity value $s(\hat{Cl}_y, \hat{Cl}_z)$ as in Eq. (13). $\Gamma(G)$ returns the uniformity of all g values with the max g value in G, as in Eq. (14).

$$s(\hat{Cl}_y, \hat{Cl}_z) = \max(G_{\hat{Cl}_y\hat{Cl}_z}) \times \Gamma(G_{\hat{Cl}_y\hat{Cl}_z}) \tag{13}$$

$$\Gamma(G) = 1 - \frac{\frac{1}{n_g}\sum_{i=1}^{n_g} |G(i) - \max(G)|}{\max(G)}. \tag{14}$$

After obtaining the similarity matrix $S \in \mathbb{R}^{\hat{C}\times\hat{C}}$ of sub-clusters where the (y, z)th element is $s(\hat{Cl}_y, \hat{Cl}_z)$, we merge sub-clusters into final clusters.

3.4 The Identification of Clusters

After inputting the similarity matrix S into a traditional linkage-based method (herein we apply the Single-linkage method), we obtain a dendrogram with clustering threshold interval $I_t = [\min(s(\hat{Cl}_y, \hat{Cl}_z)), \max(s(\hat{Cl}_y, \hat{Cl}_z))] \subseteq [0, 1]$. Then, by solving Problem (3), SMMP automatically tunes $t = \text{mean}(I_t^*)$ to generate final clusters as $Cl = \{Cl_1, Cl_2, \ldots, Cl_C\}, C = C^t$.

In summary, the overall clustering process of the proposed SMMP algorithm needs no manual tuning or supervision.

3.5 Complexity Analysis

Fig. 4 presents the overall workflow of the SMMP algorithm, where Algorithms 1, 2, and 3 show the pseudocode of the four steps of the SMMP algorithm, respectively.

Step 1: the fast calculation of kNN distances (see Algorithm 1 Line 1∼ 2). The time complexity is $O(n\log(n))$ by applying fast kNN search technique [27].

Step 2: the identification of single-peak sub-clusters (see Algorithm 1 Line 3∼39). Line 3∼20 show the auto-tuning of k with time complexity $O(cn\tilde{k})$, where c (default as $c = 20$) is the number of k-samples picked out from I_k (by setting gap as in Line 5). $\tilde{k}$ means that each point's $\tilde{k}$th neighbor (an average

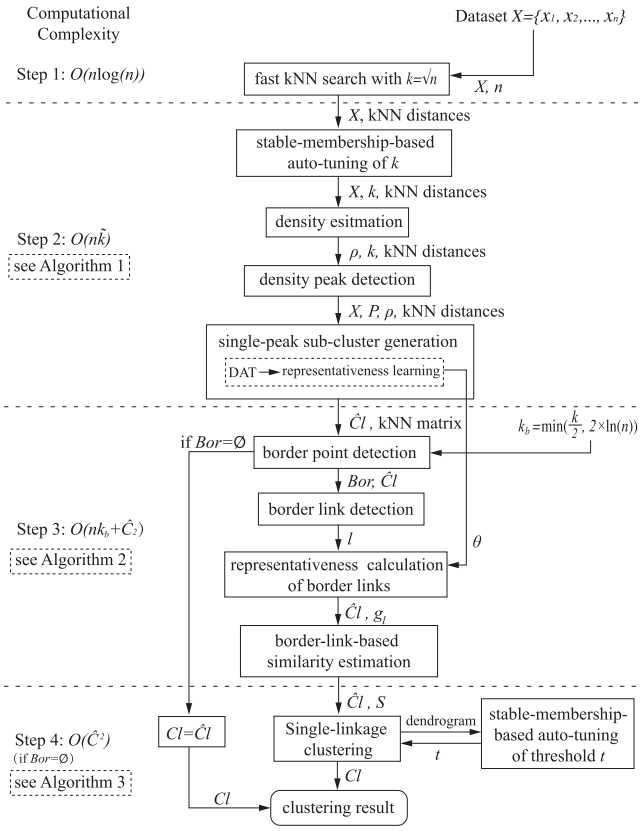


Fig. 4. The workflow of the overall SMMP algorithm.

concept) is its nearest higher density point. Since c is a constant, the overall time complexity is $O(n\tilde{k})$. In fact, most data points can find a real close higher density point, i.e., $\tilde{k} \ll \sqrt{n}$.

Line 21∼39 show the generation of sub-clusters and θ learning, where the initialization of representativeness θ (see Line 22∼24) needs complexity $O(n)$; the sub-cluster label acquisition of each point and the representativeness learning (see Line 25∼37) need complexity $O(n\tilde{k})$; the formation of sub-clusters (see Line 38∼39) needs complexity $O(n)$. So, the overall time complexity is $O(n\tilde{k})$.

Step 3: the similarity evaluation of sub-clusters (see Algorithm 2), where the identification of cross-cluster border points (see Line 1∼8) needs $O(nk_b)$; the identification of border links (Line 9∼17) with $O(|Bor|k_b)$, $|Bor| \ll n$ represents the total number of border points; the similarity evaluation (Line 18∼24) with $O(\hat{C}^2)$. So, the overall time complexity is $O(nk_b + \hat{C}^2)$.

Step 4: the adaptive merging of sub-clusters (see Algorithm 3), where the dendrogram building via Single-linkage (Line 1∼2) needs $O(\hat{C}^2)$; the auto-tuning of clustering threshold t and the adaptive merging of sub-clusters (Line 3∼12) need $O(\hat{C})$. So, the overall time complexity is $O(\hat{C}^2)$.

The overall time complexity of SMMP is $O(n(\log(n) + \tilde{k} + k_b) + \hat{C}^2)$, where $k_b, \hat{C}, \tilde{k}$, and k are all far less than n. Notably, SMMP only needs to calculate the kNN distances of data.

4 Experiments

4.1 Experimental Set Up

Datasets. Thirteen synthetic datasets of different shapes and eleven real-world datasets are selected to test the clustering

TABLE 1
Datasets

Dataset	Instances	Attributes	Clusters	Source
Agg	788	2	7	[41]
Jain	373	2	2	[41]
Spiral	312	2	3	[41]
Threecircles	299	2	3	[42]
Flame	240	2	2	[41]
D1	87	2	3	[38]
D2	85	2	4	[35]
R15	600	2	15	[41]
S3	5000	2	15	[43]
D31	3100	2	31	[41]
A3	7500	2	50	[41]
Birchrg1	100000	2	100	[41]
DIM1024	1024	1024	16	[41]
Breastcancer	569	30	2	[44]
Movementlibras	360	90	15	[44]
Parkin	195	22	2	[44]
Drivedata	606	6400	4	[44]
Waveform	5000	21	3	[44]
Lonosphere	351	34	2	[44]
Vote	345	17	2	[44]
Musk	6598	166	2	[44]
YTF	10000	10	41	[45]
REUTERS	10000	10	4	[46]
MNIST	10000	500	10	[47]

performance of the proposed algorithm, corresponding detailed summarization is in Table 1.

Comparison Methods and Settings. K-means [9] (the most typical K-centers clustering technique), the AP algorithm [11] (an excellent non-parametric partitional clustering technique), the DBSCAN [12] and MSC [28] algorithms (classic density-based non-parametric partitional clustering techniques), the DPC [19] and SSSP-DPC [31] algorithms (remarkable density peak clustering techniques), the Self-tuning Spectral Clustering algorithm (SSC) [42] (a popular Spectral Clustering technique), the KMM algorithm [20] (an outstanding multi-prototype clustering technique based on K-means), and the proposed SMMP algorithm.

In terms of the parameter setting, for K-means, we use the correct number C of clusters as input; for AP, we use its default parameter setting [11]; for SSC, DBSCAN, KMM, MSC, we select the optimal parameter setting over a full range of possible configurations; for DPC and SSSP-DPC, we manually select the correct number C of clusters according to an appropriate parameter d_c setting; while for SMMP, it adjusts parameters automatically. Besides, for iterative algorithms, such as K-means, AP, SSC, MSC, and KMM, we pick the best results among ten runs.

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

Data Preprocessing. all datasets are preprocessed by the min-max normalization method [43], aiming to reduce the influence of different metrics in different dimensions.

Evaluation Metric. the popular Adjusted Rand Index (ARI) [48], Adjusted Mutual Information (AMI) [48], Normalized Mutual Information (NMI) [49] and F-Score [50] are used to measure the clustering performance of the comparison algorithms.

4.2 Experiments on Synthetic Datasets

In this subsection, experiments on twelve synthetic datasets of different shape types are conducted to compare the clustering performance of the proposed SMMP and the other comparison algorithms: DPC [19], SSSP-DPC [31], K-means [9], KMM [20], SSC [42], DBSCAN [12], MSC [28], and AP [11].

4.2.1 Comparison With KMM

The SMMP is compared with KMM [20] on the *Jain* dataset that is composed of two moon shapes with different densities.

Fig. 5 shows the comparison results, where different colors indicate different clusters, and big dots are sub-cluster centers. As shown, KMM failed to reconstruct the structure of the two moon shapes when setting a small number (ten or thirty) of sub-clusters. But it managed to recognize the moon shapes when over-dividing the dataset into 100 sub-clusters, leaving each sub-cluster with almost no structural information (only about four data points in each sub-cluster). As an iterative algorithm, KMM's accuracy and stability are highly dependent on the initial settings of sub-cluster centers and cluster centers, which usually need prior knowledge.

In contrast, SMMP perfectly learned the data structure when setting only ten sub-clusters, for which the excellent shape recognition performance of DPC technology in single-peak clusters and our reasonable evaluation of sub-cluster similarity should get the credit. Also, sub-clusters with complete structure can provide important local cohesive information, which makes the study of sub-clusters meaningful.

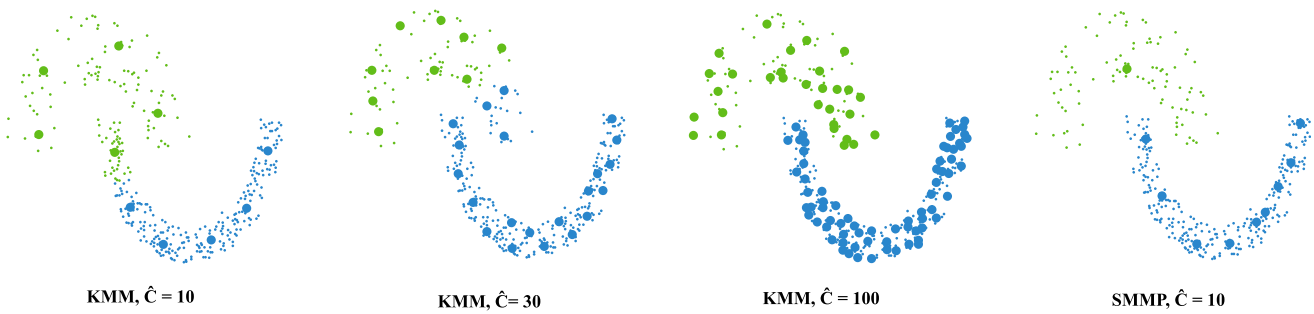


Fig. 5. The different results of KMM and SMMP on the *Jain* dataset.

Algorithm 1. SMMP: The Identification of Single-Peak Sub-Clusters

Input: dataset $X = \{x_1, x_2, \ldots, x_n\}$.
Output: density peak set P, sub-cluster result $\hat{Cl}$, and representativeness θ, **w.r.t** k.

// **the fast calculation of kNN distances**
fast obtain kNN distances of data with $k = \lceil \sqrt{n} \rceil$
// **the auto-tuning of** k
$I_k = [1, \lceil \sqrt{n} \rceil]$
$gap = \lceil \frac{\text{range}(I_k)}{c} \rceil$ // set iterate twenty times ($c = 20$ as default).
for $k = \min(I_k) : gap : \min(I_k)$ **do**
  calculate density ρ with k // Eq. (4)
  for each point $x_i \in X$ **do**
    $P^k = X$ // P^k is the density peak set **w.r.t** k
    for $x_j \in N_k(x_i)$ from near to far **do**
      if $\rho_i < \rho_j$ **then**
        $P^k = P^k \setminus \{x_i\}$ // x_i is not a density peak.
        break
      end if
    end for
  end for
  $\hat{C}^k \leftarrow |P^k|$ // $\hat{C}^k$ is the number of sub-clusters **w.r.t** k
end for
take obtained above pairs of $\hat{C}^k$ and k as input to obtain the optimal threshold interval I_k^* via solving the Problem (5).
$k = \lceil \text{mean}(I_k^*) \rceil$, $P \leftarrow P^k$, $\hat{C} \leftarrow \hat{C}^k$
//**the generation of sub-clusters and** θ **learning w.r.t** k
for each point $x_i \in X$ **do**
  $\theta_i = 1$ // initialize the representativeness
end for
for each point $x_i \in X$, from high-ρ to low-ρ **do**
  if $x_i \in P$ is a density peak **then**
    $x_i \leftarrow$ a unique sub-cluster label // x_i is a sub-cluster center
  else
    for each neighbor $x_j \in N_k(x_i)$ from near to far **do**
      if $\rho_j > \rho_i$ **then**
        $\theta_i = \theta_j \times \phi(x_i, x_j)$ // DAT, Eqs. (7) and (8)
        x_i's label $\leftarrow$ x_j's label
        break
      end if
    end for
  end if
end for
points with the same label form sub-clusters $\hat{Cl}$.
return density peaks $P = \{p_1, p_2, \ldots, p_{\hat{C}}\}$, sub-clusters $\hat{Cl} = \{\hat{Cl}_1, \hat{Cl}_2, \ldots, \hat{Cl}_{\hat{C}}\}$, and representativeness $\theta = \{\theta_1, \theta_2, \ldots, \theta_n\}$.

SMMP is demonstrated to be stable and independent without iterations or any initial setup, making it an excellent multi-prototype clustering technique.

4.2.2 Comparisons Among the State-of-the-Art Algorithms

Fig. 6 shows the comparison results of different algorithms, where "★" represents the identified cluster centers of K-means and DPC, and "×" represents the identified noise of DBSCAN. As shown, the proposed SMMP almost perfectly distinguished all datasets using multi-prototype clustering technique; SSC distinguished all the ring and spiral shapes of the *Threecircles* and *Spiral* dataset but had flaws in identifying some non-spherical clusters in the *Agg*, *Jain*, and *Flame* datasets; DPC did a satisfying job on the *Agg*, *Flame*, *Spiral*, *S3*, and *A3* datasets, but it failed on the *Jain*, *Threecircles*, and *D1* datasets due to incorrect cluster center recognition; DBSCAN almost successfully reconstructed all shapes, but it falsely identified many border points as noise points in the *S3* dataset and misdetected the number of clusters of the *Jain* and *D1* datasets; K-means failed to identify all non-spherical clusters.

Algorithm 2. SMMP: Border-Link-Based Connectivity Measure of Sub-Clusters

Input: sub-cluster result $\hat{Cl}$, and representativeness θ, and k.
Output: similarity matrix S of sub-clusters $\hat{Cl}$.

// **the identification cross-cluster border points**
$k_b = \lfloor \min(\frac{k}{2}, 2\ln(n)) \rfloor$
for each pair of sub-clusters $\hat{Cl}_y, \hat{Cl}_z \in \hat{Cl}$ **do**
  if $\exists x_i \in \hat{Cl}_y, \exists x_j \in \hat{Cl}_z, x_i \in N_{k_b}(x_j), x_j \in N_{k_b}(x_i)$ **then**
    $x_i \rightleftharpoons x_j$ are cross-cluster border points, and sub-clusters $\hat{Cl}_y$ and $\hat{Cl}_z$ are intersecting. // Definition 3
    $Bor = Bor \cup \{x_i, x_j\}$ // Bor: a set of border points.
  end if
end for
// **the representativeness calculation of border links**
for each border point $x_i \in Bor$ **do**
  $\tau_i = \arg\min_{x_j : x_j \rightleftharpoons x_i}(d_{ij})$, s.t. x_i, x_j are both unlinked // Eq. (6)
  if $\tau(i) \neq \varnothing$ **then**
    $l_i = \{x_i, \tau_i\}$
    $x_i, \tau_i \leftarrow$ linked // give linked points "linked" labels
    $g_{l_i} = \frac{\theta_i + \theta_{\tau_i}}{2}$, $l_i = \{x_i, \tau_i\}$ // g_{l_i} is the representativeness of l_i according to Eq. (9)
  end if
end for
// **the similarity evaluation of sub-clusters**
for each pair of density peaks $\hat{Cl}_y, \hat{Cl}_z \in \hat{Cl}$ **do**
  determine n_g according to Eq. (11)
  obtain $G_{\hat{Cl}_y \hat{Cl}_z} = \{g_{[1]}, g_{[2]}, \ldots, g_{[n_g]}\}$ // Eq. (10)
  get $s(\hat{Cl}_y, \hat{Cl}_z)$ according to Eqs. (13) and (14).
end for
returnthe similarity matrix S.

For further information, a comparison table of AMI, ARI, NMI, and F-Score is presented in Table 2, where the best results are highlighted and the best results of non-parametric algorithms are marked with highlighted*. As shown, the proposed SMMP stands out for its high scores on all experimental datasets; KMM's performance is secondary; SSSP-DPC is merely superior to DPC on the *Threecircles* dataset; while AP loses its competitiveness in identifying non-spherical clusters; MSC as another density-based technique is inferior to DBSCAN.

As verified, our SMMP algorithm has a quite pleasing recognition performance.

4.3 Experiments on Real-World Datasets

The real-world dataset clustering has always been a hard nut to crack for its high-dimensional and large-size characters, but this also indicates its vital importance.

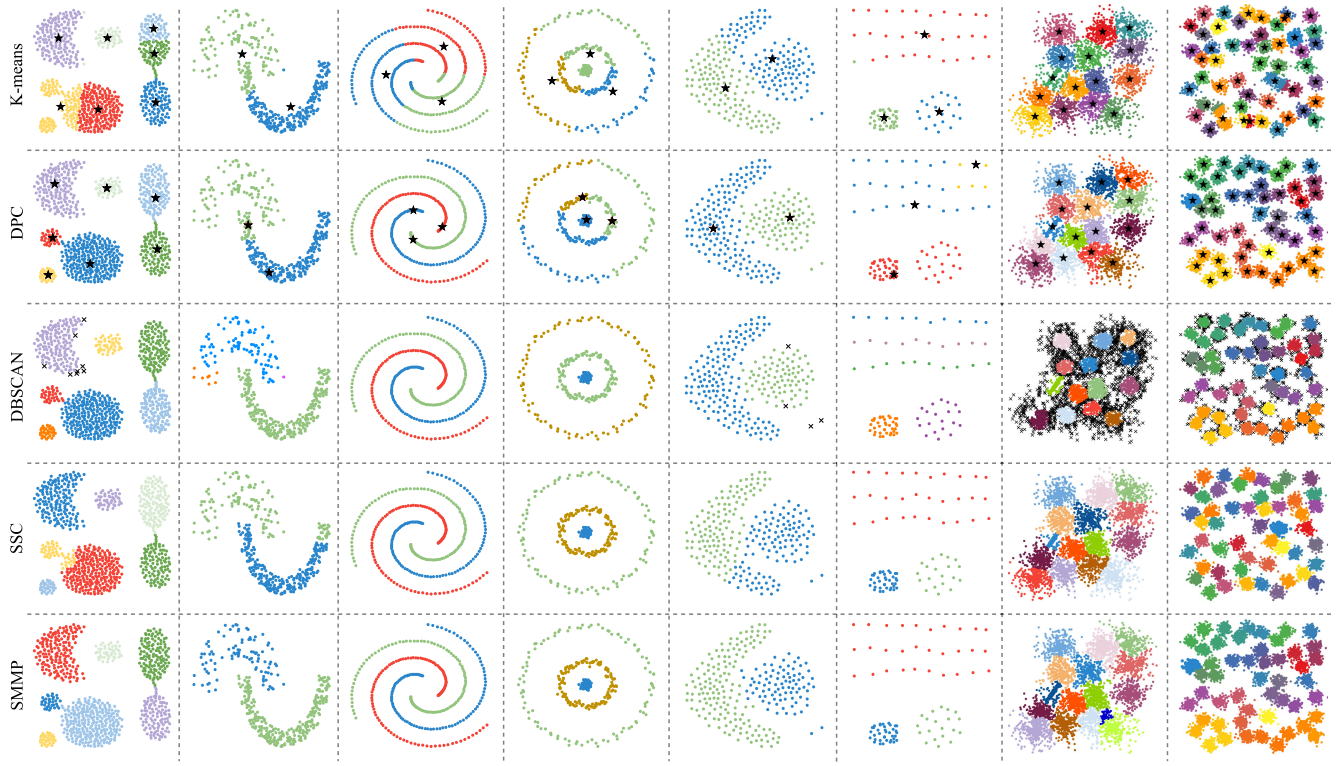


Fig. 6. The results of different algorithms on synthetic datasets. The datasets from left to right are named: *Agg*, *Jain*, *Spiral*, *Threecircles*, *Flame*, *D1*, *S3*, and *A3*.

In this subsection, experiments are conducted on eleven real-world datasets, including eight UCI [44] datasets (*Breast-cancer*, *Movementlibras*, *Parkin*, *Drivedata*, *Waveform*, *Lonosphere*, *vote* and *Musk*) and three popular large-scale machine learning datasets (*YTF* [45], *REUTERS* [46], and *MNIST* [47]) of 10,000 samples. The experimental results are reported in Table 3, where the best results are highlighted and the best results of non-parametric algorithms are marked with highlighted*.

As Table 3 shows, the overall performance of SMMP is outstanding, especially among non-parametric algorithms. SMMP algorithm is demonstrated to be a good alternative method to real-world dataset clustering.

TABLE 2
The Comparison of AMI, ARI, NMI, and F-Score on Synthetic Datasets

Dataset	Metric	DPC	SSSP-DPC	K-means	KMM	SSC	DBSCAN	MSC	AP	SMMP
Agg	AMI ARI	**0.99 1.00**	0.97 0.98	0.82 0.75	**0.99** 0.99	0.96 0.97	0.97 0.98	0.83 0.83	0.61 0.40	**0.99 1.00**
	NMI F-Score	**0.99 1.00**	0.97 0.98	0.85 0.85	**0.99 1.00**	0.97 0.99	0.98 0.99	0.90 0.89	0.76 0.59	**0.99 1.00**
Jain	AMI ARI	0.54 0.62	0.36 0.32	0.49 0.58	**1.00 1.00**	0.64 0.73	0.87 0.98	0.52 0.62	0.22 0.12	**1.00 1.00**
	NMI F-Score	0.58 0.90	0.39 0.80	0.53 0.89	**1.00 1.00**	0.67 0.93	0.93 0.99	0.56 0.90	0.37 0.32	**1.00 1.00**
Spiral	AMI ARI	**1.00 1.00**	**1.00 1.00**	-0.01 -0.01	**1.00 1.00**	**1.00 1.00**	**1.00 1.00**	0.28 0.13	0.24 0.13	**1.00 1.00**
	NMI F-Score	**1.00 1.00**	**1.00 1.00**	-0.00 0.35	**1.00 1.00**	**1.00 1.00**	**1.00 1.00**	0.46 0.37	0.37 0.34	**1.00 1.00**
Threecircles	AMI ARI	0.18 0.03	**1.00 1.00**	0.16 0.06	**1.00 1.00**	**1.00 1.00**	**1.00 1.00**	0.41 0.32	0.37 0.27	**1.00 1.00**
	NMI F-Score	0.23 0.53	**1.00 1.00**	0.17 0.43	**1.00 1.00**	**1.00 1.00**	**1.00 1.00**	0.51 0.53	0.56 0.41	**1.00 1.00**
Flame	AMI ARI	**1.00 1.00**	**1.00 1.00**	0.43 0.48	0.91 0.95	0.54 0.61	0.84 0.94	0.86 0.92	0.23 0.13	**1.00 1.00**
	NMI F-Score	**1.00 1.00**	**1.00 1.00**	0.45 0.85	0.91 0.99	0.55 0.89	0.88 0.98	0.87 0.98	0.38 0.31	**1.00 1.00**
D1	AMI ARI	0.59 0.53	0.59 0.49	0.95 0.96	**1.00 1.00**	**1.00 1.00**	0.73 0.81	0.75 0.84	0.73 0.81	**1.00 1.00**
	NMI F-Score	0.68 0.76	0.67 0.73	0.95 0.99	**1.00 1.00**	**1.00 1.00**	0.85 0.83	0.87 0.90	0.85 0.84	**1.00 1.00**
D2	AMI ARI	**0.96 0.97**	**0.96 0.97**	**0.96 0.97**	**0.96 0.97**	**0.96 0.97**	0.85 0.91	**0.96 0.97**	**0.96 0.97**	**0.96 0.97**
	NMI F-Score	**0.97 0.99**	**0.97 0.99**	**0.97 0.99**	**0.97 0.99**	**0.97 0.99**	0.90 0.97	**0.97 0.99**	**0.97 0.99**	**0.97 0.99**
R15	AMI ARI	**0.99 0.99**	**0.99 0.99**	0.94 0.89	**0.99 0.99**	**0.99 0.99**	0.98 0.98	**0.99 0.99**	**0.99 0.99**	**0.99 0.99**
	NMI F-Score	**0.99 1.00**	**0.99 0.99**	0.95 0.92	**0.99 1.00**	**0.99 1.00**	**0.99 0.99**	**0.99 1.00**	**0.99 1.00**	**0.99 1.00**
S3	AMI ARI	0.94 0.93	0.88 0.83	0.90 0.87	0.92 0.90	0.90 0.86	0.66 0.30	0.88 0.85	0.47 0.32	**0.95 0.94**
	NMI F-Score	0.94 0.96	0.88 0.91	0.90 0.94	0.92 0.95	0.90 0.93	0.70 0.79	0.88 0.92	0.67 0.49	**0.95 0.97**
D31	AMI ARI	0.95 0.93	0.96 0.94	0.91 0.82	0.96 0.95	**0.97 0.95**	0.87 0.71	0.95 0.92	0.77 0.80	**0.96* 0.94***
	NMI F-Score	0.96 0.97	0.96 0.97	0.92 0.86	0.96 0.97	**0.97 0.98**	0.88 0.93	0.95 0.96	0.87 0.79	**0.96* 0.97***
A3	AMI ARI	**0.99** 0.98	0.98 0.97	0.98 0.93	**0.99** 0.98	**0.99 0.99**	0.90 0.74	0.98 0.96	0.34 0.32	**0.99 0.98***
	NMI F-Score	**0.99 0.99**	0.98 **0.99**	0.98 0.95	**0.99 0.99**	**0.99 0.99**	0.91 0.95	0.98 0.98	0.70 0.32	**0.99 0.99**
DIM1024	AMI ARI	**1.00 1.00**	**1.00 1.00**	**1.00 1.00**	**1.00 1.00**	**1.00 1.00**	**1.00 1.00**	**1.00 1.00**	0.39 0.65	**1.00 1.00**
	NMI F-Score	**1.00 1.00**	**1.00 1.00**	**1.00 1.00**	**1.00 1.00**	**1.00 1.00**	**1.00 1.00**	**1.00 1.00**	0.73 0.53	**1.00 1.00**

TABLE 3
The Comparison of AMI, ARI, NMI, and F-Score on Real-World Datasets

Dataset	Metric	DPC	SSSP-DPC	K-means	KMM	SSC	DBSCAN	MSC	AP	SMMP
Breastcancer	AMI ARI	0.41 0.47	0.34 0.38	0.61 0.73	0.59 0.70	**0.69 0.80**	0.38 **0.50***	0.21 0.42	0.15 0.06	**0.42*** 0.49
	NMI F-Score	0.46 0.84	0.39 0.79	0.62 0.93	0.61 0.92	**0.70 0.95**	0.38 **0.85***	0.34 0.78	0.27 0.25	**0.48*** 0.84
Movementlibras	AMI ARI	0.48 0.26	0.58 0.31	0.55 0.32	0.50 0.30	**0.58 0.36**	0.27 0.09	**0.47** 0.29	**0.49* 0.39***	0.43 0.28
	NMI F-Score	**0.58** 0.46	0.60 0.51	0.61 0.51	0.59 0.50	0.63 **0.53**	0.41 0.38	**0.65** 0.48	0.64 **0.48***	0.54 0.46
Parkin	AMI ARI	0.23 0.09	0.23 0.09	0.22 0.05	0.05 -0.03	0.19 0.15	0.11 **0.18**	0.07 0.03	0.09 0.03	**0.25** 0.13
	NMI F-Score	0.25 0.68	0.25 0.68	0.24 0.65	0.02 **0.74**	0.21 0.72	0.13 **0.70***	0.16 0.46	0.19 0.27	**0.28 0.70***
Drivedata	AMI ARI	0.60 0.61	**0.65 0.68**	0.52 0.50	0.53 0.51	0.47 0.44	0.53 0.56	-0.00 0.00	0.31 0.14	**0.56* 0.59***
	NMI F-Score	0.61 0.83	**0.65 0.86**	0.53 0.73	0.54 0.74	0.48 0.71	0.61 0.77	0.35 0.01	0.49 0.28	**0.62* 0.78***
Waveform	AMI ARI	0.22 0.19	0.21 0.20	0.36 0.25	0.37 0.25	0.37 0.25	0.01 0.00	0.36 0.25	0.14 0.02	**0.39 0.31**
	NMI F-Score	0.22 0.56	0.21 0.58	0.36 0.53	0.37 0.53	0.37 0.52	0.01 0.48	0.36 0.53	0.23 0.07	**0.39 0.60**
Lonosphere	AMI ARI	0.07 0.03	0.05 -0.04	0.12 0.17	0.12 0.18	0.11 0.14	**0.61 0.72**	0.11 0.29	0.11 0.09	0.20 0.25
	NMI F-Score	0.09 0.62	0.07 0.65	0.13 0.71	0.13 0.72	0.12 0.70	**0.64 0.92**	0.28 0.57	0.22 0.45	0.26 0.74
Vote	AMI ARI	0.50 0.56	0.50 0.56	0.46 0.54	**0.53 0.60**	0.50 0.59	0.31 0.30	0.34 **0.54***	0.12 0.04	**0.36*** 0.40
	NMI F-Score	0.51 0.88	0.51 0.88	0.47 0.87	**0.54 0.89**	0.50 0.88	0.38 0.69	**0.42* 0.87***	0.23 0.18	0.38 0.82
Musk	AMI ARI	-0.00 0.00	0.03 -0.04	0.06 **0.15**	0.03 -0.04	0.03 -0.03	0.07 **0.05***	0.03 -0.04	0.04 0.00	**0.13** 0.02
	NMI F-Score	-0.00 0.59	0.03 0.64	0.06 **0.77**	0.03 0.64	0.03 0.63	0.06 0.56	0.03 **0.59***	**0.09** 0.04	**0.09** 0.33
YTF	AMI ARI	0.73 0.50	**0.80** 0.58	0.76 0.57	0.75 0.39	0.71 0.43	0.67 0.38	**0.78* 0.68**	0.53 0.24	0.72 0.53
	NMI F-Score	0.76 0.59	0.81 0.68	0.77 0.62	0.80 0.67	0.77 0.61	0.77 0.60	**0.87 0.71**	0.75 0.37	0.76 0.61
REUTERS	AMI ARI	0.27 0.28	0.25 0.22	0.51 0.57	0.12 0.01	**0.50 0.36**	0.31 0.09	0.31 0.16	0.18 0.01	**0.36* 0.34***
	NMI F-Score	0.26 0.60	0.24 0.58	0.51 0.76	0.09 0.44	**0.48 0.67**	0.28 0.43	0.31 0.48	0.30 0.05	**0.36* 0.59***
MNIST	AMI ARI	0.43 0.30	0.72 0.53	0.84 0.78	0.89 0.85	0.89 0.83	0.56 0.24	0.58 0.40	0.35 0.05	**0.92 0.93**
	NMI F-Score	0.49 0.49	0.81 0.72	0.85 0.86	0.91 0.91	0.90 0.88	0.55 0.54	0.68 0.63	0.54 0.09	**0.93 0.97**

4.4 Comparison of Cluster Number Detection

Table 4 displays the comparison of cluster number detection performance of the non-parametric clustering algorithms: DBSCAN, MSC, AP, and SMMP.

As shown in Table 4 and Fig. 6, for synthetic datasets, SMMP perfectly detected the number of clusters, except for dividing *S3* into sixteen clusters; DBSCAN mistakenly divided the sparse moon-shaped cluster of the *Jain* dataset into three clusters, the sparse cluster of the *D1* dataset into three clusters, and the *S3* into fourteen clusters; AP over-divided the *S3* dataset into 1469 clusters; MSC wrongly divided the *Threecircles* dataset into fifteen clusters and the *Agg* dataset into merely five clusters.

TABLE 4
The Comparison of Cluster Number Detection

Dataset	DBSCAN	MSC	AP	SMMP
Agg ($C = 7$)	7	6	16	7
Jain ($C = 2$)	4	2	14	2
Spiral ($C = 3$)	3	34	18	3
Threecircles ($C = 3$)	3	11	18	3
Flame ($C = 2$)	2	2	13	2
D1 ($C = 3$)	5	5	5	3
D2 ($C = 4$)	4	4	4	4
R15 ($C = 15$)	15	15	15	15
S3 ($C = 15$)	15	15	654	16
D31 ($C = 31$)	31	32	300	31
A3 ($C = 50$)	50	50	3548	50
DIM1024 ($C = 16$)	16	16	512	16
Breastcancer ($C = 2$)	1	65	43	2
Movementlibras ($C = 15$)	9	58	31	7
Parkin ($C = 2$)	2	30	21	2
Drivedata ($C = 4$)	9	606	42	6
Waveform ($C = 3$)	12	4	147	4
Lonosphere ($C = 2$)	1	103	44	4
Vote ($C = 2$)	2	20	38	2
Musk ($C = 2$)	5	3	445	19
YTF ($C = 41$)	195	129	1238	29
REUTERS ($C = 4$)	353	76	454	13

Algorithm 3. SMMP: The Adaptive Merging of Sub-Clusters

Input: the similarity matrix S and sub-clusters $\hat{Cl}$.
Output: Clustering result Cl

```
// the dendrogram building
obtain a dendrogram by apply the Single-linkage clustering technique with the similarity matrix w.r.t s as input.
// the auto-tuning of clustering threshold t
I_t = [min(s(Ĉl_y, Ĉl_z)), max(s(Ĉl_y, Ĉl_z))] ⊆ [0, 1]
gap = 0.01 // set iterate times (gap is adjustable).
for t = min(I_t) : gap : max(I_t) do
    detect cluster number C^t w.r.t t according to the dendrogram.
end for
solve the Problem (3) to obtain the optimal clustering threshold t = mean(I*_t)
obtain corresponding clustering result Cl^t w.r.t t
Cl ← Cl^t and C ← C^t.
returnClustering result Cl = {Cl_1, Cl_2, ..., Cl_C}.
```

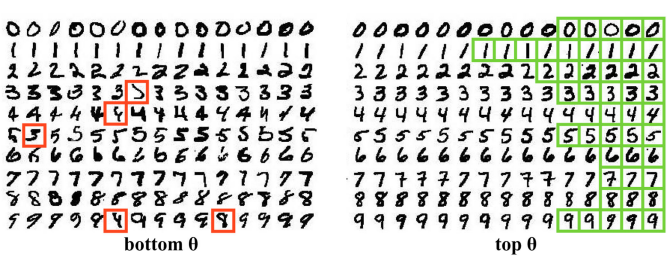


Fig. 7. Fifteen bottom θ digit images (a) and fifteen top θ digit images (b) in each clusters of the *MNIST* dataset.

TABLE 5
The Runtime Comparison of Different Algorithms (Unit: Second)

Dataset	DPC	SSSP-DPC	K-means	KMM	SSC	DBSCAN	MSC	AP	SMMP
Agg ($n = 788$)	0.112	0.320	0.009	2.653	0.580	0.013	0.008	1.470	0.027
Jain ($n = 373$)	0.032	0.107	0.004	1.146	0.619	0.005	0.007	0.263	0.011
Spiral ($n = 312$)	0.009	0.201	0.005	0.304	0.248	0.005	0.005	0.175	0.012
Threecircles ($n = 299$)	0.024	0.117	0.007	1.018	0.159	0.005	0.005	0.188	0.011
Flame ($n = 240$)	0.041	0.051	0.003	1.335	0.118	0.003	0.006	0.149	0.008
D1 ($n = 87$)	0.001	0.021	0.007	0.178	0.062	0.002	0.002	0.014	0.004
D2 ($n = 85$)	0.001	0.025	0.006	0.270	0.069	0.001	0.001	0.011	0.009
R15 ($n = 600$)	0.046	0.272	0.007	1.538	0.328	0.025	0.002	1.607	0.028
S3 ($n = 5000$)	3.150	12.492	0.069	56.610	7.592	0.457	0.124	316.723	0.132
D31 ($n = 3100$)	0.855	4.727	0.051	25.061	2.714	0.142	0.020	69.094	0.064
A3 ($n = 7500$)	5.511	32.424	0.054	110.314	8.802	0.759	0.083	506.514	0.184
DIM1024 ($n = 1024$)	0.493	0.804	0.063	0.517	1.298	0.068	0.142	22.450	0.024
Breastcancer ($n = 569$)	0.073	0.182	0.053	3.865	0.425	0.012	0.074	0.612	0.031
Movementlibras ($n = 360$)	0.023	0.082	0.021	0.996	0.257	0.013	0.032	0.374	0.018
Parkin ($n = 195$)	0.005	0.005	0.003	0.503	0.118	0.001	0.014	0.072	0.007
Drivedata ($n = 606$)	0.662	0.493	0.211	2.142	1.694	0.127	2.396	13.664	0.019
Waveform ($n = 5000$)	2.484	12.798	0.022	43.199	5.403	0.432	2.942	55.551	0.509
Lonosphere ($n = 351$)	0.043	0.071	0.003	1.346	0.138	0.016	0.022	0.216	0.023
Vote ($n = 345$)	0.049	0.189	0.005	23.685	0.265	0.004	0.012	0.573	0.019
Musk ($n = 6598$)	6.882	29.374	0.086	48.563	15.807	1.266	0.535	482.611	0.407
YTF ($n = 10000$)	11.681	83.328	0.122	149.977	28.625	4.561	0.394	2488.182	1.030
REUTERS ($n = 10000$)	11.316	86.601	0.052	72.533	29.294	3.138	1.975	3635.583	1.531
Total time	43.493	264.684	0.863	547.753	104.615	11.235	8.801	7597.096	4.108

TABLE 6
The Time Complexity of Algorithms

DPC [19]	$O(n^2)$	SSSP-DPC [31]	$O(n^2)$
K-means [9]	$O(nCT)$	SSC [42]	$O(n^2)$
DBSCAN [12]	$O(n\log(n))$	MSC [28]	$O(n^2)$
AP [11]	$O(n^2T)$	SMMP (ours)	$O(n(\log(n) + \tilde{k} + k_b) + \hat{C}^2)$
KMM [20]	$O(n((\hat{C}d + \hat{C}C + \hat{C}\hat{C})T_1 + \hat{C}d)T)$		

T and T_1 indicate iteration times.

For real-world datasets, SMMP also has a better cluster detection performance compared with DBSCAN, MSC, and AP, especially for the *YTF* and *REUTERS* datasets. Also, unlike non-parametric clustering techniques such as DBSCAN and MSC which highly rely on laborious manual parameter tuning, SMMP is an auto-tuning method.

4.5 The Handwritten Digit Recognition of *MNIST*

In handwritten digit recognition applications, a class is often composed of multiple subclasses, because different users write the same digits in different ways [1]. So, a handwritten digit class can be modeled as a multi-prototype cluster. A strong feature representation *MNIST* [47] (a well-known handwritten digit image dataset) test set of 10,000

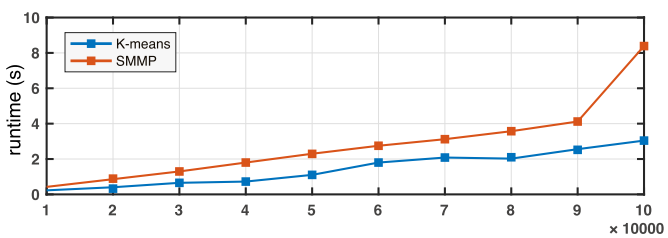


Fig. 8. The speed comparison between K-means and SMMP on ten different size sampling datasets of the *Birchrg1* dataset.

TABLE 7
The Ranges of AMI, ARI, NMI, and F-Score at $k \in I_k^*$

Dataset	AMI	ARI	NMI	F-Score
Agg	99.2(±**0**)	99.6(±**0**)	99.2(±**0**)	99.8(±**0**)
Jain	100.0(±**0**)	100.0(±**0**)	100.0(±**0**)	100.0(±**0**)
Spiral	96.6(±3.4)	97.1(±2.9)	96.6(±3.4)	99.0(±1.0)
Threecircles	100.0(±**0**)	100.0(±**0**)	100.0(±**0**)	100.0(±**0**)
Flame	100.0(±**0**)	100.0(±**0**)	100.0(±**0**)	100.0(±**0**)
D1	100.0(±**0**)	100.0(±**0**)	100.0(±**0**)	100.0(±**0**)
D2	96.4(±**0**)	96.8(±**0**)	96.6(±**0**)	98.8(±**0**)
R15	99.4(±**0**)	99.3(±**0**)	99.4(±**0**)	99.7(±**0**)
S3	95.4(±0.9)	94.9(±1.2)	96.0(±0.8)	97.3(±0.6)
D31	95.7(±0.2)	93.8(±0.3)	95.9(±0.1)	96.9(±0.1)
A3	98.8(±0.1)	98.2(±0.1)	98.8(±0.1)	99.1(±0.1)
DIM1024	100.0(±**0**)	100.0(±**0**)	100.0(±**0**)	100.0(±**0**)
Breastcancer	43.9(±0.7)	49.9(±0.8)	48.4(±0.6)	84.7(±0.3)
Movementlibras	42.4(±0.9)	27.6(±1.3)	54.3(±0.6)	45.7(±0.8)
Parkin	22.4(±2.4)	7.6(±5.0)	24.8(±2.8)	66.5(±3.7)
Drivedata	55.5(±**0**)	59.1(±**0**)	62.1(±**0**)	77.6(±**0**)
Waveform	39.4(±0.2)	31.1(±0.2)	39.3(±0.2)	59.9(±0.2)
Lonosphere	20.2(±0.5)	24.8(±0.5)	25.8(±0.5)	73.8(±0.4)
Vote	38.9(±2.3)	43.6(±3.1)	40.5(±2.4)	83.5(±1.1)
Musk	13.6(±0.4)	1.9(±0.1)	9.3(±0.3)	29.3(±**0**)
YTF	72.8(±0.0)	53.3(±0.1)	76.1(±0.0)	60.7(±0.0)
REUTERS	36.4(±**0**)	34.2(±**0**)	35.6(±**0**)	58.7(±**0**)
MNIST	92.5(±0.3)	92.7(±0.5)	92.5(±0.3)	96.6(±0.3)

TABLE 8
The Average AMI, ARI, NMI, and F-Score of All Datasets at $\eta \in [0, 1]$

Paramater η	0	0.05	0.10	0.15	0.20	0.25	0.30	0.35	0.40	0.45	0.50	0.60	0.70	0.80	0.90	1.00
AMI_arv(%)	60.46	71.38	**72.22**	72.14	72.14	71.79	71.62	71.62	71.62	69.19	69.19	68.84	65.64	64.29	62.46	62.46
ARI_arv(%)	55.65	69.16	**69.97**	69.79	69.79	69.30	69.24	69.24	69.24	66.52	66.52	66.16	62.60	60.31	58.87	58.87
NMI_arv(%)	65.10	73.17	**73.62**	73.58	73.58	73.42	73.36	73.36	73.36	71.69	71.69	71.46	68.95	68.10	66.94	66.94
F-Score_arv (%)	75.76	83.45	**84.10**	83.97	83.97	83.60	83.52	83.52	83.52	81.80	81.80	81.36	79.18	77.85	77.44	77.44

samples with 500 features from [14] is used to evaluate the performance of our proposed algorithm.

To obtain a high recognition accuracy, we define a noise threshold θ_c based on representativeness, i.e., points with $\theta < \theta_c$ are considered as noise. By applying θ_c to cut 16% of data as noise, we obtain an almost perfect clustering result with $AMI = 0.97, ARI = 0.98$. Fig. 7 shows some recognition results of the digit images of different θ values. As shown, SMMP provided an almost perfect recognition result with just five slips (red). Although digit images with low θ are often difficult to recognize, SMMP did a satisfying job. In addition, for each class, SMMP identified multiple subclasses (with centers marked by green) that represent different writing ways, which effectively reflects the real underlying structure of data.

4.6 The Speed of SMMP

In large-scale data clustering tasks, the execution speed is one of the most important factors that need special attention. As Table 5 shows, SMMP (the second-fastest clustering algorithm) only takes about one second to execute a dataset of 10,000 data points; while AP, KMM, and SSSP-DPC are prohibitively time-consuming. Also, SMMP is much faster than KMM, although the two are both multi-prototype clustering techniques. Table 6 lists the time complexity of all comparison algorithms.

To further verify the fast speed of SMMP, speed comparisons of K-means and SMMP are launched on the *Birchrg1* dataset of 100,000 points, as in Fig. 8. As shown, SMMP took about eight seconds to execute a dataset of 100,000 points, which is slower than K-means but still acceptable.

As verified, SMMP with fast speed is promising for large-scale data clustering.

4.7 Parameter Sensitivity

SMMP has three auto-tuned parameters: $k = \lceil \text{mean}(I_k^*) \rceil$, $k_b = \lfloor \min(\frac{k}{2}, 2\ln(n)) \rfloor$, $t = \text{mean}(I_t^*)$, and a fixed parameter $\eta = 0.1$, and the effectiveness of the auto-tuning has already been verified in Sections 4.2 and 4.3. In fact, when k, k_b, and η are fixed, similarity matrix S is obtained. Then, on the basis of S, we can obtain I_t^* by solving Problem (3). Because $\forall t \in I_t^*$ returns the same clustering result, SMMP is insensitivity to $t = \text{mean}(I_t^*)$. So, we only need to verify the sensitivity of $k = \lceil \text{mean}(I_k^*) \rceil$ and η, because different $k \in I_k^*$ and $\eta \in [0, 1]$ may return different clustering results.

Table 7 shows the ranges of AMI, ARI, NMI, and F-Score at $k \in I_k^*$ on different datasets. As shown, the clustering performance of SMMP at $k \in I_k^*$ is efficient and robust. Table 8 shows the average AMI, ARI, NMI, and F-Score (AMI_arv, ARI_arv, NMI_arv, and F-Score_arv) over all datasets with different $\eta \in [0, 1]$ (the best results are highlighted). As shown, SMMP had a stable performance at $\eta \in [0, 0.5]$, and obtained the best performance around $\eta = 0.1$. So, we set $\eta = 0.1$ as default. As verified, SMMP is insensitive to parameters k and η, and the setting of $\eta = 0.1$ is efficient.

5 Conclusion

Herein, a stable-membership-based parameter-free multi-peak clustering algorithm (SMMP) is proposed. As a multi-prototype non-parametric clustering technique, SMMP can achieve fast, automatic, and accurate multi-prototype clustering without iteration. Our designed DAT method can help learn the representativeness of points to sub-cluster centers during the generation of sub-clusters. Benefited from the superiority of DPC technology in recognizing shapes, the generated sub-clusters can have arbitrary shapes, which allows the algorithm to preserve sufficient local structure information. Also, our proposed border-link-based connectivity measure method can help to obtain a high-fidelity similarity evaluation of sub-clusters based on the learned representativeness. According to the similarity matrix, high-similar arbitrary-shaped sub-clusters are successfully spliced into complex-shaped clusters by applying the Single-linkage method. Besides, the introduced concept of the "stable membership", as a core criterion for a reasonable clustering state, allows SMMP to achieve auto-tuning. As analyzed, SMMP is proven to be suitable for large datasets, requiring only kNN distances of data. The effectiveness and the efficiency of SMMP are well-verified in the conducted comparisons on synthetic datasets and real-world datasets, as well as its application to the handwritten digit recognition of *MNIST*.

Nevertheless, as a non-parametric clustering technique, SMMP is more suitable for low-dimensional data clustering. Because our border detection method and similarity estimation of sub-clusters are more suitable for low-dimensional data. We believe these methods can be further refined to better suit datasets with high dimensions. Besides, we will seek some dimensionality reduction techniques to transform high-dimensional datasets into low-dimensional ones to expand the application of SMMP.

References

[1] A. K. Jain, "Data clustering: 50 years beyond k-means," *Pattern Recognit. Lett.*, vol. 31, no. 8, pp. 651–666, 2010.
[2] S. Shalev-Shwartz and S. Ben-David, *Understanding Machine Learning: From Theory to Algorithms*, Cambridge, U.K.: Cambridge Univ. Press, 2014.
[3] A. K. Jain and R. C. Dubes, *Algorithms for Clustering Data*. Hoboken, NJ, USA: Prentice-Hall, 1988.

[4] J. Li, K. Wang, and L. Xu, "Chameleon based on clustering feature tree and its application in customer segmentation," *Ann. Operations Res.*, vol. 168, no. 1, pp. 225–245, 2009.

[5] R. Achanta and S. Susstrunk, "Superpixels and polygons using simple non-iterative clustering," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 4651–4660.

[6] 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.

[7] T. Lei, X. Jia, X. Zhang, and H. Meng, "Automatic fuzzy clustering framework for image segmentation," *IEEE Trans. Fuzzy Syst.*, vol. 28, no. 9, pp. 2078–2092, Sep. 2020.

[8] M. I. Jordan and T. M. Mitchell, "Machine learning: Trends, perspectives, and prospects," *Science*, vol. 349, no. 6245, pp. 255–260, 2015.

[9] J. MacQuee, "Some methods for classification and analysis of multivariate observations," in *Proc. 5th Berkeley Symp. Math. Statist. Probability*, vol. 1, no. 14, pp. 281–297, 1967.

[10] L. Kaufman and P. J. Rousseeuw, *Finding Groups in Data: An Introduction to Cluster Analysis*, New York, NY, USA: Wiley, 2009.

[11] B. J. Frey and D. Dueck, "Clustering by passing messages between data points," *Science*, vol. 315, no. 5814, pp. 972–976, 2007.

[12] M. Ester et al., "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.

[13] 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.

[14] 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.

[15] S. Mai et al., "Incremental density-based clustering on multicore processors," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 44, no. 3, pp. 1338–1356, Mar. 2022.

[16] R. Sibson, "SLINK: An optimally efficient algorithm for the single-link cluster method," *Comput. J.*, vol. 16, no. 1, pp. 30–34, 1973.

[17] D. Defays, "An efficient algorithm for a complete link method," *Comput. J.*, vol. 20, no. 4, pp. 364–366, 1977.

[18] H. K. Seifoddini, "Single linkage versus average linkage clustering in machine cells formation applications," *Comput. Ind. Eng.*, vol. 16, no. 3, pp. 419–426, 1989.

[19] A. Rodriguez and A. Laio, "Clustering by fast search and find of density peaks," *Science*, vol. 344, no. 6191, pp. 1492–1496, 2014.

[20] F. Nie, C.-L. Wang, and X. Li, "K-multiple-means: A multiple-means clustering method with specified k clusters," in *Proc. 25th ACM SIGKDD Int. Conf. Knowl. Discov. Data Mining*, 2019, pp. 959–967.

[21] C.-D. Wang, J.-H. Lai, C. Y. Suen, and J.-Y. Zhu, "Multi-exemplar affinity propagation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 9, pp. 2223–2237, Sep. 2013.

[22] M. Liu, X. Jiang, and A. C. Kot, "A multi-prototype clustering algorithm," *Pattern Recognit.*, vol. 42, no. 5, pp. 689–698, 2009.

[23] T. Zhang, R. Ramakrishnan, and M. Livny, "BIRCH: An efficient data clustering method for very large databases," *ACM SIGMOD Rec.*, vol. 25, no. 2, pp. 103–114, 1996.

[24] C.-R. Lin and M.-S. Chen, "Combining partitional and hierarchical algorithms for robust and efficient data clustering with cohesion self-merging," *IEEE Trans. Knowl. Data Eng.*, vol. 17, no. 2, pp. 145–159, Feb. 2005.

[25] S. Guha, R. Rastogi, and K. Shim, "CURE: An efficient clustering algorithm for large databases," *ACM SIGMOD Rec.*, vol. 27, no. 2, 1998, Art. no. 73.

[26] T. Luo, C. Zhong, H. Li, and X. Sun, "A multi-prototype clustering algorithm based on minimum spanning tree," in *Proc. 7th Int. Conf. Fuzzy Syst. Knowl. Discov.*, 2010, pp. 1602–1607.

[27] N. Bhatia et al., "Survey of nearest neighbor techniques," 2010, *arXiv:1007.0085*.

[28] D. Comaniciu and P. Meer, "Mean shift: A robust approach toward feature space analysis," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 24, no. 5, pp. 603–619, May 2002.

[29] D. Cheng, J. Huang, S. Zhang, X. Zhang, and X. Luo, "A novel approximate spectral clustering algorithm with dense cores and density peaks," *IEEE Trans. Syst., Man, Cybern. Syst.*, vol. 52, no. 4, pp. 2348–2360, Apr. 2022.

[30] A. Lotfi, P. Moradi, and H. Beigy, "Density peaks clustering based on density backbone and fuzzy neighborhood," *Pattern Recognit.*, vol. 107, 2020, Art. no. 107449.

[31] D. U. Pizzagalli et al., "A trainable clustering algorithm based on shortest paths from density peaks," *Sci. Adv.*, vol. 5, no. 10, 2019, Art. no. eaax3770.

[32] 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.

[33] M. Parmar, D. Wang, A. -H. Tan, C. Miao, J. Jiang, and Y. Zhou, "A novel density peak clustering algorithm based on squared residual error," in *Proc. Int. Conf. Secur. Pattern Anal. Cybern.*, 2017, pp. 43–48.

[34] Z. Li and Y. Tang, "Comparative density peaks clustering," *Expert Syst. Appl.*, vol. 95, pp. 236–247, 2018.

[35] M. Parmar et al., "FREDPC: A feasible residual error-based density peak clustering algorithm with the fragment merging strategy," *IEEE Access*, vol. 7, pp. 89789–89804, 2019.

[36] Y. Wang et al., "McDPC: Multi-center density peak clustering," *Neural Comput. Appl.*, vol. 32, no. 17, pp. 13465–13478, 2020.

[37] X. Xu et al., "A robust density peaks clustering algorithm with density-sensitive similarity," *Knowl.-Based Syst.*, vol. 200, 2020, Art. no. 106028.

[38] M. Parmar et al., "REDPC: A residual error-based density peak clustering algorithm," *Neurocomputing*, vol. 348, pp. 82–96, 2019.

[39] J. Kleinberg, "An impossibility theorem for clustering," in *Proc. Adv. Neural Inf. Process. Syst.*, 2002, pp. 463–470.

[40] K. M. Ting, Y. Zhu, M. Carman, Y. Zhu, and Z.-H. Zhou, "Overcoming key weaknesses of distance-based neighbourhood methods using a data dependent dissimilarity measure," *Proc. 22nd ACM SIGKDD Int. Conf. Knowl. Discov. Data Mining*, 2016, pp. 1205–1214.

[41] Pasi Fränti and Sami 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/

[42] L. Zelnik-manor and P. Perona, "Self-tuning spectral clustering," in *Proc. Int. Conf. Neural Inf. Process. Syst.*, 2004, pp. 1601–1608.

[43] P. Franti and O. Virmajoki, "Iterative shrinking method for clustering problems," *Pattern Recognit.*, vol. 39, no. 5, pp. 761–775, 2006.

[44] K. Bache and M. Lichman, "UCI machine learning repository," 2013. [Online]. Available: http://archive.ics.uci.edu/ml

[45] L. Wolf et al., "Face recognition in unconstrained videos with matched background similarity," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2011, pp. 529–534.

[46] D. D. Lewis et al., "RCV1: A new benchmark collection for text categorization research," *J. Mach. Learn. Res.*, vol. 5, pp. 361–397, 2004.

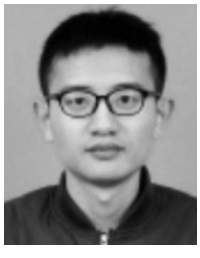
[47] Y. LeCun and C. Cortes, "MNIST handwritten digit database," 2010. [Online]. Available: http://yann.lecun.com/exdb/mnist/

[48] N. X. Vinh, J. Epps, 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.

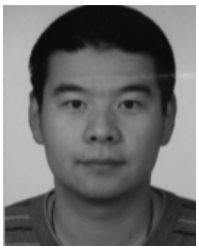
[49] P. A. Estévez, M. Tesmer, C. A. Perez, and J. M. Zurada, "Normalized mutual information feature selection," *IEEE Trans. Neural Netw.*, vol. 20, no. 2, pp. 189–201, Feb. 2009.

[50] M. Sokolova, N. Japkowicz, and S. Szpakowicz, "Beyond accuracy, f-score and ROC: A family of discriminant measures for performance evaluation," in *Proc. Australas. Joint Conf. Artif. Intell.*, 2006, pp. 1015–1021.

[51] Y.-A. Geng et al., "RECOME: A new density-based clustering algorithm using relative KNN kernel density," *Inf. Sci.*, vol. 436, pp. 13–30, 2018.

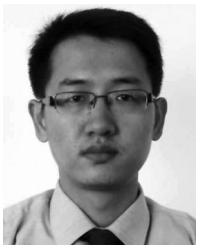


Junyi Guan received the PhD degree from the Zhejiang University of Technology (ZJUT), Hangzhou, China. He is currently a post-doctoral in ZJUT. His current research interests include data mining, pattern recognition, unsupervised learning, and machine learning.



Sheng Li received 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.

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.



Jinhui Zhu received the PhD degree in surgery from Zhejiang Chinese Medical University, Hangzhou, China. He is chief physician of Second Affiliated Hospital, Zhejiang University School of Medicine. His research interests include bioinformatics and pattern recognition.

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

Peng Si is currently working toward the master's degree with the Zhejiang University of Technology. His current research interests include pattern recognition, and medical image processing.