8-2020

A systematic density-based clustering method using anchor points

Yizhang WANG
Jilin University

Di WANG
Nanyang Technological University

Wei PANG
Heriot-Watt University

Ah-hwee TAN
Singapore Management University, ahtan@smu.edu.sg

You ZHOU
Jilin University

A systematic density-based clustering method using anchor points

Yizhang Wang[a, b], Di Wang[c, d], Wei Pang[e], Chunyan Miao[c, d, f], Ah-Hwee Tan [c, f], You Zhou[a , b],

*a College of Computer Science and Technology, Jilin University, Changchun, China
b Key Laboratory of Symbolic Computation and Knowledge Engineering of Ministry of Education, China
c Joint NTU-UBC Research Centre of Excellence in Active Living for the Elderly, Nanyang Technological University, Singapore
d Joint NTU-WeBank Research Centre on FinTech, Nanyang Technological University, Singapore
e School of Mathematical and Computer Sciences, Heriot-Watt University, Edinburgh, UK
f School of Computer Science and Engineering, Nanyang Technological University, Singapore

Published in Neurocomputing, 400 (2020), 352–370

DOI: 10.1016/j.neucom.2020.02.119

Abstract:

Clustering is an important unsupervised learning method in machine learning and data mining. Many existing clustering methods may still face the challenge in self-identifying clusters with varying shapes, sizes and densities. To devise a more generic clustering method that considers all the aforementioned properties of the natural clusters, we propose a novel clustering algorithm named Anchor Points based Clustering (APC). The anchor points in APC are characterized by having a relatively large distance from data points with higher densities. We take anchor points as centers to obtain intermediate clusters, which can divide the whole dataset more appropriately so as to better facilitate further grouping. In essence, based on the analysis of the identified anchor points, the relationship among the corresponding intermediate clusters can be better revealed. In short, the difference in local densities (densities within neighboring data points) of the anchor points characterizes their different properties, that is to say, all the intermediate clusters may fall into one or multiple identified levels with different densities. Finally, based on the properties of anchor points, APC spontaneously chooses the appropriate clustering strategies and reports the final clustering results. To evaluate the performances of APC, we conduct experiments on twelve two-dimensional synthetic datasets and twelve multi-dimensional real-world datasets. Moreover, we also apply APC to the Olivetti Face dataset to further assess its effectiveness in terms of face recognition. All experimental results indicate that APC outperforms four classical methods and two state-of-the-art methods in most cases.

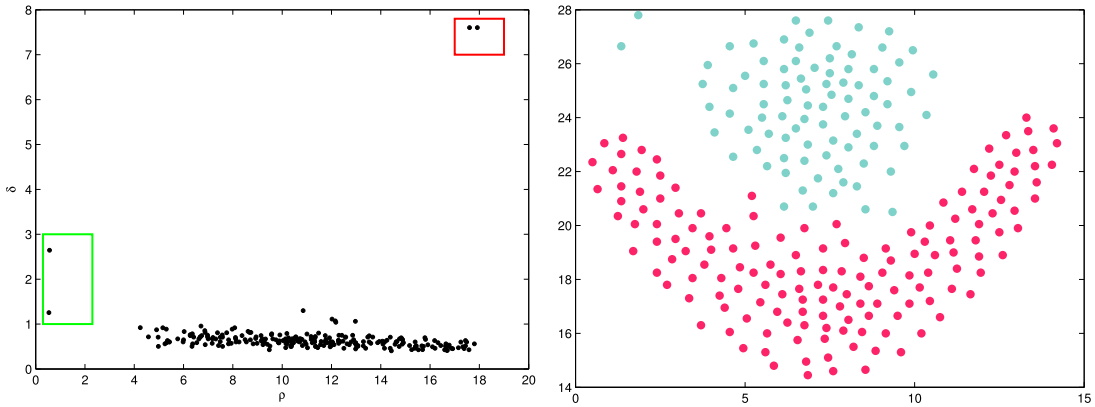
Keywords: Density based clustering, Anchor data points, Local density analysis

1. Introduction

Clustering is an important and effective way of information acquisition without labels. It has been successfully applied in many fields, such as community detection [1], [2], [3], behavioral pattern analysis [4], biological information processing [5], [6], image processing [7], [8], financial data analysis [9], [10], [11], etc. More and more applications incorporate clustering algorithms to better deal with datasets with arbitrary shapes, sizes and densities.

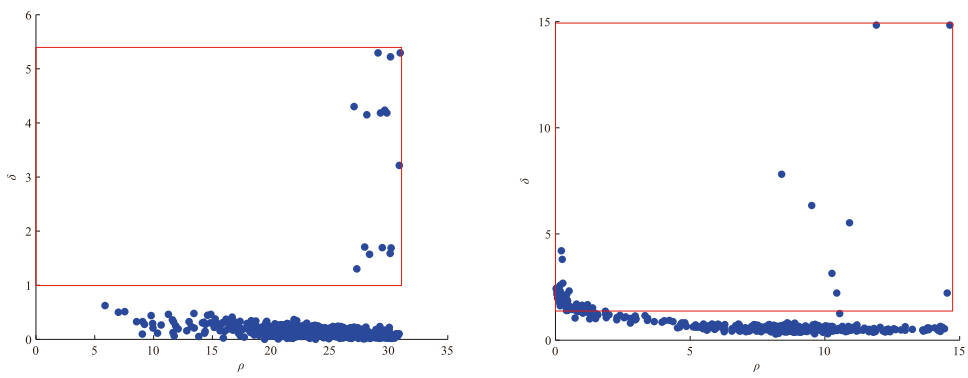
In fact, although many clustering algorithms obtain effective results in various applications, they may not well identify clusters with varying shapes, sizes and densities. K-means can easily identify convex shape clusters [12], but it may fail in identifying non-convex ones. DBSCAN (acronym of Density Based Spatial Clustering of Applications with Noise) is well known to be good at discovering clusters with uniform densities [13], but it may fail in detecting clusters of different densities [14]. SC (acronym of Spectral Clustering) is a clustering method based on graph theory [15], [16]. It works well on clusters with similar distribution, where similar distribution means that clusters are similar in terms of their shapes, structures and number of data points. However, SC is not good at identifying clusters of varying distributions. OPTICS (acronym of Ordering Points To Identify the Clustering Structure) generates an augmented cluster-ordering graph, which is subsequently used to analyze and extract clusters [17], [18]. However, it is difficult to determine the precise boundary of clusters using the augmented cluster-ordering graph.

DPC (acronym of Clustering method by fast search and find of Density Peaks) [19] is a recently proposed ingenious method [20]. For N data points in a dataset $D = (x_1, x_2, \ldots, x_N)^T$ the similarities between data points are defined as $S_{ij} = \|x_i - x_j\|, \forall i < j$, and then the elements in S, which is the set of all S_{ij}, are sorted in descending order to form a vector $s = (s_1, s_2, \ldots, s_{N(N-1)/2})$. DPC generates



(a) Decision graph (b) Clustering results generated by DPC

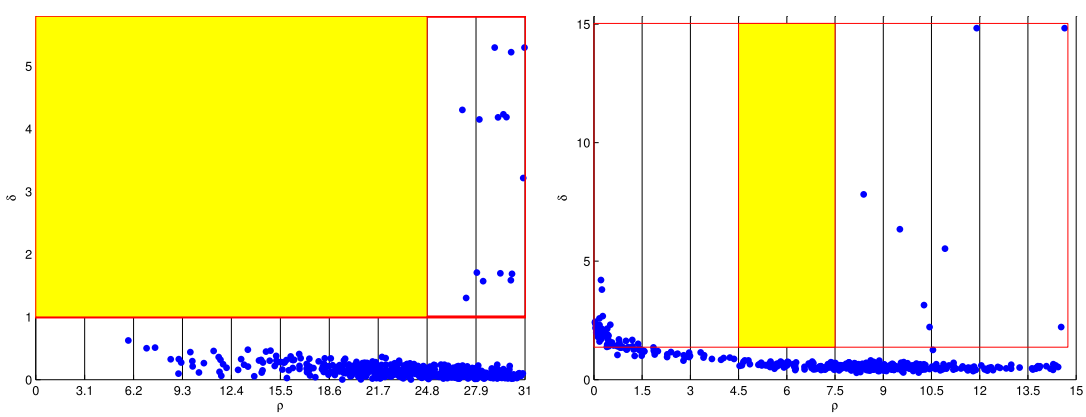
Fig. 1. Clustering results generated by DPC on Flame [23] dataset.



(a) Anchor points identified in the R15 dataset [40] ($mdelta = 1$)

(b) Anchor points identified in the Compound dataset [41] ($mdelta = 1.37$)

Fig. 2. Anchor points (data points in the red rectangle) identified by APC.



(a) One local density level identified in R15

(b) Two local density levels identified in Compound

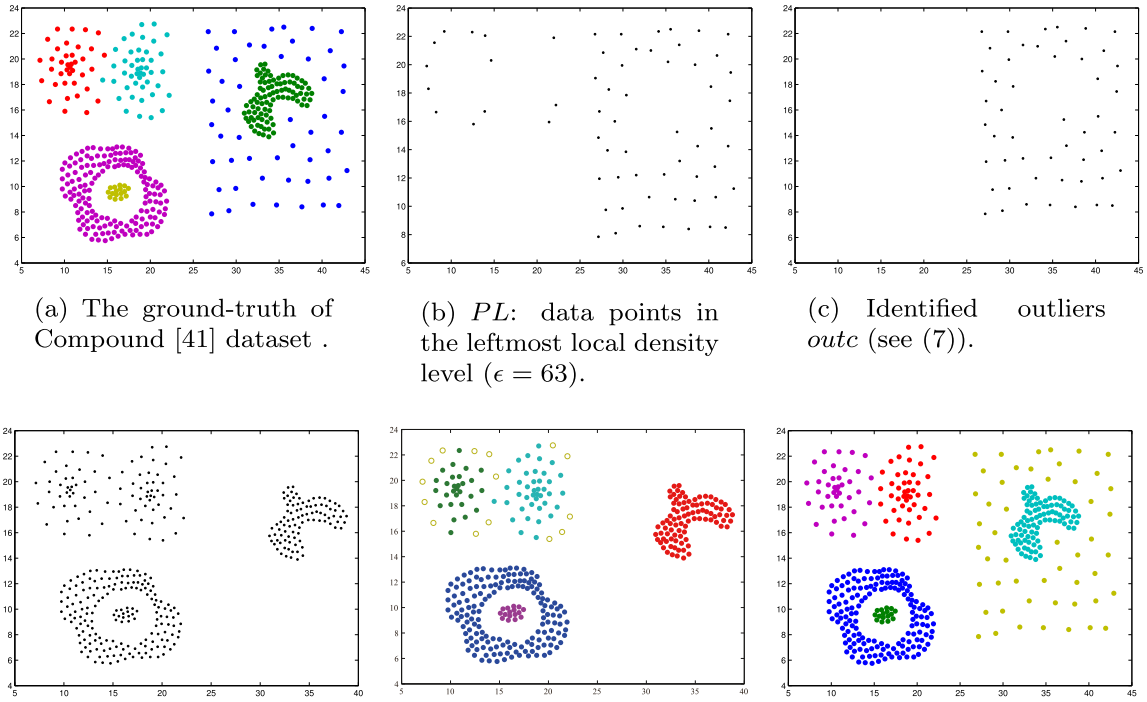
Fig. 3. Gaps and local density levels identified in two datasets. The yellow bins represent the gaps and the remaining area in the red rectangle represents the respective local density levels.

a two-dimensional decision graph (see Fig. 1(a)) according to two computed parameters, namely the local density ρ and the minimum distance between one data point and another with higher density δ [21]. DPC suggests that the data points with large ρ and δ values are appropriate to be selected as cluster centers, which represent density peaks [19]. Subsequently, each remaining data point is sequentially assigned to its nearest higher density neighbor so as to form clusters. The local density of data point x_i is computed as follows:

$$\rho_i = \sum_j e^{-\left(\frac{\|x_i - x_j\|}{d_c}\right)^2}, \tag{1}$$

where $d_c = s_{pct \cdot N \cdot (N-1)/200}$. Parameter pct is user-defined to regulate the value of d_c. Parameter δ of data point x_i is defined as

$$\delta_i = \min_{j:\rho_j > \rho_i} S_{ij}. \tag{2}$$



(a) The ground-truth of Compound [41] dataset .

(b) PL: data points in the leftmost local density level ($\epsilon = 63$).

(c) Identified outliers $outc$ (see (7)).

(d) The remaining ($|D| - |outc|$) number of data points after applying the purifying algorithm.

(e) Further processing results of (d). Empty circles represent outliers. Data points in cyan, green and purple represent three micro-clusters (see (9)), respectively.

(f) Final clustering results after applying the micro-cluster optimization algorithm.

Fig. 4. Step-by-step demonstrations of APC dynamics using the Compound dataset (in this case, $num\_level \geq 2$). Parameter values being used: $pct = 1.9$, $mdelta = 1.37$, $x_{num} = 10$, $k = 42$, $MinPts = 3$ and $Eps = 0.85$.

DPC has the ability of identifying outliers. Specifically, as the simple illustrations shown in Fig. 1, DPC can help to identify outliers with a clear definition: outliers often have relatively lower ρ and higher δ (see Page 3 of the original DPC paper [19]). As shown in the bottom left of Fig. 1(a), the two data points (corresponding to the two data points in the top left of Fig. 1(b)) in the green rectangle have relatively lower ρ and higher δ to be considered as outliers. According to [22], DPC can be regarded as a clustering-based outliers detection technique that normal data instances belong to large and dense clusters, while outliers either belong to small or sparse clusters [22]. In addition to finding outliers, DPC can also help to categorize all the data points into different levels based on their local densities. These advantages of DPC constitute why we use DPC's key mechanisms to remove outliers.

To identify clusters with varying shapes, sizes and densities, in this paper, we propose a novel clustering algorithm named Anchor Points based Clustering (APC). As a widely adopted assumption, if a data point is far away from its nearest higher density neighbor, it is likely to be a cluster center [19]. We refer certain data points to anchor points, which have relatively large δ values (both ρ and δ values in APC are computed in the same way as DPC does). The concept of anchor points (also known as representatives or landmarks in literature) has also been exploited in existing large-scale spectral clustering methods [24,25], which select anchor points by random selection or K-means based selection. Although both use anchor points to perform the clustering task, the usage differs. Our method selects anchor points to identify the data structure complexity via the density peak clustering approach, while the anchor points based spectral methods [24,25] aim to alleviate the large computational overhead.

We use local density ρ to analyze the intrinsic data structure of the underlying dataset becasue local density ρ is an effective variable used to describe the spacial characteristics of data points. After anchor points are identified by APC, they are chosen as cluster centers and each remaining data point is sequentially assigned to its nearest higher density neighbor to produce intermediate clusters. Moreover, all the anchor points are autonomously categorized into different levels based on their local densities. Anchor points belong to the same level (having similar local densities) and those belong to other levels are treated differently. If certain anchor points have similar local densities, the corresponding intermediate clusters belonging to the same local density level are deemed as having similar densities as well. At this stage, the key factor affecting the clustering results is whether there are connected data points (or connected points in short) [13]. The reason is as follows: connected points refer to the data points that locate near the boundaries of the natural clusters and they are ambiguous in the decision of which clusters they belong to. If there is only one local density level of anchor points identified in the dataset, we simply employ the DPC dynamics to proceed with the clustering process, because DPC works well on identifying clusters with similar densities (i.e., the clusters have highly identifiable single density peaks belonging to the same local density level). On the other hand, if anchor points are found in multiple density levels, we employ the improved DBSCAN dynamics to identify clusters, because DBSCAN works well on handling the connected points.

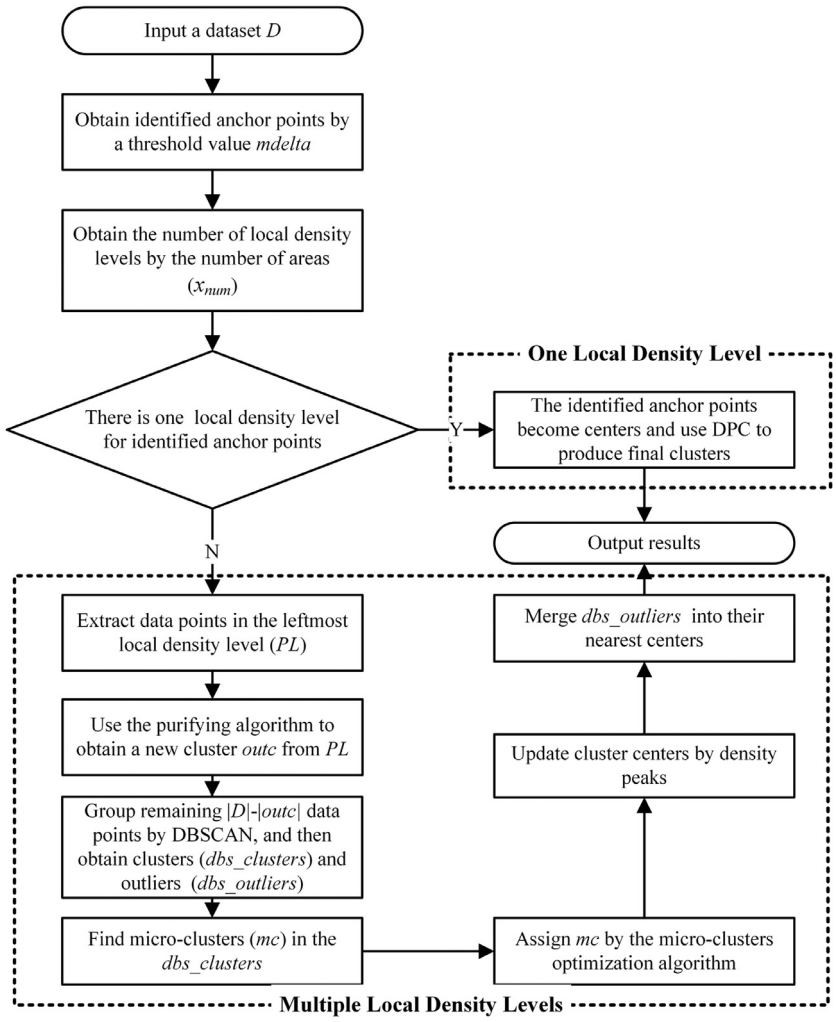


Fig. 5. Flowchart of the proposed APC algorithm.

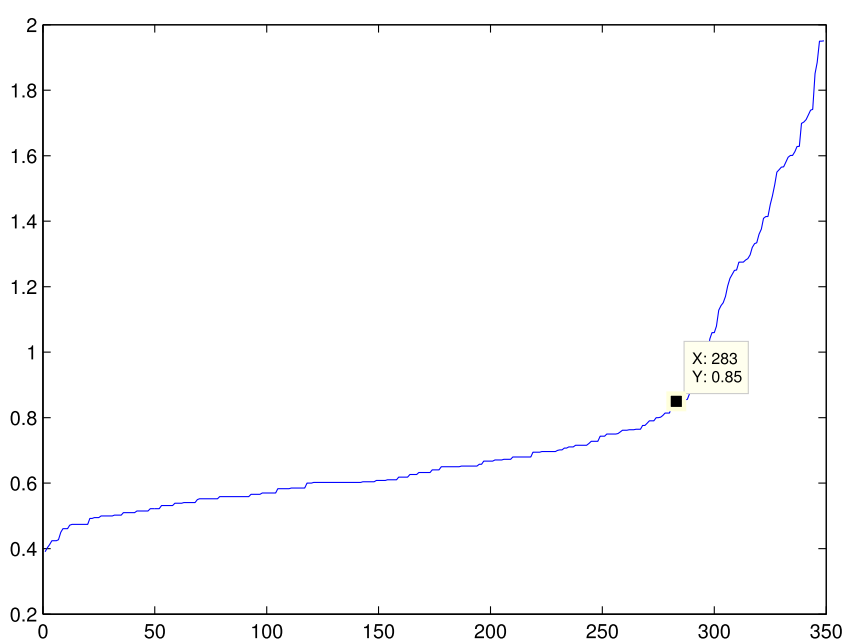
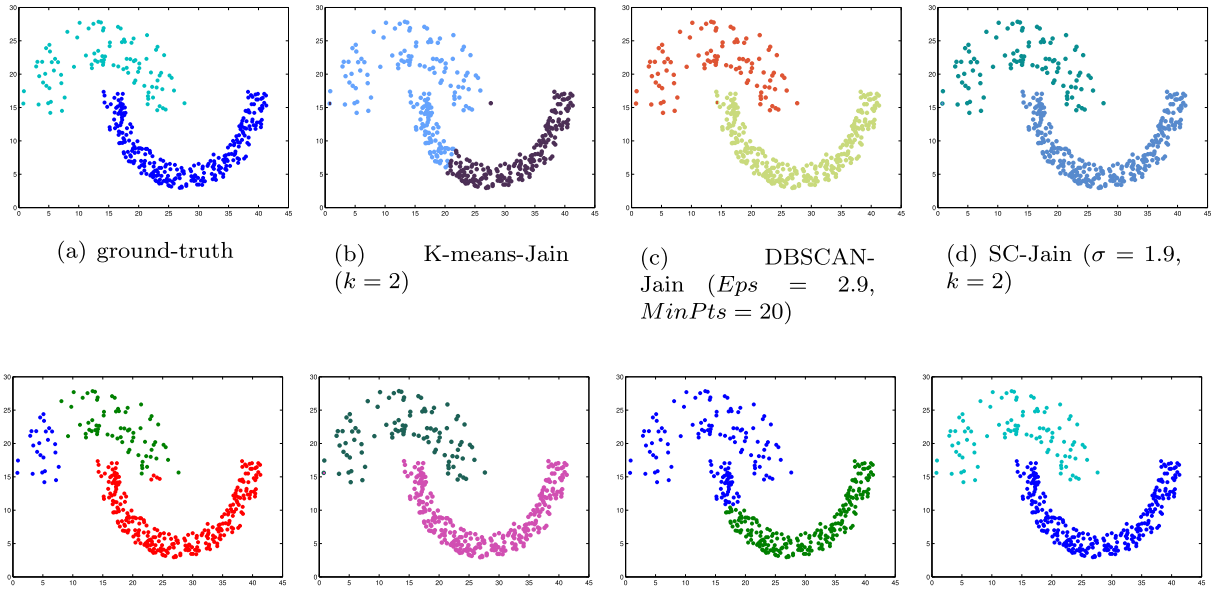


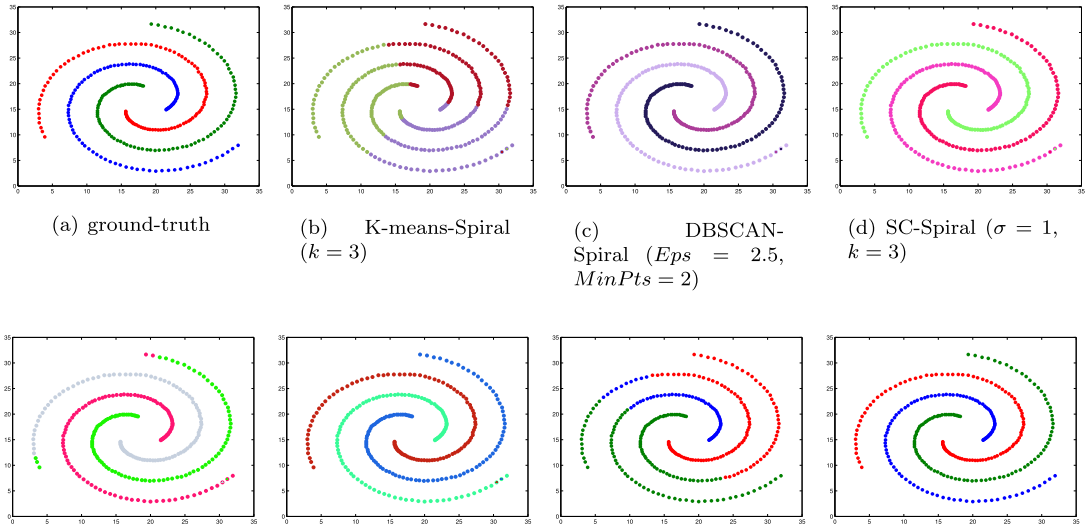
Fig. 6. Illustration of determining *MinPts* and *Eps* values in APC. As shown, in Compound dataset, the annotated point (283,0.85) is identified as inflection point (valley).



(a) ground-truth (b) K-means-Jain ($k = 2$) (c) DBSCAN-Jain ($Eps = 2.9$, $MinPts = 20$) (d) SC-Jain ($\sigma = 1.9$, $k = 2$)

(e) OPTICS-Jain ($k = 8$) (f) DPC-Jain ($pct = 40$) (g) DPC-KNN-PCA-Jain ($pct = 3$, $d = 2$, $k = 15$) (h) APC-Jain ($pct = 50$, $mdelta = 5.5$)

Fig. 7. Result comparisons on Jain.



(a) ground-truth (b) K-means-Spiral ($k = 3$) (c) DBSCAN-Spiral ($Eps = 2.5$, $MinPts = 2$) (d) SC-Spiral ($\sigma = 1$, $k = 3$)

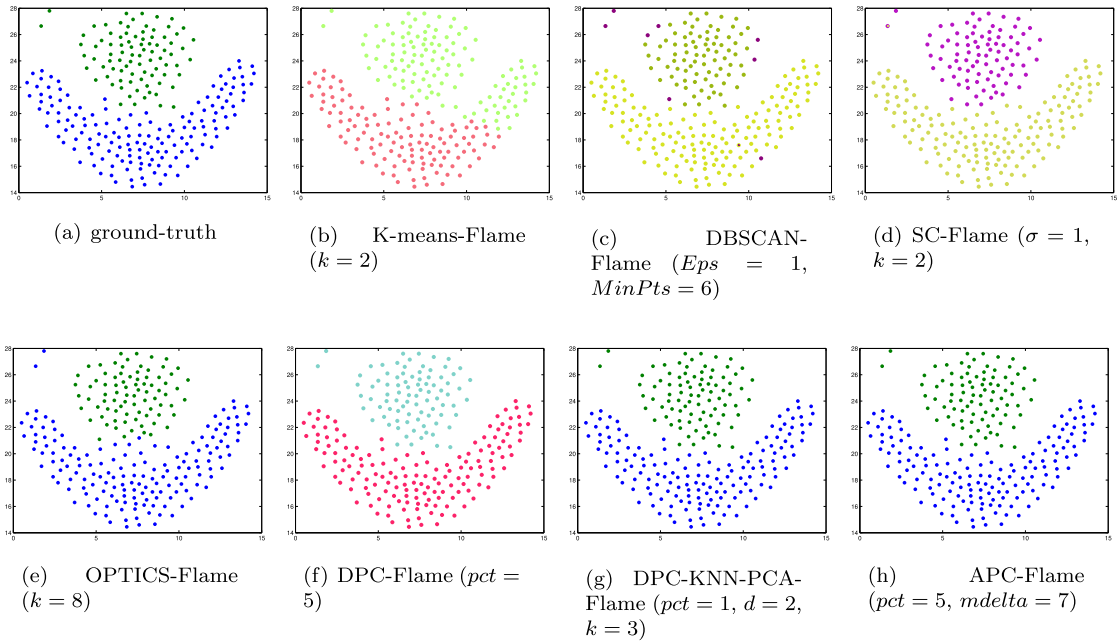
(e) OPTICS-Spiral ($k = 8$) (f) DPC-Spiral ($pct = 5$) (g) DPC-KNN-PCA-Spiral ($pct = 2$, $d = 2$, $k = 4$) (h) APC-Spiral ($pct = 5$, $mdelta = 4$)

Fig. 8. Result comparisons on Spiral.

In a nutshell, our proposed APC algorithm synthesizes the advantages of two well-known methods DPC and DBSCAN: DPC can easily detect outliers (data points with lower ρ and higher δ values) and DBSCAN is good at identifying clusters with uniform densities. Nonetheless, we did not straightforwardly combine DPC and DBSCAN to derive APC. Instead, empowered by our proposed autonomous density analysis on the intrinsic structure of the dataset, APC adopts the key mechanisms of DPC and DBSCAN in a unified clustering framework to perform the clustering procedures in an autonomous manner according to the analyzed local density distribution. As the experimental results shown, APC outperforms six benchmarking methods in most cases. The main contributions of this paper are summarized as follows:

(i) We propose a novel clustering method named APC to identify clusters of different shapes, sizes and densities from a new perspective, which relies on the identified density levels of anchor points.
(ii) We propose a new data structure analysis method based on systematic analysis on the local density levels of anchor points.
(iii) We demonstrate the effectiveness of APC using both widely adopted synthetic datasets and real-world ones.

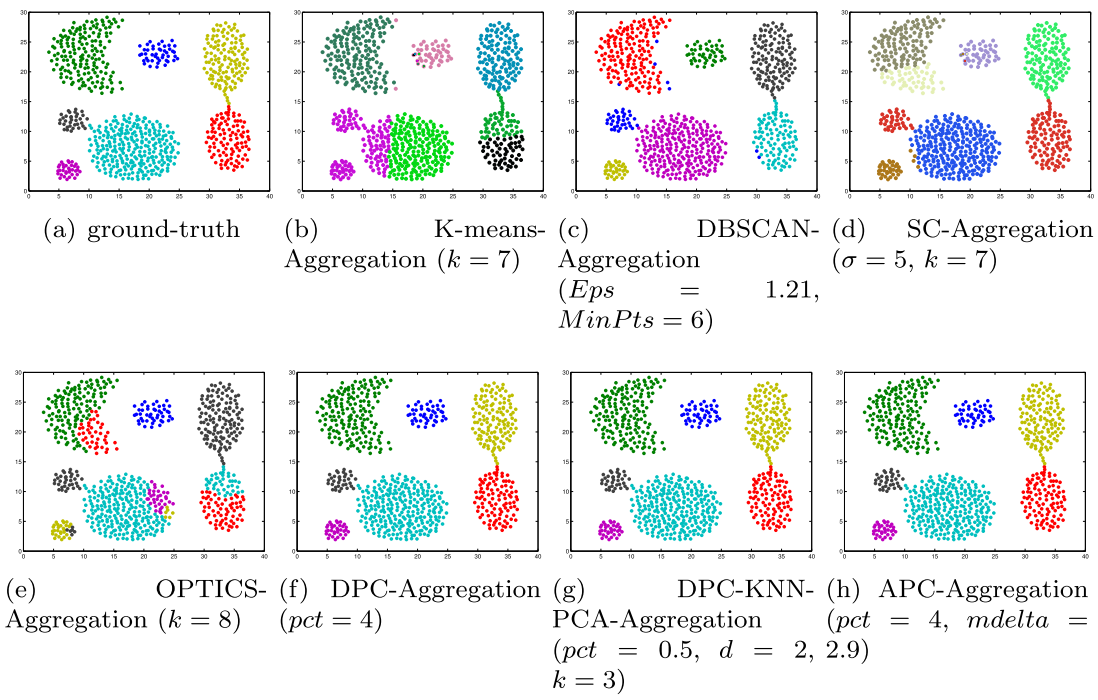
The rest of this paper is organized as follows. In Section 2, we review related work. In Section 3, we present the details of our proposed APC method. In Section 4, we report the experimental



(a) ground-truth (b) K-means-Flame ($k = 2$) (c) DBSCAN-Flame ($Eps = 1$, $MinPts = 6$) (d) SC-Flame ($\sigma = 1$, $k = 2$)

(e) OPTICS-Flame ($k = 8$) (f) DPC-Flame ($pct = 5$) (g) DPC-KNN-PCA-Flame ($pct = 1$, $d = 2$, $k = 3$) (h) APC-Flame ($pct = 5$, $mdelta = 7$)

Fig. 9. Result comparisons on Flame.



(a) ground-truth (b) K-means-Aggregation ($k = 7$) (c) DBSCAN-Aggregation ($Eps = 1.21$, $MinPts = 6$) (d) SC-Aggregation ($\sigma = 5$, $k = 7$)

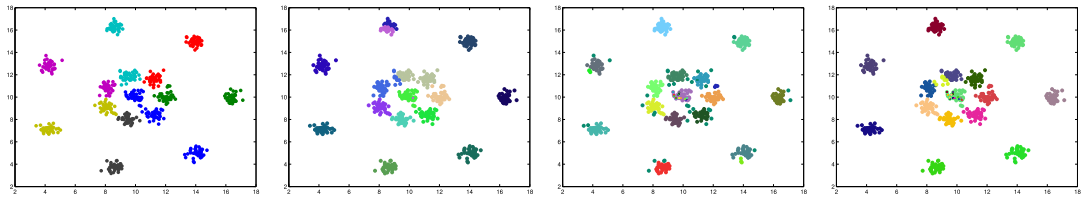
(e) OPTICS-Aggregation ($k = 8$) (f) DPC-Aggregation ($pct = 4$) (g) DPC-KNN-PCA-Aggregation ($pct = 0.5$, $d = 2$, $k = 3$) (h) APC-Aggregation ($pct = 4$, $mdelta = 2.9$)

Fig. 10. Result comparisons on Aggregation.

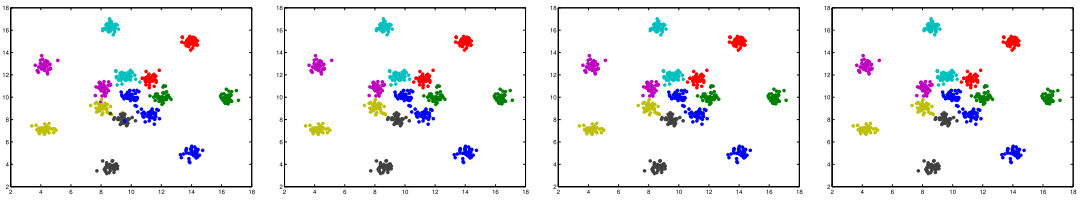
results with discussions. In Section 5, we conclude the paper and propose future work.

2. Related work

In recent years, many clustering algorithms have been proposed that can well handle various types of cluster formations [26,27]. For example, a series of methods extending KNN (K Nearest Neighbor) and SNN (Shared Nearest Neighbor) have been proposed. Du et al. proposed an improved DPC algorithm based on KNN and PCA (Principal Component Analysis) to solve the issue that DPC may overlook certain clusters and get inferior results on high dimensional data [28]. Xie et al. improved DPC by adopting new points assignment strategies based on KNN in order to enhance its robustness [29]. Liu *et al.* introduced KNN to compute the two parameters of DPC and finally aggregated clusters if they are density reachable [30]. Parmar et al. propose a residual error-based DPC algorithm to better identify overlapping clusters [31]. Xu developed a density peak based hierarchical clustering method (DenPEHC), which directly generates clusters on each possible clustering layer, and introduces a grid granulation framework to enable DenPEHC to cluster large-scale and high-dimensional datasets [32]. Mehmood et al. firstly found the local density regions and subsequently merged the density connected regions to form meaningful clusters [33]. Chen et al. proposed a novel clustering algorithm named CLUB, which finds density backbone of clusters on the basis of KNN and SNN [34]. However, it is hard for CLUB to deal with clusters that violate the nearest neighbor rule (i.e., closer data

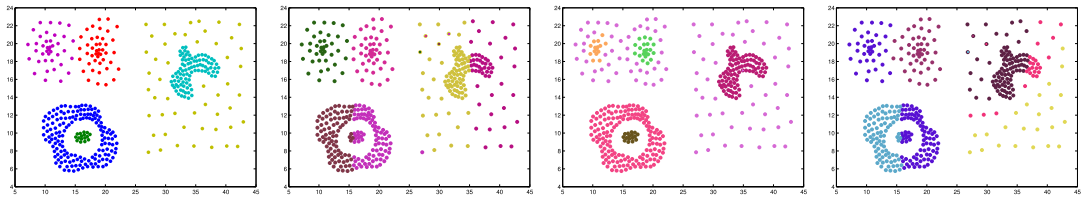


(a) ground-truth (b) K-means-R15 ($k = 15$) (c) DBSCAN-R15 ($Eps = 0.3$, $MinPts = 3$) (d) SC-R15 ($\sigma = 5$, $k = 15$)

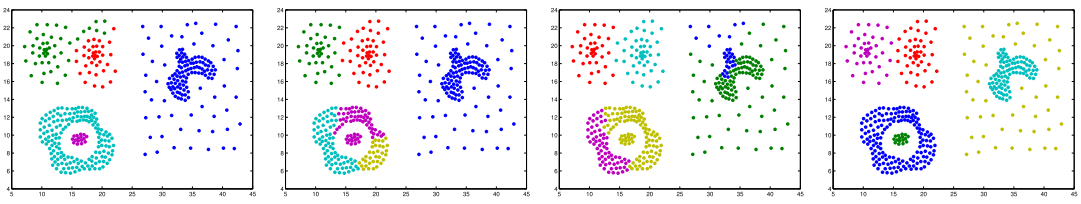


(e) OPTICS-R15 ($k = 7$) (f) DPC-R15 ($pct = 5$) (g) DPC-KNN-PCA-R15 ($pct = 0.5$, $d = 2$, $k = 3$) (h) APC-R15 ($pct = 5$, $mdelta = 1$)

Fig. 11. Result comparisons on R15.



(a) ground-truth (b) K-means-Compound ($k = 6$) (c) DBSCAN-Compound ($Eps = 1$, $MinPts = 5$) (d) SC-Compound ($\sigma = 5$, $k = 6$)

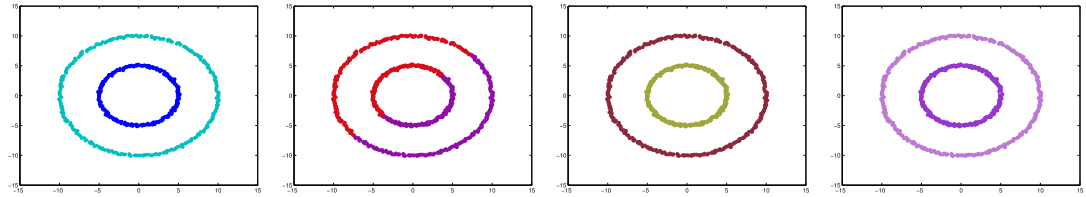


(e) OPTICS-Compound ($k = 2$) (f) DPC-Compound ($pct = 5$) (g) DPC-KNN-PCA-Compound ($pct = 5$, $d = 2$, $k = 8$) (h) APC-Compound ($pct = 1.9$, $mdelta = 1.37$, $Eps = 0.85$, $MinPts = 3$)

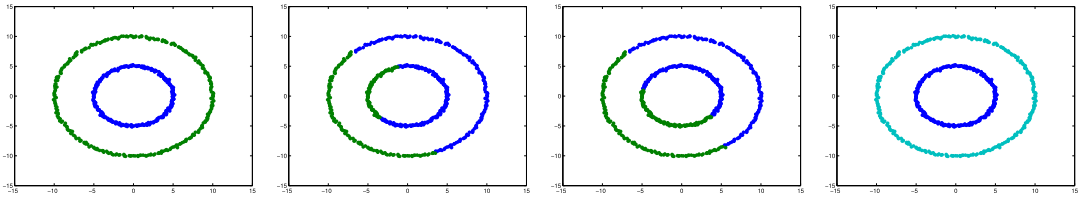
Fig. 12. Result comparisons on Compound.

points are more likely to be merged into a cluster). Ren et al. proposed two-steps deep density-based image clustering, firstly used deep convolutional autoencoder and t-SNE to obtain 2-dimensional features, then merged local clusters into final results by their density relationship [35]. Ren et al. also proposed two-steps parallel boosted clustering to address the scalability issue [36]. In addition, K-dist graph [37] is often used to divide a dataset into subsets with different densities. Specifically, K-dist graph is a two-dimensional graph, wherein y-axis represents sorted distance between kth nearest neighbor and each data point and x-axis represents corresponding data points. Gaonkar and Kedar presented their method to autonomously determine the input parameters values of DBSCAN by K-dist graph [38]. However, it is difficult for these methods based on K-dist graph to find drastic changes between two density levels and across cluster boundaries in many datasets. There are also studies in the literature using the differential evolution method to form clusters with different densities [39]. Nonetheless, the local clusters with uniform densities may often be ignored.

As a brief summary, it is difficult for the afore-reviewed clustering methods to identify clusters with arbitrary shapes, sizes and densities. Prior studies often consider a single trait of natural clusters, for example, DBSCAN mainly considers whether the density of different natural clusters are similar or not and DPC mainly considers whether the natural clusters have single density peak. In this research, we address this issue from a new perspective that we

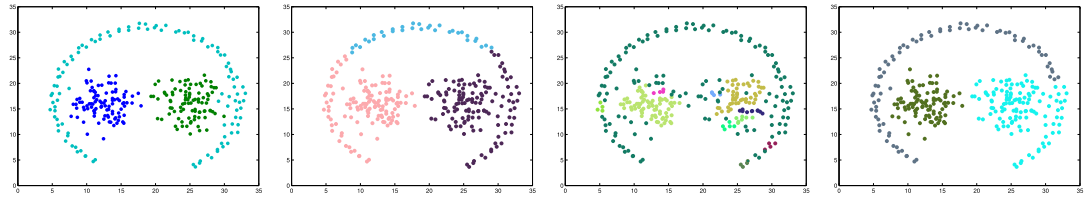


(a) ground-truth (b) K-means-2circles ($k = 2$) (c) DBSCAN-2circles ($Eps = 1, 5, k = 6$ $Minpts = 2$) (d) SC-2circles ($\sigma =$

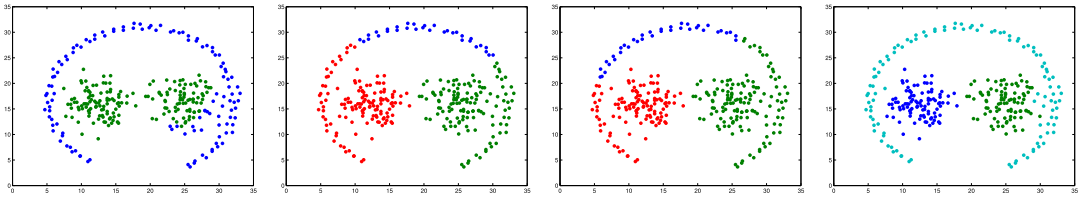


(e) OPTICS-2circles ($k = 2$) (f) DPC-2circles ($pct = 5$) (g) DPC-KNN-PCA-2circles ($pct = 1$, $d = 2$, $k = 2$) (h) APC-2circles ($pct = 2$, $mdelta = 3$, $Eps = 1$, $MinPts = 2$)

Fig. 13. Result comparisons on 2circles.



(a) ground-truth (b) K-means-Pathbased ($k = 3$) (c) DBSCAN-2circles ($Eps = 1$, $Minpts = 4$) (d) SC-Pathbased ($\sigma = 0.7$, $k = 3$)



(e) OPTICS-Pathbased ($k = 10$) (f) DPC-Pathbased ($pct = 2$) (g) DPC-KNN-PCA-Pathbased ($pct = 0.5$, $d = 2$, $k = 5$) (h) APC-Pathbased ($pct = 0.55$, $mdelta = 3.5$, $Eps = 1.5$, $MinPts = 5$)

Fig. 14. Result comparisons on Pathbased.

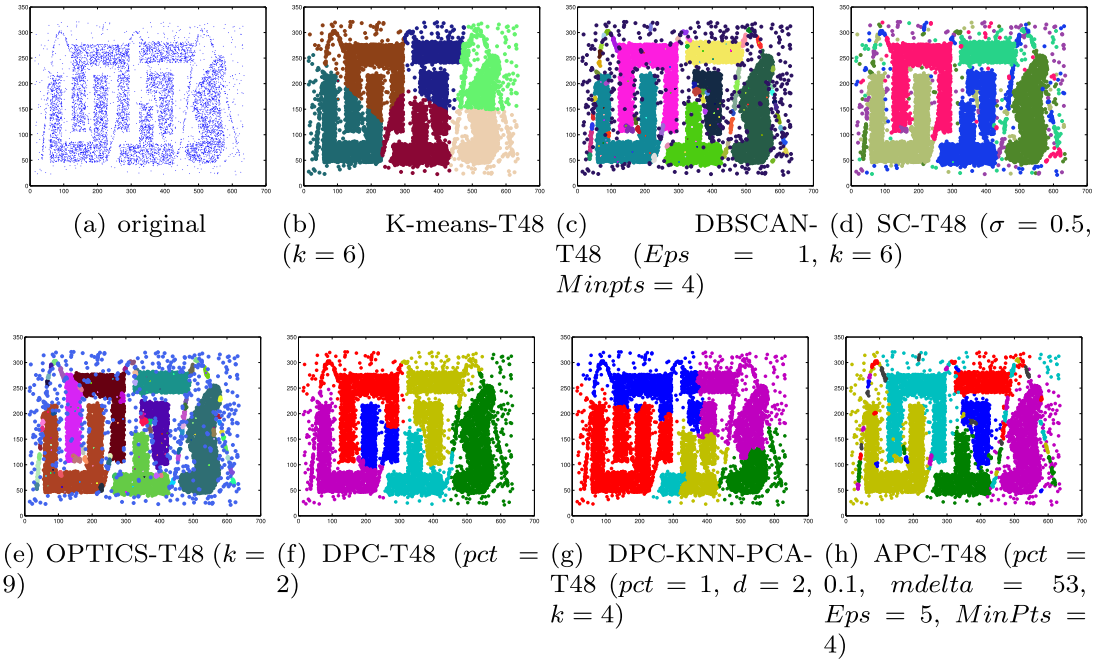
consider the whole structure of a dataset based on local density levels. All data points in a dataset are categorized into different local density levels based on the identified anchor points. Subsequently, our clustering method autonomously selects the appropriate clustering strategy (see Section 3.3) to further obtain the final clustering results.

3. Density levels of anchor points based clustering

In this research, we propose a novel Anchor Points based Clustering (APC) algorithm to identify clusters with arbitrary shapes, sizes and densities. APC consists of three main processes: (i) APC selects anchor points that have relatively higher δ values from the decision graph. (ii) Subsequently, the number of local density levels of anchor points are determined, which roughly represent the complexity of the intermediate clusters. (iii) Finally, APC autonomously chooses appropriate clustering strategies based on the identified local density levels to obtain the final cluster formation. We introduce the technical details of APC in the following subsections.

3.1. Anchor points identification

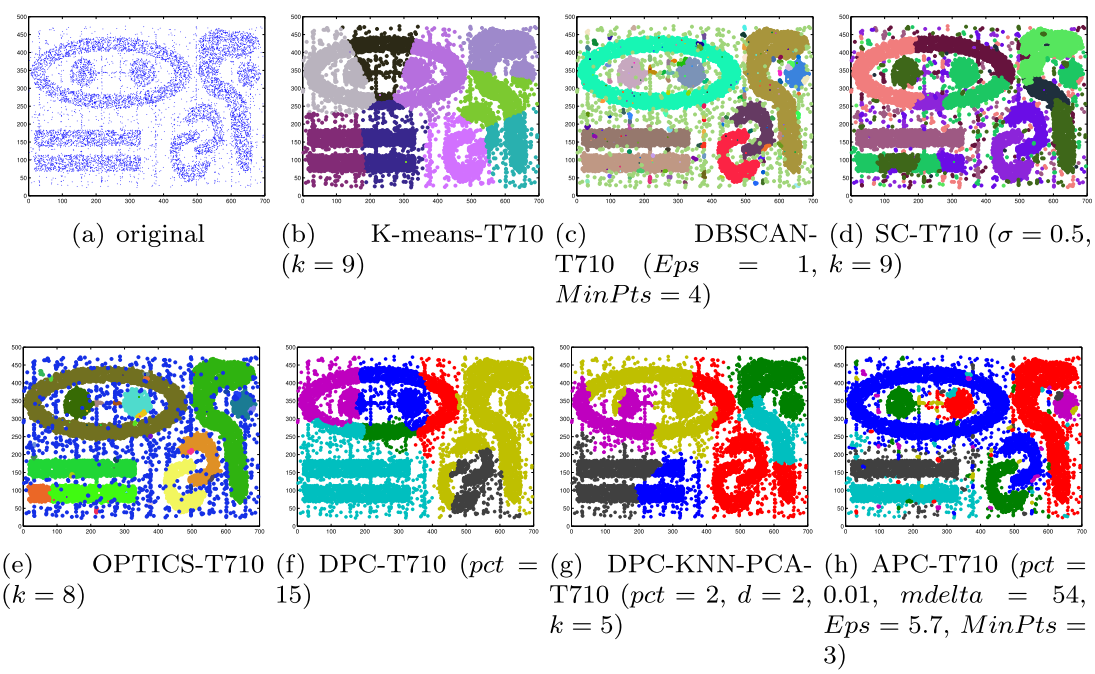
Data points with higher δ values are likely to be chosen as centers based on the intuitive definition of δ. In our proposed APC method, we first only consider δ values and use a rectangle to



(a) original (b) K-means-T48 ($k = 6$) (c) DBSCAN-T48 ($Eps = 1$, $Minpts = 4$) (d) SC-T48 ($\sigma = 0.5$, $k = 6$)

(e) OPTICS-T48 ($k = 9$) (f) DPC-T48 ($pct = 2$) (g) DPC-KNN-PCA-T48 ($pct = 1$, $d = 2$, $k = 4$) (h) APC-T48 ($pct = 0.1$, $mdelta = 53$, $Eps = 5$, $MinPts = 4$)

Fig. 15. Result comparisons on T48.



(a) original (b) K-means-T710 ($k = 9$) (c) DBSCAN-T710 ($Eps = 1$, $MinPts = 4$) (d) SC-T710 ($\sigma = 0.5$, $k = 9$)

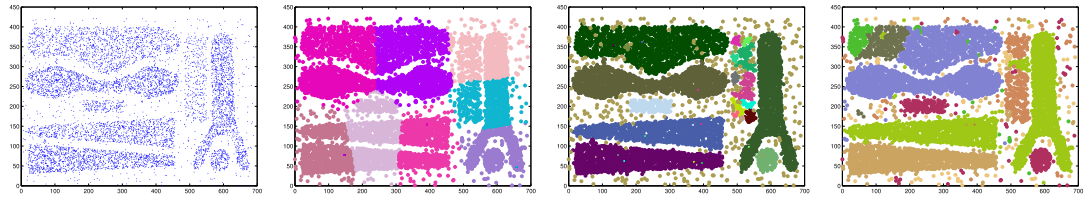
(e) OPTICS-T710 ($k = 8$) (f) DPC-T710 ($pct = 15$) (g) DPC-KNN-PCA-T710 ($pct = 2$, $d = 2$, $k = 5$) (h) APC-T710 ($pct = 0.01$, $mdelta = 54$, $Eps = 5.7$, $MinPts = 3$)

Fig. 16. Result comparisons on T710.

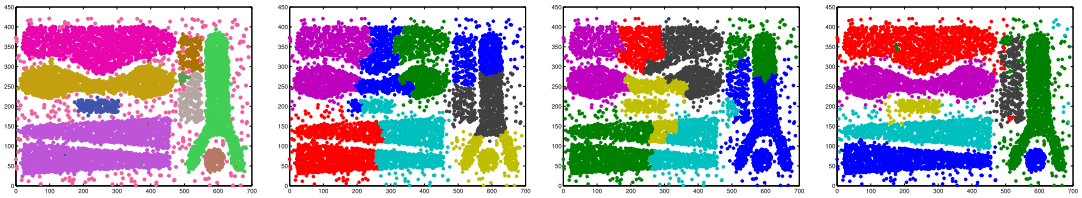
choose anchor points from the decision graph, which is generated in the same way as in DPC (see Section 1). The coordinate of the lower left corner of the rectangle is selected as (0, *mdelta*) and that of the upper right corner is selected as (max(ρ_i), max(δ_i)). The determination of this rectangle only relies on one parameter *mdelta*, which denotes the cutoff value of δ to define the range of anchor points. All the data points fall in the rectangle are selected as anchor points (see the red rectangles in Fig. 2). APC takes anchor points as cluster centers, then the remaining data points are subsequently assigned according to their densities and distance to the neighboring anchor points to form all the intermediate clusters. Then, these intermediate clusters are further processed to form the final clusters (see Section 3.3). The reason of obtaining the intermediate clusters is to decompose the potentially complex cluster structure into relatively simpler intermediate cluster generation.

The value of *mdelta* greatly affects the overall performance of APC. If its value is set too small, the intermediate clusters may include many outliers and borderline data points (the data points with smaller δ and ρ values on the decision graph). If the value of *mdelta* is set too large, the intermediate clusters may not represent simple clusters with unique density peaks.

Therefore, in terms of setting the value of *mdelta*, we make sure it fulfills the following criterion: all the data points with larger ρ values and larger δ values fall in the afore-introduced selection

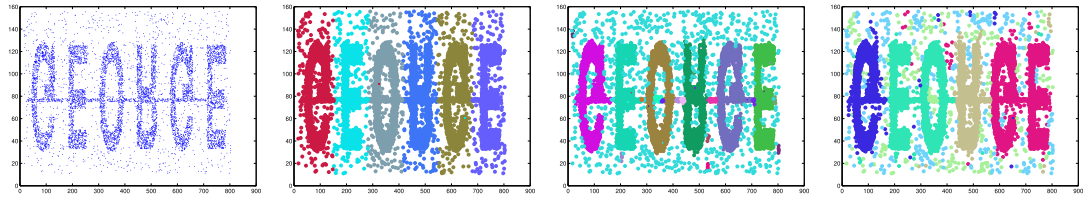


(a) original (b) K-means-T88 ($k = 8$) (c) DBSCAN-T88 ($Eps = 1$, $MinPts = 4$) (d) SC-T88 ($\sigma = 1$, $k = 8$)

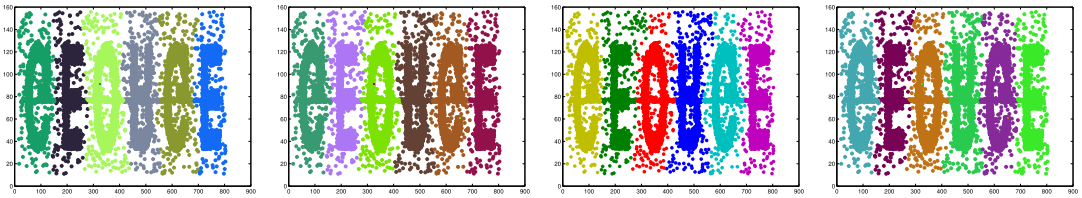


(e) OPTICS-T88 ($k = 9$) (f) DPC-T88 ($pct = 0.1$) (g) DPC-KNN-PCA-T88 ($pct = 2$, $d = 2$, $k = 5$) (h) APC-T88 ($pct = 0.005$, $mdelta = 49$, $Eps = 8.5$, $MinPts = 6$)

Fig. 17. Result comparisons on T88.



(a) original (b) K-means-T58 ($k = 6$) (c) DBSCAN-T58 ($Eps = 4$, $Minpts = 4$) (d) SC-T58 ($\sigma = 0.5$, $k = 6$)



(e) OPTICS-T58 ($k = 13$) (f) DPC-T58 ($pct = 2$) (g) DPC-KNN-PCA-T58 ($pct = 2$, $d = 2$, $k = 5$) (h) APC-T58 ($pct = 2$, $mdelta = 73.4$)

Fig. 18. Result comparisons on T58.

rectangular (see Fig. 2). Specifically, in here, a larger value means the value is larger than the median value. The rationality of setting the value of *mdelta* as such is to enable the further determination of cluster belongingness of all the outliers as they are selected as anchor points and centers to form intermediate clusters. The formation of intermediate clusters and the further processing are expected to improve the overall clustering results.

3.2. Local density levels of anchor points

We propose a method to determine whether the local density levels of the identified anchor points are distinguishable and compute the number of local density levels, denoted *num_level*. With reference to the red rectangles outlining the anchor points areas in the decision graph shown in Fig. 3(a) and (b), we segment each rectangular into $x_{num} \in \mathbb{Z}^+$ bins along the X-axis (ρ), then the average interval value x_{inte} can be computed as

$$x_{inte} = (\lceil max(\rho_i) \rceil)/x_{num}. \tag{3}$$

Moreover, the mth bin along the X-axis, denoted BIN_m, is defined as

$$BIN_m = \{(\rho, \delta) \in G \mid (m-1)x_{inte} \leq \rho \leq mx_{inte}, 0 \leq \delta \leq \lceil max(\delta_i) \rceil\}, m \in [1, x_{num}], \tag{4}$$

where G represents the set of all data points within the decision graph. If two or more continuous bins have no anchor points, we call these continuous bins a gap. As such, the gap represents distinguishable difference between anchor points in different density

Fig. 19. Clustering results obtained by APC on Olivetti Face dataset.

levels. The region of a single gap, denoted *GAP*, can be defined as follows:

$$GAP = \{(\rho, \delta) \in G \mid mx_{inte} \leq \rho \leq (m + num\_bin)x_{inte}, 0 \leq \delta \leq \lceil max(\delta_i) \rceil\}, num\_bin \geq 2, \tag{5}$$

where *num_bin* represents the number of bins in the corresponding gap. We use *GAP* to determine the number of local density levels. For example, in Fig. 3(a), if we set $pct = 5$, $mdelta = 1$, $x_{num} = 10$, then $x_{inte} = 3.1$ (see (3)). The yellow area represent a gap that $num\_bin = 8$. The remaining area within the red rectangle represent a single local density level. In Fig. 3(b), if we set $pct = 1.9$, $mdelta = 1.37$, $x_{num} = 10$, then $x_{inte} = 1.5$, $num\_bin = 2$. The yellow gap divides the anchor points into two local density levels.

3.3. Clustering based on local density levels of anchor points

Based on the number of local density levels identified, APC autonomously adopts appropriate strategies to further determine the final cluster formation delineated in the following two subsections.

3.3.1. When $num\_level = 1$

If all identified anchor points fall into one local density level (see Fig. 3(a)), the corresponding intermediate clusters have similar densities. As introduced in Section 1, connected points may affect clustering results. In this case, the connected points between different neighboring intermediate clusters can be correctly assigned, because they have relatively lower local densities. As such, the intermediate clusters are reported as final results without further processing. When $num\_level = 1$, APC has three parameters, namely *pct, mdelta* and x_{num}.

3.3.2. When $num\_level \geq 2$

In this situation, we use two-step strategies to identify clusters: (i) Extracting outliers from the whole dataset. (ii) Grouping the remaining data points for each local density level. Specifically, the key mechanisms of DPC are used to extract the outliers. Outliers often have lower local density and are located at the leftmost local density level [22]. Thus, outliers can be easily identified by adopting the key mechanisms of DPC.

(i) Extracting outliers from the whole dataset. We use *PL* to denote the set of data points belonging to the leftmost local density level (the number of data points in *PL* is denoted as ϵ). Obviously, *PL* comprises the edge points of natural clusters (data points with lower ρ and lower δ values) and outliers in the dataset (data points with lower ρ but higher δ values) (see Fig. 4(b)). To further distinguish which data points in *PL* should be considered as outliers, we employ an improved K-dist graph method to assess their densities. As introduced in Section 2, K-dist graph [37] is a useful tool to divide a dataset into subsets with different densities. In this research, we extended it by using the averaged distance to its k nearest neighbors instead. Specifically, we use $kavg_j$ to denote the average distance from data point PL_j to its k nearest neighbors in *PL* as follows:

$$kavg_j = \frac{1}{k} \sum_k \|PL_j, PL_k\|, \tag{6}$$

where k denotes the number of nearest neighbors in *PL* for PL_j and $\| \cdot \|$ computes the Euclidean distance between two data points. When $\forall x_j, kavg_{(j+1)} \geqslant 2kavg_j$, the outliers *outc* are determined in an autonomous manner as follows

$$outc = \begin{cases} \{PL_1, \ldots, PL_j\}, & j > \epsilon/2, \\ \{PL_{(j+1)}, \ldots, PL_\epsilon)\}, & j \leq \epsilon/2, \end{cases} \tag{7}$$

when none of x_j is subject to $kavg_{(j+1)} \geqslant 2kavg_j$, all the data points in PL can be merge into a new cluster because they have similar and lower local density. We refer this additional process of distinguishing outliers to the purifying algorithm (see Subalgorithm Purifying in Algorithm 1). The intuition of taking $2kavg_j$ as the threshold to differentiate whether a gap exists is that density is empirically deemed as significantly different when the value of $kavg_{(j+1)}$ is doubled (see detailed elaborations in Section 4.1).

Algorithm 1: Anchor Points based Clustering.

Input: dataset D and parameters pct, $mdelta$, x_{num}, k, $MinPts$ and Eps
Output: assigned cluster indices

based on the density distribution in D, generate a 2-dimensional decision graph using pct (see (1) and (2);
identify anchor points in the generated decision graph using $mdelta$ (see Section~3.1);
further categorize local density bins among the identified anchor points using x_{num} (see Section~3.2);
$num\_bin \leftarrow$ number of bins in gaps (see (5));
$num\_level \leftarrow$ number of identified local density levels;
if $num\_level \geq 2$ **then**
  $PL \leftarrow$ data points in the leftmost local density level;
  obtain the outlier cluster $outc$ in PL by applying **Subalgorithm Purifying**;
  for $i = 1 \rightarrow num\_level$ **do**
    obtain clusters $dbs\_clusters$ and outliers $dbs\_outliers$ in the ith local density level by applying DBSCAN with $MinPts$ and Eps;
    further process $dbs\_clusters$ by applying **Subalgorithm MCO**;
    update cluster centers by finding density peaks;
    assign $dbs\_outliers$ to their nearest centers;
  end
else
  anchor points become cluster centers and other data points are assigned to their nearest higher density cluster centers;
end
output the clustering index assignments;

Subalgorithm *Purifying* (PL, k)
  $kavg \leftarrow$ sorted average distance from data points to its k nearest neighbors (see (6));
  $\epsilon \leftarrow$ size of PL;
  for $j = 1 \rightarrow \epsilon$ **do**
    if $kavg(j+1) \geq 2kavg(j)$ **then**
      if $j > \epsilon/2$ *(see (7))* **then**
        remove data points of PL from $j+1$ to ϵ ;
      else
        remove data points of PL from 1 to j ;
      end
      the remaining data points in PL form the outlier cluster $outc$;
    else
      all the data points in PL can be merge into a new cluster;
    end
  end

Subalgorithm *Micro-Clusters Optimization (MCO)* $(dbs\_clusters)$
  $num\_dbc \leftarrow$ size of $dbs\_clusters$;
  $LDC \leftarrow$ local density of cluster centers in $dbs\_clusters$ (see (8));
  $\lambda \leftarrow 0$;
  for $n = 1 \rightarrow num\_dbc$ **do**
    if $LDC_n < mean(LDC)$ *see (9)* **then**
      $\lambda = \lambda + 1$;
    end
  end
  obtain λ micro-clusters;
  if $num\_dbc > 2(\lambda + 1)$ **then**
    merge the micro-clusters into the nearest non-micro-cluster centers;
  end

(ii) Grouping the remaining data points for each local density level. At this stage, there are $(|D| - |outc|)$ number of data points awaiting for further processing, which comprise anchor points, data points belonging to intermediate clusters, and remaining outliers (not identified in the previous stages). Because DBSCAN works well on identifying clusters of different sizes and handling the connected points at the same time, we employ the key mechanisms of DBSCAN to group the remaining $(|D| - |outc|)$ number of data points in each density level. These data points will be either identified as within the respective clusters, denoted $dbs\_clusters$, or outliers, denoted $dbs\_outliers$. The number of clusters obtained by applying DBSCAN is denoted as num_dbc.

We use LDC to denote the local densities of all the cluster centers for $dbs\_clusters$ as follows:

$$LDC = \{\rho_i | x_i \in C\}, \tag{8}$$

where C denotes all cluster centers in $dbs\_clusters$. In $dbs\_clusters$, if the local density of the nth cluster is smaller than the mean value in LDC, we consider it as a micro-cluster mc:

$$mc = \left\{ dbs\_clusters_n | LDC_n < \frac{1}{num\_dbc} \sum_{i=1}^{num\_dbc} (LDC_i) \right\}. \tag{9}$$

We use λ to denote the number of identified micro-clusters. Generally speaking, Centers of micro-clusters have lower local density and the corresponding micro-clusters may not qualify as the ultimate clusters according to DPC that a cluster center has higher local density ρ and higher δ. Moreover, based on our empirical studies, if λ is too small, i.e., $num\_dbc > 2(\lambda + 1)$, micro-clusters should be merged into their nearest non-micro-clusters (see detailed elaborations in Section 4.1). Specifically, when the number of micro-clusters (λ) is too few, i.e., less than half of num_{dbc}, these micro-clusters are merged into other core clusters with higher density centers. After the merging of micro-clusters, we update the cluster centers by finding density peaks. For outliers in $dbs\_outliers$, they are merged into their nearest clusters, wherein the distance is computed as the distance from the outlier to the cluster center. We refer to this process of further processing of clusters as the micro-cluster optimization (MCO) algorithm (see Subalgorithm MCO in Algorithm 1).

When $num\_level \geq 2$, APC requires six parameters, namely pct, $mdelta$, x_{num}, k, $MinPts$ and Eps. The latter three parameters are not being used for single density level (i.e., when $num\_level = 1$) and they are required by the improved K-dist graph method (k) and DBSCAN ($MinPts$ and Eps), respectively. An example shown in Fig. 4 demonstrates the step-by-step dynamics of APC when multiple local density levels are identified.

We present the pseudocodes of our proposed APC algorithm in Algorithm 1. Moreover, for more intuitive illustration of the dynamics of APC, we also present the flowchart of APC in Fig. 5.

Our method synthesizes the advantages of both DPC and DBSCAN to form a systematic density-based clustering method. As shown in Algorithm 1, the computation complexity of APC is $O(n^2)$, which is on the same order of magnitude as DPC and DBSCAN. The effectiveness of APC is assessed on multiple synthetic and real-world datasets. The experimental results are reported in the following section.

4. Experimental results

In this section, we use twelve synthetic datasets, twelve real-world datasets, and the Olivetti Face dataset to evaluate the performance of our proposed APC method. The properties of the synthetic and real-world datasets are listed in Table 1. Jain, Aggregation, Compound, Spiral, Flame, Pathbased and R15 datasets are downloaded from University of Eastern Finland[1]. Large scale datasets T48, T58, T710 and T88 are downloaded from Karypis Lab[2]

[1] http://cs.uef.fi/sipu/datasets/
[2] http://glaros.dtc.umn.edu/

Table 1
Dataset features.

Type	ID	Datasets	#Samples	#Dimensions	#Natural clusters
Synthetic	1	Jain	373	2	2
Synthetic	2	Spiral	312	2	3
Synthetic	3	Flame	240	2	2
Synthetic	4	Aggregation	788	2	7
Synthetic	5	R15 [40]	600	2	2
Synthetic	6	Compound [41]	399	2	6
Synthetic	7	2circles	600	2	2
Synthetic	8	Pathbased	300	2	2
Synthetic	9	T48	8000	2	*N/A*
Synthetic	10	T58	8000	2	*N/A*
Synthetic	11	T710	10,000	2	*N/A*
Synthetic	12	T88	8000	2	*N/A*
Real-world	1	Ecoli	178	13	8
Real-world	2	Sonar	208	60	2
Real-world	3	Wine	178	13	3
Real-world	4	Iris	150	4	3
Real-world	5	Liver	345	6	2
Real-world	6	Vehicle	846	18	4
Real-world	7	German	1000	24	2
Real-world	8	Pima	768	8	2
Real-world	9	Abalone	4177	8	3
Real-world	10	Gesture	1747	18	5
Real-world	11	Wine-white	4898	11	7
Real-world	12	Wine-red	1599	11	6

and they have no ground-truth defined. Dataset 2circles[3] is artificial and has multiple centers in the natural cluster. All the real-world datasets, namely Ecoli, Sonar, Wine, Iris, Liver, Vehicle, German, Pima, Abalone, Gesture, Wine-white and Wine-red, are downloaded from the UCI dataset repository[4].

4.1. Parameters setting of APC

As introduced in Section 3.3, APC requires six parameters at most in order to identify clusters with varying shapes, sizes and densities. To minimize the effort on parameter values optimization, we set certain parameters to constant values based on preliminary sensitivity tests. We first perform sensitivity tests on x_{num} using eight synthetic datasets with given ground-truth (see Table 1). The results are reported in Appendix A. Because APC obtains consistent results on all datasets for $x_{num} \in \{6, 7, \ldots, 10\}$. For the subsequent experiments included in this paper, we always set x_{num} to 10.

We also test the sensitivity of parameter k used in the k-dist graph method to identify the outliers (see Section 3.3.2). Specifically, we test two values of k: $\epsilon/3$ and $2\epsilon/3$, where ϵ denotes the size of *PL*. The results are shown in Appendix B. Because APC obtains perfect results on all datasets using either k value being tested. For the subsequent experiments included in this paper, we always set k to $2\epsilon/3$.

[3] https://github.com/mlyizhang/corepoints-clustering.git
[4] http://archive.ics.uci.edu/ml/datasets.html

Furthermore, we present the sensitivity test on the threshold value used in the distinguishing criterion of $kavg_{(j+1)} \geqslant 2kavg_j$ in the purifying algorithm from $1.4kavg_j$ to $3kavg_j$. As shown in Table 2, $2kavg$ is shown as a robust value for all the datasets that have multiple local-density levels.

Similarly, we present the sensitivity test on the threshold value used in the distinguishing criterion of $num_{dbc} > 2(\lambda + 1)$ in the MCO algorithm from $2(\lambda + 0.5)$ to $2(\lambda + 5)$. As shown in Table 3, the performance of APC is insensitive to such threshold value. Therefore, for all experiments in this paper, we use $2(\lambda + 1)$.

Based on the previous introductions of setting two parameter values to constants, APC only has four user-defined parameters (two in certain circumstances, see Section 3.3.1). In the remainder of this subsection, we present the heuristic methods that we used to fine-tune the remaining four parameters as follows.

(i) *pct*: This parameter is inherited from DPC. We can determine the value of *pct* by analyzing the decision graph. When the value of *pct* gradually increases from 0, δ values of certain data points become obviously greater than the others. In this situation, it is easier for users to find cluster centers with obviously higher δ values. As such, the corresponding *pct* value is determined. Based on our experience, $pct \in (0, 10]$ is good enough for most datasets. For each synthetic dataset studied in this paper, parameter *pct* may be taken from a range of values to obtain the best results. For example. APC always obtains best results ($ARI = 1.00, NMI = 1.00$) on the Aggregation dataset when $pct \in [3.3, 4.3]$. By following such a heuristic method, it is not difficult to find a reasonable *pct* value.

(ii) *mdelta*: This parameter is used in APC to obtain anchor points. The rational of setting the value of *mdelta* is to enable the further determination of cluster belongingness of all the outliers as they are selected as anchor points and centers to form intermediate clusters. For datasets with multiple local-density levels, if the value of *mdelta* is chosen to identify all the outliers as anchor points, the ultimate results are definitely the best. Otherwise, a few outliers will be overlooked, then the results may also be close to the best. A heuristic visual guideline when clustering datasets with different local density levels is that the setting of *mdelta* should lead to the selection of densely distributed data points with lower local densities (potentially outliers) in the decision graph as anchor points. As such, the possibility of having outliers overlooked is relatively smaller, which may lead to better results. For datasets with single local density level, *mdelta* should lead to the inclusion of all obvious cluster centers based on the DPC's assumption, i.e., data points with higher ρ and higher δ values should be selected as cluster centers.

(iii) *MinPts* and *Eps*: These two parameters are inherited from DBSCAN. The original DBSCAN paper [13] presents a parameter setting method using K-dist graph. When inflection point (the bottom of a "valley" in a plot) occurs in the K-dist plot, the corresponding distance to its kth nearest neighbour will be selected as *Eps*, and *MinPts* is set to k. In APC, we adopt

Table 2
Performance of APC (ARI values) from $1.4kavg_j$ to $3kavg_j$.

Datasets	$1.4kavg_j$	$1.6kavg_j$	$1.8kavg_j$	$2kavg_j$	$2.2kavg_j$	$2.4kavg_j$	$2.6kavg_j$	$2.8kavg_j$	$3kavg_j$
Compound	1.00	1.00	1.00	1.00	1.00	1.00	0.99	0.99	0.99
2circles	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
Pathbased	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
Sonar	0.04	0.04	0.04	0.04	0.04	0.04	0.04	0.04	0.04
German	0.08	0.08	0.08	0.08	0.08	0.08	0.08	0.08	0.08
Vehicle	0.11	0.11	0.11	0.11	0.11	0.11	0.11	0.11	0.11

Table 3
Performance of APC (ARI values) from $2(\lambda+0.5)$ to $2(\lambda+5)$.

Datasets	$2(\lambda+0.5)$	$2(\lambda+1)$	$2(\lambda+2)$	$2(\lambda+3)$	$2(\lambda+4)$	$2(\lambda+5)$
Compound	1.00	1.00	1.00	1.00	1.00	1.00
2circles	1.00	1.00	1.00	1.00	1.00	1.00
Pathbased	1.00	1.00	1.00	1.00	1.00	1.00
Sonar	0.04	0.04	0.04	0.04	0.04	0.04
German	0.08	0.08	0.08	0.08	0.08	0.08
Vehicle	0.11	0.11	0.11	0.11	0.11	0.11

Table 4
Descriptions on the parameters used by each algorithm.

Algorithm (Parameters in use)	Parameter description
K-means (k)	k: predefined number of clusters
DBSCAN (*Eps*, *MinPts*)	*Eps*: radius of underlying neighborhood *MinPts*: minimum number of data points within the neighborhood
SC (σ, k)	σ: parameter of Gaussian kernel function k: predefined number of clusters
OPTICS (k)	k: number of nearest neighbors
DPC (*pct*)	*pct*: ratio to define cutoff distance d_c
DPC-KNN-PCA (*pct*, d, k)	*pct*: same as *pct* used in DPC d: dimensionality to be reduced to by applying PCA k: number of nearest neighbors
APC (*pct*, *mdelta*) or (*pct*, *mdelta*, *Eps*, *MinPts*)	*pct*: same as *pct* used in DPC *mdelta*: threshold to identify anchor points *Eps* and *MinPts*: same as those used in DBSCAN

Table 5
Performance comparison on synthetic datasets.

Algorithms	ARI	NMI	Algorithms	ARI	NMI
Dataset Jain			**Dataset Spiral**		
K-means	0.6977	0.3181	K-means	0.3277	−0.0061
DBSCAN	**1.0000**	**1.0000**	DBSCAN	**1.0000**	**1.0000**
SC	**1.0000**	**1.0000**	SC	**1.0000**	**1.0000**
OPTICS	0.9624	0.9041	OPTICS	0.9529	0.9315
DPC	**1.0000**	**1.0000**	DPC	**1.0000**	**1.0000**
DPC-KNN-PCA	0.5692	0.5420	DPC-KNN-PCA	0.2652	0.3367
APC	**1.0000**	**1.0000**	APC	**1.0000**	**1.0000**
Dataset Flame			**Dataset Aggregation**		
K-means	0.7528	0.4880	K-means	0.8097	0.7588
DBSCAN	0.9659	0.9280	DBSCAN	0.9828	0.9749
SC	0.9769	0.9501	SC	0.9503	0.9364
OPTICS	0.9405	0.8696	OPTICS	0.7429	0.6717
DPC	**1.0000**	**1.0000**	DPC	**1.0000**	**1.0000**
DPC-KNN-PCA	**1.0000**	**1.0000**	DPC-KNN-PCA	0.9957	0.9884
APC	**1.0000**	**1.0000**	APC	**1.0000**	**1.0000**
Dataset R15			**Dataset Compound**		
K-means	0.9016	0.8938	K-means	0.6422	0.5379
DBSCAN	0.9018	0.8942	DBSCAN	0.9103	0.8774
SC	0.9175	0.9100	SC	0.6127	0.4942
OPTICS	0.9800	0.9787	OPTICS	0.8863	0.8369
DPC	**0.9928**	**0.9942**	DPC	0.6368	0.5263
DPC-KNN-PCA	**0.9928**	**0.9942**	DPC-KNN-PCA	0.5448	0.7423
APC	**0.9928**	**0.9942**	APC	**1.0000**	**1.0000**
Dataset 2circles			**Dataset Pathbased**		
K-means	0.4983	−0.0017	K-means	0.6620	0.4618
DBSCAN	**1.0000**	**1.0000**	DBSCAN	0.6288	0.4577
SC	**1.0000**	**1.0000**	SC	0.8197	0.7275
OPTICS	**1.0000**	**1.0000**	OPTICS	0.7414	0.5365
DPC	0.5043	0.0042	DPC	0.6600	0.4572
DPC-KNN-PCA	0.0012	0.0021	DPC-KNN-PCA	0.5448	0.7423
APC	**1.0000**	**1.0000**	APC	**1.0000**	**1.0000**

Table 6
Performance comparison on real-world datasets.

Algorithms	ARI	NMI	Algorithms	ARI	NMI
Dataset Ecoli			**Dataset Sonar**		
K-means	0.4070	0.6070	K-means	0.0064	0.0088
DBSCAN	0.2829	0.4118	DBSCAN	0.0093	0.0927
SC	0.4087	0.5375	SC	−0.0045	0.045
OPTICS	0.3444	**0.6148**	OPTICS	0.0029	0.036
DPC	0.4467	0.5507	DPC	0.0085	0.0105
DPC-KNN-PCA	0.1721	0.4580	DPC-KNN-PCA	−0.0046	0.0049
APC	**0.4531**	0.4630	APC	**0.0381**	**0.2329**
Dataset Wine			**Dataset Iris**		
K-means	0.3711	0.4288	K-means	0.7302	0.7582
DBSCAN	0.1978	0.3531	DBSCAN	0.5681	0.7337
SC	0.3885	**0.4327**	SC	0.7445	0.7777
OPTICS	0.2370	0.4032	OPTICS	0.5438	0.6925
DPC	0.3910	0.4308	DPC	0.6314	0.7112
DPC-KNN-PCA	0.2614	0.3336	DPC-KNN-PCA	0.7243	0.7747
APC	**0.3910**	0.4308	APC	**0.8766**	**0.8581**
Dataset Liver			**Dataset Vehicle**		
K-means	−0.0054	0.0009	K-means	0.0757	0.1001
DBSCAN	0.0274	0.1139	DBSCAN	0.0973	0.2555
SC	−0.0035	0.0021	SC	0.1103	0.1454
OPTICS	0.0015	**0.1444**	OPTICS	0.0012	0.0050
DPC	−0.0016	0.0045	DPC	0.0901	0.1335
DPC-KNN-PCA	0.0067	0.1524	DPC-KNN-PCA	0.0298	**0.3381**
APC	**0.0304**	0.0457	APC	**0.1140**	0.2443
Dataset German			**Dataset Pima**		
K-means	0.0609	0.0158	K-means	0.1046	0.0536
DBSCAN	0.0810	0.0056	DBSCAN	0.0023	0.0042
SC	0.0285	0.0021	SC	0.1040	0.0597
OPTICS	−0.0042	0.0001	OPTICS	0.0062	0.0005
DPC	0.0042	0.0027	DPC	0.0218	0.0058
DPC-KNN-PCA	0.0571	0.0237	DPC-KNN-PCA	0.0131	0.0035
APC	**0.0846**	**0.0769**	APC	**0.1507**	**0.0908**
Dataset Abalone			**Dataset Gesture**		
K-means	0.1294	0.1258	K-means	0.1897	0.2292
DBSCAN	0.0457	0.0969	DBSCAN	0.4379	0.3891
SC	0.0851	0.0991	SC	0.2422	0.2662
OPTICS	0.0006	0.0065	OPTICS	0.3081	0.5472
DPC	**0.1507**	**0.1312**	DPC	0.0652	0.1488
DPC-KNN-PCA	0.1311	0.1228	DPC-KNN-PCA	0.1400	0.2054
APC	**0.1507**	**0.1312**	APC	**0.4671**	**0.3922**
Dataset Wine-white			**Dataset Wine-red**		
K-means	0.0156	0.0338	K-means	0.002	0.0356
DBSCAN	0.0065	0.0392	DBSCAN	0.0048	0.0973
SC	0.0107	0.0264	SC	0.0057	0.0301
OPTICS	0.0031	0.0014	OPTICS	0.0013	0.0022
DPC	0.0186	0.0166	DPC	0.0486	0.0268
DPC-KNN-PCA	0.0086	0.0315	DPC-KNN-PCA	0.0100	0.0376
APC	**0.0312**	**0.1386**	APC	**0.0627**	**0.2020**

Table 7
Corresponding parameter values adopted by each clustering algorithm (the sequence of presenting the parameters values follows the sequence of introduction in Table 4).

Datasets	Parameters						
	K-means	DBSCAN	SC	OPTICS	DPC	DPC-KNN-PCA	APC
Elico	8	(0.1.2)	(0.5,8)	8	0.2	(2,2,3)	(4,0.1,0.1,2)
Sonar	2	(1,2)	(0.05,2)	12	0.1	(4,58,2)	(5,0.8,3,2)
Wine	3	(20,2)	(46,3)	8	0.1	(2,12,2)	(0.1,262,-,-)
Iris	3	(1,2)	(0.1,2)	4	0.8	(4,3,2)	(0.8,1.03,-,-)
Liver	2	(4.9,2)	(10,2)	6	6	(2,3,5)	(6,21.27,6,3)
Vehicle	4	(0.5,2)	(1,4)	10	2	(1,8,6)	(0.1,0.68,2,2)
German	2	(8.1,2)	(1,2)	6	1	(1,8,6)	(0.5,10.6,9,2)
Pima	2	(0.5,2)	(0.1,2)	4	1	(0.5,7,3)	(1,0.18,-,-)
Abalone	3	(0.1,10)	(1,3)	3	10	(2,8,8)	(10,0.9,-,-)
Gesture	5	(0.4,2)	(1,5)	5	3	(1,12,3)	(3,0.32,2,2)
Wine-white	7	(0.3,5)	(1000,7)	7	9	(0.5,5,8)	(9,1.52,-,-)
Wine-red	6	(1,3)	(1000,6)	6	9	(0.8,8,9)	(9,2.1,-,-)

Table 8
Performance comparisons on Olivetti Face dataset.

Metric	K-means (10)	DBSCAN (0.85/2)	SC(10/2)	OPTICS (10)	DPC ($d_c = 0.07$)	DPC-KNN-PCA (1/20/5)	APC (1/0.91)
ARI	0.6545	0.5918	0.6164	0.4838	0.6023	0.6790	**0.7464**
NMI	0.8213	0.7979	0.7797	0.7416	0.7802	0.8263	**0.8635**

the same procedure to obtain these two parameter values. A simple illustration is shown in Fig. 6: for Compound dataset, we plot the K-dist graph (k is set to 3 because the inflection point is much easier to be found). The inflection point is identified at the (283, 0.85) location (283 is the index of that data point). Then, *MinPts* and *Eps* are set to 3 and 0.85, respectively. For all the applicable datasets studied in this paper, we first find their respective inflection points and then fine-tune the values heuristically. This fine-tuning procedure has been applied to APC, DBSCAN and other benchmarking methods (following their respective fine-tuning procedures) to make the performance comparison on a fair basis.

4.2. Benchmarking models

For performance comparisons, we select four classical clustering methods, namely K-means, DBSCAN, SC, OPTICS and two state-of-the-art methods DPC and DPC-KNN-PCA (downloaded from https://github.com/mlyizhang/DPC-KNN-PCA.git) [28], which all have been briefly reviewed in this paper. The parameters used by all the algorithms are listed in Table 4. In terms of evaluation metrics, we adopt both adjusted rand index [42], $ARI \in [-1, 1]$ and normalized mutual information [43], $NMI \in [0, 1]$. In order to ensure the fairness of performance evaluation, we obtain the best results of all algorithms from extensive experiments by heuristically tuning all the parameters.

4.3. Experiments on synthetic datasets

Figs. 7–18 show the clustering results obtained by K-means, DBSCAN, SC, OPTICS, DPC, DPC-KNN-PCA and APC algorithms on the twelve synthetic datasets. Moreover, we summarize the results in Table 5. To better demonstrate the procedures of APC, we also show the identified gaps and local density levels (see Section 3.2) of all eight synthetic datasets with provided ground-truth (see Table 1) in Appendix C. Please note that for those four datasets without provided ground-truth, we only show the clustering results (see Figs. 15 to 18) and are not able to report the evaluation metrics in Table 5.

As shown in Table 5, DPC achieves good results on Jain, Spiral, Flame, Aggregation, R15 and T58 datasets. This is not surprising because the clusters in these datasets have only one density peak rather than multiple ones. Jain, Spiral, 2circles, T48, T58 and T710 datasets comprise clusters with uniform densities rather than varying density, so DBSCAN works well on them. For the convex datasets R15 and T58 (the silhouette of each letter is approximately convex), K-means obtains good results on them. SC obtains ideal results on only Jain, Spiral, 2circles and Flame, because each natural cluster in these datasets have close proximity distribution. OPTICS obtains good results on Jain, Spiral, Flame, R15 and 2circles, because the densities of different natural clusters are analogous. On the other hand, it is relatively more difficult to distinguish precise cluster boundaries in datasets Jain, Spiral and Flame, so OPTICS only gets sub-optimal results on them. DPC-KNN-PCA achieves good results on Flame, Aggregation, R15 and T58 because they are convex. Nonetheless, despite of the various characteristics of these datasets, APC always obtains the best clustering results in terms of both ARI and NMI as shown in Table 5.

Generally speaking, the performance of clustering algorithms may be affected by the dataset's intrinsic structure, such as convex shape or non-convex, one density peak or multiple ones, uniform densities or varying, breaking the nearest neighbor rule or obeying it, existence of connected points or not, etc. Nonetheless, APC adopts a novel approach that it assumes all the natural clusters in a dataset can be categorized into different local density levels. As shown with experimental results, APC is able to well handle all the various cluster types. Therefore, APC is shown to be able to identify clusters with arbitrary shapes, sizes, and densities.

4.4. Experiments on real-world datasets

Table 6 presents the experimental results of applying the seven afore-compared clustering algorithms on twelve UCI datasets (see Table 1). The corresponding parameter values used by all algorithms are listed in Table 7.

For Sonar, Iris, Pima, German, Abalone, Gesture, Wine-white and Wine-red (eight out of twelve datasets) datasets, APC achieves the highest ARI and NMI scores. For the other four datasets, namely Elico, Wine, Liver and Vehicle, APC only gets the highest ARI values. However, the NMI values obtained by all algorithms are close. These experimental results demonstrate that APC is able to well handle various types of real-world datasets.

4.5. Application in face recognition

In this subsection, we apply APC to the Olivetti Face dataset for face recognitions [44]. The Olivetti Face dataset is contributed by AT&T Laboratories Cambridge, which can be downloaded online[5]. This dataset is a well recognised benchmark for face recognition tasks and it consists of 400 images of 40 persons (every person has 10 images), where each image has 112×92 pixels in greyscale. Both prior studies [45] and [19] use the first 100 images to assess their algorithms, so we adopt the same approach in our study as well. The similarity between two images is computed by complex wavelet structural similarity (CW-SSIM) [44]. For benchmarking state-of-the-art model DPC, its parameter is set to the same value used in [19] that $d_c = 0.07$. As shown in Table 8, APC outperforms other models, especially the state-of-the-art models DPC and DPC-KNN-PCA in both evaluation metrics. Fig. 19 shows the face recognition results of APC, different colors represent different identified clusters.

As shown in Fig. 19, APC correctly and exclusively identifies the second, sixth, seventh and eighth persons. For the ninth and tenth persons, two images are mistakenly clustered. Three images are wrongly assigned for the fifth person. The application of APC in the face recognition again demonstrates the effectiveness of our method, which achieves better results than the state-of-the-art DPC and DPC-KNN-PCA algorithms.

5. Conclusion and future work

In this paper, we propose an effective clustering algorithm named Anchor Point based Clustering (APC). At most, APC requires

[5] http://www.cl.cam.ac.uk/research/dtg/attarchive/facedatabase.html

Table 9
Performance of APC under different x_{num} values.

Datasets	$x_{num} = 6$		$x_{num} = 7$		$x_{num} = 8$		$x_{num} = 9$		$x_{num} = 10$	
	ARI	x_{inte}	*ARI*	x_{inte}	*ARI*	x_{inte}	*ARI*	x_{inte}	*ARI*	x_{inte}
Jain	1.0000	34.5000	1.0000	29.5714	1.0000	25.8750	1.0000	23.0000	1.0000	20.7000
Spiral	1.0000	5.5000	1.0000	4.7143	1.0000	4.1250	1.0000	3.6667	1.0000	3.3000
Flame	1.0000	3.0000	1.0000	2.5714	1.0000	2.2500	1.0000	2.0000	1.0000	1.8000
Aggregation	1.0000	8.1667	1.0000	7.0000	1.0000	6.1250	1.0000	5.4444	1.0000	4.9000
R15	0.9928	5.1667	0.9928	4.4286	0.9928	3.8750	0.9928	3.4444	0.9928	3.1000
Compound	1.0000	2.5000	1.0000	2.1429	1.0000	1.8750	1.0000	1.6667	1.0000	1.5000
2circles	1.0000	2.6667	1.0000	2.2857	1.0000	2.0000	1.0000	1.7778	1.0000	1.6000
Pathbased	1.0000	0.8333	1.0000	0.7142	1.0000	0.625	1.0000	0.5556	1.0000	0.5000

the setting of four user-defined parameters, or only two in certain circumstance, which may not incur too much overhead comparing to other similar methods. APC synthesizes the advantages of two well-known methods DPC and DBSCAN. Nonetheless, we did not straightforwardly combine DPC and DBSCAN to derive APC. Instead, empowered by our proposed autonomous density analysis on the structure of the underlying dataset, APC adopts the key mechanisms of DPC and DBSCAN in a unified clustering framework to perform the clustering procedures in an autonomous manner according to the analyzed local density distribution. According to the experimental results of synthetic datasets and real-world datasets, APC achieves significantly better results than DPC and DBSCAN. Apparently, by introducing two more parameters than DPC and DBSCAN, our proposed APC method outperforms DPC and DBSCAN in almost all experiments (23 of 25 datasets).

Most importantly, APC categorizes all the data points in a dataset into one or more local density levels through a systematic approach. For example, Jain, Spiral, Flame, Aggregation and R15 datasets are identified with one local density level, while Compound, 2circles and Pathbased datasets are identified with two local density levels (see Appendix C). This novel analytical approach helps to autonomously reveal the intrinsic structure of the underlying dataset. We conduct extensive experiments on twelve synthetic datasets, twelve real-world datasets and one facial image clustering task that are all publicly available online to assess the effectiveness of APC. The experimental results show that APC performs better than all the classical and state-of-the-art benchmarking clustering algorithms.

In the future, we will apply APC to more challenging application domains, such as detecting communities in social networks and RNA sequencing analysis. Moreover, we plan to look into other methods to synergize multiple clustering algorithms such as ensemble clustering techniques, which have shown promising performance [46–48].

Declaration of competing interest

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

CRediT authorship contribution statement

Yizhang Wang: Conceptualization, Methodology, Software, Writing - original draft. **Di Wang:** Formal analysis, Validation, Writing - review & editing. **Wei Pang:** Conceptualization, Writing - review & editing. **Chunyan Miao:** Writing - review & editing, Supervision, Funding acquisition. **Ah-Hwee Tan:** Writing - review & editing, Supervision. **You Zhou:** Resources, Supervision, Writing - review & editing, Project administration, Funding acquisition.

Acknowledgment

This research is supported by the National Natural Science Foundation of China (61772227, 61572227), the Science & Technology Development Foundation of Jilin Province (20180201045GX) and the Social Science Foundation of Education Department of Jilin Province (JJKH20181315SK). This research is also supported, in part, by the National Research Foundation Singapore under its AI Singapore Programme (Award Number: AISG-GC-2019-003), the Singapore Ministry of Health under its National Innovation Challenge on Active and Confident Ageing (NIC Project No. MOH/NIC/COG04/2017), and the Joint NTU-WeBank Research Centre on Fintech, Nanyang Technological University, Singapore.

Appendix A. Performance of APC under different x_{num} values

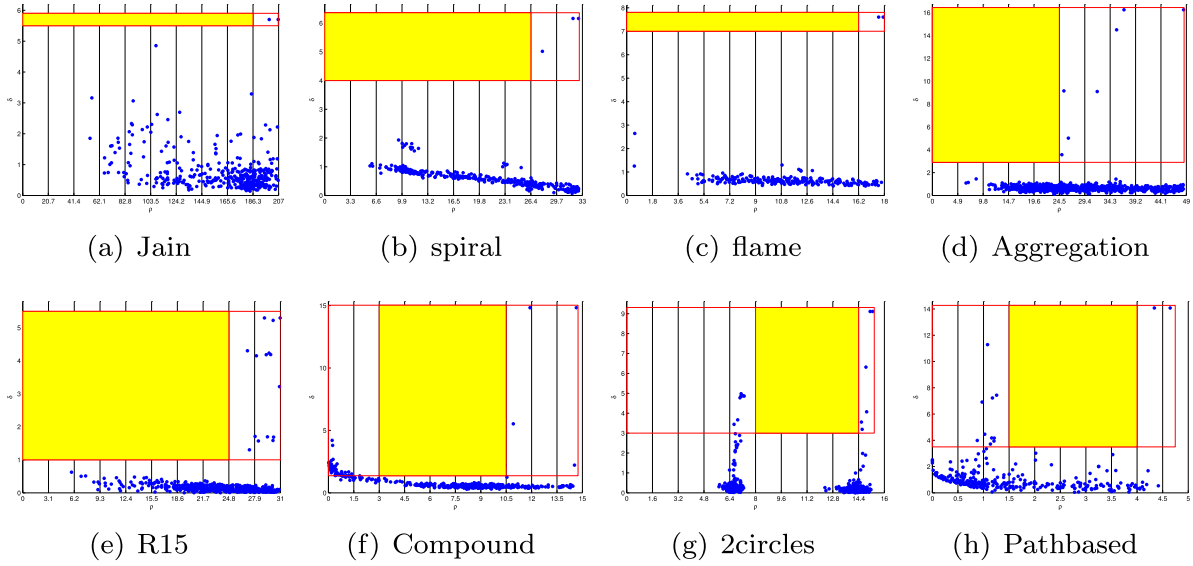
We fix the heuristically determined parameters values of *pct, mdelta, k, Eps*, and *MinPts* to test the sensitivity of x_{num} using eight synthetic datasets. The performances of APC are reported in Table 9. It is clearly shown in Table 9 that APC obtains consistent results on all datasets for $x_{num} \in \{6, 7, \ldots, 10\}$. In this paper, we always set x_{num} to 10 for all experiments.

Appendix B. Performance of APC under different k values

We fix the heuristically determined parameters values of *pct, mdelta, Eps*, and *MinPts*, and as discussed in Appendix A, we set $x_{num} = 10$. Please note that among all the synthetic datasets, only three of them require the involvement of k, namely Compound, 2circles and Pathbased, because there are more than one local density levels found in these three datasets (see Section 3.3.2). The results are shown in Table 10. It is clearly shown in Table 10 that APC obtains perfect results on all datasets using either k value. In this paper, we always set k to $2\epsilon/3$ for all experiments.

Table 10
Performance of APC under different k values.

Datasets	$k = \epsilon/3$	$k = 2\epsilon/3$
	ARI	*ARI*
Compound	1.0000	1.0000
2circles	1.0000	1.0000
Pathbased	1.0000	1.0000



(a) Jain (b) spiral (c) flame (d) Aggregation

(e) R15 (f) Compound (g) 2circles (h) Pathbased

Fig. 20. Visualization of the identified gaps and local density levels for eight synthetic datasets (those provided with ground-truth).

Appendix C. Visualization on gaps and local density levels

We show the identified gaps and local density levels of eight synthetic datasets in Fig. 20. Yellow areas in the red rectangles represent the gaps and the remaining bins in the red rectangles represent local density levels.

References

[1] M. Wang, W. Zuo, Y. Wang, An improved density peaks-based clustering method for social circle discovery in social networks, Neurocomputing 179 (2016) 219–227.

[2] L. Huang, G. Wang, Y. Wang, W. Pang, Q. Ma, A link density clustering algorithm based on automatically selecting density peaks for overlapping community detection, Int. J. Modern Phys. B 30 (24) (2016) 1650167.

[3] A. Said, R.A. Abbasi, O. Maqbool, A. Daud, N.R. Aljohani, CC-GA: a clustering coefficient based genetic algorithm for detecting communities in social networks, Appl. Soft Comput. 19 (2017) 59–70.

[4] D. Wang, A.-H. Tan, Self-regulated incremental clustering with focused preferences, in: Proceedings of International Joint Conference on Neural Networks, 2016, pp. 1297–1304.

[5] Y. Ge, S.C. Sealfon, Flowpeaks: a fast unsupervised clustering for flow cytometry data via k-means and density peak finding, Bioinformatics 28 (15) (2012) 2052–2058.

[6] S. Park, H. Zhao, Spectral clustering based on learning similarity matrix, Bioinformatics 34 (12) (2018) 2069–2076.

[7] Y. Shi, C. Otto, A.K. Jain, Face clustering: representation and pairwise constraints, IEEE Trans. Inf. Forensics Secur. 13 (7) (2017) 1626–1640.

[8] H. Zhang, S. Wang, X. Xu, T.W. Chow, Q.J. Wu, Tree2vector: learning a vectorial representation for tree-structured data, IEEE Trans. Neural Netw. Learn. Syst. (99) (2018) 1–15.

[9] S. Aghabozorgi, Y.W. Teh, Stock market co-movement assessment using a three-phase clustering method, Expert Syst. Appl. 41 (4) (2014) 1301–1314.

[10] D. Wang, Q. Chai, G.S. Ng, Bank failure prediction using an accurate and interpretable neural fuzzy inference system, AI Commun. 29 (4) (2016) 477–495.

[11] D. Wang, X. Qian, C. Quek, A.-H. Tan, C. Miao, X. Zhang, G.S. Ng, Y. Zhou, An interpretable neural fuzzy inference system for predictions of underpricing in initial public offering, Neurocomputing 319 (2018) 102–117.

[12] K. Wagstafsf, C. Cardie, S. Rogers, Constrained k-means clustering with background knowledge, in: Eighteenth International Conference on Machine Learning, 2001, pp. 577–584.

[13] M. Ester, H.-P. Kriegel, X. Xu, A density-based algorithm for discovering clusters in large spatial databases with noise, in: ACM SIGKDD Conference on Knowledge Discovery and Data Mining, 1996, pp. 226–231.

[14] P. Viswanath, R. Pinkesh, l-DBSCAN : a fast hybrid density based clustering method, in: International Conference on Pattern Recognition, 1, 2006, pp. 912–915.

[15] U. von Luxburg, A tutorial on spectral clustering, Stat. Comput. 17 (4) (2007) 395–416.

[16] A.Y. Ng, M.I. Jordan, Y. Weiss, On spectral clustering: Analysis and an algorithm, in: Advances in Neural Information Processing systems, 2002, pp. 849–856.

[17] H.-P. Kriegel, M. Pfeifle, Hierarchical density-based clustering of uncertain data, in: IEEE International Conference on Data Mining, 2005, pp. 689–692.

[18] M. Ankerst, M.M. Breunig, H.P. Kriegel, OPTICS: ordering points to identify the clustering structure, in: ACM SIGKDD Conference on Knowledge Discovery and Data Mining, 1999, pp. 49–60.

[19] A. Rodriguez, A. Laio, Clustering by fast search and find of density peaks, Science 344 (2014) 1492–1496.

[20] M. Du, S. Ding, Y. Xue, A robust density peaks clustering algorithm using fuzzy neighborhood, Int. J. Mach. Learn. Cybern. 9 (7) (2018) 1131–1140.

[21] R. Mehmood, G. Zhang, R. Bie, H. Dawood, H. Ahmad, Clustering by fast search and find of density peaks via heat diffusion, Neurocomputing 208 (2016) 210–217.

[22] V. Chandola, A. Banerjee, V. Kumar, Anomaly detection: a survey, ACM Comput Surv 41 (3) (2009) 1–15.

[23] L. Fu, E. Medico, Flame, a novel fuzzy clustering method for the analysis of dna microarray data, BMC Bioinform. 8 (1) (2007) 3.

[24] D. Huang, C.-D. Wang, J. Wu, J.-H. Lai, C.K. Kwoh, Ultra-scalable spectral clustering and ensemble clustering, IEEE Trans. Knowl. Data Eng. (2019) 1–6.

[25] D. Cai, X. Chen, Large scale spectral clustering via landmark-based sparse representation, IEEE Trans. Cybern. 45 (8) (2014) 1669–1680.

[26] H. Yu, C. Zhang, G. Wang, A tree-based incremental overlapping clustering method using the three-way decision theory, Knowl. Based Syst. 91 (2016) 189–203.

[27] J. Deng, J. Guo, Y. Wang, A novel k-medoids clustering recommendation algorithm based on probability distribution for collaborative filtering, Knowl. Based Syst. 175 (2019) 96–106.

[28] M. Du, S. Ding, H. Jia, Study on density peaks clustering based on k-nearest neighbors and principal component analysis, Knowl. Based Syst. 99 (2016) 135–145.

[29] J. Xie, H. Gao, W. Xie, X. Liu, P.W. Grant, Robust clustering by detecting density peaks and assigning points based on fuzzy weighted k-nearest neighbors, Inf. Sci. (Ny) 354 (2016) 19–40.

[30] Y. Liu, Z. Ma, F. Yu, Adaptive density peak clustering based on k-nearest neighbors with aggregating strategy, Knowl. Based Syst. 133 (2017) 208–220.

[31] M. Parmar, D. Wang, X. Zhang, A.-H. Tan, C. Miao, J. Jiang, Y. Zhou, REDPC: a residual error-based density peak clustering algorithm, Neurocomputing 348 (2019) 82–96.

[32] J. Xu, G. Wang, W. Deng, Denpehc: density peak based efficient hierarchical clustering, Inf. Sci. (Ny) 373 (2016) 200–218.

[33] R. Mehmood, S. El-Ashramand, R. Bie, H. Dawood, A. Kos, Clustering by fast search and merge of local density peaks for gene expression microarray data, Sci. Rep. 7 (2017) 45602.

[34] M. Chen, L. Li, B. Wang, J. Cheng, L. Pan, X. Chen, Effectively clustering by finding density backbone based-on knn, Pattern Recognit. 60 (2016) 486–498.

[35] Y. Ren, N. Wang, M. Li, Z. Xu, Deep density-based image clustering, 2018. arXiv preprint arXiv:1812.04287

[36] Y. Ren, U. Kamath, C. Domeniconi, Z. Xu, Parallel boosted clustering, Neurocomputing 351 (2019) 87–100.

[37] S. Wang, Y. Liu, B. Shen, MDBSCAN: multi-level density based spatial clustering of applications with noise, in: International Conference on Service Systems and Service Management, 2007, pp. 1–5.

[38] M.N. Gaonkar, K. Sawant, Autoepsdbscan: DBSCAN with eps automatic for large dataset, Int. J. Adv. Comput. Theory Eng. 2 (2) (2013) 11–16.

[39] A. Karami, R. Johansson, Choosing DBSCAN parameters automatically using differential evolution, Int. J. Comput. Appl. 91 (7) (2014) 1–11.

[40] C.J. Veenman, M.J.T. Reinders, E. Backer, A maximum variance cluster algorithm, IEEE Trans. Pattern Anal. Mach. Intell. 24 (9) (2002) 1273–1280.
[41] C.T. Zahn, Graph-theoretical methods for detecting and describing gestalt clusters, IEEE Trans. Comput. 100 (1) (1971) 68–86.
[42] N.X. Vin, J. Epps, J. Bailey, Information theoretic measures for clusterings comparison: variants, properties, normalization and correction for chance, J. Mach. Learn. Res. 11 (2010) 2837–2854.
[43] P.A. Estevez, M. Tesmer, C.A. Perez, J.M. Zurada, Normalized mutual information feature selection, IEEE Trans. Neural Netw. 20 (2) (2009) 189–201.
[44] F.S. Samaria, A.C. Harter, Parameterisation of a stochastic model for human face identification, in: Proceedings of IEEE Workshop on Applications of Computer Vision, 22, 1994, pp. 138–142.
[45] B.J. Frey, D. Dueck, Clustering by passing messages between data points., Science 315 (5814) (2007) 972–973.
[46] D. Huang, J.-H. Lai, C.-D. Wang, Robust ensemble clustering using probability trajectories, IEEE Trans. Knowl. Data Eng. 28 (5) (2015) 1312–1326.
[47] D. Huang, C.-D. Wang, J.-H. Lai, Locally weighted ensemble clustering, IEEE Trans. Cybern. 48 (5) (2017) 1460–1473.
[48] D. Huang, C.-D. Wang, H. Peng, J. Lai, C.-K. Kwoh, Enhanced ensemble clustering via fast propagation of cluster-wise similarities, IEEE Trans. Syst., Man, Cybern.: Syst. (2018) 1–13.

Yizhang Wang is a doctoral student from school of computer science and technology, Jilin University, Changchun, China. His research interests include clustering, representation learning and few-shot learning.

Di Wang received the B.Eng. degree in Computer Engineering and the Ph.D. degree in Computer Science from Nanyang Technological University, Singapore (NTU), in 2003 and 2014, respectively. He is currently working as a Senior Research Fellow and Research Manager in the Joint NTU-UBC Research Centre of Excellence in Active Living for the Elderly (LILY), NTU, Singapore. His research interests include computational intelligence, decision support systems, computational neuroscience, autonomous agents, affective computing, ubiquitous computing, etc.

Wei Pang received his Ph.D. degree in computing science from the University of Aberdeen in 2009. He is currently an Associate Professor at Heriot-Watt University, Edinburgh, UK, and he is also an Honorary Senior Lecturer at University of Aberdeen. He has authored over 100 papers, including over 40 journal papers.. His research interests include bio-inspired computing, data mining, machine learning, and qualitative reasoning.

Chunyan Miao received the B.S. degree from Shandong University, Jinan, China, in 1988, and the M.S. and Ph.D. degrees from Nanyang Technological University (NTU), Singapore, in 1998 and 2003, respectively. She is currently a Professor and the Chair of the School of Computer Science and Engineering, NTU, the Director of the Joint NTU-UBC Research Centre of Excellence in Active Living for the Elderly, and the Director of the Alibaba-NTU Singapore Joint Research Institute. Her current research interests focus on humanized artificial intelligence, which includes infusing intelligent agents into interactive new media (virtual, mixed, mobile, and pervasive media) to create novel experiences and dimensions in game design, interactive narrative, and other real world agent systems.

Ah-Hwee Tan received the B.Sc. (Hons.) and M.Sc. degrees in computer and information science from the National University of Singapore and the Ph.D. degree in cognitive and neural systems from Boston University, USA. He is currently a Professor of Computer Science and the Associate Chair (Research) of the School of Computer Science and Engineering at Nanyang Technological University (NTU). Prior to joining NTU, he was a Research Manager with the Institute for Infocom Research, Agency for Science, Technology and Research (A*STAR), Singapore, spearheading the Text Mining and Intelligent Agents research programs. His current research interests include cognitive and neural systems, brain inspired intelligent agents, machine learning, knowledge discovery, and text mining.

You Zhou received his bachelor and Ph.D. degree from Jilin University in 2002 and 2008, respectively. He is now a professor of the College of Computer Science and Technology at Jilin University, Changchun, China. His research interests include Machine Learning, Pattern Recognition and Bioinformatics.