Neurocomputing 329 (2019) 348–358

Contents lists available at ScienceDirect

Neurocomputing

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

Identifying cluster centroids from decision graph automatically using a statistical outlier detection method

Huanqian Yan, Lei Wang, Yonggang Lu*

School of Information Science and Engineering, Lanzhou University, Lanzhou, Gansu 730000, China

ARTICLE INFO

Article history:
Received 8 May 2018
Revised 25 September 2018
Accepted 28 October 2018
Available online 5 November 2018

Communicated by Dr. Haijun Zhang

Keywords:
Clustering
Outlier detection
Decision graph
Centroid identification
Image segmentation

ABSTRACT

Cluster centroid identification is a crucial step for many clustering methods. Recently, Rodriguez and Laio have proposed an effective density-based clustering method called Density Peak Clustering (DPC), in which the density value of each data point and the minimum distance from the points with higher density values are used to identify cluster centroids from the decision graph. However, there is still a lack of automatic methods for the identification of cluster centroids from the decision graph. In this work, a novel statistical outlier detection method is designed to identify cluster centroids automatically from the decision graph, so that the number of clusters is also automatically determined. In the proposed method, one-dimensional probability density functions at specific density values in the decision graph are estimated using two-dimensional Gaussian kernel functions. Then the cluster centroids are identified automatically as outliers in the decision graph using expectation values and standard deviations computed at specific density values. Experiments on several synthetic and real-world datasets show the superiority of the proposed method in centroid identification from the datasets with various distributions and dimensionalities. Furthermore, it is also shown that the proposed method can be effectively applied to image segmentation.

1. Introduction

Clustering is the process of grouping a set of data objects into multiple groups which are called clusters, so that objects within the same cluster have higher similarity than the objects from different clusters. Dissimilarities or similarities can be assessed based on attribute values describing the objects using certain distance metric [1]. Clustering is an important technique for exploratory data analysis, and has been studied for many years [2,3]. It has been shown to be useful in many practical domains such as astronomy, biology, data classification and image processing [4], *etc.*

Clustering is generally considered as a difficult problem because the optimal number of clusters cannot be easily determined and clusters may have different distributions, shapes and sizes [5]. It has been shown that clustering is a nonconvex, discrete optimization problem. Due to the existence of many local minima, it is typically difficult to find a globally minimal solution without trying many different initial conditions [6]. Although there are many clustering algorithms like hierarchical clustering algorithms [7–9] and partitional clustering algorithms [10–13], most of them are not generic enough and can only be used for solving particular clustering problems.

Density-based clustering algorithms separate dataset into clusters which have high density values at the cluster centers and are separated by low density value points. These algorithms can detect clusters with arbitrary shapes or sizes, and they are also not sensitive to noise. The most representative density-based clustering methods include DBSCAN [14], MEANSHIFT [15,16], OPTICS [17], and DENCLUE [17]. Those methods has many advantages, but most of them have a same drawback that the parameter setting is not a straightforward task.

An excellent density-based clustering method published in Science in 2014 is proposed by Rodriguez and Laio [18]. The method is called Density Peak Clustering (DPC). It is based on the simple idea that a cluster centroid has a higher density value than its neighbors and is far away from the other objects with higher density values. After the cluster centroids are selected by users, the algorithm can generate clusters by assigning a data point to the cluster that contains its nearest neighbor with higher density value. A heuristic method is designed to select cluster centroids from a two dimensional decision graph whose two axes are the density value and the minimum distance from the points with higher density values. However, it is still not an automatic method because the cluster centroids in the decision graph have to be

* Corresponding author.
E-mail addresses: yanhq15@lzu.edu.cn (H. Yan), leiwang16@lzu.edu.cn (L. Wang), ylu@lzu.edu.cn (Y. Lu).

https://doi.org/10.1016/j.neucom.2018.10.067
0925-2312/

manually decided. For the threshold-based method suggested in the paper, the optimal threshold parameter is usually difficult to be determined for different datasets.

To address these issues, a novel clustering method called Automatic Density Peak Clustering (ADPC) is proposed in the paper. The basic idea is that cluster centroids can be regarded as outliers with anomalously large minimum distance values in the decision graph. So, a newly designed statistical outlier detection method is used to identify the cluster centroids automatically from the decision graph. The proposed clustering method can identify clusters with arbitrary shapes or sizes and can determine the number of clusters automatically.

The initial results of the method have been reported in a conference paper [19]. Here, detailed explanation of the density estimation, additional experimental results, and more analysis of the proposed method are included. The rest of the paper is organized as follows. The original DPC method is introduced in Section 2. The proposed ADPC method is described in Section 3. The experimental results and analysis are presented in Section 4. Conclusions are drawn in Section 5.

2. The DPC method

For the DPC method, the centroids need to be selected from the decision graph first. During the process, two important indicators are considered in the decision graph for each data point i: local density p_i, and the minimum distance d_i from points of higher density values. After the cluster centroids are selected, the DPC method iteratively assigns data points to the cluster which contains their nearest neighbors with higher density values.

2.1. Computation of the density value and the minimum distance

The local density p_i of a point i is defined as

$$p_i = \Sigma_j \chi(r_{ij} - r_c), \tag{1}$$

where $\chi(x)$ is a kernel function, r_{ij} is the distance between point i and point j, and r_c is the cutoff distance threshold. In the DPC method, r_c is a parameter which needs to be determined manually. In our experiments, the Gaussian kernel function is used. So the local density p_i is defined as:

$$p_i = \Sigma_j e^{-r_{ij}^2/2r_c^2} \tag{2}$$

The minimum distance d_i of point i is measured by computing the minimum distance between the point i and any other points of higher density values:

$$d_i = \begin{cases} min_j(r_{ij}) & if\ \exists j\ s.t.\ p_j > p_i \\ max_j(r_{ij}) & otherwise \end{cases} \tag{3}$$

Usually, the value d_i is much larger than the typical distances between nearest neighbours if the p_i of point i is a local or global maximum density value. This observation, which is the core of the DPC algorithm, is illustrated by an example in Fig. 1. Fig. 1A shows 30 points from two normal distributions. Fig. 1B is the decision graph which shows the plot of d_i as a function of p_i for each point. From the decision graph, the two points having high local density values and large minimum distances can be easily identified. The two points are identified as cluster centroids, which are shown as filled triangle or square in both Fig. 1A and 1B.

2.2. The threshold-based method for identifying centroids

As shown in Fig. 1B, the cluster centroids are usually the points that have large d_i values and relatively high p_i values. A simple threshold-based method suggested by Rodriguez and Laio [18] selects the cluster centroids using the following formula:

$$\gamma_i = p_i \times d_i > TH_\gamma \tag{4}$$

where the threshold parameter TH_γ has to be decided by users.

The process of identifying cluster centroids using the threshold-based method from the decision graph for the dataset shown in Fig. 1A is shown in Fig. 2. In Fig. 2, the horizontal axis is the point density value p_i, the vertical axis is the γ_i value, and it can be seen that the centroids are points which have large γ_i values. The selection of the parameter TH_γ is important since it may directly affect the eventual clustering results. For some simple datasets, such as these containing non-overlapping clusters, the TH_γ value can be selected easily by users. But for complex datasets, it is not an easy task to select the optimal TH_γ values.

3. Identifying cluster centroids from the decision graph automatically

Two drawbacks of the threshold-based method are that it does not use the distribution information of the points in the decision graph and the parameter TH_γ can not be easily determined for different datasets. To deal with the above drawbacks, a novel statistical outlier detection method for selecting the cluster centroids is designed.

3.1. Concept of statistical outlier detection

Statistical outlier detection (also known as model-based outlier detection) makes assumptions that data objects satisfy a statistical or stochastic distribution model, and that some data points not following this model are outliers. The general idea behind the statistical method for outlier detection is to learn a generative model fitting the given dataset, and then identify those objects in low-probability regions of the model as outliers. In general, statistical methods for outlier detection can be divided into two major categories: parametric methods and nonparametric methods, according to how the models are specified and learned. Nonparametric methods do not assume priori statistical models and are more adaptive than parametric methods. Examples of nonparametric methods include histogram and kernel density estimation. In this work, a special nonparametric gaussian kernel density estimation method is designed for identifying the cluster centroids.

3.2. A special statistical method for identifying cluster centroids

The proposed method is developed based on the following observation: the value d_i is much larger than the typical distances between nearest neighbors if the point i has a local or global maximum density value. Thus, an important feature for identifying a cluster centroid is that its d_i value is anomalously large. So, in our method, cluster centroids are identified using a specially designed outlier detection method which contains mainly three steps:

- Firstly, the probability density $\rho_y(p_i, y)$ in the decision graph at a specific density value p_i and an arbitrary distance value y is estimated;
- Secondly, the expectation value and the variance of the distance y are computed at the specific p_i value using the probability density $\rho_y(p_i, y)$;
- Thirdly, the cluster centroids are identified using the expectation value and the variance of the distance y at the specific p_i value.

Two-dimensional Gaussian kernel functions are used to estimate the probability density at the specific p_i in the decision

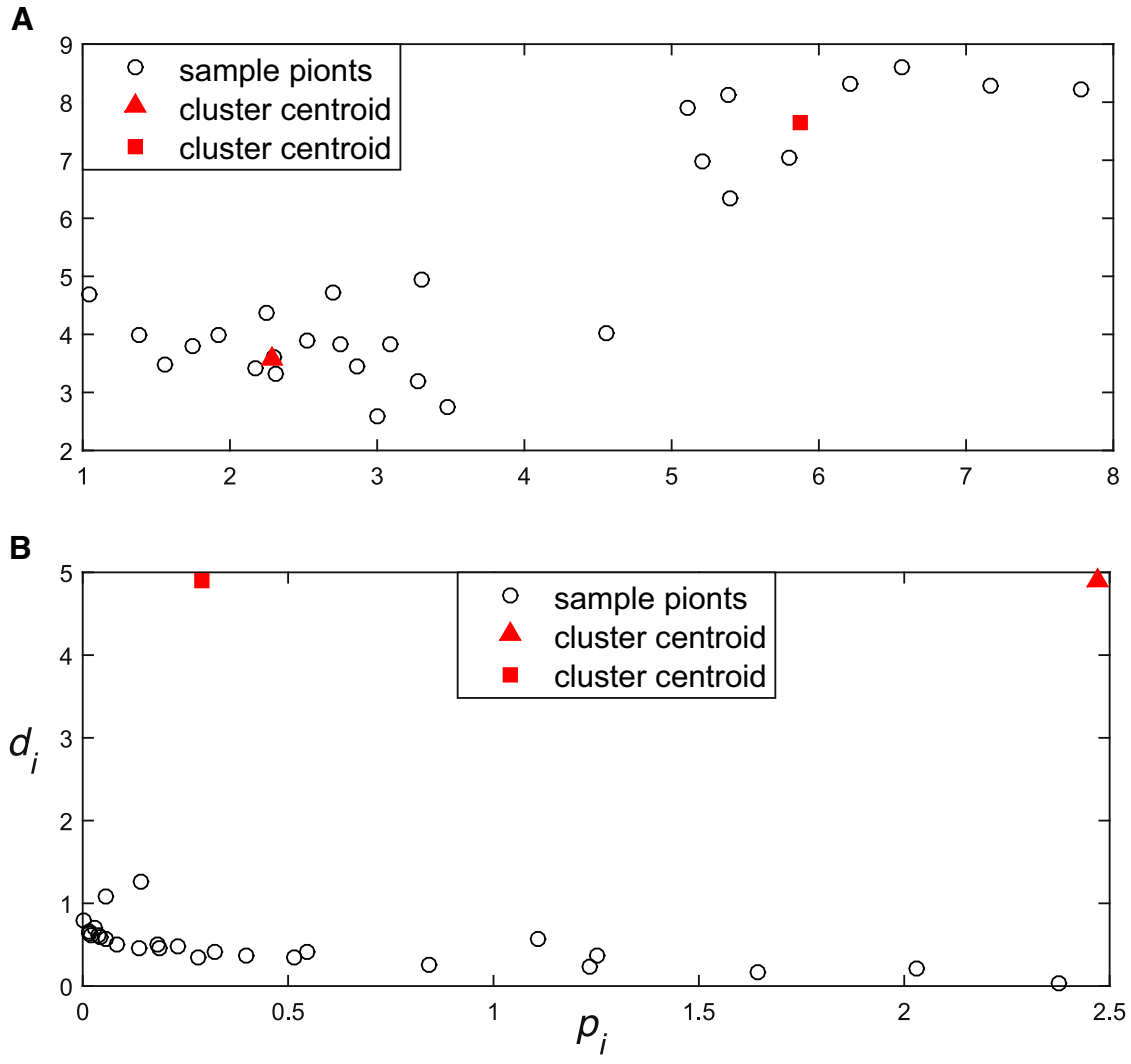

Fig. 1. (A) A data distribution in 2D space. (B) Decision graph for the data in (A).

graph, which is given by:

$$\rho_y(p_i, y) = \frac{\sum_{j=1, j\neq i}^{N} \frac{1}{2\pi ab} e^{-\frac{1}{2}(((p_j-p_i)/a)^2+((d_j-y)/b)^2)}}{\sum_{z=1, z\neq i}^{N} \frac{1}{\sqrt{2\pi}a} e^{-\frac{1}{2}(\frac{p_z-p_i}{a})^2}}, \tag{5}$$

where N is the total number of the data points, and a and b are the 2D kernel widths. The denominator is a normalization factor which is introduced to ensure that $\int_{-\infty}^{+\infty} \rho_y(p_i, y)dy = 1$, so that $\rho_y(p_i, y)$ works as a one dimensional probability density function of y.

The selection of the values for the 2D kernel widths a and b are important. It is found that a and b can be estimated using the standard deviations of p_i and d_i of all the data points:

$$\begin{cases} a = \alpha \times \sigma_{p_i}, & 0 < \alpha < 1 \\ b = \beta \times \sigma_{d_i}, & 0 < \beta < 1 \end{cases} \tag{6}$$

The selection of the parameters α and β will be discussed in Section 4.2.

Using the probability density defined in (5), the expectation value and the variance of the distance y at the specific p_i can be computed using:

$$\mu_y(p_i) = \int_{-\infty}^{+\infty} \rho_y(p_i, y) y dy \tag{7}$$

$$\sigma_y^2(p_i) = \int_{-\infty}^{+\infty} \rho_y(p_i, y)(y - \mu_y(p_i))^2 dy \tag{8}$$

By substituting (5) into (7) and let $v = \frac{y-d_i}{b}$, it follows that:

$$\begin{aligned} \mu_y(p_i) &= \frac{\frac{1}{\sqrt{2\pi}b}}{\sum_{z=1, z\neq i}^{N} e^{-\frac{1}{2}(\frac{p_z-p_i}{a})^2}} \sum_{j=1, j\neq i}^{N} \int_{-\infty}^{+\infty} y \\ &\quad \times e^{-\frac{1}{2}(\frac{y-d_i}{b})^2} \times e^{-\frac{1}{2}(\frac{p_j-p_i}{a})^2} dy \\ &= \frac{\frac{1}{\sqrt{2\pi}b} \sum_{j=1, j\neq i}^{N} e^{-\frac{1}{2}(\frac{p_j-p_i}{a})^2}}{\sum_{z=1, z\neq i}^{N} e^{-\frac{1}{2}(\frac{p_z-p_i}{a})^2}} \\ &\quad \times \sum_{j=1, j\neq i}^{N} \int_{-\infty}^{+\infty} (d_j b e^{-\frac{1}{2}v^2} + v b^2 e^{-\frac{1}{2}v^2}) dv \\ &= \frac{\frac{1}{\sqrt{2\pi}b} \sum_{j=1, j\neq i}^{N} e^{-\frac{1}{2}(\frac{p_j-p_i}{a})^2}}{\sum_{z=1, z\neq i}^{N} e^{-\frac{1}{2}(\frac{p_z-p_i}{a})^2}} \sum_{j=1, j\neq i}^{N} \sqrt{2\pi} d_j b \\ &= \frac{\sum_{j=1, j\neq i}^{N} d_j \times e^{-\frac{1}{2}\times\left(\frac{p_j-p_i}{a}\right)^2}}{\sum_{z=1, z\neq i}^{N} e^{-\frac{1}{2}\times\left(\frac{p_z-p_i}{a}\right)^2}} \end{aligned}$$

So, the expectation value is:

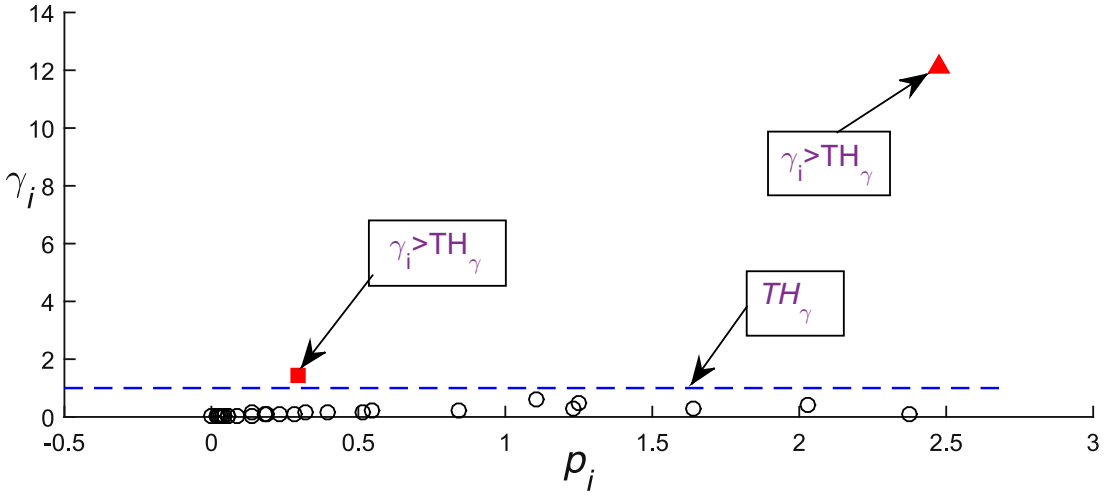

Fig. 2. The threshold-based method for identifying the cluster centroids.

$$\mu_y(p_i) = \frac{\sum_{j=1,j\neq i}^{N} d_j \times e^{-\frac{1}{2}\times\left(\frac{p_j-p_i}{a}\right)^2}}{\sum_{z=1,z\neq i}^{N} e^{-\frac{1}{2}\times\left(\frac{p_z-p_i}{a}\right)^2}} \tag{9}$$

By substituting (5) into (8) and let $t = \frac{y-d_i}{b}$, it follows that:

$$\begin{aligned}
\sigma_y^2 &= \frac{\frac{1}{\sqrt{2\pi}b}\sum_{j=1,j\neq i}^{N}\int_{-\infty}^{+\infty}(y-\mu_p(p_i))^2 e^{-\frac{1}{2}(((p_j-p_i)/a)^2+((d_j-y)/b)^2)}}{\sum_{z=1,z\neq i}^{N} e^{-\frac{1}{2}(\frac{p_z-p_i}{a})^2}}dy \\
&= \frac{\frac{1}{\sqrt{2\pi}b}\sum_{j=1,j\neq i}^{N} e^{-\frac{1}{2}(\frac{p_j-p_i}{a})^2}\int_{-\infty}^{+\infty} b(tb+d_j-\mu_y(p_i))^2 e^{-\frac{1}{2}t^2}}{\sum_{z=1,z\neq i}^{N} e^{-\frac{1}{2}(\frac{p_z-p_i}{a})^2}}dt \\
&= \frac{\frac{1}{\sqrt{2\pi}b}\sum_{j=1,j\neq i}^{N} e^{-\frac{1}{2}(\frac{p_j-p_i}{a})^2}\int_{-\infty}^{+\infty} b((tb)^2+(d_j-\mu_y(p_i))^2) e^{-\frac{1}{2}t^2}}{\sum_{z=1,z\neq i}^{N} e^{-\frac{1}{2}(\frac{p_z-p_i}{a})^2}}dt \\
&= \frac{\frac{1}{\sqrt{2\pi}}}{\sum_{z=1,z\neq i}^{N} e^{-\frac{1}{2}(\frac{p_z-p_i}{a})^2}}\sum_{j=1,j\neq i}^{N} e^{-\frac{1}{2}(\frac{p_j-p_i}{a})^2}\left[-tb^2e^{-\frac{1}{2}t^2}|_{-\infty}^{+\infty}\right. \\
&\quad \left.+\int_{-\infty}^{+\infty} b^2 e^{-\frac{1}{2}t^2}dt + \sqrt{2\pi}(d_j-\mu_y(p_i))^2\right] \\
&= \frac{\frac{1}{\sqrt{2\pi}}}{\sum_{z=1,z\neq i}^{N} e^{-\frac{1}{2}(\frac{p_z-p_i}{a})^2}}\sum_{j=1,j\neq i}^{N}\sqrt{2\pi}(b^2+(d_j-\mu_y(p_i))^2)e^{-\frac{1}{2}(\frac{p_j-p_i}{a})^2} \\
&= \frac{\sum_{j=1,j\neq i}^{N}\left[b^2+\left(d_j-\mu_y(p_i)\right)^2\right]e^{-\frac{1}{2}\times\left(\frac{p_j-p_i}{a}\right)^2}}{\sum_{z=1,z\neq i}^{N} e^{-\frac{1}{2}\times\left(\frac{p_z-p_i}{a}\right)^2}}
\end{aligned}$$

So, the variance is:

$$\sigma_y^2(p_i) = \frac{\sum_{j=1,j\neq i}^{N}\left[b^2+\left(d_j-\mu_y(p_i)\right)^2\right]e^{-\frac{1}{2}\times\left(\frac{p_j-p_i}{a}\right)^2}}{\sum_{z=1,z\neq i}^{N} e^{-\frac{1}{2}\times\left(\frac{p_z-p_i}{a}\right)^2}} \tag{10}$$

Using (9) and (10), the expectation value and the variance at the specific p_i can be easily computed using the summation instead of the integration in (7) and (8). Then, the outliers are identified using the following threshold:

$$TH_d(p_i) = \mu_y(p_i) + 3\times\sigma_y(p_i) \tag{11}$$

For any point i, if its minimum distance $d_i > TH_d(p_i)$, it is treated as an outlier, and thus is identified as a cluster centroid in our experiments. The identified the cluster centroids and the clustering result using the proposed method are shown in Fig. 3, where the data is the same as used in Fig. 1A. Using the threshold defined in (11), two points represented as a filled triangle and a filled square are identified as the cluster centroids.

3.3. An automatic density peak clustering

An automatic density-based clustering method called ADPC can be produced using the proposed cluster centroid identification method, which is summarized as follows:

- *Step 1:* Compute the distance matrix r_{ij} for all data points.
- *Step 2:* Compute local density p_i of a point i according to formula (2);
- *Step 3:* Compute the minimum distance d_i of point i according to formula (3);
- *Step 4:* Select cluster centroids using the proposed statistical method according to formula (11);
- *Step 5:* Assign each remaining data point to the nearest point which has bigger density value;

The complexity for computing the distance matrix is $O(N^2)$. The complexity for calculating the local density p_i and the minimum distance d_i of point i is $O(N^2)$. The proposed cluster centroid identification method also needs $O(N^2)$ operations. The final label assignment procedure for each point only needs $O(N)$ operations. So the total time complexity of the proposed ADPC method is $O(N^2)$.

4. Experimental results

There are 6 synthetic datasets and 8 real-world datasets used in the experiments. Two synthetic datasets, called Dataset A and Dataset B generated by ourselves, are shown in Fig. 4, where different colors represent different classes. Dataset A contains three 2D normal distributions with the same size (n=200, σ=1). Dataset B contains four 2D normal distributions with different sizes: (n_1=100, σ_1=2), (n_2=200, σ_2=3), (n_3=300, σ_3=4), (n_4=400, σ_4=2). The other 4 synthetic datasets, including Aggregation, Flame, Spiral, and R15, are downloaded from the internet [20], which are also shown in Fig. 4. Aggregation is composed of seven clusters with different sizes and shapes, which contains 788 points; Flame is composed of two clusters with different sizes and shapes, which contains 240 points; Spiral is composed of three spiral clusters which contains 312 points; R15 is composed of fifteen clusters with different sizes, which contains 600 points. The details of the 8 real-world datasets from UCI [21], including Ecoli, EEG Eye State, Glass Identification, Liver Disorder, Wine quality_red, QSAR biodegradation, Statlog (Shuttle) and Banknote authentication, are shown in Table 1.

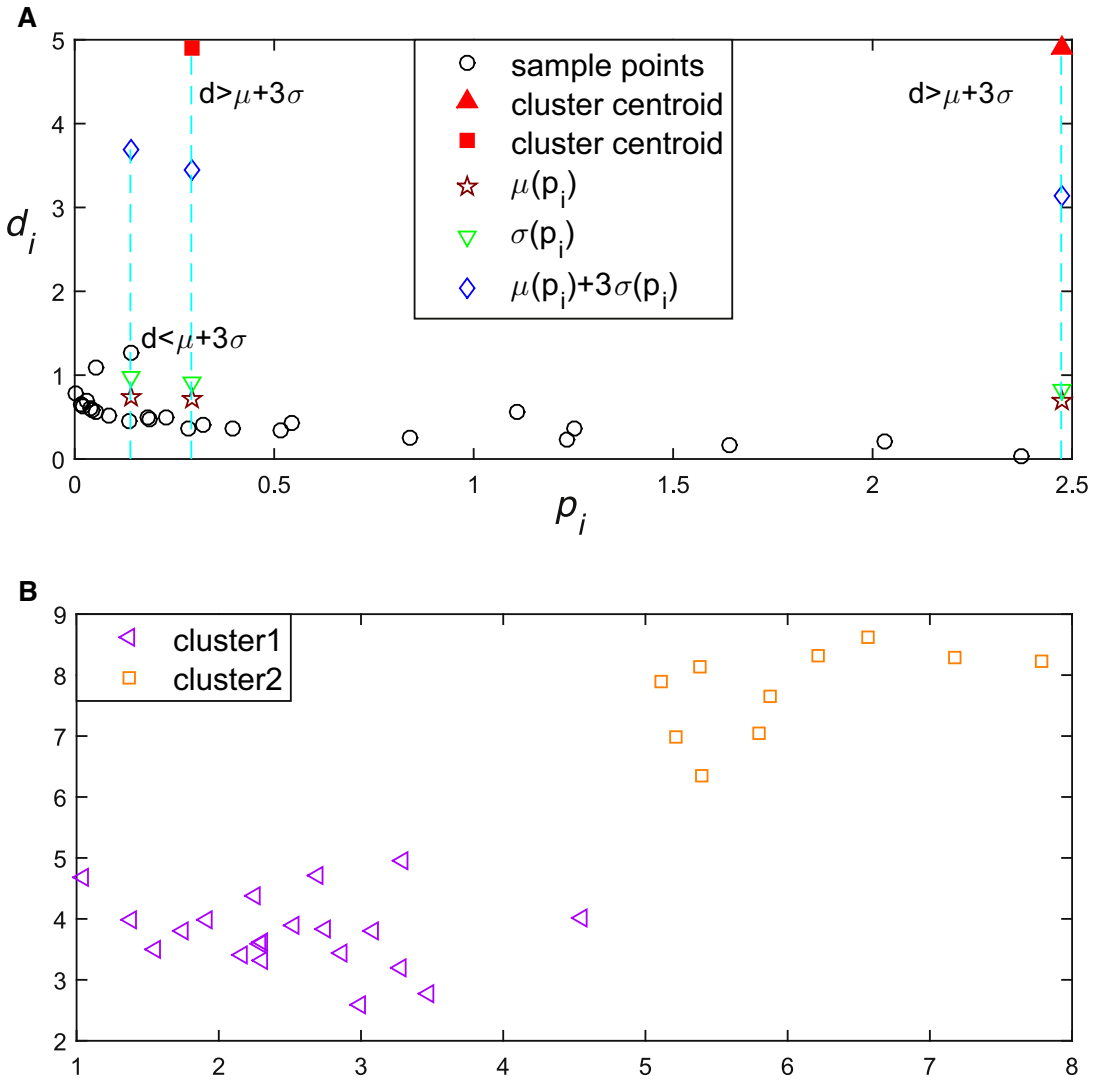

Fig. 3. (A) The process of identifying cluster centroids in the decision graph. (B) The clustering result in which different colors correspond to different clusters.

Table 1
Details of the real-world datasets used in the experiments.

Dataset	# Points	# Features	# Clusters
Ecoli	336	8	8
EEG Eye State	14,980	14	2
Glass Identification	214	9	7
Liver Disorder	345	7	2
Wine quality-red	1599	12	6
QSAR biodegradation	1055	41	2
Statlog (Shuttle)	58,000	9	7
Banknote authentication	1372	4	2

4.1. Evaluation criterion

Because for all the datasets, ground truth cluster labels are available, the F1-Measure (F1-score) [22] is used to evaluate the clustering results. which is defined as:

$$F1-score=\frac{2\times Recall(L(e),C(e^{'}))\times Precision(L(e),C(e^{'}))}{Recall(L(e),C(e^{'})+Precision(L(e),C(e^{'}))}, \tag{12}$$

where $L(e)$ is the cluster assigned to e by the clustering algorithm and $C(e)$ is the cluster assigned to e by the ground truth. To compute the Recall and the Precision, the extend BCubed Recall and BCubed Precision values [23] are used:

$$\begin{cases} ExtenndBCubed\_Recall=Avg_e[Avg_{e^{'},L(e)\cap L(e^{'})}[Mul\_recall(e,e^{'})]] \\ Mul\_recall(e,e^{'})=\dfrac{Min(|C(e)\cap C(e^{'})|,|L(e)\cap L(e^{'})|)}{|L(e)\cap L(e^{'})|} \end{cases} \tag{13}$$

$$\begin{cases} ExtendBCubed\_Precision=Avg_e[Avg_{e^{'},C(e)\cap C(e^{'})}[Mul\_precision(e,e^{'})]] \\ Mul\_precision(e,e^{'})=\dfrac{Min(|C(e)\cap C(e^{'})|,|L(e)\cap L(e^{'})|)}{|C(e)\cap C(e^{'})|} \end{cases} \tag{14}$$

A higher value of the F1-score indicates a greater similarity between the clustering result and the ground truth.

4.2. Parameter selection

In the DPC method, the parameter r_c must be determined before computing the density values. It can be chosen under the condition that the average number of neighbors is around 1% to 2% of the total number of the points, as suggested by Rodriguez and Laio

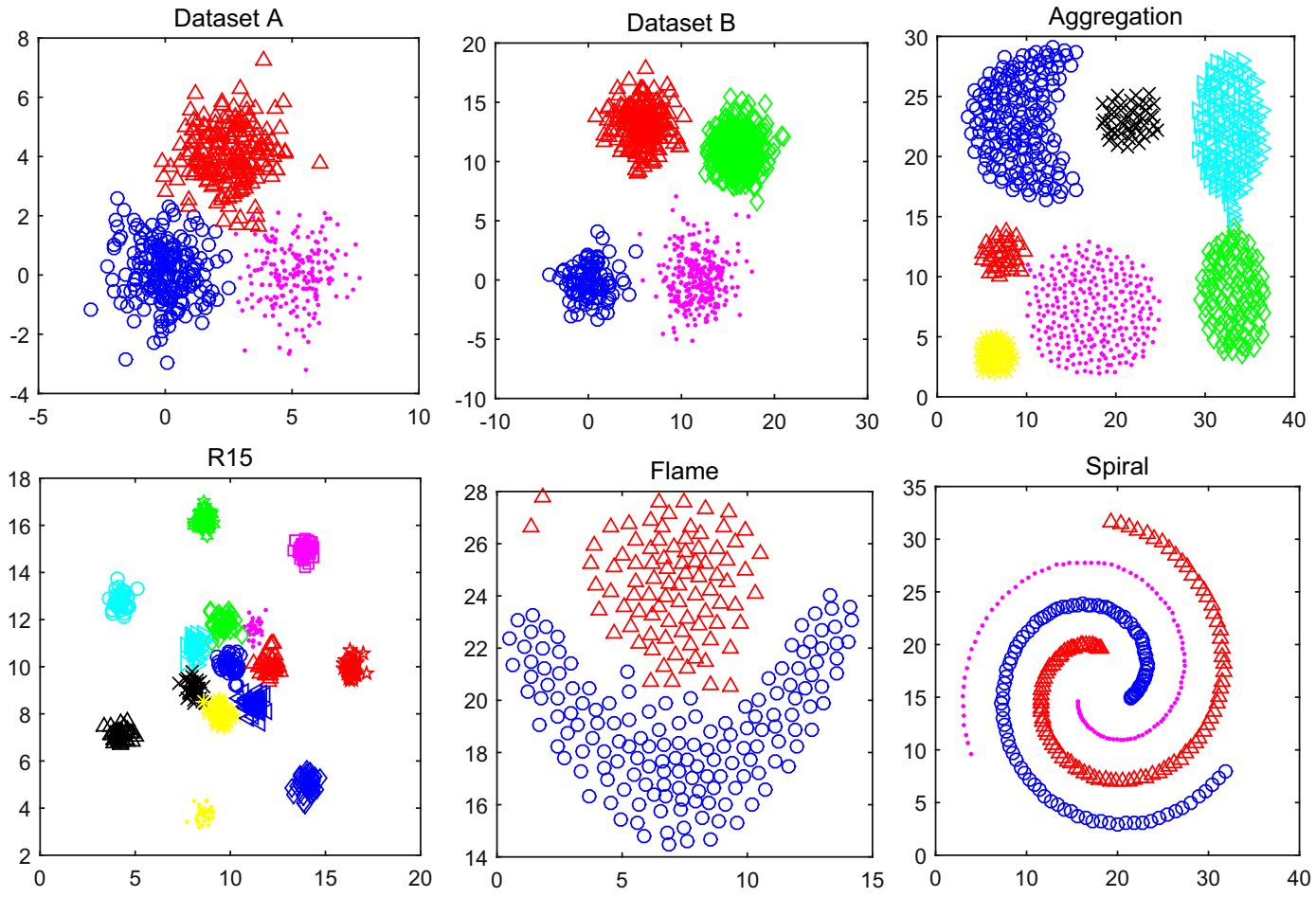

Fig. 4. The synthetic datasets: Dataset A, Dataset B, Aggregation, R15, Flame and Spiral.

[18]. In our experiments, it is found that using 4% can produce better results. So, the parameter r_c is determined with the condition that the average number of neighbors is around 4% of the total number of the points in our experiments.

In our method, the parameters α and β defined in (6) have to be determined. To decide the value of α, we first set $\beta = 0.5$, then different values of α are used to compute TH_d and identify the cluster centroids. Three datasets are used for the parameter selection, which include two UCI datasets (Iris, Seeds) [21] and a synthetic dataset (Dataset B). The clustering results at different α values for the three datasets are shown in Fig. 5A. It can be seen from Fig. 5A that the clustering results are not sensitive to the selection of the parameter α. So, in our experiments, the parameter α is set to 0.5. To determine the parameter β, the parameter α is set to 0.5, and different values of β are used in our experiments. The clustering results produced with different β values are shown in Fig. 5B. It is found that good clustering results with F1-score $>$ 0.7 can be produced within a relatively wide range of the parameter β, while $\beta = 0.5$ is a relatively good choice. So, the parameter $\beta = 0.5$ is selected in our experiments.

4.3. Comparison of the clustering results

In order to evaluate the statistical-based centroid identification in the proposed ADPC method, it is compared with the simple threshold-based DPC method proposed by Rodriguez and Laio [18]. First, the percentile-based method is used to select the centroids for DPC, the $(100\text{-}f)th$ percentile value from the set $\{\gamma_i \mid \gamma_i = p_i \times d_i, 1 \le i \le N\}$ is used to determine the threshold TH_γ. The three datasets, including Iris, Seeds and Dataset B, are used in the experiments. The clustering results under different f values for the three datasets are shown in Fig. 6. From Fig. 6, it can be found that $f = 1$ is a good choice. So the $99th$ percentile from the set $\{\gamma_i \mid \gamma_i = p_i \times d_i, 1 \le i \le N\}$ is used as the threshold TH_γ for the percentile-based method.

The comparison of the F1-scores of the clustering results for the percentile-based DPC method and the proposed ADPC method are shown in Table 2 for different datasets. As shown in Table 2,

Table 2
The F1-scores and the number of clusters produced by the percentile-based DPC method and the ADPC method on 14 different datasets.

Dataset	Percentile-based method		ADPC	
	F1-score	#Clusters	F1-score	#Clusters
Dataset A	0.853	6	**0.964**	3
Dataset B	0.824	10	**0.995**	7
Aggregation	0.854	8	**0.960**	6
Flame	0.882	3	**0.992**	3
R15	0.571	6	**0.993**	15
Spiral	0.914	4	**1**	3
Ecoli	0.590	4	**0.644**	5
EEG Eye State	0.133	150	**0.671**	4
Glass Identification	0.487	3	**0.510**	5
Liver Disorders	0.521	4	**0.669**	9
Wine quality-red	0.382	16	**0.510**	9
QSAR biodegradation	0.315	11	**0.711**	4
Statlog (Shuttle)	0.195	580	**0.784**	85
Banknote authentication	**0.466**	14	0.443	35

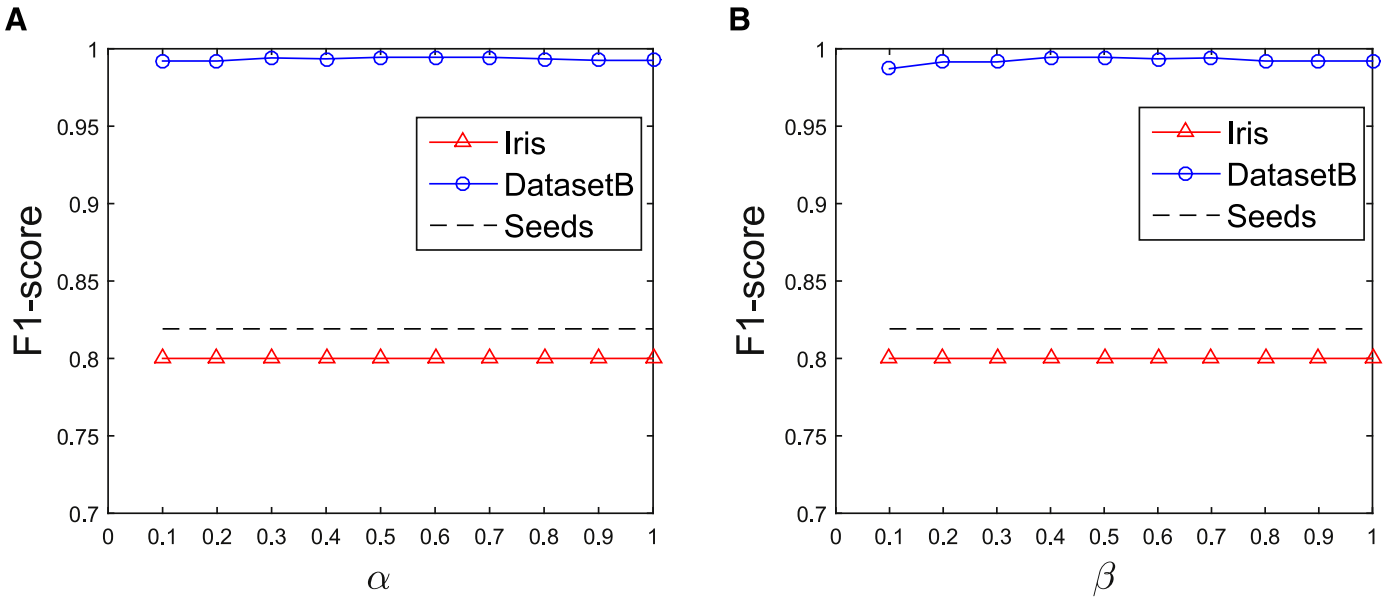

Fig. 5. The F1-scores of the clustering results produced with (A) different α values and (B) different β values.

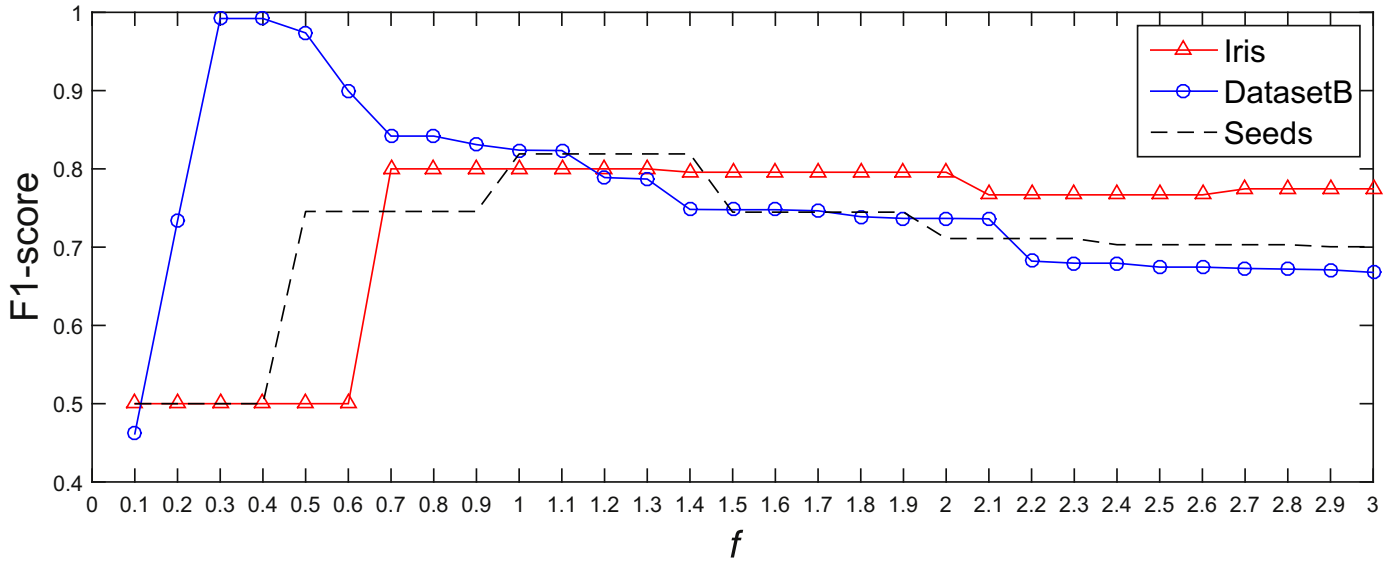

Fig. 6. The F1-score of the clustering results produced with different f values.

compared with the percentile-based method, the ADPC method has produced better results for 13 out of the total 14 datasets, and has produced worse result for only 1 dataset (the Banknote authentication). To analyze the results more strictly, a one-sided paired t-test is done with a hypothesis that the percentile-based DPC method produces larger or similar F1-scores as the ADPC method. The hypothesis is rejected at the 5% significance level with a p-value as small as 0.00093. The very small p-value indicates that the ADPC method produces significantly better results than the percentile-based method.

The clustering results of the 6 synthetic datasets are also shown in Fig. 7 where different combinations of colors and patterns represent different clusters. For all the synthetic datasets, compared with the percentile-based DPC method, the proposed ADPC method produces better results in that they are more consistent with the benchmark clusters shown in Fig. 4.

The identification of the cluster centroids from the decision graph is important for the clustering process. It can be seen that the simple threshold-based centroid identification, such as the percentile-based DPC method, may fail to work properly for the complex datasets.

To further evaluate the effectiveness of the proposed method, it is also compared with the DPC method using manual parameter tuning, in which different TH_γ values are tuned for different datasets. A lot of values of the parameter TH_γ are tried for each dataset and the best one is used in the manual DPC method. The F1-scores and the number of clusters produced by the manual DPC method and the ADPC method are recorded in Table 3 for comparison. The clustering results of the 6 synthetic datasets produced by the manual DPC method are also shown in Fig. 7.

As is can be seen from Table 3 and Fig. 7, the ADPC method has produced similar results as the manual DPC method. More specifically, the two methods have produced the same results for 5 out of the total 14 datasets. Furthermore, the ADPC method has produced better results for 6 datasets (Dataset B, Ecoli, Glass Identification, liver Disorder, Wine Quality_red, Statlog (Shuttle))

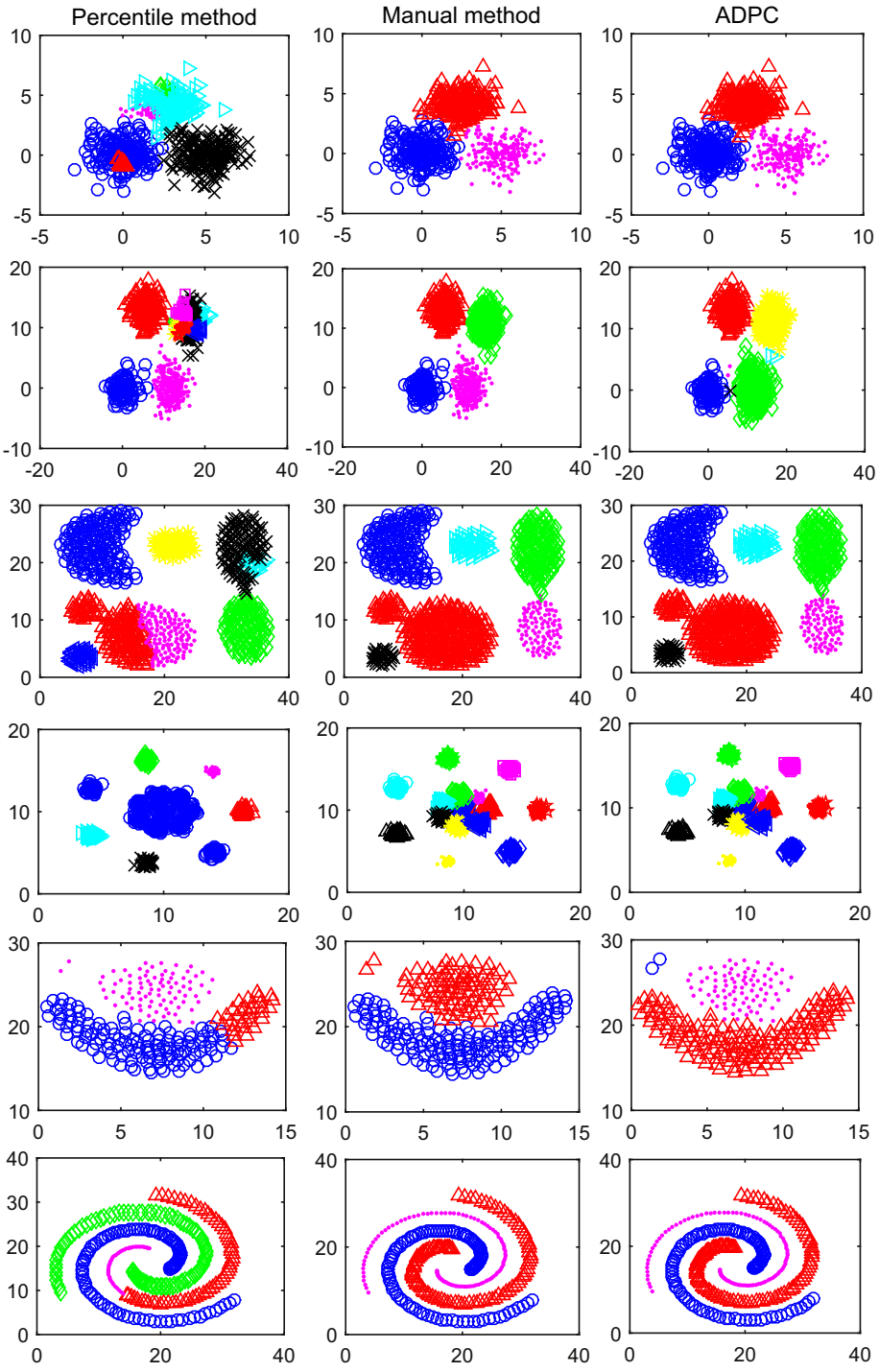

Fig. 7. The clustering results of the six synthetic datasets produced by the Percentile-based DPC method (left column), the Manual DPC method (middle column), and the ADPC method (right column).

which all have complex distribution or high dimensions, while the manual DPC method has produced better results for only 3 datasets (Flame, QSAR biodegradation and Banknote authentication). It should be noticed that the ADPC method uses the same parameter settings for all the different datasets while the manual DPC method uses different tuned parameters for different datasets. A two-sided paired t-test is done with a hypothesis that the manual DPC method produces similar F1-scores as the ADPC method. The hypothesis is accepted at the 5% significance level with a p-value $= 0.5421$, which shows that the results produced by the two methods are similar.

From Tables 2 and 3 and Fig. 7, it can be seen that the proposed ADPC method can produce better results than the percentile-based DPC method, and can produce competitive results as the manual DPC method.

In addition, we also compare the proposed ADPC method with two traditional clustering methods: Kmeans and Birch [9]. For both Kmeans and Birch, the exact number of the clusters in the ground truth is given as input. The results of these three methods are shown in Table 4. As can be seen, Kmeans, Birch and ADPC have produced 4, 7 and 8 best results respectively out of the 14 datasets. For datasets Dataset B, Wine quality_red and QSAR

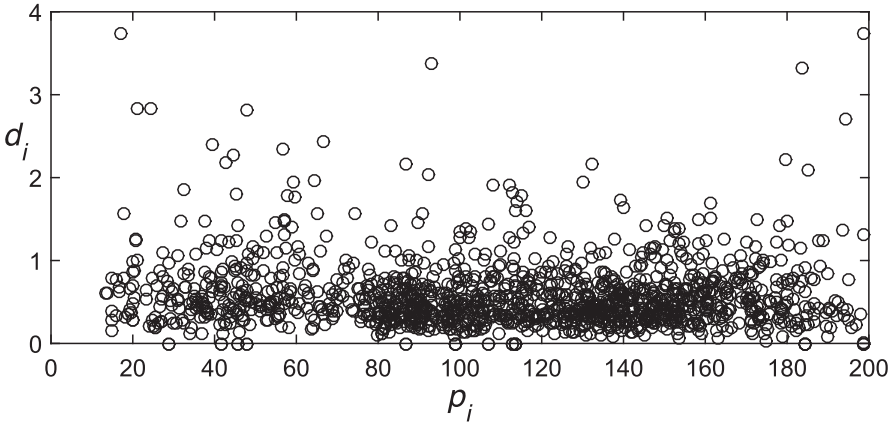

Fig. 8. The decision graph for Banknote authentication.

Table 3
The F1-scores and the number of clusters produced by the manual DPC method and the ADPC method on 14 different datasets.

Dataset	Manual method		ADPC	
	F1-score	#Clusters	F1-score	#Clusters
Dataset A	**0.964**	3	**0.964**	3
Dataset B	0.992	4	**0.995**	7
Aggregation	**0.960**	6	**0.960**	6
Flame	**1**	2	0.992	3
R15	**0.993**	15	**0.993**	15
Spiral	**1**	3	**1**	3
Ecoli	0.617	2	**0.644**	5
EEG Eye State	**0.671**	2	**0.671**	4
Glass Identification	0.492	2	**0.510**	5
Liver Disorders	0.522	3	**0.669**	9
Wine quality-red	0.506	3	**0.510**	9
QSAR biodegradation	**0.712**	2	0.711	4
Statlog (Shuttle)	0.636	6	**0.784**	85
Banknote authentication	**0.609**	3	0.443	35

Table 4
The F1-scores produced by Kmeans,Birch,and ADPC method on 14 different datasets.

Dataset	Kmeans	Birch	ADPC
Dataset A	**0.984**	0.968	0.964
Dataset B	0.990	**0.996**	0.995
Aggregation	0.827	0.916	**0.960**
Flame	0.747	0.756	**0.992**
R15	**0.993**	0.782	**0.993**
Spiral	0.334	0.370	**1**
Ecoli	0.640	0.428	**0.664**
EEG Eye State	**0.671**	**0.671**	**0.671**
Glass Identification	0.436	**0.529**	0.510
Liver Disorder	0.627	0.668	**0.669**
Wine quality_red	0.284	**0.520**	0.510
QSAR biodegradation	0.588	**0.712**	0.711
Statlog (Shuttle)	**0.784**	**0.784**	**0.784**
Banknote authentication	0.548	**0.659**	0.443

biodegradation, the gap between the results produced by the proposed ADPC method and the best results produced by Kmeans or Birch is very small. So, although the optimal number of clusters is given as input for both Kmeans and Birch, the proposed ADPC method can still produce competitive results compared to the two traditional methods.

For the Statlog (Shuttle) dataset, the ADPC method has identified much more cluster centroids than the actual ones. To study this case, the number of points in each cluster is counted. It is found that there are 76 clusters which have less than 3 points. It indicates that the ADPC method identifies some outliers in the dataset as cluster centroids. Although ADPC has produced the best result, a more effective algorithm is needed to avoid identifying these outliers as centroids in our future work. For Banknote authentication, the ADPC method fails to produce a satisfying result. To study this special case, the decision graph for the Banknote authentication dataset is shown in Fig. 8. It can be seen from Fig. 8 that it is difficult to identify the three centroids correctly even by human eyes. Clustering on datasets like Banknote authentication does not satisfy the two conditions of cluster centroids, which may be why the ADPC method fails to work well for the dataset.

4.4. Application to image segmentation

Image segmentation is the decomposition of a gray level or color image into homogeneous tiles. It is arguably the most important low-level vision task. Homogeneity is usually defined as similarity in pixel values, so clustering algorithms can be used for image segmentation. In our experiments, the proposed ADPC method and the percentile-based DPC method are used to do automatic image segmentation for four color images named Flower, Kids, City and Plane, which have 31,200, 30,070, 30,000 and 30,000 pixels respectively. In the experiments, only the RGB values of each pixel are used as features for both methods. The results of the image segmentation are shown in Fig. 9.

As shown from the Fig. 9, the first column on the left are the original images, the middle column shows the segmentation produced by the percentile-based DPC method, and the last column on the right shows the segmentation produced by the proposed ADPC method. The first row contains the results for the Flower image, the second row contains the results for the Kids image, the third row contains the results for City image, and the fourth row contains the results for the Plane image. For the Flower image, the percentile-based DPC method produces 312 clusters, while the ADPC method produces 321 clusters. For the Kids image, the percentile-based DPC method produces 301 clusters, while the ADPC method produces 227 clusters. For the City image, the percentile-based DPC method produces 300 clusters, while the ADPC method produces 280 clusters. For the Plane image, the percentile-based DPC method produces 300 clusters, while the ADPC method produces 225 clusters. So in most cases, the percentile-based DPC method produces more clusters than the proposed ADPC method for the images.

From the segmentation results in Fig. 9, it can be seen that the proposed ADPC method can identify good homogenous segmented regions, such as sky, clouds, walls, roofs, cars, flowers, stamens, patterns on the T-shirts, the wheels and the missiles of the plane, *etc*. The percentile-based DPC method can also identify

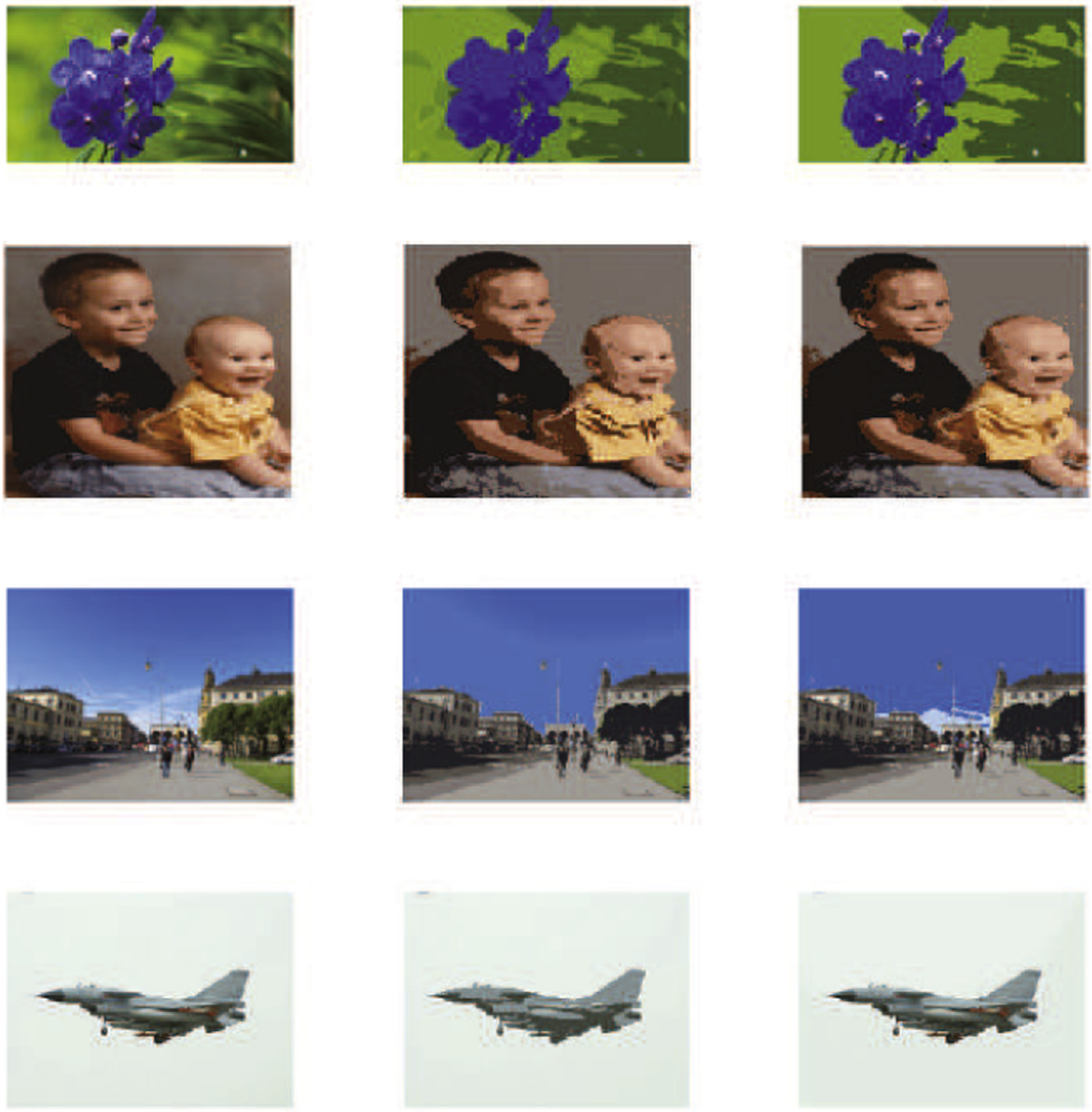

Fig. 9. The image segmentation results by the percentile-based DPC method (middle) and the proposed ADPC method (right) for the color images (Flower, Kids, City and Plane) (left).

homogenous segmented regions, but it fails to identify the clouds, cars, stamens, and the wheels of the plane. So, in most cases, although the percentile-based DPC method produces more clusters, it identifies fewer details of the images compared to the ADPC method. It can be seen that with the proposed automatic centroid identification in the ADPC method, good image segmentation can be produced with only the RGB features.

5. Conclusion

In this paper, a novel clustering method is proposed based on statistical automatic centroid identification from the decision graph. It is shown that the proposed ADPC method can deal with datasets of various distributions and dimensionalities, and the proposed statistical-based centroid identification is better than the simple threshold-based centroid identification. Moreover, the ADPC method can also be used for image segmentation effectively. In the future work, we plan to improve the ADPC method for dealing with difficult decision graphs so that it can estimate the number of clusters more accurately.

Acknowledgments

This work is supported by the National Key R&D Program of China (Grants no. 2017YFE0111900, 2018YFB1003205), and the Lanzhou Talents Program for Innovation and Entrepreneurship (Grants No. 2016-RC-93).

Supplementary material

Supplementary material associated with this article can be found, in the online version, at doi:10.1016/j.neucom.2018.10.067.

References

[1] M.T. Law, R. Urtasun, R.S. Zemel, Deep spectral clustering learning, in: Proceedings of the International Conference on Machine Learning, 2017, pp. 1985–1994.
[2] H. Zhang, T.W.S. Chow, Q.M.J. Wu, Organizing books and authors by multilayer som, IEEE Trans. Neural Netw. Learn. Syst. 27 (12) (2015) 2537.
[3] H. Zhang, S. Wang, X. Xu, T. Chow, Q. Wu, Tree2vector: learning a vectorial representation for tree-structured data., IEEE Trans. Neural Netw. Learn. Syst. PP (99) (2018) 1–15.
[4] P. Kulczycki, M. Charytanowicz, P.A. Kowalski, S. Lukasik, The complete gradient clustering algorithm: properties in practical applications, J. Appl. Stat. 39 (6) (2012) 1211–1224.
[5] Y. Lu, Y. Wan, Clustering by sorting potential values (CSPV): a novel potential-based clustering method, Pattern Recognit. 45 (9) (2012) 3512–3522.
[6] J.M. Kleinberg, An impossibility theorem for clustering, in: Proceedings of the Advances in Neural Information Processing Systems, 2003, pp. 463–470.
[7] M. Charikar, V. Chatziafratis, Approximate hierarchical clustering via sparsest cut and spreading metrics, in: Proceedings of the Twenty-Eighth Annual ACM-SIAM Symposium on Discrete Algorithms, Society for Industrial and Applied Mathematics, 2017, pp. 841–854.
[8] S. Guha, R. Rastogi, K. Shim, Cure: an efficient clustering algorithm for large databases, in: Proceedings of the ACM Sigmod Record, 27, ACM, 1998, pp. 73–84.
[9] T. Zhang, R. Ramakrishnan, M. Livny, Birch: A new data clustering algorithm and its applications, Data Min. Knowl. Discov. 1 (2) (1997) 141–182.
[10] K. Wagstaff, C. Cardie, S. Rogers, S. Schrödl, et al., Constrained k-means clustering with background knowledge, in: Proceedings of the ICML, 1, 2001, pp. 577–584.
[11] B.J. Frey, D. Dueck, Clustering by passing messages between data points, Science 315 (5814) (2007) 972–976.
[12] A.M. Serdah, W.M. Ashour, Clustering large-scale data based on modified affinity propagation algorithm, J. Artif. Intell. Soft Comput. Res. 6 (1) (2016) 23–33.
[13] F. Höppner, Fuzzy Cluster analysis: Methods for Classification, Data analysis and Image Recognition, John Wiley & Sons, 1999.
[14] M. Ester, H.-P. Kriegel, J. Sander, X. Xu, et al., A density-based algorithm for discovering clusters in large spatial databases with noise., in: Proceedings of the KDD, 96, 1996, pp. 226–231.
[15] K. Fukunaga, L. Hostetler, The estimation of the gradient of a density function, with applications in pattern recognition, IEEE Trans. Inf. Theory 21 (1) (1975) 32–40.
[16] Y. Guo, A. Şengür, Y. Akbulut, A. Shipley, An effective color image segmentation approach using neutrosophic adaptive mean shift clustering, Measurement 119 (2018) 28–40.
[17] R.J. Campello, D. Moulavi, A. Zimek, J. Sander, Hierarchical density estimates for data clustering, visualization, and outlier detection, ACM Trans. Knowl. Discov. Data (TKDD) 10 (1) (2015) 5.
[18] A. Rodriguez, A. Laio, Clustering by fast search and find of density peaks, Science 344 (6191) (2014) 1492–1496.
[19] H. Yan, Y. Lu, H. Ma, Density-based clustering using automatic density peak detection, in: Proceedings of the International Conference on Pattern Recognition Applications and Methods, 2018, pp. 95–102.
[20] (http://www.cs.joensuu.fi/sipu/datasets).
[21] (http://www.archive.ics.uci.edu/ml/datasets/).
[22] C.J.V. Rijsbergen, Foundation of evaluation, J. Document. 30 (4) (1974) 365–373.
[23] E. Amigó, J. Gonzalo, J. Artiles, F. Verdejo, A comparison of extrinsic clustering evaluation metrics based on formal constraints, Inf. Retr. 12 (4) (2009) 461–486.

Huanqian Yan received his bachelor's degree in School of Computer Science and Engineering, Changchun University of Science and Technology, now he is a graduate in the School of Information Science and Engineering, Lanzhou University, majored in computer science. His research interests include pattern recognition and image processing.

Lei Wang received the bachelor's degree in software engineering from Northwest A&F University in 2016. He is currently a graduate in the School of Information Science and Engineering, Lanzhou University. His major is software engineering and his research interest is machine learning, big data, and data mining.

Yonggang Lu received both the B.S. and M.S. Degrees in Physics from Lanzhou University, Lanzhou, China in 1996 and 1999 respectively. Later he received the M.S. and Ph.D. Degrees in Computer Science from New Mexico State University, Las Cruces, NM, USA in 2004 and 2007 respectively. He finished some of the Ph.D. work at Los Alamos National Lab, NM, USA. He is now a professor in the School of Information Science and Engineering, Lanzhou University, Lanzhou, China. His main research interests include pattern recognition, image processing, neural networks, and bioinformatics.