International Journal of Machine Learning and Cybernetics (2021) 12:1243–1254
https://doi.org/10.1007/s13042-020-01231-2

ORIGINAL ARTICLE

A robust spectral clustering algorithm based on grid-partition and decision-graph

Lijuan Wang[1,3] · Shifei Ding[1,2] · Yanru Wang[1] · Ling Ding[1]

Received: 16 April 2020 / Accepted: 2 November 2020 / Published online: 19 November 2020

Abstract
Spectral clustering (SC) transforms the dataset into a graph structure, and then finds the optimal subgraph by the way of graph-partition to complete the clustering. However, SC algorithm constructs the similarity matrix and feature decomposition for overall datasets, which needs high consumption. Secondly, k-means is taken at the clustering stage and it selects the initial cluster centers randomly, which leads to the unstable performance. Thirdly, SC needs prior knowledge to determine the number of clusters. To deal with these issues, we propose a robust spectral clustering algorithm based on grid-partition and decision-graph (PRSC) to reduce the amount of calculation and improve the clustering efficiency. In addition, a decision-graph method is added to identify the cluster centers quickly to improve the algorithm stability without any prior knowledge. A numerical experiments validate that PRSC algorithm can effectively improve the efficiency of SC. It can quickly obtain the stable performance without any prior knowledge.

Keywords Spectral clustering · Grid-partition method · Decision-graph method · Stable performance

1 Introduction

In the big data era, clustering is widely used in data mining, pattern recognition, image segmentation and maybe it is the most important technique of unsupervised learning [1–3]. The mainly goal of clustering is to divide the original data points into different clusters according to their similarity and the similarity of the same cluster is larger and the dissimilarity between different clusters is larger [4]. However, the performance of most well-known traditional clustering algorithms, such as the k-means algorithm, the affinity propagation (AP) algorithm, and the density-based spatial clustering of applications with noise (DBSCAN) algorithm are powerless when processing the datasets with cross-classes and complex distribution structure [5–7].

Spectral clustering (SC) algorithm is a clustering method based on graph theory [8], which is a classical kernel-based method. For a given dataset clustering, it constructs an undirected weighted graph, where the vertices of the graph represent data points, and each edge of the graph has a weight to describe the similarity between the vertices [9]. Aggregating data points into multiple clusters is equivalent to dividing the graph into several subgraphs, so that the connection weights inside the subgraphs are the largest, and the connection weights between the subgraphs are the smallest [10]. In today's society, data has become a valuable resource. SC algorithm, as an effective method of data analysis, can discover the internal connections of different data and provide valuable information for decision makers [11]. Due to its reliable theoretical foundation and good clustering performance, SC has been successfully applied in many fields, such as computer vision, data analysis, image processing, video surveillance, automatic control and other fields [12–14].

Although SC algorithm has achieved good performance in real practice, there still exist a great deal of situations when it faces large scale dataset to be further studied [15]. Firstly, the whole dataset is taken for similarity matrix

✉ Shifei Ding
dingsf@cumt.edu.cn

1 School of Computer Science and Technology, China University of Mining and Technology, Xuzhou 221116, China

2 Mine Digitization Engineering Research Center of Ministry of Education of the People's Republic of China, Xuzhou 221116, China

3 School of Information and Electrical Engineering, Xuzhou College of Industrial Technology, Xuzhou 221400, China

construction and feature decomposition, which costs high computational complexity and will affect the applications of SC. Secondly, SC algorithm relies on k-means to complete clustering, which is sensitive with initial cluster centers, and results in an unstable performance. Thirdly, not only SC itself but also k-means need prior knowledge to determine the number of clusters in advance. In order to overcome the shortcomings of SC, many studies have been proposed to improve it. To tackle the difficulty of finding the clusters in the cases that dataset has a large difference in density and overcome its clustering effect depends on the selection of initial centers, Chen et al. [16] propose a novel spectral clustering algorithm based on membrane computing framework, called MSC algorithm, whose idea is to use membrane clustering algorithm to realize the clustering component in spectral clustering. A tissue-like P system is used as its computing framework, where each object in cells denotes a set of cluster centers and velocity-location model is used as the evolution rules. Under the control of evolution-communication mechanism, the tissue-like P system can obtain a good clustering partition for each dataset. In order to dispose with multi-scale datasets, Zelnik-Manor et al. [17] proposed an adaptive spectral clustering technique. Instead of selecting a global parameter σ, it calculates an adaptive parameter σ_i for each point x_i based on its neighborhoods information, where σ_i is the Euclidean distance from point x_i to its p nearest neighbors. The similarity measure is called an adaptive Gauss kernel function. Considering the data distribution of each point's neighborhoods, self-adjusting spectral clustering can effectively separate the compact clusters contained in the sparse background cluster. Facing with the traditional spectral clustering algorithms use Gaussian kernel function to construct the similarity matrix, and they are sensitive to the selection of scale parameter. In addition, they need to randomly determine the initial cluster centers at the clustering stage and the clustering performance is not stable. Wang et al. [18] present an algorithm on the basis of message passing, which makes use of a density adaptive similarity measure, describing the relations between data points and obtaining high-quality cluster centers through message passing mechanism in AP clustering. The performance of clustering is optimized by this method. There are three pivotal bottlenecks that are usually encountered in conventional spectral clustering methods: (1) equally treat each data point, so that easily affected by the outliers; (2) are sensitive to the initialization; (3) need to specify the number of cluster. To conquer these issues, Wen et al. [19] have proposed a novel spectral clustering algorithm, via employing an affinity matrix learning to learn an intrinsic affinity matrix, using the local PCA to resolve the intersections; and further taking advantage of a robust clustering that is insensitive to initialization to automatically generate clusters without an input of number of cluster.

Despite the theoretical and practical advantages of aforementioned algorithms, all these introduce new parameters to help improve the performance of SC. Consequently, the challenge of finding a robust algorithm is interesting. In order to tackle the aforementioned problems, we proposed a robust spectral clustering algorithm based on grid-partition and graph-decision (PRSC). PRSC introduces a grid-partition method for data mapping to reduce the scale of processed dataset. At the same time, PRSC introduces a decision-graph method to identify the cluster centers for preparation. The major contributions of the paper are:

- We propose a robust spectral clustering algorithm based on grid-partition and graph-decision (PRSC) to improve the performance of the traditional SC.
- PRSC algorithm introduces a grid-partition method to improve the efficiency of SC and introduces a decision-graph method to identify the cluster centers without any prior knowledge.
- Our experimental results show that the PRSC can obtain stable clustering result than SC on most datasets. Besides, PRSC based on grid-partition and graph-decision can outperform several traditional algorithms in terms of obtaining optimal clustering.

The remainder of this paper is organized by this way: We introduce the main principle of SC and DPC in Sect. 2. The proposed PRSC algorithm is presented in Sect. 3. Section 4 designs experiments to illustrate the efficiency of PRSC. Finally, concluding generalization and further challenges are summarized.

2 Related works

The proposed PRSC is based on density peaks clustering and spectral clustering. In this section we give a briefly introduction about the spectral clustering (SC) algorithm and density peaks clustering (DPC) algorithm.

2.1 Basic theory of SC algorithm

The idea of SC algorithm is derived from spectral graph theory, which transforms the data clustering problem into a graph-partition problem to solve, especially suitable for the case of non-convex datasets [20]. Given a dataset $X = \{x_1, x_2, \ldots, x_n\}$, an $n \times n$ similarity matrix can be constructed based on the pairwise similarity of the data points, and the spectral method is clustering based on the feature vectors and feature values of the similarity matrix [21]. Using these feature vectors, a low-dimensional embedding subspace of the data points can be constructed, and in this embedding space, the data points can be clustered using

k-means [22]. Normalized cut is a typical spectral clustering algorithm [23]. The basic principle of non-normalized SC algorithm is briefly introduced below.

An undirected weighted graph $G(V,E)$, where each point $x_i \in X$ is taken as a vertex i of the graph, and the similarity value w_{ij} between any two points (x_i, x_j) is taken as the connection between the two vertices (i,j). Gaussian kernel functions are commonly used in SC for constructing the similarity matrix. It is defined by:

$$\mathrm{W}_{ij} = \exp(-||x_i - x_j)||^2/2\sigma^2) \tag{1}$$

where, d_{ij} represents the European distance between x_i and x_j. σ is a scale parameter. In graph G, the sum of the weights of the edges connected to vertex i is defined as the degree of vertex i, which is represented by:

$$d_{ii} = \sum_{j=1}^{n} w_{ij} \tag{2}$$

Therefore, the degree matrix $D \in R^{n\times n}$ of graph G is a diagonal matrix. The elements on the diagonal are d_{ii} and the elements outside the diagonal are 0. After the graph is constructed, clustering results can be obtained by the way of decomposing the graph into independent connected components using some graph-partition methods [24]. There are many traditional graph-partition methods, such as minimum cut set method, proportional cut set method, canonical cut set method, and minimum and maximum cut set method [25]. After mathematical derivation, it can be proved that the eigenvalues and eigenvectors of the Laplacian matrix of the graph contains classification information of vertices, and make full use of these information can get better clustering results [26]. Laplacian matrix is calculated by:

$$L = D - W \tag{3}$$

Algorithm 1 Non-normalized SC algorithm

Input: Dataset $X = \{x_1, x_2, \cdots, x_n\}$, number of clusters k.

Output: Cluster label.

Step 1: Construct a similarity matrix for overall data points;
Step 2: Calculate the non-normalized graph Laplacian matrix L;
Step 3: Calculate the first k eigenvectors of L;
Step 4: Clustering feature vectors by k-means algorithm;
Step 5: Output clustering results.

Compared with other traditional clustering algorithms, SC has shown extraordinary effectiveness in processing datasets with cross-classes and complex distribution structure and the brief steps of non-normalized SC is described as Algorithm 1 [27].

Despite the superior performance, the SC method easily encounter the problem of scalability, because both storage and computation of a large affinity matrix and eigen-decomposition are high-consuming [28, 29]. Without too many details, it mainly includes three parts of computational complexity. Firstly, the construction of similarity matrix requires $O(n^2)$ time complexity. The second part of the time-consuming operation is to calculate the eigenvector of the smallest eigenvalue. A common method is the Lanczos algorithm, which requires $O(m^3) + (O(mn) + O(nt) + O(p(m-k)))$ time complexity, where $m > k$ is the Arnoldi length defined by the user, and p is the number of Arnoldi iterations for each execution of the algorithm, m is usually set to several times the size of k. The third part is to run k-means on the feature vector matrix. In each iteration, the distance between each point and all cluster centers must be calculated. This results in time complexity $O(l \cdot nk^2)$, where l is the number of iterations of k-means. In general, we can see that SC is a time-consuming clustering algorithm. In addtion, SC needs to set the number of clusters in advance, and the sensitivity of k-means to the initial cluster centers will lead to unstable clustering results [30]. Thus, we need to propose some solutions to speed up the clustering process and remove dependence on initial cluster centers.

2.2 Basic theory of DPC algorithm

Recently, a novel density-based algorithm named density peaks clustering (DPC) algorithm has attracted widely attention [31, 32]. DPC is proposed to obtain arbitrary clusters with fewer parameters and no iteration [33]. The main particularity of DPC is the decision-graph for identifying cluster centers firstly without any prior knowledge [34]. Cluster centers can been identified by a 2D decision-graph on the basis of their characteristics [35].

Assume that the dataset is $X = \{x_1, x_2, \cdots, x_n\}$, let d_{ij} represents the similarity between x_i and x_j, which is calculated by European distance in DPC. Implementing DPC is on the basis of two assumptions: cluster centers have higher local-density than their neighbours; cluster centers are positioned far from each other [36, 37].

Assume 1 The local-density of a cluster center is higher than its neighbours, which is calculated as:

$$\rho_i = \sum_j \chi\left(d_{ij} - d_c\right)$$
$$\chi(x) = \begin{cases} 1, x < 0 \\ 0, x \geq 0 \end{cases} \tag{4}$$

where d_c, an input parameter is represents the cut-off distance.

Assume 2 The relative-distance between any two cluster centers is relatively large, which is defined as:

$$\delta_i = \begin{cases} \min\limits_{j:\rho_j>\rho_i} \left(d_{ij}\right), & if\ \exists j\ s.t. \rho_j > \rho_i \\ \max\limits_{j} \left(d_{ij}\right), & otherwise \end{cases} \tag{5}$$

Once ρ and δ of all data points are obtained, a decision-graph is constructed as Fig. 1a with ρ as the abscissa and δ as the ordinate. Also DPC algorithm can select the cluster centers based on Fig. 1b. Decision-graph drawn in this way is based on γ, which is calculated as:

$$\gamma_i = \rho_i \cdot \delta_i \tag{6}$$

DPC selects points with large attributes as the cluster centers according to the decision-graph [38]. After that, the non-centers are classified. They are traversed in the order of ρ from large to small, and the data points are categorized to the nearest higher-density point. At the same time, a boundary threshold is defined, that is, the boundary threshold of a cluster is represented as the point with the highest density in the set of points that belong to the cluster but is less than d_c from other clusters, and is set as ρ_b. Finally, the points with density not greater than ρ_b are regarded as noise points. The brief steps of DPC is described as Algorithm 2 [39].

Algorithm 2 DPC algorithm

Input: Dataset $X = \{x_1, x_2, \cdots, x_n\}$, input parameter $d_c = 2$.

Output: Cluster labels.

Step 1: Calculate a similarity matrix for overall data points;

Step 2: Calculate local-density and relative-distance for each data based on Assume 1 and Assume 2;

Step 3: Identify cluster centers according to the local-density and relative-distance;

Step 4: Assign non-center points to the nearest high-density point class;

Step 5: Screen outliers with local-density not exceed ρ_b;

Step 6: Output clustering results.

As a simple and effective density-based clustering algorithm, DPC has been playing a prominent role in data mining such that community discovery, genetic disease, biology and so on [40–42]. However, it will result in an unsatisfactory clustering result when dealing with a high-dimensional sparse dataset in DPC.

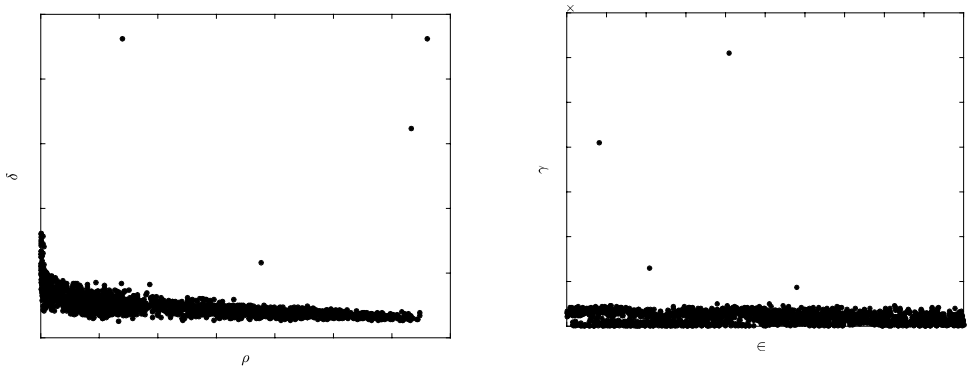
3 Robust spectral clustering algorithm based on grid-partition and decision-graph

In order to improve the clustering efficiency of SC and reduce its dependence on the initial cluster centers, an improved robust spectral clustering algorithm (PRSC) is proposed. We focus on the topic of reduce the data scale of clustering on large-scale data, a grid-partition method is introduced, which replaces all data points with non-empty grids, and adopted SC on the grid units to enhance the algorithm efficiency. Decision-graph method in DPC is taken for identifying cluster centers to improve the stableness of SC.

3.1 The description of the PRSC algorithm

In PRSC, the dataset needs to be pre-processed, that is, the data points are mapped into the corresponding grid units. Assume $X = \{x_1, x_2, \ldots, x_n\}$, $x_i = [l_i, h_i)$ is a data space with d dimensions. First, feature space is divided into several equal and mutually disjoint intervals according to the data structure. The grid side of partition is defined as:

$$\xi = a\left(\prod_{i=1}^{d} \frac{h_i - l_i}{n}\right)^{\frac{1}{d}} \tag{7}$$



(a) **(b)**

Fig. 1 Two methods of drawing decision-graph. **a** Decision-graph based on ρ and δ. **b** Decision-graph based on γ

where, $a = 0.5$ is the proportional coefficient and is given by the experience. After that, data points are mapped into grids and only non-empty grids are seen as objects for subsequent operations.

Firstly, assuming the data points are mapped into $\{u_1, u_2, \ldots, u_g\}$ non-empty grids, similarity matrix is constructed based on the non-empty grids. Different with SC algorithm, similarity matrix here only measures by Euclidean distance. For example, let s_{ij} denote the Euclidean distance between the grids $u_i = (u_{i1}, u_{i2}, \ldots u_{id})$ and $u_j = (u_{j1}, u_{j2}, \ldots, u_{jd})$ as follows:

$$s_{ij} = \sqrt{\sum_{k=1}^{d} \left(u_{ik} - u_{jk}\right)^2} \quad (\text{where } i, j = 1 \ldots g) \tag{8}$$

Secondly, cluster centers are identified on the basis of decision-graph in DPC. Non-empty grids $\{u_1, u_2..., u_g\}$ and the local-density of grid u_i is:

$$\rho_{u_i} = count(u_i) \tag{9}$$

where *count*() represents the number of points. The relative-distance is calculated for non-empty grids by Eq. 5. Finally, objects with higher local-density and relative-distance are selected as cluster centers based on the decision-graph.

Thirdly, the cluster centers selected by the decision-graph is taken as the initial cluster centers of SC, and the non-empty grids are used as the clustering objects. SC is performed according to the similarity matrix constructed previously, and each data point is assigned to its mapped grid cluster. The brief steps of DPC is described as Fig. 2.

We introduce the grid-partition method into the SC algorithm, only non-empty grids are clustered, which can reduce the data scale and improve the efficiency of SC. In addition, decision-graph method in DPC is taken for identifying cluster centers, but local-density and relative-distance are calculated based on the non-empty grids are effective, which can improve the stableness of SC without more time complexity. More importantly, the number of clusters are same as the cluster centers selected by the decision-graph. Thus, PRSC is an algorithm without any more prior knowledge and the cluster result is robust.

3.2 Performance analysis of PRSC

Next, we take a deep exploration into the proposed PRSC. The main principle of PRSC algorithm is taking grid-partition and decision-graph to improve the efficiency of SC. The method adopt a two-stage process, which is composed of identifying cluster centers and SC performing mainly. First of all, it identifies cluster centers according to the non-empty grids. Although the information that the cluster centers can rely on is reduced, the non-empty grids can represent almost all the points. Thus, the cluster centers selected are almost the same as the cluster centers selected by DPC based on all information, but it is more efficient than DPC. Secondly, SC is performaned only on the non-empty grids, since non-empty grids represent almost all points, the accuracy has little effect, but it can greatly reduce time complexity. Greatly improved the algorithm performance and reduce the intricacy of spectral clustering algorithm.

The application of grid-partition will greatly reduce the amount of computation. Suppose that there is a dataset $X = \{x_1, x_2, \ldots, x_n\}$, it will be mapped into non-empty grids $\{u_1, u_2, \ldots, u_g\}$. Ignoring details, the time complexity of PRSC algorithm is dependent on two main stages. For identifying cluster centers, according to DPC, the main time complexity comes from the construction of similarity, which is $O(g^2)$. The second part of the time-consuming is same as SC, but the similarity matrix has been constructed above, it only contains the remaining two complexity. As the clustering object is non-empty grids $g \ll n$, thus, PRSC is must be more efficient than SC.

PRSC obtains optimal cluster centers based on the decision-graph in DPC, but more efficient than DPC. And PRSC obtains clustering results based on SC but more efficient than SC. Moreover, PRSC clustering automatically without any more parameters and is stable.

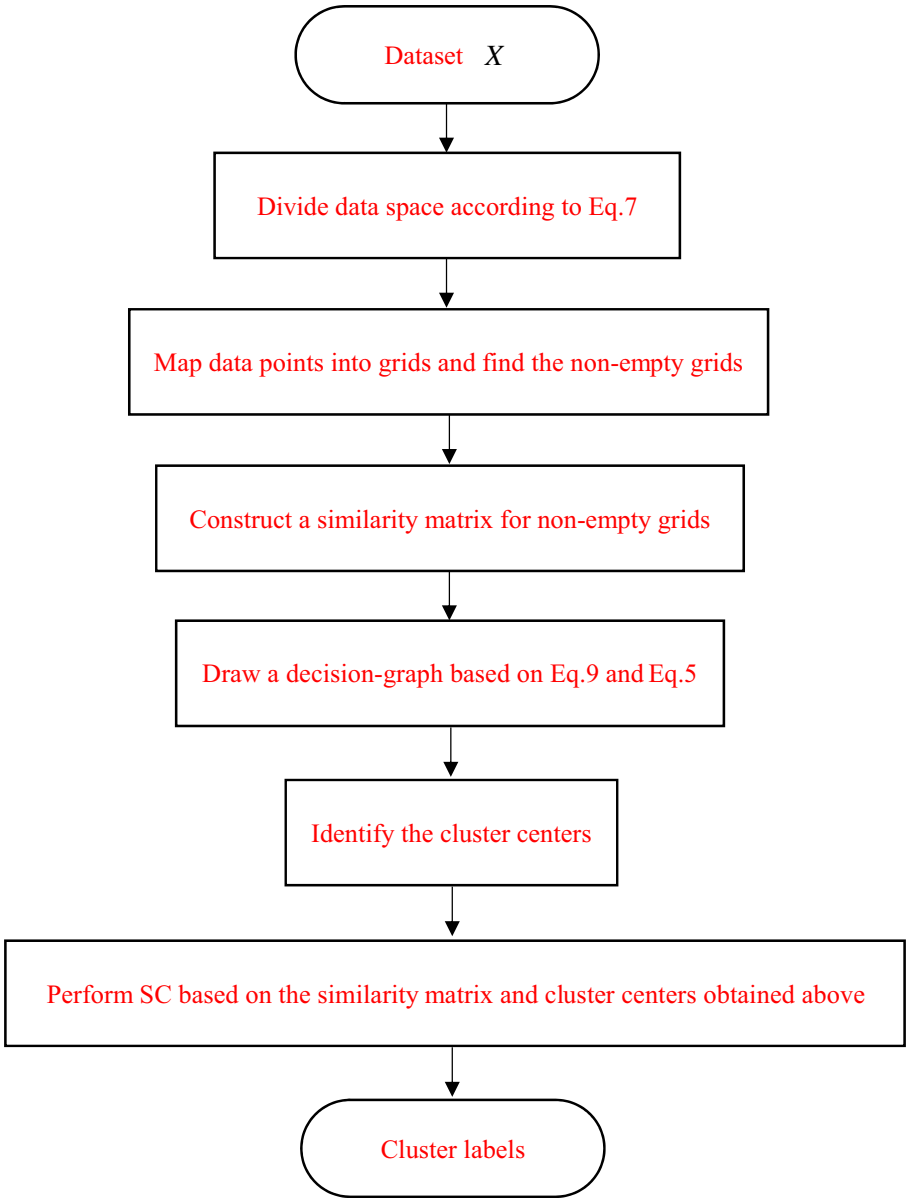


Fig. 2 Flowchart of PRSC algorithm

Table 1 Characteristics of four synthetic datasets

	Instances	Attributes	Clusters
Two cluster	400	2	2
Three cluster	600	2	3
Square	1000	2	4
Five cluster	2000	2	5

4 Experiments and results

To verify the effectiveness of the proposed PRSC algorithm, we select four synthetic datasets and eight UCI datasets. Their data characteristics are shown in Tables 1 and 3. The cluster numbers of datasets are various, which can prove the superiority of the proposed PRSC algorithm. The experiments are conducted on a desktop computer with a core i5 3.10 GHz processor, Window10 operating system and 4 GB RAM by running MATLAB 2015. In this part, we compare both clustering performance and running time among different clustering algorithms. The code of the comparison algorithm is provided by the authors and shows the best clustering results.

4.1 Experiments on synthetic datasets

Different characteristics of four synthetic datasets are presented in Table 1. We test PRSC and SC on these datasets and display the clustering results as colored plots, which are

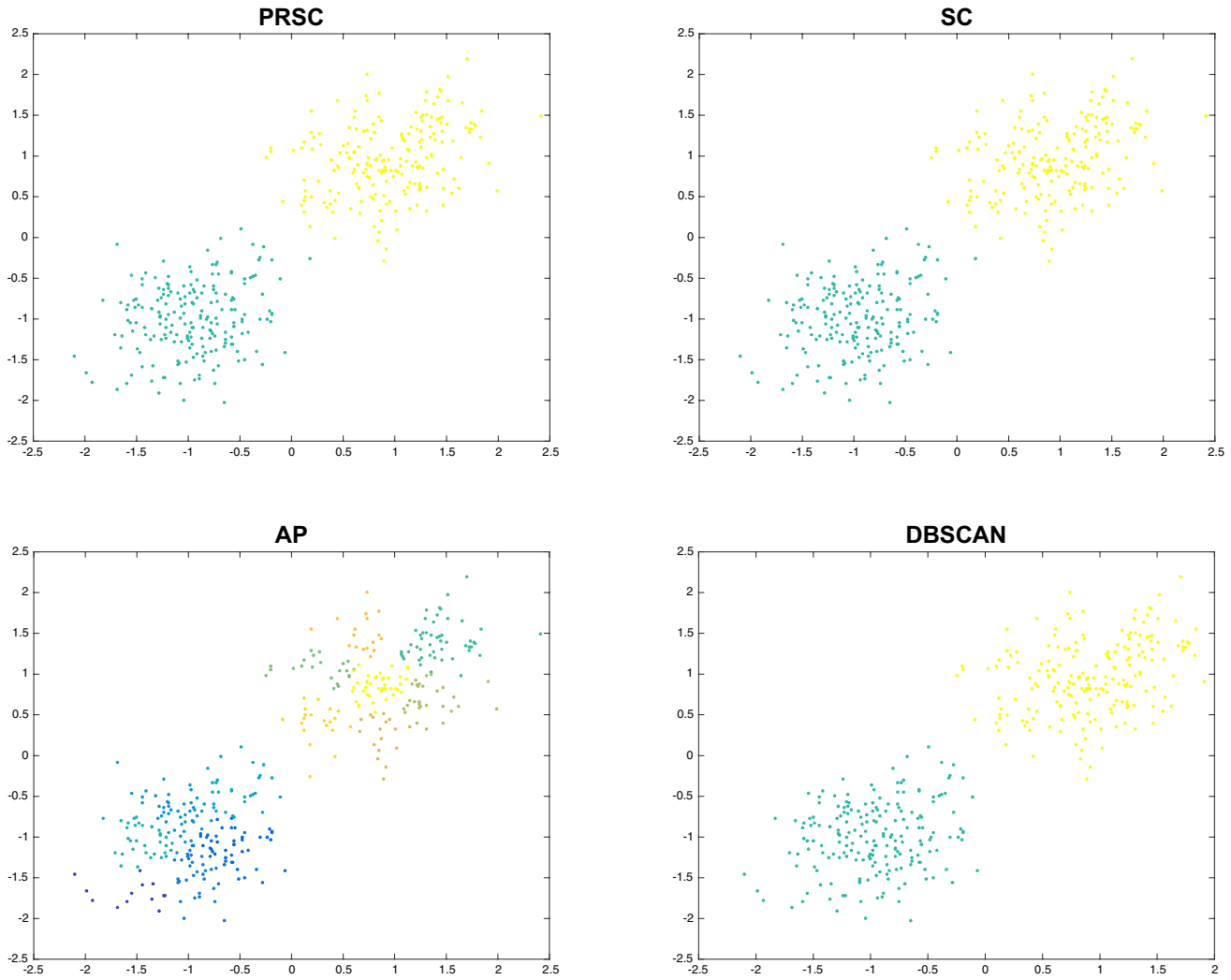


Fig. 3 Clustering results on two cluster

more straightforward. In addition, well-known traditional AP and DBSCAN algorithms are compared. The experiment results demonstrate the robustness of the PRSC clustering method in terms of the efficiency. The clustering results are shown in Figs. 3, 4, 5 and 6.

We can see from Figs. 3, 4, 5 and 6, although SC can obtain the accurate number of clusters on these datasets, it cannot perform satisfactory on Five cluster as the randomness of cluster centers. The traditional AP algorithm cannot obtain satisfactory clustering results. Although DBSCAN can get satisfactory results on most datasets, the selection of parameters is very difficult. However, PRSC selects the cluster centers based on the decision-graph, which is more robust. From Figs. 3, 4, 5 and 6, PRSC can obtain the satisfactory clustering results, which can prove that the grid-partition has little effect on clustering results.

After that, running time of PRSC, SC, AP, and DBSCAN are compared on these dataset as Table 2. The running time of PRSC and its comparison algorithms is the average of 10 runs.

It can be indicated from Table 2, PRSC is more efficient than SC, AP and DBSCAN. PRSC only performs clustering on the non-empty grids, which will significantly reduce the amount of calculation. Since these are two-dimensional data, the advantages of PRSC are not obvious. Next subsection takes eight UCI datasets to prove the superiority of the introduced PRSC algorithm.

4.2 Experiments on UCI datasets

Different characteristics of eight UCI datasets are presented in Table 3, and the dimensions of these datasets are relatively

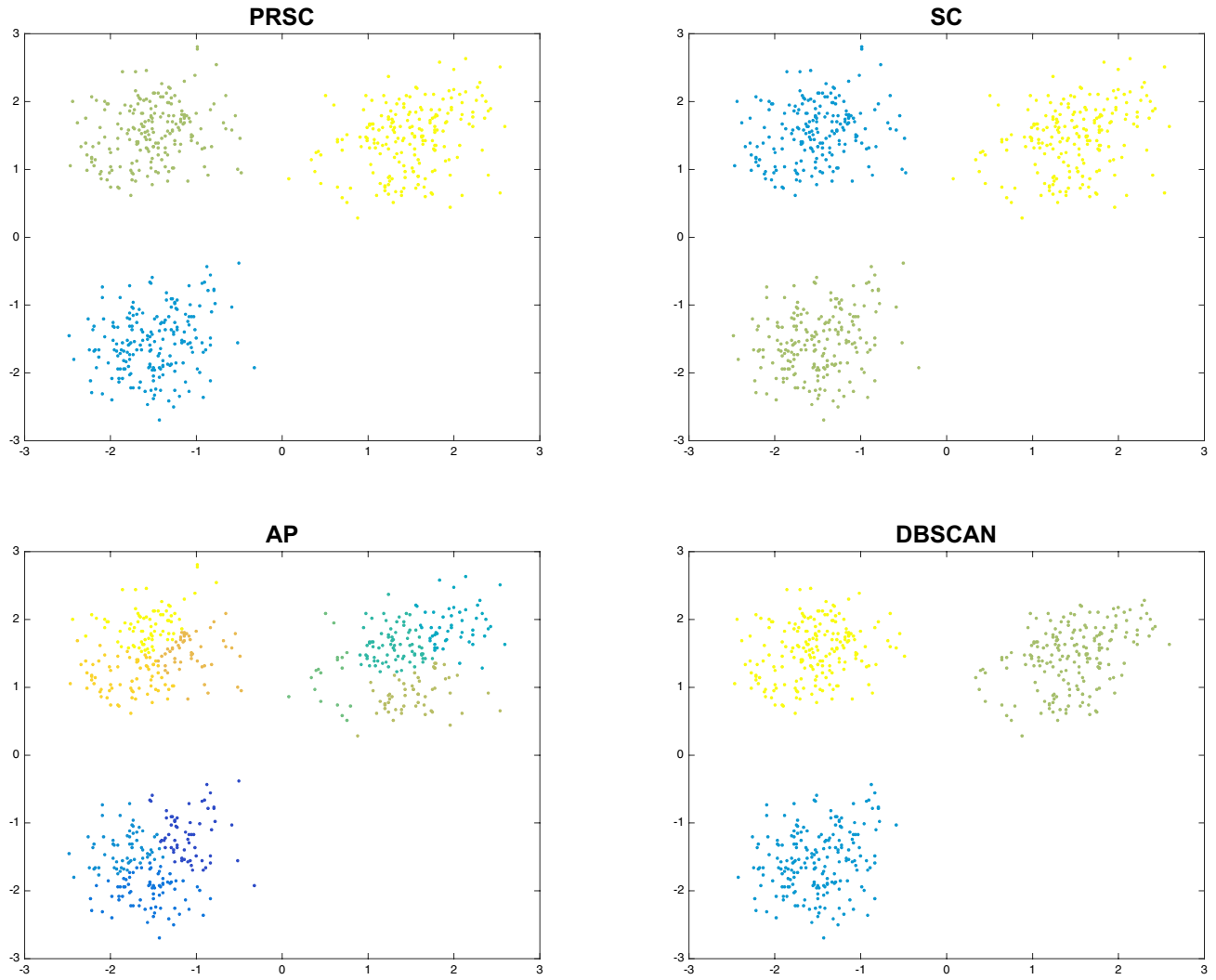


Fig. 4 Clustering results on three cluster

high. We test PRSC and SC on these datasets, and display the clustering results as Table 4. As SC is unstable, we run 10 times to take the average. In addition, DPC is compared as a new density-based clustering algorithm. We evaluate the clustering results in terms of Acc [42]. Assume y_i and z_i are the true classification labels and predictive classification labels respectively. $\delta(x, y) = 1$ if $x = y$ or $\delta(x, y) = 0$ otherwise. Each cluster label is mapped by the Hungarian algorithm optimally:

$$Acc = \sum_{i=1}^{N} \delta(y, map(z_i))/n \tag{10}$$

As can be seen from Table 3, the clustering accuracy of PRSC is as satisfactory as SC. Although DPC is superior to them on some datasets, it is not satisfactory on high-dimensional and sparse datasets, such as Sonar, Breast and Segmentation. PRSC is more suitable for high-dimensional datasets than DPC. It further proves that grid-partition method has little effect on the clustering results.

In order to prove PRSC algorithm is more efficient than SC, we also compare the running time of these algorithms and testify the feasibility of PRSC. Each algorithm runs 10 times for average, and the running time is shown as Table 5.

We can see from Table 5, on small-scale datasets PRSC is almost the same as SC and DPC. This is because the number of non-empty grid is not much different from the scale of the dataset. The running time of PRSC is superior to SC and DPC on large-scale datasets. The running time of PRSC is consist of two parts including the calculation for similarity matrix and SC process. Compared with SC and DPC, non-empty grids is less than data points. Thus, PRSC is efficient.

In summary, concluded by Tables 4 and 5, PRSC can obtain the satisfactory clustering results without prior knowledge. It processes large-scale datasets superior than DPC and SC.

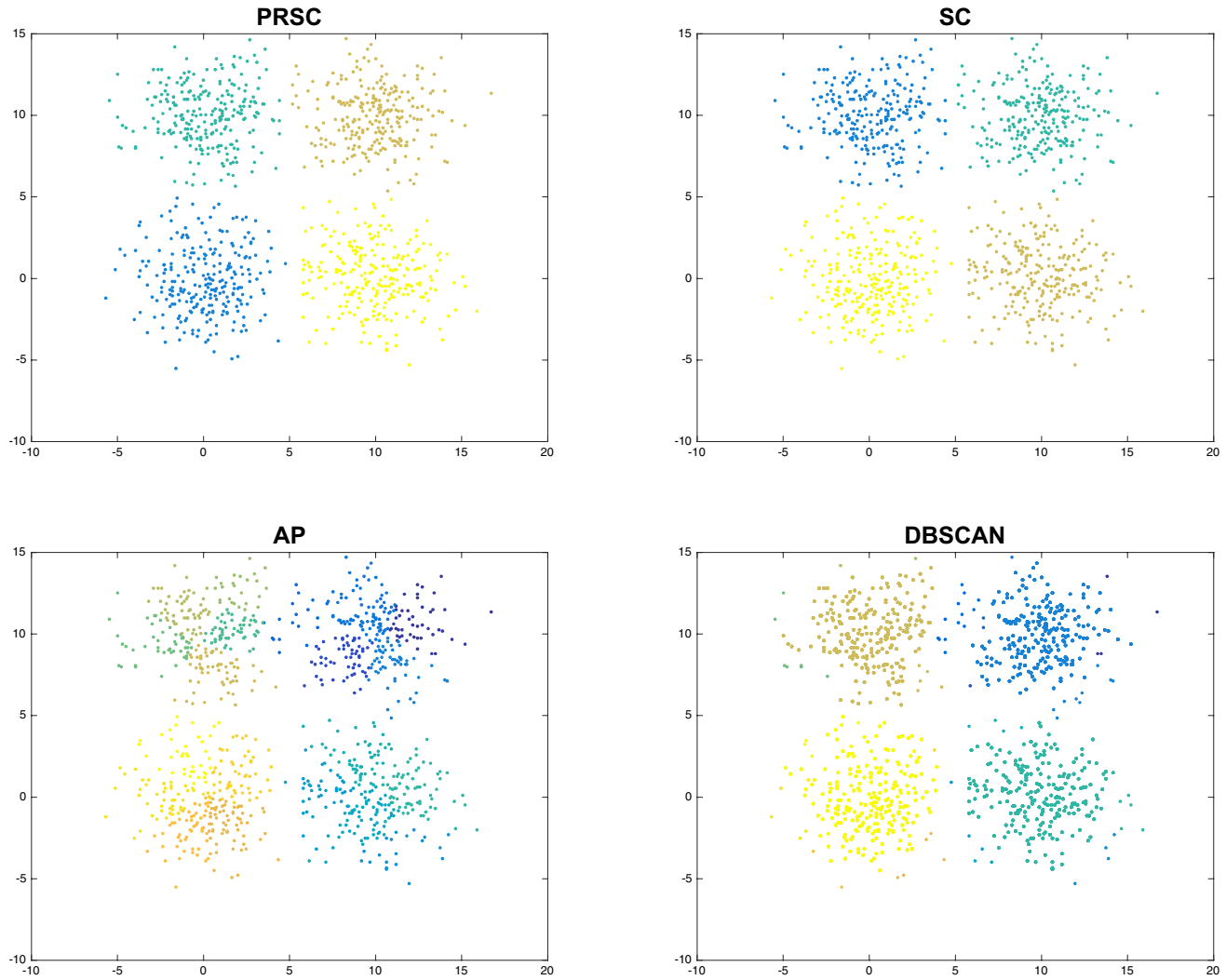


Fig. 5 Clustering results on square

5 Conclusions

In this paper, we proposed a robust spectral clustering (PRSC) algorithm to improve the SC. PRSC can effectively reduce the computational complexity through the grid-partition using non-empty grid units equivalent to the real datasets. Grid-partition is used to reduce the calculation cost. The calculation of the similarity and the clustering performance based on the non-empty grids will improve the efficiency of SC. Although PRSC reduces data information, it partitions the datasets based on the data structure with a little effect on the clustering accuracy. In addition, cluster centers are identified by a decision-graph for SC will eliminate parameters and keep clustering results stable.

The clustering performance on synthetic datasets and UCI datasets verify that PRSC is better than traditional SC. It not only can improve the efficiency of SC, but also can obtain the cluster centers automatically without prior knowledge. PRSC is feasible and robust. Furthermore, we consider using heuristic sampling or semi-supervised techniques [43] to further improve the performance of the proposed algorithm. Whether it is the proposed PRSC algorithm or the SC algorithm, the clustering effectiveness needs to be further explored, and expand more practical applications.

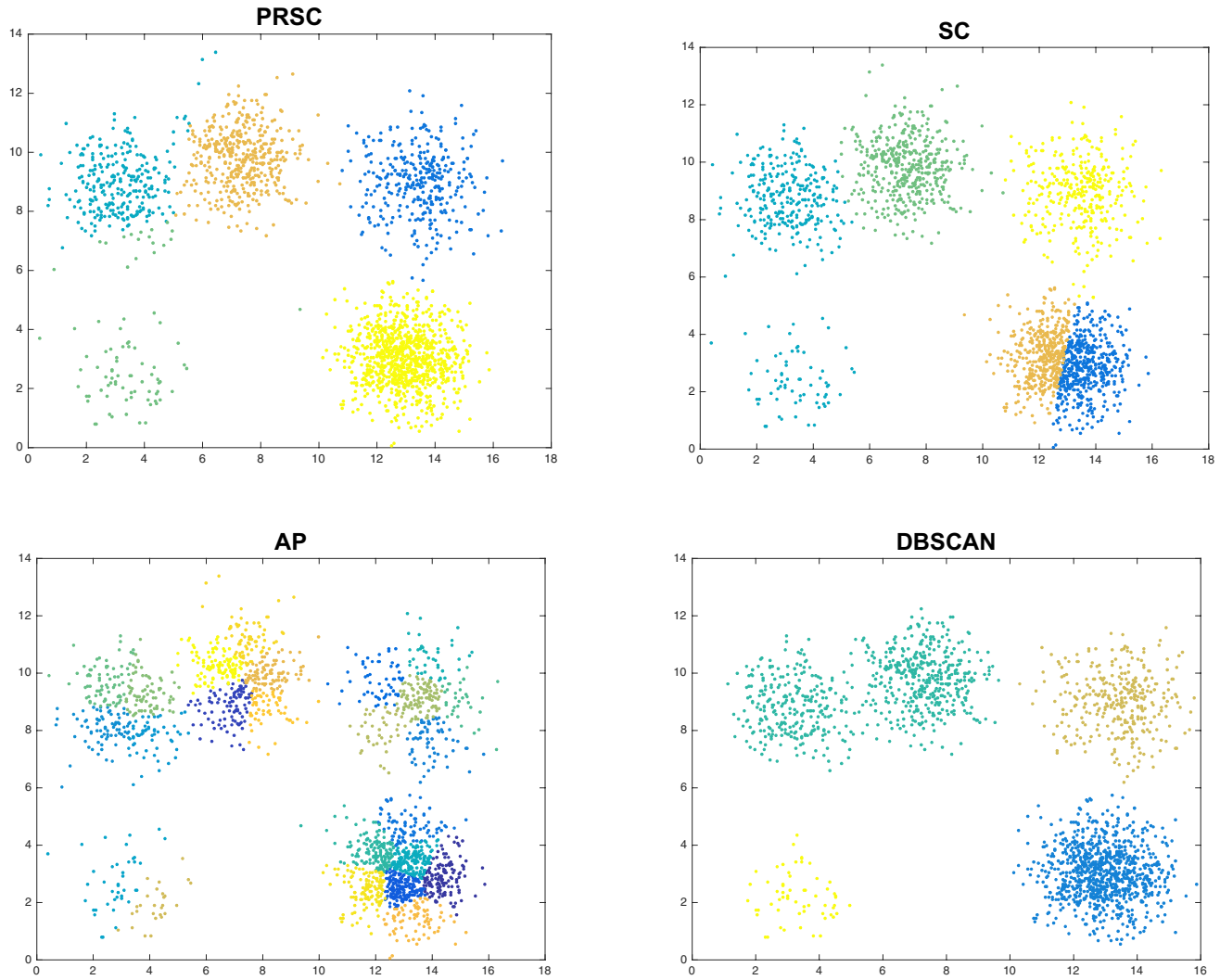


Fig. 6 Clustering results on five cluster

Table 2 Running time of different algorithms on synthetic datasets

	PRSC	SC	AP	DBSCAN
Two cluster	0.24	0.29	1.52	0.39
Three cluster	0.27	0.35	3.18	0.31
Square	0.40	0.59	8.83	0.43
Five cluster	0.61	1.23	26.74	0.69

Table 3 Characteristics of eight UCI datasets

	Instances	Attributes	Clusters
Sonar	208	60	2
Musk	476	166	2
Breast	569	30	2
Statlog	2000	36	6
Madelon	2000	500	2
Segmentation	2310	19	7
Waveform	5000	21	3
Two norm	7400	20	2

Table 4 Clustering accuracy of different algorithms on UCI datasets

	PRSC	SC	DPC
Sonar	54.81	54.81	51.44
Musk	52.73	51.47	50.42
Breast	89.63	89.63	79.09
Statlog	62.55	67.55	70.70
Madelon	58.8	58.2	54.75
Segmentation	58.01	56.75	48.23
Waveform	50.62	50.54	57.96
Two norm	61.59	97.84	53.92

Table 5 Running time of different algorithms on UCI datasets

	PRSC	SC	DPC
Sonar	0.26	0.28	0.23
Musk	0.26	0.31	0.28
Breast	0.32	0.34	0.28
Statlog	1.03	1.31	0.95
Madelon	0.35	1.17	0.88
Segmentation	0.92	1.94	1.13
Waveform	0.85	7.75	5.37
Two norm	0.37	18.1	12.52

Acknowledgements This work is supported by the National Natural Science Foundations of China (No. 61672522, No. 61976216),in part of the Cyan and Blue project of Jiangsu of China(2020).

References

1. Deng T, Ye D, Ma R et al (2020) Low-rank local tangent space embedding for subspace clustering. Inf Sci 508:1–21
2. Zhang Q, Zhu C, Yang L et al (2017) An incremental CFS algorithm for clustering large data in industrial internet of things. IEEE Trans Ind Inf 13(3):1193–1201
3. Pang Y, Ye L, Li X et al (2016) Incremental learning with saliency map for moving object detection. IEEE Trans Circuits Syst Video Technol 28(3):640–651
4. Zhu Y, Ting K, Carman M (2016) Density-ratio based clustering for discovering clusters with varying densities. Pattern Recognit 60:983–997
5. MacQueen J Some methods for classification and analysis of multivariate observations. In: Proceedings of 5th Berkeley symposium on mathematical statistics and probability, vol 1, pp 281–297
6. Frey B, Dueck D (2007) Clustering by passing messages between data points. Science 315(5814):972–976
7. Ester M, Kriegel H, Sander J et al (1996) A density-based algorithm for discovering clusters in large spatial databases with noise. In: Proceedings of the 2nd international conference on knowledge discovery and data mining, vol 226, p 231
8. Cour T, Benezit F, Shi J (2005) Spectral segmentation with multiscale graph decomposition. In: Proceedings of the IEEE computer society conference on computer vision and pattern recognition (CVPR'05), vol 2, pp 1124–1131
9. Chung F, Graham F (1997) Spectral graph theory. American Mathematical Soc., Providence
10. Huang J, Nie F, Huang H (2013) Spectral rotation versus k-means in spectral clustering. In: Proceedings of the twenty-seventh AAAI conference on artificial intelligence
11. Li Z, Nie F, Chang X et al (2018) Rank-constrained spectral clustering with flexible embedding. IEEE Trans Neural Netw Learn Syst 29(12):6073–6082
12. Mehrkanoon S, Alzate C, Mall R et al (2014) Multiclass semisupervised learning based upon kernel spectral clustering. IEEE Trans Neural Netw Learn Syst 26(4):720–733
13. Yang Y, Ma Z, Yang Y et al (2014) Multitask spectral clustering by exploring intertask correlation. IEEE Trans Cybern 45(5):1083–1094
14. Panda R, Kuanar S, Chowdhury AS (2017) Nyström approximated temporally constrained multisimilarity spectral clustering approach for movie scene detection. IEEE Trans Cybern 48(3):836–847
15. Pang Y, Wang S, Yuan Y (2014) Learning regularized LDA by clustering. IEEE Trans Neural Netw Learn Syst 25(12):2191–2201
16. Chen G, Hu J, Peng H et al (2018) A spectral clustering algorithm improved by P systems. Int J Comput Commun Control 13(5):759–771
17. Zelnik-Manor, L, Perona, P, Self-Tuning Spectral Clustering. In: Advances in Neural Information Processing Systems 17 (NIPS, (2004) The MITPress, December 13-18, Vancouver, British Columbia, Canada, pp 1601–1608
18. Wang L, Ding S, Jia H (2019) An improvement of spectral clustering via message passing and density sensitive similarity. IEEE Access 7:101054–101062
19. Wen G, Zhu Y, Cai Z et al (2018) Self-tuning clustering for high-dimensional data. World Wide Web 21(6):1563–1573
20. Zhang H, Cao L (2014) A spectral clustering based ensemble pruning approach. Neurocomputing 139:289–297
21. Cheng D, Nie F, Sun J et al (2017) A weight-adaptive Laplacian embedding for graph-based clustering. Neural Comput 29(7):1902–1918
22. Hu Z, Nie F, Wang R et al (2020) Multi-view spectral clustering via integrating nonnegative embedding and spectral embedding. Inf Fusion 55:251–259
23. Pang Y, Xie J, Nie F et al (2018) Spectral clustering by joint spectral embedding and spectral rotation. IEEE Trans Cybern 50(1):247–258
24. Tautenhain C, Nascimento M (2020) An ensemble based on a bi-objective evolutionary spectral algorithm for graph clustering. Expert Syst Appl 141:112911
25. Liu J, Guo X, Liu Y (2020) Hyperspectral remote sensing image feature extraction based on spectral clustering and subclass discriminant analysis. Remote Sens Lett 11(2):166–175
26. Allab K, Labiod L, Nadif M (2018) Simultaneous spectral data embedding and clustering. IEEE Trans Neural Netw Learn Syst 29(12):6396–6401
27. Wang Y, Duan X, Liu X et al (2018) A spectral clustering method with semantic interpretation based on axiomatic fuzzy set theory. Appl Soft Comput 64:59–74
28. Nataliani Y, Yang M (2019) Powered Gaussian kernel spectral clustering. Neural Comput Appl 31(1):557–572
29. Zhang H, Zhang R, Nie F et al (2019) An efficient framework for unsupervised feature selection. Neurocomputing 366:194–207
30. Ding S, Jia H, Du M et al (2018) A semi-supervised approximate spectral clustering algorithm based on HMRF model. Inf Sci 429:215–228
31. Rodriguez A, Laio A (2014) Clustering by fast search and find of density peaks. Science 344(6191):1492–1496

32. Xu X, Ding S, Du M et al (2018) DPCG: an efficient density peaks clustering algorithm based on grid. Int J Mach Learn Cybern 9(5):743–754
33. Bai L, Cheng X, Liang J et al (2017) Fast density clustering strategies based on the k-means algorithm. Pattern Recognit 71:375–386
34. Du M, Ding S, Jia H (2016) Study on density peaks clustering based on k-nearest neighbors and principal component analysis. Knowl Based Syst 99:135–145
35. Ding S, Du M, Sun T et al (2017) An entropy-based density peaks clustering algorithm for mixed type data employing fuzzy neighborhood. Knowl Based Syst 133:294–313
36. Xie J, Gao H, Xie W et al (2016) Robust clustering by detecting density peaks and assigning points based on fuzzy weighted K-nearest neighbors. Inf Sci 354:19–40
37. Liu R, Wang H, Yu X (2018) Shared-nearest-neighbor-based clustering by fast search and find of density peaks. Inf Sci 450:200–226
38. Xu X, Ding S, Wang L et al (2020) A robust density peaks clustering algorithm with density-sensitive similarity. Knowl Based Syst 200:1–11
39. Xu X, Ding S, Shi Z (2018) An improved density peaks clustering algorithm with fast finding cluster centers. Knowl Based Syst 158:65–74
40. Wang M, Zuo W, Wang Y (2016) An improved density peaks-based clustering method for social circle discovery in social networks. Neurocomputing 179:219–227
41. Meng H, Yuan F, Yan T et al (2019) Indoor positioning of rbf neural network based on improved fast clustering algorithm combined with LM algorithm. IEEE Access 7:5932–5945
42. Li X, Wong K (2019) Evolutionary multiobjective clustering and its applications to patient stratification. IEEE Trans Cybern 49(5):1680–1693
43. Jiang J, Yan X, Yu Z et al (2015) A Chinese expert disambiguation method based on semi-supervised graph clustering. Int J Mach Learn Cybern 6(2):197–204