International Journal of Pattern Recognition
and Artificial Intelligence
Vol. 33, No. 8 (2019) 1950012 (19 pages)
© World Scientific Publishing Company
DOI: 10.1142/S0218001419500125

HaloDPC: An Improved Recognition Method on Halo Node for Density Peak Clustering Algorithm

Jianhua Jiang[*], Wei Zhou[†] and Limin Wang[‡]

Department of Data Science
Jilin University of Finance and Economics
Changchun 130117, P. R. China
[*]jianhuajiang@yahoo.com
[†]1549225062@qq.com
[‡]wlm_new@163.com

Xin Tao

School of Management, Jilin University, 5988 Renming Street
Changchun 130022, P. R. China
459978415@qq.com

Keqin Li

Department of Computer Science
State University of New York
Science Hall 249, New Paltz, New York 12561, USA
lik@newpaltz.edu

Received 7 November 2017
Accepted 8 November 2018
Published 8 January 2019

The density peaks clustering (DPC) is known as an excellent approach to detect some complicated-shaped clusters with high-dimensionality. However, it is not able to detect outliers, hub nodes and boundary nodes, or form low-density clusters. Therefore, halo is adopted to improve the performance of DPC in processing low-density nodes. This paper explores the potential reasons for adopting halos instead of low-density nodes, and proposes an improved recognition method on Halo node for Density Peak Clustering algorithm (HaloDPC). The proposed HaloDPC has improved the ability to deal with varying densities, irregular shapes, the number of clusters, outlier and hub node detection. This paper presents the advantages of the HaloDPC algorithm on several test cases.

Keywords: Clustering algorithm; density peak; anomaly detection; halo node.

[*]Corresponding author.

1. Introduction

The goal of clustering is to separate a series of finite unlabeled objects into different clusters with characteristics of internal homogeneity and external separation. Clustering has been applied in a wide variety of fields, ranging from engineering (machine learning, artificial intelligence, pattern recognition, mechanical engineering, electrical engineering),[2] computer sciences (web mining, spatial database analysis, textual document collection, image segmentation),[14] life and medical sciences (genetics, biology, microbiology, psychiatry, clinic, pathology), to earth sciences (geography, geology, remote sensing), social sciences (sociology, psychology, education),[16] and economics (marketing, business).[7,10,23,27]

Traditional methods of clustering can be broadly categorized into those of hierarchical, partitioning, density-based, model-based, grid-based, and soft-computing.[17,19,21,25] Inspired by DBSCAN,[6] many density-based clustering methods[5,6,20] have been proposed. In the last decade, DBSCAN has had a big impact on the data mining research community due to its capability of discovering clusters with arbitrary shapes and noise detection. Furthermore, DBSCAN is able to detect outliers easily by its parameters of *MinPts* and *Eps*. However, it is also vital when choosing an appropriate density threshold. Expert-defined threshold values are so sensitive; a slightly different threshold setting may result in an entirely different clustering on a dataset.[8]

Rodriguez and Laio proposed a novel density peaks clustering (DPC) algorithm that cluster centers are characterized by a higher density than their neighbors and by a relatively large distance to higher density points.[18] Both the idea of local density maxima from mean-shift[3] and the idea of only one parameter of the distance between data points from K-Medoids[15] are adopted by DPC. Focusing on this method, several researches[1,4,11–13,22,25,26] have been carried out for improving its capabilities.

As illustrated in Table 1, the DPC has limitations in processing irregular shapes[26] and varying densities,[4] and in detecting the number of clusters.[4] Furthermore, an issue about hub node detection needs to be solved as illustrated in Fig. 8(c). In fact, these limitations of DPC are caused by its incapability of processing low density nodes. In the DPC algorithm,[18] halos are adopted to solve low density nodes in two ways: (1) one is *halo generation* that low density nodes are considered as a whole of halos, and (2) the other is *no halo generation* that low density nodes are assigned to its cluster with a simple principle. Inspired by SCAN,[24] halo nodes can be classified

Table 1. Advantages of HaloDPC, K-Means, DBSCAN and DPC (where "×" refers to disable, "$\surd$" refers to able and "∂" is partial)

Algorithms	Sizes	Irregular Shapes	Densities	# of Clusters	Outliers	Hub Nodes
K-Means	×	×	×	×	×	×
DBSCAN	$\surd$	∂	×	×	$\surd$	×
DPC	$\surd$	∂	$\surd$	∂	∂	×
HaloDPC	$\surd$	∂	$\surd$	$\surd$	$\surd$	$\surd$

into hub nodes, anomalies and boundary nodes. Therefore, a new approach is required to improve the capability of halo processing in the DPC algorithm.

This paper tests our proposed Halo node for density peaks clustering (HaloDPC) algorithm on the most popular clustering benchmarks and demonstrates its feasibility. In order to assess its performance, this paper compares HaloDPC with DPC and other algorithms on several UCI datasets. The HaloDPC overcomes the above limitations of DPC with satisfactory results on synthetic datasets. The rest of this paper is organized as follows: in Sec. 2, basic principles of DPC, DBSCAN and SCAN algorithms are described; in Sec. 3, the innovative enhanced DPC algorithm is explained; in Sec. 4, experimental results on synthetic datasets and some UCI datasets are analyzed; in Sec. 5, some discussions are given; the final conclusions are drawn in Sec. 6.

2. Related Work

The proposed HaloDPC algorithm is based on DPC[18] and inspired by DBSCAN[6] and SCAN[24]. Therefore, brief reviews of the three algorithms should be given in the following sections.

2.1. *DPC: A density peaks clustering approach*

The DPC algorithm is based on the idea that cluster centers are characterized by a higher density than their neighbors and by relatively large distance to higher density points.[18] Cutoff distance d_c is the only parameter in this method. For each data point x_i, it computes two quantities: its local density ρ_i and its distance δ_i to higher density points.

$$d_{ij} = \text{distance}(x_i, x_j), \tag{1}$$

where the distance can be measured by distance functions, e.g. Euclidean distance.

$$\rho_i = \sum_{i=1} \chi \times (d_{ij} - d_c), \tag{2}$$

where $\chi(x) = 1$, if $x < 0$, otherwise $\chi(x) = 0$. As a rule of thumb, one can choose d_c so that the average number of neighbors is around 1% to 2% of the total number of points in a dataset.[18]ρ_i, similar to *MinPoints* in DBSCAN,[6] is defined as the number of neighbor points to point x_i in Eq. (2). Another local density of point x_i is presented in Eq. (3), as follows:

$$\rho_i = \sum_j \exp\left(\frac{-d_{ij}^2}{d_c^2}\right), \tag{3}$$

$$\delta_i = \min_{j:\rho_j > \rho_i} d_{ij}. \tag{4}$$

Note that δ_i is measured by computing the minimum distance between point x_i and any other points with relatively high density.

The DPC algorithm can be summarized from Rodriguez *et al.*[18] and Du *et al.*[4] shown in Algorithm 1.

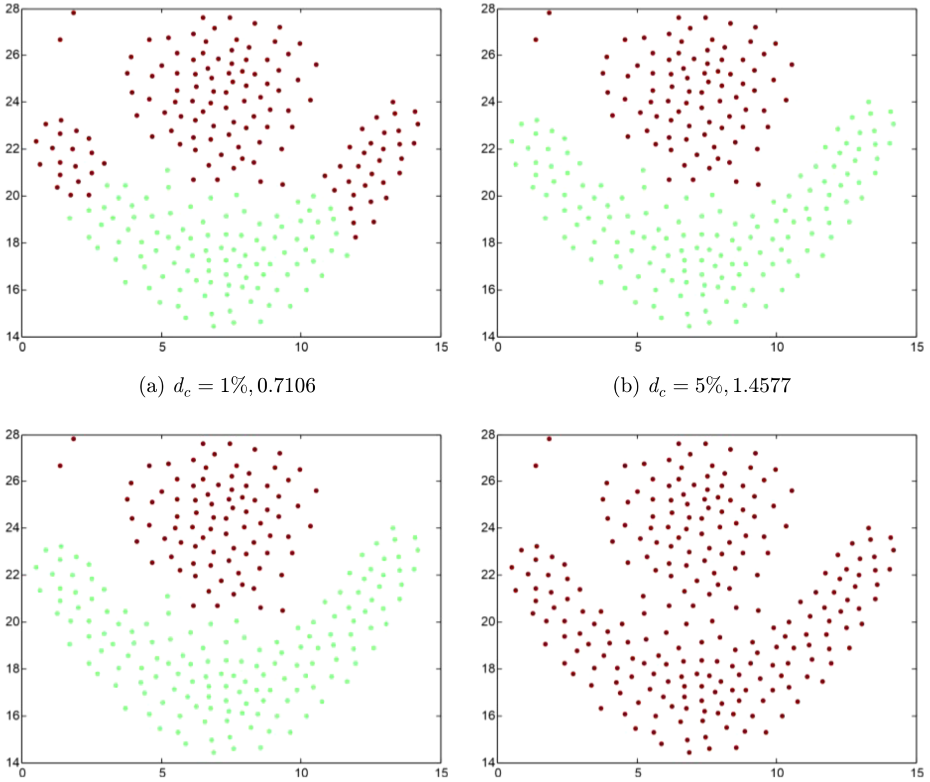
In this algorithm, a border region for each cluster is found, which can be defined as the set of points assigned to one cluster but being within a distance d_c from data points in another cluster. Then the point with highest density is found within the border region, and is denoted by ρ_b. The points with densities higher than ρ_b are considered as the core of the cluster. The others are considered as the halo of the cluster (can be considered as noise area).[18]

As illustrated in Algorithm 1, lower density nodes can be solved in two ways. One is to label these lower density nodes as halo nodes without classification, and the other is to assign them into different clusters based on the value of δ_i. However, it is known from Figs. 4(c), 5(c) and 8(c), that it is hard to classify the low density nodes into any categories among boundary nodes, hub nodes, outliers, or even new clusters.

As illustrated in Fig. 1, these outliers in the dataset of *Flame* is in the top left corner. After experiments with different d_c values, it is obvious in Fig. 1 that the DPC algorithm is not able to detect these two outliers correctly. The major reason is that it is difficult to determine the values of d_c and δ_i in the *Flame* dataset. Naturally, an extension of DPC on the processing of low density nodes should be proposed to enhance its performance.

Algorithm 1. The DPC algorithm

```
Data: Initial nodes $X \in R_{N \times M}$, $d_c$
Result: The label vector of cluster index: $y \in R_{N \times M}$
Step 1: Calculate $d_c$
begin
    Calculate $d_{ij}$ from $R_{N \times M}$ based on Eq. (1);
    Sort $d_{ij}$ in an ascending order;
    Determine $d_c$ by finding value of 1% to 2% position in the order above.
Step 2: Detect cluster centroids by density peaks
begin
    Calculate $\rho_i$ based on Eqs. (2) or (3);
    Calculate $\delta_i$ based on Eq. (4);
    Put all nodes based on $\rho$ in a descending order;
    Detect cluster centroids with relatively higher $\rho$ and $\delta$.
Step 3: Assign each node to different clusters
begin
    Detect halo nodes based on their densities;
    Determine their affiliations of relatively higher density nodes by $\delta_i$ in
    each cluster.
```



(a) $d_c = 1\%, 0.7106$

(b) $d_c = 5\%, 1.4577$

(c) $d_c = 10\%, 2.1213$

(d) $d_c = 25\%, 3.7483$

Fig. 1. Failure outlier detection by DPC on *Flame* dataset with varying d_c.

2.2. *DBSCAN: A density-based clustering approach with noise*

DBSCAN is an efficient clustering algorithm for reasons that (1) it is significantly effective in discovering clusters of arbitrary shapes, (2) it is efficient in cluster processing, (3) it is good for outlier detection.[6] It is inspired by an intuition that within each cluster, there are points whose densities are considerably higher than those outside of the cluster. Furthermore, the density within the noise area is much lower than the one in any of the clusters.

The DBSCAN[6] has two major advantages. It can extract arbitrary-shaped clusters and is also able to detect outliers. The structure and the features of core node chain have contributed to extracting arbitrary-shaped clusters and detecting outliers. Therefore, the inspiration is that its *reachability* or *connectivity* is a basic function to extract complex clusters.

2.3. *SCAN: A structural network clustering approach*

The SCAN[24] algorithm is a well-known structural network clustering approach to discover underlying community structures in complex networks. It has the ability to detect clusters, hubs and outliers in complex networks with the following features[24]:

(1) It detects clusters, hubs and outliers by criteria of the structure and the connectivity of the vertices;
(2) It is fast with a running time of $\mathrm{O}(m)$, on a network with n vertices and m edges.

Similar to the DBSCAN algorithm, the SCAN algorithm inherits the idea of node density, such as concepts of *direct-density-reachability*, *density-reachability* and *density-connectivity*.

In the SCAN algorithm, each nonmember vertice v can be classified into *hub nodes* and *outliers* based on its network structural similarity and cluster IDs of neighbor nodes. Therefore, as inspired by the SCAN algorithm, there is a difference between outliers and hub nodes due to their different locations in datasets. Furthermore, both outliers and hub nodes can be classified by their relationship with their neighbor nodes.

3. Halo Processing of DPC

HaloDPC inherits the strengths of *centroid detection* of DPC, *density-connectivity* of DBSCAN and *network structural similarity* of SCAN, and it is capable of processing datasets with varying densities or sizes, with irregular shapes, and also capable of detecting the number of clusters, outliers and hub nodes. Halos are defined as a set of relatively low density nodes where outliers, boundary nodes, hub nodes and new clusters may exist. HaloDPC assumes that node classification in halos is based on *connectivity analysis* and *network structural analysis*. As illustrated in Algorithm 2, HaloDPC includes three major steps: halo node generation, halo network generation and halo classification.

3.1. *Halo node generation*

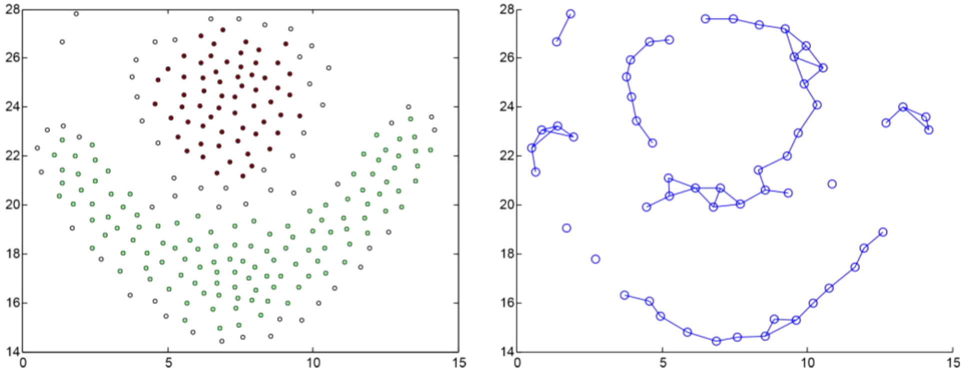
Halo nodes are generated by finding low density nodes from the result of the DPC processing. All of these halo nodes will be collected as a halo node set in which each node is labeled with a predicted cluster ID based on the DPC. As illustrated in Figs. 2(a) and 3(a), halos in datasets *Flame* and *Pathbase* are labeled with blank circles.

3.2. *Halo network generation*

Halo network can be defined as a network composed of connective nodes from the halo node set. *Halo network generation* is based on *density-connectivity*. The main idea of *density-connectivity* is that there will be a connected line if the distance between $node_i$ and $node_j$ is shorter than the cutoff distance d_c. Obviously, different halo network structures can be generated from *density-connectivity*. As illustrated in

Algorithm 2. The HaloDPC algorithm

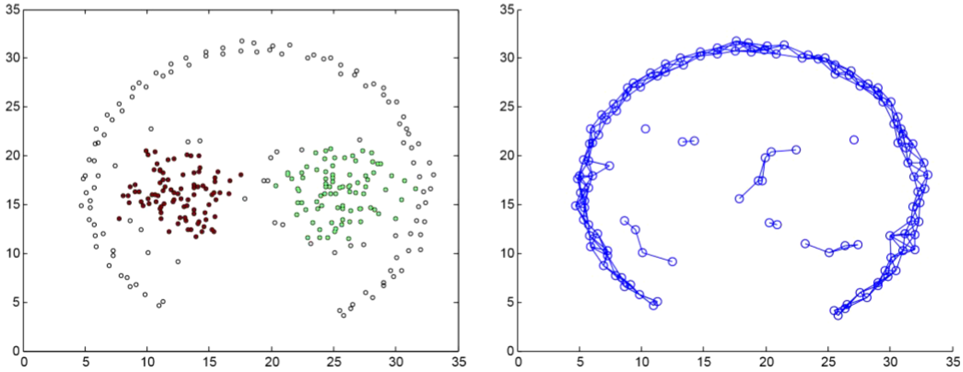
```
Data: $Node_{DPC}$, $dc$, $\rho$, $\varphi$
Result: All halo nodes classified
/* STEP 1. Halo Node Generation                                      */
for $node_i \in node_{DPC}$ do
    if the $\rho$ of $node_i \leq \rho$ then
        Put $node_i$ into the array of $HaloNode$;
        $HaloNode_k = node_i$;
/* STEP 2. Halo Network Generation                                   */
for $hNode_i \in HaloNode$ do
    /* STEP 2.1. Determine statistics for each halo node             */
    Count the number of core nodes of $hNode_i$ with the distance of $d_c$;
    /* STEP 2.2. Form halo networks                                  */
    if $DensityConnetivity(node_i, \exists HaloNetwork \in HaloNetwork)$ then
        Put $hNode_i$ to $HaloNetwork$;
    else
        Generate another $HaloNetwork$;
        Put $HaloNetwork$ into $HaloNetworks$;
/* STEP 3. Halo Classification                                       */
for $HaloNetwork \in HaloNetworks$ do
    /* STEP 3.1. Label hub nodes and generate a new cluster          */
    if $numberOfClusters(HaloNetwork) \geq 2$ then
        /* STEP 3.2. Classification in a complex $HaloNetwork$        */
        if $\frac{numberOfCoreNodes(HaloNetwork)}{numberOfHaloNodes(HaloNetwork)} \leq \varphi$ then
            /* STEP 3.2.1. Generate a new cluster                    */
            Assign a new $ClusterID$ to $HaloNetwork$;
        else
            /* STEP 3.2.2. Detect hub nodes                          */
            Generate another $HaloNetwork$;
            while $DensityConnectivity(x, y) \mid x, y \in HaloNetwork$ do
                if the $clusterID$ of $HaloNode_x \neq HaloNode_y$ then
                    Remove $x$, $y$ from $HaloNetwork$;
                    Label $x$, $y$ as hub nodes;
                    /* STEP 3.2.3. Generate sub halo networks        */
                    Divide $HaloNetwork$ into different sub halo networks;
                    Put these sub halo networks into $HaloNetworks$;
                else
                    /* STEP 3.3. Classification in a simple
                        $HaloNetwork$                                 */
                    if $distance(\forall coreNode, \forall HaloNode) \geq d_c$ then
                        /* STEP 3.3.1. Outlier prediction            */
                        Resign the $clusterID$;
                        All $HaloNode \in HaloNetwork$ are outliers;
                    else
                        /* STEP 3.3.2. Boundary node prediction      */
                        Do not change the $clusterID$ of $HaloNetwork$;
```



(a) Halo nodes

(b) Halo networks

Fig. 2. Halo nodes and halo networks of *Flame* with $d_c = 1.3865$.



(a) Halo nodes

(b) Halo networks

Fig. 3. Halo nodes and halo networks of *Pathbase* with $d_c = 2.55$.

Figs. 2(b) and 3(b), nodes that are relatively adjacent can be connected together depending on the value of δ_i. Therefore, network structure, based on the principles of the SCAN algorithm, is good for finding hub nodes.[24] Furthermore, it is efficient in finding new underlying clusters in Fig. 3(b), provided that new clusters are in existence.

3.3. *Halo classification*

Halo classification is the prediction of halo node types based on *halo networks* by characteristics including *numberOfClusters*, ratio of core nodes and *density-connectivity*. Both outliers and boundary nodes are in existence if *numberOf Clusters*=1. However, new clusters or hub nodes may exist simultaneously in more than two clusters. Ratio of core nodes is an important factor to distinguish new clusters from boundary nodes. The approach of halo classification is described in detail in Step 3 of Algorithm 2.

4. Simulation Experiment and Analysis

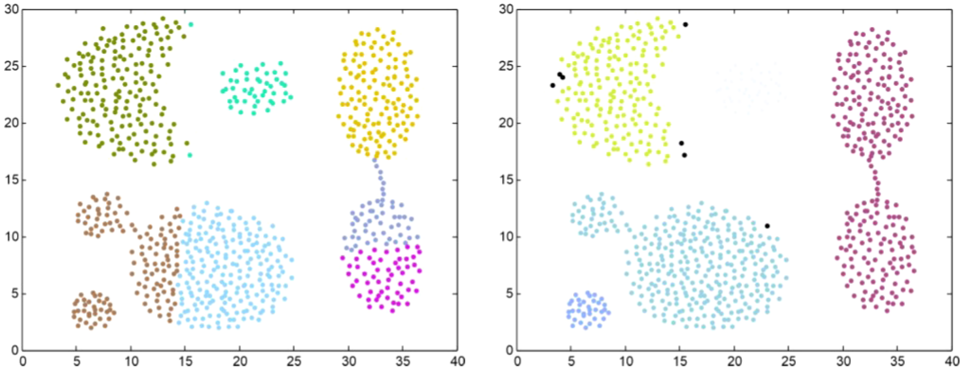
To test the feasibility and effectiveness of the proposed HaloDPC algorithm, this paper compares it with K-Means,[9] DBSCAN,[6] AP[7] and DPC[18] on synthetic datasets listed in Table 2.

4.1. *Detecting clusters with varying sizes*

As illustrated in Figs. 4, 5 and 8, the AP algorithm has some difficulties in handling datasets with varying sizes such as *Aggregation*, *Pathbase* and *Flame*, and although both DBSCAN and DPC are able to detect correct clusters in *Aggregation* and *Flame* datasets, they fail in dealing with the path-based dataset such as *Pathbase*. However, the proposed HaloDPC is found capable of handling these complex datasets with varying sizes.

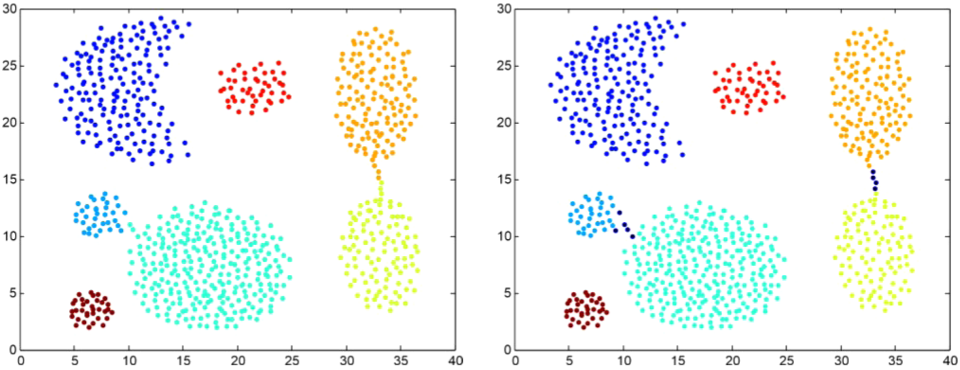
Table 2. Five different types of datasets.

Datasets	Nodes	Dimensions	Clusters
Flame	240	2	2
Aggregation	788	2	7
Spiral	312	2	3
Pathbase	312	2	3
D	87	2	3



(a) K-Means, $k = 7$ (b) DBSCAN, $MinPts = 3$

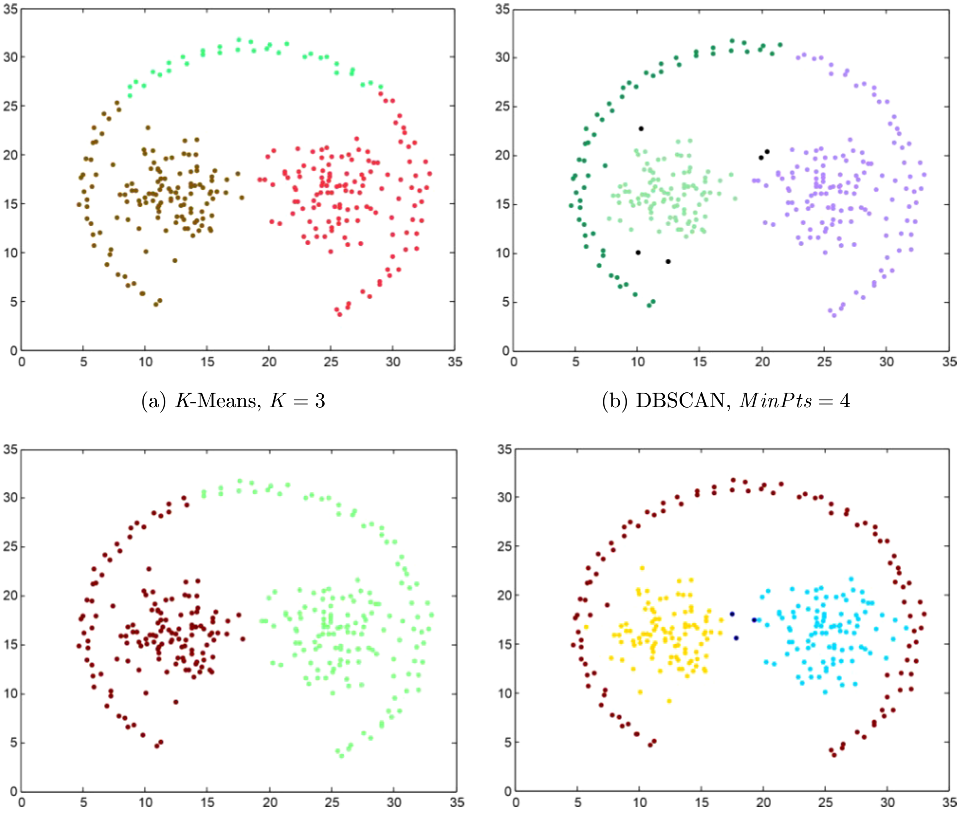
Fig. 4. Clustering result of *Aggregation*.



(c) DPC, $d_c = 1.8601$

(d) HaloDPC, $d_c = 1.8601$

Fig. 4. (*Continued*)



(a) K-Means, $K = 3$

(b) DBSCAN, $MinPts = 4$

(c) DPC, $d_c = 2.55$

(d) HaloDPC, $d_c = 2.55$

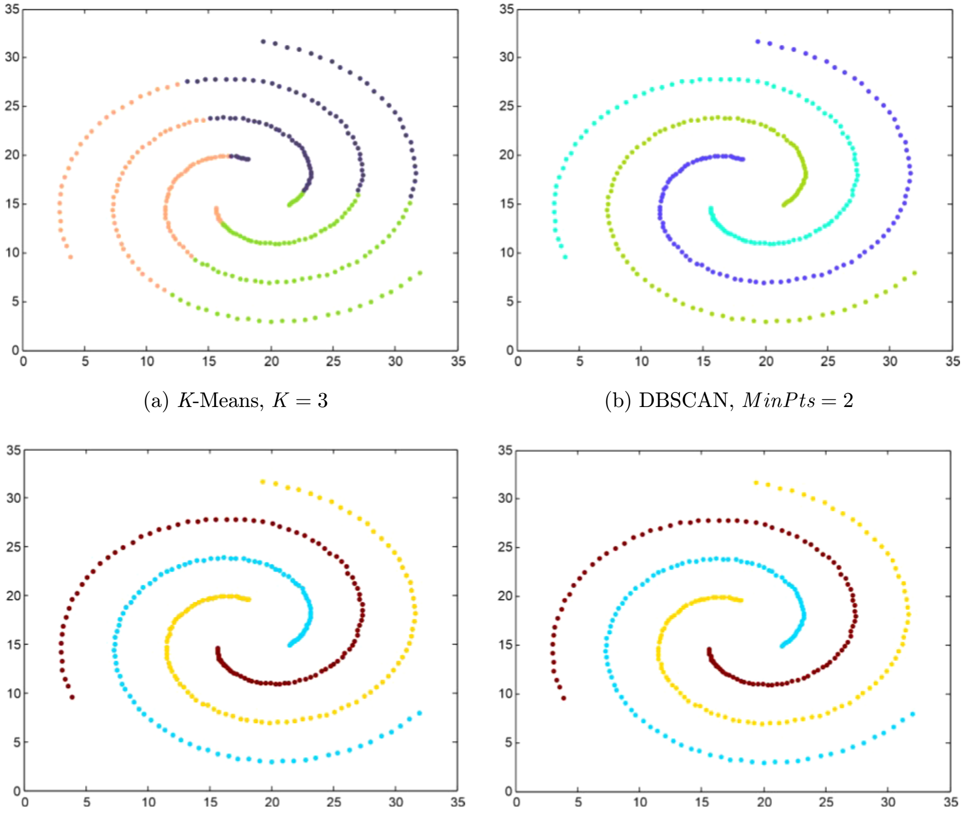
Fig. 5. Clustering result of *Pathbase*.

4.2. *Detecting clusters with irregular shapes*

Datasets of *Pathbase*, *Spiral* and *Flame* are adopted to test its capability to deal with irregular shapes. As illustrated in Figs. 5, 6 and 8, the AP algorithm is unable to handle these three datasets, DBSCAN and DPC can process datasets of *Flame* and *Spiral* but cannot *Pathbase*. Only the proposed HaloDPC has the ability to handle all the three complex datasets with irregular shapes.

4.3. *Detecting clusters with varying densities*

A dataset of D is utilized to evaluate its capability to cope with varying densities. As shown in Fig. 7, the AP algorithm has problems in processing dataset with varying densities, while other algorithms of DBSCAN, DPC and HaloDPC are able to handle this kind of dataset with a good result.



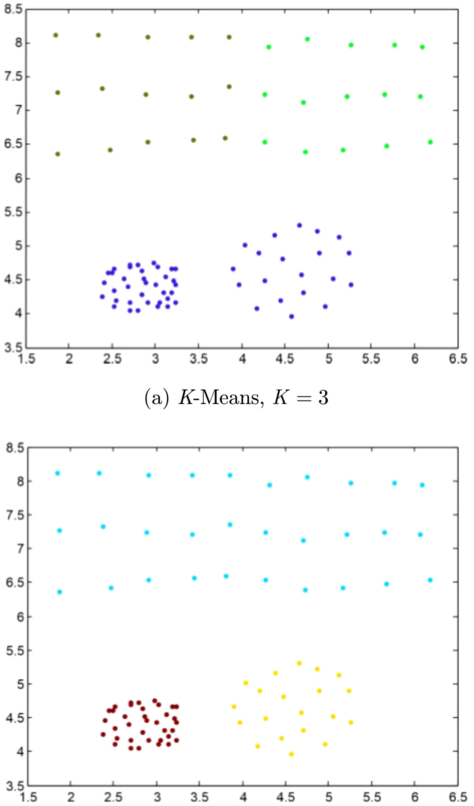
(a) K-Means, $K = 3$

(b) DBSCAN, $MinPts = 2$

(c) DPS, $d_c = 2.5812$

(d) HaloDPC, $d_c = 2.5812$

Fig. 6. Clustering result of *Spiral*.



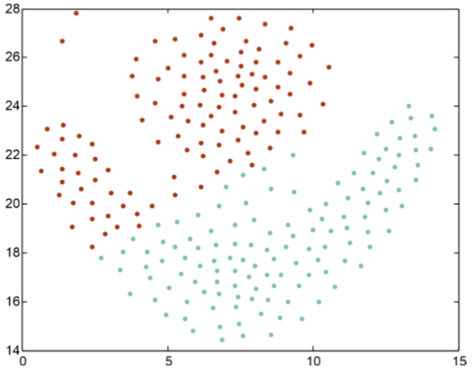
(a) K-Means, $K = 3$

(b) DBSCAN, $MinPts = 5$

(c) DPC, $d_c = 0.9441$

(d) HaloDPC, $d_c = 0.9441$

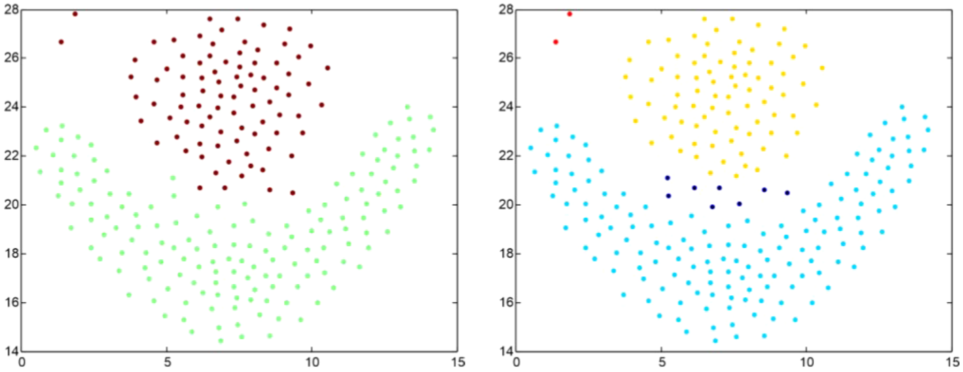
Fig. 7. Clustering with varying densities on D dataset.



(a) K-Means, $K = 2$

(b) DBSCAN, $MinPts = 2.5$

Fig. 8. Clustering result of *Flame*.



(c) DPC, $d_c = 1.3124$

(d) HaloDPC, $d_c = 1.3124$

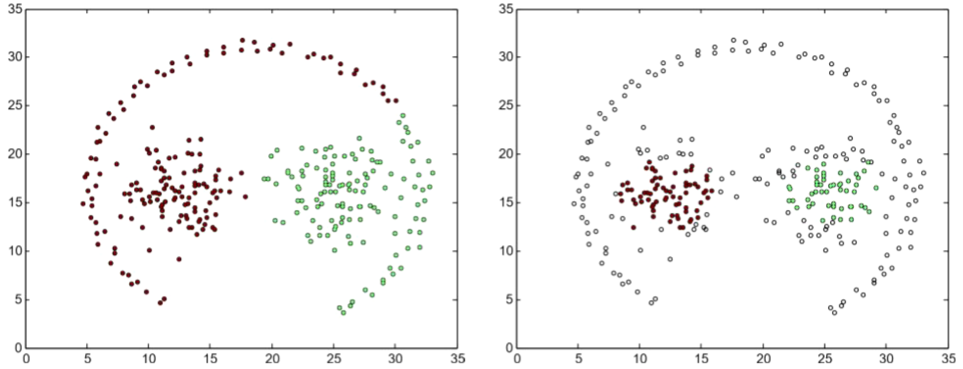
Fig. 8. (*Continued*)

4.4. *Detecting the number of clusters*

Datasets of *Aggregation* and *Pathbase* are of great use to access the ability of detecting the number of clusters. As shown in Figs. 4 and 5, both DPC and DBSCAN have the ability to detect number of clusters in *Aggregation* but not *Pathbase*. The DPC algorithm cannot make certain the number of clusters with different d_c values illustrated in Fig. 9. However, the proposed HaloDPC is able to detect the number of clusters in datasets of *Aggregation* and *Pathbase*.

4.5. *Detecting clusters with outliers and hub nodes*

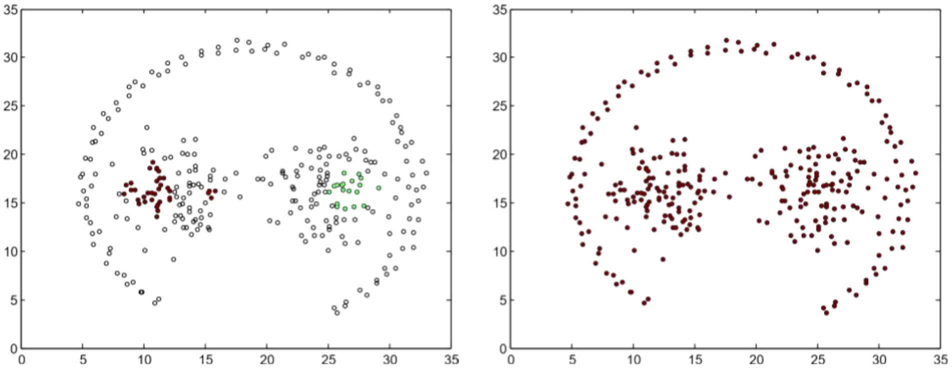
Datasets of *Aggregation*, *Pathbase* and *Flame* are selected to test the ability of detecting outliers and hub nodes. For outlier detection, both DBSCAN and HaloDPC have the ability to detect outliers but DPC cannot detect them in datasets



(a) $d_c = 1.5403$

(b) $d_c = 4.9010$

Fig. 9. Clustering dataset of *Pathbase* by DPC with different d_c values.



(c) $d_c = 7.1309$

(d) $d_c = 10.4519$

Fig. 9. (*Continued*)

of *Flame* and *Pathbase*. In the top left corner of Fig. 8(c), the two outliers with different d_c values cannot be detected by DPC. In Fig. 5(c), the DPC is not able to find outliers in low density nodes. However, the proposed HaloDPC is able to find outliers correctly in all the three datasets. For hub node detection, hub nodes can be defined as nodes that belong to more than two clusters. It is found that only the proposed HaloDPC has the ability to find hub nodes in datasets of *Aggregation*, *Pathbase* and *Flame* in Figs. 4(d), 5(d) and 8(d).

5. Discussion

Density-based clustering approaches have attracted extensive study and researches in recent years. The application of these approaches has played an important role in many subject and fields. Compared with other kinds of clustering approaches, density-based clustering approaches have the advantage of extracting arbitrary-shaped clusters with low computing complexity. Density-based clustering approaches can be classified into *connectivity-oriented* methods and *peak-oriented* methods. *Connectivity-oriented* methods, such as DBSCAN, is an excellent approach, and the DPC algorithm in *peak-oriented* method. The DPC algorithm has the ability to find arbitrary-shaped clusters efficiently. Currently, many modified processing methods of density peaks have been put into application to handle many kinds of datasets with complicated structure. However, halo processing is yet the biggest unsolved problem in the DPC algorithm. As illustrated in Fig. 9, no matter how d_c values change, it is impossible to generate three reasonable clusters simultaneously, for the underlying new cluster can be generated only by HaloDPC.

With principles of the DPC, it is inefficient to process low-density nodes. Based on *no halo generation* in the DPC, each node should be categorized into the cluster that its nearest node of relatively higher density belongs to. But there follows a contradiction that outliers, boundary nodes, hub nodes and new clusters should be determined by their neighbor nodes, rather than by the relatively higher density nodes. The lower the density becomes, the higher the possibility of difference exists between neighbor nodes and relatively higher density nodes. Therefore, how to classify these low density nodes in the DPC is really a big challenge.

5.1. *Analysis of processing datasets with varying sizes*

As illustrated in Figs. 4(c) and 4(d), HaloDPC inherits the advantage of processing datasets with varying sizes. In Algorithm 2 on Sec. 3, it is found that the HaloDPC is an extension for post-processing of low-density nodes in the DPC. With principles of the DPC, the size of a cluster is determined by δ_i.

5.2. *Analysis of processing datasets with irregular shapes*

Density-based clustering algorithms, such as DPC, DBSCAN and HaloDPC, have the advantage of clustering irregular shaped datasets, which is achieved by utilizing density relationship among nodes, as described in Algorithms 1 and 2 and proved in Figs. 6 and 8. However, the DPC is not able to find new clusters with low density nodes or halo nodes. In Sec. 4.2, it shows that only the proposed HaloDPC is able to handle the complex and irregular shaped dataset of *Pathbase*.

5.3. *Analysis of processing datasets with varying densities*

The ability to process datasets with varying densities is an essential part of a clustering algorithm. The AP algorithm adopts distance-based approach to determine clusters, but it cannot adjust its decision rules in an environment of varying densities. The DBSCAN algorithm adopts *MinPts* and *Eps* to detect densities in a dataset. However, it is difficult to make a global parameter to satisfy a dataset of varying densities. In density-peak-oriented clustering algorithm, such as DPC, a node is assigned to the same cluster as its neighbors with relatively higher density. The proposed HaloDPC has the capability of processing datasets with varying densities, since that it inherits the basic aggregating approach of DPC. Even more, the HaloDPC utilizes halo networks to improve its capability of processing low density nodes.

5.4. *Analysis of detecting the number of clusters*

It is difficult to find the number of clusters with irregular shapes and varying densities from a dataset. Both DBSCAN and DPC have problems in detecting the number of clusters in the dataset of *Pathbase* illustrated in Figs. 5(b) and 5(c). DBSCAN will generate a new cluster if there is a core node. However, there is no core node in the long, thin and circled cluster in the *Pathbase* dataset. Similar to the DBSCAN, the DPC assumes that a cluster has a relatively higher density centroid, but in fact, the *Pathbase* dataset has a long, thin and circled cluster without centroids. For low density nodes, they linger among boundary nodes, hub nodes, outliers, and can be regarded as even new clusters. The HaloDPC assumes that there is no centroid if new clusters are generated with low density nodes, and it forms new clusters by classifying halo networks in Algorithm 2 in Sec. 3.3.

5.5. *Analysis of detecting outliers and hub nodes*

In the DPC, both outliers and hub nodes are regarded as halo nodes as shown in Algorithm 1. As illustrated in Fig. 1, the two outliers in the left top corner cannot be detected by any d_c value. Therefore, the DPC has a weakness of detecting outliers in the *Flame* dataset, which further demands that the capability of detecting outliers and hub nodes should be improved. *Halo network* is adopted to analyze the difference between outliers and hub nodes. In Algorithm 2, hub nodes can be detected from halo networks that have more than two different *ClusterIDs*, and outliers can be found by their neighbor distance. After improvement on DPC halo processing, the HaloDPC is able to detect outliers and hub nodes correctly as illustrated in Figs. 4(d), 5(d) and 8(d).

6. Conclusion

This paper proposes an innovative clustering algorithm of density peaks, with the enhancement of halo processing (HaloDPC) which combines advantages of *peaks-oriented* density methods, *connectivity-oriented* density methods and *structure-oriented* classification methods for low-density node processing. The advantage of HaloDPC algorithm lies in its capability of extracting arbitrary-shaped clusters with varying densities in many complicated datasets. There are two major steps in HaloDPC: the first is to determine clusters by *no halos generation* from the DPC, and the second is to post-process clusters by *density connectivity* and *network structural similarity* for relatively low density nodes in each cluster. In density-based clustering approaches, it is difficult to process a dataset with clusters of varying densities, especially when processing low-density nodes that can be ambiguously classified into outliers, boundary nodes, hub nodes and even new clusters. Furthermore, *connectivity-oriented* density methods are able to find outliers easily due to their nonreachability, but they still have problems in detecting boundary nodes and hub nodes. By contrast, HaloDPC not only puts forward a reasonable halo processing method but it also generates high quality results, which is highly competitive compared with other density-based algorithms.

In limitations of the DPC algorithm, HaloDPC is proposed by the inspiration from DBSCAN and SCAN algorithms. HaloDPC is designed aiming at improving the capability of processing low density nodes by adoption of *halo networks, network structural similarity, adjacent distance* and *ratio of core nodes.* After comparison with experiments on some synthetic datasets, such as *Flame, Pathbase, Spiral* and

Aggregation, the proposed HaloDPC has improved the capability of DPC in processing complex datasets with irregular shapes, and in detecting outliers and hub nodes.

In the future, the HaloDPC algorithm will be applied in other areas, such as financial data analysis. Furthermore, the HaloDPC will be extended to network structural datasets, such as supply chain networks.

Acknowledgments

The authors are grateful for the financial support by the National Science Foundation of China (No. 61572225), the Foundation of Jilin University of Finance and Economics (No. 2018Z05), and the Natural Science Foundation of the Science and Technology Department of Jilin Province (No. 20180101044JC).

References

1. E. Aksehirli, B. Goethals and E. Müller, Efficient cluster detection by ordered neighbourhoods, in *Big Data Analytics and Knowledge Discovery*, eds. S. Madria and T. Hara, DaWak 2015, Lecture Notes in Computer Science, Vol. 9263, Springer, Cham.
2. H. Cecotti, Hierarchical k-nearest neighbor with GPUs and a high performance cluster: Application to handwritten character recognition, *Int. J. Pattern Recognit. Artif. Intell.* **31**(2) (2017) 1–24.
3. Y. Cheng, Mean shift, mode seeking, and clustering, *IEEE Trans. Pattern Anal. Mach. Intell.* **17**(8) (1995) 790–799.
4. M. Du, S. Ding and H. Jia, Study on density peaks clustering based on k-nearest neighbors and principal component analysis, *Knowl. Based Syst.* **99** (2016) 135–145.
5. L. Ertoz, M. Steinbach and V. Kumar, Finding clusters of different sizes, shapes, and densities in noisy, high dimensional data, *SIAM Int. Conf. Data Mining* (CA, USA, 1–3 May 2003), pp. 47–58.
6. M. Ester, H. Kriegel and X. Xu, A density-based algorithm for discovering clusters in large spatial databases with noise, in *Proc. Int. Conf. Knowledge Discovery and Data Mining* 2–4, August 1996, Port Land, Dregon, pp. 264–323.
7. B. J. Frey and D. Dueck, Clustering by passing messages between data points, *Science* **315**(5814) (2007) 972–976.
8. J. Han, M. Kamber and J. Pei, Data mining: Concepts and techniques, *Data Mining Concepts Models Methods and Algorithms*, 3rd edn. (Morgan Kaufmann Publishers, 2011).
9. J. A. Hartigan and M. A. Wong, A K-means clustering algorithm, *Appl. Stat.* **28**(1) (1979) 100–108.
10. E. R. Hruschka, R. J. G. B. Campello, A. A. Freitas and A. C. P. L. F. Carvalho, A survey of evolutionary algorithms for clustering, *IEEE Trans. Syst. Man Cybernet.* **39**(2) (2009) 133–155.
11. J. Jiang, D. Hao, Y. Chen, M. Parmar and K. Li, GDPC: Gravitation-based density peaks clustering algorithm, *Phys. A* **502** (2018) 345–355.
12. J. Jiang, Y. Chen, D. Hao and K. Li, DPC-LG: Density peaks clustering based on logistic distribution and gravitation, *Phys. A* **514** (2019) 25–35.
13. J. Jiang, X. Tao and K. Li, DFC: density fragment clustering without peaks, *J. Intell. Fuzzy Syst.* **34**(1) (2018) 525–536.

14. X. L. Li, G. S. Cui and Y. S. Dong, Graph regularized non-negative low-rank matrix factorization for image clustering, *IEEE Trans. Cybernet.* **47**(11) (2016) 3840–3853.
15. H. S. Park and C. H. Jun, A simple and fast algorithm for K-medoids clustering, *Expert Syst. Appl.* **36**(2) (2009) 3336–3341.
16. X. Qi, R. Luo, E. Fuller, R. Luo and C. Q. Zhang, Signed quasi-clique merger: A new clustering method for signed networks with positive and negative edges, *Int. J. Pattern Recognit. Artif. Intell.* **30**(3) (2016) 1–20.
17. X. Z. Qian, J. Deng, H. Qian and Q. Wu, An efficient density biased sampling algorithm for clustering large high-dimensional datasets, *Int. J. Pattern Recognit. Artif. Intell.* **29**(8) (2015) 1–17.
18. A. Rodriguez and A. Laio, Clustering by fast search and find of density peaks, *Science* **344**(6191) (2014) 1492–1496.
19. L. Rokach, A survey of clustering algorithms, *Data Min. Knowl. Discov. Handbook* **16**(3) (2009) 269–298.
20. J. Sander, M. Ester, H. Kriegel and X. Xu, Density-based clustering in spatial databases: The algorithm gdbscan and its applications, *Data Mining Knowl. Discov.* **2**(2) (1998) 169–194.
21. N. Sharet and I. Shimshoni, Analyzing data changes using mean shift clustering, *Int. J. Pattern Recognit. Artif. Intell.* **30**(7) (2016) 1–29.
22. M. Wang, W. Zuo and Y. Wang, An improved density peaks-based clustering method for social circle discovery in social networks, *Neurocomputing* **179** (2016) 219–227.
23. R. Xu and D. C. WunschII, Survey of clustering algorithms, *IEEE Trans. Neural Networks* **16**(3) (2005) 645–678.
24. X. Xu, N. Yuruk, Z. Feng and T. A. J. Schweiger, SCAN: A structural clustering algorithm for networks, *ACM SIGKDD Int. Conf. Knowledge Discovery and Data Mining*, 2007, pp. 824–833.
25. W. Zang, L. Ren, W. Zhang and X. Liu, Automatic density peaks clustering using DNA genetic algorithm optimized data field and Gaussian process, *Int. J. Pattern Recognit. Artif. Intell.* **31**(8) (2017) 1–28.
26. W. Zhang and J. Li, Extended fast search clustering algorithm: Widely density clusters, no density peaks, *Computer Science* **5**(7) (2015) 1–17.
27. J. Zhao, J. Sun, Y. Zhai, Y. Ding, C. Wu and M. Hu, A novel clustering-based sampling approach for minimum sample set in big data environment, *Int. J. Pattern Recognit. Artif. Intell.* **32**(2) (2018) 1–20.

Jianhua Jiang is an Associate Professor of Computer Science. His current research interests include cloud computing, data mining, operation management and finance risk. He has published over 30 research papers, and has received 3 best paper awards.

Wei Zhou received the B.A. degree in Electronical Commerce from Jilin University of Finance and Economics, China, in 2017. His current research interests include data mining and e-commerce.

Limin Wang received a Master's degree and a Doctorate in Computer Science and Technology from Jilin University in 2004 and 2007, respectively. She was a ACM member, member of China computer society CCF, executive council member of society, in Jilin province at the 7th International Conference of the Electronic Commerce and Electronic Government Affairs Program Committee, new century excellent talents in Jilin Province Colleges and Universities. She has published more than 70 papers.

Xin Tao is a PH.D candidate from Jilin University major in Information Science. His current research includes data-mining, intelligent algorithm, think tank research, Knowledge aggregation and social media analyzing.

Keqin Li is a SUNY Distinguished Professor of Computer Science. His current research interests include parallel computing and high-performance computing, distributed computing, energy-efficient computing and communication, heterogeneous computing systems, cloud computing, big data computing, CPU-GPU hybrid and cooperative computing, multicore computing, storage and file systems, wireless communication networks, sensor networks, peer-to-peer file sharing systems, mobile computing, service computing, Internet of things and cyberphysical systems. He has published over 410 journal articles, book chapters, and refereed conference papers, and has received several best paper awards. He is currently or has served on the editorial boards of IEEE Transactions on Parallel and Distributed Systems, IEEE Transactions on Computers, IEEE Transactions on Cloud Computing, Journal of Parallel and Distributed Computing. He is an IEEE Fellow.