DFC: Density Fragment Clustering without Peaks

Jianhua Jiang\textsuperscript{a}, Xing Tao\textsuperscript{b} and Keqin Li\textsuperscript{c,*}
\textsuperscript{a}School of Management Science and Information Engineering, Jilin University of Finance and Economics, Changchun, China
\textsuperscript{b}School of Management, Jilin University, Changchun, China
\textsuperscript{c}Department of Computer Science, State University of New York, New Paltz, NY, USA

Abstract. The density peaks clustering (DPC) algorithm is a novel density-based clustering approach. Outliers can be spotted and excluded automatically, and clusters can be found regardless of the shape and of dimensionality of the space in which they are embedded. However, it still has problems when processing a complex data set with irregular shapes and varying densities to get a good clustering result with anomaly detection. A density fragment clustering (DFC) algorithm without peaks algorithm is proposed with inspiration from DPC, DBSCAN and SCAN to cope with a larger number of data sets. Experimental results show that our algorithm is more feasible and effective when compared to DPC, AP and DBSCAN algorithms.

Keywords: Density peak, fragment clustering, anomaly detection

1. Introduction

Clustering is known as unsupervised classification in pattern recognition, or nonparametric density estimation in statistics [1–3]. The goal of clustering is to separate a 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), computer sciences (web mining, spatial database analysis, textual document collection, image segmentation), life and medical science (genetics, biology, microbiology, psychiatry, clinic, pathology), to earth sciences (geography, geology, remote sensing), social sciences (sociology, psychology, education), and economics (marketing, business) [4, 5].

Traditional methods in clustering can be broadly categorized into hierarchical, partitioning, density-based, model-based, grid-based, and soft-computing methods [6]. Many density-based clustering methods [7–9] have been proposed and inspired by DBSCAN [7] because its capability of discovering clusters with arbitrary shapes with an overall average runtime complexity of $O(n \times \log n)$. In addition, DBSCAN can detect the number of clusters automatically. However, choosing an appropriate density threshold can be nontrivial. It is very sensitive to the user-defined threshold values, often producing very different clustering results in a data set even for slightly different threshold settings [10].

Density peaks clustering (DPC) algorithm is based on the idea that cluster centers are characterized by a higher density than their neighbors and by a relatively...
large distance from points with higher densities [11]. Concretely speaking, DPC adopts the idea of local density maxima from mean-shift [12] and the basic idea of only one parameter of the distance between data points from K-Medoids [13]. The DPC algorithm is effective with two assumptions that cluster centers are surrounded by neighbors with lower local density and that they are a relatively large distance from any points with a higher local density [11]. Several researches [14–23] have been going on around this method.

As illustrated in Table 1, DPC still has some limitations: (1) The number of clusters cannot be determined automatically. It is determined by its decision graph in two dimensional space with a cutoff distance $d_c$ and its density $\rho$. Du M. et al. [14] have figured out that DPC cannot detect the number of clusters when clusters are in the lower-left corner. Unfortunately, it cannot be changed with human interference. (2) Like DBSCAN [7], relative thin clusters would not be detected by its decision graph. Especially, when a path-based data set consists of a circular cluster with an opening near the bottom and two Gaussian distributed clusters inside, the number of clusters cannot be found via DPC [16] correctly. (3) It is not reasonable that there must be a density peak and only one centroid in each cluster. Otherwise, DPC will split natural groups [16].

The proposed DFC algorithm has solved these above problems via applying both density fragment and network structural similarity that inspired by DPC [11], SCAN [24] and DBSCAN [7] algorithms. When compared with clustering algorithms such as DBSCAN [7], AP [2] and DPC [11] illustrated in Table 1, the proposed novel DFC algorithm has the capability of determining the number of clusters automatically, clustering more reasonable clusters, and detecting anomalies.

We test the novel DFC algorithm in the most popular clustering benchmarks to demonstrate its feasibility. Our algorithm has overcome these above problems with satisfactory results when compared with DPC, SCAN and DBSCAN algorithms on some UCI and synthetic data sets. The rest of this paper is organized as follows. In Section 2, we describe principles of DPC, DBSCAN and SCAN algorithms. In Section 3, we propose the novel density fragment clustering without peaks algorithm. In Section 4, we present experimental results on synthetic data sets and some UCI data sets. In Section 5, some discussions are made to explain the major reasons. Finally, we make conclusions.

2. Related work

The proposed DFC algorithm is inspired by DPC, DBSCAN and SCAN. Brief reviews of these three algorithms should be given in the following subsections.

2.1. DPC: A density peaks clustering approach

Density peaks clustering (DPC) algorithm is based on the idea that cluster centers are characterized by a higher density than their neighbors and by relatively large distance from points with higher density [11]. Cutoff distance $d_c$ is the only parameter in this method. For each data point $x_i$, it computes two quantities: its local density $\rho_i$ and its distance $\delta_i$ from points of higher density.

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

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

$$\rho_i = \sum_j \chi \times (d_{ij} - d_c)$$ (2)

Where $\chi(x) = 1$ if $x < 0$ and $\chi(x) = 0$ otherwise, and $d_c$ is the only user defined parameter to describe its cutoff distance. 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 data set [11]. $\rho_i$, similar to MinPoints in DBSCAN [7], is defined as the number of neighbour points to point $x_i$ in Equation 2.

<table>
<thead>
<tr>
<th>Algorithms</th>
<th>Varying sizes</th>
<th>Irregular shapes</th>
<th>Varying densities</th>
<th>$#$ of clusters</th>
<th>Anomaly detection</th>
</tr>
</thead>
<tbody>
<tr>
<td>DBSCAN</td>
<td>$\partial$</td>
<td>$\partial$</td>
<td>$\partial$</td>
<td>$\partial$</td>
<td>$\partial$</td>
</tr>
<tr>
<td>AP</td>
<td>$\checkmark$</td>
<td>$\times$</td>
<td>$\partial$</td>
<td>$\partial$</td>
<td>$\times$</td>
</tr>
<tr>
<td>DPC</td>
<td>$\checkmark$</td>
<td>$\partial$</td>
<td>$\checkmark$</td>
<td>$\partial$</td>
<td>$\partial$</td>
</tr>
<tr>
<td>DFC</td>
<td>$\checkmark$</td>
<td>$\partial$</td>
<td>$\checkmark$</td>
<td>$\checkmark$</td>
<td>$\checkmark$</td>
</tr>
</tbody>
</table>
Another local density of a point \( x_i \) is presented in Equation 3, as follows:

\[
\rho_i = \sum_j \exp\left(-\frac{d_{ij}^2}{d_c^2}\right) \quad (3)
\]

\[
\delta_i = \min_{j: \rho_j > \rho_i} d_{ij} \quad (4)
\]

Note that \( \delta_i \) is measured by computing the minimum distance between point \( x_i \) and any other points with relative higher density.

The DPC algorithm can be summarized from Rodriguez et al. [11] and Du et al. [14] in Algorithm 1.

**Algorithm 1 The DPC algorithm**

**Input:** Initial nodes \( X \in \mathbb{R}^{N \times M} \), \( d_c \)

**Output:** The label vector of cluster index: \( y \in \mathbb{R}^{N \times M} \)

**Step 1:** Calculate \( d_c \)
1.1 Calculate \( d_{ij} \) from \( R_{N \times M} \) based on Equation 1;
1.2 Sort \( d_{ij} \) in an ascending order;
1.3 Determine \( d_c \) by finding value of 1 to 2% position in the above order.

**Step 2:** Detect cluster centroids by density peaks
2.1 Calculate \( \rho_i \) based on Equation 2 or 3;
2.2 Calculate \( \delta_i \) based on Equation 4;
2.3 Sort nodes based on \( \rho \) in a descending order;
2.4 Detect centroids with relative high \( \rho \) and \( \delta \).

**Step 3:** Assign each node to different clusters
3.1 Detect halo nodes based on its density;
3.2 Determine its affiliation of relative high density nodes by \( \delta_i \) to each cluster.

As illustrated in Algorithm 1, there is an inspiration that each node to be a centroid has two major characteristics: one is relative high density, and the other is relative long distance from its higher density nodes.

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

DBSCAN is a time proved and efficient clustering algorithm with the following features. (1) It is significantly effective in discovering clusters of arbitrary shapes. (2) It is easy for anomaly detection. (3) It is efficient in clustering process [7].

The DBSCAN can determine a cluster if two parameters \( Eps \) and \( MinPts \) are satisfied. Therefore, it is easy to figure outliers out. The capability of finding arbitrary shaped cluster is achieved by its characteristics of *directly-density-reachable*, *density-reachable* and *density-connected*. It is the inspiration that each cluster is consisted of reachable core nodes and their boundary nodes classified by density parameters of \( Eps \) and \( MinPts \). Furthermore, the density within the areas of noise is lower than the density in any of the clusters.

#### 2.3. SCAN: A structural network clustering approach

The SCAN algorithm is a famous structural network clustering approach to discover underlying structures in networks. It can detect clusters, hubs and outliers in networks with the following features [24]:

- It detects clusters, hubs and outliers by using the structure and the connectivity of the vertices as clustering criteria.
- It is fast with its running time of \( O(m) \) on a network with \( n \) vertices and \( m \) edges.

Similar to DBSCAN algorithm, SCAN algorithm inherits the ideas of node density, such as definitions of *directly density-reachable*, *density-reachable*, and *density-connected*. Different from DBSCAN, each edge between two vertices is measured by network structural similarity in Equation 5.

\[
\sigma(v, w) = \frac{|\Gamma(v) \cap \Gamma(w)|}{\sqrt{|\Gamma(v)||\Gamma(w)|}} \quad (5)
\]

In SCAN algorithm, network structural similarity in Equation 5 is adopted to determine core and member vertex in a cluster. It is the inspiration that network structural similarity is an efficient measurement to determine the role of each vertex in a network. In other words, network-based modelling maybe a good way to aggregate clusters.

### 3. Methods

The proposed DFC algorithm inherits the strengths of DPC, DBSCAN and SCAN. *Density fragment* can be defined as the set of density decreasing nodes with relative nearby distance. DFC assumes that (1) a cluster is consisted of many density fragments; (2) two density fragments can be merged together if their network structural similarity is relative high. As illustrated in Algorithm 2, DFC includes three major steps: density decreasing, density fragment generation and density fragment aggregation.
Algorithm 2 The DFC algorithm.

Input: Initial nodes $X \in \mathbb{R}^{N \times M}$, $d_c$, $\rho$

Output: All clusters and outliers are found

Step 1: Similarity matrix($d$) calculation
1.1 Calculate $d_{ij}$ from $d_{N \times M}$ based on Equation 1;
1.2 Sort $d_{ij}$ in an ascending order.

Step 2: Density($\rho$) calculation
2.1 Determine $d_c$ value with principles of DPC;
2.2 Calculate $\rho_i$ value based on Equations 2 or 3;
2.3 Put all $\rho$ in a descending order.

Step 3: Delta($\delta$) calculation
for each node $i$ do
    $\delta_i = d_{ij}$ based on Equation 1
end for

Step 4: Density fragment generation
4.1 Core node link generation;
for each node $i$ in a descending order do
    if $\delta_i < d_c$ then
        Node $i$ becomes a core node $i$;
        Get slot nodes for the core node $i$ that their distances within $d_c$;
        Connect core node $i$ to its neighbour core node $i + 1$;
        Get a core node link based on the above rule;
        Get the core node link and their slot nodes together and then generate a density fragment.
    end if
end for

Step 5: Density fragment aggregation
for each density fragment do
    Calculate density fragment similarity based on Equation 7
    for each $DFSim(x, y)$ do
        if $DFSim(x, y) > 0.5$ then
            Aggregate the fragment $x$ and fragment $y$ as one cluster.
        end if
    end for
end for

Step 6: Anomaly detection
for each core node link do
    if $DFSim(x, y) > 0.5$ then
        Node $i$ is an outlier.
    else
        if Core node link has no slot nodes then
            It is an outlier.
        end if
    end if
end for

Fig. 1. An example of core node chains.

3.1. Density decreasing

As similar with DPC, a suitable cutoff distance $d_c$ is selected based on principles of density to calculate their local density $\rho$ for sorting these density values in a descending order. $\delta_i$ is an Euclidean distance between the node $i$ and its next node $i + 1$ in the density decreasing order depicted in Equation 6.

$$\delta_i = \text{Distance}(\text{node}_i, \text{node}_{i+1})$$

3.2. Density fragment generation

Density fragment generation is processed in two stages: in stage I, node $i$ can become a core node if its $\delta_i \leq d_c$, then check its next node $i + 1$ with density-reachability to form a core node chain illustrated in Fig. 1; in stage II, a core node $i$ can expand its density-connectivity to its nearby nodes in the range of $d_c$. These nearby nodes can be defined as slot nodes. A core node chain and its slot nodes constitute a density fragment. All density fragments can be formed after processing this two stages.

As illustrated in Fig. 1, 40 nodes with two dimensions are generated randomly, core node chains are depicted with solid line, and density fragments are formed by these core node chains and their slot nodes.

3.3. Density fragment aggregation

As described in the sub Section 3.2, density fragment is a network structure. Density fragment aggregation can be processed by principles
of network structural similarity and a priori probability. The network structural similarity of density fragments can be defined in Equation 7. Frag\textsubscript{x} and Frag\textsubscript{y} refer to density fragment of x and y respectively. The larger of DFSim(x, y) value, the higher probability of aggregation between Frag\textsubscript{x} and Frag\textsubscript{y}. The aggregation of density fragments is processed with a priori probability that the density fragment with higher density has a higher priority to merge other density fragments.

\[
    DFSim(x, y) = \frac{|Frag_x \cap Frag_y|}{\sqrt{|Frag_x||Frag_y|}}
\]  

(7)

The proposed density fragment clustering approach is depicted in Algorithm 2.

4. Results

To test its feasibility and effectiveness of the proposed DFC algorithm, we compare it with DBSCAN, AP and DPC in the above four UCI data sets and the next three synthetic data sets listed in Table 2. F-Measure can be defined in Equation 10, which is an index to evaluate the performance of clustering results. 7 different data sets (Iris, Seeds, Wine, Glass, Flame, Aggregation and Spiral) are selected to evaluate the clustering performance of DBSCAN, AP, DPC and DFC in Table 3 and Fig. 2. The results in Table 3 and Fig. 2 are mean values based on 30 times run.

F-Measure index measures accuracy. It considers both the precision P and the recall R of clustering algorithms: P is the ratio of the number of correct results to the number of all returned results, and R is the ratio of the number of correct results to the number of results that should have been returned. P, R and F-Measure(F) are defined as the following Equations 8, 9 and 10.

\[
P = \frac{|M_j \cap C_i|}{|C_i|}
\]

(8)

\[
R = \frac{|M_j \cap C_i|}{|M_j|}
\]

(9)

\[
F(M_j, C_i) = \frac{2 \cdot P(M_j, C_i) \cdot R(M_j, C_i)}{P(M_j, C_i) + R(M_j, C_i)}
\]

(10)

Besides the clustering performance depicted in Fig. 2, many experiments have been done to evaluate the DFC capability with limitations such as varying sizes, irregular shapes, varying densities, number of clusters and anomaly detection.

4.1. Varying sizes

Clusters with varying sizes are normal in a real data set, and they should be detected correctly without interference. Aggregation, Flame and D are selected to evaluate the capability of processing varying sizes of clusters in a data set.

As illustrated in Fig. 3, to test the accuracy of the DBSCAN algorithm without any other factors, the number of cluster is controlled to the true cluster number \(Eps = 0.9\), \(MinPts = 5\). AP’s preference is set to the medium value of the similarity matrix according to the original AP algorithm. DPC’s \(d_c\) value is set to 2.7212 and DFC’s \(DFSim\) is set to 0.3024 when they can get their best result. This figure shows that both DFC and DPC can find clusters correctly while AP cannot cluster in an intuitive way. And some points are close to their actual clusters while DBSCAN cannot recognize them. And the parameters in these four algorithms are set for the same purpose in the following data sets. Figures 5 and 8 show the same result that DFC has the ability to detect data sets with varying size while most algorithms can not.

<table>
<thead>
<tr>
<th>Data sets</th>
<th>Nodes</th>
<th>Dimensions</th>
<th>Clusters</th>
</tr>
</thead>
<tbody>
<tr>
<td>Iris</td>
<td>150</td>
<td>4</td>
<td>3</td>
</tr>
<tr>
<td>Seeds</td>
<td>210</td>
<td>7</td>
<td>3</td>
</tr>
<tr>
<td>Wine</td>
<td>178</td>
<td>13</td>
<td>3</td>
</tr>
<tr>
<td>Glass</td>
<td>214</td>
<td>9</td>
<td>6</td>
</tr>
<tr>
<td>Flame</td>
<td>240</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>Aggregation</td>
<td>788</td>
<td>2</td>
<td>7</td>
</tr>
<tr>
<td>Spiral</td>
<td>312</td>
<td>2</td>
<td>3</td>
</tr>
<tr>
<td>D</td>
<td>87</td>
<td>2</td>
<td>3</td>
</tr>
</tbody>
</table>

Table 2 7 different types of data sets

<table>
<thead>
<tr>
<th>Data sets</th>
<th>DBSCAN</th>
<th>AP</th>
<th>DPC</th>
<th>DFC</th>
</tr>
</thead>
<tbody>
<tr>
<td>Iris</td>
<td>0.7045</td>
<td>0.4851</td>
<td>0.7715</td>
<td>0.7554</td>
</tr>
<tr>
<td>Seeds</td>
<td>0.4784</td>
<td>0.3877</td>
<td>0.8026</td>
<td>0.793</td>
</tr>
<tr>
<td>Wine</td>
<td>0.5794</td>
<td>0.3142</td>
<td>0.5892</td>
<td>0.5894</td>
</tr>
<tr>
<td>Glass</td>
<td>0.3315</td>
<td>0.2874</td>
<td>0.5418</td>
<td>0.5571</td>
</tr>
<tr>
<td>Flame</td>
<td>0.6242</td>
<td>0.2874</td>
<td>1</td>
<td>0.9713</td>
</tr>
<tr>
<td>Aggregation</td>
<td>0.8140</td>
<td>0.3429</td>
<td>1</td>
<td>0.9909</td>
</tr>
<tr>
<td>Spiral</td>
<td>1</td>
<td>0.2853</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>D</td>
<td>0.8143</td>
<td>0.6117</td>
<td>0.8780</td>
<td>1</td>
</tr>
</tbody>
</table>

Table 3 F-Measure evaluation
Fig. 2. F-Measure with 7 different data sets.

Fig. 3. Clustering the data set of Aggregation with varying sizes and irregular shapes.

(a) DBSCAN, $Eps = 0.9$, $MinPts = 5$

(b) AP, $Preference = -16.5170$

(c) DPC, $d_c = 2.7212$

(d) DFC, $DFSim = 0.3024$
4.2. Irregular shapes

In general, density-based clustering algorithm has advantage of processing data sets with irregular shapes. *Aggregation, Spiral* and *Flame* can be adopted to evaluate its capability of processing irregular shapes. When compared with the DPC algorithm, the proposed DFC algorithm has the equal capability of clustering irregular shaped clusters shown in Figs. 3, 4 and 8. Compared with AP in these three data sets, DFC has the superior performance. DBSCAN gets a good result on *Spiral* clustering the same as the DPC and DFC in Fig. 4. However, DBSCAN cannot deal with the data sets such as *Aggregation* and *Flame* due to its incorrect anomaly detection. The DFC can detect the linear data like *Spiral* and also has the ability to handle irregular spherical data like *Aggregation* and *Flame*.

4.3. Varying densities

Data set $D$ in Fig. 5 is used to test the advantage of the DFC in varying densities clustering. AP cannot solve this kind of data set when it gets the right cluster number. DBSCAN only detect the two cluster centers while the data set has the three centers actually, and all the points in the third cluster are recognized as anomalies which shown black color in Fig. 5. The DPC and DFC have a good performance in the data set $D$.

4.4. Number of clusters

The DPC has the advantage of finding cluster number by its decision graph [11]. However the DPC doesn’t work well in data sets *Iris* or *Glass* which shown in Figs. 6 and 7, because it is difficult to select the suitable cluster centroids. The proposed
Fig. 5. Clustering with varying densities on D data set.

(a) DBSCAN, $Eps = 0.6$, $MinPts = 5$

(b) AP, $Preference = -2.1523$

(c) DPC, $d_c = 0.9441$

(d) DFC, $DFSim = 0.2673$

DFC can achieve reasonable clusters without human intervention.

4.5. Anomaly detection

Anomaly detection is a basic function for any clustering algorithm. DBSCAN adopts $MinPts$ and density-reachability to detect outliers, but this algorithm meets unsatisfied anomaly detection which shown in Figs. 3, 5 and 8. DPC takes outliers as halos without anomaly detection. As illustrated in Figs. 8 and 9, only the proposed DFC algorithm can figure these two outliers correctly in the top left corner in data set Flame.

5. Discussion

With the aggregation principle in DFC, clusters are formed by merging density fragments based on network structural similarity and density decreasing sequence. To analyse the strengths and weaknesses of the proposed DFC algorithm, the elaborate analysis of simulation results shown in Section 4 are made to know its capability of processing data sets with varying sizes, irregular shapes and varying densities, of detecting the number of clusters and outliers.

5.1. Analysis of processing varying sizes

As illustrated in Section 4.1, both DFC and DPC can detect clusters correctly with varying sizes. In DBSCAN algorithm, the true cluster cannot be found when there is a close line, and the wrong anomaly detecting disturb border points to reach their correct clusters. In AP algorithm, it is difficult to adjust global parameters to satisfy the data distribution in Fig. 3(b). To overcome the limitation of global parameters, each cluster in DPC is formed by determining its centroid.
and by absorbing lower density nodes based on inter-
distances $\delta_i$ in Equation 6. The proposed DFC also has
this capability because that the size of each cluster is
determined by network structural similarity among
density fragments.

5.2. Analysis of detecting irregular shapes

As illustrated in Figs. 3, 4 and 8, irregular shapes
in data sets Aggregation, Spiral and Flame are
extracted correctly by DPC and DFC algorithms. The
DBSCAN algorithm gets good result in Spiral while
fault in clustering Aggregation and Flame. The AP
algorithm can get many fragments of each clusters,
however it cannot combine them together to a rea-
sonable cluster as depicted in Fig. 4(b). The DPC
algorithm can get the correct clusters because each
centroid is in the front part of a cluster and the fol-
lowing density decreasing nodes are near to them
iteratively shown in Fig. 4(c). Similar as illustrated
in Figs. 3 and 8, the reason for successful cluster-
ing in DFC is that density fragments are materials
of any irregular shaped cluster and the merging prin-
iple is based on network structural similarity and
priori probability without any direction guidance.

5.3. Analysis of processing varying densities

Generally speaking, it is an important capabil-
ity of handling data sets with varying densities.
DBSCAN algorithm cannot handle varying den-
sity data sets because it is distance-based clustering
approach. As principles in DBSCAN, a node will
be absorbed to a cluster if it is density-reachable or
density-connectivity [7]. However, it is difficult to set
the global density parameter in DBSCAN. Both DPC
and DFC can handle data sets with varying densities
because both of them do not adopt density principles
in the stage of cluster generation. A cluster in DPC is
formed by finding its centroid to absorb decreasing
density nodes one by one [11]. The proposed DFC
forms its cluster by merging density fragments with
scalability in Equation 7.

5.4. Analysis of detecting the number of clusters

As illustrated in Figs. 6 and 7 in Section 4.4, the
DPC algorithm will determine the number of clus-
ters with human involvement. However, it is difficult
for experts to determine the number of centroids in
the decision graphs of Fig. 6(a) and (b). However,
the proposed DFC algorithm has a relative advan-
tage of detecting the number of clusters automatically
when compared with DPC. The reason is that it
will be unreasonable if the centroids of clusters are
determined in a priori. In other words, more rea-
sonable number of clusters should be determined
by density-reachability and density-connectivity as
DBSCAN. The proposed DFC algorithm does
not determine each centroid of a cluster, while
it forms a cluster by merging density fragments
as SCAN.

5.5. Analysis of anomaly detection

Compared with DPC, AP and DBSCAN shown
in Figs. 8 and 9, only the proposed DFC algorithm
can figure these two outliers correctly in the left top
corner in the data set of Flame. DBSCAN algorithm
has problems of finding outliers, and AP algorithm
cannot find these two outliers with optimization by
large amount of iterations. The state-of-the-art DPC
algorithm cannot detect outliers when the distance
between outliers and relative higher density nodes
is less than $d_c$. However, the $d_c$ will not be a small
value to find a centroid of a cluster in a data set.
Fig. 7. Decision graph of DPC with Glass data set.

Fig. 8. Irregular shapes and anomaly detection of Flame data set.
However, outliers are detected by density fragments and their density connectivity in our proposed DFC algorithm. Therefore, DFC has its advantage to detect outliers.

6. Conclusion

With inspiration by DPC, DBSCAN and SCAN algorithms, the proposed DFC is processed by merging density fragments with principles of network structural similarity and density decreasing sequence to improve its capability with varying sizes, varying densities and irregular shapes, with detecting the number of clusters and outliers. In this paper, the proposed DFC algorithm shows the power in some synthetic data sets. Besides the good feasibility, the DFC gets better clustering performances when compared with some classical methods, such as DBSCAN, AP and DPC in UCI data sets.

However, the proposed DFC algorithm does not perform well in a path-based data set such as Pathbase. Future research is to improve the performance of DFC algorithms on more complex data sets.

Acknowledgments

The authors are grateful to the financial support by the National Natural Science Foundation of China (No. 61572225), the Foundation of Jilin University of Finance and Economics (No. 2016P46) and the Foundation of Education Department of Jilin Province in China (No. JJKH20170119KJ).

References


