Physica A 523 (2019) 702–713

Contents lists available at ScienceDirect

Physica A

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

PHYSICA A

STATISTICAL MECHANICS AND ITS APPLICATIONS

A novel density peaks clustering algorithm based on *k* nearest neighbors for improving assignment process[☆]

Jianhua Jiang [a,b,*], Yujun Chen [a], Xianqiu Meng [a], Limin Wang [a], Keqin Li [c,**]

[a] *Department of Data Science, Jilin University of Finance and Economics, Changchun 130117, PR China*
[b] *Key Laboratory of Symbolic Computation and Knowledge Engineering, Ministry of Education, Jilin University, Changchun, 130012, PR China*
[c] *Department of Computer Science, State University of New York, New Paltz, NY 12561, USA*

HIGHLIGHTS

- K nearest neighbors is adopted to solve domino effect problem in density peaks clustering.
- The capability of aggregating some non-spherical clusters is enhanced effectively.
- Experimental results show that the DPC-KNN algorithm is more effective.

ARTICLE INFO

Article history:
Received 25 December 2018
Received in revised form 8 March 2019
Available online 11 March 2019

MSC:
00-01
68T10

Keywords:
Density clustering algorithm
Density peaks
k nearest neighbors

ABSTRACT

Density Peaks Clustering (DPC) algorithm is a kind of density-based clustering approach, which can quickly search and find density peaks. However, DPC has deficiency in assignment process, which is likely to trigger domino effect. Especially, it cannot process some non-spherical data sets such as *Spiral*. The research results indicate that assignment process appears to be the most significant step in deciding the success of the clustering performance. Therefore, we propose a density peaks clustering based on *k* nearest neighbors (DPC-KNN) which aims to overcome the weakness of DPC. The proposed DPC-KNN integrates the idea of *k* nearest neighbors into the distance computation and assignment process, which is more reasonable. It can be seen from experimental results that the DPC-KNN algorithm is more feasible and effective, compared with K-means, DBSCAN and DPC.

1. Introduction

Clustering is to divide objects into several sensible clusters according to their similarity [1–3]. Objects in the same cluster are characterized by higher similarity, but objects in different clusters have lower similarity. Clustering approaches have been applied widely in engineering, computer sciences fields, and so on [4–7].

[☆] The authors are grateful to the financial support by the National Natural Science Foundation of China (no. 61572225), the Social Science Foundation of Jilin Province, China (no. 2017BS28), the Natural Science Foundation of the Science and Technology Department of Jilin Province, China (no. 20180101044JC), the Foundation of the Education Department of Jilin Province, China (no. JJKH20180465kJ) and the Foundation of Jilin University of Finance and Economics, China (no. 2018Z05).

[*] Corresponding author at: Department of Data Science, Jilin University of Finance and Economics, Changchun 130117, PR China.

[**] Corresponding author.

E-mail addresses: jjh@jlufe.edu.cn (J. Jiang), lik@newpaltz.edu (K. Li).

https://doi.org/10.1016/j.physa.2019.03.012
0378-4371/© 2019 Elsevier B.V. All rights reserved.

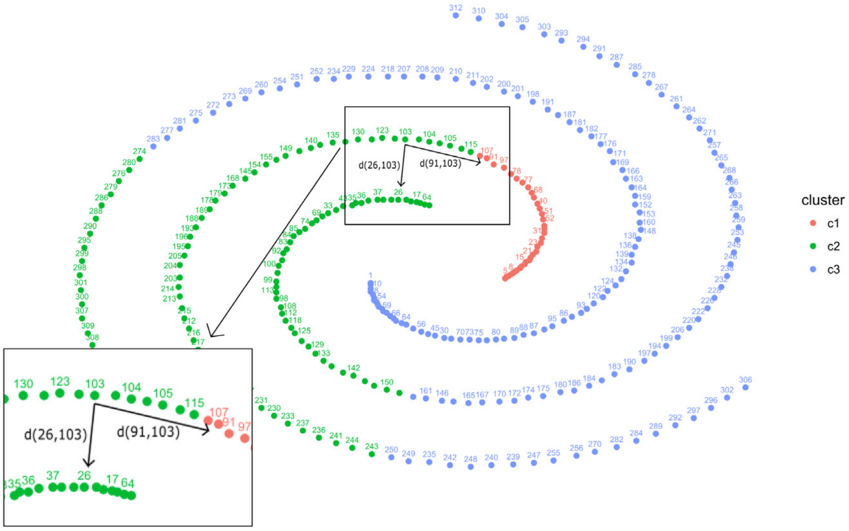


Fig. 1. Sort density by **DPC** on *Spiral* data set, $d_c = 13.6041$, ×.

Clustering algorithms are divided into different categories by different starting points and criteria [1]. K-means [8] is a simple, well-known algorithm. It is very fast and can be easily implemented in solving spherical data sets. The drawbacks of K-means are that it is hard to decide the initial partitions and the number of clusters, it is sensitive to outliers and noise, and it has weak ability of discovering non-spherical clusters [1,5,9]. DBSCAN [10] is a density-based clustering algorithm. It is able to discover arbitrary shape clusters. DBSCAN depends on two parameters: ϵ and *MinPts*. ϵ is radius of neighborhood for an object, and *MinPts* is the minimum number of points in a neighborhood, but the two parameters need to be specified by users [1,5,9,11]. It is difficult to pre-set the two parameters appropriately.

In 2014, DPC (density peaks clustering) [12] algorithm was published in the journal *Science*. It is a kind of density-based clustering algorithm based on the idea that cluster centers are characterized by a higher density and a relatively longer distance [12]. Cutoff distance d_c is the only user-defined parameter. DPC requires two quantities that are local density ρ and distance δ. It is able to quickly search and find density peaks. DPC introduces the concept of cluster centers, it can determine the clustering center automatically and it is able to deal with arbitrary shape clusters. Due to the good performance of DPC algorithm, it has attracted the attention of many scholars. Focusing on this method, several researches [13–22] have been carried out to improve its capabilities.

DPC finds out cluster centers by decision graph. For the remaining points, DPC adopts one-step strategy that each point is assigned to the cluster of its nearest point with higher density. The assignment rules makes DPC efficient. However, once a data point is assigned incorrectly, it will cause cluster allocation errors among the remaining points, triggering the **domino effect** [11,22]. As is shown in Fig. 1, c1, c2, c3 represent three different clusters respectively, and data points are marked by numbers. No. 1 means the point is of highest density. The larger the number of points is, the lower their density becomes. Point No. 103 is of higher density than its neighbors, so the assignment of it will also influence the assignment of its neighbors. Apparently, point No. 103 is assigned to cluster c2 incorrectly, triggering the **domino effect** and generating wrong cluster result, which leads to the failure of cluster aggregation.

In DPC algorithm, distance δ will influence assignment process. In assignment process of DPC algorithm, cluster allocation of each point is determined by its distance δ. For point No. 103, distance δ_{103} is distance $d_{(26,103)}$, which is the minimum distance from points higher than its density. As a result, it is absorbed to point No. 26 and assigned to cluster c2 incorrectly, which belongs to cluster c1. Therefore, it is unreasonable if the calculation of distance δ only takes into consideration the distance between a point and its nearest neighbors of higher density. In order to overcome the problem, we propose a density peaks clustering based on k nearest neighbors (DPC-KNN) which integrates the idea of k nearest neighbors into DPC, which further improves the distance δ computation. This approach is tested with *Seeds* [23], *Wine* [23] data sets and five shaped data sets, namely *Aggregation* [24], *Flame* [25], *Spiral* [26], *Jain* [27] and *R15* [28]. Compared with K-means [8], DBSCAN [10] and DPC [12], the proposed DPC-KNN has three advantages:

(1) Cluster center in decision graph is more notable than DPC;

(2) Non-spherical clusters are aggregated more effectively than DPC;

(3) Various sizes clusters are aggregated more correctly than K-means and DBSCAN.

The rest of this paper is organized as follows. The functions of DPC algorithm and of DPC-KNN algorithm will be described in Section 2. The process of DPC-KNN algorithm will be proposed in Section 3. Experimental results on some data sets will be presented in Section 4. Some discussions will be made to explain the major reasons in Section 5. Finally, conclusions will be drawn in the last Section.

2. Related work

The proposed DPC-KNN algorithm is inspired by DPC [12] and k nearest neighbors. Brief reviews ought to be given in the following subsections.

2.1. DPC: a density peaks clustering approach

DPC algorithm is based on the hypothesis that cluster centers are characterized by a higher density than their neighbors and by a relatively longer distance from points of higher density [12]. P_i means point *i*. It is a data point in the data set of $N * M$ dimensions, $P_i \in N$. For each point of P_i, it computes two parameters: its local density ρ_i and its distance δ_i from points with higher density. These two parameters are relied on distance d_{ij} between data points P_i and P_j.

$$d_{ij} = distance(P_i, P_j) \tag{1}$$

where the formula can be calculated by distance formula, e.g. Euclidean distance.

The local density ρ_i of point P_i is given by Eq. (2).

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

where $\chi(d_{ij} - d_c) = 1$ if $(d_{ij} - d_c) < 0$ and $\chi(d_{ij} - d_c) = 0$ otherwise, cutoff distance d_c is the only user-defined parameter. As a rule of thumb, one can choose a d_c so that the average number of neighbors is around 1% to 2% of the total number of points in a data set [12,18,22].

For point P_i of the highest density, its distance δ_i is given by Eq. (3).

$$\delta_i = max(d_{ij}) \tag{3}$$

For the rest of points, distance δ_i are defined by Eq. (4).

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

2.2. DPC-KNN: density peaks clustering based on k nearest neighbors

In DPC-KNN algorithm, for each point P_i, the formula of local density is the same as DPC shown in Eq. (2). The proposed DPC-KNN integrates the idea of k nearest neighbors into the formula of distance δ. The set of k nearest neighbors of point P_i is defined by Eq. (5). k represents the number of the nearest neighbors.

$$N_i^k = \{P_j | \underset{k}{min}(d_{ij}), P_j \in N, P_j \neq P_i\} \tag{5}$$

The set that consists of point P_i and its k nearest neighbors is defined by Eq. (6).

$$S_i = \{N_i^k, P_i\} \tag{6}$$

The set of the points of higher density than point P_i is given by Eq. (7).

$$H_i = \{P_t | \rho_t > \rho_i, P_t \in N, P_t \neq P_i\} \tag{7}$$

For each point P_i, except the point of the highest density, distance δ_i is defined by Eq. (8).

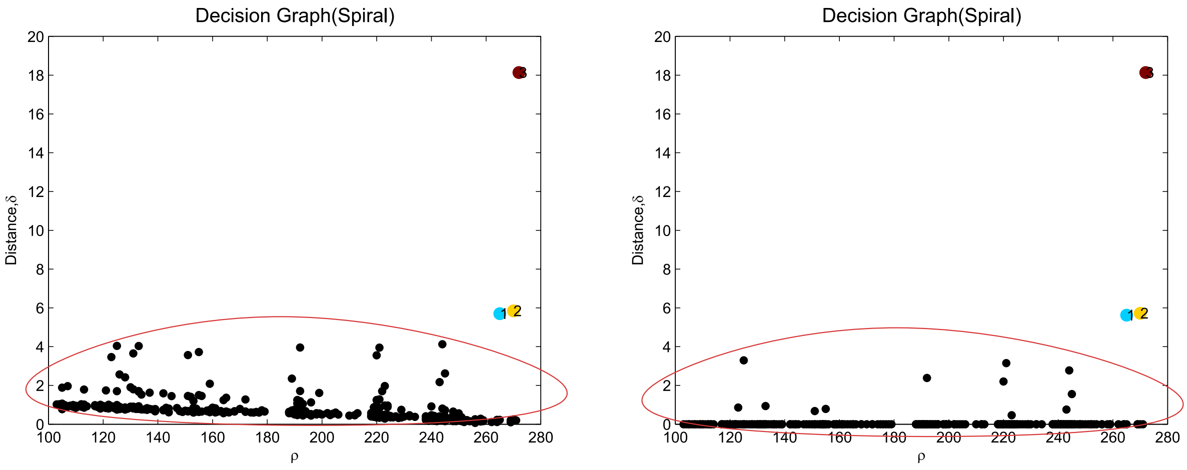
$$\delta_i = min\{distance(P_l, P_t)\}, P_l \in S_i, P_t \in H_i \tag{8}$$

3. Methods

The proposed DPC-KNN is different from DPC algorithm. In DPC-KNN algorithm, the formula of distance δ and assignment rules are redefined. DPC-KNN algorithm includes three major steps: (1) calculate the density and distance of points; (2) generate decision graph; (3) aggregate clusters.

3.1. CaLculate the density and distance of points

A suitable cutoff distance d_c is selected to calculate the local density ρ_i [12], and the formula of local density of DPC-KNN algorithm is the same as DPC algorithm shown in Eq. (2). But the calculation of distance δ_i is different in DPC-KNN algorithm, to which k nearest neighbors is introduced. k nearest neighbors will influence distance δ_i for each point P_i, the latter already made clear by Eqs. (3) and (8).



(a) DPC

(b) DPC-KNN, $k = 7$

Fig. 2. The decision graph of *Spiral* data set with $d_c = 14.1182$.

3.2. Generate decision graph

Distance δ is adopted as the vertical axis and density ρ as the horizontal axis of decision graph. Cluster center is characterized by a higher density and by a relatively longer distance. As shown in Fig. 2(b), all the points except the cluster centers have lower value of distance δ in decision graph of DPC-KNN algorithm, so it is easy to find cluster centers which are prominent in the decision graph.

3.3. Aggregate clusters

Cluster centers are selected by decision graph, and the remaining point P_i is assigned to each cluster. In the assignment rules of DPC-KNN algorithm, point P_i is absorbed to the point P_t in the set H_i which has minimum distance to point P_l in the set S_i. Therefore, in assignment process, point P_i is assigned to the cluster where lies the nearest point of higher density, which is determined by δ_i based on Eq. (8). Finally, we can get the clusters aggregated by assignment process. DPC-KNN algorithm is depicted in Algorithm 1.

Algorithm 1 Density peaks clustering based on k nearest neighbors

```
Require: Initial points $P_i \in R_{N\times M}$ ($R_{N\times M}$ is the matrix of $N\times M$ dimensions), $d_c$ ($d_c$ is a cutoff distance), $k$ ($k$ nearest neighbors)
Ensure: The label vector of cluster index: $y \in R_{N\times M}$
  Step 1: Calculate $d_c$
  1.1 Calculate $d_{ij}$ from $R_{N\times M}$ based on Eq. (1);
  1.2 Sort $d_{ij}$ in an ascending order;
  1.3 Determine $d_c$ by finding value of certain percentage position in the above order.
  Step 2: Detect cluster centers by decision graph
  2.1 Calculate $\rho_i$ based on Eq. (2);
  2.2 Sort points based on $\rho$ in a descending order;
  2.3 Calculate $\delta$ based on Eq. (3) for point of the highest density, and calculate $\delta$ based on Eq. (8) for the remaining points;
  2.4 Generate the decision graph with density $\rho$ and with distance $\delta$;
  2.5 Find cluster centers from decision graph.
  Step 3: Assign each point to different clusters
  3.1 Point $P_i$ is absorbed to the nearest point of higher density which is determined by $\delta_i$ based on Eq. (8);
  3.2 Iterate until all points are assigned.
```

Table 1
Seven different types of data sets.

Data sets	Points	Dimensions	Clusters
Seeds	210	7	3
Wine	178	13	3
Aggregation	788	2	7
Flame	240	2	2
Spiral	312	2	3
Jain	373	2	2
R15	600	2	15

4. Results

To test its feasibility and effectiveness of the proposed DPC-KNN algorithm, it is compared with K-means [8], DBSCAN [10] and DPC [12] on *Seeds* [23], *Wine* [23] data sets and five shaped data sets, which are, *Aggregation* [24], *Flame* [25], *Spiral* [26], *Jain* [27] and *R15* [28] respectively. The attributes of these data sets are listed in Table 1.

4.1. Evaluate clustering results

We adopt *F-Measure* [29], *NMI* (Normalized Mutual Information) [30] and *ARI* (Adjust Rand Index) [30] to test the performance of K-means, DBSCAN, DPC and DPC-KNN. The upper limit of the three indexes is 1. The larger the three indexes are, the better is the cluster result.

F-Measure involves both the precision P and the recall R: P is the ratio between the number of correct positive results and the number of all positive results returned by the classifier, and R is the ratio between the number of correct positive results and the number of all samples that should have been identified as positive. P, R and *F-Measure* are defined by Eqs. (9), (10) and (11). M_j is set of the number of all samples that should have been identified as positive. C_i is set of the number of all positive results returned by the classifier.

$$P(M_j, C_i) = \frac{|M_j \cap C_i|}{|C_i|} \tag{9}$$

$$R(M_j, C_i) = \frac{|M_j \cap C_i|}{|M_j|} \tag{10}$$

$$F(M_j, C_i) = \frac{2 \times P(M_j, C_i) \times R(M_j, C_i)}{P(M_j, C_i) + R(M_j, C_i)} \tag{11}$$

The mutual information (MI) [30] can be used to measure the information shared by two clusters. Given a set S of N data points, and two partitions of set S, namely $X = \{X_1, X_2, \dots, X_r\}$, and $Y = \{Y_1, Y_2, \dots, Y_s\}$. Suppose that we pick an object at random from S, then the probability that the object falls into cluster X_i is

$$P(i) = \frac{|X_i|}{N} \tag{12}$$

Entropy can be described as the information conveyed by the uncertainty that a randomly selected point belongs to a certain cluster. Entropy of the cluster X is given by Eq. (13).

$$H(X) = -\sum_{i=1}^{r} P(i) \times \log P(i) \tag{13}$$

The MI [30] between the clusters X and Y is defined by Eq. (14).

$$I(X, Y) = \sum_{i=1}^{r} \sum_{j=1}^{s} P(i, j) \times \log \frac{P(i, j)}{P(i)P(j)} \tag{14}$$

The NMI [31] is calculated as Eq. (15).

$$NMI(X, Y) = \frac{2 \times I(X, Y)}{H(X) + H(Y)} \tag{15}$$

The overlap between X and Y can be summarized in a contingency table shown in Table 2. N_{rs} denotes the number of objects in common between X_r and Y_s.

Table 2
The contingency table.

X	Y				Sums
	Y_1	Y_2	$\cdots$	Y_s	
X_1	N_{11}	N_{12}	$\cdots$	N_{1s}	a_1
X_2	N_{21}	N_{22}	$\cdots$	N_{2s}	a_2
$\vdots$	$\vdots$	$\vdots$	$\ddots$	$\vdots$	$\vdots$
X_r	N_{r1}	N_{r2}	$\cdots$	N_{rs}	a_r
Sums	b_1	b_2	$\cdots$	b_s	

Table 3
F-Measure evaluation.

Data sets	K-means	DBSCAN	DPC	DPC-KNN
Seeds	0.8106	0.5543	**0.8169**	**0.8169**
parameter	$k = 3$	$\epsilon = 1.2/MinPts = 2$	$d_c = 1.3110$	$d_c = 1.3110/k = 5$
Wine	0.5835	0.5813	0.6000	**0.6425**
parameter	$k = 3$	$\epsilon = 2/MinPts = 4$	$d_c = 145.3290$	$d_c = 96.4202/k = 5$
Aggregation	0.8159	0.9003	**1**	**1**
parameter	$k = 7$	$\epsilon = 1.05/MinPts = 4$	$d_c = 3.1185$	$d_c = 3.1185/k = 7$
Flame	0.7586	0.9840	**1**	**1**
parameter	$k = 2$	$\epsilon = 0.93/MinPts = 4$	$d_c = 1.4577$	$d_c = 1.6008/k = 4$
Spiral	0.3276	**1**	0.7795	**1**
parameter	$k = 3$	$\epsilon = 2/MinPts = 4$	$d_c = 13.6041$	$d_c = 13.6041/k = 7$
Jain	0.6977	0.9767	**1**	**1**
parameter	$k = 2$	$\epsilon = 2.5/MinPts = 4$	$d_c = 13.0124$	$d_c = 13.0124/k = 9$
R15	**0.9932**	0.9402	0.9916	**0.9932**
parameter	$k = 15$	$\epsilon = 0.35/MinPts = 5$	$d_c = 0.5887$	$d_c = 0.6551/k = 8$

Table 4
Normalized mutual information evaluation.

Data sets	K-means	DBSCAN	DPC	DPC-KNN
Seeds	**0.6949**	0.0948	0.6938	0.6938
parameter	$k = 3$	$\epsilon = 1.2/MinPts = 2$	$d_c = 1.3110$	$d_c = 1.3110/k = 5$
Wine	0.4287	5.551e−07	0.4240	**0.4298**
parameter	$k = 3$	$\epsilon = 2/MinPts = 4$	$d_c = 145.3290$	$d_c = 96.4202/k = 5$
Aggregation	0.8805	0.9207	**1**	**1**
parameter	$k = 7$	$\epsilon = 1.05/MinPts = 4$	$d_c = 3.1185$	$d_c = 3.1185/k = 7$
Flame	0.4622	0.9275	**1**	**1**
parameter	$k = 2$	$\epsilon = 0.93/MinPts = 4$	$d_c = 1.4577$	$d_c = 1.6008/k = 4$
Spiral	0.0007	**1**	0.6951	**1**
parameter	$k = 3$	$\epsilon = 2/MinPts = 4$	$d_c = 13.6041$	$d_c = 13.6041/k = 7$
Jain	0.3672	0.8729	**1**	**1**
parameter	$k = 2$	$\epsilon = 2.5/MinPts = 4$	$d_c = 13.0124$	$d_c = 13.0124/k = 9$
R15	**0.9942**	0.9459	0.9928	**0.9942**
parameter	$k = 15$	$\epsilon = 0.35/MinPts = 5$	$d_c = 0.5887$	$d_c = 0.6551/k = 8$

Adjusted Rand Index (*ARI*) [30] is defined as Eq. (16).

$$ARI = \frac{\sum_{ij}\binom{N_{ij}}{2} - \left[\sum_i \binom{a_i}{2} \sum_j \binom{b_j}{2}\right] / \binom{N}{2}}{\frac{1}{2}\left[\sum_i \binom{a_i}{2} + \sum_j \binom{b_j}{2}\right] - \left[\sum_i \binom{a_i}{2} \sum_j \binom{b_j}{2}\right] / \binom{N}{2}} \tag{16}$$

where N_{ij} is an entry in the contingency table, a_i and b_j are its marginal sums.

If the three indexes are higher, the performance of algorithm is better. The cluster results are depicted in Tables 3–5, which are mean values based on 20 times run. Overall, DPC-KNN and DPC are superior to K-means and DBSCAN. DPC-KNN and DPC can achieve maximum value on data sets of *Aggregation*, *Flame* and *Jain*. DPC-KNN performs better than DPC on *Spiral* data set. In summary, DPC-KNN gets highest value, compared with K-means, DBSCAN and DPC.

Table 5
Adjust rand index evaluation.

Data sets	K-means	DBSCAN	DPC	DPC-KNN
Seeds	0.7166	0.0025	**0.7264**	**0.7264**
parameter	$k = 3$	$\epsilon = 1.2/MinPts = 2$	$d_c = 1.3110$	$d_c = 1.3110/k = 5$
Wine	0.3711	0.0	0.2796	**0.3818**
parameter	$k = 3$	$\epsilon = 2/MinPts = 4$	$d_c = 145.3290$	$d_c = 96.4202/k = 5$
Aggregation	0.7624	0.8662	**1**	**1**
parameter	$k = 7$	$\epsilon = 1.05/MinPts = 4$	$d_c = 3.1185$	$d_c = 3.1185/k = 7$
Flame	0.4998	0.9659	**1**	**1**
parameter	$k = 2$	$\epsilon = 0.93/MinPts = 4$	$d_c = 1.4577$	$d_c = 1.6008/k = 4$
Spiral	−0.0057	**1**	0.6686	**1**
parameter	$k = 3$	$\epsilon = 2/MinPts = 4$	$d_c = 13.6041$	$d_c = 13.6041/k = 7$
Jain	0.3181	0.9411	**1**	**1**
parameter	$k = 2$	$\epsilon = 2.5/MinPts = 4$	$d_c = 13.0124$	$d_c = 13.0124/k = 9$
R15	**0.9928**	0.9357	0.9910	**0.9928**
parameter	$k = 15$	$\epsilon = 0.35/MinPts = 5$	$d_c = 0.5887$	$d_c = 0.6551/k = 8$

Table 6
Compare clustering performance with different k by DPC-KNN on *Spiral* data set.

k	4	5	6	7	8	9	16
FM	0.8878	0.8878	0.8878	1	1	0.8028	0.4801
NMI	0.8491	0.8491	0.8491	1	1	0.7196	0.2344
ARI	0.8312	0.8312	0.8312	1	1	0.7039	0.1724

4.2. Generate decision graph

We use DPC algorithm and DPC-KNN algorithm to compare the decision graph of *spiral*. As illustrated in Fig. 2, all the points except the cluster centers have lower value of distance δ in decision graph of DPC-KNN algorithm, which makes cluster centers more notable in decision graph than DPC.

4.3. Detect clusters of irregular shapes

Spiral is applied to evaluate the performance of DPC-KNN algorithm in processing irregular-shaped clusters. In Fig. 3, K-means is unable to aggregate satisfactory cluster result in *Spiral* data set. DPC can find three cluster centers, but it cannot aggregate *Spiral* data set correctly. DBSCAN and DPC-KNN are able to aggregate it efficiently and achieve good cluster results.

4.4. Detect clusters of varying size

As is shown in Fig. 4, K-means is unable to process *Flame* data set successfully. DBSCAN is able to detect two clusters, but it incorrectly identifies two points in the upper left corner and on the edge as noise points. However, DPC and DPC-KNN can aggregate two clusters efficiently.

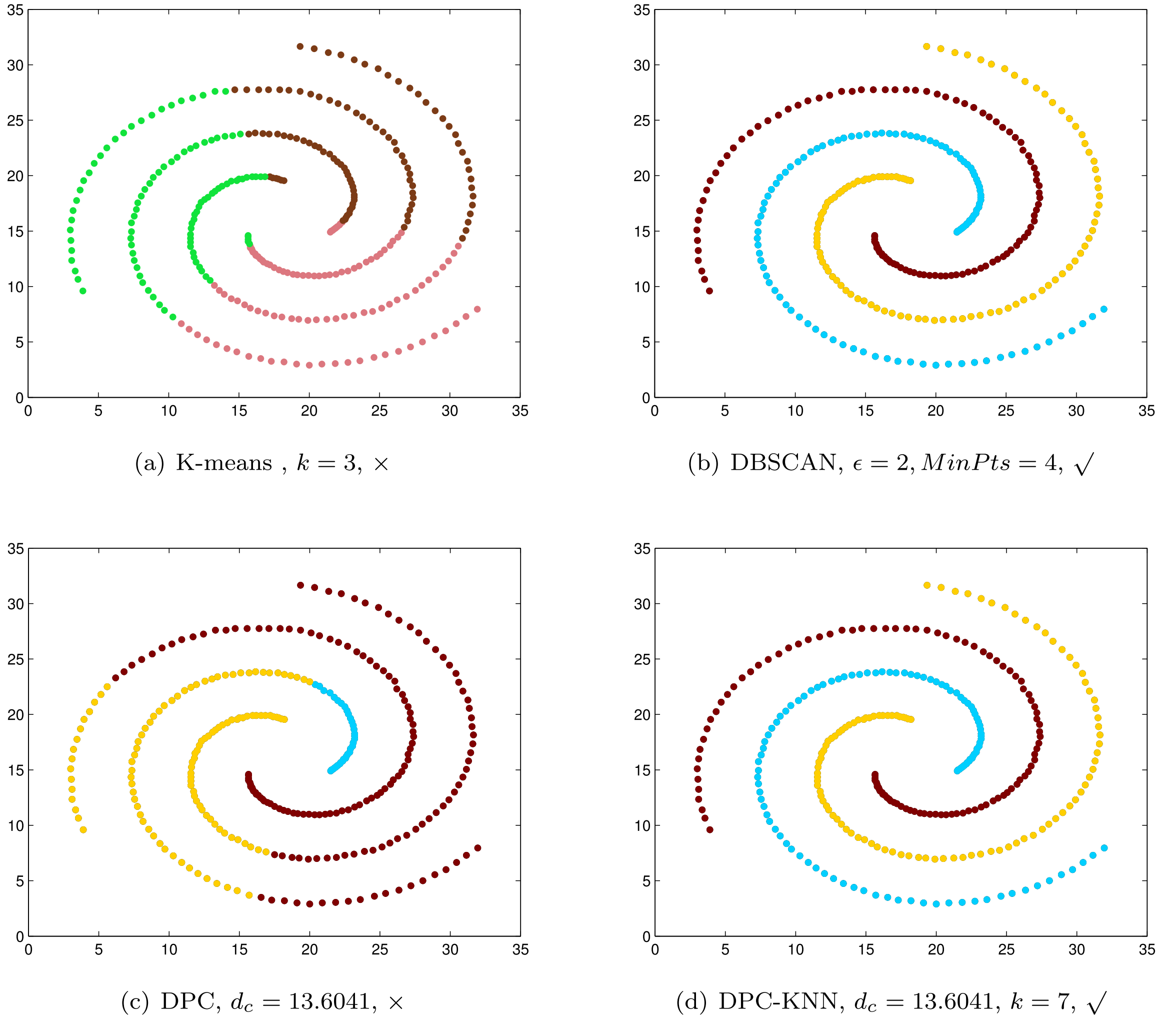
As is illustrated in Fig. 5, K-means can only recognize some certain clusters, but cannot detect all clusters correctly. DBSCAN is unable to find the two clusters on the right and it identifies some edge points as noise points. However, DPC and DPC-KNN are able to perform well on the *Aggregation* data set and achieve good cluster results.

4.5. Aggregate clusters in different values of k

As is shown in Fig. 6, when d_c is 13.6041, DPC detects *Spiral* data set incorrectly. When k is 16, DPC-KNN gets the same cluster result with the same d_c. As is illustrated in Fig. 7, when d_c is 13.6041, DPC-KNN gets different cluster results in different k. In Table 6, it is shown that DPC-KNN can achieve good cluster result when k is 7 or 8. When k is 4 to 6, DPC-KNN gets the same cluster evaluation on *Spiral* data set. When k is 16, DPC-KNN gets lower values of indexes on *Spiral* data set. It is seen that DPC-KNN algorithm is influenced by different values of k.

5. Discussion

To analyze the strengths and weaknesses of DPC-KNN algorithm, its performance is discussed in cluster detection.



(a) K-means , $k = 3$, ×

(b) DBSCAN, $\epsilon = 2, MinPts = 4$, $\surd$

(c) DPC, $d_c = 13.6041$, ×

(d) DPC-KNN, $d_c = 13.6041$, $k = 7$, $\surd$

Fig. 3. Aggregate the data set of *Spiral*.

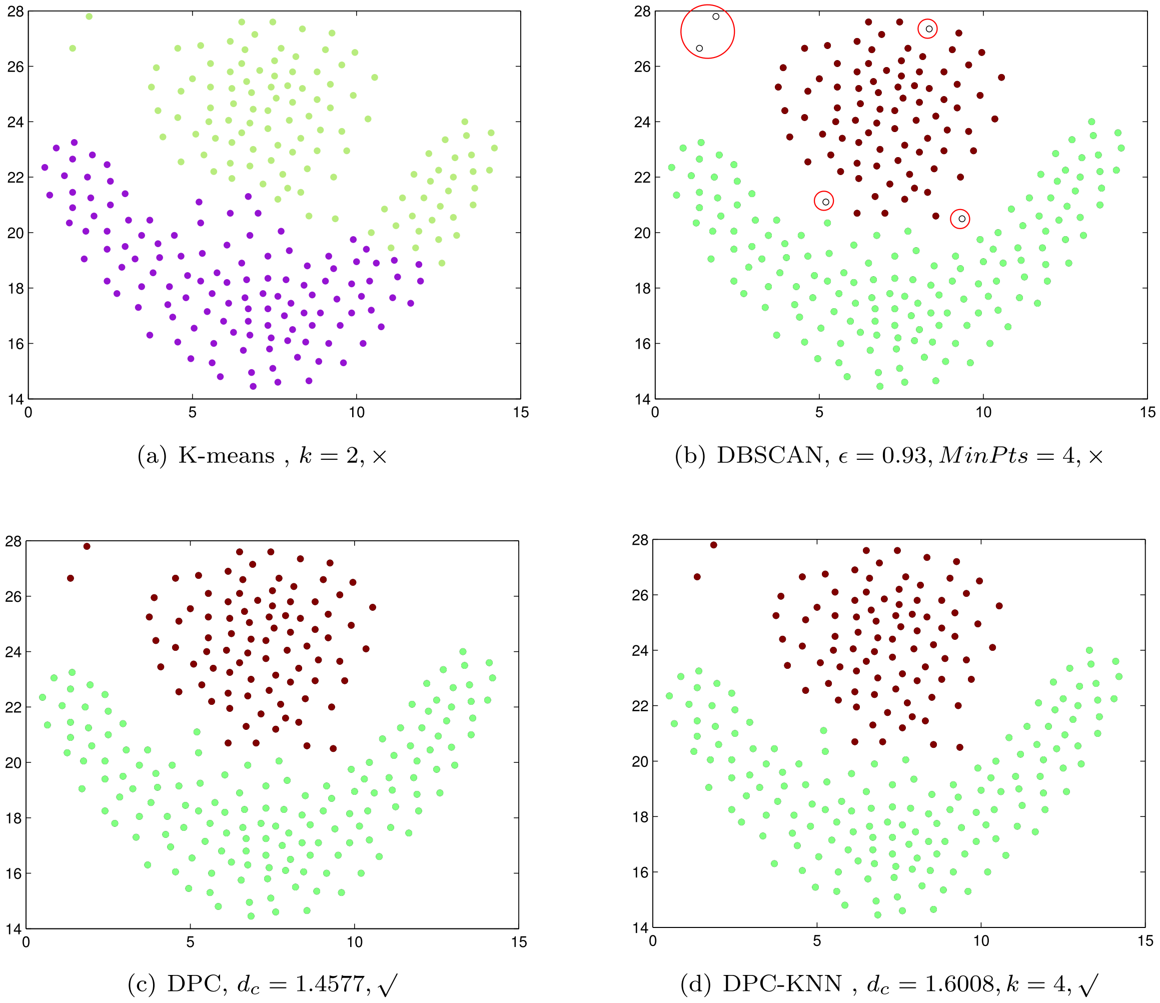
5.1. Analysis of generating decision graph

In DPC algorithm, the value δ of point P_i is minimum distance between point P_i and point P_t belonged to set H_i. DPC-KNN considers k nearest neighbors and applies it to improve the calculation of distance δ. For any point P_i except the cluster centers, if its k nearest neighbors are of lower density than point P_i, then δ_i is minimum distance between point P_l that belongs to set S_i and point P_t that belongs to set H_i. If one of its neighbors' density is higher than point P_i, then the minimum distance is zero, so δ_i is zero. Therefore, in the decision graph generated by DPC-KNN, the δ of most points are zero, and the δ of some points are smaller, so the cluster centers are more notable.

5.2. Analysis of detecting irregular shapes

Spiral data set is a typical non-spherical data set, which brings challenges to most clustering algorithms. As is shown in Fig. 3, K-means and basic DPC algorithm cannot detect *Spiral* data set. DBSCAN and DPC-KNN are able to aggregate it efficiently. K-means is able to process data sets of regular shapes but not to non-spherical data sets. DBSCAN has the capability of processing data sets of arbitrary shapes.

According to assignment process of DPC algorithm, each point is assigned to the same cluster to which its nearest point of higher density belongs. In Fig. 1, data points are marked by numbers following density order and No. 1 represents the data point of the highest density. Points No. 26 and No. 91 are the two nearest neighbors of higher density to point No. 103. It is obvious that distance $d_{(26,103)}$ is shorter than distance $d_{(91,103)}$. Therefore, point No. 103 is assigned to the same cluster as point No. 26 is in. Once point No. 103 is assigned to the wrong cluster, other points of lower density around it



(a) K-means , $k = 2, \times$

(b) DBSCAN, $\epsilon = 0.93, MinPts = 4, \times$

(c) DPC, $d_c = 1.4577, \surd$

(d) DPC-KNN , $d_c = 1.6008, k = 4, \surd$

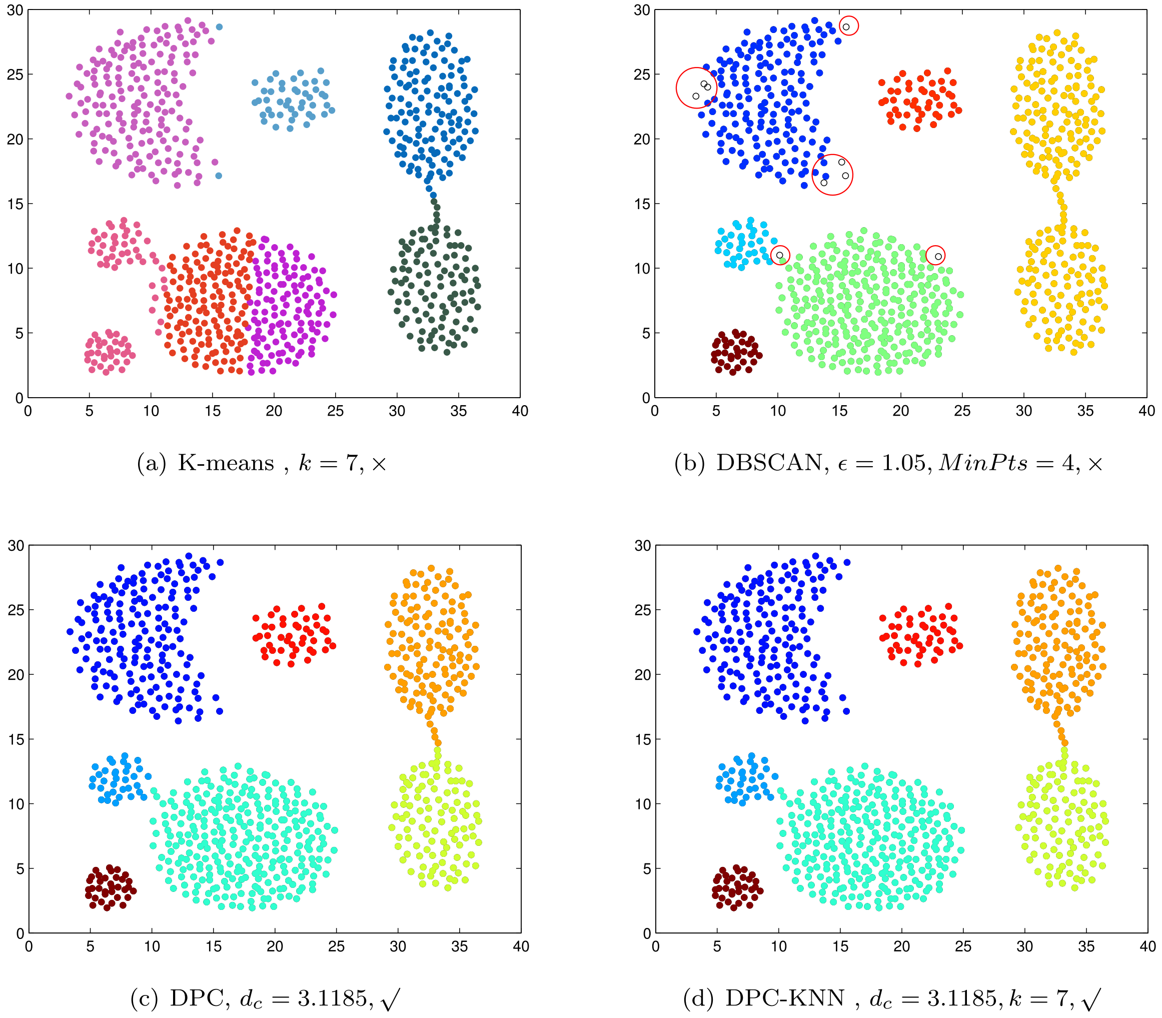
Fig. 4. Aggregate the data set of *Flame*.

are assigned to the same wrong cluster, triggering the domino effect. Therefore, DPC is unable to process *Spiral* data set successfully.

In assignment process of DPC-KNN algorithm, point P_i is assigned to the same cluster as point P_t in the set H_i which has minimum distance to point P_l in the set S_i. In Fig. 8, for point No. 103, its k nearest neighbors are No. 104, No. 105, No. 111, No. 114, No. 119, No. 123 and No. 126 when k is 7. It is seen that $d_{(91,105)}$ is the minimum distance from point No. 103 and its k nearest neighbors to the points of higher density than point No. 103. Therefore, the value of distance δ of point No. 103 is $d_{(91,105)}$. And the point No. 103 is assigned to the same cluster as point No. 91 is in. Once point No. 103 is assigned to the correct cluster, its neighbors of lower density are assigned to the same cluster, which achieves good cluster results. After integrating the k nearest neighbors into DPC-KNN algorithm, the calculation of distance δ and the assignment process become more reasonable. Therefore, DPC-KNN is able to process *Spiral* data set successfully.

5.3. Analysis of detecting varying size

K-means is hard to decide the initial partitions, which has weak ability of discovering arbitrary-shaped clusters [1,5,9], therefore, it is unable to process *Flame* and *Aggregation* data sets. In Figs. 4(b) and 5(b), DBSCAN is able to find some cluster centers and detect some clusters correctly, but it is unable to find all clusters completely. Although DBSCAN has a notion of noise, and is robust to outliers [10], its accuracy in noise detection needs to be improved. DPC can find correct cluster centers on both two data sets, and it can aggregate them by one-step assignment process. DPC-KNN inherits the advantages of DPC, so it can also detect *Flame* and *Aggregation* data sets.



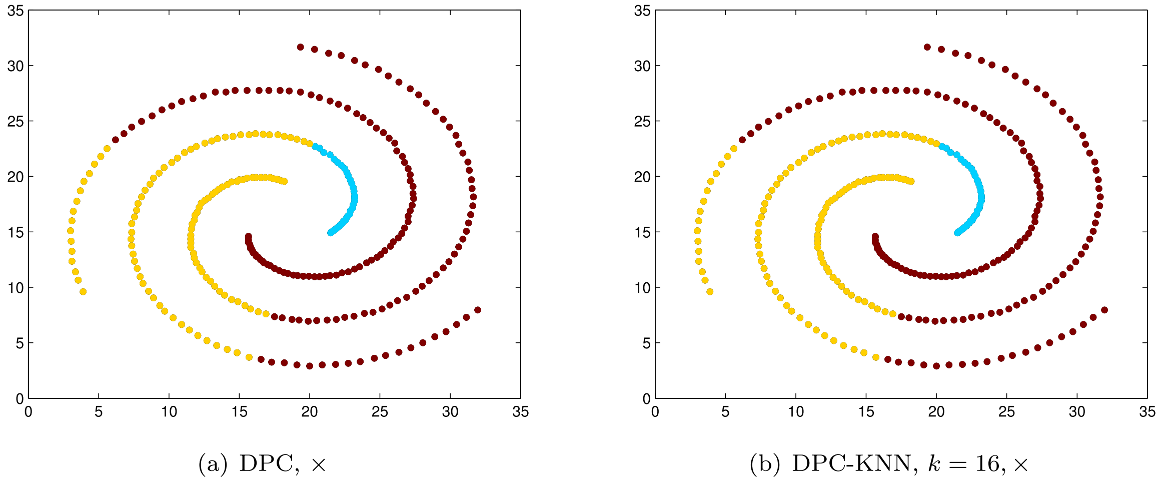
(a) K-means , $k = 7, \times$

(b) DBSCAN, $\epsilon = 1.05, MinPts = 4, \times$

(c) DPC, $d_c = 3.1185, \surd$

(d) DPC-KNN , $d_c = 3.1185, k = 7, \surd$

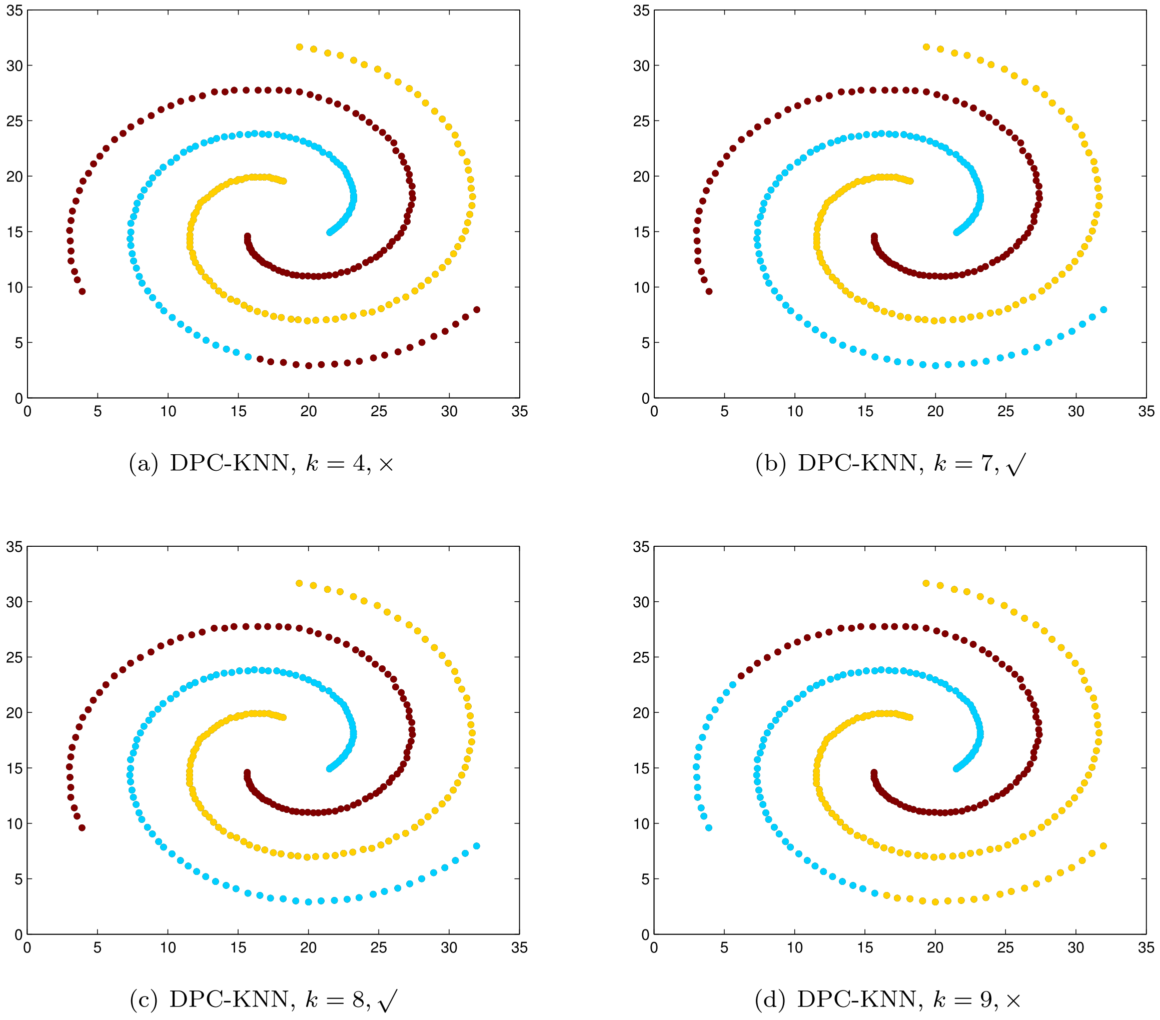
Fig. 5. Aggregate the data set of *Aggregation*.



(a) DPC, $\times$

(b) DPC-KNN, $k = 16, \times$

Fig. 6. Aggregate the data set of *Spiral* with $d_c = 13.6041$.



(a) DPC-KNN, $k = 4$, ×

(b) DPC-KNN, $k = 7$, $\surd$

(c) DPC-KNN, $k = 8$, $\surd$

(d) DPC-KNN, $k = 9$, ×

Fig. 7. Aggregate the data set of *Spiral* with $d_c = 13.6041$ in different k.

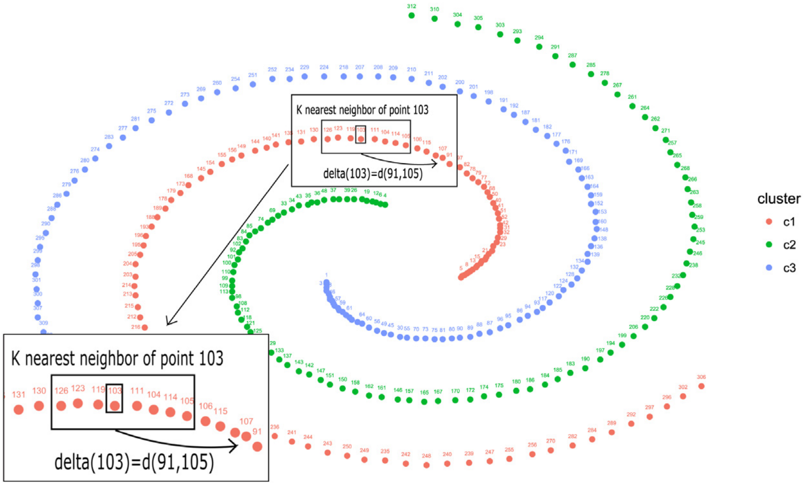


Fig. 8. Sort density by **DPC-KNN** on *Spiral* data set, $d_c = 13.6041$, $\surd$.

5.4. Analysis of aggregating clusters in different values of k

As is shown in Fig. 6, when k is 16, DPC-KNN gets the same cluster result as DPC with $d_c = 13.6041$. In Fig. 7, DPC-KNN is influenced by the values of k, different values of k have different results when d_c is the same. As is illustrated in Table 6, DPC-KNN achieves good performance on *Spiral* data set when k is 7 or 8 with $d_c = 13.6041$. Therefore, k is not fixed in DPC-KNN algorithm, and the effect of k value will affect the cluster result to a large extent.

6. Conclusion

DPC has deficiency in assignment process, which is easy to trigger **domino effect**. Especially, it cannot process some non-spherical data sets such as *Spiral*. Absorbing k nearest neighbors, DPC-KNN is able to make assignment process more reasonable and ensure the effectiveness of the algorithm. DPC-KNN integrates the idea of k nearest neighbors into the distance computation and assignment process. It can be seen from experimental results that the DPC-KNN algorithm is more feasible and effective, compared with K-means, DBSCAN and DPC. DPC-KNN has good performance in processing non-spherical clusters and various sizes clusters. It is able to generate decision graph with cluster centers that are more notable. However, how to determine the k value of DPC-KNN algorithm automatically, and find the relationship between parameters d_c and k needs a further research.

References

[1] R. Xu, D. Wunsch, Survey of clustering algorithms, IEEE Trans. Neural Netw. 16 (3) (2005) 645–678.
[2] B.J. Frey, D. Dueck, Clustering by passing messages between data points, Science 315 (5814) (2007) 972–976.
[3] E.R. Hruschka, R.J. Campello, A.A. Freitas, A.D. Carvalho, A survey of evolutionary algorithms for clustering, IEEE Trans. Syst. Man Cybern. 39 (2) (2009) 133–155.
[4] A. Jain, M. Murty, P. Flynn, Data clustering: a review, ACM Comput. Surv. 31 (3) (1999) 264–323.
[5] A.K. Jain, Data Clustering: 50 Years Beyond K-means, Springer Berlin Heidelberg, 2008, pp. 3–4.
[6] Y. Shi, L. Li, Y. Wang, J. Chen, H.E. Stanley, A study of Chinese regional hierarchical structure based on surnames, Physica A 518 (2019) 169–176.
[7] M. Wang, L. Zhao, R. Du, C. Wang, L. Chen, L. Tian, H.E. Stanley, A novel hybrid method of forecasting crude oil prices using complex network science and artificial intelligence algorithms, Appl. Energy 220 (2018) 480–495.
[8] S.P. Lloyd, Least squares quantization in PCM, IEEE Trans. Inform. Theory 28 (2) (1982) 129–137.
[9] J. Han, M. Kamber, Data Mining: Concepts and Technique, third ed., Morgan Kaufmann Publishers Inc., 2011.
[10] M. Ester, H.P. Kriegel, J. Sander, X. Xu, A density-based algorithm for discovering clusters in large spatial databases with noise, in: International Conference on Knowledge Discovery and Data Mining, 1996, pp. 226–231.
[11] J. Xie, H. Gao, W. Xie, K-nearest neighbors optimized clustering algorithm by fast search and finding the density peaks of a dataset, Sci. Sin. 46 (2) (2016) 258.
[12] A. Rodriguez, A. Laio, Clustering by fast search and find of density peaks, Science 344 (6191) (2014) 1492–1496.
[13] M. Du, S. Ding, H. Jia, Study on density peaks clustering based on k-nearest neighbors and principal component analysis, Knowl. Based Syst. 99 (2016) 135–145.
[14] M. Wang, W. Zuo, Y. Wang, An improved density peaks-based clustering method for social circle discovery in social networks, Neurocomputing 179 (2016) 219–227.
[15] Y. Chen, D. Lai, H. Qi, J. Wang, J. Du, A new method to estimate ages of facial image for large database, Multimedia Tools Appl. 75 (5) (2016) 2877–2895.
[16] C. Wiwie, J. Baumbach, R. Rottger, Comparing the performance of biomedical clustering methods, Nature Methods 12 (11) (2015) 1033–1038.
[17] K. Sun, X. Geng, L. Ji, Exemplar component analysis: a fast band selection method for hyperspectral imagery, IEEE Geosci. Remote Sens. Lett. 12 (5) (2015) 998–1002.
[18] J. Jiang, D. Hao, Y. Chen, M. Parmar, K. Li, GDPC: Gravitation-based density peaks clustering algorithm, Physica A 502 (15) (2018) 345–355.
[19] J. Jiang, X. Tao, K. Li, DFC: Density fragment clustering without peaks, J. Intell. Fuzzy Systems 34 (1) (2018) 525–536.
[20] X. Xu, S. Ding, Z. Shi, An improved density peaks clustering algorithm with fast finding cluster centers, Knowl. Based Syst. 158 (2018) 65–74.
[21] S. Liu, B. Zhou, D. Huang, L. Shen, Clustering mixed data by fast search and find of density peaks, Math. Probl. Eng. 2017 (2017) 1–7.
[22] J. Jiang, Y. Chen, D. Hao, K. Li, DPC-LG: Density peaks clustering based on logistic distribution and gravitation, Physica A 514 (2019) 25–35.
[23] D. Dua, C. Graff, UCI Machine Learning Repository, University of California, Irvine, School of Information and Computer Sciences, 2017, http://archive.ics.uci.edu/ml.
[24] P. Tsaparas, H. Mannila, A. Gionis, Clustering aggregation, ACM Trans. Knowl. Discov. Data 1 (1) (2007) 4.
[25] L. Fu, E. Medico, FLAME, a novel fuzzy clustering method for the analysis of DNA microarray data, BMC Bioinformatics 8 (1) (2007) 3.
[26] H. Chang, D.Y. Yeung, Robust path-based spectral clustering, Pattern Recognit. 41 (1) (2008) 191–203.
[27] A.K. Jain, M.H.C. Law, Data clustering: a user's dilemma, Lecture Notes in Comput. Sci. 3776 (2005) 1–10.
[28] C.J. Veenman, M.J. Reinders, E. Backer, A maximum variance cluster algorithm, IEEE Trans. Pattern Anal. Mach. Intell. 24 (9) (2002) 1273–1280.
[29] D.M.W. Powers, Evaluation: from precision, recall and f-measure to ROC, informedness, markedness and correlation, J. Mach. Learn. Technol. 2 (1) (2011) 37–63.
[30] N.X. Vinh, J. Epps, J. Bailey, Information theoretic measures for clusterings comparison: variants, properties, normalization and correction for chance, J. Mach. Learn. Res. 11 (2010) 2837–2854.
[31] D. Pfitzner, R. Leibbrandt, D. Powers, Characterization and evaluation of similarity measures for pairs of clusterings, Knowl. Inf. Syst. 19 (3) (2009) 361–394.