Int. J. Mach. Learn. & Cyber. (2018) 9:1335–1349
DOI 10.1007/s13042-017-0648-x

ORIGINAL ARTICLE

Density peaks clustering using geodesic distances

Mingjing Du[1] · Shifei Ding[1,2] · Xiao Xu[1] · Yu Xue[3]

Received: 18 December 2015 / Accepted: 24 January 2017 / Published online: 2 March 2017

Abstract Density peaks clustering (DPC) algorithm is a novel clustering algorithm based on density. It needs neither iterative process nor more parameters. However, it cannot effectively group data with arbitrary shapes, or multi-manifold structures. To handle this drawback, we propose a new density peaks clustering, i.e., density peaks clustering using geodesic distances (DPC-GD), which introduces the idea of the geodesic distances into the original DPC method. By experiments on synthetic data sets, we reveal the power of the proposed algorithm. By experiments on image data sets, we compared our algorithm with classical methods (kernel k-means algorithm and spectral clustering algorithm) and the original algorithm in accuracy and NMI. Experimental results show that our algorithm is feasible and effective.

Keywords Data clustering · Density peaks clustering · Geodesic distances

✉ Shifei Ding
dingsf@cumt.edu.cn

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

[2] Key Laboratory of Intelligent Information Processing, Institute of Computing Technology, Chinese Academy of Sciences, Beijing 100090, China

[3] School of Computer and Software, Nanjing University of Information Science and Technology, Nanjing 210044, China

1 Introduction

Clustering is perhaps the most important and widely used method of unsupervised learning: [1, 2] It is the problem of identifying groupings of similar points which are relatively 'isolated' from each other, or in other words, partitioning the data into dissimilar groups of similar items [3–5] . Clustering methods are generally divided into five groups: hierarchical clustering, partitioning clustering, density-based clustering, grid-based clustering and model-based clustering [6–9].

In density-based clustering, clusters are defined as areas of higher density than the remainder of the data set. Density peaks clustering (DPC) algorithm [10] proposed by Rodriguez and Laio is a new density-based clustering method and does not require one to specify the number of clusters. This method is robust with respect to choosing d_c as the only parameter. DPC 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. It is important to note that this method is robust with respect to changes in the metric which do not significantly affect the distances below d_c.

Currently density peaks clustering algorithm is used in outlier detection [11], image processing [12–14], document processing [15], etc. However, DPC still has some defects. DPC is not quite appropriate for solving the problem which requires more cluster centers for subsequent analysis. In order to improve its capacity, Tang et al. [16] proposed an enhanced fast density peak-based clustering (E-FDPC). In addition, DPC dose not perform well when there are more than one density peak in one cluster, which was named as "no density peaks". Inspired by the idea of a hierarchical clustering algorithm CHAMELEON, Zhang et al. [17] proposed an

extension of CFSFDP (E_CFSFDP). The selection of the key intrinsic parameters in DPC algorithm was not systematically investigated, so Wang et al. [18] proposed a clustering procedure with adaptive density peak detection. In addition, Du et al. [19] proposed a density peaks clustering based on k nearest neighbors (DPC-KNN).

Despite these drawbacks, a significant one is that the geometry of the distribution of the data has not been taken into account in DPC. DPC cannot effectively group data with arbitrary shapes, or multi-manifold structures. However, the data with multi-manifold structures is ubiquitous in real-world pattern recognition tasks. In many practical problems, e.g., handwritten digit recognition, image segmentation, and web analysis etc. These multiple low-dimensional manifolds embedded in high-dimensional data always are non-spherical shapes. Specifically, this algorithm is not able to find clusters of twisted, folded, or curved data on synthetic data sets. For example, Fig. 1 presents that clusters cannot be all detected.

Many manifold learning algorithms [20–23] are used to find the geometry of the data. In these algorithms, the earliest and most simple one is Tenenbaum's algorithm, Isomap [20], which uses the geodesic distances (GD), in the combination with multidimensional scaling (MDS), to learn the geometric structure of nonlinear manifolds. In order to overcome this problem, we propose a density peaks clustering using geodesic distances (DPC-GD) which introduces the idea of GD into DPC.

The proposed DPC-GD is based on density peaks clustering algorithm and the geodesic distances. The rest of this paper is organized as follows. In Sect. 2, we describe the principle of the DPC method and introduce the geodesic distances. In Sect. 3, we make a detailed description of DPC-GD. In Sect. 4, we present experimental results on synthetic data sets and image data sets, then we analyze the performance of the proposed algorithm. Finally, some conclusions and the intending work are given in the last section.

2 Related works

This section provides brief reviews of DPC and GD.

2.1 Density peaks clustering

Rodriguez and Laio proposed an algorithm published in the US journal Science. Its idea is that cluster centers are characterized by a higher density than their neighbors and by a relatively large distance from points with higher densities. This method utilizes two important quantities: One is the local density ρ_i of each point i, and the other is its distance δ_i from points of higher density. The two quantities correspond to two assumptions with respect to the cluster centers. One is that the cluster centers are surrounded by neighbors with a lower local density. The other is that they have relatively larger distance to the points of higher density. In the following, we will describe the computation of ρ_i and δ_i in much more detail.

Assume that the data set is $\mathbf{X}_{N\times M} = \left[\mathbf{x_1}, \mathbf{x_2}, \ldots, \mathbf{x_N}\right]^T$, where $\mathbf{x}_i = \left[x_{1i}, x_{2i}, \ldots, x_{Mi}\right]^T$ is the vector with M attributes and N is the number of points. The distance matrix of the data set needs to be computed first. Let $d(i,j)$ denote the Euclidean distance between the point i and the point j, as follows:

$$d(i,j) = \left\|\mathbf{x}_i - \mathbf{x}_j\right\| \tag{1}$$

The local density of a point i, denoted by ρ_i, is defined as:

$$\rho_i = \sum_j \chi\left(d(i,j) - d_c\right)$$
$$\chi(x) = \begin{cases} 1, & x<0 \\ 0, & x>0 \end{cases} \tag{2}$$

where d_c is a cutoff distance. ρ_i is defined as the number of points which are adjacent to point i. There is another local density computation in the code presented by Rodriguez and Laio. If the former is called a hard threshold, the latter will be called a soft threshold. Specifically, ρ_i is defined as a Gaussian kernel function, as follows:

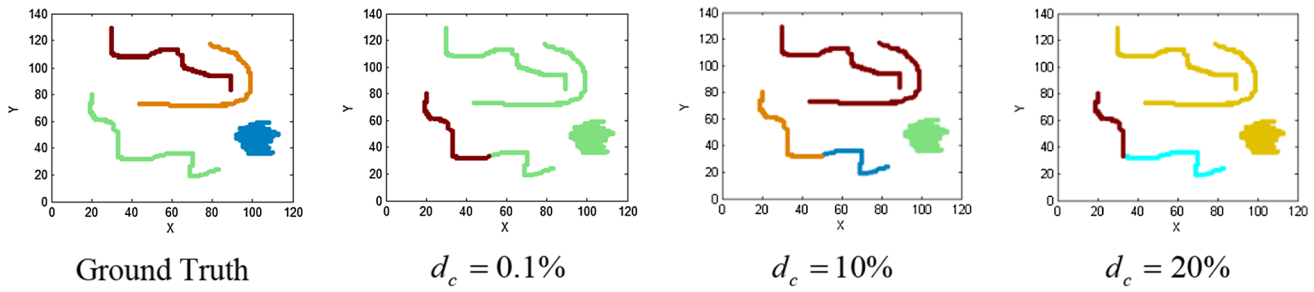


Fig. 1 DPC on these clusters of twisted and folded manifold

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

where d_c is an adjustable parameter, controlling the weight degradation rate.

$d_c$is the only variable in Eqs. 2 and 3. The process for selecting d_c is actually that for selecting the average number of neighbors of all points on the data set. In the code, d_c is defined as:

$$d_c = d_{\left\lceil N_d \times \frac{p}{100} \right\rceil} \tag{4}$$

where $N_d = \binom{N}{2}$ and $d_{\left\lceil N_d \times \frac{p}{100} \right\rceil} \in D = \left[d_1, d_2, \ldots, d_{N_d}\right]$. D is a set of all the distances between every two points on the data set, which are sorted in ascending order. N denotes the number of points on the data set. $\left\lceil N_d \times \frac{p}{100} \right\rceil$ is the subscript of $d_{\left\lceil N_d \times \frac{p}{100} \right\rceil}$, where $\lceil \bullet \rceil$ is the ceiling function and p is a percentage.

The computation of δ_i is quite simple. The minimum distance between the point i and any other points with higher density, denoted by δ_i, is defined as:

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

Only those points with relatively high ρ_i and high δ_i are considered as cluster centers. The points with high ρ_i and δ_i value are also called as peaks which have higher densities than other points. A point is assigned to the same cluster as its nearest neighbor peak.

After cluster centers have been found, DPC assigns each of the remaining points to the same cluster as its nearest neighbors with higher density. A representation named as decision graph is introduced to help one to make a decision. This representation is the plot of δ_i as a function of ρ_i for each point.

The following algorithm is a summary of DPC.

Algorithm1. DPC algorithm.

Inputs:

The samples $\mathbf{X} \in \mathbb{R}_{N \times M}$

The parameter d_c

Outputs:

The label vector of cluster index: $\mathbf{y} \in \mathbb{R}_{N \times 1}$

Method:

Step 1: Calculate distance matrix according to Eq. 1 or other metrics

Step 2: Calculate ρ_i for point i according to Eq.2 or Eq. 3

Step 3: Calculate δ_i for point i according to Eq. 5

Step 4: Plot decision graph and select cluster centers

Step 5: Assign each remaining point to the nearest cluster center

Step 6: **Return y**

2.2 Geodesic distances

Indeed, DPC encounters difficulties when the data are nonlinear structures like the spiral illustrated in Fig. 2a. The distance between two points is measured by the traditional metric like in Fig. 2b. Some traditional metrics represented by Euclidean distance are able to reveal the intrinsic geometry of the data. Instead, they have to be measured like in Fig. 2c, along the spiral.

We seek to reveal the intrinsic geometry of the data, as captured in the geodesic manifold distances between all pairs of data points. The crux is estimating the geodesic distance between faraway points, given only input-space distances. For neighboring points, input space distance provides a good approximation to the geodesic distance. For faraway

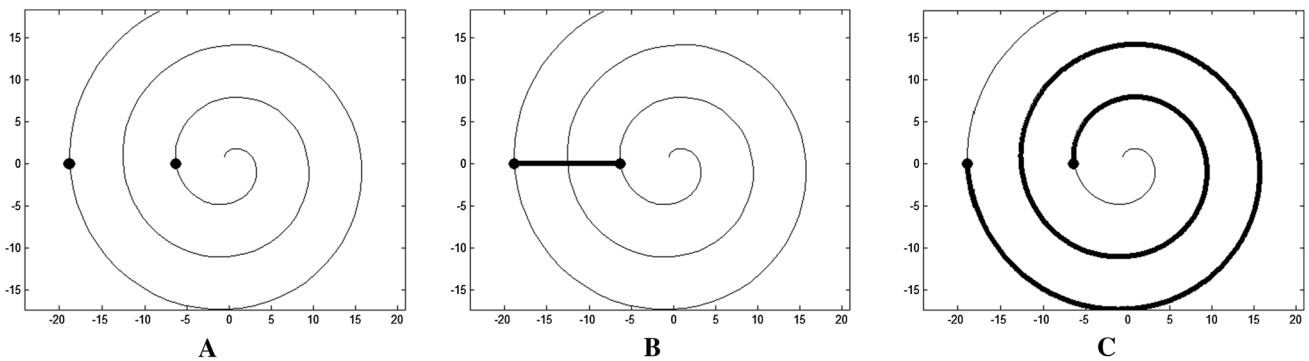


Fig. 2 Two points in a spiral

points, geodesic distance can be approximated by adding up a sequence of "short hops" between neighboring points. These approximations are computed efficiently by finding shortest paths in a graph with edges connecting neighboring data points.

3 Density peaks clustering using geodesic distances

The intrinsic geometry of data is not assessed in DPC algorithm. For this reason, we propose a novel density peaks clustering using geodesic distances (DPC-GD).

3.1 The description of the algorithm

DPC-GD tries to combine the bests of DPC and Isomap by applying geodesic distances to DPC. Calculating distance matrix consists of two steps: calculate "old" distance matrix and apply the geodesic distances to calculating "new" distance matrix. These two steps are shown in Fig. 3.

This so-called geodesic distance is approximated in the following way. First, the neighborhood for each point is calculated. For example, the neighborhood of a point may be the k nearest points. Once the neighborhoods are known, a graph is built, by linking all neighboring points. Next, each arc of the graph is labelled with the Euclidean distance or other metrics between the corresponding linked points. Finally, the geodesic distance between two points is approximated by the sum of the arc lengths along the shortest path linking both points. Practically, the shortest path is computed by Floyd's algorithm.

Define the graph G over all data points by connecting points i and j, if i is one of the k nearest neighbors of j. Set edge lengths equal to $d(i,j)$ according to Eq. 1.

Let $d_G(i,j)$ denote the geodesic distance between the point i and the point j, as follows:

$$d_D(i,j) = \begin{cases} d(i,j), & \text{if } i,\ j \text{ are linked by an edge} \\ \infty, & \text{otherwise} \end{cases} \tag{6}$$

Then for each value of $l = 1, 2, \ldots N$ in turn, replace all entries $d_G(i,j)$ by

$$d_G(i,j) = \min\left\{d_G(i,j), d_G(i,l) + d_G(l,j)\right\} \tag{7}$$

A new distance matrix of final values $D_G = \left\{d_G(i,j)\right\}$ will contain the shortest path distances between all pairs of points in G.

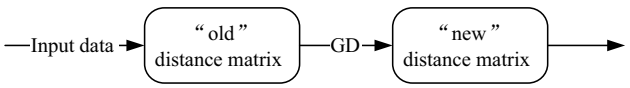


Fig. 3 Two steps of calculating distance matrix

Then we can adopt the idea of the geodesic distances to DPC. The following algorithm is a summary of the proposed DPC-GD.

Algorithm2. DPC-GD algorithm.

Inputs:

The samples $\mathbf{X} \in \mathbb{R}_{N\times M}$

The parameter d_c

The parameter k

Outputs:

The label vector of cluster index: $\mathbf{y} \in \mathbb{R}_{N\times 1}$

Method:

Step 1: Calculate distance matrix according to Eq. 1 or other metrics

Step 2: Calculate the k neighborhoods and link neighboring landmarks

Step 3: Run Floyd's algorithm to get the matrix D_G which contains all pairwise geodesic distances

Step 4: Calculate ρ_i for point i according to Eq. 2 or Eq. 3

Step 5: Calculate δ_i for point i according to Eq. 5

Step 6: Plot decision graph and select cluster centers

Step 7: Assign each remaining point to the nearest cluster center

Step 8: **Return y**

3.2 Performance analysis

Complexity Analysis: Suppose N is the total number of points on the data set. The complexity in calculating the similarity matrix is $O(N^2)$. This procedure, known as Floyd's algorithm, requires $O(N^3)$ operations. DPC-GD also needs $O(N^2)$ to compute the local density. In addition, we cost $O(N \log N)$ in the sorting process with quick sort. For the progress to determine the cluster centers, we take no account of the time. As the complexity in assignment procedure is $O(N)$, the total time complexity of our DPC-GD method is$O(N^2) + O(N^3) + O(N^2) + O(N \log N) + O(N) + O(N)$.

4 Experiments and results

In this section, we test the performance of DPC-GD through two types of the experiments. By experiments on synthetic data sets, we compare the proposed algorithm with the original algorithm for grouping data with arbitrary shapes. By experiments on image data sets, we compared our algorithm with kernel k-means algorithm, spectral clustering (SC) algorithm in accuracy.

We conduct experiments in a work station with a core i7 DMI2-Intel 3.6 GHz processor and 18 GB RAM running MATLAB 2012B. We run kernel k-means algorithm, SC algorithm, 10 times on real-world data sets. On synthetic data sets, we measure the similarity between data points with the famous Euclidean distance, which is widely used to measure the similarity of spatial data, as shown in Eq. 1. On synthetic data sets, the similarity between two images is computed with

a new image similarity method, the Complex-Wavelet Structural SIMilarity (CW-SSIM) [24], which is used to measure the similarity between images in the original paper. In DPC-GD, the parameter d_c is selected from {0.1 0.2 0.5 1 2 4 6%}, otherwise, the parameter k of the k neighborhoods is selected from {2 3 4 6 10}. In DPC, we also select the parameter d_c from {0.1 0.2 0.5 1 2 4 6%}. Note that the CW-SSIM index fall in range from 0 to 1, and the higher the value of the CW-SSIM is, the more similar the images will be. So we can apply these indices to producing a similarity matrix of kernel k-means and spectral clustering. However, for ease of utilizing DPC and DPC-GD, we map CW-SSIM index, denoted by $d_O(i,j)$, to the range 0–∞, as follow:

$$d_N(i,j) = -\log\left(d_O(i,j)\right) \tag{8}$$

In this case, the smaller the value of the $d_N(i,j)$ is, the more similar the images will be. If the value of the $d_N(i,j)$ is 0, two images are the same. The most important thing is that converted values are applied to geodesic distances directly.

4.1 Experiments on synthetic data sets

We test the performance of the original algorithm and our algorithm on synthetic data sets. The synthetic data sets are 2 dimension, which makes things easy from the visualization point of view.

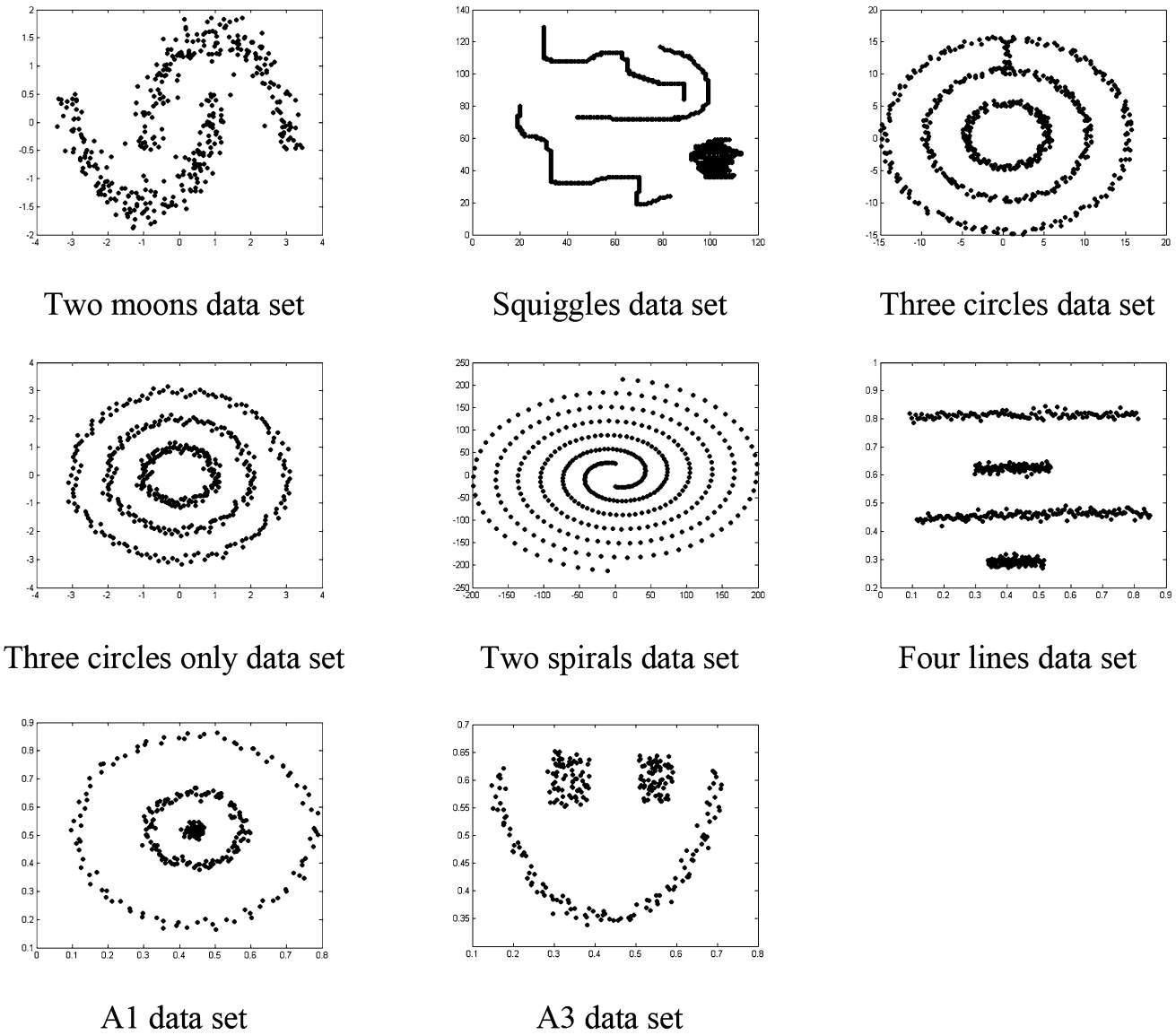


Fig. 4 Visualization of two-dimensional data sets

4.1.1 Synthetic data sets

The original algorithm and proposed algorithm are tested by 8 data sets whose geometric shapes are shown in Fig. 4. The first data set, two moons data set, which is widely used in some manifold learning algorithms [25–28], contains 2 clusters and 400 points. The second data set, squiggles data set [29], consists of 4 clusters which are of 602 points. The third data set, three circles data set [29], contains 2 clusters and 625 points. The fourth data set which is similar to three circles data set contains 450 points. But three circles only data set [30] are of 3 clusters. There are 2 clusters and 378 points in the fifth data set, two spirals data set [30]. The sixth data set [30] contains 4 lines and 512 points. The seventh data set, A1 data set [31], contains 3 clusters and 299 points. The last data set, A3 data set [31], contains 3 clusters and 266 points. We demonstrate the power of our algorithm on these test cases.

4.1.2 The evaluation of clustering results on synthetic data sets

On two moons data set, clustering results proposed by DPC have been given in Fig. 5. DPC is not able to find clusters, when the parameter d_c is selected from {0.1 0.2 0.5 1 2 4 6%}. Even if we give it a greater range of options, the performance of the original algorithm is still poor.

The proposed algorithm gets better clustering performance compared to DPC on two moons data set. Figure 6 presents two clusters found by our algorithm with different d_c, when the parameter k is fixed at 6.

Figure 7 shows clustering results proposed by DPC on four squiggles data set. DPC is not able to find clusters,

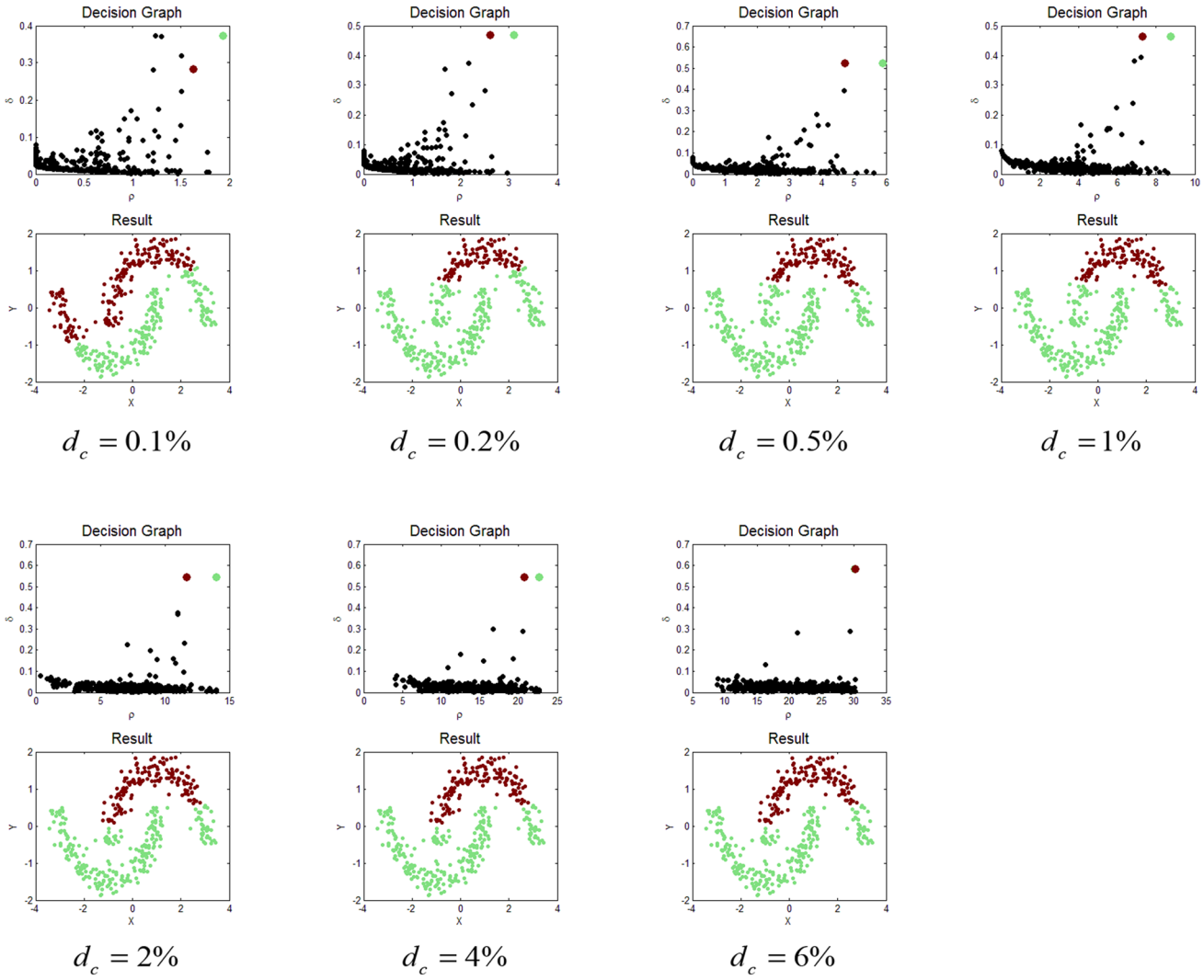


Fig. 5 DPC on two moons data set with different values of d_c

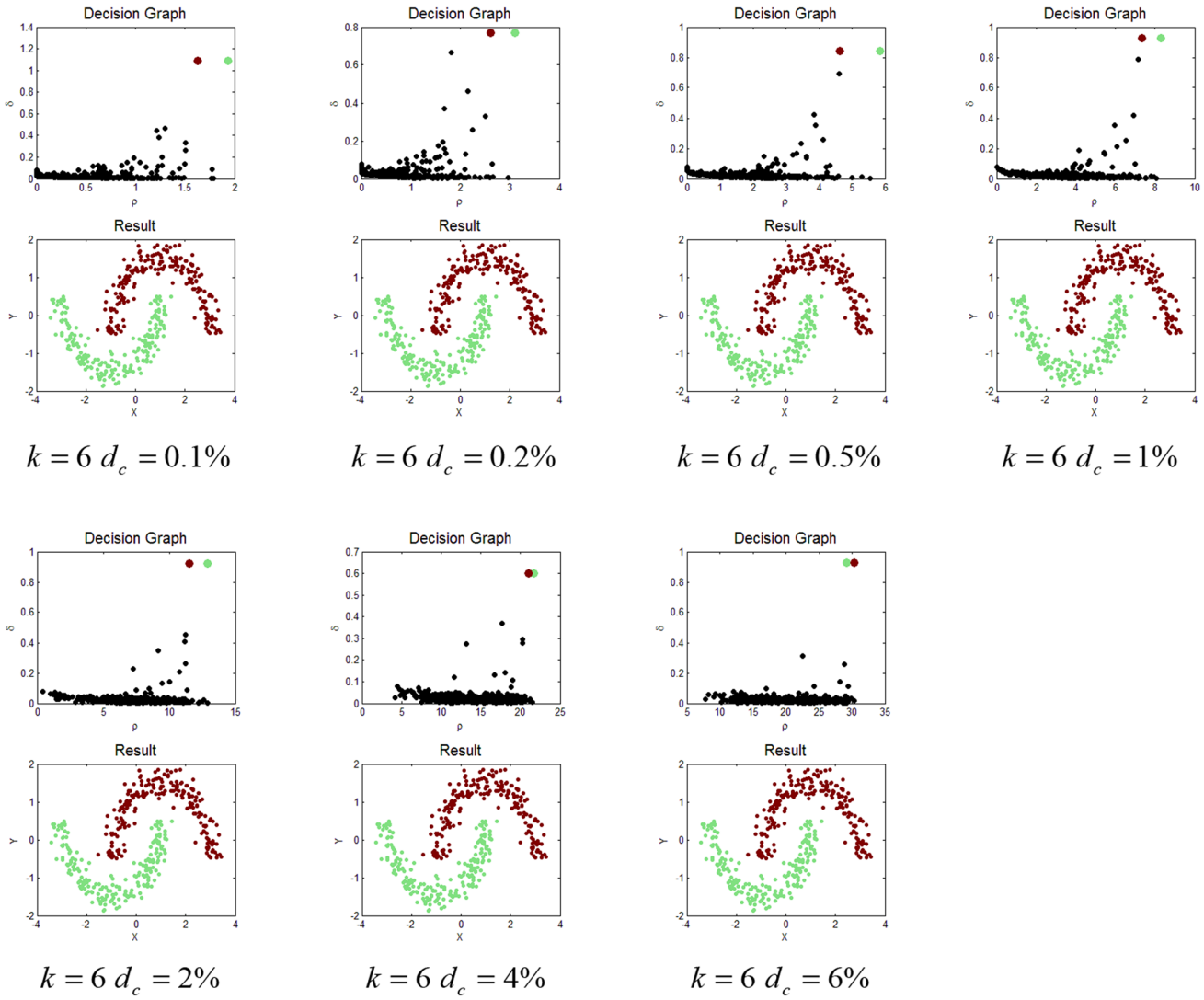


Fig. 6 DPC-GD on two moons data set

when the parameter d_c is selected from {0.1 0.2 0.5 1 2 4 6%}. In the first section, we give it a greater range of options, as shown in Fig. 1. The performance of the original algorithm is still poor.

On four squiggles data set, the experimental results which demonstrate DPC-GD produces very good performance are shown in Fig. 8. Our algorithm achieves satisfactory results with different d_c, when the parameter k is fixed at 4 and 6. The results demonstrate that our approach is robust respect to the parameters.

The original algorithm performs good results in two spirals data set as shown in Fig. 9. Because this data set contains non-Gaussian clusters. Figure 10 shows that our algorithm can also detect clusters very well.

On the remaining few data sets, clustering results proposed by DPC have been given in Fig. 11. And our algorithm produces the results in Fig. 12. For sake of saving space, we only show the best performance on these data sets.

Three circles only data set is similar to three circles data set. The only difference is that there is a line that links up with the two outer circles. The proposed algorithm sharply finds this difference and produces different clustering results. Four lines data set, A1 data set and A3 data set consist of some clusters that are of different size and shape. Our algorithm performs significantly better in all these tests. These experiments illustrate that our algorithm is very effective in grouping data with arbitrary shapes.

4.2 Experiments on image data sets

In this section, we show that challenges such as manifold proximity and non-uniform sampling are also common on

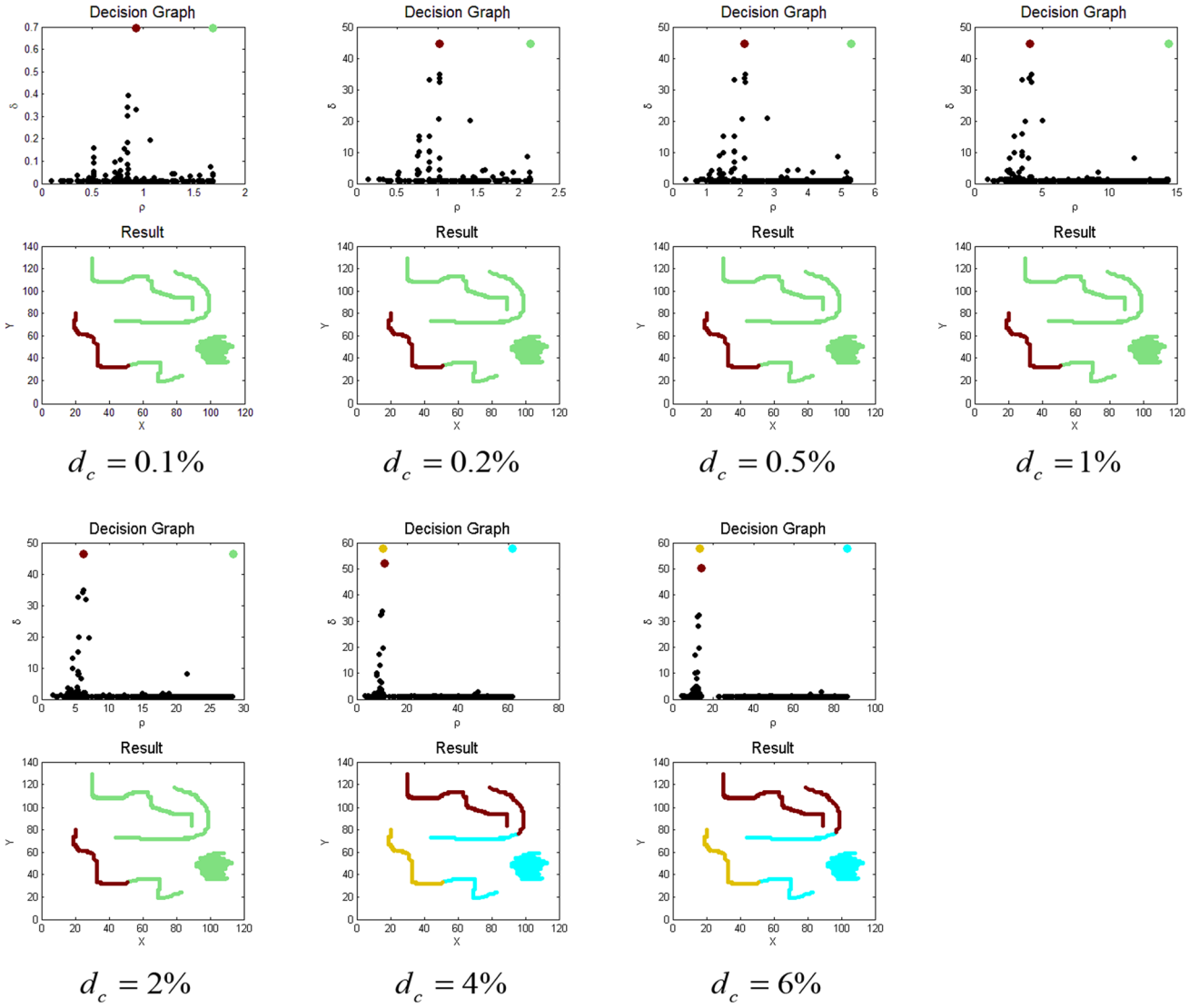


Fig. 7 DPC on four squiggles data set

image data sets, and our algorithm is able to handle these issues effectively.

In order to be more convincible, the performances of our algorithm are compared with that of classical methods (kernel k-means algorithm and spectral clustering algorithm).

4.2.1 Image data sets

Three image data sets used in the experiments include Columbia University Image Library (COIL-20) [32], the UMIST face database [33], the USPS handwritten digit database [34].

COIL-20 consists of 32×32 grey images drawn from 20 objects. Thus, each image is represented by a 1024-dimensional vector. The images of each object are taken 5 degrees apart as the object is rotated on a turntable and each object has 72 images [35]. Figure 13 shows some examples of the first object from this data set.

The UMIST face database (UMIST) consists of 575 images of 20 people. Each covers a range of poses from profile to frontal views. The original files are all in PGM format, approximately 220×220 pixels in 575 shades of grey. Cropped versions of 112×92 images are manually cropped by Daniel Graham. All images of the first 10 people on cropped data set are taken to form this data set which is used in this paper. It consists of 265 images of 10 people. And each image is represented by a 10,304-dimensional vector [36]. Figure 14 shows some images on this data set.

The USPS handwritten digit database (USPS) contains 9298 16×16 handwritten digit images in total, which is then split into 7291 training images and 2007 test images. Finally, we select the clustering of the digits

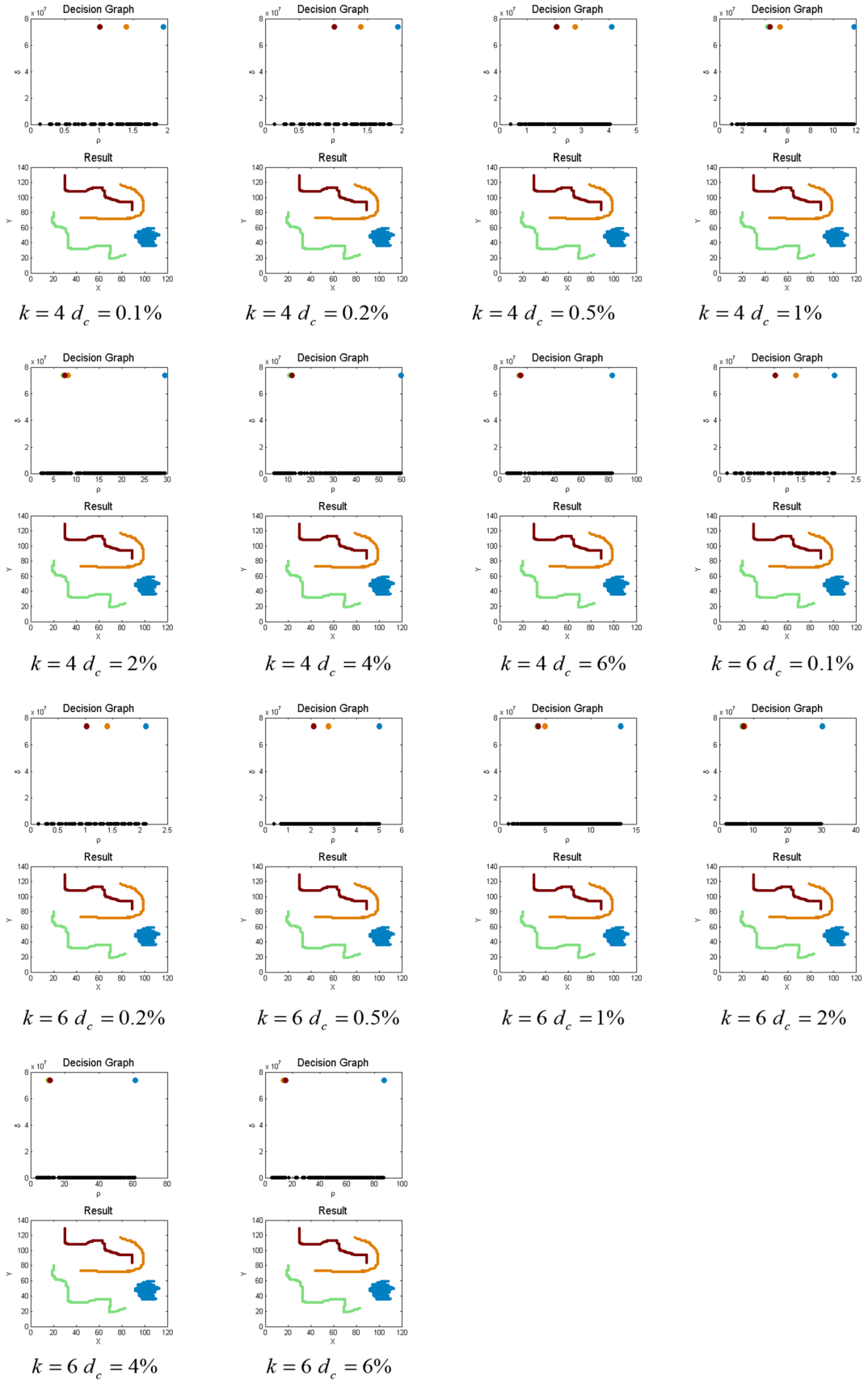


Fig. 8 DPC-GD on four squiggles data set

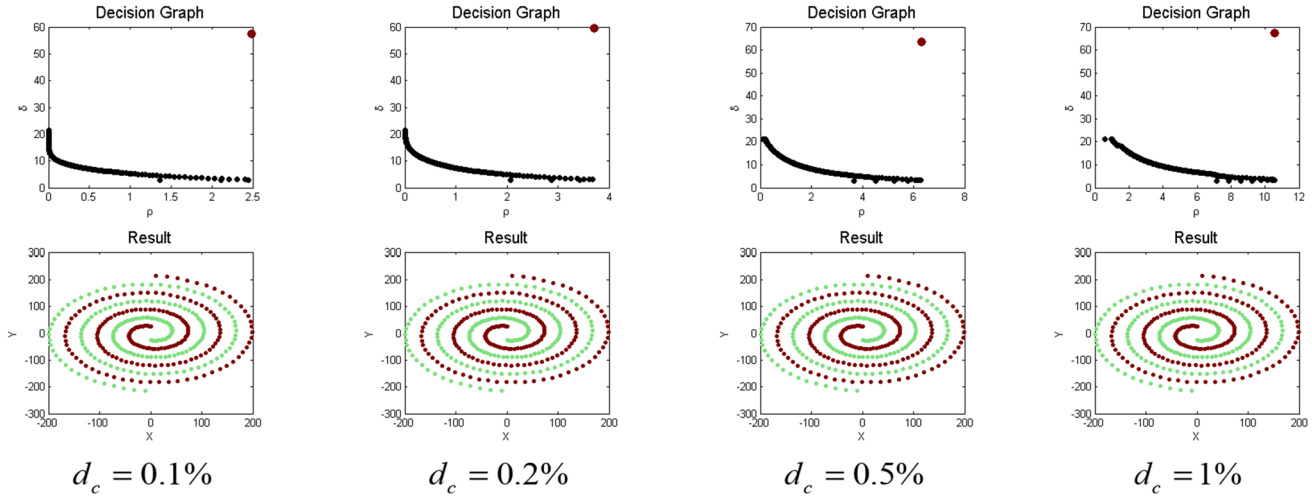


Fig. 9 DPC on two spirals data set

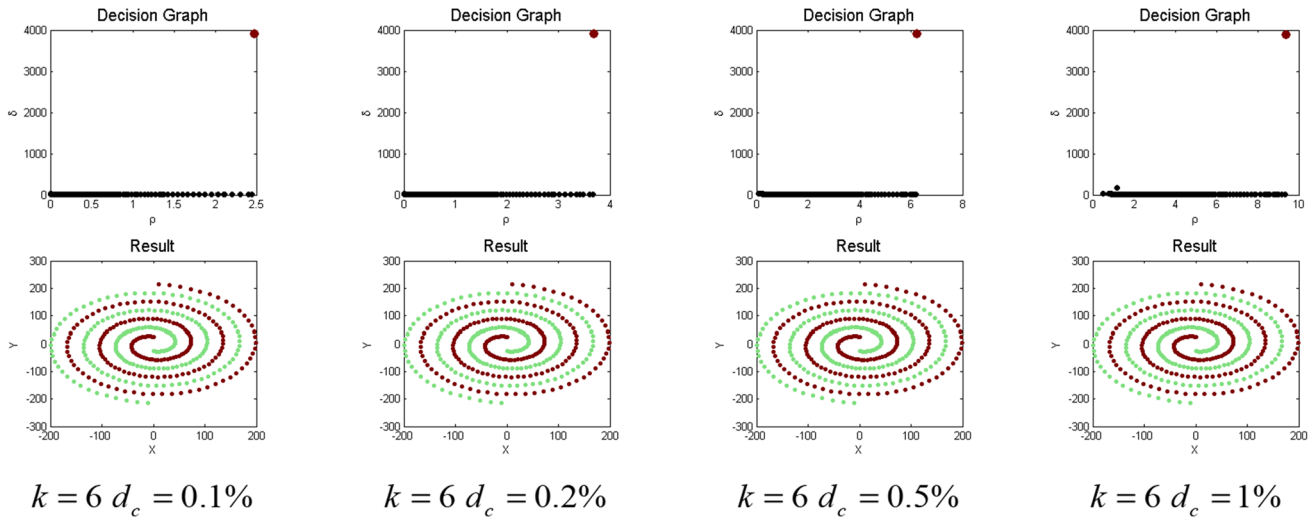


Fig. 10 DPC-GD on two spirals data set

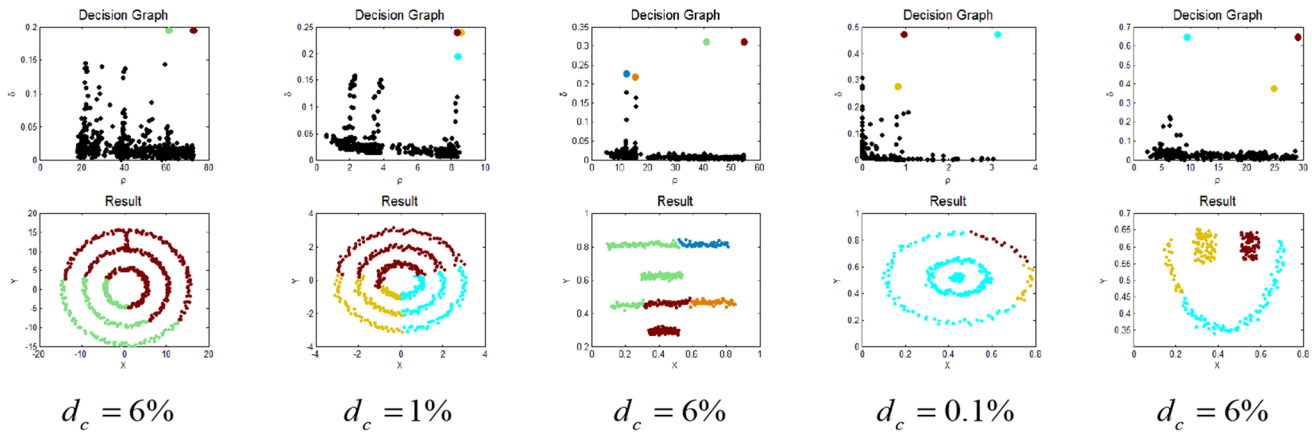


Fig. 11 DPC on the remaining few data sets

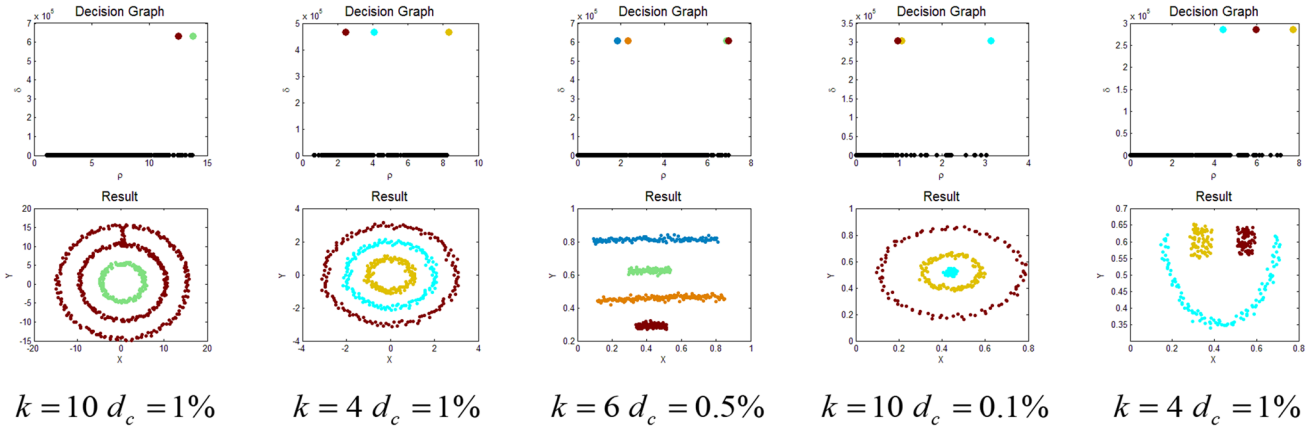


Fig. 12 DPC-GD on the remaining few data sets

Fig. 13 COIL-20

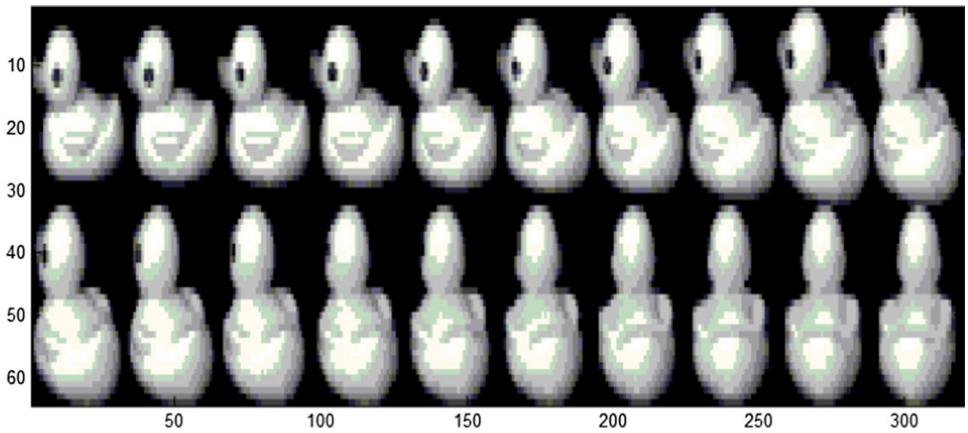
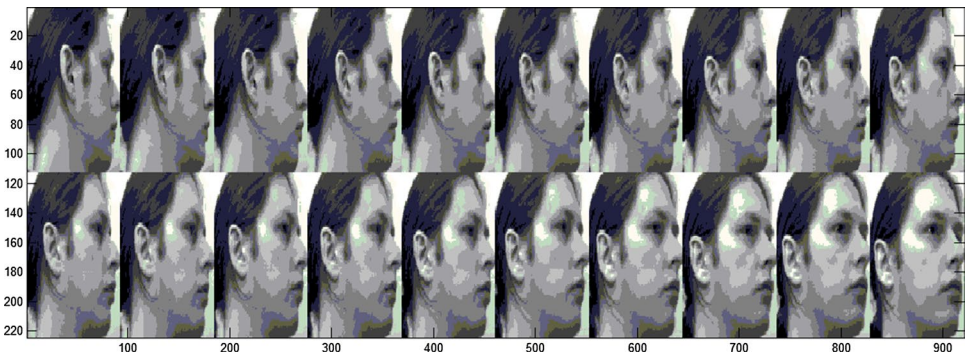


Fig. 14 The UMIST face database



from the USPS test database. We only use the images from five digits {0 2 4 6 7} on the data set. Thus, it contains 1024 images, and each image is represented by a 256-dimensional vector. Figure 15 shows some images of digits on this data set.

The details of these data sets are given in Table 1.

4.2.2 *Quality of the clustering results*

We measured the quality via Clustering Accuracy (ACC) [37–39] and the Normalized Mutual Information (NMI) [40, 41] between the produced clusters and the ground truth categories.

First, we introduce the Clustering Accuracy. For N distinct samples $x_i \in R^j$, y_i and c_i are the inherent category

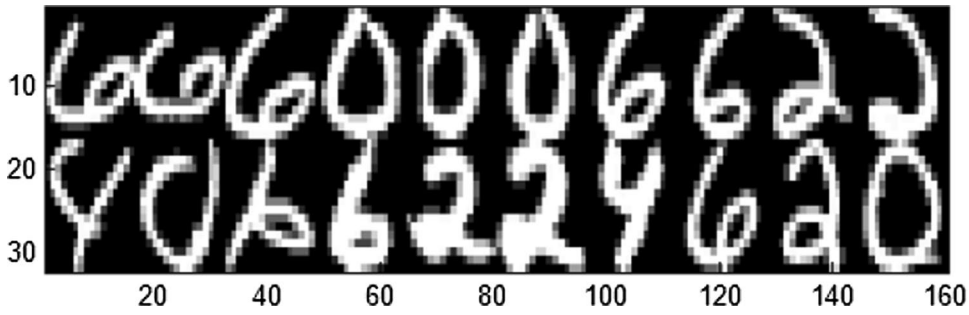


Fig. 15 The USPS handwritten digit database

Table 1 The details of these image data sets

Data sets	Cluster	Dimension	N
COIL-20	20	1024	1440
UMIST	10	10,304	265
USPS	5	256	1024

label and the predicted cluster label of $\mathbf{x_i}$, respectively, the calculation formula of ACC is

$$ACC = \frac{\sum_{i=1}^{N} \delta(y_i, map(c_i))}{N} \tag{9}$$

where $map(\cdot)$ maps each cluster label to a category label by the Hungarian algorithm [42] and this mapping is optimal, let $\delta(y_i, c_i)$ equal to 1 if $y_i = c_i$ or equals to 0 otherwise. The higher the values of the ACC are, the better the clustering performance will be.

Finally, we introduce the Normalized Mutual Information. The normalized mutual information between two random variables CAT (category label) and CLS (cluster label) is defined as

$$NMI(CAT;CLS) = \frac{I(CAT;CLS)}{\sqrt{H(CAT)H(CLS)}} \tag{10}$$

where $I(CAT;CLS)$ is the mutual information between CAT and CLS. The entropies $H(CAT)$ and $H(CLS)$ are used to normalize the mutual information in the range of [0, 1]. In practice, we made use of the following formulation to estimate the NMI score [43]:

$$NMI = \frac{\sum_{i=1}^{K}\sum_{i=1}^{K} N_{i,j} \log\left(\frac{N \cdot N_{i,j}}{N_i \cdot N_j}\right)}{\sqrt{\left(\sum_i N_i \log\frac{N_i}{N}\right)\left(\sum_j N_j \log\frac{N_j}{N}\right)}} \tag{11}$$

where N is the number of images, N_i and N_j denote the number of images in category i and cluster j, respectively, and $N_{i,j}$ denotes the number of images in category $i$as well as in cluster j. The NMI score is 1 if the clustering results perfectly match the category labels, and the score is close to 0 if data is randomly partitioned. The higher the NMI score, the better the clustering quality.

The comparison of these algorithms is shown in Table 2. In Table 2, the symbol—means that the algorithm cannot work on the data set. On these image data sets, DPC does a poor job of finding their clusters. In the case that DPC deals with data in the USPS set, only one cluster center is found by DPC with different d_c on decision graph, as shown in Fig. 16. It is unacceptable that we are incapable of making the right choices. DPC-GD has a favorable performance comparing to the original algorithm, as shown in Fig. 17.

Firstly, we discuss the problem of clustering from COIL-20. Each object has 72 images captured under varying angles. Because the space of images under varying angles

Table 2 The performance comparison of the proposed algorithm

	Quality	DPC	DPC-GD	SC	kernel k-means
COIL-20	ACC	–	1.0	0.6676 ± 0.0125	0.6320 ± 0.0509
	NMI	–	1.0	0.7492 ± 0.0084	0.7485 ± 0.0226
	Parameter		$k = 2$ $d_c = 6\%$	c = 20	c = 20
UMIST	ACC	–	0.8830	0.3970 ± 0.0218	0.3491 ± 0.0247
	NMI	–	0.8739	0.4362 ± 0.0252	0.3900 ± 0.0352
	Parameter		$k = 3$ $d_c = 0.2\%$	c = 10	c = 10
USPS	ACC	–	0.9097	0.5778 ± 0	0.5762 ± 0.0595
	NMI	–	0.7724	0.4273 ± 0	0.5288 ± 0.0366
	Parameter		$k = 10$ $d_c = 0.5\%$	c = 5	c = 5

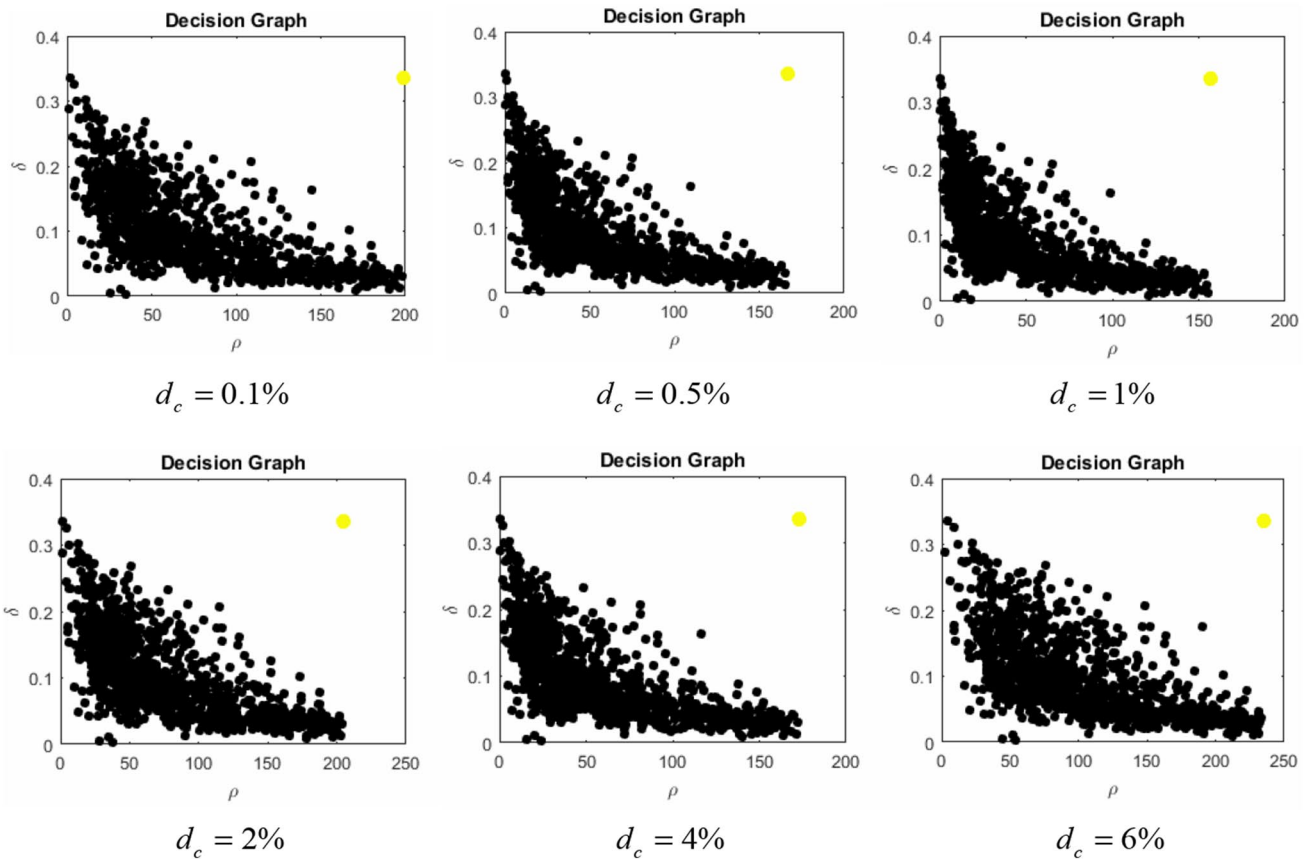


Fig. 16 DPC on USPS set with different values of d_c

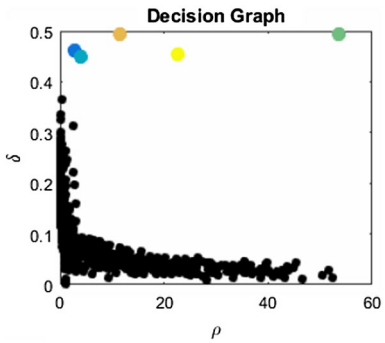


Fig. 17 DPC-GD on USPS set

is relatively densely sampled and the two images are very close to each other as shown in Fig. 13, there are several points whose closest neighbor comes from the other point. The experimental results which demonstrate DPC-GD produces very good performance on this data set are shown in Table 2.

Then, we consider the face images in the UMIST face database. The number of pictures per person is from around 19–38 and the space of face images covers varying poses. It has a non-uniform sampling and a higher dimension compared to COIL-20. The proposed algorithm also obtains satisfactory results on this database.

Finally, we explain the problem of clustering from the USPS handwritten digit database. This data set has a higher sampling density comparing to the first two, but the geometry of the distribution of each cluster may not be obvious. Experimental results show that our algorithm is effective.

In consequence, the proposed algorithm outperforms the original algorithm on some image data sets of different sampling density and different distribution.

5 Conclusions

In order to reveal the geometry of the data, the combination of the geodesic distances with density peaks clustering yields a novel algorithm, DPC-GD. The pairwise distances are recalculated by the geodesic distances. In this way, distances of points from the same cluster of arbitrary shape are shrunk.

In this paper, we test our algorithm on synthetic data sets which makes things easy from the visualization point of view. The presented algorithm shows the power on some data sets. Meanwhile, in order to assess the performance of the proposed algorithm, we compare the proposed algorithm with classical methods (kernel k-means algorithm and spectral clustering algorithm) and the original algorithm on image data sets which contain complex natural observations. Our algorithm has achieved satisfactory results on most data sets.

Future work should also be devoted to improvements of the time (and the space) complexities of the proposed algorithm. For example, we can apply Dijkstra's algorithm or other method to the geodesic distances. In addition, we will develop an automatic cluster centroid selection method on the basis of the proposed algorithm.

Acknowledgements This work is supported by the National Natural Science Foundation of China (Nos. 61379101 and 61672522), the National Key Basic Research Program of China (No. 2013CB329502). The Priority Academic Program Development of Jiangsu Higer Education Institutions (PAPD), and the Jiangsu Collaborative Innovation Center on Atmospheric Environment and Equipment Technology (CICAEET).

References

1. Wen X, Shao L, Xue Y et al (2015) A rapid learning algorithm for vehicle classification. Inf Sci 295:395–406
2. Iam-On N, Boongoen T, Kongkotchawan N (2014) A new link-based method to ensemble clustering and cancer microarray data analysis. Int J Collab Intell 1(1):45–67
3. Jia H, Ding S, Du M et al (2016) Approximate normalized cuts without Eigen-decomposition. Inf Sci 374:135–150
4. Zheng Y, Jeon B, Xu D et al (2015) Image segmentation by generalized hierarchical fuzzy C-means algorithm. J Intell Fuzzy Syst 28(2):961–973
5. Han J, Kamber M (2000) Data mining: concepts and techniques. Morgan Kaufman, San Francisco
6. Zhang Y, Sun X, Wang B (2016) Efficient algorithm for k-barrier coverage based on integer linear programming. China Commun 13(7):16–23
7. Li X, Liang Y, Cai Y (2016) CC-K-means: a candidate centres-based K-means algorithm for text data. Int J Collab Intell 1(3):189–204
8. Dong CR, Ng WWY, Wang XZ et al (2014) An improved differential evolution and its application to determining feature weights in similarity-based clustering. Neurocomputing 146:95–103
9. Xu L, Ding S, Xu X et al (2016) Self-adaptive extreme learning machine optimized by rough set theory and affinity propagation clustering. Cognit Comput 8(4):720–728
10. Rodriguez A, Laio A (2014) Clustering by fast search and find of density peaks. Sci 344(6191):1492–1496
11. Chen GJ, Zhang XY, Wang ZJ et al (2015) Robust support vector data description for outlier detection with noise or uncertain data. Knowl-Based Syst 90:129–137
12. Lu KY, Xia SY, Xia C (2015) Clustering based road detection method. In: Proceedings of the 34th Chinese Control Conference (CCC). pp 3874–3879
13. Xie K, Wu J, Yang W, Sun CY (2015) K-means clustering based on density for scene image classification. In: Proceedings of the 2015 Chinese Intelligent Automation Conference. pp 379–386
14. Du M, Ding S, Xue Y (2017) A robust density peaks clustering algorithm using fuzzy neighborhood. Int J Mach Learn Cybern. doi:10.1007/s13042-017-0636-1
15. Zhang Y, Xia Y, Liu Y et al (2015) Clustering sentences with density peaks for multi-document summarization. In: Proceedings of human language technologies: the 2015 annual conference of the north american chapter of the ACL. pp 1262–1267
16. Tang GH, Jia S, Li J (2015) An enhanced density peak-based clustering approach for hyperspectral band selection. In: Proceedings of the international geoscience and remote sensing symposium. pp 1116–1119
17. Zhang WK, Li J (2015) Extended fast search clustering algorithm: widely density clusters, no density peaks. arXiv preprint arXiv:1505.05610. doi:10.5121/csit.2015.50701
18. Wang XF, Xu YF (2015) Fast clustering using adaptive density peak detection. Stat Methods Med Res. doi:10.1177/0962280215609948
19. Du M, Ding S, Jia H (2016) Study on density peaks clustering based on k-nearest neighbors and principal component analysis. Knowl-Based Syst 99:135–145
20. Tenenbaum JB, Silva VD, Langford JC (2000) A global geometric framework for nonlinear dimensionality reduction. Sci 290(5500):2319–2323
21. Roweis ST, Saul LK (2000) Nonlinear dimensionality reduction by locally linear embedding. Sci 290(5500):2323–2326
22. Liu Z, Wang W, Jin Q et al (2016) Manifold alignment using discrete surface Ricci flow. CAAI Trans Intell Technol 1(3):285–292
23. Belkin M, Niyogi P (2003) Laplacian eigenmaps for dimensionality reduction and data representation. Neural Comput 15(6):1373–1396
24. Sampat MP, Wang Z, Gupta S et al (2009) Complex wavelet structural similarity: a new image similarity index. IEEE Trans Image Process 18(11):2385–2401
25. Belkin M, Niyogi P, Sindhwani V (2006) Manifold regularization: a geometric framework for learning from labeled and unlabeled examples. J Mach Learn Res 7:2399–2434
26. Ding SF, Hua XP (2014) Recursive least squares projection twin support vector machines for nonlinear classification. Neurocomputing 130:3–9
27. Xu X, Law R, Chen W et al (2016) Forecasting tourism demand by extracting fuzzy Takagi–Sugeno rules from trained SVMs. CAAI Trans Intell Technol 1(1):30–42
28. Chen WJ, Shao YH, Hong N (2014) Laplacian smooth twin support vector machine for semi-supervised classification. Int J Mach Learn Cybern 5(3):459–468
29. Ng AY, Jordan MI, Weiss Y (2002) On spectral clustering: analysis and an algorithm. Proc Adv Neural Inf Process Syst 2:849–856
30. Wang L, Bo LF, Jiao LC (2007) Density-sensitive spectral clustering. Acta Electron Sin 35(8):1577–1581
31. Zelnik-Manor L, Perona P (2004) Self-tuning spectral clustering. Proc Adv Neural Inf Process Syst 1601–1608
32. Nene SA, Nayar SK, Murase H (1996) Columbia object image library (COIL-20). Technical Report CUCS-005-96. Columbia University, USA
33. Graham DB, Allinson NM (1998) Characterising virtual eigensignatures for general purpose face recognition. Face recognition. Springer, Berlin Heidelberg, pp 446–456
34. Friedman J, Hastie T, Tibshirani R (2001) The elements of statistical learning. Springer, Berlin

35. Ma Z, Liu Q, Sun K et al (2016) A syncretic representation for image classification and face recognition. CAAI Trans Intell Technol 1(2):173–178
36. Zeng S, Yang X, Gou J et al (2016) Integrating absolute distances in collaborative representation for robust image classification. CAAI Trans Intell Technol 1(2):189–196
37. Xia Z, Wang X, Sun X et al (2016) Steganalysis of LSB matching using differences between nonadjacent pixels. Multimed Tools Appl 75(4):1947–1962
38. Gu B, Sheng VS, Wang Z et al (2015) Incremental learning for ν-support vector regression. Neural Netw 67:140–150
39 Jia HJ, Ding SF, Meng LH et al (2014) A density-adaptive affinity propagation clustering algorithm based on spectral dimension reduction. Neural Comput Applic 25(7–8):1557–1567
40 Wang XZ, He YL, Wang DD (2014) Non-naive bayesian classifiers for classification problems with continuous attributes. IEEE Trans Cybern 44(1):21–39
41 Chen WY, Song YQ, Bai HJ et al (2011) Parallel spectral clustering in distributed systems. IEEE Trans Pattern Anal Mach Intell 33(3):568–586
42 Papadimitriou CH, Steiglitz K (1998) Combinatorial optimization: algorithms and complexity. Courier Dover Publications, Mineola
43 Strehl A, Ghosh J (2003) Cluster ensembles- knowledge reuse framework for combining multiple partitions. J Mach Learn Res 3:583–617