

Natural neighbor-based clustering algorithm with local representatives



Dongdong Cheng, Qingsheng Zhu*, Jinlong Huang, Lijun Yang, Quanwang Wu

Chongqing Key Lab. of Software Theory and Technology, College of Computer Science, Chongqing University, Chongqing 400044, China

ARTICLE INFO

Article history:

Received 18 November 2016

Revised 21 February 2017

Accepted 22 February 2017

Available online 24 February 2017

Keywords:

Clustering

Natural neighbor

Local representatives

ABSTRACT

Clustering by identifying cluster centers is important for detecting patterns in a data set. However, many center-based clustering algorithms cannot process data sets containing non-spherical clusters. In this paper, we propose a novel clustering algorithm called NaNLORE based on natural neighbor and local representatives. Natural neighbor is a new neighbor concept and introduced to compute local density and find local representatives which are points with local maximum density. We first find local representatives and then select cluster centers from the local representatives. The density-adaptive distance is introduced to measure the distance between local representatives, which helps to solve the problem of clustering data sets with complex manifold structure. Cluster centers are characterized by higher density than their neighbors and a relatively large density-adaptive distance from any local representatives with higher density. In experiments, we compare the proposed algorithm NaNLORE with existing algorithms on synthetic and real data sets. Results show that NaNLORE performs better than existing algorithm, especially on clustering non-spherical data and manifold data.

© 2017 Elsevier B.V. All rights reserved.

1. Introduction

With the development of information technology, various industries have accumulated large amounts of data. How to mine useful information from these data is very challenging. Data mining is used to solve the problems.

Clustering analysis is an important research subject of data mining. It aims to divide a database into groups based on the similarity among these objects. It helps to recognize the intrinsic structure of data by arranging the objects in a way that objects in the same group have high similarity while objects belonging to different groups have low similarity. Many different clustering algorithms have been proposed, such as partitioning methods [1,2], hierarchical methods [3,4], fuzzy clustering methods [5,6]. The method proposed in [7] uses the meta-knowledge to solve algorithm selection problem. Among them, partitioning methods and hierarchical methods are the most primary methods.

Partitioning methods, such as K-means [1] and K-centers [2], can effectively cluster data with Gaussian-like distributions. However, these methods are sensitive to the selection of initial cluster centers. In order to get good results, they have to be launched many times with different initializations. Since a point is always

assigned to the nearest center, these methods are not applicable to non-spherical clusters.

Hierarchical clustering seeks to build a hierarchy of clusters. Single-Link [4], Complete-Link [3], BIRCH [8], CURE [9], Rock [10] and Chameleon [11] are representative algorithms. Chameleon is a hierarchical method which integrates bottom-up and top-down strategies. It can find more natural clusters of various shapes since it measures the similarity of two clusters dynamically considering data distribution within a cluster. But it suffers the problem of parameter selection.

Recently, some novel center-based clustering algorithms are proposed. In 2007, Frey and Dueck proposed a new clustering algorithm AP [12] by passing messages between data points. In 2014, Rodriguez and Laio proposed a novel center-based clustering algorithm [13] (denoted as DP for convenience in this paper) which is on the assumption that cluster centers are surrounded by neighbors with lower local density and that they are at a relatively large distance from any points with a higher local density. DPC-KNN-PCA [14] introduces k nearest neighbor and PCA into DP algorithm to process high dimensional data sets. Nevertheless, these center-based algorithms still have difficulty in clustering data sets containing manifold clusters. A density-adaptive affinity propagation clustering algorithm based on spectral dimension reduction (DAAP) [15] which is an improved algorithm of AP, is proposed to deal with data sets with complex manifold structure, but DAAP algorithm has high time complexity.

* Corresponding author.

E-mail address: qszyu@cqu.edu.cn (Q. Zhu).

In order to solve the problems of the above algorithms, we propose a new clustering algorithm based on natural neighbor and local representatives (NaNLORE) in this paper. It does not need to set parameters and it only needs to select cluster centers according to the new constructed decision graph like DP algorithm. At first, we define a new density metric based on natural neighbor to evaluate the local density of each object. Then NaNLORE finds the local representatives which are with local maximum density. After that, we compute density-adaptive distances between local representatives which help to cluster manifold data sets. Subsequently, according to the new defined density and distance, we use DP algorithm to cluster local representatives. Finally, each remaining object is assigned to the cluster its representative belonging to. Since we only need to compute the density-adaptive distances between local representatives rather than all objects in a data set, NaNLORE has lower computational complexity than DAAP. In the experiment, we compare our algorithm NaNLORE with two primary algorithms (i.e., K-means and Chameleon), three center-based algorithms (i.e., AP, DP and DPC-KNN-PCA), and a spectral-based algorithm (i.e., DAAP) on synthetic data sets and real data sets. The experimental results show that NaNLORE is more effective than other algorithms.

The rest of this paper is organized as follows. In Section 2, we introduce the related work. Section 3 reveals the proposed clustering algorithm NaNLORE. Experimental results on synthetic data sets and real data sets are reported in Section 4, followed by conclusions and future work in Section 5.

2. Related work

2.1. Center-based clustering

For many classical algorithms, a key step is to find cluster centers. For example, K-means [1] and K-centers [2] obtain clustering results by optimize an objective function, typically the sum of the distance to a set of putative cluster centers. However, the initialization of centers is a difficult task. The main purpose of AP algorithm [12] is also to find the optimal representative points (cluster centers). Different from k-centers, AP algorithm does not need to specify the initial cluster centers in advance and regards all points as potential cluster centers, avoiding the selection of the initial cluster centers. However, AP cannot obtain satisfactory results since the number of clusters is affected by user-defined preference parameter. Zhang *et al.* proposed K-AP algorithm [16] to solve the problem by introducing a constraint in the process of message passing. However, like K-means and K-centers, AP and K-AP algorithms both suffer from the problem of discovering non-spherical clusters. Sparse Subspace Clustering(SSC) [17] using sparse representation techniques to cluster multisubspace data. DP algorithm [13] uses decision graph to map the data into a two-dimension graph w.r.t the local density ρ_i and the distance δ_i . Cluster centers stand out with anomalously large value of ρ_i and δ_i . There are also some improved algorithms [14,18–20] of DP algorithm. However, these center-based algorithms still have difficulty in clustering data sets containing manifold clusters.

More algorithms have been proposed to solve the problem of clustering manifold data sets. The spectral clustering algorithms [21–23] utilize the spectrum of the similarity matrix to map the data into a lower-dimensional space in which the objects can be easily clustered by traditional algorithms. These algorithms perform well when discovering non-spherical clusters. Yet, they can not successfully cluster data sets containing clusters with multiple manifold structures and scales. The work in [24] presents a graph-based k-means algorithm called GKM, which fully exploits the underlying manifold structure of a data set to produce appropriate clustering results, meanwhile, identifies a suitable representative for each subset by the similarity between points on nonlin-

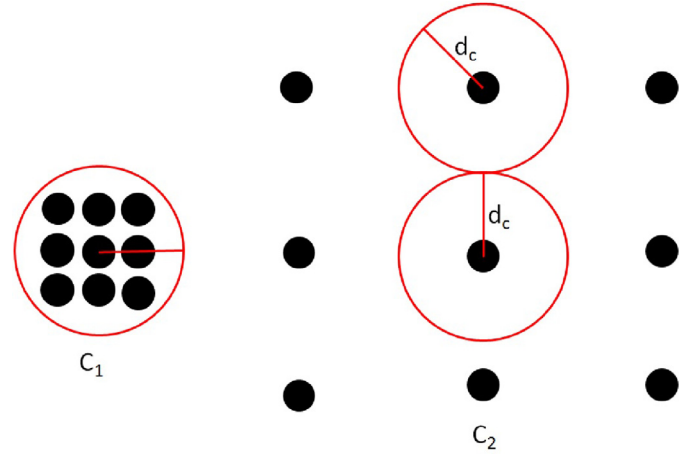


Fig. 1. Illustration of great density variations, the value of cutoff distance is hard to set.

ear manifold. However, similar to the k-means, GKM also suffers from the initial conditions, which has a significant influence on the final results. The basic idea of DAAP [15] is: firstly, map the original data to a low-dimension R^k space according to the spectral graph theory; subsequently, construct similarity measurement; according to K-AP clustering method, every data point competes for cluster center or selects other point as its center through message passing between each other; and finally obtain K cluster centers. A density-adaptive similarity measure is introduced into DAAP to describe the reactions between objects, helping cluster data sets with manifold structure. However, it has to compute the shortest distance between two points, and the time complexity is $O(N^3)$ (N is the number of objects in a data set), making it hard to apply to large scale data sets.

2.2. Decision graph

DP clustering algorithm proposes decision graph which plots the distance δ_i between point i and its nearest neighbor with a higher density as a function of the point density ρ_i . In decision graph, δ_i is much larger than typical nearest neighbor distance only for points that are local or global maxima in the density. Cluster centers are recognized as points for which the values of δ_i and ρ_i are anomalously large. For each data point i , the local density ρ_i is defined as

$$\rho_i = \sum_j \chi(d_{ij} - d_c) \quad (1)$$

where $\chi(x) = 1$ if $x < 0$ and $\chi(x) = 0$ otherwise, and d_c is a cutoff distance. In fact, ρ_i is the number of points that are closer than d_c to point i . The distance δ_i is measured by computing the minimum distance between the point i and any other point with higher density:

$$\delta_i = \min_{j: \rho_j > \rho_i} (d_{ij}) \quad (2)$$

For the point with highest density, $\delta_i = \max_j (d_{ij})$.

However, when the density of two clusters is significantly different, it will be hard to set an appropriate cutoff distance, which will influence clustering results, just as shown in Fig. 1, if the value of d_c is set inappropriately, points in Cluster C_2 are all with local maximum density. It will be hard to select Cluster C_2 's center.

2.3. Density-adaptive distance measurement

A good similarity measure should satisfy the following clustering hypothesis: (a) adjacent points should have high similarity;

(b) if points located at the same architecture (e.g. same cluster or manifold), they should have high similarity as well [25]. But the Euclidean distance only takes local information of data into account and is unable to describe the global architecture of the data set very well. In [15,22], a new distance measurement, density-adaptive distance, is proposed to depict the distribution characteristic of clusters. This measurement does not guarantee that the distance between two directly connected points is shortest. In other words, if two objects are in the same cluster, there will exist a continuous connecting curve only through high-density region; otherwise, every curve like this must cross over a whole low-density valley. First of all, the authors define a density-sensitive line length:

$$L(v_i, v_j) = (e^{\rho \|v_i - v_j\|} - 1)^{1/\rho} \quad (3)$$

where $\|v_i - v_j\|$ is Euclidean distance between two points v_i and v_j . $\rho > 1$ is density factor. This method compresses the high-density regions distance, and meanwhile magnifies the low-density regions distance. It can be adapted to convex and non-convex data sets, since it does not satisfy triangle inequality.

Let $P = \{p_1, p_2, \dots, p_l\} \in V$ present the path from p_1 to p_l , and the length $l = |P|$, $(p_k, p_{k+1}) \in E$, $1 \leq k < l$. Make P_{ij} indicate the collection of paths connecting data points v_i and v_j , $1 \leq i, j \leq n$. The density-adaptive distance measurement method is defined as:

$$D_{ij} = \min_{k=1}^{l-1} \sum_{k=1}^{l-1} L(p_k, p_{k+1}), p \in P_{ij} \quad (4)$$

where $L(p_k, p_{k+1})$ is the density-sensitive line length along the path p between two adjacent points. This measurement is data-correlated and will automatically adjust the size along with local density. As pointed out in [15], the distance measurement is insensitive to the value of ρ , we set $\rho = 2$ in this paper.

3. The proposed algorithm

The social structure is becoming more and more complex with the growth of population. Consequently, it is more and more difficult that all of the social members collaborate to manage social affairs. A good idea is to select representatives, and the representatives have more power than the others. The power comes from their neighbors and in turn, acts on their neighbors. Analogously, when analyzing data, objects with local maximum density can be selected as representatives, and cluster centers are chosen from representatives. Inspired by this idea, we present a novel clustering algorithm based on natural neighbor and local representative (NaNLORE), into which natural neighbor is introduced to compute points density and find local representatives. Since we introduce the density-adaptive distance, NaNLORE can effectively process data sets with multiple manifold structures and scales.

3.1. Natural neighbor

Jarvis–Patrick algorithm [26] presents shared near neighbor which is used to measure the similarity between objects. Natural Neighbor (NaN) [27–29] is a new concept of neighbor. The concept originates from the reality that the number of ones real friends should be the number of how many people are taken him or her as friends and he or she takes them as friends at the same time. It is conventional that objects lying in sparse region should possess few neighbors, whereas objects lying in dense region should possess great neighbors.

The key idea of natural neighbor is that we continuously expand neighbor searching range, and every time compute the number of each object is considered as other object's neighbor, until all objects are considered as neighbor or the number of objects without being other objects' neighbors does not change. The

Algorithm 1: NaN-Searching.

Input: D : the data set

Output: sup_k, nb, NN_r

Initializing: $r=1, nb(i)=0, NN_0(i) = \phi, RNN_0(i) = \phi$;

while true do

for each data point x in D do

 Use kdtree to find the r -th neighbor y of x ;

$nb(y) = nb(y) + 1$;

$NN_r(x) = NN_{r-1}(x) \cup \{y\}$;

$RNN_r(y) = RNN_{r-1}(y) \cup \{x\}$;

end

 Compute the number of points with no neighbor (i.e., $nb(x)=0$) $Numb$;

if the number $Numb$ does not change then

 Break;

end

$r=r+1$;

end

$sup_k = r$;

Output the $sup_k, nb, NN_r(i)$;

whole computation procedure of natural neighbor can be automatically fulfilled without any parameters. If the formation of K -nearest neighbor is regarded as an active neighbor searching procedure, then the forming of Natural Neighbor is completely passive.

The search cost of KNN and RNN for each object in the database is huge. So we introduce the KD-tree [30] into the Natural Neighbor searching algorithm. The Natural Neighbor searching algorithm is described in Algorithm 1.

In Algorithm 1, r is the searching range and $nb(y)$ represents the times that point y is contained by the neighborhood of other points. $NN_r(x)$ is the r -neighbors of x . $RNN_r(y)$ is the r -reverse neighbors of y ; sup_k is Natural characteristic value. Since KD-tree is introduced into NaN-Searching, the time complexity of NaN-searching algorithm is $O(N \log N)$ (N is the number of objects in a dataset).

Definition 1 (Natural Neighbor). Based on Natural Neighbor searching algorithm, if X belongs to the neighbors of Y and Y belongs to the neighbors of point X . Then X and Y are Natural Neighbor of each other.

Compared with the published concept of neighbor that has been widely used, k -nearest neighbor proposed in [31], the main difference is that Natural Neighbor is a scale-free concept of neighbor, the great advantage is that the search method of Natural Neighbor is non-parameter. Scale-free means that the number of neighbors for each object is not necessarily identical.

Definition 2 (Natural characteristic value sup_k). According to NaN-Searching algorithm, each point x has different number of neighbors $nb(x)$. There exists an average number of neighbors sup_k . The formula of computing sup_k is as follows.

$$sup_k = \min\{r | \forall x \exists y (y \neq x \cap x \in NN_r(y)) \text{ or } \forall x (|RNN_r(x)| = 0) = |RNN_{r-1}(x)| = 0\} \quad (5)$$

where x and y are points in the dataset. $NN_r(y)$ is the r th nearest neighbor of y . $RNN_r(y)$ is the r th reverse nearest neighbor of y . This formula means that with the increase of r , the minimum value of r satisfying one of the following conditions is natural characteristic value sup_k : (1) all points are considered as neighbor other points; (2) the number of points with no neighbor (i.e., $nb(x) = 0$) is the same in two successive iterations. Given a data set D containing

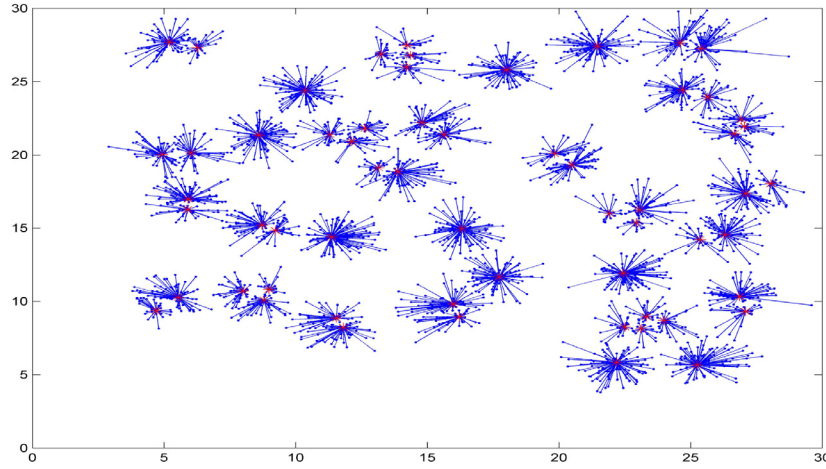


Fig. 2. The points are assigned to the its Representatives, and the red star points are Local Representatives found by LORE algorithm. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

n points, according to the graph theory, we know that $n \times \sup_k = \sum_{x \in D} nb(x)$, thus, \sup_k is the average number of neighbors.

Definition 3 (Natural Neighborhood Graph-NNG). The graph constructed by linking natural neighbors of each point is Natural Neighborhood Graph (NNG).

Definition 4 (Maximum Mutual Neighborhood Graph-MMNG). When X is one of the $\max\{nb\}$ nearest neighbors of Y , and Y is one of the $\max\{nb\}$ nearest neighbors of X , then X is Maximum Mutual neighbor of Y and Y is Maximum Mutual neighbor of X . The graph constructed by linking Maximum Mutual neighbors of each point is Maximum Mutual Neighborhood Graph (MMNG).

3.2. Natural neighbor-based density

The density of point in dense region is larger than that in sparse region, and distance between points in dense region is smaller than that in sparse region. The density of a point is inversely proportional to distance between itself and its neighbors. As for two different points, we find the same number of neighbors. The larger the sum of distance between the point and its neighbors is, the sparser region the point lies, contrarily, the smaller, the denser. The k nearest neighbor is used to compute points density in [20]. In this paper, natural neighbor-based density is defined in a similar way.

Definition 5 (The density of a given point i). the Natural Neighbor-based density of point i is defined as:

$$\rho_i = \frac{\sup_k}{\sum_{j \in N(i, \sup_k)} \text{dist}(i, j)} \quad (6)$$

where \sup_k is the Natural characteristic value. $N(i, \sup_k)$ is the \sup_k nearest neighbors of point i , and $\text{dist}(i, j)$ is the distance between point i and j . Different from the definition in [20], natural neighbor-based density does not need to set parameter k , and NaN-Searching algorithm is committed to find the appropriate parameter k .

3.3. Local representative

Definition 6 (Representative). If the density of q is the maximum in the neighbors of p , then the object q is the representative of p and its neighbors, which is denoted as $p \in q$.

According to the above definition, there exists a situation that a point belongs to two or more representatives at the same time. The representatives will compete for being representative of the point, and the representative competition rule is defined as follows.

Representative Competition Rule (RCR): For point p , $p \in R1$ and $p \in R2$, then $p \in \{X | \text{density}(X) = \max\{\text{density}(R1), \text{density}(R2)\}\}$. That is, comparing the density of $R1$ and $R2$, the larger one will be representative of point p .

If the representative of point p is q , and the representative of q is r , the representative of point p will be changed to r . We call it representative transfer rule, which is defined as follows.

Representative Transfer Rule (RTR): If $p \in q$, and $q \in r$, then $p \in r$.

Definition 7 (Local Representative). From the definition of Representative, RCR and RTR, we know that each point possesses one representative. The point whose representative is itself is called Local Representative.

Local Representatives are points with local maximum density. Each point belongs to one of local representatives. As shown in Fig. 2, the red star points are local representatives of the data set found by Local Representative searching algorithm (LORE), and each point is assigned to its representative. Obviously, the local representatives are points with local maximum density. Then the data set is divided into several clusters according to the local representatives. LORE algorithm is shown in Algorithm 2.

3.4. The density-adaptive distance between local representatives

The Euclidean distance is not a proper measure on manifold data. In [24], geodesic distance is used as a similarity measurement between two points. However, in clustering settings, the exact geodesic distance between two samples cannot be obtained directly, because we have no prior information about the underlying manifolds. As pointed in [32], if a data set has sufficient points from the manifold, the graph distance will be a good approximation of the geodesic distance. The graph distance between two vertices is defined as the shortest path of all paths connecting them. Density-adaptive distance [15,22] not only adapts to the data set on a manifold, but also satisfies the local and global coherence of the cluster hypothesis. Here we introduce density-adaptive distance to describe relationships between local representatives on a manifold.

Algorithm 2: LORE.

Input: NN: the nearest neighbors of each point, rho: the natural neighbor-based density of each point
Output: Re: the representative of each point, localRe: the local representatives, cl: the cluster label of each point

Initializing: $Re(i)=\phi$, $localRe=\phi$, $cl(i)=\phi$;
for each point i in the data set do
 Find the point i with the maximum density in $NN(i)$;
 for each point p in $NN(i)$ do
 if $Re(p) == \phi$ then
 $Re(p) = y$;
 end
 if $Re(p) == x$ and $x \neq y$ then
 Determine $Re(p)$ according to RCR;
 end
 for each point z in the data set do
 if $Re(z) == p$ then
 Determine $Re(z)$ according to RTR;
 end
 end
 end
end
 $K=1$;
for each point x in the data set do
 if $Re(x) == x$ then
 $localRe(K)=x$;
 $cl(x)=K$;
 $K=K+1$;
 end
end
for each point x in the data set do
 $cl(x)=cl(Re(x))$;
end

In order to ensure the local representatives have good connectivity, we construct MMNG on the original data set. Then we compute the density-sensitive line length between two adjacent points. Finally, we exploit Dijkstra algorithm to compute the shortest distance between local representatives, and the shortest distance is density-adaptive distance between local representatives. Similar to DP, δ_i is measured by computing the minimum distance between the local representative i and any other local representative with higher density:

$$\delta_i = \min_{j: \rho_j > \rho_i} (D_{ij}) \quad (7)$$

where D_{ij} is the density-adaptive distance between two local representatives i and j , which is computed as Formula (4).

3.5. Natural Neighbor-based clustering algorithm with local representatives (NaNLORE)

This paper presents a new clustering algorithm, natural neighbor-based clustering algorithm with local representatives (NaNLORE). Its basic idea is: firstly, find local representatives and compute the density-adaptive distance between local representatives; then construct decision graph on local representatives according to natural neighbor-based density and the adaptive distance; subsequently, select cluster centers from local representatives according to the decision graph, and local representatives are classified into several clusters; finally, each initial cluster is assigned to the corresponding cluster which its local representative belongs to. The NaNLORE algorithm is detailed in Algorithm 3.

Algorithm 3: NaNLORE.

Input: D: the data set
Output: cl: the final cluster label of each point
 $(sup_k, nb, NN)=NaN\text{-Searching}(D)$;
for each point x in D do
 Compute the density $\rho(x)$ according to Formula (6);
end
 $(re, localRe, cl)=LORE(NN, \rho)$;
Construct MMNG on the data set D ;
for each pair of representatives $R1$ and $R2$ do
 Compute the density-adaptive distance $dist(R1, R2)$
 between $R1$ and $R2$ on MMNG according to Formula (4);
end
for each local representative r do
 $ReCl(r)=\phi$;
end
 $ReCl=DP(\rho(localRe), dist)$;
for each point x in D do
 $cl(x)=ReCl(re(x))$;
end

NaNLORE contains three main components: NaN-Searching, the search of local representatives and clustering local representatives using DP algorithm. Since KD-tree is introduced into NaN-Searching, the time complexity of NaN-Searching algorithm is $O(N \log N)$. There are three steps in the search of local representatives, choosing representatives for each point from limited neighbors, finding local representatives and assigning each object to its local representative. The time complexity of them are $O(N)$, respectively. Consequently, the second step's time complexity is $O(N)$. Before using DP algorithm, we have to compute the shortest path between local representatives, which is the main cost of computation, because the time complexity of Dijkstra is $O(N^2)$. Assuming the number of local representatives is $N_L (N_L \ll N)$, so the time complexity of computing shortest path is $O(N_L \cdot N^2)$. Through heap optimization, the time complexity of Dijkstra can be reduced to $O((N + E) \cdot \log(N))$, and E is the edge number in the graph. MMNG is a sparse graph, and the number of its edges is less than $N \cdot \max_nb/2$ (\max_nb is a constant), then the time complexity of computing the shortest path is $O(N_L \cdot N \log N)$. Thus the overall time complexity of NaNLORE is $O(N_L \cdot N \log N)$.

4. Experimental analysis

4.1. Assessment of clustering performance

We evaluate the clustering performance using two criteria. The first one is clustering accuracy [33]. Usually, the serial number of every cluster will be resorted after the clustering results are obtained, e.g. the first cluster of original data set may be specified as the second cluster by an algorithm. Thus, we need to construct a displacement mapping function and match the serial number obtained by clustering and real cluster label, one by one. Cluster accuracy (ACC) is based on this mapping relation, and the calculation formula is as follows:

$$ACC = \frac{1}{n} \sum_{i=1}^n \delta(y_i, map(c_i)) \quad (8)$$

where y_i is the real cluster label, c_i is the serial number obtained by clustering, and $\delta(x, y) = \begin{cases} 1 & x = y \\ 0 & x \neq y \end{cases}$ is a discriminate function.

$ACC \in [0, 1]$, the larger the value of ACC is, the better the clustering performance of the algorithm means.

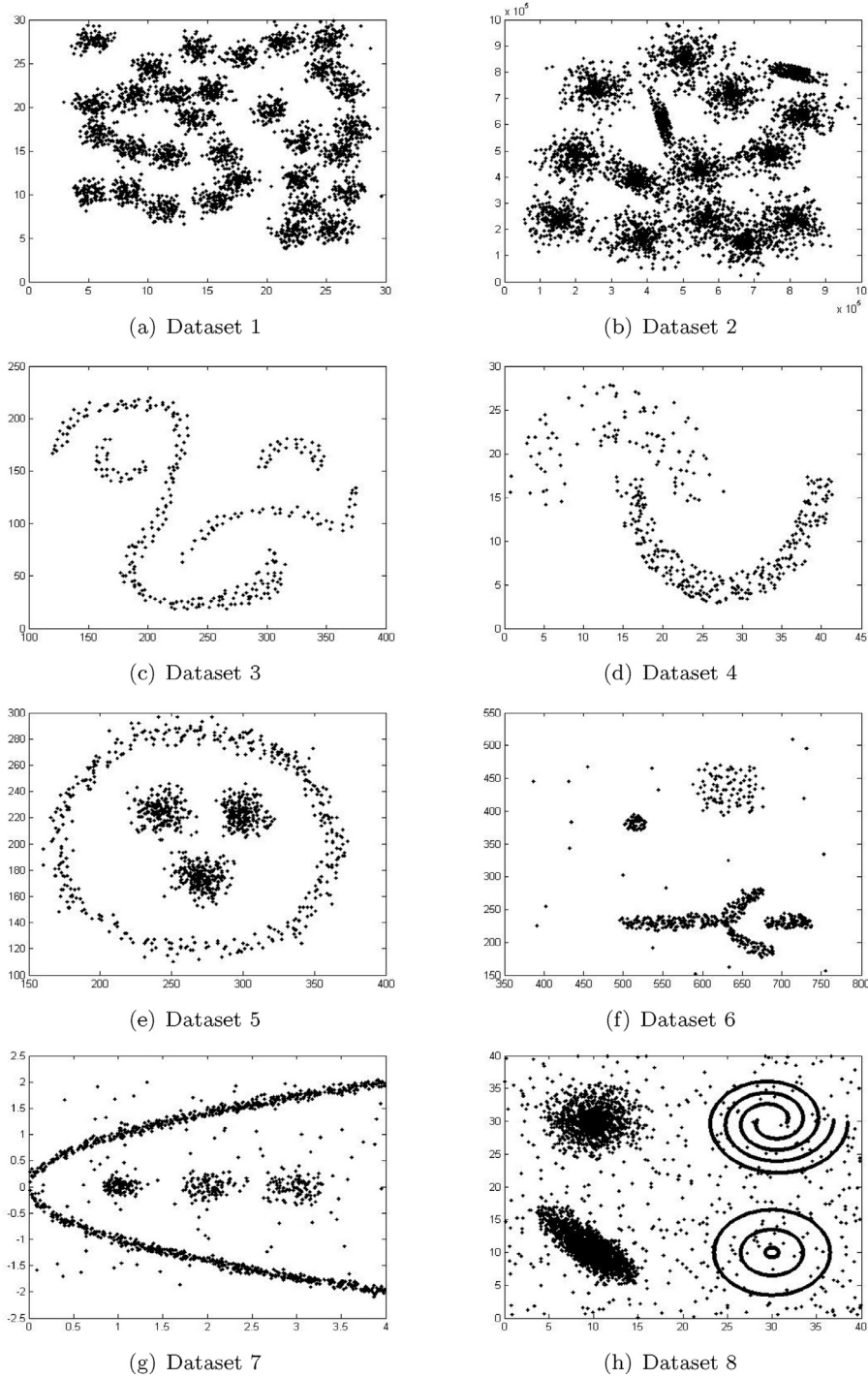


Fig. 3. Eight original synthetic datasets.

The second performance evaluation criterion is Normalized Mutual Information (NMI). The NMI is defined as

$$NMI(X, Y) = \frac{MI(X, Y)}{\sqrt{H(X)H(Y)}} \quad (9)$$

Where $MI(X, Y)$ is the mutual information between two random variables X and Y (X and Y are the ground-truth label and the clustering label, respectively.), and $H(*)$ is the random variable entropy, which is used to normalize the mutual information to be in the range of $[0,1]$. In practice, we make use of the following formula-

tion to estimate the NMI score.

$$NMI = \frac{\sum_{i=1}^k \sum_{j=1}^k n_{i,j} \log\left(\frac{n \cdot n_{i,j}}{n_i \cdot n_j}\right)}{\sqrt{(\sum_i n_i \log \frac{n_i}{n})(\sum_j n_j \log \frac{n_j}{n})}} \quad (10)$$

where n is the number of instances in a data set, n_i and n_j denote the number of instances in category i and cluster j , respectively, and $n_{i,j}$ denotes the number of instances in category i as well as in cluster j . NMI measures how good the clustering result is, with respect to the ground-truth information. $NMI=1$ means the clustering result is perfect and $NMI=0$ means the clustering result is

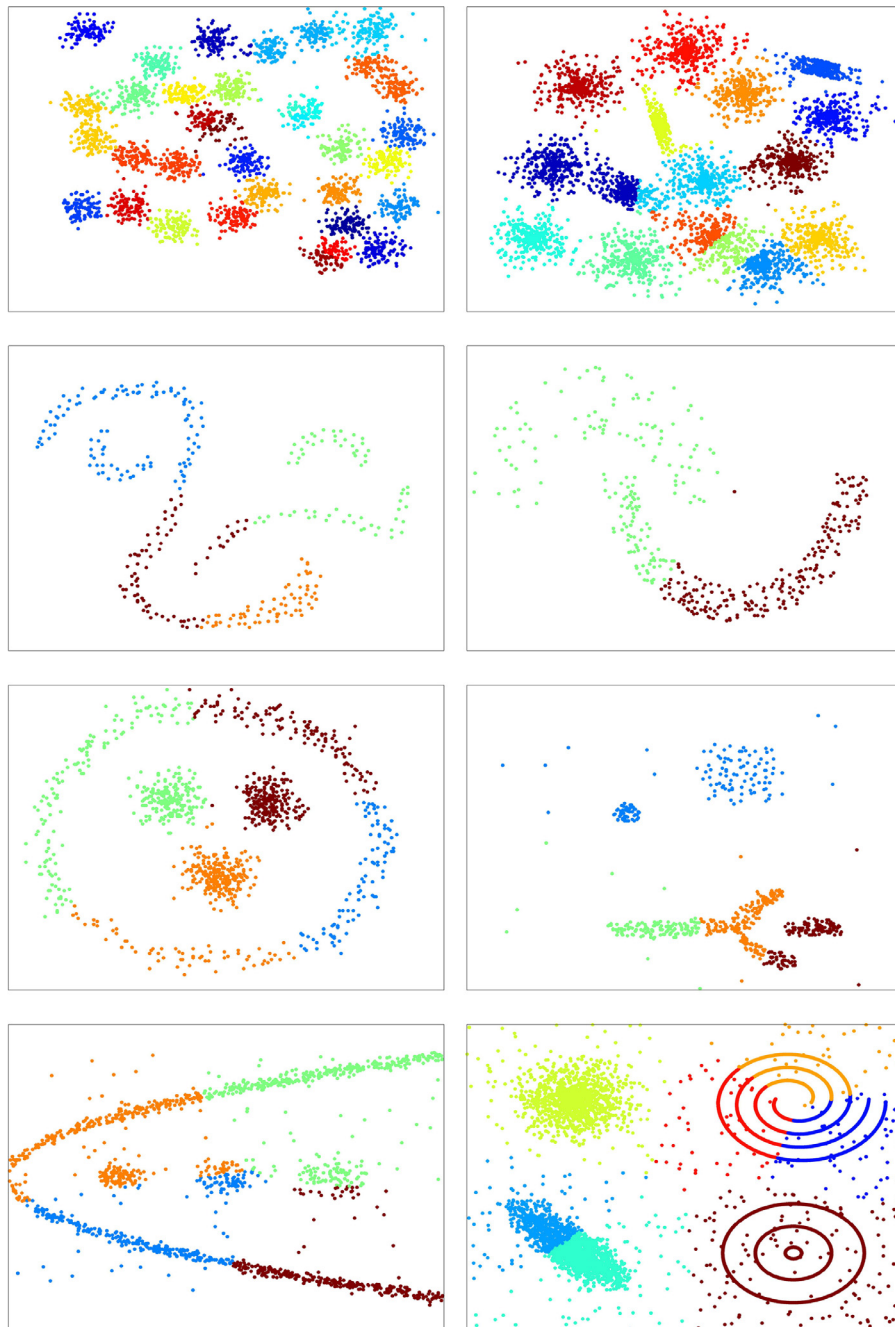


Fig. 4. Clustering results of K-means.

useless. Other value between 1 and 0 measures the quality of the clustering result.

4.2. Clustering on synthetic data sets

In order to demonstrate the effectiveness of NaNLORE, we compare the proposed algorithm NaNLORE with K-means, Chameleon, AP, DAAP, DP and DPC-KNN-PCA on eight challenging synthetic data sets, which are illustrated in Fig. 3. The first two data sets contain spherical clusters. Dataset 1, taken from [34], contains 3100 points and 31 spherical clusters. Dataset 2, taken from [35], a total of 5000 points, contains 15 clusters with overlap in data distribution. The rest data sets contain clusters with complex structure. Dataset 3 and Dataset 6 is from [36]. Dataset 3 is composed of 4 manifold clusters, a total of 315 points and Dataset 6 consists of

two spherical clusters, two manifold clusters and a few outliers, a total of 582 points. Dataset 4, taken from [37], is composed of two different density manifold clusters, a total of 373 points. Dataset 5 includes 3 spherical clusters in a circle cluster, a total of 1016 points. Dataset 7, taken from [38], consists of 3 spherical clusters, one manifold clusters and some noise points. Dataset 8 from [39], is composed of 7 clusters with multi shapes and some noises, a total of 8573 points. The clustering results of these algorithms are shown in Figs. 4–10. The comparison of these algorithms on ACC and NMI scores are shown in Table 1. The running time is shown in Table 2. The configuration of the computer used in our experiment is that processor is Intel Core i5 2.80GHZ; memory size is 4GB; programming environment is MATLAB R2013a.

For K-means algorithm, the parameter K is set as the desired cluster number and the initial cluster centers are selected ran-

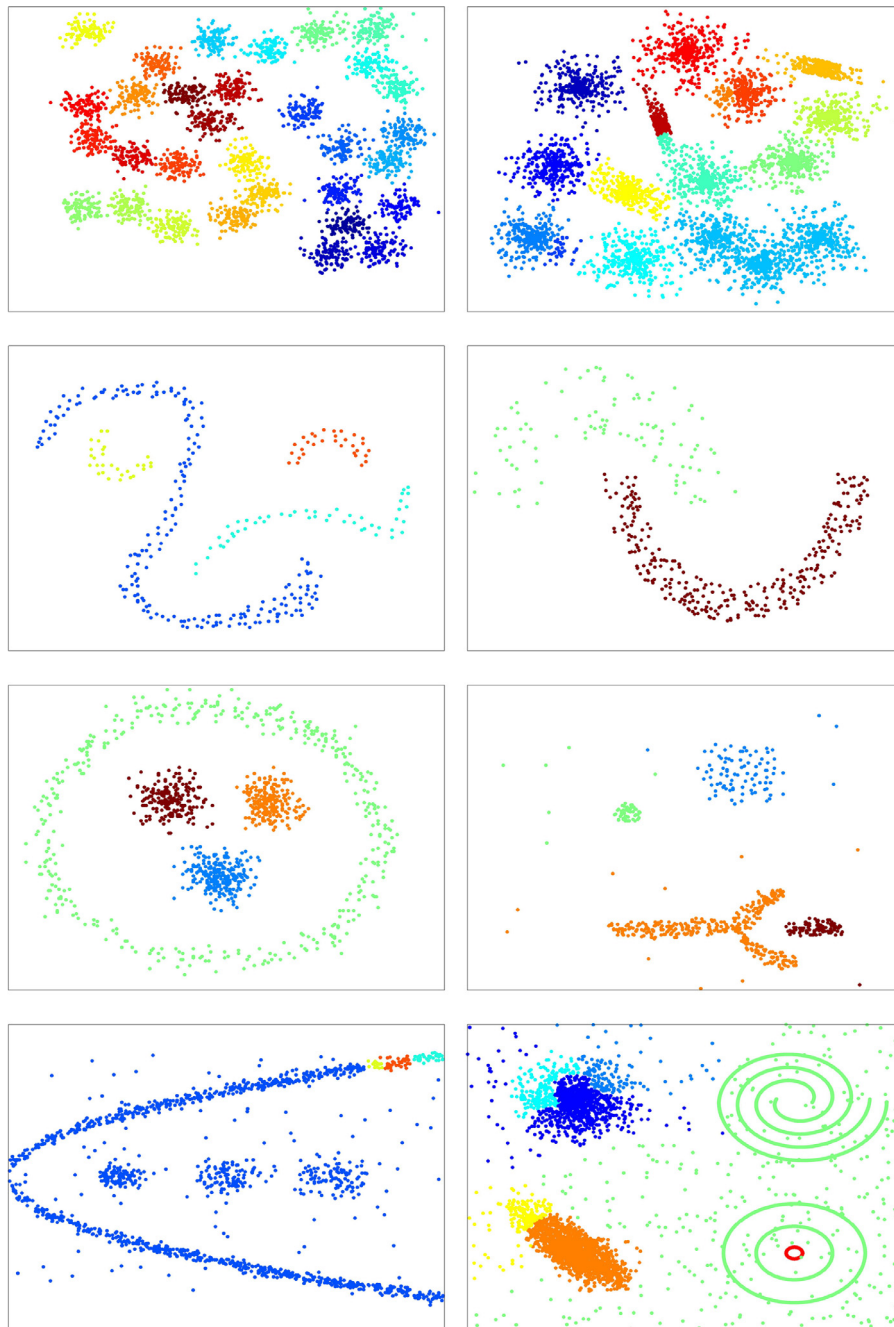


Fig. 5. Clustering results of Chameleon.

domly. The clustering result of K-means is shown in Fig. 4. From the results, we can see that K-means performs well on Dataset 1 and Dataset 2 which contain spherical clusters. However, since a point is always assigned to the nearest center, K-means can not deal with other data sets which contain clusters with multiple manifold structures and scales.

Chameleon is a hierarchical clustering algorithm. It uses two different schemes to implement merge process. The first merges only those pairs of clusters exceeding user-specified thresholds for relative interconnectivity and relative closeness. The second scheme uses the product of relative interconnectivity and relative closeness as the similarity between clusters and repeatedly merges the most similar clusters until the desired cluster number is reached. In order to avoid setting too many parameters, here we

select the second scheme to implement the merge process. The results of Chameleon is shown in Fig. 5 and we can learn that Chameleon can process data sets containing clusters with arbitrary shapes. However, Chameleon is susceptible to noise points. When there are many noise points, it cannot correctly discover the clusters, such as for Dataset 2, 7 and 8.

The results of AP algorithm are affected by the preference parameter p . In order to get the best results, we set different values for p , and the best clustering results of AP algorithm are shown in Fig. 6. The results show that AP algorithm is hard to get the right cluster number of the data sets and cannot deal with manifold data sets.

For DAAP algorithm, it reruns many times with different parameters values in an attempt to find the best clustering results.

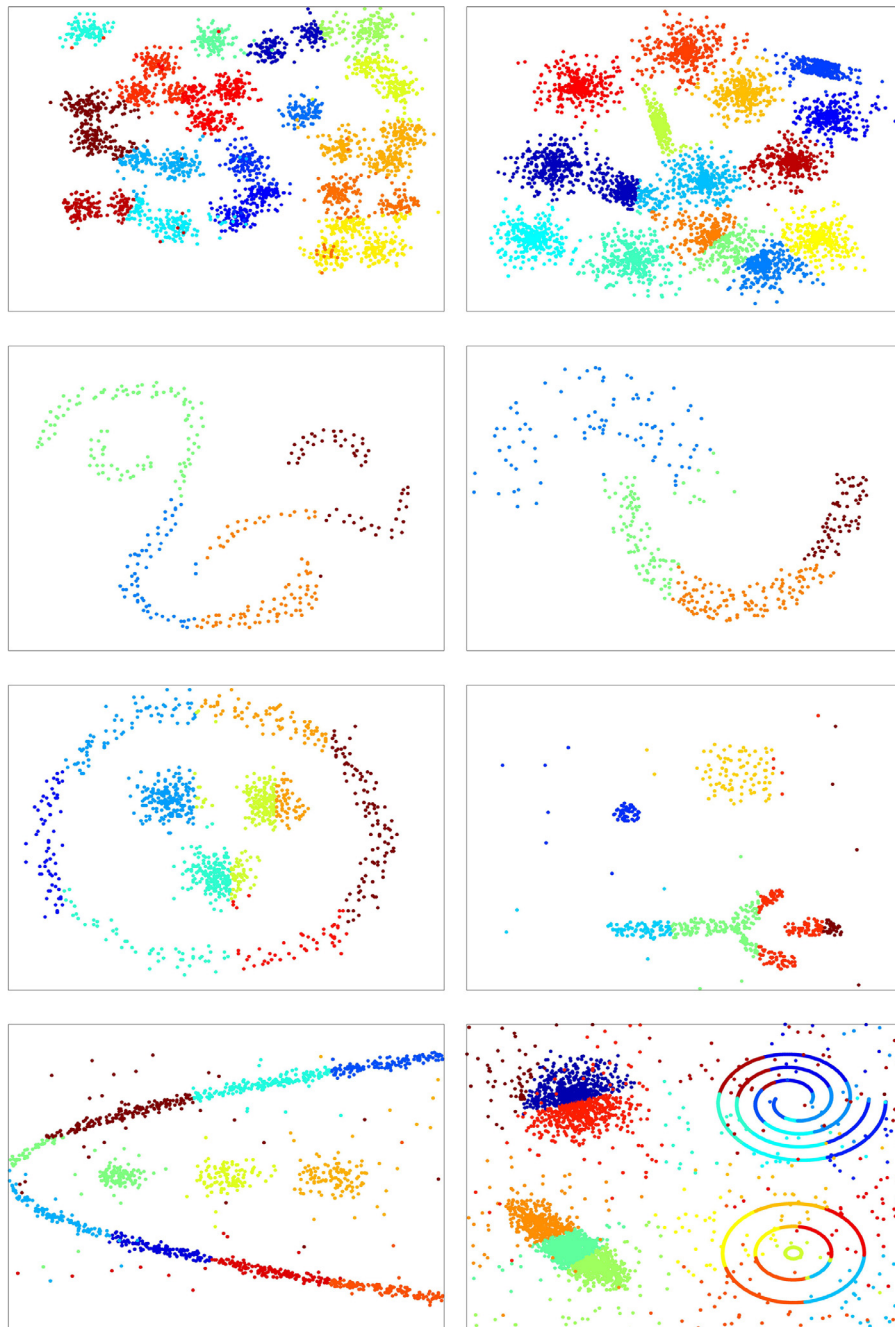


Fig. 6. Clustering results of AP. AP respectively detects 17, 16, 4, 4, 7, 6, 10, 17 clusters for the eight data sets.

The best results and the corresponding parameters are displayed in Fig. 7. DAAP introduces density-adaptive similarity measurement, making it adaptive to the manifold data sets, and uses K-AP to remedy the disadvantages of AP algorithm. However, when processing the last four data sets, DAAP still can not obtain the desired results. Besides, DAAP has high computational complexity and spends much more time processing big data sets than NaNLORE algorithm.

The cutoff distance d_c of DP algorithm is set as the 2%th shortest distance and the density is computed with exponential kernel which are suggested by DP algorithm. Fig. 8 reveals the clustering results of DP algorithm. The results show that DP algorithm is a fast and relative effective method, especially for spherical data sets, and it has a certain capacity to non-spherical data sets, but

just as shown in the results of the last six data sets, DP algorithm can hardly correctly discover the clusters in manifold data sets.

The results of DPC-KNN-PCA algorithm are displayed in Fig. 9. Like DP algorithm, DPC-KNN-PCA is good at dealing with spherical data sets, but for data sets containing complex manifold structure, it fails to get the desired clustering results.

The results of NaNLORE algorithm on these synthetic data sets are presented in Fig. 10. NaNLORE algorithm avoids setting parameters by introducing natural neighbor. It uses natural neighbor-based density and density-adaptive distances between local representatives to construct decision graph, making it applicable to deal with spherical data sets and complex manifold data sets. From the ACC and NMI scores in Table 1, we can see that NaNLORE outperforms all other methods. Additionally, NaNLORE algorithm only needs to compute the density-adaptive distances between lo-

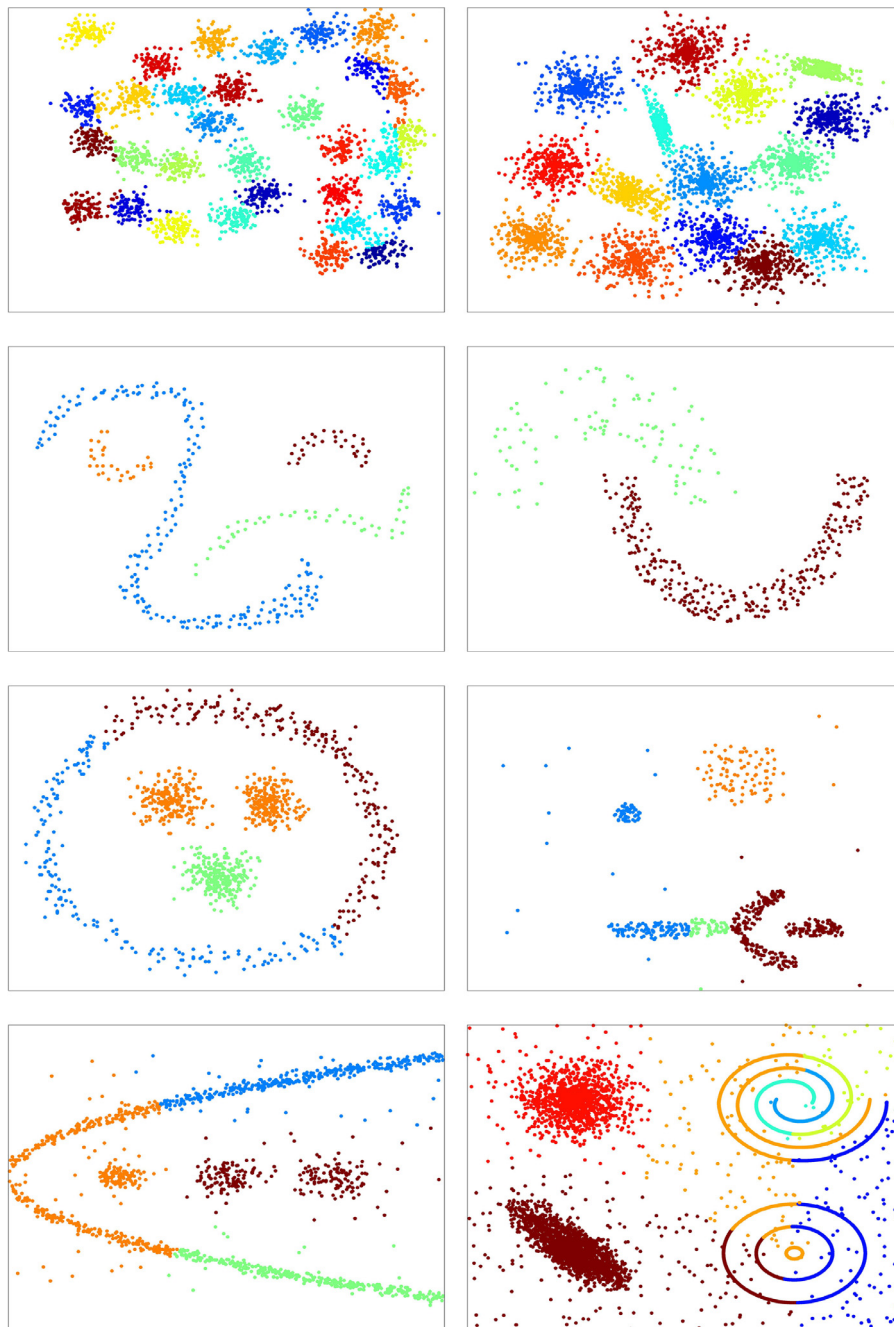


Fig. 7. Clustering results of DAAP. The preference parameter p of K-AP used in DAAP is set different for each point in a data set. For example, for point i , $p(i) = \text{median} \{S(i, k) | S(i, k) \neq -\text{inf}, k = 1, 2, \dots, n\}$, where $S(i, j)$ is the similarity of point i and point j .

cal representatives whose number is far less than data points number, therefore, the run time of NaNLORE is much less than that of DAAP algorithm.

4.3. Clustering on real data sets from UCI

To further demonstrate the effectiveness of NaNLORE, we do experiment on several benchmarking real data sets from UCI. The characteristics of these data sets are shown in Table 3. The comparison of ACC and NMI scores are illustrated in Table 4 and the running time of these algorithms is shown in Table 5.

According to Table 4, ACC and NMI are almost consistent. K-means gets the best results on Breast, which is as good as NaNLORE. However, it is not as effective as NaNLORE for other data

sets. K-means, AP, DP and DPC-KNN-PCA algorithms are center-based clustering methods. For AP algorithm, the accuracy of clustering is not high, as it is hard to control the number of clusters. The ACC and NMI scores of DP and DPC-KNN-PCA are relatively high for several simple data sets. However, as for data sets with high dimension and complex clusters, they cannot do as well as NaNLORE. Chameleon and DAAP are able to discover clusters with arbitrary shapes. However, the ACC and NMI scores of Chameleon are not high for most data sets, because it is susceptible to noise points. DAAP can make good use of the prior knowledge and divide each data set into a given number of clusters. However, the total running time of DAAP algorithm is the highest. For most data sets, the ACC and NMI scores of NaNLORE are higher than other algorithms. For Iris and Seed, NaNLORE gets the second best results.

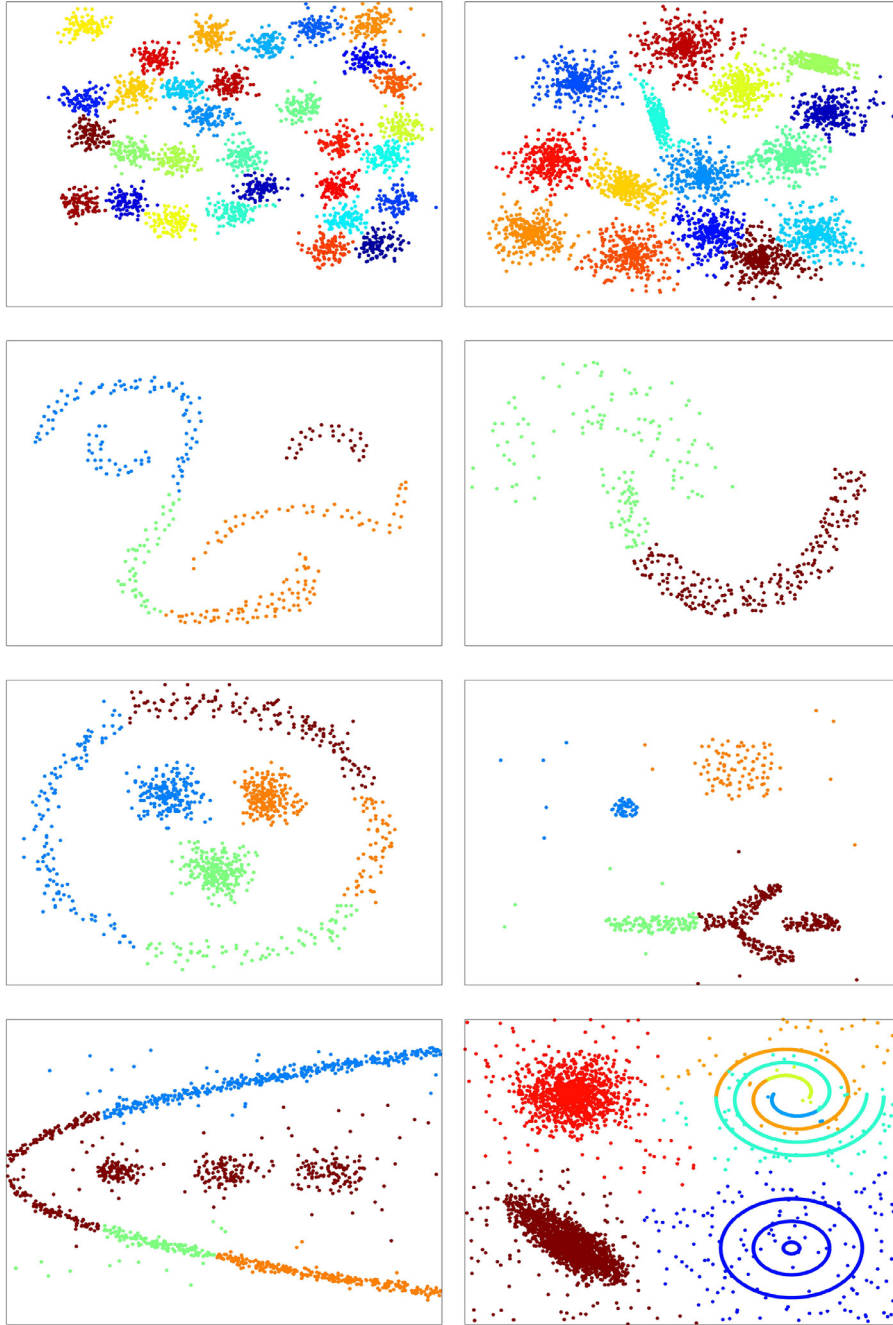


Fig. 8. Clustering results of DP.

The running time of NaNLORE algorithm is significantly less than Chameleon, AP and DAAP. Combining comparison results in terms of ACC, NMI and running time, we can conclude that NaNLORE algorithm is more effective than the compared algorithms.

4.4. Clustering on Olivetti Face Database

We also compare our algorithm with AP, DAAP and DP algorithms on the Olivetti Face Database [40]. The Olivetti Face Database contains 400 face images from 40 persons, taken at different times and varying the lighting, facial expressions and facial details. The size of each images is 92×112 pixels. We use the first 100 images, that is, 10 clusters of the database to do the experiment. The similarity between two images, denoted as $S(A, B)$, is

computed by the following equation.

$$S(A, B) = \frac{\sum_m \sum_n (A_{mn} - \bar{A})(B_{mn} - \bar{B})}{\sqrt{(\sum_m \sum_n (A_{mn} - \bar{A})^2)(\sum_m \sum_n (B_{mn} - \bar{B})^2)}} \quad (11)$$

Here A and B are the objects of Olivetti Face Database. A_{mn} and B_{mn} represent the pixels of the two images. The value of S is scaled to $[0,1]$. The bigger the value of S is, the more similar the two images are. The distance between two images, denoted as $d(A, B)$, is computed as follows:

$$d(A, B) = 1 - S(A, B) \quad (12)$$

Fig. 11 has shown clustering result of NaNLORE. In DP algorithm, the density is estimated by a Gaussian kernel with variance $d_c = 0.07$ and we do not consider finding noises. Clustering

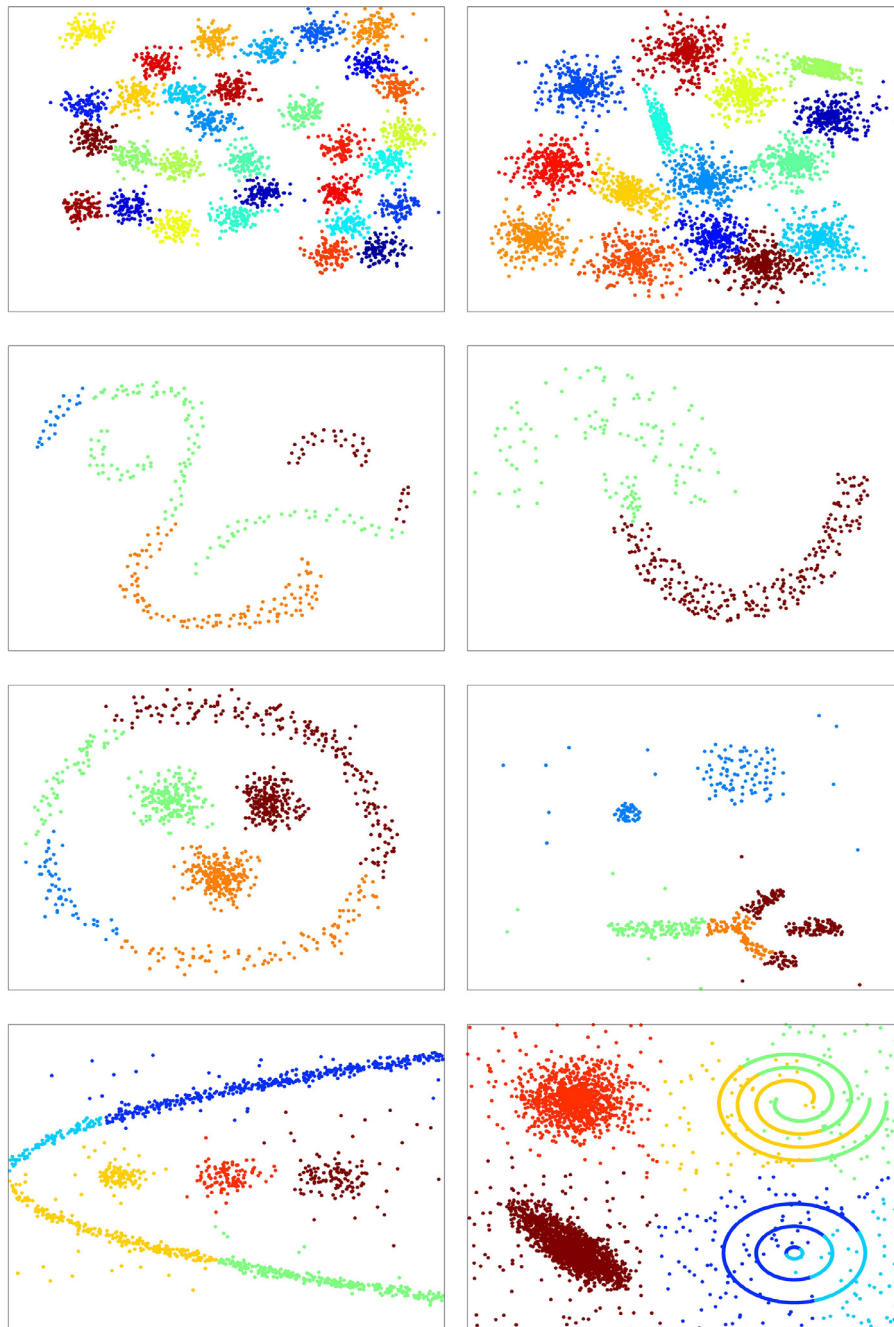


Fig. 9. Clustering results of DPC-KNN-PCA.

result of DP is shown in Fig. 12. AP and DAAP respectively discovered 11 clusters and 10 clusters, which are shown in Figs. 13 and 14. To analyze and compare the efficiency and the performance of AP, DAAP, DP and NaNLORE, we also consider ACC and NMI as the measurement standards. The results of our experiments are shown in Table 6.

The results of our experiments show that values of ACC and NMI of NaNLORE are higher than other algorithms. AP and DAAP identify cluster centers by passing messages between data points, and ignores the essential characteristic of cluster centers, leading to the undesired clustering results. DP assumes that cluster centers are surrounded by neighbors with lower local density and that they are at a relatively large distance from any points with a higher local density. Cluster centers are found quickly and accurately. NaNLORE inherits the advantage of DP algorithm and is

more capable to cluster data sets with multiple manifold structures and scales.

5. Conclusions and future work

In this paper, we propose a new clustering algorithm NaNLORE, and its core idea is to find local representatives. We first introduce a new neighbor concept Natural Neighbor which can automatically adapt to the distribution of data sets. On the basis of neighbors found by NaN-searching algorithm, point density is computed and local representatives are found. Subsequently, the density-adaptive distance between local representatives is computed on MMNG, and in this way, the global coherence of the local representatives is maintained. Last, we use DP algorithm to cluster the local representatives. The introduction of density-adaptive distance between

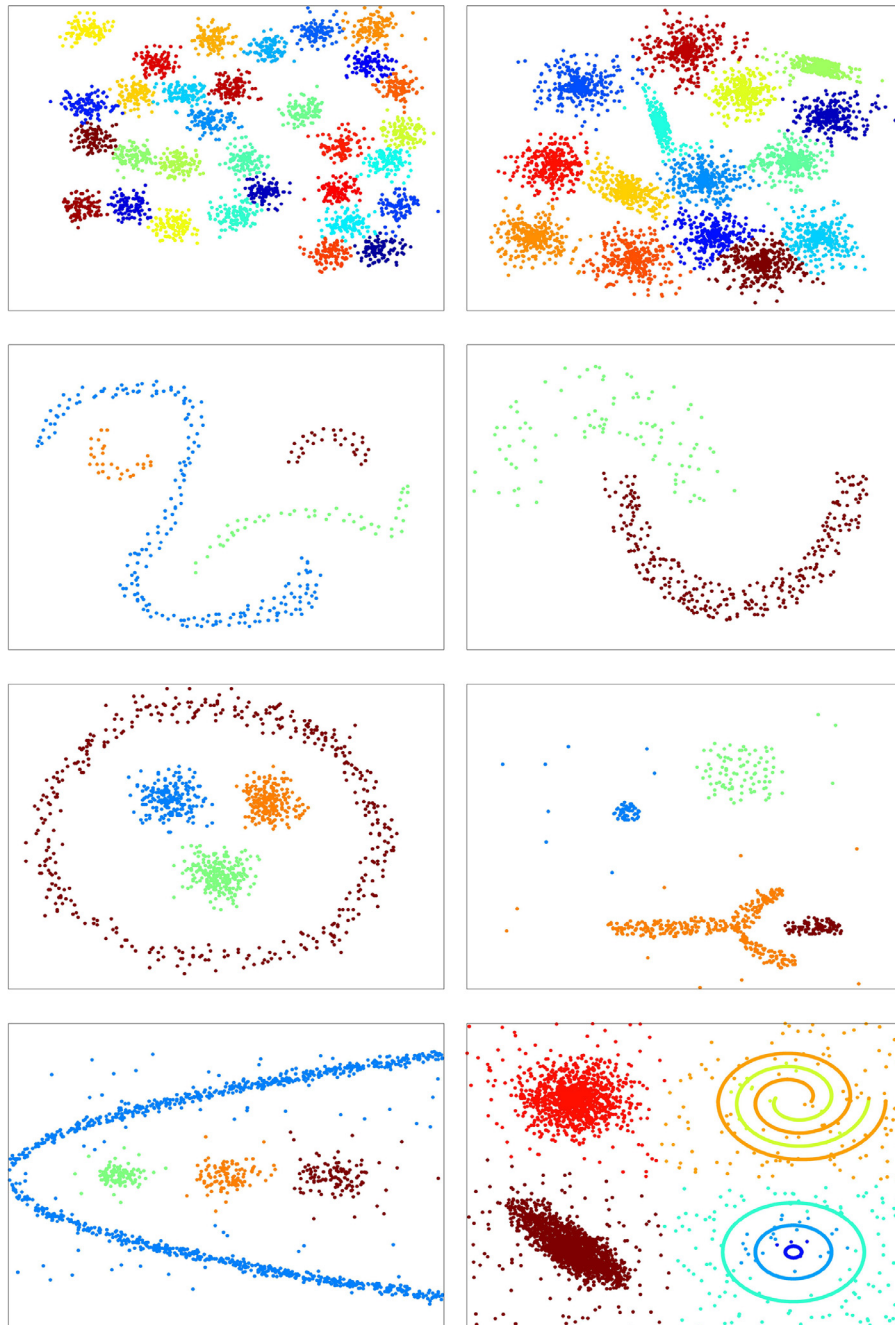


Fig. 10. Clustering results of NaNLORE.

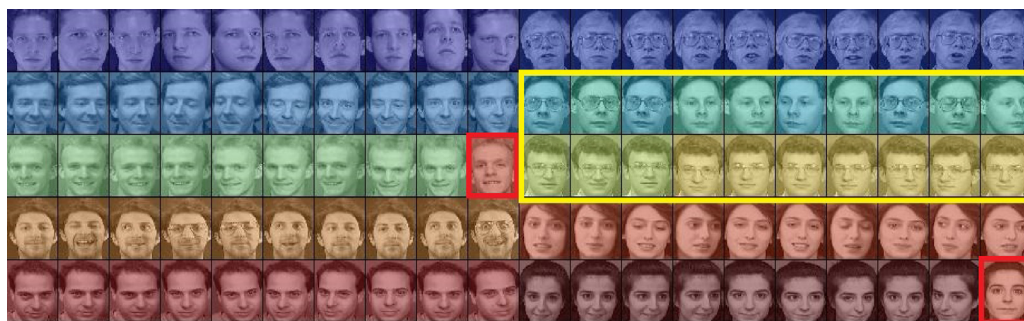


Fig. 11. Clustering results of NaNLORE on the Olivetti Face Database. Faces with the same color belong to the same cluster. NaNLORE detects 12 clusters, and two clusters enclosed by yellow box are divided into two clusters respectively. The faces enclosed by red box are assigned to the wrong cluster. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

Table 1

The comparison of the algorithms on synthetic datasets.

Datasets		K-means	Chameleon	AP	DAAP	DP	DPC-KNN-PCA	NaNLORE
Dataset 1	ACC	0.803	0.968	0.535	0.919	0.967	0.968	0.969
	NMI	0.919	0.958	0.817	0.921	0.957	0.957	0.958
Dataset 2	ACC	0.877	0.831	0.950	0.970	0.965	0.966	0.969
	NMI	0.909	0.905	0.939	0.945	0.939	0.940	0.946
Dataset 3	ACC	0.448	1.000	0.448	1.000	0.511	0.603	1.000
	NMI	0.472	1.000	0.418	1.000	0.487	0.460	1.000
Dataset 4	ACC	0.788	1.000	0.568	1.000	0.861	0.922	1.000
	NMI	0.374	1.000	0.536	1.000	0.507	0.646	1.000
Dataset 5	ACC	0.686	1.000	0.578	0.639	0.745	0.652	1.000
	NMI	0.564	1.000	0.510	0.799	0.612	0.573	1.000
Dataset 6	ACC	0.617	0.995	0.631	0.560	0.627	0.540	0.998
	NMI	0.640	0.974	0.632	0.546	0.660	0.568	0.991
Dataset 7	ACC	0.371	0.702	0.365	0.411	0.383	0.540	0.995
	NMI	0.183	0.035	0.514	0.479	0.276	0.546	0.964
Dataset 8	ACC	0.571	0.562	0.339	0.619	0.734	0.721	0.999
	NMI	0.704	0.657	0.613	0.642	0.797	0.741	0.995

Table 2

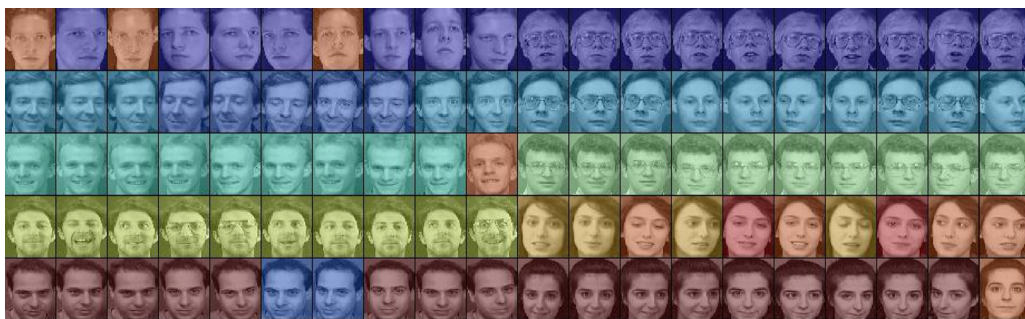
The running time of the algorithms on synthetic datasets.

Datasets	K-means	Chameleon	AP	DAAP	DP	DPC-KNN-PCA	NaNLORE
Dataset 1	0.060	349.345	121.508	4472.537	3.906	3.300	80.468
Dataset 2	0.077	633.283	224.687	12738.000	7.137	6.813	217.181
Dataset 3	0.005	3.238	9.351	17.041	0.081	0.078	1.093
Dataset 4	0.004	36.501	8.259	28.047	0.084	0.086	1.272
Dataset 5	0.007	124.339	23.668	183.178	0.412	0.437	3.741
Dataset 6	0.009	68.639	9.717	56.431	0.176	0.177	2.353
Dataset 7	0.010	209.780	22.525	608.790	0.733	0.587	13.178
Dataset 8	0.027	779.908	1976.100	51788.000	19.528	18.792	441.882

Table 3

Data characteristics of real data sets.

	Number of instances	Number of attributes	Number of clusters
Iris	150	4	3
Wine	178	13	3
Seed	210	7	3
Inosphere	351	34	2
Cancer	569	30	2
Control	600	60	6
Breast	699	10	2
Banknote	1372	4	2
Segment	2310	19	7
Letter	3864	16	5
Pageblocks	5473	10	5
Pendigits	7494	16	10

**Fig. 12.** Clustering results of DP on the Olivetti Face Database without considering noises. Faces with the same color belong to the same cluster. DP detects 12 clusters. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

local representatives enables the algorithm to deal with complex manifold data sets. The experimental results on both synthetic and real data sets suggest that NaNLORE algorithm can efficiently recognize both spherical clusters and complex manifold clusters. Compared with other methods, the time complexity of NaNLORE is relatively low. Moreover, NaNLORE also has broad application prospects, and it can be used to process some practical problems,

such as data analysis in 3D reconstruction, image segmentation, speech separation and character recognition.

A limitation of our method is that it cannot detect cluster centers automatically and we have to select cluster centers according to the new constructed decision graph. Our future research includes finding better solutions for this and verifying NaNLORE on different application domains.

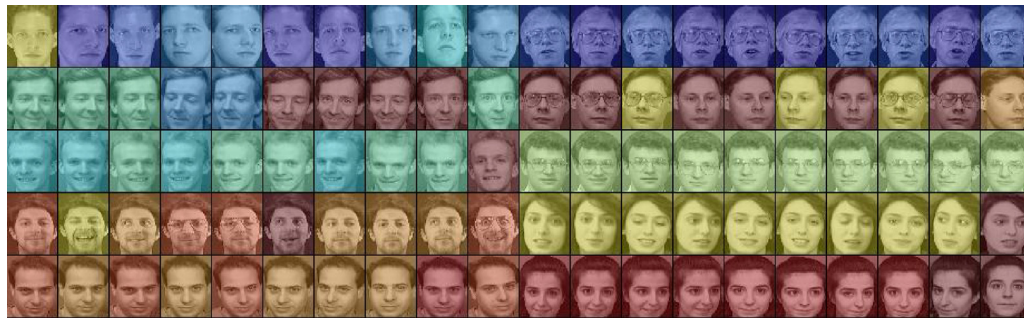


Fig. 13. Clustering results of AP on the Olivetti Face Database. Faces with the same color belong to the same cluster. AP detects 11 clusters. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

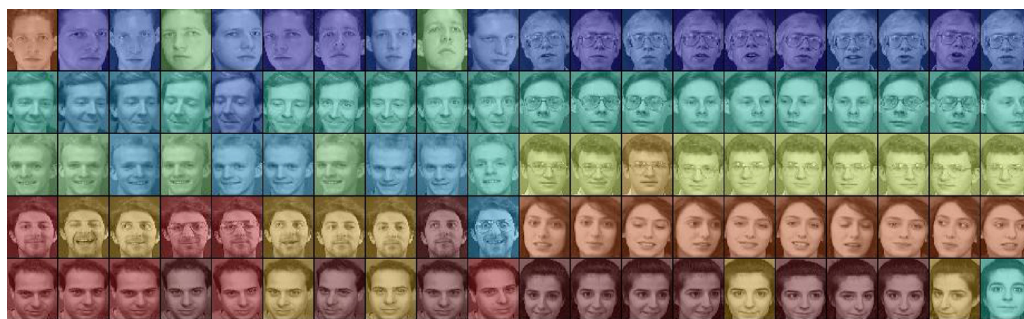


Fig. 14. Clustering results of DAAP on the Olivetti Face Database. Faces with the same color belong to the same cluster. DAAP detects 10 clusters. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

Table 4

The comparison of the algorithms on real datasets from UCI.

Datasets		K-means	Chameleon	AP	DAAP	DP	DPC-KNN-PCA	NaNLORE
Iris	ACC	0.887	0.693	0.667	0.960	0.907	0.880	0.907
	NMI	0.742	0.723	0.761	0.871	0.806	0.780	0.806
Wine	ACC	0.944	0.382	0.618	0.624	0.978	0.730	0.978
	NMI	0.816	0.040	0.576	0.389	0.909	0.765	0.911
Seed	ACC	0.890	0.519	0.848	0.895	0.886	0.914	0.905
	NMI	0.710	0.391	0.646	0.695	0.698	0.724	0.695
Inonsphere	ACC	0.712	0.647	0.726	0.684	0.678	0.729	0.732
	NMI	0.135	0.039	0.123	0.067	0.084	0.125	0.145
Cancer	ACC	0.663	0.659	0.619	0.706	0.564	0.735	0.886
	NMI	0.031	0.106	0.366	0.001	0.039	0.299	0.488
Control	ACC	0.582	0.608	0.463	0.698	0.557	0.340	0.705
	NMI	0.709	0.809	0.588	0.797	0.746	0.368	0.816
Breast	ACC	0.944	0.641	0.601	0.595	0.848	0.939	0.944
	NMI	0.683	0.014	0.521	0.069	0.425	0.661	0.683
Banknote	ACC	0.610	0.577	0.342	0.577	0.741	0.742	0.875
	NMI	0.029	0.067	0.410	0.067	0.347	0.348	0.582
Segment	ACC	0.481	0.532	0.493	0.367	0.482	0.146	0.631
	NMI	0.461	0.506	0.433	0.312	0.529	0.043	0.609
Letter	ACC	0.496	0.275	0.253	0.525	0.416	0.408	0.666
	NMI	0.339	0.151	0.453	0.405	0.292	0.405	0.534
Pageblocks	ACC	0.711	0.672	0.778	0.302	0.899	0.898	0.912
	NMI	0.049	0.164	0.030	0.111	0.110	0.091	0.294
Pendigits	ACC	0.689	0.699	0.368	0.704	0.766	0.593	0.785
	NMI	0.682	0.775	0.672	0.659	0.765	0.647	0.764

Table 5

The running time of the algorithms on real datasets from UCI.

Datasets	K-means	Chameleon	AP	DAAP	DP	DPC-KNN-PCA	NaNLORE
Iris	0.004	25.787	6.177	4.557	0.051	0.095	0.185
Wine	0.005	31.708	5.945	6.394	0.079	0.077	0.365
Seed	0.007	32.051	7.139	13.839	0.056	0.117	0.640
Inonsphere	0.048	53.234	6.855	78.590	0.234	0.302	0.366
Cancer	0.039	69.989	13.540	412.175	0.542	0.854	5.792
Control	0.036	68.740	8.181	92.342	1.625	2.104	2.898
Breast	0.033	116.459	22.496	128.217	0.157	0.421	1.924
Banknote	0.026	288.077	26.493	1488.029	3.393	3.721	39.680
Segment	0.064	378.225	53.659	4525.677	5.364	5.610	97.375
Letter	0.031	621.883	236.279	15826.944	13.514	14.021	235.497
Pageblocks	0.135	991.271	307.259	38942.025	22.295	20.722	291.245
Pendigits	0.313	1000.958	1309.610	86167.895	52.241	51.443	442.273

Table 6

The comparison of ACC and NMI on the Olivetti Face Database.

	AP	DAAP	DP	NaNLORE
ACC	0.59	0.61	0.78	0.91
NMI	0.67	0.68	0.85	0.95

Acknowledgments

We thank Xiangliang Zhang for providing the K-AP matlab code. We also thank the anonymous reviewers for their helpful comments which have led to many improvements in this paper. This work is supported by Project of Chongqing Education Commission (No.KJZH17104), National Natural Science Foundation of China (61272194) and National Natural Science Foundation of China (61502060).

Supplementary material

Supplementary material associated with this article can be found, in the online version, at [10.1016/j.knosys.2017.02.027](https://doi.org/10.1016/j.knosys.2017.02.027).

References

- [1] J. MacQueen, Some methods for classification and analysis of multivariate observations, in: *Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability*, vol. 1, 1967, pp. 281–297. Oakland, CA, USA.
- [2] P.J. Rousseeuw, *Finding Groups in Data: An Introduction to Cluster Analysis*, John Wiley & Sons, 2009.
- [3] B. King, Step-wise clustering procedures, *J. Am. Stat. Assoc.* 62 (317) (1967) 86–101.
- [4] W.W. Moss, H. Ja, Numerical taxonomy, *Annu. Rev. Entomol.* 18 (1973) 227–258.
- [5] K. Zhou, S. Yang, Exploring the uniform effect of fcm clustering: a data distribution perspective, *Knowl. Based Syst.* 96 (2016) 76–83.
- [6] K. Zhou, C. Fu, S. Yang, Fuzziness parameter selection in fuzzy c-means: the perspective of cluster validation, *Sci. China-Inf. Sci.* 57 (11) (2014) 1–8.
- [7] D.G. Ferrari, L.N. de Castro, Clustering algorithm selection by meta-learning systems: a new distance-based problem characterization and ranking combination methods, *Inf. Sci.* 301 (2015) 181–194.
- [8] T. Zhang, R. Ramakrishnan, M. Livny, Birch: An efficient data clustering method for very large databases, in: *ACM Sigmod Record*, vol. 25, 1996, pp. 103–114. ACM.
- [9] S. Guha, R. Rastogi, K. Shim, Cure: An efficient clustering algorithm for large databases, in: *ACM SIGMOD Record*, vol. 27, 1998, pp. 73–84. ACM.
- [10] S. Guha, R. Rastogi, K. Shim, Rock: A robust clustering algorithm for categorical attributes, in: *Data Engineering, 1999. Proceedings., 15th International Conference on, IEEE, 1999*, pp. 512–521.
- [11] G. Karypis, E.-H. Han, V. Kumar, Chameleon: hierarchical clustering using dynamic modeling, *Computer* 32 (8) (1999) 68–75.
- [12] B.J. Frey, D. Dueck, Clustering by passing messages between data points, *Science* 315 (5814) (2007) 972–976.
- [13] A. Rodriguez, A. Laio, Clustering by fast search and find of density peaks, *Science* 344 (6191) (2014) 1492–1496.
- [14] M.J. Du, S.F. Ding, H.J. Jia, Study on density peaks clustering based on k-nearest neighbors and principal component analysis, *Knowl. Based Syst.* 99 (2016) 135–145.
- [15] H. Jia, S. Ding, L. Meng, S. Fan, A density-adaptive affinity propagation clustering algorithm based on spectral dimension reduction, *Neural Comput. Appl.* 25 (2014) 1557–1567. 7–8.
- [16] X. Zhang, W. Wang, K. Norvag, M. Sebag, K-ap: generating specified k clusters by efficient affinity propagation, in: *Data Mining (ICDM), 2010 IEEE 10th International Conference on, IEEE, 2010*, pp. 1187–1192.
- [17] E. Elhamifar, R. Vidal, Sparse subspace clustering: algorithm, theory, and applications, *IEEE Trans. Pattern Anal. Mach. Intell.* 35 (11) (2013) 2765–2781.
- [18] Z. Liang, P. Chen, Delta-density based clustering with a divide-and-conquer strategy: 3dc clustering, *Pattern Recognit. Lett.* 73 (2016) 52–59.
- [19] J.Y. Xie, H.C. Gao, W.X. Xie, X.H. Liu, P.W. Grant, Robust clustering by detecting density peaks and assigning points based on fuzzy weighted k-nearest neighbors, *Inf. Sci.* 354 (2016) 19–40.
- [20] G. Wang, Q. Song, Automatic clustering via outward statistical testing on density metrics, *IEEE Trans. Knowl. Data Eng.* 28 (8) (2016) 1971–1985.
- [21] Y. Wang, Y. Jiang, Y. Wu, Z.-H. Zhou, Spectral clustering on multiple manifolds, *Neural Netw. IEEE Trans.* 22 (7) (2011) 1149–1161.
- [22] Q. Yang, P. Zhu, B. Huang, Spectral clustering with density sensitive similarity function, *Knowl. Based Syst.* 24 (5) (2011) 621–628.
- [23] U. von Luxburg, A tutorial on spectral clustering, *Stat. Comput.* 17 (4) (2007) 395–416.
- [24] E. Tu, L. Cao, J. Yang, N. Kasabov, A novel graph-based k-means for nonlinear manifold clustering and representative selection, *Neurocomputing* 143 (2014) 109–122.
- [25] D. Zhou, O. Bousquet, T.N. Lal, J. Weston, B. Schölkopf, Learning with local and global consistency, *Adv. Neural Inf. Process. Syst.* 16 (16) (2004) 321–328.
- [26] R.A. Jarvis, E.A. Patrick, Clustering using a similarity measure based on shared near neighbors, *Comput. IEEE Trans.* 100 (11) (1973) 1025–1034.
- [27] Q. Zhu, J. Feng, J. Huang, Natural neighbor: a self-adaptive neighborhood method without parameter k, *Pattern Recognit. Lett.* 80 (2016) 30–36.
- [28] J. Huang, Q. Zhu, L. Yang, J. Feng, A non-parameter outlier detection algorithm based on natural neighbor, *Knowl. Based Syst.* 92 (2016) 71–77.
- [29] L. Yang, Q. Zhu, J. Huang, D. Cheng, Adaptive edited natural neighbor algorithm, *Neurocomputing* 230 (2017) 427–433.
- [30] J.L. Bentley, Multidimensional binary search trees used for associative searching, *Commun. ACM* 18 (9) (1975) 509–517.
- [31] X. Luo, Y. Xia, Q. Zhu, Y. Li, Boosting the k-nearest-neighborhood based incremental collaborative filtering, *Knowl. Based Syst.* 53 (2013) 90–99.
- [32] J.B. Tenenbaum, V. De Silva, J.C. Langford, A global geometric framework for nonlinear dimensionality reduction, *Science* 290 (5500) (2000) 2319–2323.
- [33] M. Wu, B. Schölkopf, A local learning approach for clustering, in: *Advances in Neural Information Processing Systems, 2006*, pp. 1529–1536.
- [34] C.J. Veenman, M.J.T. Reinders, E. Backer, A maximum variance cluster algorithm, *IEEE Trans. Pattern Anal. Mach. Intell.* 24 (9) (2002) 1273–1280.
- [35] P. Frnti, O. Virmajoki, Iterative shrinking method for clustering problems, *Pattern Recognit.* 39 (5) (2006) 761–775.
- [36] M. Ester, H.-P. Kriegel, J. Sander, X. Xu, A density-based algorithm for discovering clusters in large spatial databases with noise, in: *Kdd*, vol. 96, 1996, pp. 226–231.
- [37] A.K. Jain, M.H. Law, *Data Clustering: A Users Dilemma*, Springer, 2005, pp. 1–10.
- [38] J. Ha, S. Seok, J.S. Lee, Robust outlier detection using the instability factor, *Knowl. Based Syst.* 63 (2014) 15–23.
- [39] G.X. Ritter, J.-A. Nieves-Vzquez, G. Urcid, A simple statistics-based nearest neighbor cluster detection algorithm, *Pattern Recognit.* 48 (3) (2015) 918–932.
- [40] F.S. Samaria, A.C. Harter, Parameterisation of a stochastic model for human face identification, in: *Applications of Computer Vision, 1994., Proceedings of the Second IEEE Workshop on, IEEE, 1994*, pp. 138–142.