Expert Systems with Applications 38 (2011) 9373–9381

Contents lists available at ScienceDirect

Expert Systems with Applications

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

A new hybrid method based on partitioning-based DBSCAN and ant clustering

Hua Jiang *, Jing Li, Shenghe Yi, Xiangyang Wang, Xin Hu

College of Computer Science, Northeast Normal University, Changchun, Jilin 130117, China

ARTICLE INFO

Keywords:
Clustering
Density-based clustering
DBSCAN
PDBSCAN
Ant clustering algorithm

ABSTRACT

Clustering problem is an unsupervised learning problem. It is a procedure that partition data objects into matching clusters. The data objects in the same cluster are quite similar to each other and dissimilar in the other clusters. Density-based clustering algorithms find clusters based on density of data points in a region. DBSCAN algorithm is one of the density-based clustering algorithms. It can discover clusters with arbitrary shapes and only requires two input parameters. DBSCAN has been proved to be very effective for analyzing large and complex spatial databases. However, DBSCAN needs large volume of memory support and often has difficulties with high-dimensional data and clusters of very different densities. So, partitioning-based DBSCAN algorithm (PDBSCAN) was proposed to solve these problems. But PDBSCAN will get poor result when the density of data is non-uniform. Meanwhile, to some extent, DBSCAN and PDBSCAN are both sensitive to the initial parameters. In this paper, we propose a new hybrid algorithm based on PDBSCAN. We use modified ant clustering algorithm (ACA) and design a new partitioning algorithm based on 'point density' (PD) in data preprocessing phase. We name the new hybrid algorithm PACA-DBSCAN. The performance of PACA-DBSCAN is compared with DBSCAN and PDBSCAN on five data sets. Experimental results indicate the superiority of PACA-DBSCAN algorithm.

1. Introduction

Clustering is a popular data analysis technique. Clustering algorithms can be widely applied in many fields including: pattern recognition, machine learning, image processing, information retrieval and so on. It also plays an important role in data mining (Sun, Liu, & Zhao, 2008).

The clustering algorithms usually can be classified into the following four categories: (a) partitional clustering; (b) density-based and grid-based clustering; (c) hierarchical clustering; (d) other clustering (Birant & Kut, 2007).

All the existing clustering algorithms have their own characteristics, but also have there own flaws. As a kind of partitional clustering, *k*-means algorithm is simple and high efficiency, but it can only discover spherical clusters. The hierarchical clustering algorithms can find non-elliptical clusters, but they are sensitive to noise and are not suitable for large databases. DBSCAN can discover clusters of arbitrary shape. But it is sensitive to the input parameters, especially when the density of data is non-uniform. On the other hand, DBSCAN has difficulties with high-dimensional databases (Gan, Ma, & Wu, 2007). So, partitioning-based DBSCAN algorithm (PDBSCAN) was proposed to solve some defects of DBSCAN, but it is still sensitive to the input parameters and density of clusters.

Traditional optimization algorithms are also used with clustering algorithms to improve clustering effect. Traditional optimization algorithms include: greedy algorithm, exhaustive search algorithm, local search heuristics, method of dynamic programming and so on. However, the traditional optimization algorithms have their own weakness. They are usually designed for specific issues, and only effective on certain types of issues. So, many new optimization algorithms have been proposed, such as Tabu search (TS), simulated annealing (SA), particle swarm optimization (PSO), ant clustering algorithm (ACA), genetic algorithm (GA). Tabu search (TS) is a search method used to solve the combinatorial optimization problems. Particle swarm optimization (PSO) is a popular stochastic optimization technique developed by Kennedy and Eberhart.

Recently, many researches have combined clustering algorithms with optimization algorithms to improve the results of clustering algorithms. Simulated annealing (SA) algorithm is always used to solve the combinatorial problems. Simulated annealing heuristic was used with *k*-harmonic means to overcome local optimal problem (Güngör & Ünler, 2007). The algorithm TabuKHM (Tabu *K*-Harmonic Means) was proposed in 2008 (Güngör & Ünler, 2008). In 2009 a new hybrid algorithm based on PSO and KHM was proposed (Yang & Sun, 2009). Moreover, some other hybrid heuristic methods such as genetic simulated annealing or Tabu search with simulated annealing were ever used with clustering algorithm to solve local optimal problem (Chu & Roddick, 2003; Huang, Pan, Lu, Sun, & Hang, 2001; Kao, Zahara, & Kao, 2008). Although there are many researches on clustering algorithms combining optimization algorithms, little research choose density-based clustering algorithms.

* Corresponding author. Fax: +86 0431 84536331.
E-mail address: jiangh289@nenu.edu.cn (H. Jiang).

0957-4174/$ - see front matter © 2011 Elsevier Ltd. All rights reserved.
doi:10.1016/j.eswa.2011.01.135

In this paper, we propose a new hybrid partitioning-based DBSCAN algorithm combining with modified ant clustering algorithm (PACA-DBSCAN). The rest of the paper is organized as follows. Section 2 describes the clustering algorithm and provides the related work. Section 3 introduces the ant clustering algorithm. Section 4 describes how to partition database in PDBSCAN algorithm and its defects. Section 5 presents our new hybrid clustering algorithm PACA-DBSCAN in detail. After that, we explain the data sets and the experimental results in Section 6. Finally, Section 7 makes conclusions.

2. Clustering

Clustering is the unsupervised classification of patterns (observations, data items, or feature vectors) into groups (Jain, Murty, & Flynn, 1999). This process does not need prior knowledge about the database. Clustering procedure partition a set of data objects into clusters such that objects in the same cluster are more similar to each other than objects in different clusters according to some predefined criteria (Güngör & Ünler, 2007). These data objects are also called data points or points, and the database is usually refered as a data set.

There are four categories of clustering. The partitional clustering, such as *k*-means, can only discover spherical clusters. It is sensitive to the noise and the center points. The better center points we choose, the better results we get (Hammerly & Elkan, 2002). Generally, the computational complexity of the hierarchical clustering is O (n^2), where n is the total number of objects. So they are usually used to analyze small database and cannot revoke the prior work. The grid-based clustering algorithms are not suitable for high-dimensional database (Xu & Wunshc, 2005). In this paper, we pay attention to density-based clustering, and especially focus on DBSCAN.

2.1. DBSCAN: a density-based clustering

Density-based clustering defines cluster as region, the objects of the region are dense. The clusters are separated from one another by low-density regions (Thanh, Wehrens, & Lutgarde, 2006). The reason we choose density-based clustering is that it has significant advantages over partitional and hierarchical clustering algorithms. It can discover clusters of arbitrary shapes. The computational complexity can be reduced to O (n*lgn) by building some special data structures. In addition it is able to effectively identify noise points (Cao, Ester, Qian, & Zhou, 2006). But density-based clustering algorithms easily lead to memory problem when facing large databases. Some researches show that current density-based clustering algorithms often have difficulties with complex data sets in which the clusters are different densities (Comaniciu & Meer, 1999).

As a kind of density-based clustering, the DBSCAN algorithm was first introduced by Ester, et al. The key idea in DBSCAN is that for each data object of a cluster, the neighborhood of a given radius (Eps) has to contain at least a minimum number (MinPts) of objects (Zhou, Zhou, Cao, Fan, & Hu, 2000). Some basic concepts related with DBSCAN are as follow (Ester, Kriegel, Sander, & Xu, 1996):

Definition 1 (*Eps-neighborhood of a point).* The Eps-neighborhood of a point p, denoted by $NEps(p)$, is defined by $NEps(p) = \{q \in D|dist(p,q) \leqslant Eps\}$.

Definition 2 (*Directly density-reachable*). An object p is directly density-reachable from an object q wrt. Eps and MinPts in the set of objects D if

(1) $p \in NEps(q)$ ($NEps(q)$ is the Eps-neighborhood of q),
(2) $|NEps(q)| \geqslant MinPts$ (Core point condition).

Definition 3 (*Core object & border object*). An object is core object if it satisfies condition (2) of Definition 2, and a border object is such an object that is not a core object itself but is density-reachable from another core object.

Definition 4 (*Density-reachable*). A point p is density reachable from a point q wrt. Eps and MinPts if there is a chain of points $p_1,\ldots,p_n$, $p_1 = q$, $p_n = p$ such that $p_i + 1$ is directly density-reachable from p_i.

Definition 5 (*Density-connected*). An object p is density-connected to an object q wrt. Eps and MinPts in the set of objects D if there is an object $o \in D$ such that both p and q are density-reachable from o wrt. Eps and MinPts in D.

Definition 6 (*Cluster*). Let D be a database of points. A cluster C wrt. Eps and MinPts is a non-empty subset of D satisfying the following conditions:

(1) $\forall p, q$: if $p \in C$ and q is density-reachable from p wrt. Eps and MinPts, then $q \in C$. (Maximality).
(2) $\forall p, q \in C$: p is density-connected to q wrt. Eps and MinPts. (Connectivity).

Definition 7 (*Noise*). Let $C_1,\ldots,C_k$ be the clusters of the database D wrt. parameters Eps_i and MinPts_i, $i = 1,\ldots,k$. Then the noise is the set of points in the database D not belonging to any cluster C_i, i.e. noise = $\{p \in D|\forall i : p \notin C_i\}$.

DBSCAN algorithm starts from an arbitrary point q, and retrieves all points density-reachable from q wrt. Eps and MinPts. If q is a core point, create a new cluster and assign the point q and its neighbors into this new cluster. Then the algorithm iteratively collects the neighbors within Eps distance from the core points. The process is repeated until all of the points have been processed. If q is a border point, no points are density-reachable from q and DBSCAN visits the next point of the database (Birant & Kut, 2007; Viswanath & Pinkesh, 2006).

2.2. PDBSCAN: partitioning-based DBSCAN algorithm

Because of the memory problem, the researches begin to partition large data set into some small partitions. In 2000 partitioning technique was first used in DBSCAN algorithm. It run DBSCAN algorithm on each partition which is partitioned by special rules.

With PDBSCAN, the R^*-tree should be built. DBSCAN requires to specify two global parameters Eps and MinPts. In order to reduce the computational complexity, MinPts is fixed to 4 usually. Then the k-dist graph must be plotted to decide the value of Eps. K-dist graph needs to calculate the distance of an object and its kth nearest neighbors for all the objects. Next, sort all the objects on the basis of the previous distances. Finally, plot the k-dist graph according to all the sorted objects and distances. Considering that building the R^*-tree and plotting the k-dist graph have to cost much time especially for a large database, the initial database is partitioned into N partitions to reduce the time cost. Partitioning database can also alleviate the burden of memory and find more precise parameter Eps for every partition.

The steps of PDBSCAN are as follow:

(1) Partitioning the initial database into N partitions.
(2) For each partition, building local R^*-tree, analyzing and selecting local Eps and MinPts, and then clustering it with DBSCAN.
(3) Merging the partial clusters.

In the first step of PDBCAN, some articles partition database over the data dimensions. This method will lead to many problems. We will introduce the partitioning technique of PDBSCAN in Section 4 and analyze the second step of PDBSCAN through experiment in Section 6. So we describe the third step of PDBSCAN first.

2.2.1. Merging the partial clusters

For two partial clusters A and B, they can be merged if and only if (Zhou et al., 2000):

(1) A and B are distributed in two adjacent regions P_A and P_B respectively;
(2) $Eps(P_A)$ is the Eps value of P_A and $Eps(P_B)$ means the Eps value of P_B. And $MEps(P_A,P_B) = \min\{Eps(P_A), Eps(P_B)\}$. E_A and E_B are the border point sets of A and B, then $\forall p_i \in E_A$, $\forall q_j \in E_B$, $N_A = |E_A|$, $N_B = |E_B|$,

$$\overline{dist\ 1} = \frac{\sum_{i=1}^{N_A}\sum_{j=1}^{N_B} DISTANCE(p_i,q_j)}{N_A \bullet N_B} \leqslant MEps(P_A,P_B),$$

where $DISTANCE(p_i,q_j) = \sqrt{\sum_{k=1}^{m}(p_{ik}-q_{jk})^2}$, m is the dimension of data points p_i and q_j.

2.2.2. Merging noise points into the partial cluster

Some noise points nearby the partitioning line maybe belong to another partial cluster located at neighbor region of these noise points. So we need to merge this kind of noise points. One noise point p_n can be merged into a cluster C if and only if:

(1) p_n and C are distributed in two adjacent regions respectively;
(2) E_C is the set of all the border points in cluster C, then $\forall p_i \in E_C$, $N_C = |E_C|$,

$$\overline{dist\ 2} = \frac{\sum_{i=1}^{N_C} DISTANCE(p_n,p_i)}{N_C} \leqslant Eps(P_c).$$

2.2.3. Merging noise points into a new cluster

When a quite small cluster locates at the partition line, it is possibly disrupted and labeled as noise points because the neighborhood of Eps does not contain the minimum number (MinPts) of points which actually belong to the small cluster. Suppose P_A and P_B are two adjacent regions. E_A and E_B are sets of points in two partitioning border area. SN is the set of noise points in E_A and E_B. If $\exists p_0 \in SN$, $\exists p_i \in SN(i = 1,2,\ldots,n,\ \ n \geqslant MinPts)$, $p_i \neq p_0$, and $DISTANCE(p_0,p_i) \leqslant MEps(p_A,p_B)$ holds true, then p_0 is the core point of the new cluster. The points of set $\{p_i\}(i = 1,2,\ldots,n,\ n \geqslant MinPts)$ are marked as the members of new cluster. Then go onto treat the other points in SN similarly.

3. Ant clustering algorithm

Ant clustering was introduced by Deneubourg et al. (1991). Then Lumer and Faieta (1994) proposed the Standard Ant Clustering Algorithm (SACA). It closely mimics the ant behavior of gathering the corpses and sorting the larvas. SACA defines a two-dimensional grid. The size of grid is dependent on the number of objects. The SACA algorithm scatters the data objects onto this grid and makes a group of agent ants work on this two-dimensional grid at the same time. The agent ants have three actions: picking up the objects, moving on the grid and dropping the objects (Handl & Meyer, 2007). Every agent ant on the grid will occur in the following two situations:

(1) An agent ant holds an object i and evaluates the probability of dropping it on its current position;
(2) An agent ant is unloaded and evaluates the probability of picking up an object i.

The SACA can cluster data without any initial knowledge. The agent ants just pick up and drop the objects influenced by the similarity and density of the objects within the agent ant's current neighborhood. The probability of picking up an object increases with low density neighborhoods, and decreases with high similarity among objects in the surrounding area. On the contrary, the probability of dropping an object will increase with high density neighborhoods. Gathering process is the positive feedback of the behavior of the ants (Handl, Knowles, & Dorigo, 2003; Vizine & de Castro, 2005).

The probability of picking up and dropping an object i is described as follows:

$$p_{pick}(i) = \left(\frac{k_p}{k_p + f(i)}\right)^2,$$

$$p_{drop}(i) = \begin{cases} 2f(i) & \text{if } f(i) < k_d, \\ 1 & \text{otherwise}, \end{cases}$$

where k_p and k_d are constants and $f(i)$ is a similarity density measure for object i in its current neighborhood Γ. It is defined as

$$f(i) = \begin{cases} \frac{1}{s^2}\sum\limits_{j\in Neigh(\Gamma)}(1 - d(i,j)/\alpha) & \text{if } f(i) > 0, \\ 0 & \text{otherwise}, \end{cases}$$

where s^2 is the size of local neighborhood Γ around object i, and α is a scaling factor explained the dissimilarity measure $d(i,j)$ between objects i and j. $d(i,j)$ is the Euclidean distance between object i and j in their N-dimensional space.

Before the pre-defined number of iterations of algorithm, the agent ants keep on working to take the objects to the area where the objects are similar enough. When the iteration terminates, the objects on the grid are clustered automatically according to the density of the data itself.

4. Problems of partitioning approach in PDBSCAN

Partitioning database is one of the most important steps for PDBSCAN. In this step, the algorithm needs to divide the database into N so that the parameter Eps of each partition can be specified more exactly. If a database is partitioned at random, the results of second step that run DBSCAN algorithm with every data partition will be affected, and the third step that merge the partial clusters will be difficult.

We can see Figs. 1 and 2 that the cluster C_1, C_2 and C_3 are dense, the cluster C_4 and C_5 are sparse. If the database is partitioned into two areas as Fig. 1, the density of data points in each area is uniform. So the parameter Eps can be fixed in terms of the densities of C_1, C_2 or C_3 for one area, and specify parameter Eps based on the densities of C_4 or C_5 for another area. Then cluster C_1, C_2, C_3, C_4 and C_5 will be clustered easily by DBSCAN algorithm in the next step. On the contrary, if the database is partitioned as Fig. 2, data points differ greatly in density in any one of areas. If the parameter Eps for each area is specified based on the densities of C_1, C_2 or C_3, the cluster C_4 and C_5 will be disrupted into some small clusters. Otherwise, if the parameter Eps is fixed according to the densities of C_4 or C_5, the cluster C_1, C_2 and C_3 will be merged into a big cluster.

In the past, many articles partitioned database over the data dimensions. In this method, they should determine the dimension over which the partitioning is carried out first. Usually, histogram is needed to analyze the distribution of data. But if the data is multi-dimensional, this partitioning technique has three serious defects:

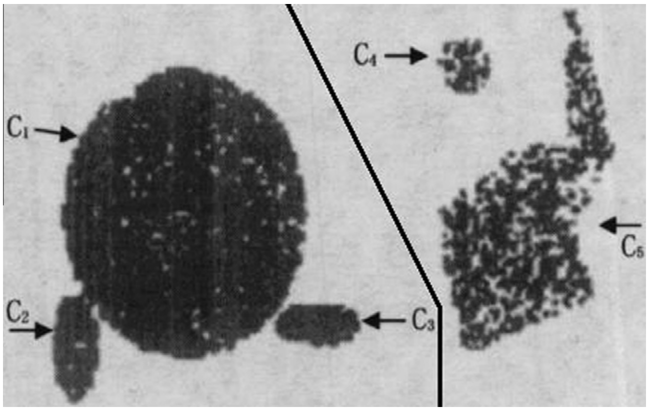


Fig. 1. The first partitioning case.

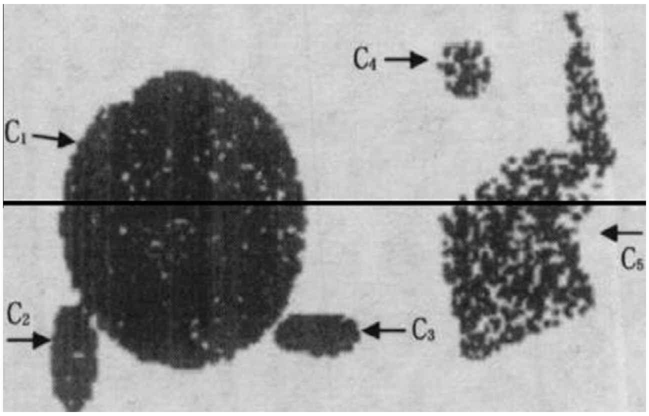


Fig. 2. The second partitioning case.

(1) The algorithm has to check the situation of every dimension and plot histogram for every dimension, this process not only costs much time but also reduces the accuracy of algorithm.
(2) It is quite difficult to merge the partial clusters in the third step because of the multi-dimensional data.
(3) When the shape of database is ring type or spirality entangled together, this approach that make data project to one dimension can not be effective.

Although PDBSCAN can conquer some drawback of DBSCAN in some degree, it also has its own limitations. Because PDBSCAN cannot divide data based on density, the densities of each partition are uneven. So PDBSCAN is still sensitive to the input parameter Eps and dimension of data. For the special shapes of data, the result of algorithm is also poor. In order to partition database in terms of density of data, we propose a new partitioning technique based on point density and describe it in detail in the next section.

5. The new hybrid method based on partitioning-based DBSCAN with ant clustering (PACA-DBSCAN)

In this section, we propose a new hybrid clustering algorithm. The new algorithm divides data according to the 'point density' that will be defined later. And we use modified ant clustering algorithm to partition multi-dimensional data to reduce time cost and improve accuracy.

5.1. Partitioning method based on point density

We first define a few new concepts and then propose a new partitioning method.

Definition 8 (*γ-neighborhood of a point).* The γ-neighborhood of a point i, denoted by $N_\gamma(i)$, is defined by $N_\gamma\ (i) = \{q \in D | dist(q, i) \leqslant \gamma\}$.

Definition 9 (*Point density, or PD).* The relative density with point i, denoted by $Den(i)$, is defined by $Den(i) = |N_\gamma(i)|/|D|$. The steps of the PD-based partitioning method are as follow:

(1) Set the initial parameter N (N is the number of partitions);
(2) Calculate the $Den(i)$ for each object i;
(3) Run the k-means algorithm to cluster data into N partitions based on the value of $Den(i)$.

5.2. Partitioning method based on modified ant clustering algorithm

To deal with high-dimensional data, we give another partitioning method which combine modified ant clustering algorithm and PD-based partitioning method (PACA). The algorithm PACA is illustrated in Fig. 3.

The PACA algorithm first uses modified ant clustering algorithm to present multi-dimensional data on a two-dimensional grid, then PD-based partitioning method is employed to calculate and partition the objects. Because this partitioning method need not consider each dimension respectively as usually done, so it save much time and improve the accuracy.

5.3. The new hybrid method based on partitioning-based DBSCAN and ant clustering (PACA-DBSCAN)

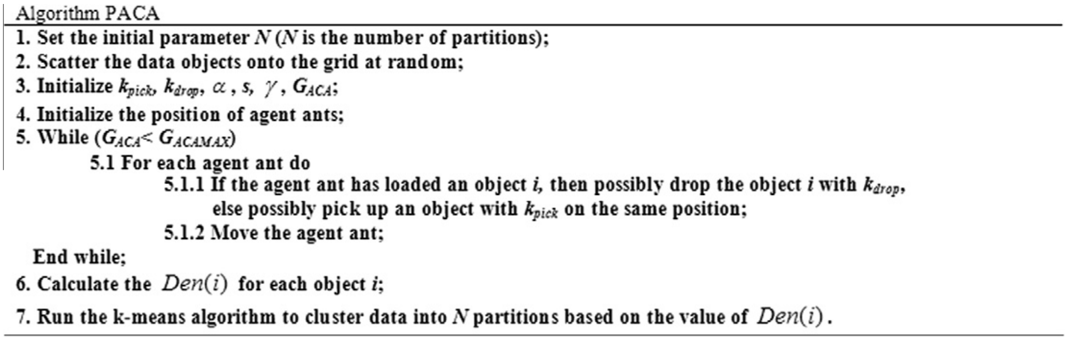
Sections 5.1 and 5.2 have presented the PD-based partitioning method and PACA partitioning algorithm. These two methods can divide data points with similar density into the same area. Our new PACA-DBSCAN algorithm will employ one of these two partitioning methods according to the number of data dimension. If the data is two-dimensional, the algorithm uses PD-based partitioning method to partition data directly. Else if the data is multi-dimensional, the algorithm will partition data with PACA algorithm. Then for each partition, PACA-DBSCAN algorithm builds R*-tree, plots k-dist graph and runs DBSCAN algorithm. At last, the partial clusters will be merged based on predetermined rules. The PACA-DBSCAN algorithm makes a better use of the advantages of both PD-based partitioning and PACA partitioning. The detail of the proposed PACA-DBSCAN algorithm is explained in Fig. 4.

6. Experimental studies

In order to test the effectiveness of our PACA-DBSCAN algorithm, we used five data sets. All the experiments were performed on a Pentium (R) CPU 2.50 GHZ with 512 MB RAM. We have coded with C++. All data sets were run with the DBSCAN, PDBSCAN and PACA-DBSCAN algorithms. The initialization of the parameters used in PACA-DBSCAN algorithm is summarized in Table 1. The experimental results are evaluated and compared by two criteria.

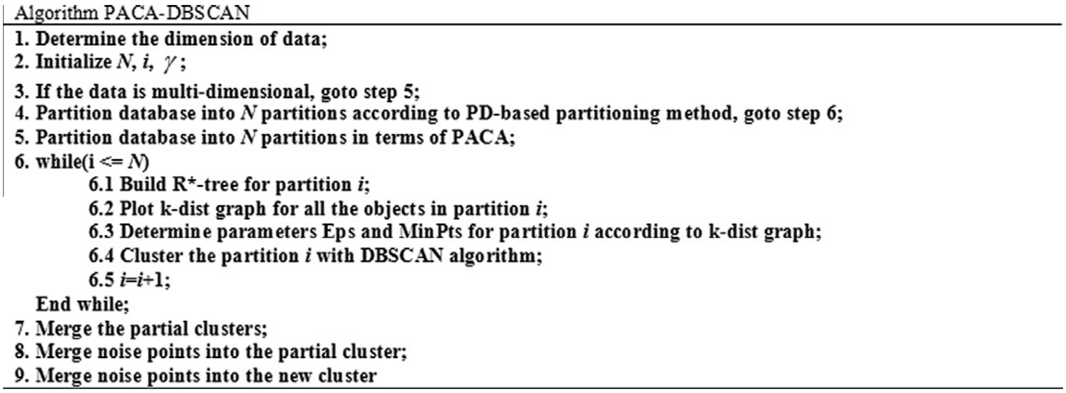
6.1. Data sets

The data sets are Artset1, Artset2, Artset3, Iris and Wine. They are summarized in Table 2. The Artset1, Artset2 and Artset3 are dummy data sets. Iris and Wine are two well-known data sets available at ftp://ftp.ics.uci.edu/pub/machine-learning-databases/.



```
Algorithm PACA
1. Set the initial parameter $N$ ($N$ is the number of partitions);
2. Scatter the data objects onto the grid at random;
3. Initialize $k_{pick}$, $k_{drop}$, $\alpha$, s, $\gamma$, $G_{ACA}$;
4. Initialize the position of agent ants;
5. While ($G_{ACA} < G_{ACAMAX}$)
        5.1 For each agent ant do
                5.1.1 If the agent ant has loaded an object i, then possibly drop the object i with $k_{drop}$,
                      else possibly pick up an object with $k_{pick}$ on the same position;
                5.1.2 Move the agent ant;
  End while;
6. Calculate the $Den(i)$ for each object i;
7. Run the k-means algorithm to cluster data into $N$ partitions based on the value of $Den(i)$.
```

Fig. 3. The process of the PACA algorithm.



```
Algorithm PACA-DBSCAN
1. Determine the dimension of data;
2. Initialize $N$, i, $\gamma$;
3. If the data is multi-dimensional, goto step 5;
4. Partition database into $N$ partitions according to PD-based partitioning method, goto step 6;
5. Partition database into $N$ partitions in terms of PACA;
6. while(i <= $N$)
        6.1 Build R*-tree for partition i;
        6.2 Plot k-dist graph for all the objects in partition i;
        6.3 Determine parameters Eps and MinPts for partition i according to k-dist graph;
        6.4 Cluster the partition i with DBSCAN algorithm;
        6.5 i=i+1;
  End while;
7. Merge the partial clusters;
8. Merge noise points into the partial cluster;
9. Merge noise points into the new cluster
```

Fig. 4. The process of the PACA-DBSCAN algorithm.

Table 1
The initialization of the parameters used in the PACA-DBSCAN algorithm.

Parameter	Value
kpick	0.15
kdrop	0.15
s	5
γ	4
α	4

Table 2
Characteristics of data sets considered.

Data set	No. of data classes	No. of features	Size of data set
Artset1	3	2	300(257,32,211)
Artset2	4	2	1572(528,348,272,424)
Artset3	7	2	1043(343,30,38,241,72,157,155)
Iris	3	4	150(50,50,50)
Wine	3	13	178(59,71,48)

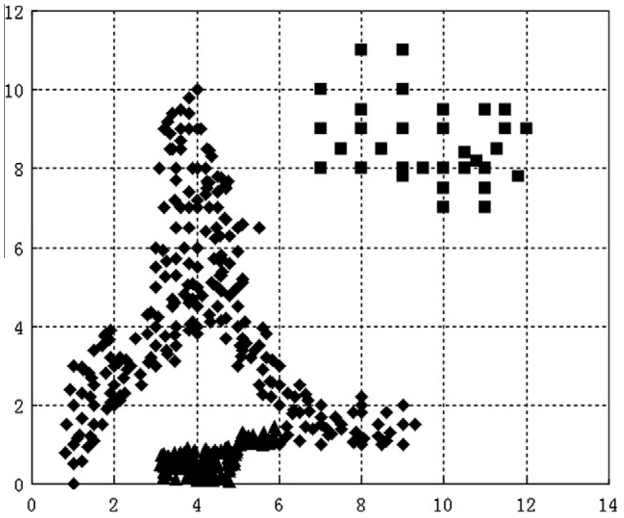


Fig. 5. The artificial data set: Artset1.

All the five data sets are described as follows:

(1) Artset1 ($n = 500$, $d = 2$, $k = 3$), this is an artificial data set. It is a two-featured problem with three unique classes and 500 patterns. In order to prove the superiority of the new algorithm that it can be applied to non-uniform density of data and less sensitive to the input parameters, the shapes of three clusters is irregular. The data set is illustrated in Fig. 5.

(2) Artset2 ($n = 1572$, $d = 2$, $k = 4$), this is an artificial data set. It is a two-featured problem with four classes and 1572 patterns. These data is obtained using a data generator for multivariate Gaussian clusters. It can prove that the result of PACA-DBSCAN is better than PDBSCAN and DBSCAN on standard Gaussian data set. The data set is illustrated in Fig. 6.

(3) Artset3 ($n = 1043$, $d = 2$, $k = 7$), this is an artificial data set. It is a two-featured problem with four classes and 1043 patterns. These data are obtained using a data generator. There are two kinds of shapes in Artset3. One is spiral, and another is annular. The data set also includes some noise points. We create it to show that PACA-DBSCAN can overcome the drawback of PDBSCAN which cannot recognize the special shape of data.

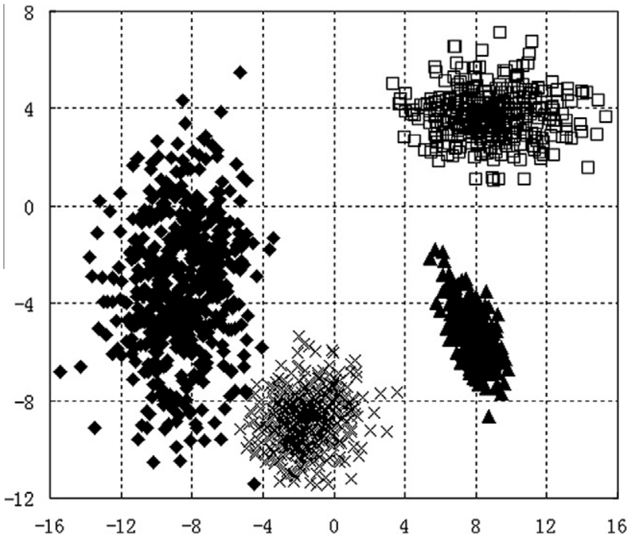


Fig. 6. The artificial data set: Artset2.

(4) Iris ($n = 150$, $d = 4$, $k = 3$), it is a well-known data set from machine learning databases which consists of three different species of iris flower: Iris Setosa, Iris Versicol-our and Iris Virginica. For each species, 50 samples with four features (sepal length, sepal width, petal length, and petal width) were collected. The purpose of choosing this data set is to prove that PACA-DBSCAN has better results when the data is multi-dimensional.

(5) Wine ($n = 178$, $d = 13$, $k = 3$), it is also a famous machine learning database which is the result of a chemical analysis of wines grown in a region in Italy but derived from three different cultivars. The data set consists of 178 objects each with 13 continuous attributes.

6.2. Experimental results

In our experiment, five data sets are normalized according to the Eq. (1) and used with the DBSCAN, PDBSCAN and PACA-DBSCAN algorithms.

$$x = \frac{x - \min(x)}{\max(x) - \min(x)}. \tag{1}$$

We compare the performances of the DBSCAN, PDBSCAN and PACA-DBSCAN algorithms in terms of classical F-Measure and a proposed criterion that we call it 'Eps rate' (ER):

(1) The F-Measure: it is based the precision and the recall from the information retrieval (Dalli, 2003; Handl et al., 2003). The precision and the recall are defined as:

$$p(i,j) = \frac{n_{ij}}{n_j}, \quad r(i,j) = \frac{n_{ij}}{n_i},$$

where each class i (as given by the class labels of the used data set) is regarded as the set of n_i items desired for a query, and each cluster j(generated by the algorithm) is regarded as the set of n_j items retrieved for a query. n_{ij} is the number of data points of the class i within cluster j. For a class i and a cluster j, the F-Measure is defined as:

$$F(i,j) = \frac{(b^2 + 1) \cdot p(i,j) \cdot r(i,j)}{b^2 \cdot p(i,j) + r(i,j)},$$

where we set $b = 1$ to obtain equal weighting for $p(i,j)$ and $r(i,j)$. The overall F-Measure for the data set of size n is given by

$$F = \sum_i \frac{n_i}{n} \max_j \{F(i,j)\}.$$

So, the bigger the F-Measure is, the better the clustering algorithm is.

(2) Eps rate (ER): it shows the sensitivity of DBSCAN algorithm to the parameter Eps.
For each subset of data, $ER_i = \frac{\text{the scope of Eps corresponding the optimal results}}{\text{the scope of Eps of partition i}}$,
For the whole data set, $ER = \frac{\sum_{i=1}^{N} ER_i}{N}$, where N is the number of partitions.

The bigger the ER is, the less sensitive to Eps the algorithm is.

DBSCAN, PDBSCAN and PACA-DBSCAN algorithms need two parameters Eps and MinPts. We set the value of MinPts 4. In order to set Eps, we need to build R^*-tree, and plot 4-dist graph. DBSCAN and PDBSCAN algorithm is sensitive to the parameter Eps. Once the Eps change a little, the results of the algorithm will change a lot. In

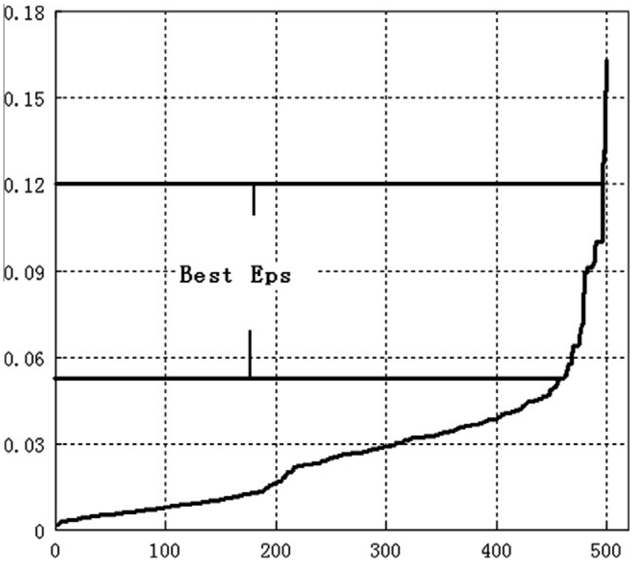


Fig. 7. The 4-dist of Artset1.

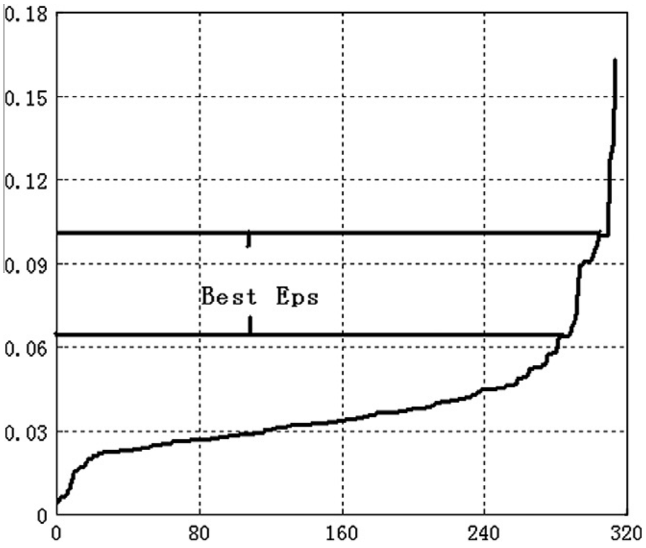


Fig. 8. The 4-dist of the first partition of Artset1.

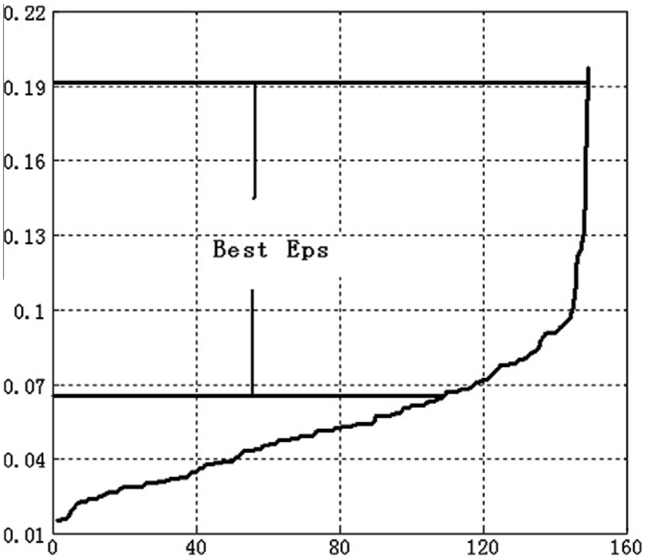


Fig. 9. The 4-dist of the second partition of Artset1.

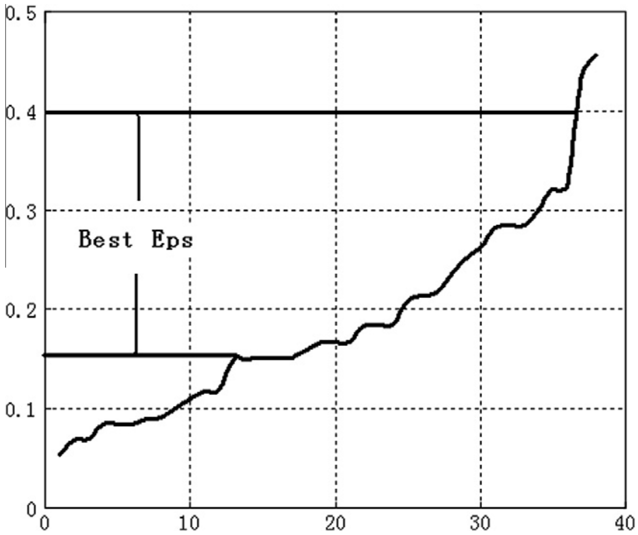


Fig. 10. The 4-dist of the third partition of Artset1.

this paper, we compare the sensitivity to Eps of three algorithms. Fig. 7 is the 4-dist graph of artificial data set Artset1 which have not been partitioned ($ER_{Artset1}$ = (0.12–0.05)/(0.163–0.02) = 0.07/0.143 = 0.4895). PACA-DBSCAN partitions Artset1 into three partitions: Artset1–1, Artset1–2 and Artset1–3. The 4-dist graph of the three partitions is respectively illustrated in Figs. 8–10. The ER of every partition of Artset1 and the total ER of Artset1 are summarized in Table 3. Table 4 lists the ER of multi-dimensional data set Wine.

The Tables 3 and 4 show that the ER with PACA-DBSCAN algorithm is greater than the ER with DBSCAN algorithm. So, the new PACA-DBSCAN algorithm can reduce the sensitivity to the input parameter even if the data is multi-dimensional.

We run all data sets with the DBSCAN, PDBSCAN and PACA-DBSCAN algorithms. Table 5 summarizes the results of DBSCAN, PDBSCAN and PACA-DBSCAN algorithms on five data sets. The quality of clustering is evaluated using the *F*-Measure and ER. Bold and italic face indicates the best result out of the three algorithms.

From Table 5 we can see clearly that, for the artificial data sets of Artset1, Artset2 and Artset3, the average values of *F*-Measure and ER of the PACA-DBSCAN algorithm are better than the values of DBSCAN and PDBSCAN. For the data sets of Iris and Wine, the values of the *F*-Measure and ER of PACA-DBSCAN are also higher than the other two algorithms.

Table 5
Results of DBSCAN, PDBSCAN and PACA-DBSCAN algorithms according to *F*-Measure and ER on five data sets.

Data set	DBSCAN	PDBSCAN	PACA-DBSCAN
Artset1			
F-Measure	0.689	0.689	0.996
ER	0.49	0.497	0.503
Artset2			
F-Measure	0.99	1.00	1.00
ER	0.002	0.291	0.396
Artset3			
F-Measure	0.551	0.608	0.962
ER	0.114	0.216	0.425
Iris			
F-Measure	0.772	0.978	0.981
ER	0.234	0.248	0.154
Wine			
F-Measure	0.625	0.442	0.654
ER	0.016	0.003	0.021

Table 3
The ER of artificial data set Artset1.

Algorithm	Data set	The scope of Eps corresponding the optimal results	The total scope of Eps	The ER of each area	The final ER
DBSCAN	Artset1	[0.05–0.12]	[0.02–0.163]	0.49	0.49
PACA-DBSCAN	Artset1–1	[0.065–0.1]	[0.005–0.163]	0.22	
	Artset1–2	[0.065–0.191]	[0.015–0.197]	0.692	
	Artset1–3	[0.16–0.4]	[0.054–0.455]	0.599	0.503

Table 4
The ER of data set Wine.

Algorithm	Data set	The scope of Eps corresponding the optimal results	The total scope of Eps	The ER of each area	The final ER
DBSCAN	Wine	[0.48–0.49]	[0.306–0.938]	0.016	0.016
PACA-DBSCAN	Wine-1	[0.5–0.52]	[0.391–1.072]	0.029	
	Wine-2	[0.5 –0.51]	[0.382 –1.147]	0.013	0.021

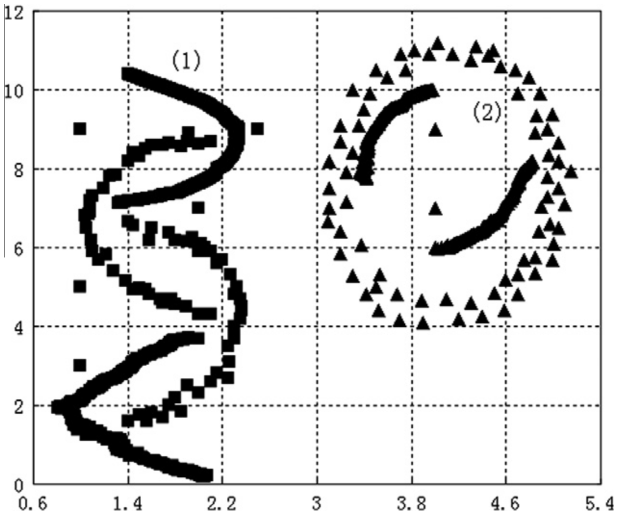


Fig. 11. The result of Artset3 with DBSCAN.

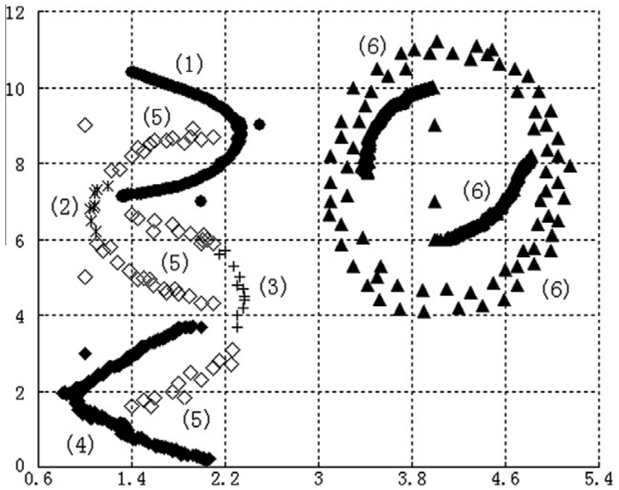


Fig. 12. The result of Artset3 with PDBSCAN.

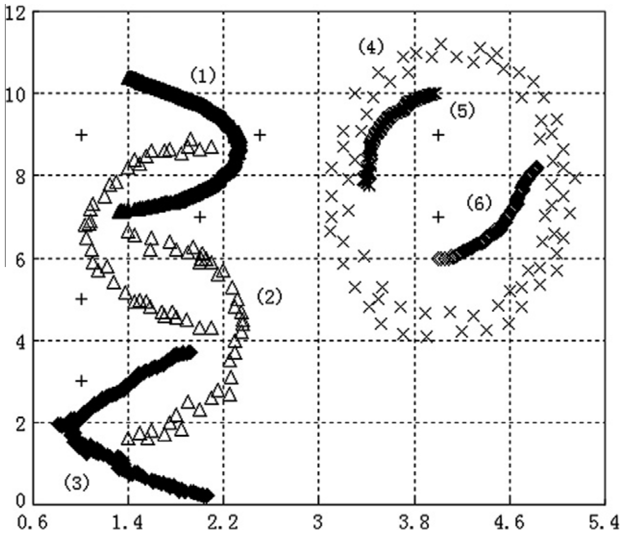


Fig. 13. The result of Artset3 with PACA-DBSCAN.

The results of Artset3 with DBSCAN, PDBSCAN, PACA-DBSCAN are illustrated in Figs. 11–13. DBSCAN only cluster Artset3 into two clusters. Fig. 12 shows that PDBSCAN cannot deal with data of special shape very well. It merges the three clusters together on the right side, and divides the data on the left side into five clusters. Fig. 13 shows that the Artset3 can be clustered more correctly by PACA-DBSCAN than the other two algorithms.

7. Conclusions

In this paper we present a new hybrid algorithm (PACA-DBSCAN) based on partitioning-based DBSCAN and modified ant clustering algorithms. It can partition database into *N* partitions according to the density of data, then cluster each partition with DBSCAN. Superior to DBSCAN and PDBSCAN, The new hybrid algorithm reduces the sensitivity to the initial parameters, and can deal with data of uneven density very well. For multi-dimensional data, The PACA-DBSCAN algorithm does not need to discuss the distribution of data on each dimension. In contrast with PDBSCAN, The PACA-DBSCAN can correctly cluster data of very special shape. We employ five data sets to prove the performance of our new proposed algorithm. The result of PACA-DBSCAN are evaluated and compared by the classical *F*-Measure and a proposed criterion (ER). The experiment has proved that the performance of PACA-DBSCAN is better than DBSCAN and PDBSCAN.

In the future, we intend to improve the performance of ant clustering algorithm. And we are going to research how to specify the parameters Eps and MinPts quickly and exactly.

References

Birant, D., & Kut (2007). ST-DBSCAN: An algorithm for clustering spatial-temporal data. *Data and Knowledge Engineering, 60*, 208–221.

Cao, F., Ester, M., Qian, W., & Zhou, A. (2006). Desity-based clustering over an evolving data stream with noise. In *2006 SIAM conference on data mining, Bethesda* (pp. 328–339).

Chu, S., & Roddick, J. (2003). A clustering algorithm using Tabu search approach with simulated annealing for vector quantization. *Chinese Journal Electronics, 12*(3), 349–353.

Comaniciu, D., & Meer, P. (1999). Distribution free decomposition of multivariate data. *Pattern Analysis, 2*, 22–30.

Dalli, A. (2003). Adaptation of the *F*-measure to cluster-based Lexicon quality evaluation. In *EACL 2003*. Budapest.

Deneubourg, J.-L., Goss, S., Franks, N., Sendova-Franks, A., Detrain, C., & Chretien, L. (1991). The dynamics of collective sorting: robot-like ants and ant-like robots. *Proceedings of the First International Conference On Simulation of Adaptive Behavior*, 356–365.

Ester, M., Kriegel, H.P., Sander, J., & Xu, X. (1996). A density-based algorithm for discovering clusters in large spatial databases with noise. In *Proceedings of 2nd International Conference on Knowledge Discovery and Data Mining. Porland; Oregon*.

Gan, G.J., Ma, C.Q. & Wu, J.H. (2007). Data Clustering Theory, Algorithms, and Applications, ASA-SIAM Series on Statistics and Applied Probability, SIAM, Philadelphia, ASA, Alexandria, VA, (Chapter 7).

Güngör, Z., & Ünler, A. (2007). *K*-harmonic means data clustering with simulated annealing heuristic. *Applied Mathmatics and Computation, 184*, 199–209.

Güngör, Z., & Ünler, A. (2008). *K*-harmonic means data clustering with tabu-search method. *Applied Mathematical Modelling, 32*, 1115–1125.

Hammerly, G., & Elkan, C. (2002). Alternatives to the *k*-means algorithm that find better clusterings.In: *Proceedings of the 11th international conference on information and knowledge management* pp. 600–607.

Handl, J., Knowles, J., & Dorigo, M. (2003). On the performance of ant-based clustering. *Design and Application of Hybrid Intelligent Systems. Frontiers in Artificial Intelligence and Applications, 104*, 204–213.

Handl, J., & Meyer, B. (2007). Ant-based and swarm-based clustering. *Swarm Intelligince, 1*, 95–113.

Huang, C. H., Pan, J. S., Lu, Z. H., Sun, S. H., & Hang, H. M. (2001). Vector quantization based on genetic simulated annealing. *Signal Processing, 81*, 1513–1523.

Jain, A. K., Murty, M. N., & Flynn, P. J. (1999). Data clustering: a review. *ACM Computing Surveys, 31*(3), 264–323.

Kao, Y. T., Zahara, E., & Kao, I. W. (2008). A hybridized approach to data clustering. *Expert Systems with Applications, 34*, 1754–1762.

Lumer, E. D., & Faieta, B. (1994). Diversity and Adaptation in Populations of Clustering Ants. *Proceedings of the Third International Conference on the Simulation of Adaptive Behavior, 3*, 499–508.

Sun, J. G., Liu, J., & Zhao, L. Y. (2008). Clustering algorithms research. *Journal of Software, 19*(1), 48–61.

Thanh, N. Tran, Wehrens, Ron, & Lutgarde, M. C. (2006). Buydens KNN-Kernel density-based clustering for high-dimensional multivariate data. *Computational Statistics and Data Analysis, 51*, 513–525.

Viswanath, P., & Pinkesh, R. (2006). l-DBSCAN: A Fast Hybrid Density Based Clustering Method. *Pattern Recognition, 1*, 912–915.

Vizine, A. L., & de Castro, L. N. (2005). Towards Improving Clustering Ants: An Adaptive Ant clustering algorithm. *Informatica, 29*, 143–154.

Xu, R., & Wunshc, D. (2005). Survey of Clustering Algorithms. *IEEE Transactions on Neural Networks, 16*, 645–678.

Yang, F. Q., & Sun, T. L. (2009). Particle swarm optimization based *K*-harmonic means data clustering. *Expert Systems with Applications, 36*, 9847–9852.

Zhou, A., Zhou, S. G., Cao, J., Fan, Y., & Hu, Y. F. (2000). Approaches for Scaling DBSCAN Algorithm to Large Spatial Databases. *Journal of Computer Science and Technology, 15*(6).