Applied Soft Computing 12 (2012) 2387–2393

Contents lists available at SciVerse ScienceDirect

Applied Soft Computing

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

Chaotic ant swarm approach for data clustering

Miao Wan[a,b,c,*], Cong Wang[a,c], Lixiang Li[a,c], Yixian Yang[a,b,c]

[a] Information Security Center, Beijing University of Posts and Telecommunications, P.O. Box 145, Beijing 100876, China
[b] Research Center on Fictitious Economy and Data Science, Chinese Academy of Sciences, Beijing 100190, China
[c] National Engineering Laboratory for Disaster Backup and Recovery, Beijing University of Posts and Telecommunications, Beijing 100876, China

ARTICLE INFO

Article history:
Received 20 February 2010
Received in revised form 10 March 2011
Accepted 18 March 2012
Available online 5 April 2012

Keywords:
Data mining
Data clustering
Chaotic ant swarmoptimization
Optimization based clustering

ABSTRACT

Clustering divides data into meaningful or useful groups (clusters) without any prior knowledge. It is a key technique in data mining and has become an important issue in many fields. This article presents a new clustering algorithm based on the mechanism analysis of chaotic ant swarm (CAS). It is an optimization methodology for clustering problem which aims to obtain global optimal assignment by minimizing the objective function. The proposed algorithm combines three advantages into one: finding global optimal solution to the objective function, not sensitive to clusters with different size and density and suitable to multi-dimensional data sets. The quality of this approach is evaluated on several well-known benchmark data sets. Compared with the popular clustering method named *k*-means algorithm and the PSO-based clustering technique, experimental results show that our algorithm is an effective clustering technique and can be used to handle data sets with complex cluster sizes, densities and multiple dimensions.

1. Introduction

Clustering is the unsupervised classification of patterns (observations, data items, or feature vectors) into groups (clusters) [1]. In the past fifty years, many attentions have been focused on the problem of clustering from the theoretical and the practical point of view. Such problem has been addressed in diverse areas such as pattern recognition, data analysis, image processing, economic science (especially market research) and biology. So the study about new clustering algorithms is an important issue in the research fields including data mining, machine learning, statistics, and biology.

In recent years, different clustering algorithms have been proposed, such as partitioning [14,16], hierarchical [5], density-based [7], grid-based [23] and model-based [2]. Partitioning approach constructs different partitions based on a certain criterion. For hard partitional clustering, each pattern belongs to one and only one cluster. Fuzzy clustering [9,10] extends this notion that each pattern may belong to all clusters with a degree of membership. Apart from the above techniques, kernel *k*-means and spectral clustering have both been used to identify clusters that are non-linearly separable in input space [11–13].

k-means algorithm [14] is the most popular approach because of its simplicity, efficiency and low cost of computation. However, since criterion functions for clustering are usually non-convex and nonlinear, traditional approaches, especially the *k*-means algorithm, is sensitive to initializations and easy to be trapped in local optimal solutions [4]. As the increasing numbers and dimensions of data sets, finding solutions to the criterion functions has become an NP-hard problem. Since the importance of clustering strategies in many fields, global optimization methods, such as genetic algorithms (GA), ant colony optimization (ACO) and particle swarm optimization (PSO), have been applied to solve clustering problems [8,6,24,15]. When solving clustering problems, these algorithms start from an initial population or position and explore the solution space through a number of iterations to reach a near optimal solution.

Swarm Intelligence (SI) is an innovative distributed intelligent paradigm for solving optimization problems that originally took its inspiration from the biological examples by swarming, flocking and herding phenomena in vertebrates [6]. Chaotic ant swarm (CAS) [17] is an optimization algorithm inspired by chaotic behavior of ant swarm which has been applied in several fields [18,20,19]. However, there is few application of CAS in data clustering. Moreover, as a latest optimization methodology, mathematical modelling, modification, and adaptation of the algorithm might be a major part of the research on CAS in future.

In this paper, we propose a clustering algorithm based on the principles of chaotic ant swarm search method and build a new optimization model for discovering clusters, which is a chaotic optimization version to solve clustering problems. In our algorithm, no centroid or center needs to be selected in the initial step. Meanwhile, in order to overcome the drawbacks of traditional algorithms, the proposed algorithm combines the following three

* Corresponding author at: Information Security Center, Beijing University of Posts and Telecommunications, P.O. Box 145, Beijing 100876, China.
E-mail address: wanmiao120@163.com (M. Wan).

1568-4946/$ – see front matter © 2012 Elsevier B.V. All rights reserved.
doi:10.1016/j.asoc.2012.03.037

advantages into one: (1) Find a global optimum clustering result; (2) Have a good algorithm performance for high-dimensional data; (3) Not sensitive to clusters with different size and density.

2. Background

2.1. Optimization based clustering

Clustering is a data mining technique which classifies objects into groups (clusters) without any prior knowledge. The problem of common clustering can be formally started as follows. Given a sample data set $X = \{x_1, x_2, \ldots, x_n\}$, determine a partition of the objects into K clusters $C_1, C_2, \ldots, C_K$ which satisfies:

$$\begin{cases} \bigcup_{i=1}^{K} C_i = X; \\ C_i \bigcap C_j = \emptyset, \quad i, j = 1, 2, \ldots, K; \quad i \neq j; \\ C_i \neq \emptyset, \quad i = 1, 2, \ldots, K. \end{cases} \tag{1}$$

In the viewpoint of mathematics, cluster C_i can be determined by:

$$\begin{cases} C_i = \{x_j \mid \|x_j - z_i\| \leq \|x_j - z_p\|, \quad x_j \in X\}, p \neq i, \ p = 1, 2, \ldots, K, \\ z_i = \dfrac{1}{\mid C_i \mid} \displaystyle\sum_{x_j \in C_i} x_j, \quad i = 1, 2, \ldots, K, \end{cases} \tag{2}$$

where $\|\cdot\|$ denotes the distance of any two data points in the sample set. z_i is the center of cluster C_i, which is represented by the average(mean) of all the points in the cluster.

We can see from Eq. (2) that C_i is composed by some data items nearest to z_i. So the task of clustering can be seen as a process of determining k centers of $\{C_1, C_2, \ldots, C_K\}$.

A clustering criterion must be adopted. The most commonly used criterion in clustering task is the sum of squared error (SSE) [26]:

$$\mathrm{SSE} = \sum_{i=1}^{K} \sum_{x_j \in C_i} \|x_j - z_i\|^2. \tag{3}$$

For each data in the given set, the error is the distance to the nearest cluster. The general objective of clustering is to obtain that partition which, for fixed number of clusters, minimizes the square-error.

Thus, the clustering problem is converted to a process of searching K centers $z_1, z_2, \ldots, z_K$, which can minimize the sum of distance between all the sample data x_i and its closest center. This could be considered as a function optimization issue with the objective function as SSE.

2.2. Chaotic ant swarm (CAS) optimization

Social insects with self-organizing behavior, for example, ants, have attained the attention of many scientists and researchers. The colony of these insects can achieve high level of structure and success of foraging activities while individuals in the colony only take simple tasks and act aperiodically. Existing ant-inspired optimization algorithms are mainly based on the random meta-heuristic of nondeterministic probability theory. However, Cole has pointed out that ant colony exhibits a periodic behavior while single ant shows low-dimensional deterministic chaotic activity patterns [21]. Moreover, the problem of how the chaotic behaviour of single ant relates to the self-organization and foraging behaviours of the ant colony has received little attention. From the perspective of dynamics, there are interactions between the two kinds of behaviors. These interactions help ants to find food and survive, which can be adapted to the solution of optimization problems. Consequently, inspired by the chaotic and self-organization behaviours of ants, chaotic ant swarm (CAS) [17] was developed to solve the optimization problems, which incorporated chaotic dynamics of ant, swarm organization and optimization principles.

In CAS, an ant colony composed of M ants is considered. These ants are located in a D-dimensional search space S and they try to minimize a function J. The ant colony undergoes two successive phases, chaotic phase and organization phase. To achieve self-organization from chaotic state, a successively decrement of organization variable y_i is introduced into CAS. The influence of the organization variable on the ant's behaviour is very weak in the first process and the behaviour of single ant is chaotic. The motion of the ants approximately are governed by the following equation:

$$x_{id}(t) = x_{id}(t-1) e^{3 - \psi_d x_{id}(t-1)}. \tag{4}$$

Eq. (4) is a chaotic map suggested by Solé [22]. With the continual small change of y_i evolving in time, the influence of the organization on the behaviour of individual ant becomes stronger and stronger. When the effect of the organization is sufficiently large, the chaotic behaviour of the individual ant disappears. Then ants will do some further searches and move to the position that they can find in the search space. Throughout the whole process, they exchange information with their neighbors continually. Mathematically, the changing process of position for ant i can be described as [17]:

$$\begin{cases} y_i(t) = y_i(t-1)^{(1+r_i)}, \\ x_{id}(t) = (x_{id}(t-1) + V_{id}) e^{(1 - e^{-a y_i(t)})(3 - \psi_d (x_{id}(t-1) + V_{id}))} \\ \qquad + (pbest_{id}(t-1) - x_{id}(t-1)) e^{(-2a y_i(t) + b)} - V_{id}, \end{cases} \tag{5}$$

where

1. t means the current iteration step, and $t-1$ is the previous iteration step;
2. $y_i(t)$ is the ith ant's organization variable of the current iteration step, $y_i(0) = 0.999$;
3. $x_{id}(t)$ is the current state of the dth dimension of ant i;
4. $pbest_{id}(t-1)$ is the best position found by the ith ant and its neighbors within $t-1$ steps;
5. $V_i (0 < V_{id} < 1)$ determines the search region of ant i;
6. a is a sufficiently large positive constant and can be selected as $a = 2000$;
7. b is a constant, $0 \leq b \leq 2/3$.

In addition, r_i and ψ_d are two important parameters. r_i is the organization factor of ant i, which affects the convergence speed of the CAS directly. The larger r_i is, the faster the system converges, and the shorter the runtime is. The format of r_i can be designed according to concrete problems and runtime. Each ant could have different r_i, such as $r_i = 0.1 + 0.2 \times rand(1)$. ψ_d affects the search ranges of the CAS. If the interval of the search is $[-\omega_d/2, \omega_d/2]$, we can obtain an approximate formula $\omega_d \approx 7.5/\psi_d$, and $V_{id} = \omega_d/2 (0 < V_{id} < 1)$ will make the search interval shift to $[0, \omega_d]$. The impacts to the optimization result by adjusting each parameter in Eq. (5) are fully discussed in [17].

The procedure of CAS algorithm can be illustrated in Fig. 1.

3. Proposed methodology: the CAS-based clustering algorithm (CAS-C)

In this section we will give the formal mathematical model of data clustering and express how CAS optimization solves general clustering problem in detail.

As data clustering can be seen as an optimization problem of seeking a global optimal solution to Eq. (3), in this paper we present

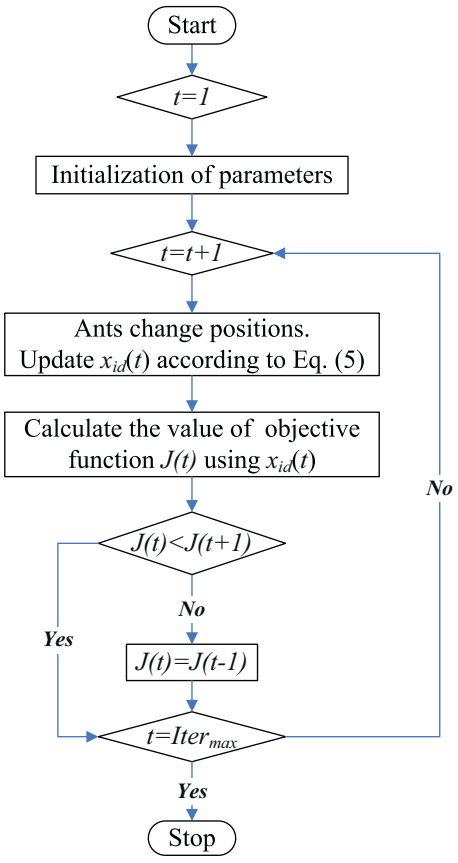

Fig. 1. Working flow of CAS algorithm.

a CAS based clustering approach, called CAS-C algorithm, to solve this problem. In CAS-C, clustering is regarded as a process of ant foraging, and the centers or centroid can be seen as the goal (food) to search.

Unlike the partitional clustering techniques, there is no initial partition selected in our algorithm. In the initial step, several data in the sample set are randomly picked as the positions of the ants. After steps of iteration, the ants move and converge to some points that are considered as centers of each cluster in the data space. Generally, there are two ways to stop the iteration of optimization-based algorithm. A maximum number of iterations can be specified by experience to prevent endless oscillation. The other way is calculating the value of the objective function to find a converging state when all the patterns do not change between two successive iterations. We choose the former way throughout this paper and preset $Iter_{max}$ as a number of maximum steps of iteration. When the algorithm arrives to $Iter_{max}$, the clustering process will stop with the results output.

Based on Eq. (5), the equation for CAS-C algorithm is given as

$$\begin{cases} y_i(t) = y_i(t-1)^{(1+r_i)}, \\ z_{pid}(t) = (z_{pid}(t-1) + V_{id})e^{(1-e^{-ay_i(t)})(3-\psi_d(z_{pid}(t-1)+V_{id}))} \\ \qquad +(zbest_{pid}(t-1) - z_{pid}(t-1))e^{(-2ay_i(t)+b)} - V_{id}, \end{cases} \quad (6)$$

where

1. t means the current iteration step, and $t-1$ is the previous iteration step;
2. $z_{pid}(t)$ is the current state of the dth dimension ($d = 1, 2, \ldots, D$) of ant i for the pth desired center z_p ($p = 1, 2, \ldots, K$, where K is the desired cluster number and needed to be pre-assigned before the algorithm starts);
3. $zbest_{pid}(t-1)$ presents the best position of the dth dimension found by all the ants within $(t-1)$ steps for ant i;
4. Other parameters have the same meaning with Eq. (5).

Now we will give a full explain about how CAS-C algorithm is implemented. Algorithm 3.1 introduces the procedure of proposed CAS-C algorithm. Given the desired cluster number K, the CAS-C algorithm is carried out in the following steps:

1. Initialization. There are several parameters to be preassigned before iteration starts in CAS-C. Initialize ψ_d for scope of searching in the data space, the ant number M, the maximum number of generations $Iter_{max}$ and the organization factor r_i. Then set $t = 1$ and generate positions of $M \times K$ ants randomly from the data space for each center (line 1 in Algorithm 3.1).
2. Iteration process. At the tth step, the best position found by all ants within $(t-1)$ steps is picked out as $zbest_{pi}(t-1)$ for ant i. Then every individual ant changes their places in the data space as Eq. (6). After moving, select proper $zbest_{pi}(t)$ for each ant based on minimizing the square-error in Eq. (3), and remember it to the next iteration. The iteration process is terminated and goes to Step 4 when t equals to the maximum iteration step $Iter_{max}$ (lines 2-14 in Algorithm 3.1).
3. Mark final centers. All the ants will converge to the global optimal points in the search space after the iteration process. The final positions of the ants are considered as the required centers(lines 15-16 in Algorithm 3.1).
4. Perform clustering on the data set. Allocate all the data according to Eq. (2) into different clusters which are represented by the final centers gained after the iteration process. Mark every data object with its corresponding label (lines 18-25 in Algorithm 3.1).

Algorithm 3.1 *(The CAS-C algorithm).*

Require:
Data set, $X = \{x_1, x_2, \ldots, x_n\}$;
Cluster number, K.
Ensure:
Clusters: $\{C_1, C_2, \ldots, C_K\}$.
1: Initialize the search scope ψ, organization factor r and position z of M ants randomly, in which each single ant z_i $(i = 1, 2, \ldots, K)$ contains K randomly generated centroid vectors: $z_i = \{z_{i1}, z_{i2}, \ldots, z_{iK}\}$.
2: **for** $t = 1 : Iter_{max}$ **do**
3: **for** $i = 1 : M$ **do**
4: Calculate the objective function $J(i, t)$ with current $z_i(t)$
5: $J_{last} = J(i, t-1)$
6: $y_i(t) = y_i(t-1)^{(}1 + r_i)$,
7: $z_i(t) = (z_i(t-1) + V_i)e^{(1-e^{-ay_i(t)})(3-\psi(z_i(t-1)+V_i))} +$ $(zbest_i(t-1) - z_i(t-1))e^{(-2ay_i(t)+b)} - V_i$
8: Calculate $J(i, j+1)$ with current $z_i(j+1)$ according to Eq. (3)
9: **if** $J(i, t) < J_{last}$ **then**
10: $zbest_i(t) = z_i(t)$ //$zbest_i$ represents the local best position, the best position found so far for ant i.
11: **else**
12: $zbest_i(t) = zbest_i(t-1)$
13: **end if**
14: **end for**
15: Update the global best position $zbest_g$: Select the best $zbest_i$ from $\{zbest_1, zbest_2, \ldots, zbest_M\}$ as $zbest_g$. //$zbest_g$ represents the global best position in the neighborhood of each ant.
16: $\{z_1, z_2, \ldots, z_K\} = zbest_g$
17: **end for**
18: **for** $j = 1 : n$ **do**
19: **for** $c = 1 : K$ **do**
20: Calculate distance $d_c = \|x_j - z_c\|$
21: **end for**
22: $d = \{d_1, d_2, \ldots, d_K\}$
23: Find the position p of $min(d)$
24: C_p.add(x_j)
25: **end for**

4. Cluster validity

The prudent and enlightened validation of clustering results is the essential step that changes a qualitative analysis into hard evidence [25]. The clustering results will be presented by comparison with two previous clustering algorithms using four different evaluation measures.

4.1. Methods for comparison

For presenting the priority of the proposed CAS-C algorithm, we select some previous clustering techniques for algorithm comparisons.

Firstly we choose the *k*-means algorithm [14] as a method to be compared because it is the most famous conventional clustering technique. The *k*-means algorithm is a partition-based clustering approach and has been widely applied for decades of years.

Moreover, as an swarm-based methodology, the CAS-C algorithm will be compared with the PSO-based clustering technique. Particle swarm optimization (PSO) [27] is a formerly proposed swarm-based algorithm which simulates bud flocking or fish schooling behavior to achieve a self-evolution system. PSO compares favorably with many global optimization algorithms like genetic algorithms (GA) [28], simulated annealing (SA) [29] and other global optimization algorithms. The clustering approach using PSO can search automatically the data centers of K groups data set by optimizing an objective function.

4.2. Evaluation functions

In order to evaluate the performances of CAS-C and its comparison algorithms, in this article we use four measure functions to validate the convergence and clustering quality.

4.2.1. Sum of squared error (SSE)

SSE is the common criteria of evaluating clustering results which sums the squared error of each data together. It is also taken as the fitness function of both CAS-C and PSO-clustering algorithms. The objective function will be measured in each single iteration of CAS-C and PSO techniques and should be calculated as:

$$\mathrm{SSE} = \sum_{i=1}^{k} \sum_{x_j \in C_i} \|x_j - z_i\|^2, \tag{7}$$

It is easy to see that Eq. (7) tries to make the results of clustering more compact and independent.

All the three methods used in this paper aim to minimize SSE during their iterations. Thus, the smaller the final value of SSE is, the better the optimization algorithm performs.

4.2.2. Intra-cluster and inter-cluster distances

Clustering results can be measured by calculating the average intra-cluster distance (*Intra*) and average inter-cluster distance (*Inter*).

The intra-cluster distance measure is the distance between a point and its cluster center. We take the average of all of these distances and call it *Intra* which is defined as

$$Intra = \frac{1}{n} \sum_{i=1}^{K} \sum_{x_j \in C_i} \|x_j - z_i\|^2, \tag{8}$$

where n is the total number of objects in a data set.

The inter-cluster distance between two clusters is defined as the distance between the centers of them. We calculate the average of all of these distances as follows

$$Inter = \frac{1}{K} \sum \|z_i - z_j\|^2, \quad i = 1, 2, \ldots, K-1, \quad j = i+1, \ldots, K. \tag{9}$$

A good clustering method should produce clusters with high intra-class similarity while low inter-class similarity. The similarity is expressed in terms of a distance function which is usually very different in diverse applications. Therefore, we want to minimize the value of measure *Intra* and maximize the value of *Inter*.

4.2.3. F-measure

F-measure is the widely used statistical validation which considers both the *Precision* and *Recall* information.

Generally, some symbols are introduced for the convenience of evaluating the clustering results. C_i^0 ($i = 1, 2, \ldots, k$) is used to represent the actual target clusters, where k is the number of desired clusters set in the algorithms. The corresponding clusters detected by algorithms are named as C_i^s ($i = 1, 2, \ldots, k$). The performance of clustering can be evaluated in terms of *Precision* and *Recall* which can be calculated as follows:

$$Precision = \frac{|\, C_i^0 \,| \bigcap |\, C_i^s \,|}{|\, C_i^s \,|} = \frac{\text{sum of correctly detected objects for cluster } i}{\text{sum of detected objects for cluster } i}, \tag{10}$$

$$Recall = \frac{|\, C_i^0 \,| \bigcap |\, C_i^s \,|}{|\, C_i^0 \,|} = \frac{\text{sum of correctly detected objects for cluster } i}{\text{sum of objects actrually in cluster } i}. \tag{11}$$

Both the two criterions vary from 0 to 1. In principle, we want both high *Precision* and high *Recall* in the experiments.

F-measure is the harmonic mean of *Precision* and *Recall* and is calculated by

$$F = \frac{(b^2 + 1) \cdot Precision \cdot Recall}{b^2 \cdot (Precision + Recall)}, \tag{12}$$

where we chose $b = 1$, to obtain equal weighting for *Precision* and *Recall*.

F-measure is limited to the interval [0, 1] and should be maximized.

5. Simulation experiments

In this section, we will present several simulation experiments on the platform of Matlab to give a detailed illustration on the superiority and feasibility of the proposed approach.

5.1. Data source

Two different types of benchmark data sets are used: two synthetic data sets [30] that permit the modulation of specific data properties and three real data sets provided by UCI Machine Learning Repository [31].

Both of the two synthetic data sets in our work follow x-dimensional normal distributions $N(\vec{\mu}, \vec{\sigma})$ from which the data items are located into the y different clusters. The sample size s of each cluster, the mean vector $\vec{\mu}$ and the vector of the standard deviation $\vec{\sigma}$ are themselves randomly determined using uniform distributions over fixed ranges (with $s \in [50, 450]$, $\mu_i \in [-10, 10]$ and $\sigma_i \in [0, 5]$). Consequently, clusters in each data set are with

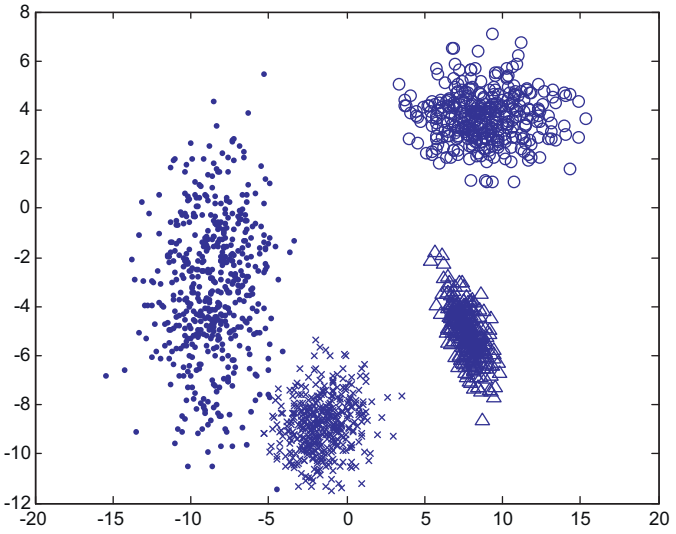

Fig. 2. The original two-dimensional data distribution in space.

different size and different density. The first one, which we call it 2D-4C, is a two-dimensional data set arranged in ([−20, 20], [−12, 8]) and contains 4 clusters with 528, 348, 272 and 424 instances each (see Fig. 2). The second data set, named 10D-4C, contains a total number of 1289 items that spread in 4 clusters based on 10 different features.

All the 3 data sets from UCI that we employ in our experiments are famous databases that can be easily found in data mining and pattern recognition literatures. Iris data set contains 3 classes of 50 instances each, where each class refers to a type of iris plant and can be treated as a cluster in the experiments. Each instance has 4 features representing sepal length, sepal width, petal length and petal width, respectively. Wine data are the results of a chemical analysis of wines grown in the same region in Italy but derived from three different cultivars. This set contains 3 clusters and has 59, 71, 48 instances for each cluster. Glass data set has 214 instances describing 6 classes of glass based on 9 features. The data points in all the 3 data sets are scattered in high-dimensional spaces.

Table 1
Summarization of data sets.

Data sets	Instances	Featrues/dimensions	Clusters
2D-4C	1572	2	4
10D-4C	1289	10	4
Iris	150	4	3
Wine	178	13	3
Glass	214	9	6

The description of all the data sets used in our study can be summarized in Table 1.

5.2. Parameter settings

In the initialization step, there are a number of parameters that need to be set for both CAS-C and PSO-clustering algorithms.

For PSO, we use 50 particles, and set $w = 0.72$ and $c_1 = c_2 = 1.49$. These values were chosen to ensure good convergence [32].

For CAS-C, 30 ants are used. We default $a = 2000$, $b = 7/12$, $y(0) = 0.999$, $r_i = 0.02 + rand(1) * 0.5$ and $Istep = 800$. The value of parameter ψ_d can be selected according to the ranges of intervals [17].

5.3. Results

For all the results reported, average measures over 30 simulations are given. Euclidean distance is chosen to measure the distance among data points in our work.

Table 2
Values of performance measures by the *k*-means, PSO and the proposed CAS-C algorithms.

Data sets	Method	*F*	SSE	Intra	Inter
2D-4C	*k*-means	0.7682	4.2041×10^3	2.2376	7.4135
		(0.0104)	(53.6571)	(0.3850)	(0.8914)
	PSO	0.9553	3.7578×10^3	1.8331	9.6757
		(0.0062)	(44.3622)	(0.13687)	(0.47761)
	CAS-C	**0.9966**[a]	**3.5906×10^3**[a]	**1.4112**[a]	**10.7462**[a]
		(0.0027)	(38.2317)	(0.0925)	(0.2351)
10D-4C	*k*-means	0.6306	1.7512×10^4	3.4861	12.0786
		(0.067369)	(343.2058)	(0.17326)	(1.25532)
	PSO	0.6436	2.0309×10^4	4.0123	14.1810
		(0.037196)	(358.3691)	(0.14857)	(1.20055)
	CAS-C	**0.7089**[a]	1.9008×10^4[b]	3.6957[b]	**14.8365**[a]
		(0.012749)	(316.5987)	(0.05923)	(0.09487)
Iris	*k*-means	0.8853	97.3462	0.8023	3.5196
		(0.13340)	(9.6661)	(0.002076)	(0.06771)
	PSO	0.9333	97.3259	0.7898	3.5483
		(0.042715)	(10.77132)	(0.00349)	(0.0486)
	CAS-C	**0.9333**[a]	**97.3094**[a]	**0.7794**[a]	**3.5767**[a]
		(0.010368)	(8.8901)	(0.00095)	(0.02074)
Wine	*k*-means	0.702	1.6656×10^4	0.9039	44.5206
		(0.14568)	(76.3192)	(0.051477)	(0.72684)
	PSO	0.6937	1.6548×10^4	0.9069	43.1749
		(0.33945)	(72.14098)	(0.05796)	(0.41683)
	CAS-C	**0.735**[a]	**1.6363×10^4**[a]	**0.9037**[a]	43.3477[b]
		(0.030169)	(44.0654)	(0.00569)	(0.08893)
Glass	*k*-means	0.4768	240.5743	1.1044	7.9543
		(0.0014282)	(24.43907)	(0.07563)	(1.8499)
	PSO	0.4801	238.9055	1.0672	8.6098
		(0.001387)	(21.70175)	(0.06820)	(1.34023)
	CAS-C	**0.4957**[a]	**236.7708**[a]	**1.0276**[a]	**10.23569**[a]
		(0.001291)	(18.46152)	(0.01298)	(0.83705)

The bold value is the best-performed result among 3 results acquired by 3 different approaches (Rank 1).
[a] Rank 1.
[b] Rank 2.

Table 3
Results of *t*-test on *Intra*.

Data sets	*t*	Degree of freedom (*DF*)	Significant probability (*p*)
2D-4C	−11.4	58	2.31E−13
10D-4C	−8.85	58	1.83E−09
Iris	−2.94	58	0.00382
Wine	−2.49	58	0.01102
Glass	−2.55	58	0.00944

For all the results reported, average values of different performance indices over 30 simulations and their corresponding standard deviations (shown in bracket) for each data set are given in Table 2. Euclidean distance is chosen to measure the distance between data points in our work.

A *Student's t-test* has been conducted *Intra* and *Inter* for PSO-based and CAS-C algorithms. Tables 3 and 4 present the statistical results comparisons, where *p* is the probability if the null hypothesis (*H*0) is supported.

For the *Intra* measure, we implement our *t*-test based on:

$$H0 : Intra(\text{CAS-C}) < Intra(\text{PSO}),$$
$$H1 : Intra(\text{CAS-C}) \geq Intra(\text{PSO}).$$

For the *Inter* measure, we implement our *t*-test based on:

$$H0 : Inter(\text{CAS-C}) > Inter(\text{PSO}),$$
$$H1 : Inter(\text{CAS-C}) \leq Inter(\text{PSO}).$$

From the clustering results shown in Tables 2–4, and according to the properties of data sets which are described in Table 1, some conclusions could be revealed as follows:

(1) After label comparison for each data set, CAS-C acquired the largest average *F-measure* value for all 5 data sets, which means the CAS-C algorithm has the lowest error rate and performs the best accuracy ability of clustering. PSO-based clustering technique also gained larger *F* values than the *k*-means algorithm, but smaller than the CAS-C approach for 4 data sets. It can be found that the global search ability indeed help the optimization-based clustering techniques make significant improvement to the *k*-means algorithm. Furthermore, for the 10D-4C and the Wine data sets, the CAS-C algorithm performed distinct improvements than the PSO-based approach, which means CAS-C are more suitable to group the data with high-dimension and multiple cluster densities.

(2) Consider the fitness of solutions, i.e. the minimum value of objective function, SSE. For all data sets, except 10D-4C, the CAS-C algorithm had the smallest value of SSE. Moreover, based on the comparison of SSE, the CAS-C algorithm minimized the objective function better than PSO algorithm for all 5 data sets. These results show that with the help of global and chaotic search, the proposed CAS-C methodology can reach the global optimal solutions, which has covered the shortage of the *k*-means algorithm. Meanwhile, as an optimization-based clustering algorithm, CAS-C reached the optimal points more closer and exhibited better convergence than the PSO algorithm.

(3) When considering intra-cluster and inter-cluster distances, the former ensures compact clusters with little deviation from the cluster centers, while the latter ensures larger separation between the different clusters. With reference to these criteria, the CAS-C algorithm succeeded most in finding clusters with larger separation than the *k*-means algorithm and PSO technique, although *k*-means algorithm did the best for the wine data set. It is also the CAS-C algorithm that succeeded in forming the more compact clusters than the other two methods on all data sets but 10D-4C.

(4) Then come to result comparisons between data sets. The proposed CAS-C algorithm performs better than PSO-based and the *k*-means approaches for all the data sets: It had the best SSE, *F-measure*, *Intra* and *Inter* values for the 2D-4C, Iris and Glass data sets; It got the best SSE, *F-measure*, and *Intra* values and the second best inter-cluster distance for the Wine data set; It acquired the best *F-measure* and *Inter* measures for the 10D-4C data set with the second best SSE and *Intra* values. We can see that the PSO-based approach exhibited worse than CAS-C and even *k*-means algorithm on 10D-4C and Wine data sets, which shows the PSO-based clustering techniques are not good at handling high-dimensional data sets, while CAS-C can overcome this drawback.

(5) The standard deviations of different measures obtained by different methods are shown in bracket. Stability of CAS-C over different data sets can also be seen from the smallest values of standard deviation of all indices. These comparisons present that the results of the CAS-C algorithm change less at different experiments, and CAS-C is a more stable clustering technique than the *k*-means and PSO-based clustering algorithms. The results of one-tailed *t*-test presented in Tables 3 and 4 reveal that *Intra*(CAS-C) is significant smaller than *Intra*(*PSO*) ($t<0$ with $p>0.05$), while *Inter*(CAS-C) is significant larger than *Inter*(*PSO*) ($t>0$ with $p<0.05$) on all data sets in our experiments.

To sum up, CAS-C is a high-quality clustering algorithm which can find the global optimum clustering result and have a good algorithm performance for the data set with high-dimension and multiple cluster densities.

Table 4
Results of *t*-test on *Inter*.

Data sets	*t*	Degree of freedom (*DF*)	Significant probability (*p*)
2D-4C	8.99	58	5.23E−10
10D-4C	2.43	58	0.01253
Iris	2.4	58	0.0119
Wine	1.81	58	0.04217
Glass	4.6	58	3.18E−05

6. Conclusion

This paper presents an efficient clustering algorithm based on the chaotic ant swarm optimization. The clustering problem is converted to that of seeking the center for each cluster by optimizing the objective function. Numerical simulations are given to show that the proposed CAS-based clustering algorithm have better convergence ability and could be used to achieve high quality on multi-dimensional data sets and can detect clusters with different sizes or densities with encouraging results.

Acknowledgments

This work is supported by the National Natural Science Foundation of China (Grant Nos. 61070209, 61121061), the Program for New Century Excellent Talents in University of the Ministry of Education of China (Grant No. NCET-10-0239), the Specialized Research Fund for the Doctoral Program of Higher Education (Grant No. 20100005110002), the National Science Foundation of China Innovative Grant (Grant No. 70921061), the CAS/SAFEA International Partnership Program for Creative Research Teams and the Asia Foresight Program under NSFC Grant (Grant No. 61161140320).

References

[1] A.K. Jain, M.N. Murty, P.J. Flyn, Data clustering: a review, ACM Computing Surveys 31 (3) (1999) 264–323.
[2] A. Dempster, N. Laird, D. Rubin, Maximum likelihood from incomplete data via the EM algorithm, Journal of the Royal Statistical Society. Series B (Methodological) 39 (1) (1977) 1–38.
[4] R. Xu, D. Wunsch II, Survey of clustering algorithms, IEEE Transactions on Neural Networks 16 (3) (2005) 645–678.

[5] S. Guha, R. Rastogi, K. Shim, Cure: an efficient clustering algorithm for large databases, in: Proceedings of ACM SIGMOD Conference on Management of Data, 1998, pp. 73–84.
[6] J. Handl, B. Meyer, Ant-based and swarm-based clustering, Swarm Intelligence 1 (1) (2007) 95–113.
[7] A. Hinneburg, D. Keim, An efficient approach to clustering in large multimedia databases with noise, in: Proceedings of the 4th International Conference on Knowledge Discovery and Data Mining (KDD-98), 1998, pp. 58–65.
[8] E. Hruschka, R. Campello, L. de Castro, Evolving clusters in gene-expression data, Information Sciences 176 (13) (2006) 1898–1927.
[9] J.C. Bezdek, Pattern Recognition with Fuzzy Objective Function Algorithms, Plenum Press, New York, 1981, pp. 95–107.
[10] J. Zhang, Y. Leung, Improved possibilistic C-means clustering algorithms, IEEE Transactions Fuzzy Systems 12 (2) (2004) 209–217.
[11] I.S. Dhillon, Y. Guan, B. Kulis, Weighted graph cuts without eigenvectors: a multilevel approach, IEEE Transactions on Pattern Analysis Machine Intelligence 29 (11) (2007) 1944–1957.
[12] I.S. Dhillon, Y. Guan, B. Kulis, A unified view of kernel *k*-means, spectral clustering and graph partitioning, Technical Report TR-04-25, UTCS, 2005.
[13] I.S. Dhillon, Y. Guan, B. Kulis, Kernel *k*-means: spectral clustering and normalized cuts, in: Proceedings of the Tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, ACM Press, NY, USA, 2004, pp. 551–556.
[14] J. MacQueen, Some methods for classification and analysis of multivariate observations, in: Proceedings of the 5th Berkeley Symposium on Mathematical Statistics and Probability, 1967, pp. 281–297.
[15] D.W. van der Merwe, A.P. Engelbrecht, Data clustering using particle swarm optimization, in: Proceedings of IEEE Congress on Evolutionary Computation, 2003, pp. 215–220.
[16] R.T. Ng, J. Han, Efficient and effective clustering methods for spatial data mining, in: Proceedings of the 20th International Conference on Very Large Data Bases Conference, 1994, pp. 144–155.
[17] L. Li, Y. Yang, H. Peng, X. Wang, An optimization method inspired by chaotic ant behavior, International Journal of Bifurcation and Chaos 16 (2006) 2351–2364.
[18] L. Li, Y. Yang, H. Peng, X. Wang, Parameters identification of chaotic systems via chaotic ant swarm, Chaos Solitons and Fractals 28 (2006) 1204–1211.
[19] J. Cai, X. Ma, L. Li, Y. Yang, H. Peng, X. Wang, Chaotic ant swarm optimization to economic dispatch, Electric Power Systems Research 77 (2007) 1373–1380.
[20] L. Li, Y. Yang, H. Peng, Fuzzy system identification via chaotic ant swarm, Chaos Solitons and Fractals 40 (2009) 1399–1407.
[21] B.J. Cole, Is animal behavior chaotic? Evidence from the activity of ants, Proceedings Royal Society London B–Biological Sciences 244 (1991) 253–259.
[22] R.V. Solé, O. Miramontes, B.C. Goodwill, Oscillations and chaos in ant societies, Journal of Theoretical Biology 161 (1993) 343–357.
[23] G. Sheikholeslami, S. Chatterjee, A.D. Zhang, WaveCluster: A multi-resolution clustering approach for very large spatial databases, in: Proceedings of the 24th International Conference on Very Large Data Bases, 1998, pp. 428–439.
[24] P.S. Shelokar, V.K. Jayaraman, B.D. Kulkarni, An ant colony approach for clustering, Analytica Chimica Acta 509 (2004) 187–195.
[25] A.K. Jain, R.C. Dubes, Algorithms for Clustering Data, Prentice-Hall Advanced Reference Series, Prentice-Hall, Inc., 1988.
[26] P-N. Tan, M. Steinbach, V. Kumar, Introduction to Data Mining, Pearson Addison-Wesley, 2006.
[27] J. Kennedy, R. Eberhart, Particle swarm optimization, in: Proceedings of the IEEE International Joint Conference on Neural Networks (ICW), Vol. 4, Perth, Australia, 1995, pp. 1942–1948.
[28] Y. Shi, R. Eberhart, A modified particle swarm optimizer, in: Proceedings of IEEE International Conference on Evolutionary Computation (ICEC'98), Anchorage, 1998, pp. 69–73.
[29] H. Spath, Cluster Analysis Algorithms for Data Reduction and Classification, Ellis Horwood, Upper Saddle River, NJ, 1980.
[30] Cluster Generators: Synthetic Data for the Evaluation of Clustering Algorithms, http://dbkgroup.org/handl/generators/.
[31] UCI Machine Learning Repository, http://archive.ics.uci.edu/ml/index.html, University of California, Irvine, Department of Information and Computer Science, Center for Machine Learning and Intelligent Systems, 2007.
[32] F. van den Bergh, An Analysis of Particle Swarm Optimizers, Ph.D. Thesis, Department of Computer Science, University of Pretoria, Pretoria, South Africa, 2002.