Pattern Recognition 47 (2014) 2143–2152

Contents lists available at ScienceDirect

Pattern Recognition

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

A particle swarm optimization based simultaneous learning framework for clustering and classification

Ruochen Liu*, Yangyang Chen, Licheng Jiao, Yangyang Li

Key Laboratory of Intelligent Perception and Image Understanding of Ministry of Education of China, Xidian University, Xi'an 710071, China

ARTICLE INFO

Article history:
Received 27 November 2012
Received in revised form
30 September 2013
Accepted 20 December 2013
Available online 4 January 2014

Keywords:
Classification
Particle swarm optimization
Clustering
Image segmentation
Global factor

ABSTRACT

A particle swarm optimization based simultaneous learning framework for clustering and classification (PSOSLCC) is proposed in this paper. Firstly, an improved particle swarm optimization (PSO) is used to partition the training samples, the number of clusters must be given in advance, an automatic clustering algorithm rather than the trial and error is adopted to find the proper number of clusters, and a set of clustering centers is obtained to form classification mechanism. Secondly, in order to exploit more useful local information and get a better optimizing result, a global factor is introduced to the update strategy update strategy of particle in PSO. PSOSLCC has been extensively compared with fuzzy relational classifier (FRC), vector quantization and learning vector quantization (VQ+LVQ3), and radial basis function neural network (RBFNN), a simultaneous learning framework for clustering and classification (SCC) over several real-life datasets, the experimental results indicate that the proposed algorithm not only greatly reduces the time complexity, but also obtains better classification accuracy for most datasets used in this paper. Moreover, PSOSLCC is applied to a real world application, namely texture image segmentation with a good performance obtained, which shows that the proposed algorithm has a potential of classifying the problems with large scale.

1. Introduction

The primary goal of pattern recognition is supervised and unsupervised classifications [1,2]. The frame of the pattern recognition is taken as: definition of pattern classes, sensing environment, pattern representation, feature extraction and selection, cluster analysis, classifier design and learning, selection of training and test samples, and performance evaluation. No doubt pattern recognition has already caused a high level of concern because of its importance in a variety of scientific disciplines and engineering such as medicine, computer vision, and artificial intelligence.

In supervised classification, the classifier is designed using samples with class labels, and then always classifies those data with unknown class labels. Up to now, many techniques for such classification have been developed, such as linear and logistic regression, decision trees [3] and rules, *k*-nearest neighbor classifiers [4], neural networks [5], and support vector machines [6]. They have been proven to achieve good performance successfully. In unsupervised classification (e.g., clustering), the main aim is to group a given collection of unlabeled patterns into meaningful clusters, which is performed to uncover the distribution of the whole data, that is, reveal the underlying structure in data.

Different from the approaches proposed in Refs. [1,2], another type of classifiers is proposed by incorporating structural information into their classification schemes [7–10]. Firstly, the clustering analysis is utilized to discover the natural structure in data. Then, a classifier based on the obtained structural information is designed. The results show that using this way not only reveals the data distribution but also improves the classification learning to some extent. In radial basis function neural network (RBFNN) [7], training samples are clustered so as to determine the parameters of the hidden layer by using fuzzy c-means (FCM) [8]. While in fuzzy relational classifier (FRC) [9], the training phase includes two phases. Firstly, execute the clustering on the training samples, then a relation matrix between clusters and the class labels is established, the matrix here is constructed by some operator rather than by optimizing a function. In vector quantization and learning vector quantization (VQ+LVQ3) [10], only the positions and class labels of the centers are used to classify new samples, and neither of them need to predetermine the relational matrix between clusters and classes.

Obviously, these algorithms have a common point: the clustering learning and the classification learning optimize their own criteria sequentially and separately, which means that the clustering learning here just aids the classification learning but does not benefit from the classification learning. Recently, a simultaneous learning framework for clustering and classification (SCC) is presented [11]. In SCC, the Bayesian theory and the cluster posterior probabilities of classes are employed to associate the classification learning with the clustering

* Corresponding author. Tel.: +86 29 88202661; fax: +86 29 88201023.
E-mail address: aliang3399@gmail.com (R. Liu).

0031-3203/$ - see front matter © 2014 Elsevier Ltd. All rights reserved.
http://dx.doi.org/10.1016/j.patcog.2013.12.010

learning. Based on this point, authors define a single objective function to which the clustering process is directly embedded, which ensures the evaluation of both clustering and classification performance at the same time. In the process of optimization, not only the positions of the individuals are optimized, but a relation matrix is also obtained, which represents the connection between classification and clustering. Therefore, some more meaningful information becomes transparent. Compared to other algorithms, it has been proved that SCC can get a visible superiority, possessing a simultaneous frame formed of clusters and classes.

However, in the cluster learning stage of SCC [11], the number of cluster k is set in the range from $l_{\max}$ to $c_{\max}$, suggested by Bezdek [12]. Obviously, there also exists a large uncertainty and randomness. Suppose that k is a large value, this kind of selection strategy will lead to a higher dimension structure with solutions encoded. Similarly, the trial and error [13], which is the traditional method for finding out the value of cluster number k, demands running algorithm time by time to determine the number of clusters. In this paper, an improved simultaneous framework for clustering and classification (PSOSLCC) is proposed. In order to compensate for the shortcoming mentioned above, an automatic clustering using an improved differential evolution algorithm (ACDE) [14] is used to obtain the number of clusters, which not only save the time, but also can always get approximate true number of clusters. Meanwhile, SCC used a modified particle swarm optimizer (PSOm) to optimize the objective function, while the proposed algorithm adopts a global factor to update the positions of particles in the process of the PSOm, which can get more local information from neighborhood and improve the performance to some extent. PSOSLCC is compared against FRC, VQ+LVQ3, RBFNN and SCC. The experimental results show that the PSOSLCC is better than other algorithms in term of classification accuracy. In addition, PSOSLCC is also applied to a real world application, namely texture image segmentation.

The rest of this paper is organized as follows: in Section 2, the proposed algorithm (PSOSLCC) is described in detail. The experimental sets, results on two synthetic datasets and several real-life benchmark datasets are elaborated in Section 3. In addition, the algorithm is executed on several texture images. Finally, the conclusions are displayed in Section 4.

2. The proposed algorithm

In PSOSLCC, by running an automatic clustering using improved differential evolution algorithm (ACDE) [14] for many times, the optimal cluster number of dataset is found and meanwhile a set of clustering centers corresponding to the optimal clustering number are obtained. Then a simultaneous learning framework for clustering and classification based on an improved particle swarm optimization is adopted to classify the dataset. In the proposed algorithms, these clustering centers will act as the initial swarm in the improved PSO, and a special objective function used in [11] will be optimized to obtain a best clustering centers for training datasets and can evaluate the classification and clustering ability simultaneously, in which, a relational matrix is established through Bayesian theorem to formulate the relationship between clusters and classes. Moreover, a global factor is introduced into PSOm [14] to update the position of particles in order to obtain a better performance.

The procedure of the proposed PSOSLCC will be described as follows:

Step 1: Load dataset, and divide the dataset into two parts, training dataset and testing dataset.
For training dataset (training phase):
Step 2: Run the automatic clustering algorithm (ACDE) to determine the optimal cluster number, obtain a set of clustering centers, which are defined as the initial positions of the particles in PSO, and set the current iteration number t to 1 and the ideal value of the objective function of PSO to 0.
Step 3: Calculate $p(c_j|x_i)$ using fuzzy c-means (FCM) based on the training dataset by using Eq. (11). Here x_i is the training sample and c_j represents the jth cluster, $p(c_j|x_i)$ is the posterior probability of sample x_i belonging to jth cluster.
Step 4: Calculate relational matrix P by using Eq. (13), then determine the labels for the training dataset based on $p(c_j|x_i)$ and P by using Eq. (9).
Step 5: Based on the results of steps 3 and 4, calculate the objective value of each particle according to Eq. (16).
Step 6: If $t=1$, set p_i of each particle and its objective value equal to its current position and objective value; otherwise compare the current particle's objective value with the previous, and choose the better one as the current position and objective value.
Step 7: Determine the best particle p_g with the best objective value.
Step 8: Update the velocity and position of each particle by using Eqs. (28) and (29).
Step 9: Update the inertia weight ω using Eq. (22).
Step 11: If $t \geq t_{\max}$ or the objective function value of $p_g \leq 0$, then stop; otherwise return back to step 3.
For testing dataset (testing phase):
Step 12: Determine the labels of the testing data using position of global optimum particle and its corresponding relational matrix P.
Step 13: Calculate the classification accuracy.
Step 14: End.

In the following sections, we will give a detailed introduction of each step of the proposed algorithm.

2.1. The automatic clustering algorithm

In this paper, the simultaneous frame consists of two parts: clustering and classification. For the clustering part, the number of clusters k becomes a key issue to be determined. One of the intuitive manners is the "trial-and-error" [13]. First, the dataset is divided into c clusters, and then the classification rate of the algorithm is evaluated. Finally, choose the appropriate value for parameter k. In [11], the parameter k is set in the range from the number of classes $l_{\max}$ up to $c_{\max}$, $c_{\max}$ is set as $\sqrt{N}$ and N is the number of the training samples [12]. Due to the uncertainty of k, the method above has a higher computational complexity.

To settle this problem, we adopt an automatic clustering algorithm rather than the trial-and-error method to determine the parameter k. Das et al. [14] put forward an automatic clustering using improved differential evolution algorithm (ACDE) for the clustering of large unlabeled datasets. Two techniques are used in the algorithm to find the correct number of clusters as well as the optimal partitioning. One is the algorithm adopting a real-coding chromosome with fixed length which contains activation threshold and clustering center. The activation threshold controls whether the corresponding clustering center is to be activated. The other is using an improved differential evolution to further improve the algorithm's accuracy, convergence speed, and robustness.

By introducing ACDE into the proposed algorithm, not only the computation complexity is reduced greatly, but also a set of clustering centers can be achieved at the same time, which most of the times can reveal the underlying structure of data correctly. We will describe ACDE as follows.

Table 1
Solution encoding in ACDE.

$\underbrace{T_{i,1}\ T_{i,2}\ \cdots\ T_{i,K_{\max}}}_{\text{Activation Threshholds}}$	$\underbrace{\mathbf{c}_{i,1}\ \mathbf{c}_{i,2}\ \cdots\ \mathbf{c}_{i,K_{\max}}}_{\text{Clusher Centers}}$

$K_{\max}$ is the maximum number of clusters. $T_{ij}, j \in [1, K_{\max}]$ is the decision-maker. When $T_{ij} > 0.5$, the corresponding $c_{i,j}$ is active, otherwise, it is inactive.

2.1.1. Solution encoding

In ACDE, the encoding of the solution contains two parts: threshold and feature attributes with d dimensions. For the first part, all the values are initialized in the range [0,1], each of which controls whether the corresponding individual is to be activated or not. In the second part, every sample consists of the same dimensions as the training datasets. The whole solution is shown in Table 1.

2.1.2. Fitness function

In ACDE, PBM index [15] is used to evaluate the solutions, which is proposed by Pakhira, Bandyopadhyay and Maulik in 2004. PBM index can be defined as

$$PBM(k) = \left(\frac{1}{k} \times \frac{E_1}{E_k} \times D_k\right)^2 \tag{1}$$

where k is the number of clusters. And the other factors can be shown as

$$E_k = \sum_{i=1}^{k} E_i \tag{2}$$

$$E_i = \sum_{j=1}^{n} u_{ij} ||x_j - c_i|| \tag{3}$$

$$D_k = \max_{i,j=1}^{k} ||c_i - c_j|| \tag{4}$$

where n is the scale of the training data, u_{ij} is the partition matrix, x_{ij} is the sample with label, c_j is the cluster center. Thus, the function value of the ith individual is

$$f_i = PBM_i(k) \tag{5}$$

Maximizing Eq. (5) will result in compact clusters and k in PBM index got will be the optimal number of clusters.

2.1.3. Update strategy

Differential evolution is also an evolutionary algorithm, which varies with iteration [14]. For each individual, DE algorithm selects three other individuals randomly as reference. Suppose that x_k^t is the current individual, and the others are x_i^t, x_j^t, and x_m^t. Then, the offspring can be updated as

$$u_{k,d}^{t+1} = \begin{cases} x_{m,d}^t + F \cdot (x_{i,d}^t - x_{j,d}^t) & \text{if} \quad rand_d(0,1) < C_r \\ x_{k,d}^t & \text{otherwise} \end{cases} \tag{6}$$

where $u_{k,d}^{t+1}$ represents the offspring, C_r is a scalar parameter between 0 and 1, called the crossover rate, the choice of F is as:

$$F = 0.5 \times (1 + rand(0.1)) \tag{7}$$

Then, the new individual is selected between u_i^{t+1} and x_i^t, according to the following equation:

$$x_i^{t+1} = \begin{cases} u_i^{t+1} & \text{if } f(u_i^{t+1}) > f(x_i^t) \\ x_i^t & \text{if } f(u_i^{t+1}) \le f(x_i^t) \end{cases} \tag{8}$$

where $f(\ )$ is the fitness value of the sample.

The procedure of the ACDE used in the proposed algorithm can be shown below:

Step 1: Load dataset.
Step 2: Initialization. Each individual in the population consists of thresholds and cluster centers.
Step 3: Find out the valid cluster centers in each individual using the value of thresholds.
Step 4: For $t=1$ to $t_{\max}$ do
1. Clustering using the cluster centers and dataset for the current individual.
2. Calculate the fitness of the current individual and select three other individuals in the population.
3. Check whether the number of data belonging to the same class is larger than 2. If so, update the position of the current individual according to Eqs. (6) and (8). Use the fitness of the individuals to guide the evolution of the population.

Step 5: Record the final global optimum individual and the corresponding cluster number at $t=t_{\max}$.
Step 6: End.

2.2. Relational matrix

After running the ACDE, a set of clustering centers have gotten, then a classification mechanism depending only on them is adopt in [11], which uses both cluster information and the relational matrix to determine the class labels for new samples. In the training process, a relational matrix of each clustering centers is established firstly between the cluster and class labels. After optimizing, which is described in next section, we can obtain the relational matrix of the optimum clustering centers. In the testing process, the classification mechanism combined with the relational matrix of the optimum clustering centers is used to determine the class labels for new samples. The following details the classification mechanism.

Suppose the posterior probabilities $p(w_l|x_i)$ for each sample are available. Then the label for sample x_i can be obtained by the following equation:

$$f(x_i) = \arg \max_{1 \le l \le L} p(\omega_l|x_i) \tag{9}$$

here w_l denotes the lth class, x_i is the ith sample in the input samples (in the training process they are training samples and in the testing process they are testing samples), L is the number of classes, $f(x)$ is the new label for the sample. In order to combine the classification with the clustering, the total probability theorem is used to incorporate the clusters into the reformulation $p(w_l|x_i)$ shown by the following equation:

$$p(\omega_l|x_i) = \sum_{j=1}^{k} p(\omega_l, c_j|x_i) = \sum_{j=1}^{k} p(c_j|x_i) p(\omega_l|c_j, x_i) = \sum_{j=1}^{k} p(c_j|x_i) p(\omega_l|c_j) \tag{10}$$

where k is the number of clusters, c_j is the jth cluster center, $p(c_j|x_i)$ is the posterior probability of sample x_i belonging to jth cluster. From $p(\omega_l|c_j, x_i)$, it can be seen that $p(\omega_l|c_j, x_i)$ is independent of x_i, therefore, it can be taken as $p(\omega_l|c_j)$, and denotes the lth class posterior probabilities in the condition of jth cluster.

In Eq. (10), $p(c_j|x_i)$ can be calculated by using different clustering models. In this paper, FCM is applied to compute the posterior probability and it can be obtained by the following equation:

$$p(c_j|x_i) = \frac{dist(x_i, c_j)^{-1}}{\sum_{r=1}^{K} dist(x_i, c_r)^{-1}} \tag{11}$$

where c_j denotes the jth cluster center, and it can be observed that the formula only depends on the positions of the cluster centers. Thus, the optimization for centers becomes vital important in the whole training process.

In Eq. (10), the posterior probability is calculated through Bayesian theory as shown in the following equation:

$$p(\omega_l|c_j)=\frac{p(\omega_l,c_j)}{p(c_j)} \tag{12}$$

where $p(c_j)$ is the proportion of the samples in the jth cluster to the whole training samples, such as $Num(x\in c_j)/N$, N is the size of training samples, $p(\omega_l,c_j)$ represents the probability of the samples in the lth class and jth cluster simultaneously, shown as $Num(x\in\omega_l \text{ and } x\in c_j)/N$. So the whole expression can be written as the following equation:

$$p(\omega_l|c_j)=\frac{Num(x\in\omega_l \text{ and } x\in c_j)}{Num(x\in c_j)} \tag{13}$$

From Eq. (13), a relation matrix can be derived as

$$\sum_{l=1}^{L} p(\omega_l|c_j)=1 \tag{14}$$

And the whole expression can be expanded as the following equation:

$$P=\begin{bmatrix} p(\omega_1|c_1) & p(\omega_2|c_1) & \dots & p(\omega_L|c_1) \\ p(\omega_1|c_2) & p(\omega_2|c_2) & \dots & p(\omega_L|c_2) \\ \dots & \dots & \dots & \dots \\ p(\omega_1|c_k) & p(\omega_2|c_k) & \dots & p(\omega_L|c_k) \end{bmatrix} \tag{15}$$

It can be seen that $p(\omega_l|c_j)$ is a matrix with $k\times L$, revealing a statistical relationship between the lth class and the jth cluster. $p(\omega_l|c_j)$ represents the probability of the samples in the lth class and the jth cluster simultaneously to the jth cluster. When $p(\omega_l|c_j)$ is small, the number of samples in the jth cluster from the lth class is fewer.

Here, we note that the classification mechanism is only relevant to the cluster centers. From Eq. (13), $p(\omega_l|c_j)$ only depends on the cluster centers, each sample of the training samples only belongs to one cluster. According to Eq. (10), $p(c_j|x_i)$ just relies on the positions of the cluster centers as well. Thus, the posterior probability $p(\omega_l|x_i)$ is determined by the clustering centers, which means that it is crucial to optimize the cluster centers.

2.3. Objective function

Based on the above classification mechanism, the classification and clustering is optimized in a single objective function [11]. To realize the purpose, the objective function consists of two parts: misclassification rate and clustering impurity. Suppose the samples and labels are x_i and ω_i, where $x_i\in R^d$ and $\omega_i\in\{1,2,\dots,L\}$, and then the objective function is used in proposed algorithm given by the following equation [11]:

$$J(\{c_j\})=\sum_{i=1}^{N}\delta(f(x_i),\omega_i)/N+\beta q(X) \tag{16}$$

where $f(x_i)$ denotes the label obtained from the classification mechanism. When $f(x_i)=\omega_i$, the value of δ function is 0, otherwise 1. $q(X)$ represents the clustering impurity explained by the following equation:

$$\begin{aligned} q(X)&=1-\sum_{j=1}^{k}\max_{l=1,2,\dots,L} p(\omega_l,c_j) \\ &=1-\sum_{j=1}^{k}\max_{l=1,2,\dots,L} p(\omega_l|c_j)\times p(c_j) \\ &=1-\sum_{j=1}^{k}\frac{\max_{l=1,2,\dots,L} p(\omega_l|c_j)\times Num(x\in c_j)}{N} \end{aligned} \tag{17}$$

As referred in [11], the parameter β is an important factor used to balance the classification rate and clustering impurity, and its value is taken among {0.01, 0.1, 1}. Obviously, the process of optimizing the centers $\{c_j\}$ is the process of adjusting the simultaneous framework, composed of misclassification rate and the clustering impurity.

In Eq. (17), in order to calculate $Num(x\in c_j)$, various distance metrics can lead to different algorithms. In this paper, a distance metric based on kernel-induced is used to improve the performance of the algorithm. Thus, the distance formula can be written as follows:

$$dist(x_i,c_j)=2-2\exp\left(\frac{-||x_i-c_j||^2}{\delta^2}\right) \tag{18}$$

where δ is a parameter in the distance metric. Due to the great effect, parameter δ is defined in terms of [16]:

$$\delta^2=\frac{\sum_{i=1}^{N}||x_i-\bar{x}||^2}{\lambda} \tag{19}$$

$$\bar{x}=\sum_{i=1}^{N}x_i/N \tag{20}$$

To get the value of δ, a trial-error-approach is used to seek λ in a range of {0.01, 0.05, 0.1, 0.5, 1, 5, 10, 15}[17].

2.4. Particle swarm optimization of objection function

PSO originates from the study of behavior mechanism, generated from birds flocking and fish schooling in the nature [18]. Due to the simplicity of single unity, the main idea of PSO is to complete the complicated task relying on the collaboration with each other in a population. Compared to the basic PSO [19], an inertia weight is introduced into the improved PSO [20] to balance the global search and local search. Obviously, the larger the value of ω is, the greater the impact of the global search will produce. And the algorithm uses a local search when the value of ω is small. Compared to some other evolutionary algorithms, the improved PSO has been proved to get a competitive performance.

2.4.1. The basic PSO

PSO as an evolutionary technique was first put forward by J. Kennedy and R. Eberhart in 1995 [18], a social psychologist and electrical engineer respectively in USA.

In PSO, each individual is treated as a 'particle', which, in fact, represents a solution to the problem. A particle is defined as a point $X_i(x_{i1},x_{i2},\dots,x_{id},\dots,x_{iD})$ in a D-dimensional space. Here, the D-dimensional space is just the search environment. Meanwhile, each particle has its own velocity constituted as $V_i(v_{i1},v_{i2},\dots,v_{id},\dots,v_{iD})$. Thus, it allows the particles to own the ability to move around in the search space. Above all, an array of solutions is initialized randomly, and then updating their positions and velocities according to their own experience and the others. During this phase, a *pbest* (personal best) and *gbest* (global best) will be selected out to be introduced into the following equation:

$$v_{id}^{t+1}=\omega^t\cdot v_{id}^t+c_1\cdot r_1\cdot(p_{id}^t-x_{id}^t)+c_2\cdot r_2\cdot(p_{gd}^t-x_{id}^t) \tag{21}$$

where t is the current iteration number, d is the dth element defined as $d\in[1,D]$, ω is an inertia weight, c_1 and c_2 are two factors, c_1 represents its own dependence, c_2 determines the effect other particles has on the current one. In [21], an adaptive parameter selection in PSO is applied by learning automata, including the inertia weight and acceleration coefficients. Factors r_1 and r_2 are two random values uniformly distributed in the range [0,1], simulating the slight disturbance in the nature environment. p_{id}^t is the dth component of *pbest* at generation t, and p_{gd}^t is the one of *gbest*. Considering the search ability, the inertia weight is defined by the

following equation [22]:

$$\omega^t = 1.4 - 0.4 \times t/T\max \tag{22}$$

where T max is the maximum number of iteration. As shown in Eq. (22), the algorithm tends to have a larger search range at the beginning. While with t increasing, it has more local search around the preferable solution. In [23], a comparison is utilized between the linearly decreasing inertia weight PSO and the adaptive population size PSO, which is also an improved manner for PSO and can be studied in a further step.

The position update strategy is introduced by the following equation:

$$x_{id}^{t+1} = x_{id}^t + v_{id}^{t+1} \tag{23}$$

2.4.2. Global factor

In PSO, to search for the optimal solution, each particle updates its flying velocity and current position iteratively according to its own flying experience and other particles' flying experience. At each generation, the velocity vector for each particle is updated based on three terms: previous velocity of the particle, the private thinking of the particle and the collaboration among the particles. After that, the particle is moved to its next position. In a word, PSO is a kind of evolutionary method through cooperation and competition among different particles. Because of the simple concept and fewer parameters to tune, PSO has been successfully used to solve real-valued optimization problems and classification problems. In Refs [24–26], PSO is used to extract classification rules. Thus, we also adopt PSO to optimize the clustering centers.

In PSO, updating strategy is a kernel problem. In [19], it is performed without considering all the particles, but only using two individuals *pbest* and *gbest*. To overcome this shortcoming, we introduce an influence factor into the iteration equation. In that case, an attractive force between different particles is defined first [27]:

$$F_{ij}^d = G(t) \cdot \frac{M_{pi}(t) \times M_{aj}(t)}{R_{ij}(t) + \varepsilon} \cdot (x_j^d(t) - x_i^d(t)) \tag{24}$$

where i and j represent two particles, $G(t)$ changes with the variation of t using the equation $G(t) = G_0 \times e^{(-\alpha \cdot (t/T_{\max}))}$. In this paper, we adopt $G_0 = 100$ and $\alpha = 20$. The value of ε is 0.01.R_{ij} is the Euclidian distance between i and j, there exists a formula $M_{ai} = M_{pi} = M_{ii} = M_i$, $i = 1, 2, ..., N$, while M_i can be calculated as

$$M_i(t) = \frac{m_i(t)}{\sum_{j=1}^{N} m_j(t)} \tag{25}$$

$$m_i(t) = \frac{fit_i(t) - worst(t)}{best(t) - worst(t)} \tag{26}$$

where fit_i is the fitness value of particle i, *worst* is the maximum fitness value in the particles, and *best* is the minimum fitness value in the particles. Furthermore, it can be seen that a better solution represents a higher attraction and moves more slowly. Hence, the total force on one individual in dth dimension can be modified as follows:

$$F_i^d(t) = \sum_{j \neq i} rand_j F_{ij}^d(t) \tag{27}$$

where $rand_j$ is uniformly distributed in the range [0,1]. Therefore, the fitness values of particles and the distances between them become more important in the updating formula. That promises a better performance for the algorithm. Thus, the update equation can be improved as

$$v_{id}^{t+1} = \omega^t \cdot v_{id}^t + c_1 \cdot r_1 \cdot (p_{id}^t - x_{id}^t) + c_2 \cdot r_2 \cdot (p_{gd}^t - x_{id}^t) + F_{id}^t \tag{28}$$

$$x_{id}^{t+1} = x_{id}^t + v_{id}^{t+1} \tag{29}$$

where F_{id}^t is the factor introduced into the formula.

The procedure of PSO using global factor is shown as follows.

Step 1: Initialization. A set of clustering centers, obtained by ACDE, are defined as the positions of the particles.
Step 2: Calculate the fitness value of the samples according to Eq. (16), set the obtained value and the current position of each particle as its objective value and p_i, set the best one and its position as the global optimum individual and p_g.
Step 3: Update the velocity and position of each particle by Eqs. (28) and (29).
Step 4: Update the inertia weight ω using Eq. (22).
Step 5: Evaluate the fitness value of each particle by Eq. (16).
Step 6: Update the personal best position and its function value.
Step 7: Update the global best position and its function value.
Step 8: If the function value and t satisfy the terminate condition, then stop; otherwise go back to step 3.

3. Experimental set and results

In order to evaluate the performance of the proposed algorithm, we apply the proposed algorithm to seven UCI datasets, one synthesized datasets and three texture images. One synthesized data is used to examine the effectiveness of clustering learning and classification learning. Classification accuracy is used to make a comparison among RBFNN [7], FRC [9], VQ+LVQ3 [10], and SCC [11], of which results are directly from [11]. A detailed analysis of time cost is also given in this part.

All experiments were carried out on a desktop computer with Intel(R) Core(TM) 2CPU (1.86 GHz) and 2 GB of RAM, Running Windows XP.

3.1. Experimental results of synthetic dataset

In this section, a two-dimensional synthetic dataset [11] is produced to test the effectiveness of clustering learning and classification of the algorithm. The distribution of dataset 1 is listed in Table 2.

From Table 2, it can be seen that the dataset has two classes and four groups. In other words, it has four centers for the whole dataset. To show whether our algorithm can reveal the underlying structure in data and meanwhile obtain good classification accuracy, the proposed algorithm is used to classify the data and is compared with SCC. The obtained clustering centers and the distribution structure are shown in Fig. 1.

In Fig. 1, samples from different classes are marked with different colors. Similarly, samples in the same class are signed with the same color. It can be seen that, data in the same class can be formed with multi-piles. The individuals with "*" are the centers gained through the simultaneous frame, which can reflect the data distribution correctly. It can be seen clearly that PSOSLCC can uncover the distribution of whole data better than SCC. In the meanwhile, the classification accuracy of SCC is 95.83%, it is less

Table 2
Synthetic dataset.

Group	Class label	Group center	Variance
Gaussian distribution 1	ω_1	(−2, 2)	(0.5, 0.5)
Gaussian distribution 2	ω_1	(2, −2)	(0.5, 0.5)
Gaussian distribution 3	ω_2	(−2, −2)	(0.5, 0.5)
Gaussian distribution 4	ω_2	(2, 2)	(0.5, 0.5)

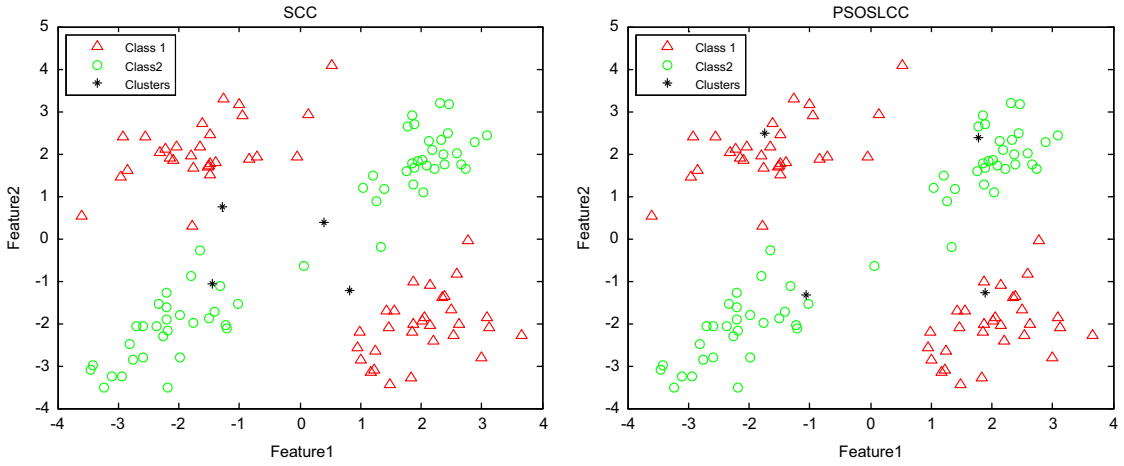

Fig. 1. Synthetic dataset.

Table 3
Benchmark datasets.

Dataset	Number	Attributes	Classes
Thyroid	215	5	3
Wine	178	13	3
Iris	150	4	3
Lung_cancer	32	56	3
Diabetes	768	8	2
Balance_scale	625	4	3
Bupa	345	6	2

Table 4
Selection of cluster number.

Dataset	k	k_1	k_2
Thyroid	3	8	3
Wine	3	4.6	3
Iris	3	7	3
Lung_cancer	3	4	2
Diabetes	2	10.3	3
Balance_scale	3	10.2	3
Bupa	2	7.7	2

than 97.33% obtained by PSOSLCC. Thus, PSOSLCC has the ability to reflect the relationship between the clusters and classes.

3.2. Experimental results of UCI datasets

In this section, seven datasets, namely Thyroid, Wine, Iris, Lung_cancer, Diabetes, Balance_scale and Bupa, are used to test the effectiveness of classification learning of the proposed algorithm. The description of the datasets is given in Table 3.

From Table 3, it can be seen that the seven datasets are all from UCI, with certain size, different dimensions, and fewer classes. They are all well-known real problems and usually used for making a comparison among some typical classification algorithms.

3.2.1. Experimental results of the basic PSO plus ACDE

In this section, we first combine the basic PSO [19] with ACDE and denote it as PSOSLCC1. In order to verify the effectiveness of ACDE, the performance of PSOSLCC1 is compared with SCC on seven UCI datasets. The experimental results are shown in Tables 4 and 5.

In Table 4, k represents the actual classes of the data, k_1 is the average value of SCC selected from the range of $[l_{\max}, c_{\max}]$. Obviously, there exists a relatively large randomness and uncertainty in the selection of k_1, while the clustering number k_2 obtained by ACDE more approximates k.

In this paper, ACDE is executed for 40 times to get 40 values, and the results may not be all the same for the whole experiment. Thus, the value with the highest frequency is selected as the optimal cluster number. From Table 4, it can be seen that k_2 can always get the same as or similar to the true cluster number. A set of clustering centers are also obtained by performing ACDE and they are corresponding to the optimal cluster number. In PSO, these clustering centers are defined as the initial positions of the particles.

Table 5
Comparison of mean classification accuracy between SCC and PSOSLCC1.

Dataset	SCC	PSOSLCC1
Thyroid	93.27	**95.70**
Wine	92.27	**95.34**
IRIS	95.20	**95.27**
Lung_cancer	42.31	**46.15**
Diabetes	74.17	**75.36**
Balance_scale	**89.77**	89.17
Bupa	**65.52**	62.3

Table 5 shows the classification accuracy of SCC and PSOSLCC1. From Table 5, it can be seen that PSOSLCC1 performs better than SCC on the most of the UCI problems used in this study, except for balance_scale and bupa.

Table 6 gives a comparison of running time of SCC and PSOSLCC1. From Table 6, it is easy to see that PSOSLCC1 reduces greatly the running time compared to SCC, which proves that adopting ACDE to determine the number of cluster of dataset is a good alternative of selecting from the range of $[l_{\max}, c_{\max}]$ utilized in SCC.

3.2.2. Experimental results of basic PSO plus global factor and ACDE

In this section, PSOSLCC is compared with PSOSLCC1 in terms of the classification accuracy and the running time. Twenty independent runs on each test problem are performed. As mentioned in Section 3.2.1, PSOSLCC1 is the basic particle swarm optimization plus ACDE based on a simultaneous learning framework for classification clustering, and PSOSLCC is the improved PSO by using the global factor plus ACDE, which is also a final version of the proposed algorithm. Mean classification accuracy of PSOSLCC and PSOSLCC1 is shown in Table 7.

Table 6
Comparison of mean running time between SCC and PSOSLCC1.

Dataset	SCC	PSOSLCC1
Thyroid	1.2659e+004	**56.94**
Wine	5.1919+003	**49.59**
Iris	4.7762e+003	**32.89**
Lung_cancer	1.0596e+003	**29.94**
Diabetes	3.5956+004	**223.57**
Balance_scale	2.4454e+004	**130.31**
Bupa	1.3326e+004	**72.69**

Table 7
Comparison of mean classification accuracy between PSOSLCC1 and PSOSLCC.

Dataset	PSOSLCC1	PSOSLCC
Thyroid	95.70	**96.07**
Wine	95.34	**95.90**
Iris	95.27	**96.13**
Lung_cancer	46.15	**50.77**
Diabetes	75.36	**76.09**
Balance_scale	**89.17**	88.36
Bupa	62.3	**65.81**

Table 8
Comparison of mean running time between PSOSLCC1 and PSOSLCC.

Dataset	PSOCAC1	PSOCAC
Thyroid	56.94	**179.39**
Wine	49.59	**170.57**
Iris	32.89	**32.89**
Lung_cancer	29.94	**210.43**
Diabetes	223.57	**751.76**
Balance_scale	130.31	**371.23**
Bupa	72.69	**199.72**

As shown in Table 7, PSOSLCC is better than PSOSLCC1 for the most problems, namely bupa, iris, lung_cancer, and diabetes in terms of classification accuracy, and it gets a similar result to PSOSLCC1 for thyroid, wine and diabetes.

Due to using the global factor, it may add the computation complexity of the PSOSLCC compared to PSOSLCC1, and this is proved by Table 8, which gives the running time of the two algorithms.

In Table 8, for all the problems, the mean running time of PSOSLCC is greater than that of PSOSLCC1. But considering the improvement of the classification rate, PSOSLCC displays an advantage over PSOLCC1.

3.2.3. Comparison of the proposed algorithm with four existed classification methods

In order to verify the effectiveness of PSOSLCC, in this part, we compare the proposed algorithm with four existing classification algorithms such as fuzzy relational classifier (FRC) [9], vector quantization and learning vector quantization (VQ+LVQ3) [10], radial basis function neural network (RBFNN) [7], and a simultaneous learning framework for clustering and classification (SCC) [11] in terms of classification accuracy, all six algorithms are executed for 20 runs independently.

Table 9 gives all statistical results of mean classification accuracy obtained by these six algorithms. It can be seen that PSOSLCC performs superiority over the other algorithms for most of the datasets. It is easy to see from Table 9 that PSOSLCC obtains a higher accuracy for most of the testing problems with less running time compared to other five classification algorithms.

Table 9
Comparison of classification accuracy between different algorithms.

Dataset	FRC	VQ+LVQ3	RBFNN	SCC	PSOSLCC1	PSOSLCC
Thyroid	80.93	88.53	44.30	93.27	95.70	**96.07**
Wine	94.31	95.11	90.11	92.27	95.34	**95.90**
Iris	93.89	89.07	95.33	95.20	95.27	**96.13**
Lung_cancer	34.61	48.57	43.84	42.31	46.15	**50.77**
Diabetes	65.10	65.98	61.40	74.17	75.36	**76.09**
Balance_scale	46.15	66.42	49.60	**89.77**	89.17	88.36
Bupa	59.94	57.80	58.08	65.52	62.3	**65.81**

Experimental results in this section just show the effect of the proposed algorithm on the problems with small scale. In the following section, we will apply the proposed algorithm to the image segmentation problem, which shows that PSOSLCC is also a possible choice for dealing with problems with a large scale.

3.3. Experimental results of texture image segmentation

Image segmentation is defined as a process of dividing an image into disjoint homogeneous regions, and these homogeneous regions usually contain similar objects of interest part of them. The extent of homogeneity of the segmented regions can be measured using some image properties (e.g., pixel intensity) [18]. In this section, we try to apply PSOSLCC to the segmentation of texture images.

In the image segmentation experiments, we use the gray-level co-occurrence matrix (GLCM) [28] and the undecimated wavelet decomposition [29] to extract texture features of each pixel from the images. For GLCM, there are many statistics that can be determined from each GLCM, such as angular second moment, contrast, correlation, sum of squares, and entropy. In this paper, we chose four statistics, including contrast, angular second moment, entropy and correlation with four directions. In this study, we set the number of gray levels at 16 and the window size at 16×16. For the undecimated wavelet decomposition, we set the window size at 21×21 for artificial texture images. Two-level wavelet transform can get seven-dimensional energy features. We combine the 16-dimension GLCM features with 7-dimension wavelet features, so each pixel could be represented by a 23-dimension** feature vector.

All the artificial texture images used in this part are 256×256 images. As shown in Fig. 2(a) and (e), Images 1 and 3 both contain two textures. Fig. 2(b) and (f) represents the true partitioning of Images 1 and 3. Image 2 contains four textures as shown in Fig. 2 (c), and Fig. 2(d) represents its true partitioning. 50% representative samples of each feature are selected as the training samples, and all samples are used as testing samples. We perform 20 independent runs on each problem, and the resulting images are shown in Figs. 3–5. Each contains four images, the first image is their true partitioning. (b), (c) and (d) are the results obtained by FRC, VQLVQ3, and PSOSLCC, respectively.

From Figs. 3 and 5, the visual results of PSOSLCC are obviously better than that of other two algorithms. But for image 2, VQLVQ3 performs the best among these three algorithms

The mean classification accuracies of artificial texture images are shown in Table 10.

From Table 10, it can be seen that the algorithm is probable to get a competitive result for the image segmentation, but considering time cost, PSOSLCC performs superiority over other algorithms obviously.

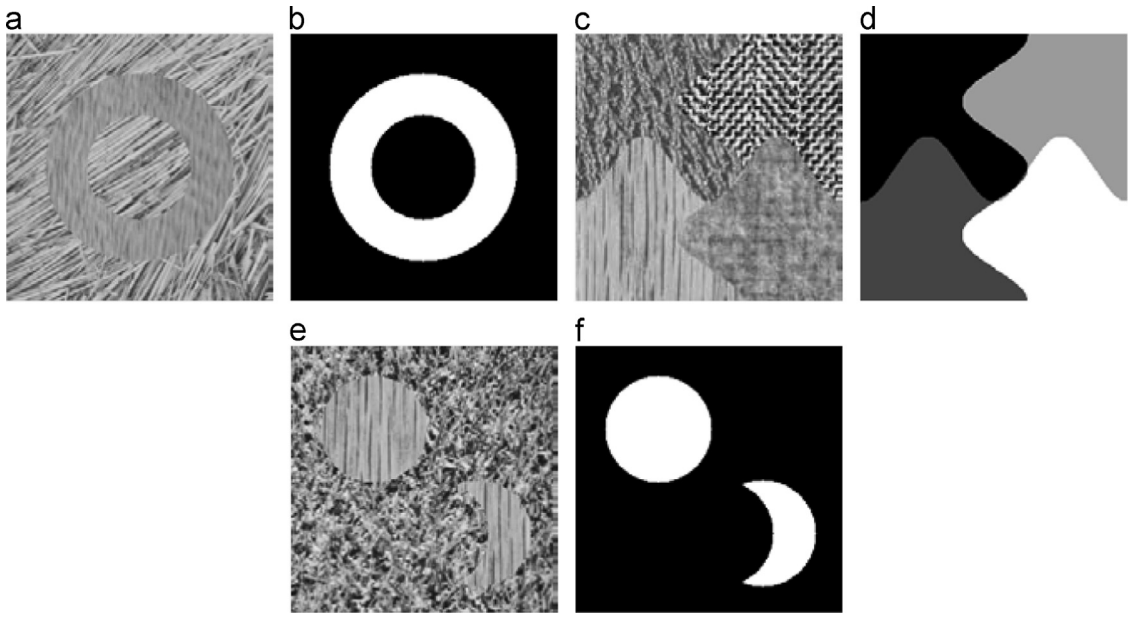

Fig. 2. Artificial texture images and their true partitioning: (a) original Image 1; (b) true partitioning of image 1; (c) original image 2; (d) true partitioning of image 2; (e) original image 3; (f) true partitioning of image 3.

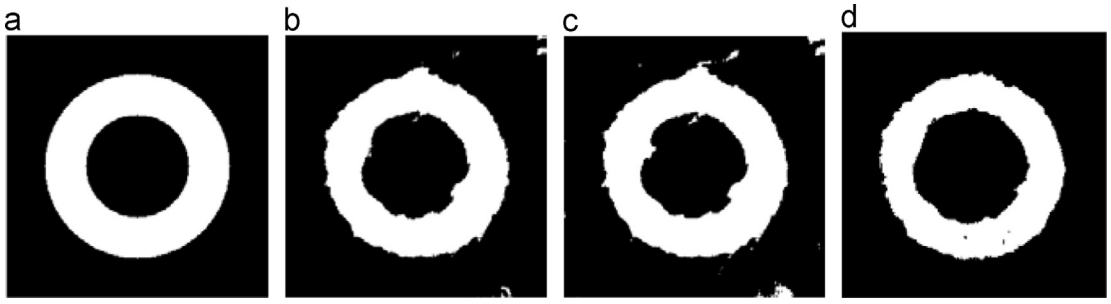

Fig. 3. Segmentation results by three classification algorithms for text image 1. (a) True partitioning of texture image 1; (b) FRC; (c) VQLVQ3; (d) PSOSLCC.

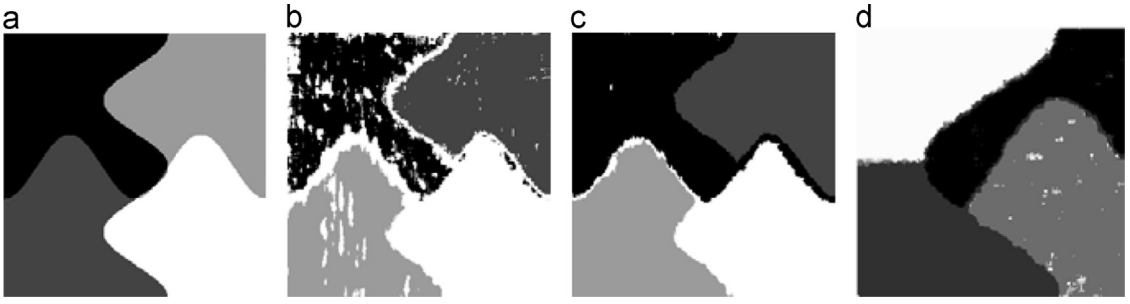

Fig. 4. Segmentation results by three classification algorithms for text image 2. (a) True partitioning of texture image 2; (b) FRC; (c) VQLVQ3; (d) PSOSLCC.

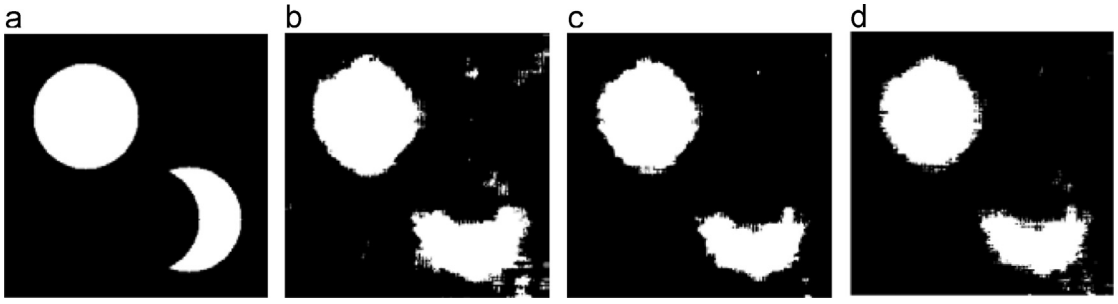

Fig. 5. Segmentation results by three classification algorithms for text image 3. (a) True partitioning of texture image 2; (b) FRC; (c) VQLVQ3; (d) PSOSLCC.

Table 10
Comparison of classification accuracy between FRC, VQLVQ3, and PSOSLCC.

Texture image	FRC	VQLVQ3	PSOSLCC
Image 1	95.21	94.62	**96.43**
Image 2	87.38	**94.53**	94.31
Image 3	95.77	98.13	**99.02**

4. Conclusions

In this paper, a novel PSOSLCC is proposed. The main ideas of the proposed algorithm include three aspects. Firstly, an automatic clustering algorithm (namely automatic clustering using improved differential evolution algorithm (ACDE) is performed for 40 runs to find out the distribution structure of data, the optimal cluster number and the corresponding clustering centers are obtained. Then for the training data selected, an improved PSO by using a global factor is used to optimize a special single objective function so as to find an optimal cluster center, by using a relational matrix established through Bayesian theorem, the relationship between the different cluster centers and classes is determined. Finally, the test dataset is classified by using the relational matrix obtained. Experimental results show that by using the automatic clustering algorithm rather than the trial and error or selecting from the range of $[l_{\max}, c_{\max}]$ not only reduces the computation complexity, but also reveals the natural distribution of the data and a global factor adopted in the process of PSO can improve the performance of the algorithm compared with SCC. It is worth noting that the time cost of the proposed algorithm can be decreased greatly compared to SCC, which makes it possible to deal with such problem with large scale as the texture images segmentation.

Conflict of interest

No conflict of interest exits in the submission of this manuscript, and manuscript is approved by all authors for publication.

Acknowledgments

This work was supported by the National Natural Science Foundation of China (Nos. 61373111, 61272279, 61103119 and 61203303); the Fundamental Research Funds for the Central Universities (Nos. K50511020014, K5051302084, K50510020011, K5051302049, and K5051302023); the Fund for Foreign Scholars in University Research and Teaching Programs (the 111 Project) (No. B07048); and the Program for New Century Excellent Talents in University (No. NCET-12–0920).

Appendix A. Supporting information

Supplementary data associated with this article can be found in the online version at http://dx.doi.org/10.1016/j.patcog.2013.12.010.

References

[1] A.K. Jain, R.P.W. Duin, J. Mao, Statistical pattern recognition: a review, IEEE Trans. Pattern Anal. Mach. Intell. PAMI 22 (1) (2000) 4–37.
[2] A.K. Jain, M.N. Murty, P.J. Flynn, Data clustering: a review, ACM Comput. Surv. 31 (3) (1999) 264–323.
[3] J.R. Quinlan, Induction of Decision Trees, Kluwer Academic Publishers, Boston, Manufactured in The Netherlands (Mach. Learn. 1 1986, 81–106).
[4] M. J. Islam, Q. M. Jonathan Wu, Majid Ahmadi, Maher A. Sid-Ahmed, Investigating the performance of Naive–Bayes classifiers and *K*-nearest neighbor classifiers, in: International Conference on Convergence Information Technology, 2007, pp. 1541–1546.
[5] G.Q. Zhang, Neural networks for classification: a survey, IEEE Trans. Syst. Man Cybernet. Part C: Appl. Rev. 30 (4) (2000) 451–462.
[6] F. De, A. Della Cioppa, E. Tarantion, Evaluation of particle swarm optimization effectiveness in classification. In: I. Bloch, A. Petrosino, A.G.B. Tettamanzi (Eds.), Springer-Verlag Berlin Heidelberg, Lecture Notes in Computer Science, Fuzzy Logic and Applications, vol. 3849, pp. 164–171.
[7] M.T. Musavi, W. Ahmed, K.H. Chan, K.B. Faris, D.M. Hummels, On the training of radial basis function classifiers, Neural Netw. 5 (4) (1992) 595–603.
[8] J.C. Bezdek, Pattern Recognition with Fuzzy Objective Function Algorithms, Plenum, New York, 1981.
[9] M Setnes, R Babuska, Fuzzy relational trained by fuzzy clustering, IEEE Trans. Syst. Man Cybernet. Part B: Cybernet. 29 (5) (1999) 619–625.
[10] S.W. Kim., B.J. Oommen, Enhancing prototype reduction schemes with LVQ3-type algorithms, Pattern Recognit. 36 (5) (2003) 1083–1093.
[11] Songcan Chen Weiling Cai, A Daoqiang Zhang, Simultaneous learning framework for clustering and classification, Pattern Recognit. 42 (7) (2009). (1248–1259).
[12] J.C. Bezdek, Pattern Recognition in Handbook of Fuzzy Computation, IOP Publishing Ltd, Boston, NY, 1998. (Chapter 6).
[13] S. Abe, Training of support vector machines with Mahalanobis kernels, in: International Conference on Artificial Networks, Lecture Notes in Computer Science, vol. 3697, 2005, 571–576.
[14] Swagatam Das, Ajith Abraham, Senior Member, Amit Konar, Automatic clustering using an improved differential evolution algorithm, IEEE Trans. Syst. 38 (1) (2008) 218–237.
[15] M.K. Pakhira, S. Bandyopadhyay, U. Maulik, Validity index for crisp and fuzzy clusters, Pattern Recognit. 37 (3) (2004) 487–501.
[16] S.C. Chen, D.Q. Zhang, Robust image segmentation using FCM with spatial constraints based on new kernel-induced distance measure, IEEE Trans. SMC Part B 34 (4) (2004) 1907–1916.
[17] M. Omran, A. Salman, A. Engelbrecht, Dynamic clustering using particle swarm optimization with application in unsupervised image classification, Pattern Anal. Appl. 8 (4) (2006) 332–344.
[18] J. Kennedy, R. Eberhart, Particle swarm optimization, in: Proceedings of the IEEE International Conference on Neural Network, IEEE Press, Piscataway, NJ, 1995, 1942–1948.
[19] J. Bratton and D. Kennedy, Defining a standard for particle swarm optimization, in: Proceedings of the IEEE Swarm Intelligence Symposium, 2007, 120–127.
[20] Y. Shi,, R. C. Eberhart, A modified Particle Swarm Optimization, Proceedings of IEEE International Conference on Evolutionary Computation, Piscataway, NJ, 69–73.1998,.
[21] A.B. Hashemi, M.R. Meybodi, A note on the learning automata based algorithms for adaptive parameter selection in PSO, Appl. Soft Comput. 11 (1) (2011) 689–705.
[22] C. Blake, E. Keogh, C.J. Merz,: UCI Repository of Machine Learning Databases (⟨http://www.ics.uci.edu/∼mlearn/MLRepository.html⟩), Department of Information and Computer Science, University of California, Irvine, CA, 1998.
[23] Chen DeBao, Zhao ChunXia, Paritcle swarm optimization with adaptive population size and its application, Appl. Soft Comput. 9 (1) (2009) 39–48.
[24] Z. Wang, X. Sun, D. Zhang, A PSO-based classification rule mining algorithm. In: D.-S. Huang, L. Heutte, M. Loog (Eds.), Springer-Verlag Berlin Heidelberg LNAI 4682, Lecture Notes in Computer Science, Advanced Intelligent Computing Theories and Applications. With Aspects of Artificial Intelligence (2007) 377–384.
[25] A.A.A. Esmin, Generating fuzzy rules from examples using the particle swarm optimization algorithm, in: Proceedings of the Seventh International Conference on Hybrid Intelligent Systems, 2007, pp. 340–343.
[26] A. Cervantes, P. Isasi, I. Galván, Binary particle swarm optimization in classification, Neural Netw. World 15 (3) (2005) 229–241.
[27] Hossein Nezamabadi-pour Esmat Rashedi, GSA Saeid Saryazdi, A Gravitational Search Algorithm, Inf. Sci. (NY) 179 (13) (2009) 2232–2248.
[28] E. Rignot, R. Kwok, Extraction of textural features in sar images: statistical model and sensitivity, Proceedings of the IGARSS, IEEE Press, New York (1990) 1979–1982.
[29] S. Fukuda, H. Hirosawa, A Wavelet-based Texture, Set applied to classification of multifrequency polarimetric SAR images, IEEE Trans. Geosci. Remote Sens. 37 (5) (1999) 2282–2286.

Ruochen Liu is currently an associate professor with the Intelligent Information Processing Innovative Research Team of the Ministry of Education of China at Xidian University, Xi'an, China. She received her Ph.D. degree from Xidian University, Xi'an, China, in 2005. Her research interests are broadly in the area of computational intelligence. Her areas of special interest include artificial immune systems, evolutionary computation, data mining, and optimization.

Yangyang Chen received her B.Sc. degree from Xidian University, Xi'an, China. She is currently working toward her M.S. degree in Xidian University. Her current research focuses on data mining.

Licheng Jiao received the Ph.D. degree from Xi'an Jiaotong University, Xi'an, China in 1990. He is currently a professor and the dean of Electronic Engineering School at Xidian University, China. His current research focuses on the intelligent information processing.

Yangyang Li received her Ph.D. degree in Pattern Recognition and Intelligent System from Xidian University, Xi'an, China, in 2007. She is currently a lecturer in the school of Electronic Engineering at Xidian University. Her current research interests include quantum-inspired evolutionary computation, artificial immune systems, and data mining.