Neural Comput & Applic (2015) 26:713–721
DOI 10.1007/s00521-014-1757-z

ORIGINAL ARTICLE

A chaos embedded GSA-SVM hybrid system for classification

Chaoshun Li · Xueli An · Ruhai Li

Received: 18 March 2013 / Accepted: 16 October 2014 / Published online: 26 October 2014

Abstract Parameter optimization and feature selection influence the classification accuracy of support vector machine (SVM) significantly. In order to improve classification accuracy of SVM, this paper hybridizes chaotic search and gravitational search algorithm (GSA) with SVM and presents a new chaos embedded GSA-SVM (CGSA-SVM) hybrid system. In this system, input feature subsets and the SVM parameters are optimized simultaneously, while GSA is used to optimize the parameters of SVM and chaotic search is embedded in the searching iterations of GSA to optimize the feature subsets. Fourteen UCI datasets are employed to calculate the classification accuracy rate in order to evaluate the developed CGSA-SVM approach. The developed approach is compared with grid search and some other hybrid systems such as GA-SVM, PSO-SVM and GSA-SVM. The results show that the proposed approach achieves high classification accuracy and efficiency compared with well-known similar classifier systems.

Keywords Support vector machine · Gravitational search algorithm · Chaotic search · Parameter optimization · Feature selection

C. Li (✉) · X. An
State Key Laboratory of Simulation and Regulation of Water Cycle in River Basin, China Institute of Water Resources and Hydropower Research, Beijing 100044, China
e-mail: csli@hust.edu.cn

C. Li · R. Li
School of Hydropower and Information Engineering, Huazhong University of Science and Technology, Wuhan 430074, China

1 Introduction

Support vector machines (SVMs) first proposed by Vapnik [1] have recently been used in a range of problems including pattern recognition, bioinformatics, text categorization and fault diagnosis [2–6]. SVM classifies data with different class labels by determining a set of support vectors that are members of the set of training inputs and outline a hyperplane in the feature space [7].

Two major problems exist when using SVM for classification: how to set the optimal parameters for SVM and how to choose the optimal feature subset of the target dataset. How to set the parameters has a direct effect on the classification accuracy. Parameters that need be optimized include the penalty parameter C and the kernel function parameters such as the gamma (γ) for the radial basis functions (RBF) kernel [8, 9]. For parameter determination, grid search is often used. The search process consists of varying parameters by a fixed step size through a wide range of values and then evaluating the performance of each combination. Because of its computational complexity, grid search is only suitable for the optimization when there are very few parameters [10]. With the development of heuristic optimization methods, certain optimization techniques such as the genetic algorithm (GA) [7], particle swarm optimization (PSO) [9], simulated annealing (SA) [11] and CMA-ES [10] have been adopted in parameter optimization for SVM.

Classification problems generally involve a number of features. However, not all of these features are equally important. Some features may be redundant or even irrelevant, and if not eliminated from classification could result in increased computational time complexity or decreased classification accuracy. In order to achieve better performance, feature selection or feature reduction is necessary for complex dataset in classification. As predicted by [12], the feature selection problem is an NP-hard combinatorial problem and requires efficient solution algorithms. Efforts have been made to develop feature selection methods, including stochastic gradient descent algorithm [12], tabu search [13] and discrete particle swarm optimization [14]. Real-value PSO was used for feature selection in kernel clustering to optimize the subsets of classes [15].

Facing the challenges of parameter optimization and feature selection in SVM classification, some researchers have tried to solve them synchronously. Huang and Wang proposed a feature selection and parameter optimization approach based on GA [7]. Huang employed a PSO-SVM model to combine the discrete PSO with the continuous-valued PSO to simultaneously optimize the input feature subsets selection and the SVM kernel parameter setting [9]. In [16], a PSO-SVM model with multi-class SVM was constructed, while PSO was employed to optimize the kernel parameters and input feature subsets together. In [16], the labels of input features and kernel parameters of SVM were represented by particles in real value in optimization process. If the value of a label was less than or equal to 0.5, then its corresponding feature was not chosen. Compared with PSO and GA, the GSA is a swarm-based meta-heuristic search algorithm based on the law of Newtonian gravity [17]. Recently, GSA has been proved to be an excellent optimization method in different kinds of applications, including parameter identification [18, 19], fuzzy model identification [20, 21], wind turbine control [22], capacitor placement optimization in radial distribution system [23], economic emission load dispatch [24], synthesis gas production [25] and so on; especially, GSA has also shown potential in handing problems of feature subset selection [26] and parameter optimization of SVM [27]. Based on the discussion, it is interesting to solve the problem of SVM in classification by using GSA.

Sarafrazi and Nezamabadi-pour have promoted the research of applying GSA to solve the mentioned problems of SVM, while GSA has been used to handle the feature selection and SVM parameter optimization problems together [28]. In their work, a GSA-SVM hybrid system was proposed, in which real-valued GSA was used to optimize parameters and discrete GSA was used to optimize feature subsets simultaneously. Although the discrete GSA is a good method for feature subset selection, many other methods have also been reported to be effective in handling this problem. Among them, chaotic search is an attractive choice.

The superiority of chaotic sequence and chaotic search has been reported widely [29–31]. Recently, chaos embedded methods have been developed and applied in parameter optimization of SVMs. In [32], an optimal selection approach for SVM parameters was put forward based on mutative scale optimization algorithm (MSCOA). Wu proposed a new PSO method that uses chaotic mappings for parameter adaptation of wavelet v-support vector machine (Wv-SVM) [33]. In [34], a new chaotic differential evolution optimization approach based on Ikeda map was proposed to optimize kernel function parameters of SVM. Besides, chaotic sequence has also been used in feature selection problems. In [35], two kinds of chaotic maps, the logistic maps and tent maps, were embedded in PSO to handle feature selection problems. Since chaotic sequences have been successfully applied in various optimization problems, it is reasonable that chaotic sequences have good potential for optimization of input feature subsets.

Although SVM classification performance has been improved significantly, there is still motivation to push this work further. This study tries a new system, a hybrid chaos embedded GSA-SVM (CGSA-SVM) by combining GSA and chaotic search. In this hybrid system, the GSA is used to optimize parameters of SVM, while the chaotic search is employed to optimize the input feature subsets. Compared with [28], we introduce chaotic search to replace the discrete GSA for feature selection. Benefiting from the properties of ergodicity and stochasticity, chaos is efficient in real-value optimization and discrete value optimization.

The remainder of this paper is organized as follows. Section 2 reviews pertinent literature on SVM and GSA. Section 3 presents in detail the developed CGSA-SVM approach for determining the parameter values for SVM and selecting feature subsets. Next, Sect. 4 compares the experimental results with those of existing approaches. Conclusions are finally drawn in Sect. 5.

2 Literature review

2.1 Support vector machine

The SVM technique was first proposed by Vapnik [1]. The principles of SVM stem from statistical learning theory. In this section, the technique is briefly introduced as follows.

Let (x_i, y_i), $i = 1,\ldots,l$, denote a set of training data, where $x_i \in R^d$ is the input data with d dimensions and $y_i \in \{-1, 1\}$is the corresponding bipolar label. A linear decision surface can be defined by the equation $f(x) = \langle w, x\rangle + b = 0$, where w is a weight vector orthogonal to the decision surface, b is an offset term and $\langle \cdot\,, \cdot\rangle$ is the inner product operator. The original formulation of SVM algorithm seeks a linear decision surface that separates the two opposite classes with a maximal margin $1/\|w\|$ by solving the following optimization problem:

$$\min_{w,b} \frac{1}{2}\|w\|^2 \tag{1}$$

Subject to $y_i(\langle w, x_i\rangle + b) \geq 1, \quad i = 1, \ldots, l.$

This optimization problem can be transformed into its corresponding dual problem:

$$W(a) = \sum_{i=1}^{l} a_i - \frac{1}{2}\sum_{i,j\neq 1}^{l} a_i a_j y_i y_j \langle x_i, x_j \rangle \tag{2}$$

With constraints: a_i, $a_j \geq 0$, $i = 1, \ldots, l$, $\sum_{i=1}^{l} y_i a_i = 0$. The Lagrange multipliers a_i can be obtained by optimizing $W(a)$.

Considering $w = \sum_{i=1}^{l} a_i y_i x_i$, the decision function is expressed as follows:

$$f(x) = \mathrm{sgn}\left(\sum_{i=1}^{l} a_i y_i \langle x_i, x_j \rangle + b\right) \tag{3}$$

In order to relax the margin constraints for the nonlinearly separable data, the slack variables are introduced into the optimization problem. The following two forms of soft margin SVMs have generally been discussed and applied:

$$\min_{w,\xi} \frac{1}{2}\|w\|^2 + C\sum_{i=1}^{l} \xi_i \tag{4}$$

Subject t$y_i(\langle w, x_i \rangle + b) \geq 1 - \xi_i, \quad i = 1, \ldots, l, \quad \xi_i \geq 0$ where ξ_i, $i = 1,\ldots,l$, are slack variables and C is the penalty parameter of error.

In practice, most of the problems are linearly inseparable, even with soft margin SVM. Thus, the input data are mapped into a high dimensional feature space, in which the data are sparse and possibly more separable. Suppose the mapping function is $\varphi(\cdot)$, then the inner product $\langle x_i, x_j \rangle$ can be replaced by $\langle \varphi(x_i), \varphi(x_j) \rangle$. Given a symmetric and positive kernel function $K(x, y)$ which satisfies Mercer's theorem, the inner product in feature space could be expressed by $K(x_i, x_j) = \langle \varphi(x_i), \varphi(x_j) \rangle$. Consequently, the decision function becomes:

$$f(x) = \mathrm{sgn}\left(\sum_{i=1}^{l} a_i y_i K(x_i, x_j) + b\right) \tag{5}$$

The radial basis function is a common kernel function given as:

$$K(x_i, x_j) = \exp\left(-\gamma \|x_i - x_j\|^2\right) \tag{6}$$

where γ is the parameter of the kernel function.

The performance of an SVM can be controlled through the penalty parameter C and the kernel parameter γ. These parameters influence the number of support vectors and the maximization margin of the SVM.

2.2 Gravitational search algorithm (GSA)

GSA was first proposed by Rashedi [17]. Assumed there are N agents (masses), the position of the ith agent is $L_i = (l_i^1, \ldots l_i^d, \ldots, l_i^n)$, $i \in \{1,\ldots,N\}$. The mass of each agent is calculated according to the fitness function value of the agent. It is easy to visualize that a good agent possesses a strong gravitational field and moves slowly, as it has a larger inertial mass.

Based on fitness function value, the mass of the ith agent in the kth iteration is defined:

$$M_i(k) = \frac{\mathrm{fit}_i(k) - \mathrm{worst}(k)}{\sum_{j=1}^{N} (\mathrm{fit}_j(k) - \mathrm{worst}(k))} \tag{7}$$

where $\mathrm{fit}_i(k)$ is the fitness function value of the ith agent. For an optimization problem about seeking minimal value, $\mathrm{worst}(k) = \max_{j\in[1,\ldots,N]} \mathrm{fit}_j(k)$.

According to Newton gravitation theory, the dth dimension of the force acting on the ith mass from the jth mass in the kth iteration is defined as:

$$F_{ij}^d(k) = G(k)\frac{M_i(k) \times M_j(k)}{||\mathbf{L}_i(k), \mathbf{L}_j(k)||_2}(l_j^d(k) - l_i^d(k)) \tag{8}$$

where M_i and M_j are masses of agents, $\mathbf{L}_i$ is the position vector of agent, l_i^d is the dth element in $\mathbf{L}_i$ and G(k) is the gravitational constant in the kth iteration.

It must be pointed out that the gravitational constant G(k) is important in determining the performance of GSA, and G(k) is defined as a function of iteration number k:

$$G(k) = G_0 \cdot \exp\left(-\beta \cdot \frac{k}{\text{max\_iter}}\right) \tag{9}$$

where G_0 is the initial value, β is a constant, k is the iteration number, max_iter is the maximum number of iterations.

For the ith agent, the randomly weighted sum of the forces exerted from other agents can be calculated by:

$$F_i^d(k) = \sum_{j\neq i} rand_j F_{ij}^d(k) \tag{10}$$

Based on the law of motion, the acceleration of the ith agent can be calculated by:

$$a_i^d(k) = \frac{F_i^d(k)}{M_i(k)} \tag{11}$$

where M_i is the inertial mass of the ith agent.

Thus, the searching strategy for this concept can be described by following equations, which describe velocity and location, respectively:

$$v_i^d(k+1) = \mathrm{rand}_i \cdot v_i^d(k) + a_i^d(k) \tag{12}$$

$$l_i^d(k+1) = l_i^d(k) + v_i^d(k+1) \tag{13}$$

In above equations, l_i^d represents the position of ith agent in the dth dimension, v_i^d is the velocity, a_i^d is the acceleration and rand_i is a random number among [0, 1].

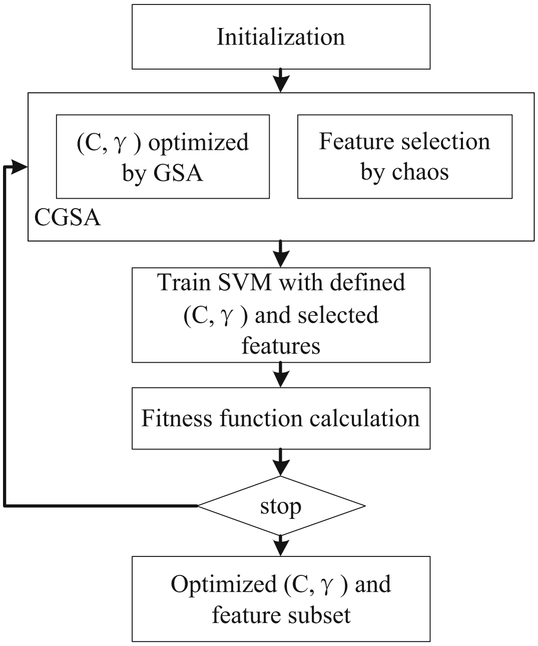
Initialization
(C, γ) optimized by GSA
Feature selection by chaos
CGSA
Train SVM with defined (C, γ) and selected features
Fitness function calculation
stop
Optimized (C, γ) and feature subset

Fig. 1 The scheme of the proposed CGSA-SVM

3 The hybrid CGSA-SVM classifier

Parameter optimization and feature selection are of great importance for improving classification ability of SVM. As reported in literature, PSO, GA, GSA and other optimization methods have been used to improve SVM. This study develops a hybrid approach based on chaotic search and GSA, termed CGSA-SVM, for parameter determination and feature selection in SVM. The scheme of the proposed CGSA-SVM can be illustrated by Fig. 1. In this scheme, GSA is adopted to optimize the key parameter pair (C, γ) of SVM, while chaotic search is implemented for the feature subset optimization. For the parameter optimization process, GSA is suitable for the coding and searching of the real-valued parameter. For the feature selection process, if the needed features must be selected from a total of n features, each feature would be assigned a variable valued between 0 and 1. If the value of a variable is less than or equal to 0.5, then its corresponding feature is not chosen. Conversely, if the value of a variable is >0.5, then its corresponding feature is chosen. Chaotic sequence is used to make variance and represents different selection of feature subsets. The solution representation is shown in Fig. 2.

3.1 Feature selection based on chaotic search

Chaos is a bounded unstable dynamic behavior that exhibits sensitive dependence on initial conditions and includes infinite unstable periodic motions in nonlinear systems. Benefiting from the properties of ergodicity and stochasticity, chaos has been employed in numerous optimization problems. Considering that the feature selection problem is an optimization problem with searching range of [0, 1], chaos is well suited to handle this problem.

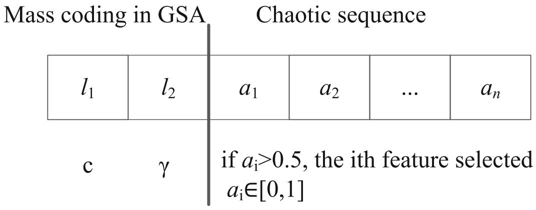

Fig. 2 Solution representation

A chaotic map with n dimension is a discrete-time dynamical system that can be expressed as:

$$cx_i^{(k+1)} = f(cx_i^{(k)}), \quad i = 1, \ldots, n \tag{14}$$

By defining the initial state of $cx_i^{(0)}$, a chaotic sequence can be obtained through running the system function. A chaotic sequence is usually denoted by $\{cx_i^{(k)}, k = 0,1,2,\ldots\}$.

As a well-known chaotic map, the logistic map was introduced by Robert May in 1976 [36]. It is often used to explain how complex behavior can arise from a simple deterministic dynamic system without any stochastic disturbance. This map is defined as:

$$\begin{aligned} cx^{(k+1)} &= a \cdot cx^{(k)}(1 - cx^{(k)}) \quad \text{for} \quad 0 < a \le 4, \\ cx(k) &\in (0, 1) \end{aligned} \tag{15}$$

where $cx^{(k)}$ is the kth chaotic number, and a is the control parameter that determine the chaotic behavior of the dynamic system. Typically, $a = 4$.

In this paper, logistical map is adopted to represent the selection of features as follows:

$$F_i = \begin{cases} 1 & cx_i^{(k)} > 0.5 \\ 0 & cx_i^{(k)} \le 0.5 \end{cases} \tag{16}$$

where $F_i = 1$ means the ith feature is selected; $F_{\mathrm{i}} = 0$ means the ith feature is not selected.

3.2 Objective function

Classification accuracy and the number of selected features are the two criteria used to design an objective function for classification [28]. In this paper, an objective function is proposed, which combines the two goals into one by setting weights for the goals. The weight accuracy can be adjusted to a high value (such as 100 %) if accuracy is the most important.

$$\text{obj} = w \cdot \text{SVM\_acc} + (1 - w)\left(1 - \frac{\sum_{i=1}^{n_f} F_i}{n_f}\right) \tag{17}$$

where SVM_acc is the SVM classification accuracy, n_{f} is the number of features, w is the weight of SVM

Fig. 3 The pseudo-code of the CGSA-SVM hybrid system

```
Initialize a 2-dimensional swarm $L_i$ for $(C, \gamma)$,i=1,…,$N$;
Initialize a n-dimensional vector $cx_i^{(0)} \in (0,1)$,i=1,…,n;
repeat
for each mass i=1,2,…,$N$
    Get the feature subset: if iteration<$max\_it/2$, get feature subset
      with chaos variables by calculating Eq. (16); else set the feature subset with the best
      feature subset;
    Decoding position to get $(C, \gamma)$ and then train SVM and calculate fitness function Eq. (17);
    Update the best $(C, \gamma)$ and the best feature subset.
    Calculate mass using Eq. (7);
    Calculate gravitational force using Eq. (10);
    Calculate acceleration using Eq. (11);
    Update the velocity using Eq. (12);
    Update the position using Eq. (13);
    Update chaotic sequence by Eq. (15);
End
Until the number of iteration reaches $max\_it$
Retrain SVM parameters and selected feature subset.
```

classification accuracy and $F_i \in \{0, 1\}$, where "1" represents that feature i is selected and "0" represents that feature *i* is not selected.

The weight w is used to control the significance of classification accuracy and number of features. For any SVM-based classifier, the first objective is always improving classification accuracy; thus, w is usually set to be near one. We set $w = 0.9$ in the following experiments.

3.3 The proposed CGSA-SVM

Based on the above description, the proposed CGSA-SVM algorithm can be illustrated as shown in Fig. 3. In this algorithm, some further explanation is needed for the number of iteration where chaotic search is used. We set the number of iterations of chaotic search to be one half of the maximum number of iterations in GSA. In the first batch of iterations, chaotic search is adopted to search feature subsets by using its ergodicity and stochasticity. At this stage, feature selection by chaotic search and SVM parameter search by GSA take place simultaneously. In each iteration, feature subset represented by chaotic sequence is used to reduce the original dataset. And then, SVM parameter optimization is conducted on reduced dataset. In the second batch of iterations, the optimal feature subset of the first batch is used to reduce dataset, and SVM parameter optimization is going on with the reduced dataset. This design will help to maintain the stability of GSA in searching optimal parameters.

4 Experiments

In order to verify the performance of the proposed CGSA-SVM, classification experiments were designed. The datasets used in the experimentation were all obtained from the well-known machine learning data repository of UCI Machine Learning Repository, Center for Machine Learning and Intelligent Systems, University of California [37]. The number of features, number of instances and actual number of classes in the datasets are shown in Table 1.

Table 1 Dataset from the UCI repository

No.	Dataset	ID	Number of features	Number of instances	Number of classes
1	Australian	AUS	15	653	2
2	Breast cancer	BRE	10	683	2
3	German	GER	30	1,000	2
4	Ionosphere	IO	34	351	2
5	Iris	IR	4	150	3
6	Pima	PM	8	768	2
7	Vehicle	VEH	18	846	4
8	Vowel	VW	10	528	11
9	Wine	WN	13	175	3
10	Mammographic	MAM	5	829	
11	Teaching assistant evaluation	TAE	5	151	3
12	Hayes-Roth	HR	5	160	3
13	Mushroom	MR	22	8,124	2
14	Spambase	SB	57	4,601	2

The k-fold method [38] presented by Salzberg was employed in the experiments. In this study, the value of k was set to 10. Thus, a dataset was split into 10 parts, with nine data parts used as training data and the last one used as testing data in SVM classification experiments. For each UCI dataset, the experiments were repeated 10 times, so that each of ten parts would be used as testing data to verify the performance of the hybrid SVM system. The ten

Table 2 Experimental results and accuracy statistics of CGSA-SVM in the UCI datasets

Dataset	Accuracy (%)	Optimized C	Optimized γ	No. of features
AUS	100 ± 0	453.44 ± 242.13	13.92 ± 8.24	5.20 ± 1.98
BRE	100 ± 0	648.99 ± 312.39	10.99 ± 9.34	4.20 ± 1.22
GER	95.50 ± 5.79	566.53 ± 211.06	10.51 ± 7.84	6.40 ± 1.89
IO	84.41 ± 11.35	407.83 ± 310.28	8.48 ± 6.84	13.80 ± 4.02
IR	100 ± 0	596.47 ± 271.77	9.35 ± 8.13	3.10 ± 0.87
PM	100 ± 0	545.47 ± 258.49	7.93 ± 8.84	4.70 ± 2.00
VEH	99.63 ± 0.82	423.08 ± 274.44	9.98 ± 8.41	7.90 ± 2.18
VW	100 ± 0	629.29 ± 227.30	4.78 ± 3.93	5.50 ± 2.06
WN	100 ± 0	527.97 ± 277.90	7.66 ± 4.85	5.40 ± 1.89

classification accuracy results for a dataset were recorded, and the mean accuracy was used to compare with the classification accuracy of other methods in literature.

In the first group of experiments, the classification results of CGSA-SVM were compared with existing results of similar methods in literature. In this part of experiments, parameters of CGSA-SVM were set as: the size of population $N = 10$, the maximum number of iterations max_it = 50, G0 = 100, $\beta = 20$. Therefore, for the proposed CGSA-SVM, the number of objective function calculations is equal to 500. In comparative algorithms GSA-SVM, PSO-SVM and GA-SVM as reported in [7, 8, 28], the number of objective function calculations was 100, 2,000 and 5,000, respectively. The number of objective function calculations by CGSA-SVM was much low than PSO-SVM and GA-SVM, but higher than GSA-SVM. This number was chosen to make the comparison as reliable as possible and to examine the classification capability of the proposed approach more thoroughly.

Table 2 presents the accuracy results of CGSA-SVM (mean, standard deviation, optimized γ, optimized C and number of selected features) applied to the UCI datasets over 10 runs. The results obtained by CGSA-SVM were compared with grid search-based SVM (Grid-SVM) [7], GA-SVM [7], PSO-SVM [8] and GSA-SVM [28] in Table 3, while the results of latter methods were cited from literature. The results in Table 3 show that the proposed approach is more accurate than the other methods for all datasets except for "Ionosphere." In some dataset, PSO-SVM performs the same as CGSA-SVM. However, the number of objective function calculations used in CGSA-SVM was much lower than PSO-SVM: The former was 500, compared with the latter's 2,000. The comparison between GA-SVM and PSO-SVM shows that the proposed algorithm is more efficient and accurate in classification.

By comparing CGSA-SVM with GSA-SVM [28] in Table 3, it can be seen that the proposed chaos embedded hybrid system is more accurate in classification. However, it also has a higher number of objective function calculations. In next group of experiments, the number of objective function calculations was set to 100, for a better comparison with GSA-SVM.

Table 3 Classification accuracy comparison between different methods

Dataset	Grid-SVM [7]	GA-SVM [7]	PSO-SVM [8]	GSA-SVM [28]	CGSA-SVM
AUS	84.84	88.10	91.03	95.71	100
BRE	96.64	96.19	99.18	95.57	100
GER	75.30	85.60	81.62	94.51	95.5
IO	93.08	98.56	99.01	96.01	84.41
IR	96.00	100	99.20	–	100
PM	76.69	81.50	82.68	94.09	100
VEH	84.28	84.06	89.83	–	99.63
VW	98.91	99.30	100	–	100
WN	96.60	–	100	–	100
Average	89.15	91.66	93.62	95.18	97.73

– Approach did not use the dataset for test

In order to further verify the performance of CGSA-SVM, we compared PSO-SVM with CGSA-SVM by running them in the same circumstance. In this part of experiments, parameters of CGSA-SVM were set as: the size of population $N = 5$, the maximum iteration max_it = 20, G0 = 100, $\beta = 20$. Therefore, for the proposed CGSA-SVM, the number of objective function calculations was equal to 100. For PSO-SVM, both the cognition learning factor c_1 and the social learning factor c_2 were set to 2, and the number of particles and generations was set to be 8 and 200, respectively. Thus, the number of objective function calculations was 1,600. These two approaches were applied for all the 14 UCI datasets, and the k-fold method was used in experiments with $k = 10$.

Table 4 shows the comparison between CGSA-SVM and PSO-SVM in classification accuracy and number of selected features. The results of Table 4 were calculated based on 10 repeated experiments for each dataset, with the mean and standard deviation analyzed. In order to examine

Table 4 Comparison of CGSA-SVM and PSO-SVM and results of pair *t* tests

Dataset	PSO-SVM	CGSA-SVM	*p* value of pair *t* test	Significance
AUS	88.67 ± 3.92	99.85 ± 0.46	<0.001	Extremely significant
BRE	98.80 ± 1.17	99.25 ± 0.78	<0.001	Extremely significant
GER	78.5 ± 3.77	91.3 ± 6.48	<0.001	Extremely significant
IO	95 ± 7.21	82.64 ± 14.23	0.0248	Very significant
IR	99.33 ± 2.10	100 ± 0	0.3305	Not quite significant
PM	79.47 ± 2.07	100 ± 0	<0.001	Extremely significant
VEH	82.07 ± 4.84	99.02 ± 0.96	<0.001	Extremely significant
VW	93.40 ± 14.62	98.40 ± 5.03	0.3202	Not quite significant
WN	100 ± 0	97.5 ± 4.37	0.0871	Not quite significant
MAM	86.34 ± 3.43	100 ± 0	<0.001	Extremely significant
TAE	67.85 ± 13.14	99.28 ± 2.25	<0.001	Extremely significant
HR	76.15 ± 11.14	100 ± 0	<0.001	Extremely significant
MR	81.32 ± 13.65	98.67 ± 3.71	<0.001	Extremely significant
SB	93.37 ± 2.09	99.43 ± 0.81	<0.001	Extremely significant

Confidence level $\alpha = 0.05$

the effectiveness of experimental results, pair *t* test was used for analysis. Table 4 shows that the proposed approach performs better than PSO-SVM on most datasets, with the exception of "Ionosphere" and "Wine." On most datasets, CGSA-SVM performed significantly better than PSO-SVM. Results of pair *t* tests show that there is significant difference between the results of these two approaches, which proves the effectiveness of the results.

In this part of experiments, the results of PSO-SVM are slightly different from those in [8]. The average accuracy of the first nine datasets in [8] is 93.62 %, compared with 90.59 % which we got on the same datasets. The reason would be that the number of evaluations of fitness function in [8] was 2000, while this number had been decreased to 1,600 in our experiments. It also shows that this type of heuristic search algorithm-based system is sensitive to the number of fitness evaluations.

Figure 4 exhibits the average iteration processes of PSO-SVM and CGSA-SVM, where the maximum numbers of iterations are 200 and 20, respectively. The iteration processes are based on results of executing these two approaches 10 times on all 14 UCI datasets. From Fig. 4, it is manifest that CGSA-SVM could find the best fitness value more efficiently and effectively.

Comparing the results of CGSA-SVM in these experiments with those of GSA-SVM in literature, it can be seen that CGSA-SVM is more accurate when the number of fitness evaluations for these two approaches was both set as 100. The average accuracy of CGSA-SVM is 97.53 % in Table 4, which is greater than the 95.18 % of GSA-SVM in Table 3.

Experiments of classification on UCI datasets show the superiority of the proposed approach over the existing similar methods. Compared with PSO and GA, the GSA has many advantages [28]. Among other advantages, the GSA is memoryless, which makes it robust to escape from local optima, and the direction of any agent is adjusted based on the position of all agents, which makes it more capable of exploring the search space. The chaos embedded GSA-SVM has inherited the advantage of GSA in searching SVM parameters, so it is acceptable that CGSA-SVM performed better than PSO-SVM and GA-SVM.

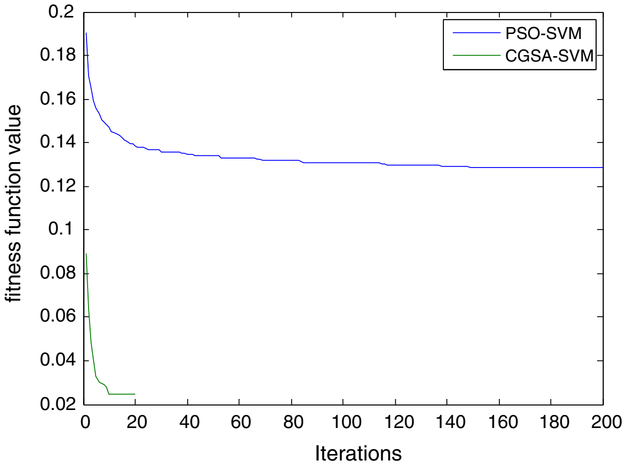

Fig. 4 Average iteration process of classification on 14 UCI dataset

The difference between CGSA-SVM and GSA-SVM is the implement of feature selection method. CGSA-SVM employs chaotic sequence to search for the optimal feature subset, while GSA-SVM uses binary GSA for this task. The chaotic sequence possesses the advantages of ergodicity and stochasticity, thus allowing CGSA-SVM to search in the feature space more efficiently.

5 Conclusions

In order to solve the problems of SVM facing in classification, this study presents a hybrid SVM system based on chaotic search and gravitational search algorithm. The proposed CGSA-SVM is capable of searching for the optimal SVM parameters and the optimal feature subset simultaneously. Fourteen UCI datasets were employed in experiments to test the performance of the proposed CGSA-SVM system. Comparison of the obtained results with those of other approaches demonstrates that the developed CGSA-SVM approach has a better classification accuracy than others tested. With 500 evaluations of fitness function, the CGSA-SVM obtains much higher classification accuracy than GSA-SVM, PSO-SVM and GA-SVM, although evaluations of fitness function of latter approaches are 2,000 and 5,000. With the same evaluations of fitness function, CGSA-SVM performs better than GSA-SVM. The experimental results make it convincible that the proposed approach in this paper is an efficient and effective classification method.

Acknowledgments This paper is supported by the Open Research Fund of State Key Laboratory of Simulation and Regulation of Water Cycle in River Basin,China Institute of Water Resources and Hydropower Research (Grant No. IWHR-SKL-201220), the National Natural Science Foundation of China (Grant Nos. 51479076, 51109088, 51309258), the Research Fund for the Doctoral Program of Higher Education of China (Grant No. 20110142120020) and the Fundamental Research Funds for the Central Universities, HUST (No. 2013QN114).

References

1. Vapnik VN (1995) The nature of statistical learning theory. Springer, New York
2. Guo XC, Yang JH, Wu CG, Wang CY, Liang YC (2008) A novel LS-SVMs hyper parameter selection based on particle swarm optimization. Neurocomputing 71:3211–3215
3. Wu Q, Wu Sh, Liu J (2010) Hybrid model based on SVM with Gaussian loss function and adaptive Gaussian PSO. Eng Appl Artif Intell 23:487–494
4. Shin K-S, Lee T-S, Kim H-J (2005) An application of support vector machines in bankruptcy prediction model. Expert Syst Appl 28:127–135
5. Zhang X, Zhou J, Wang C et al (2012) Multi-class support vector machine optimized by inter-cluster distance and self-adaptive deferential evolution. Appl Math Comput 218:4973–4987
6. Yuan S-F, Chu F-L (2007) Fault diagnostics based on particle swarm optimization and support vector machines. Mech Syst Signal Process 21:1787–1798
7. Huang C-L, Wang C-J (2006) A GA-based feature selection and parameters optimization for support vector machines. Expert Syst Appl 31:231–240
8. Lin S-W, Ying K-C, Chen S-C et al (2008) Particle swarm optimization for parameter determination and feature selection of support vector machines. Expert Syst Appl 35:1817–1824
9. Huang Ch-L, Dun J-F (2008) A distributed PSO–SVM hybrid system with features selection and parameter optimization. Appl Soft Comput 8:1381–1391
10. Friedrichs F, Igel Ch (2005) Evolutionary tuning of multiple SVM parameters. Neurocomputing 64:107–117
11. Boardman M, Trappenberg T (2006) A heuristic for free parameter optimization with support vector machines. In: International joining conference on neural networks. IEEE Computer Society Press, Silver Springer, MD, pp 1337–1344
12. Gadat S, Younes L (2007) A stochastic algorithm for feature selection in pattern recognition. J Mach Learn Res 8:509–547
13. Zhang H, Sun G (2002) Feature selection using tabu search method. Pattern Recognit 35:701–711
14. Unler A, Murat A (2010) A discrete particle swarm optimization method for feature selection in binary classification problems. Eur J Oper Res 206:528–539
15. Das S, Abraham A, Konar A (2008) Automatic kernel clustering with a Multi-Elitist Particle Swarm Optimization Algorithm. Pattern Recognit Lett 29:688–699
16. Ranaee V, Ebrahimizadeh A, Ghaderi R (2010) Application of the PSO–SVM model for recognition of control chart patterns. ISA Trans 49(4):577–586
17. Rashedi E, Nezamabadi-pour H, Saryazdi S (2009) GSA: a gravitational search algorithm. Inf Sci 179(13):2232–2248
18. Li C, Zhou J (2011) Parameters identification of hydraulic turbine governing system using improved gravitational search algorithm. Energy Convers Manag 52(1):374–381
19. Li C, Li H, Kou P (2014) Piecewise function based gravitational search algorithm and its application on parameter identification of AVR system. Neurocomputing 124:139–148
20. Li C, Zhou J, Fu B et al (2012) T-S fuzzy model identification with gravitational search based hyper-plane clustering algorithm. IEEE Trans Fuzzy Syst 20:305–317
21. Li C, Zhou J, Xiao J, Xiao H (2013) Hydraulic turbine governing system identification using T-S fuzzy model optimized by chaotic gravitational search algorithm. Eng Appl Artif Intell 26(9):2073–2082
22. Chatterjee A, et al (2014) A gravitational search algorithm (GSA) based photo-voltaic (PV) excitation control strategy for single phase operation of three phase wind-turbine coupled induction generator. Energy. doi:10.1016/j.energy.2014.07.037
23. Shuaib YM, Kalavathi MS, Rajan CCA (2015) Optimal capacitor placement in radial distribution system using gravitational search algorithm. Int J Electr Power Energy Syst 64:384–397
24. Jiang S, Ji Z, Shen Y (2014) A novel hybrid particle swarm optimization and gravitational search algorithm for solving economic emission load dispatch problems with various practical constraints. Int J Electr Power Energy Syst 55:628–644
25. Ganesan T, Elamvazuthi I, Ku Shaari KZ, Vasant P (2013) Swarm intelligence and gravitational search algorithm for multi-objective optimization of synthesis gas production. Appl Energy 103(2013):368–374
26. Han X, Chang X, Quan L et al (2014) Feature subset selection by gravitational search algorithm optimization. Inf Sci 281:128–146
27. Zhang W, Niu P, Li G et al (2013) Forecasting of turbine heat rate with online least squares support vector machine based on gravitational search algorithm. Knowl Based Syst 39:34–44
28. Sarafrazi S, Nezamabadi-pour H (2013) Facing the classification of binary problems with a GSA-SVM hybrid system. Math Comput Model 57:270–278
29. Li C, Zhou J, Kou P et al (2012) A novel chaotic particle swarm optimization based fuzzy clustering algorithm. Neurocomputing 83:98–109
30. Li C, Zhou J, Xiao J, Xiao H (2012) Parameters identification of chaotic system by chaotic gravitational search algorithm. Chaos, Solitons Fractals 45(4):539–547
31. Liu B, Wang L, Jin Y et al (2005) Improved particle swarm optimization combined with chaos. Chaos, Solitons Fractals 25:1261–1271

32. Hu Y, Zhang H (2012) Chaos Optimization Method of SVM Parameters Selection for Chaotic Time Series Forecasting. Phys Procedia 25:588–594
33. Wu Q (2011) A self-adaptive embedded chaotic particle swarm optimization for parameters selection of Wv-SVM. Expert Syst Appl 38:184–192
34. Dos Santos GS, Luvizotto LGJ, Mariani VC, Coelho LDS (2012) Least squares support vector machines with tuning based on chaotic differential evolution approach applied to the identification of a thermal process. Expert Syst Appl 39:4805–4812
35. Chuang L, Yang C, Li J (2011) Chaotic maps based on binary particle swarm optimization for feature selection. Appl Soft Comput 11:239–248
36. May R (1976) Simple mathematical models with very complicated dynamics. Nature 261:459–467
37. Hettich S, Blake C, Merz C (1998) UCI repository of machine information and computer sciences. http://www.ics.uci.edu/~mlearn/MLRepository.html
38. Salzberg SL (1997) On comparing classifiers: pitfalls to avoid and a recommended approach. Data Min Knowl Disc 1:317–327