



Adaptive CCR-ELM with variable-length brain storm optimization algorithm for class-imbalance learning

Jian Cheng¹ · Jingjing Chen¹ · Yi-nan Guo¹ · Shi Cheng¹ · Linkai Yang¹ · Pei Zhang¹

Published online: 27 February 2019
© Springer Nature B.V. 2019

Abstract

Class-specific cost regulation extreme learning machine (CCR-ELM) can effectively deal with the class imbalance problems. However, its key parameters, including the number of hidden nodes, the input weights, the biases and the tradeoff factors are normally generated randomly or preset by human. Moreover, the number of input weights and biases depend on the size of hidden layer. Inappropriate quantity of hidden nodes may lead to the useless or redundant neuron nodes, and make the whole structure complex, even cause the worse generalization and unstable classification performances. Based on this, an adaptive CCR-ELM with variable-length brain storm optimization algorithm is proposed for the class imbalance learning. Each individual consists of all above parameters of CCR-ELM and its length varies with the number of hidden nodes. A novel merge operator is presented to incorporate two parent individuals with different length and generate a new individual. The experimental results for nine imbalance datasets show that variable-length brain storm optimization algorithm can find better parameters of CCR-ELM, resulting in the better classification accuracy than other evolutionary optimization algorithms, such as GA, PSO, and VPSO. In addition, the classification performance of the proposed adaptive algorithm is relatively stable under varied imbalance ratios. Applying the proposed algorithm in the fault diagnosis of conveyor belt also proves that ACCR-ELM with VLen-BSO has the better classification performances.

Keywords Variable-length · Adaptive · Brain storm optimization algorithm · Class-specific cost regulation extreme learning machine · The class imbalance problems

1 Introduction

In many real-world classification problems, the total number of data for a class is less than the other class (He and Ma 2013; Shen et al. 2018). We call it the imbalance data sets. Classifying the imbalance data is a challenging problem that has attracted growing attention of the scholars. Many outstanding works, normally grouped into data- and algorithm-based strategies (Liu et al. 2009), had been done on the class imbalance learning. Data-based strategies weaken the imbalance degree of data by over-sampling for

minority samples and under-sampling for majority samples. Synthetic minority over-sampling technique (SMOTE) (Nitesh et al. 2002) and SMOTE-based improved sampling algorithms were the popular data-based strategies (Ramentol 2012; Zeng et al. 2009; Gao et al. 2011). Algorithm-based strategies improve the traditional classification algorithms according to the data features, with the purpose of enhancing the classification performances. The cost-sensitive boosting methods (Li and Mao 2014) updated the weight of AdaBoost in terms of the classification cost. Kernel-based learning methods were employed to adjust the classification boundary between minority and majority samples (Wu and Chang 2005; Wu et al. 2016). Three approaches including the boundary movement, the biased penalty and the class-boundary alignment were presented to adjust boundary skews (Wu and Chang 2005). In the kernel boundary alignment algorithm, the kernel matrix was adjusted in terms of the skew distribution of dataset (Wu et al. 2016). A weighted

✉ Yi-nan Guo
guoyinan@cumt.edu.cn
Jian Cheng
chengjian@cumt.edu.cn

¹ School of Information and Control Engineering, China University of Mining and Technology, Xuzhou 221008, Jiangsu, China

extreme learning machine (weighted-ELM) (Zong et al. 2013) was presented to strengthen the role of minority class by assigning an extra weight to each minority sample. Xiao et al. (2017) introduced class-specific regulation cost to penalize the misclassified data, and then proposed a class-specific cost regulation extreme learning machine (CCR-ELM), together with its kernel-based extension, to solve binary or multiclass classification problems with imbalance distribution.

In recent years, extreme learning machine (ELM) has become an attention-attracting method due to its faster speed and better generalization in regression and classification applications. It provides a unified learning platform with a widespread type of feature mappings by employing generalized single hidden layer feed-forward networks (SLFNs) and least-square-based learning algorithm (Huang et al. 2005). The input weights and hidden biases of SLFNs are chosen randomly. Corresponding output weights are analytically determined by generalized inverse operation of output matrix in the hidden layer. Some improved ELM algorithms were proposed to solve different kinds of regression or classification problems. Online sequential extreme learning machines employed additive hidden nodes with radial basis function in SLFNs, with the purpose of better generalization for dealing with an online data flow or concept-drifting data stream (Rong et al. 2009; Zhao et al. 2012; Mirza et al. 2015). Incremental extreme learning machine (Yang et al. 2017) randomly added hidden nodes and manually adjusted the output weights linking the hidden layer and the output layer. Semi-supervised ELM and unsupervised ELM fit for tackling the classification problems that collecting a large amount of labeled data is hard and time-consuming (Huang et al. 2014). For the class imbalance problems, weighted ELM (W-ELM) (Zong et al. 2013), total error rate ELM (TER-ELM) (Mirza et al. 2013), and class-specific cost regulation extreme learning machine (CCR-ELM) (Xiao et al. 2017) were presented, and CCR-ELM show better performances.

All above ELM-based learning algorithms may result in ill-condition problems due to the randomly selected input weights and hidden biases. Non-optimal or redundant input weights and hidden biases also make them responding slowly (Huang et al. 2005) and lead to worse generalization. In view of the shortcomings above, some population-based intelligence optimization algorithms, such as genetic algorithm (GA) (Ertam and Avci 2017), particle swarm optimization (PSO) (Han et al. 2013), fruit fly optimization (FOA) (Li and Zhang 2014) and krill herd algorithm (KH) (Guo et al. 2017), were introduced to find the optimal parameters of the activation function in ELM-based learning algorithms, with the purpose of improving the classification accuracy.

In this paper, we focus on the class imbalance problem, in which majority class with negative label and minority class with positive label. Class-specific cost regulation ELM is employed to weaken the influence of the imbalance and dispersion degree of the data on the classification accuracy. The structure of CCR-ELM decided by the number of hidden nodes, the input weights, the hidden biases and two tradeoff factors, plays the direct role in its classification performances. In general, above parameters are pre-set by human or randomly generated. Inappropriate values of the above parameters may lead to the useless or redundant neuron nodes, and make the whole structure complex. Based on this, brain storm optimization algorithm (BSO) is employed to find the optimal parameters' values of CCR-ELM. Because the number of input weights and biases depend on the number of hidden nodes, the networks with a varying number of hidden nodes are depicted by the individuals with various length. The novel mutation and merge operators generating the new individuals are proposed, with the purpose of effectively exchanging evolutionary information between the individuals with variable-length. Following that, an adaptive CCR-ELM based on variable-length brain storm optimization algorithm is presented to solve the class imbalance problems.

The rest of paper is organized as follows. A brief review of CCR-ELM is given in Sect. 2. BSO-based CCR-ELM for classifying imbalance data is proposed in Sect. 3. The classification performances of the proposed method are compared with other class imbalance learning algorithms in Sect. 4. Finally, the whole paper is concluded and work planned to do next are given.

2 Class-specific cost regulation extreme learning machine

In extreme learning machine originally proposed by Huang et al. (2005), the input weights and biases of SLFNs are randomly generated. Corresponding output matrix of hidden-layer is explicitly calculated in terms of the output weights by the fewer steps. Thus, the computation cost of ELM is less.

Suppose there are N distinct samples, depicted by $(X_i, y_i), i = 1, 2, \dots, N$. $X_i = [x_{i1}, x_{i2}, \dots, x_{in}]^T \in R^n$ and $y_i = [y_{i1}, y_{i2}, \dots, y_{im}]^T \in R^m$. Let a_j and β_j be the input and output weights, respectively. b_j is the bias of a hidden unit. A SLFN with L hidden nodes is modeled as follows.

$$\sum_{j=1}^L \beta_j g(a_j, b_j, X_i) = o_i, i = 1, \dots, N \quad (1)$$

In above formula, $g(\bullet)$ is the activation function and normally employs traditional nonlinear functions, such as

sigmoid, sine, radial basis functions, and so on (Huang et al. 2012). The error between the estimated output o_i and the actual output y_i shall be zero once SLFNs can exactly approximate the feature of data. That is,

$$\sum_{j=1}^L \beta_j g(a_j, b_j, X_i) = y_i, i = 1, \dots, N \tag{2}$$

Let $\beta = [\beta_1^T, \dots, \beta_L^T]^T$ and $Y = [y_1^T, \dots, y_N^T]^T$. Above model can be simplified as $H\beta = Y$.

$$H = \begin{bmatrix} g(a_1, b_1, X_1) & \dots & g(a_L, b_L, X_1) \\ \vdots & \dots & \vdots \\ g(a_1, b_1, X_N) & \dots & g(a_L, b_L, X_N) \end{bmatrix}_{N \times L} \tag{3}$$

H is so-called output matrix of hidden layer. h_{ij} represents the output of j th hidden node corresponding to the input X_i . During the training process, the parameters of a hidden node, including a_j and b_j , are not adjusted after initially generated. Corresponding output weights are estimated as follows.

$$\hat{\beta} = H^\dagger Y = \begin{cases} (\frac{1}{C} + H^T H)^{-1} H^T Y, L < N \\ H^T (\frac{1}{C} + H^T H)^{-1} Y, L \geq N \end{cases} \tag{4}$$

H^\dagger is the Moore–Penrose generalized inverse of H . C is a pre-set parameter, with the purpose of providing a tradeoff between minimizing the training error and maximizing the marginal distance. I is the unit matrix. The optimal output weights are gotten by minimizing the cost function $\|O - Y\|$.

After introducing class-specific regulation cost, CCR-ELM was presented to solve the class imbalance problems. Two tradeoff factors, including C^+ for minority positive samples and C^- for majority negative samples, are employed to rebalance both classes (Xiao et al. 2017). Assuming that the number of minority positive samples and majority negative samples are expressed by l_1 and l_2 , respectively. CCR-ELM is modelled as follows.

$$\min \left(\frac{1}{2} \|\beta\|^2 + \frac{1}{2} C^+ \sum_{i=1}^{l_1} \xi_i^2 + \frac{1}{2} C^- \sum_{i=1}^{l_2} \xi_i^2 \right) \tag{5}$$

s.t. $h(x_i)\beta = y_i - \xi_i, i = 1, \dots, N$.

Corresponding output weights $\hat{\beta}$ are computed as:

$$\hat{\beta} = H^\dagger Y = \begin{cases} (\frac{1}{C^+} + \frac{1}{C^-} + H^T H)^{-1} H^T Y, L < N \\ H^T (\frac{1}{C^+} + \frac{1}{C^-} + H^T H)^{-1} Y, L \geq N \end{cases} \tag{6}$$

For the binary classification problems, the decision function of CCR-ELM-based classifier is $f(x) = \text{sign}h(x)\beta$.

$$f(x) = \begin{cases} \text{sign} h(x) (\frac{1}{C^+} + \frac{1}{C^-} + H^T H)^{-1} H^T Y, L < N \\ \text{sign} h(x) H^T (\frac{1}{C^+} + \frac{1}{C^-} + H^T H)^{-1} Y, L \geq N \end{cases} \tag{7}$$

In CCR-ELM, there are five key parameters having a direct impact on the classification accuracy, including the number of hidden nodes L , the input weights a_j , the biases b_j , C^+ for the minority positive samples and C^- for the majority negative samples. The former three parameters decide the structure of SLFNs and normally are pre-set by human.

3 Adaptive CCR-ELM with variable-length brain storm optimization algorithm

Brain storm optimization algorithm (BSO) proposed by Cheng et al. (2015) is a promising swarm intelligence algorithm simulating the human brainstorming process. Implicit parallel search ability of BSO makes the population having superior diversity and converges to the optimum faster, especially for multi-modal or large-scale optimization problems. Many improved BSO algorithms have been presented, with the purpose of avoiding premature convergence and accelerating the convergence speed. Cheng et al. (2014) proposed the population diversity enhanced brain storm optimization algorithm (en-BSO). Two partial re-initializing solutions strategies were given to help the population jump out of local optima. The algorithm steps of en-BSO with re-initialization strategy is shown in Algorithm 1 (Cheng et al. 2014).

In en-BSO, a new individual is formed by the following mutation or mergence operators. Let $\bar{x}_{ij}(t)$ and $x_{ij}(t)$ be j th gene of x_i after and before mutation operation. T and t denote the maximum iterations and the current generation. logsig is a logarithmic sigmoid transfer function. c is a constant. The mutation operation corresponds to Step 3.1 shown as follows.

$$\bar{x}_{ij}(t) = x_{ij}(t) + \xi(t) \times \text{rand}() \tag{8}$$

$$\xi(t) = \text{log sig} \left(\frac{0.5T - t}{c} \right) \times \text{rand}() \tag{9}$$

The mergence operation corresponding to Step 3.2 is done based on two selected parent individuals or cluster centers expressed by $x_{ij}(t)$ and $x_{kj}(t)$. $\bar{\bar{x}}_{ij}(t)$ denotes the new individual. Let $\alpha \sim U(0, 1)$ be a random number.

$$\bar{\bar{x}}_{ij}(t) = \alpha x_{ij}(t) + (1 - \alpha) x_{kj}(t) \tag{10}$$

Algorithm 1 The en-BSO

-
- 1: **Initialization:**
Randomly generate n individuals, and evaluate them.
 - 2: **Clustering:**
 - 2.1: Partition n individuals into m clusters by k-means clustering algorithm;
 - 2.2: Rank individuals in each cluster and record the best individual as cluster center in each cluster;
 - 2.3: A new individual is generated based on a randomly selected cluster center as $r_c < p_c \cdot r_c \in [0, 1]$ and p_c is a pre-determined probability.
 - 3: **Generating new individual:**
One or two cluster(s) are randomly selected to generate new individual in terms of a random value $r_g \in [0, 1]$
 - 3.1: A new individual is generated by the mutation operation based on a randomly selected cluster center or an individual in this cluster as a pre-determined probability $p_g > r_g$;
 - 3.2: A new individual is formed by merging two randomly selected cluster centers or individuals from corresponding clusters when $p_g < r_g$.
 - 4: **Selection:**
Compared the newly generated individual with the original individual and the one with better fitness will be retained.
 - 5: **Re-initialization:**
Re-initialize half solutions which is selected randomly after certain iterations.
 - 6: Evaluate all of individuals.
 - 7: If the terminate condition is not satisfied, jump to Step 2. Otherwise, output the optimal individual.
-

Above en-BSO is employed to find the optimal parameters of CCR-ELM. Among five key parameters of CCR-ELM, the number of input weights $a_{ij} \in R^{N \times L}$ and the biases $b_j \in R^L$ vary with the number of hidden nodes L . Based on this, the networks with different L can be described by the individuals with various length, expressed by $p_i = (C^+, C^-, a_{11}, \dots, a_{1N}, \dots, a_{L1}, \dots, a_{LN}, b_1, b_2, \dots, b_L)$. The length of each individual is $L(N + 1) + 2$. During the evolution, once two parent individuals have different length, no matching genes for the shorter individual can be merged with the longer one by Eq. (10). Therefore, a novel merge operator is proposed, and then variable-length en-BSO algorithm (VLen-BSO) is constructed.

Assuming that l_j and l_k are the length of parent individuals. Only the genes in the matching positions can be merged each other and the new genes in the mismatching parts are randomly generated based on corresponding parents.

$$\bar{x}_{ij}(t) = \begin{cases} \alpha x_{ij}(t) + (1 - \alpha)x_{kj}(t) & j \leq \min(l_i, l_k) \\ x_{ij}(t) + \alpha \xi(t) & \min(l_i, l_k) < j \leq \lceil \max(l_i, l_k) \times \text{rand}() \rceil, l_i > l_k \\ 0 & j > \lceil \max(l_i, l_k) \times \text{rand}() \rceil \end{cases} \quad (11)$$

Above variable-length en-BSO algorithm is introduced to find the optimal structure and tradeoff factors of CCR-

ELM for the class imbalance learning. We call it adaptive CCR-ELM with VLen-BSO. In order to evaluate the classification accuracy of each individual, the fitness function is defined as the root mean square error (RMSE) on the training set X_{train} .

$$F = \sqrt{\frac{\sum_{i=1}^N \left\| \sum_{j=1}^L \hat{\beta}_j g(a_j, b_j, X_i) - y_i \right\|_2^2}{N}} \quad (12)$$

The detailed algorithm steps of adaptive CCR-ELM with VLen-BSO are shown as follows.

4 Experimental results and discussion

Nine binary-class imbalanced datasets are chosen from UCI machine learning repository preprocessed by Frank and Asuncion (2010) in the experiments, with the purpose of fully analyzing and comparing the classification performances of the proposed method. Especially, the data in Wine1 record chemical composition of wine in the same area of Italy, and can be classified into three categories in fact. It is transformed into binary-classification problem by combining the data in any two classes into majority samples, and employing the others as minority samples. Detailed information about these datasets is listed in

Algorithm 2 Adaptive CCR-ELM with VLen-BSO

-
- 1: The imbalanced dataset $X = \{(x_i, y_i), i = 1, 2, \dots, N\}$ is divided into the training set X_{train} and test set X_{test} .
 - 2: The initial population P is randomly generated based on the defined lower and upper bounds of variables.
 - 3: Finding the optimal parameters of CCR-ELM by VLen-BSO and output these optimal parameters.
 - 4: Classify the imbalanced dataset by CCR-ELM with the optimal parameters.
-

Table 1 Detailed information about the datasets from UCI

Dataset	Number of attributes	Number of classes	Number of training data	Number of test data	Imbalance ratio
Wine1	13	2	100	78	0.3333
Wine2	12	2	2898	2000	0.2461
Adult	14	2	28,842	20,000	0.2500
WPBC	34	2	100	98	0.2533
WDBC	32	2	300	317	0.4000
Dota2	116	2	50,000	52,944	0.1000
Iris	4	2	75	75	0.3333
Car	6	2	1000	728	0.2997
Poker hand	11	2	25,010	100,000	0.2382

Table 1. The imbalanced ratio (IR) of datasets varies from 0.1 to 0.4. In addition, a real-world class imbalance problem for the fault diagnosis of conveyor belt is employed to further analyze the classification performances.

To evaluate the class imbalance algorithms, we give more insight into not only the overall accuracy of the whole samples, but also the classification accuracy obtained within each class. Suppose TP , TN , FP , FN are the number of data with true positive, true negative, false positive and false negative, respectively. The overall accuracy of the dataset is defined in Eq. (13). However, this index is sensitive to the class distribution and cannot fully reflect the imbalance classification performances (Xiao et al. 2017). Based on this, minority accuracy, majority accuracy and G-mean are employed to evaluate the binary-classification performances (Yu et al. 2016).

$$\text{overall accuracy} = \frac{TP + FN}{TP + FP + TN + FN} \quad (13)$$

$$\text{minority accuracy} = \frac{TP}{TP + FN} \quad (14)$$

$$\text{majority accuracy} = \frac{TN}{TN + FP} \quad (15)$$

$$G - \text{mean} = \sqrt{\frac{TP}{TP + FN} \times \frac{TN}{TN + FP}} \quad (16)$$

4.1 The influence of parameters on the performance of CCR-ELM

In CCR-ELM, sigmoid expressed by $g(a, b, X) = 1/(1 + \exp(a * X + b))$ is employed as the activation function and the bound of input weights and biases are set as $a_j, b_j \in [-1, 1]$. C^+ and C^- are the discrete values chosen from $\{2^{-24}, 2^{-23}, \dots, 2^{25}\}$ and the possible values of L are $\{10, 20, \dots, 1000\}$.

The statistical classification performances under different L are shown in Fig. 1. Taking Car dataset as example, the effect of L on G-mean is shown in Fig. 2. With the

increasing of L , more hidden nodes are contained in the network, bringing the better testing accuracy, and relatively stable G-mean when $L \geq 500$ for most of the datasets. However, G-mean for large-scale datasets, such as Poker hand and Dota2, becomes larger with the increasing number of hidden nodes, causing the weak generalization of CCR-ELM. Iris, as a small-scale dataset, the classification performance is apparently independent on L because larger L causes more redundant nodes having meaningless impact on the algorithm performances. In summary, the classifier with the complex structure is in favor of the classification accuracy, but obviously increases the computational complexity, causing the time-consuming for learning.

As L is pre-set to 400, G-mean under different C^+ and C^- are shown in Fig. 3. Obviously, G-mean is sensitive to both C^+ and C^- . When C^+ and C^- are less than 2^{-10} , the classification accuracy retains worse because of less punishment for the misclassified data.

4.2 Comparison of the classification performances among ACCR-ELM combining with different evolutionary optimization algorithms

In this paper, five kinds of swarm intelligence optimization algorithms, including GA, PSO (Guo et al. 2016), variable-length PSO (VPSO) (Wang et al. 2018), VLen-BSO and group-based VLen-BSO (GVLen-BSO) are introduced to find the optimal parameters of CCR-ELM, with the purpose of improving the generalization and classification accuracy. Group-based VLen-BSO employes a simple grouping method (Guo et al. 2015) in the cluster operator instead of the k-means method to reduce the computational cost. During the experiments, the maximum terminal iteration and the population size are respectively set to 1500 and 20 for all methods. For ACCR-ELM with GA, the crossover probability is 0.9, the mutation probability is 0.01, and the generation gap is 0.9. For ACCR-ELM with PSO and

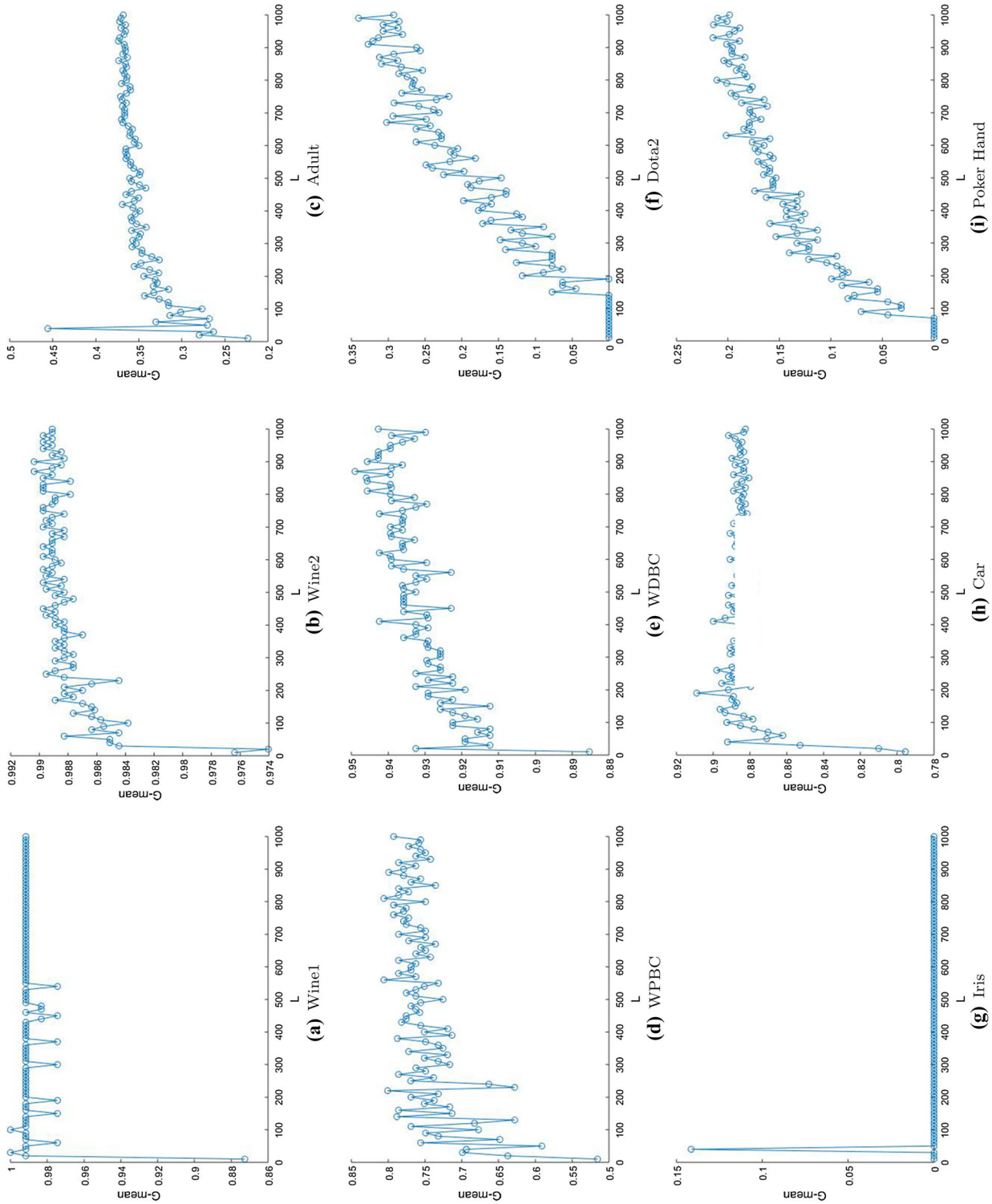


Fig. 1 The G-mean under different L

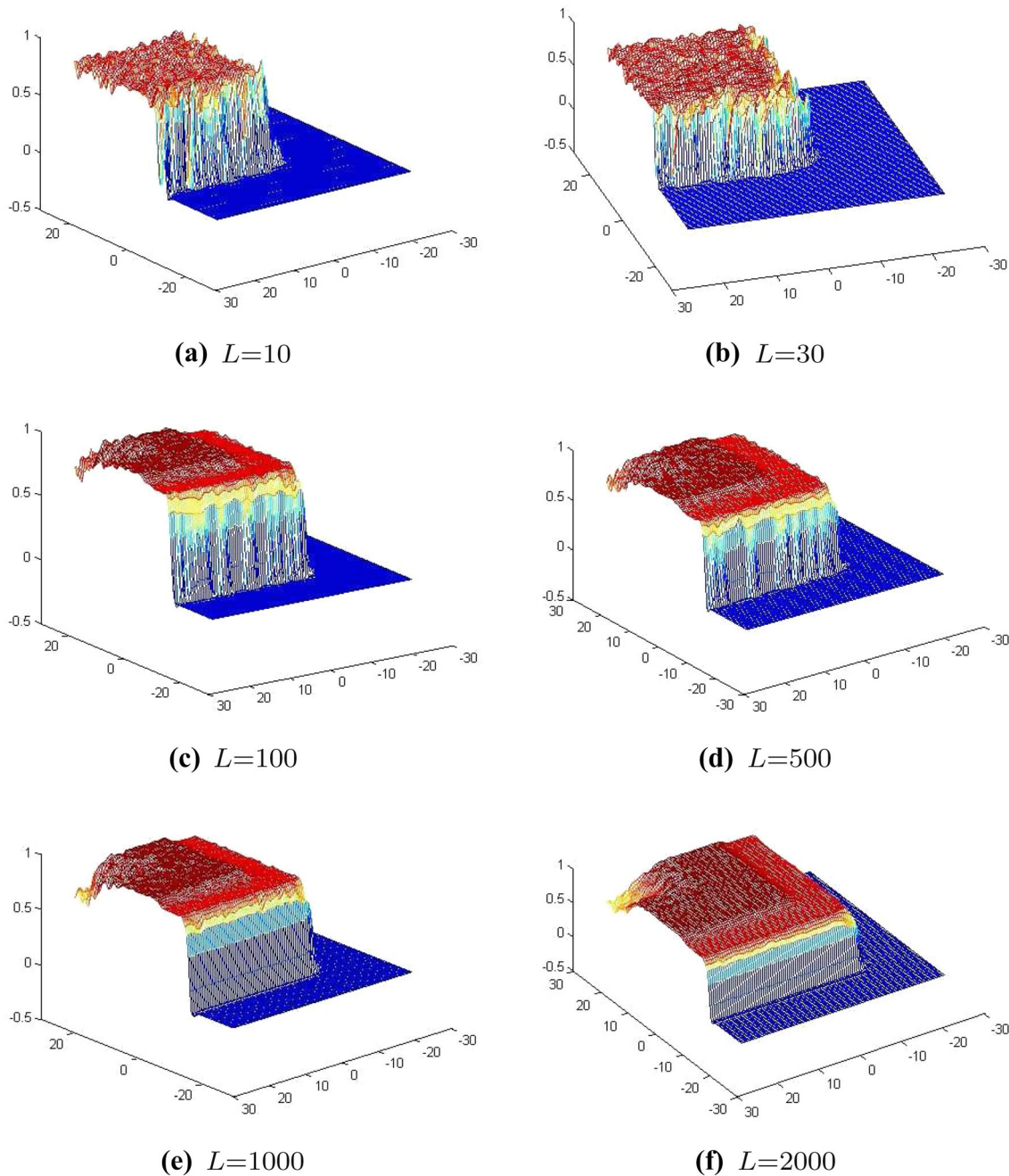


Fig. 2 G-mean of ACCR-ELM with different number of hidden nodes L

VPSO, the learning factors $c_1 = 1.5$, $c_2 = 1.7$ and the inertia weight $\omega = 1$. For ACCR-ELM with VLen-BSO, $p_c = 0.2$, $p_g = 0.6$. The number of the clusters is set to 20. Only half of the solutions will be re-initialized every 200 iterations.

The classification performances of above all adaptive CCR-ELM are compared with the original CCR-ELM and the statistical results under 30 running times are listed in Tables 2 and 3.

The statistical results of overall accuracy for 30 running times shown in Table 2 indicate that ACCR-ELMs with VLen-BSO and GVLen-BSO have the better classification accuracy than other algorithms. By further comparing G-mean shown in Table 3, we see that traditional CCR-ELM with randomly generated parameters has the worse classification accuracy, especially for Iris, Dota2 and Poker hand datasets. In contrast, adaptive CCR-ELMs with the optimal parameters have the better classification

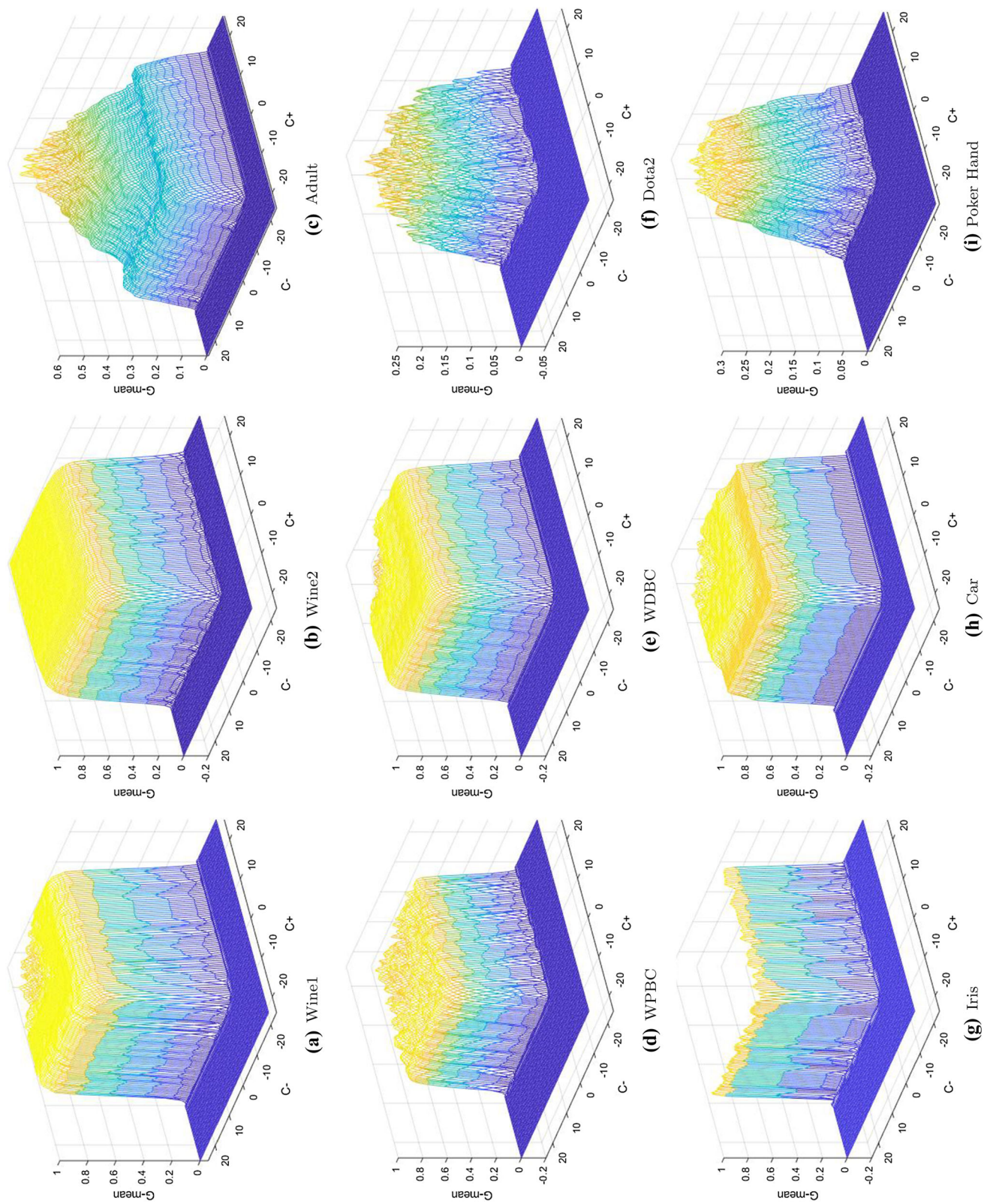


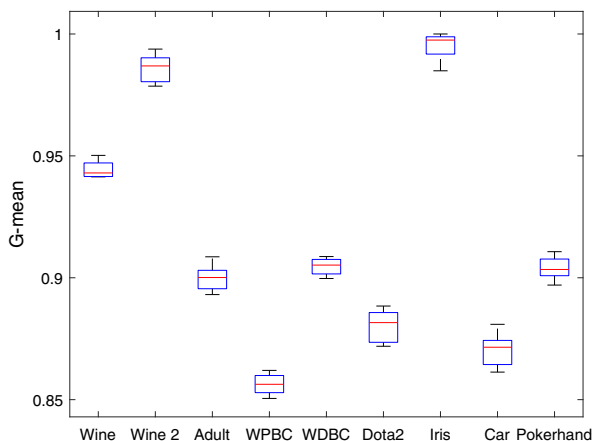
Fig. 3 The G-mean under different L

Table 2 Compare the overall accuracy ($(C^+, C^-, L)Max/Average(\%)$) among CCR-ELM and ACCR-ELM with GA, PSO, VPSO, VLen-BSO and GVLen-BSO

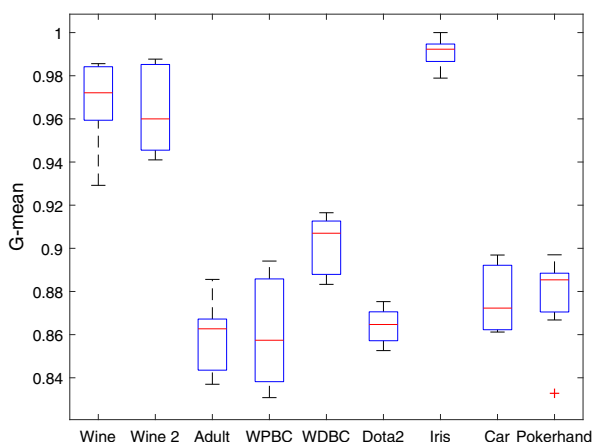
Dataset	CCR-ELM	ACCR-ELM with GA	ACCR-ELM with PSO	ACCR-ELM with VPSO	ACCR-ELM with VLen-BSO	ACCR-ELM with GVLen-BSO
Wine	$(2^7, 2^6, 70)$	$(6.59^7, 3.44^6, 70)$	$(3.76^6, 9.78^6, 50)$	$(1.82^5, 1.72^5, 70)$	$(3.25^6, 3.40^6, 80)$	$(1.68^7, 2.08^7, 30)$
	96.87/95.41	92.55/91.17	97.75/97.23	98.75/97.10	98.88/97.75	100/97.42
Wine2	$(2^3, 2^{-10}, 10)$	$(5.37^3, 1.12^3, 250)$	$(1.74^3, 1.81^4, 170)$	$(5.75^6, 3.70^6, 60)$	$(5.95^6, 2.27^7, 120)$	$(2.38^7, 6.55^6, 80)$
	97.30/96.55	98.91/97.96	99.06/98.95	99.24/99.03	99.31/99.28	99.37/99.14
Adult	$(2^7, 2^{-1}, 600)$	$(1.43^7, 5.11^6, 310)$	$(6.34^7, 5.12^6, 130)$	$(2.81^4, 1.68^4, 30)$	$(2.45^7, 5.76^7, 80)$	$(1.94^7, 2.97^7, 500)$
	76.53/75.71	81.72/79.77	80.48/78.98	86.14/84.85	88.64/87.95	79.51/77.89
WPBC	$(2^{-12}, 2^4, 20)$	$(3.38^6, 2.89^7, 90)$	$(3.32^7, 1.77^7, 90)$	$(6.77^3, 5.32^3, 150)$	$(4.79^6, 9.19^6, 10)$	$(5.92^6, 1.02^7, 60)$
	82.47/80.82	85.30/83.99	72.16/70.61	92.88/91.43	90.38/89.42	92.31/89.62
WDBC	$(2^7, 2^{-21}, 60)$	$(4.46^7, 7.43^6, 80)$	$(5.38^6, 1.22^7, 220)$	$(8.39^2, 1.54^3, 40)$	$(1.76^7, 8.79^6, 110)$	$(1.85^7, 2.52^7, 90)$
	90.82/90.13	90.12/89.01	91.50/90.37	92.41/91.55	91.82/90.77	92.45/91.22
Dota2	$(2^{13}, 2^6, 90)$	$(2.32^5, 6.65^6, 340)$	$(4.42^5, 1.33^6, 60)$	$(2.81^4, 1.68^4, 90)$	$(3.45^6, 4.78^6, 90)$	$(5.66^6, 1.39^6, 150)$
	85.69/85.14	84.42/81.11	83.10/81.32	85.42/85.04	87.75/86.94	85.76/85.66
Iris	$(2^0, 2^1, 30)$	$(1.22^6, 8.09^6, 120)$	$(4.99^5, 2.54^7, 280)$	$(2.53^7, 4.92^6, 30)$	$(2.06^7, 2.38^7, 50)$	$(2.31^7, 1.45^6, 70)$
	41.33/37.86	60.33/59.63	35.52/34.66	100/99.54	100/100	100/98.35
Car	$(2^{-8}, 2^6, 60)$	$(4.96^7, 3.78^7, 60)$	$(3.09^7, 2.30^7, 420)$	$(9.62^4, 1.20^6, 60)$	$(3.24^7, 1.57^7, 30)$	$(1.62^7, 2.47^7, 80)$
	84.14/82.81	84.64/83.77	84.74/83.73	85.69/81.94	86.53/84.66	85.01/84.36
PokerHand	$(2^{11}, 2^{-17}, 70)$	$(5.41^4, 3.86^4, 130)$	$(2.42^6, 2.33^7, 120)$	$(8.42^7, 5.50^6, 130)$	$(4.32^6, 6.66^6, 20)$	$(1.01^7, 2.85^6, 100)$
	78.42/74.98	85.33/83.46	86.53/83.23	89.87/88.98	90.03/89.83	71.45/71.42

Table 3 Compare G-mean ($(C^+, C^-, L)Max/Average(\%)$) among CCR-ELM and ACCR-ELM with GA, PSO, VPSO, VLen-BSO and GVLen-BSO

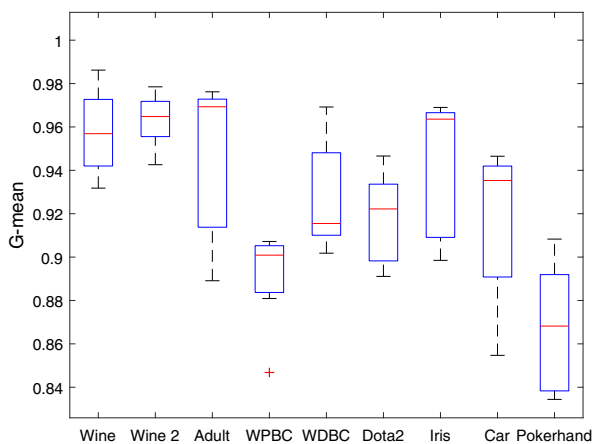
Dataset	CCR-ELM	ACCR-ELM with GA	ACCR-ELM with PSO	ACCR-ELM with VPSO	ACCR-ELM with VLen-BSO	ACCR-ELM with GVLen-BSO
Wine	$(2^{-4}, 2^{-3}, 20)$	$(6.77^6, 1.46^6, 160)$	$(1.97^7, 2.38^7, 120)$	$(6.13^4, 3.60^4, 60)$	$(3.54^6, 2.23^7, 70)$	$(1.64^7, 1.98^7, 60)$
	0.9616/0.9540	0.9263/0.9143	0.9832/0.9788	0.9858/0.9794	0.9916/0.9741	0.9916/0.9726
Wine2	$(2^{-2}, 2 - 23, 20)$	$(3.42^5, 9.47^7, 120)$	$(2.73^7, 1.51^7, 10)$	$(9.49^5, 7.71^4, 130)$	$(2.57^7, 3.13^6, 190)$	$(2.79^7, 2.63^6, 100)$
	0.9612/0.9526	0.9804/0.9724	0.9819/0.9800	0.9891/0.9877	0.9874/0.9869	0.9725/0.9648
Adult	$(2^{18}, 2^{-12}, 370)$	$(6.31^4, 6.78^5, 30)$	$(4.20^6, 1.55^6, 40)$	$(6.93^4, 6.23^6, 70)$	$(4.33^6, 8.37^6, 30)$	$(3.30^7, 1.29^7, 150)$
	0.9616/0.9540	0.8407/0.8367	0.8223/0.8108	0.8858/0.8647	0.8858/0.8647	0.8913/0.8891
WPBC	$(2^7, 2^{-1}, 30)$	$(3.43^6, 3.97^6, 140)$	$(6.69^6, 3.15^7, 100)$	$(9.90^6, 1.21^6, 60)$	$(1.06^7, 1.69^7, 80)$	$(2.56^7, 1.47^6, 110)$
	0.6896/0.6062	0.8426/0.8352	0.7009/0.6364	0.8905/0.8766	0.8789/0.8603	0.8946/0.8809
WDBC	$(2^4, 2^{-13}, 10)$	$(5.23^6, 6.21^6, 60)$	$(1.54^6, 8.07^6, 30)$	$(1.40^7, 2.32^7, 160)$	$(1.55^7, 1.69^7, 10)$	$(1.94^7, 2.97^7, 20)$
	0.9616/0.9540	0.9057/0.8982	0.9123/0.9037	0.9185/0.9127	0.9162/0.9052	0.9229/0.9100
Dota2	$(2^4, 2^{21}, 30)$	$(4.76^5, 3.53^6, 90)$	$(3.67^5, 7.43^4, 50)$	$(6.45^5, 2.41^6, 30)$	$(3.66^7, 4.11^6, 140)$	$(1.19^7, 2.05^5, 70)$
	0.0036/0.020	0.8508/0.8437	0.8406/0.8312	0.8779/0.8696	0.8895/0.8816	0.8996/0.8911
Iris	$(2^{-8}, 2^{-16}, 60)$	$(4.23^4, 2.61^6, 120)$	$(2.94^7, 2.25^7, 150)$	$(1.97^7, 1.32^7, 10)$	$(2.58^7, 1.77^7, 30)$	$(2.87^7, 2.84^4, 40)$
	0.2464/0.1963	0.6421/0.6372	0.2132/0.1414	1/0.9926	1/1	0.9738/0.9653
Car	$(2^{-1}, 2^5, 530)$	$(8.44^7, 8.00^7, 80)$	$(2.25^7, 2.80^7, 100)$	$(5.11^4, 2.21^4, 90)$	$(2.65^7, 1.00^7, 90)$	$(1.26^7, 2.65^7, 150)$
	0.8802/0.8419	0.8522/0.8436	0.8640/0.8486	0.8927/0.8738	0.8720/0.8710	0.8702/0.8541
PokerHand	$(2^4, 2^{17}, 80)$	$(6.67^5, 3.98^4, 210)$	$(1.10^6, 7.84^6, 130)$	$(5.53^5, 2.65^4, 220)$	$(4.55^7, 4.73^6, 40)$	$(1.61^7, 2.71^7, 100)$
	0.4266/0.3676	0.8212/0.8011	0.7963/0.7748	0.8958/0.8821	0.9020/0.9013	0.8589/0.8477



(a) G-mean for adaptive CCR-ELM with VLen-BSO



(b) G-mean for adaptive CCR-ELM with VPSO



(c) G-mean for adaptive CCR-ELM with GVLen-BSO

Fig. 4 The classification robustness of adaptive CCR-ELMs under varied imbalanced ratios

performances for most of datasets. Especially, VLen-BSO

and GVLen-BSO provides the more rational parameters for CCR-ELM, classifying the minority data more accurately.

4.3 The effect of the imbalance ratio on the algorithm performance

In real-world classification problems, the imbalance ratios reflecting the quantitative difference between the data of majority class and minority class, depend on the problem properties. The experiments under the imbalanced ratios changing from 0.1 to 0.4 are done for above nine imbalance datasets, with the purpose of analyzing the robustness of the class imbalance learning algorithms under varied imbalanced ratios.

The statistic experimental results of G-mean under different imbalanced ratio are shown in Fig. 4. Comparing the classification performances among ACCR-ELM with VLen-BSO, VPSO and GVLen-BSO indicate that no matter the imbalance ratios change or not, ACCR-ELM with VLen-BSO has the relatively stable classification performance. The G-means of ACCR-ELMs with GVLen-BSO and VPSO both fluctuate in the range about 0.5, meaning relatively worse robustness, but GVLen-BSO obtains the more rational parameters with the relatively better classification accuracy than VPSO.

4.4 Application of ACCR-ELM algorithm in fault diagnosis of conveyor belt

Conveyor belt is the most important transportation equipment in coal mine. Any fault, such as overcurrent, off-tracking, overload, and so on, may make the belt shut down immediately, and cause production delay, even reduce production efficiency. Given that the faults only occur occasionally and the belt generally holds a normal operation, the monitoring data for a conveyor belt is in fact imbalance.

The experiment is done on the monitoring data collected from two belts in certain coal mine. Belt 1 represents a belt for the underground transportation of coal mine. Belt 2 is a conveyor belt in a coal separating plant. Detailed information about two belts are listed in Tables 4 and 5. In ACCR-ELM with VLen-BSO, the number of hidden nodes is set to 500, sigmoid activation function with $C = 1$ is employed. The learning factors and inertia weight of VPSO are denoted as $c_1 = 1.5$, $c_2 = 1.6$, and $\omega = 1$. The probability of VLen-BSO is preset to $p_c = 0.2$, $p_g = 0.6$. The statistic experimental results with 30 running times shown in Table 6 indicate that ACCR-ELM with VLen-BSO has

Table 4 The information of two belts

Belt ID	Length (m)	Running speed (s/m)	Capacity (kW)	The type of revolution speed transducer
Belt1	1512	3.45	1650	SZMB-9
Belt2	980	4.8	840	SZMB-5

Table 5 Detailed information on the fault of belt in coal mine

Belt ID	Number of attributes	Number of classes	Number of training data	Number of test data	Imbalance ratio	Number of attributes
Belt1	40	2	7867	529	8396	0.0630
Belt2	40	2	9772	428	10,200	0.0420

Table 6 Comparison of the classification performance for the belt fault between ACCR-ELM with VLen-BSO and VPSO

Belt ID	ACCR-ELM with VLen-BSO	ACCR-ELM with VPSO
Belt1		
(C^+, C^-, L)	(1.95 ⁵ , 2.01 ⁵ , 1490)	(1,1500)
Max/average	0.8973/0.8724	0.8104/0.7441
SD	3.445e−4	2.147e−3
Belt2		
(C^+, C^-, L)	(2.84 ⁴ , 5.11 ³ , 1940)	(1,1500)
Max/average	0.9100/0.8841	0.8422/0.8003
SD	5.796e−4	8.776e−4

the better classification accuracy and stability that VPSO because of the more reasonable structure of ELM.

5 Conclusions

An adaptive CCR-ELM combining with VLen-BSO was proposed to deal with the class imbalance problems, with the purpose of improving the generalization and classification accuracy. VLen-BSO is employed to find the optimal parameters of CCR-ELM, including the number of hidden nodes, the input weights, the biases and the tradeoff factors. All above parameters compose of an individual in VLen-BSO and the length of individuals vary with the number of hidden nodes. In order to incorporate two parent individuals with different length, a novel mergence operator is presented to generate a new individual. The experimental results for nine imbalance datasets show that VLen-BSO can find better parameters of CCR-ELM, resulting in the better classification accuracy than other evolutionary optimization algorithms, such as GA, PSO, and VPSO. In addition, the classification performance of the proposed adaptive algorithm is relatively stable under varied imbalance ratios. Applying the proposed algorithm in the fault diagnosis of conveyor belt also proves that ACCR-ELM with VLen-BSO has the better classification

accuracy and stability. The optimal structure for ensemble imbalance learning with more than one CCR-ELM is our future work.

Acknowledgements This work is supported by National Natural Science Foundation of China under Grant 61573361, Research Program of Frontier Discipline of China University of Mining and Technology under Grant 2015XKQY19, Six Talent Peak Project in Jiangsu Province under Grant 2017-DZXX-046 and National Key Research and Development Program under Grant 2016YFC0801406.

References

- Cheng S, Qin QD, Chen JF et al (2014) Maintaining population diversity in brain storm optimization algorithm. *Proc IEEE Congr Evolut Comput* 4(3):3230–3237
- Cheng S, Qin QD, Chen JF et al (2015) Brain storm optimization algorithm: a review. *Artif Intell Rev* 6728:1–14
- Ertam F, Avci E (2017) A new approach for internet traffic classification: GA-WK-ELM. *Measurement* 95:135–142
- Frank A, Asuncion A (2010) UCI machine learning repository. University of California, Irvine, School of Information and Computer Sciences. <http://archive.ics.uci.edu/mlS>
- Gao M, Hong X, Chen S et al (2011) A combined smote and PSO based RBF classifier for two-class imbalanced problems. *Neurocomputing* 74(17):3456–3466
- Guo XP, Wu YL, Xie LX et al (2015) An adaptive brain storm optimization algorithm for multiobjective optimization problems. *Lect Notes Comput Sci* 9140(4):365–372

- Guo YN, Zhen Y, Wang C et al (2016) Cultural particle swarm optimization algorithms for uncertain multi-objective problems with interval parameters. *Nat Comput* 16(4):1–22
- Guo YN, Zhang P, Cheng J et al (2017) An improved weighted ELM with krill herd algorithm for imbalanced learning. In: *Advances in swarm intelligence* Springer, pp 371–378
- Guo YN, Cheng J, Luo S, Gong DW, Xue Y (2018) Robust dynamic multi-objective vehicle routing optimization method. *IEEE/ACM Trans Comput Biol Bioinform* 15(6):1891–1903
- Han F, Yao HF, Ling QH (2013) An improved evolutionary extreme learning machine based on particle swarm optimization. *Neurocomputing* 116:87–93
- He HB, Ma YQ (2013) *Imbalanced learning: foundations, algorithms, and applications*. Wiley, New York
- Huang GB, Zhu QY, and Siew CK (2005) Extreme learning machine: a new learning scheme of feed forward neural networks. In: *IEEE international joint conference on neural networks*, vol 2, pp 985–990
- Huang GB, Zhou HM, Ding XJ et al (2012) Extreme learning machine for regression and multiclass classification. *IEEE Trans Syst Man Cybern Part B* 42(2):513–529
- Huang GB, Song SJ, Gupta JND et al (2014) Semi-supervised and unsupervised extreme learning machines. *IEEE Trans Cybern* 44(12):2405–2417
- Li QJ, Mao YB (2014) A review of boosting methods for imbalanced data classification. *Pattern Anal Appl* 17(4):679–693
- Li D, Zhang W (2014) Stock price prediction based on ELM and FOA. *Comput Eng Appl* 50(18):14–18
- Liu XY, Wu JX, Zhou ZH (2009) Exploratory undersampling for class-imbalance learning. *IEEE Trans Syst Man Cybern Part B* 39(2):539–550
- Mirza B, Lin ZP, Toh KA (2013) Weighted online sequential extreme learning machine for class imbalance learning. *Neural Process Lett* 38(3):465–486
- Mirza B, Lin ZP, Liu N (2015) Ensemble of subset online sequential extreme learning machine for class imbalance and concept drift. *Neurocomputing* 149(Part A):316–329
- Nitesh VC, Bowyer KW, Lawrence OH et al (2002) SMOTE: synthetic minority over-sampling technique. *J Artif Intell Res* 16(1):321–357
- Ramentol E (2012) SMOTE-RS: a hybrid preprocessing approach based on oversampling and undersampling for high imbalanced data-sets using smote and rough sets theory. *Knowl Inf Syst* 33(2):245–265
- Rong HJ, Huang GB, Narasimhan S et al (2009) Online sequential fuzzy extreme learning machine for function approximation and classification problems. *IEEE Trans Syst Man Cybern Part B* 39(4):1067
- Shen F, Liu J, Wu P (2018) Double complete D-LBP with extreme learning machine auto-encoder and cascade forest for facial expression analysis. In: *25th IEEE international conference on image processing (ICIP)*, pp 1947–1951
- Wang B, Sun YN, Xue B et al (2018) Evolving deep convolutional neural networks by variable-length particle swarm optimization for image classification. In: *IEEE congress on evolutionary computation*
- Wu G, Chang EY (2005) KBA: kernel boundary alignment considering imbalanced data distribution. *IEEE Trans Knowl Data Eng* 17(6):786–795
- Wu DH, Wang ZL, Ye C et al (2016) Mixed-kernel based weighted extreme learning machine for inertial sensor based human activity recognition with imbalanced dataset. *Neurocomputing* 190:35–49
- Xiao WD, Zhang J, Li Y et al (2017) Class-specific cost regulation extreme learning machine for imbalanced classification. *Neurocomputing* 261:70–82
- Yang LX, Yang SY, Li SJ et al (2017) Incremental laplacian regularization extreme learning machine for online learning. *Appl Soft Comput* 59:546–555
- Yu HL, Sun CY, Yang XB et al (2016) ODOC-ELM: optimal decision outputs compensation-based extreme learning machine for classifying imbalanced data. *Knowl Based Syst* 92:55–70
- Zeng ZQ, Qun WU, Liao BS et al (2009) A classification method for imbalance data set based on kernel smote. *Acta Electron Sin* 37(11):2489–2495
- Zhao JW, Wang ZH, Dong SP (2012) Online sequential extreme learning machine with forgetting mechanism. *Neurocomputing* 87(15):79–89
- Zong WW, Huang GB, Chen YQ (2013) Weighted extreme learning machine for imbalance learning. *Neurocomputing* 101(3):229–242

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.