Machine Vision and Applications
DOI 10.1007/s00138-017-0828-4

SPECIAL ISSUE PAPER

Decay-weighted extreme learning machine for balance and optimization learning

Qing Shen[1] · Xiaojuan Ban[1] · Ruoyi Liu[1] · Yu Wang[1,2]

Received: 20 September 2016 / Revised: 26 January 2017 / Accepted: 19 February 2017

Abstract The original extreme learning machine (ELM) was designed for the balanced data, and it balanced misclassification cost of every sample to get the solution. Weighted extreme learning machine assumed that the balance can be achieved through the equality of misclassification costs. This paper improves previous weighted ELM with decay-weight matrix setting for balance and optimization learning. The decay-weight matrix is based on the sample number of each class, but the weight sum values of each class are not necessarily equal. When the number of samples is reduced, the weight sum is also reduced. By adjusting the decaying velocity, classifier could achieve more appropriate boundary position. From the experimental results, the decay-weighted ELM obtains the better effects in solving the imbalance classification tasks, particularly in multiclass tasks. This method was successfully applied to build the prediction model in the urban traffic congestion prediction system.

Keywords Extreme learning machine · Weighted extreme learning machine · Multiclass classification

✉ Qing Shen
shenqingcc222333@gmail.com

✉ Xiaojuan Ban
banxj@ustb.edu.cn

Ruoyi Liu
lryliuruoyi@163.com

Yu Wang
hejohejo@126.com

[1] University of Science and Technology Beijing, Beijing 100083, China

[2] North Electronic Instrument Institute, Beijing 100191, China

1 Introduction

Extreme learning machine (ELM) essentially provides a unified solution for SLFNs, which include but not limit to support vector network, traditional neural network and regularized network [1–6]. Different from most the existing learning algorithms, particularly backpropagation (BP) algorithm and support vector machine (SVM), ELM randomly assigns the input weights and hidden layer biases and then solves the minimization problem of training error and norm of output weights, which simplifies the training process to a least-squares problem [7]. Therefore, ELM displayed its unique advantages such as high training efficiency, easy implementation and unification of multi-classification and regression. In recent years, a lot of new research has emerged such as Semi-Supervised ELM [8,9], Incremental ELM [4,10], R-ELM [11]. These researches significantly extend the capacity of ELM to deal with various problem.

However, few studies have addressed the problem of imbalanced data distribution. In practical applications, most of the objects are classified as imbalanced data sets. In many cases, the information contained minority class data is often more important, such as network attack data [12] and health data [13]. Classification algorithms always assume that the data are balanced. The original ELM was designed for the balanced data, and it balanced misclassification cost of every samples to get the global optimal solution. By assuming balanced class distribution or equal misclassification cost, ELM prefers to divide the sample into the majority class. In practical applications, this defect will lead to some important messages that cannot be found timely.

Typically, there are two approaches to deal with imbalanced data [14,15]. One establish balanced data distribution through various sampling methods or algorithmic methods. There are mainly two sampling techniques: under-sampling

Published online: 01 March 2017

approach [16] which randomly drives representative sample from majority samples, and the over-sampling approach [17] which duplicates samples or creates new samples into the minority class. Because the assumption that the neighborhood of a minority sample shares the same label is not always satisfied with different types of data, under-sampling approach may lead to loss of majority class information, and over-sampling approach may cause distortion of minority class.

The other approach to deal with imbalanced data is to set different misclassification cost for each particular class or sample. A simple and straightforward way is to balance the cost by assigning weights according to the data distribution. Weighted Regularized ELM was proposed in the work of Deng et al. [18]. By minimizing the weighted mean square error function, weighted ELM allows the training instances nearer to the query to offer bigger contributions to the estimated output, so that optimal networks can be obtained. Zong et al. [19] generalized weighted ELM and proposed a unified solution of weighted ELM to tackle the imbalanced classification tasks.

Weighted ELM assumed that the balance can be achieved through the equality of misclassification costs. This method does make ELM difficult to ignore minority samples. But it also makes more of the majority class samples misclassified. In comparison with the original ELM, the experiment results showed that the G-mean value increased but the accuracy value decreased. The fact is that the weighted ELM ignores some reasonable distribution of the imbalance data, which leads to the over-extension of the class boundary. Moreover, weighted ELM even focuses on recognizing outlier samples of minority class, so that the noise immunity of the classifier is weakened. Therefore, the trained classifier does not necessarily get the best G-mean and accuracy values.

In this paper, the optimal method is to set the weight based on the sample number of each class, but the weight sum values of each class are not necessarily equal. This paper proposed a decaying weight setting method, to assign the weight sum of each class based on the number of samples. The majority class has larger weight sum but smaller weight, while minority class has smaller weight sum but larger weight. By optimal selection of decaying coefficient, the weighted ELM can adjust the weight sum value of each class and then adjust class boundary. In the experiment, the method proposed by this paper can increase the correct classification of the minority class without much misclassification of majority class. Our method significantly extends the capacity of weighted ELM to deal with imbalanced tasks and improves antinoise performance. It can be applied to different imbalanced problems of both binary and multiclass classification.

This paper is organized as follows. Section 2 outlined the related principle of unweighted ELM and weighted ELM. Section 3 analyzed the influence of weighted settings on the classification boundary offset and proposed the decaying weight setting method. Experiments results are analyzed in Sect. 4. Section 5 introduced the application of the proposed method in the traffic congestion prediction. Section 6 ends this paper with a conclusion and future work.

2 Related principle

2.1 Unweighted extreme learning machine

For N arbitrary distinct samples $(\mathbf{x_i}, \mathbf{t_i})$, where $\mathbf{x_i} = [x_{i1}, x_{i2}, \ldots x_{in}]^T \in \mathbf{R}^n$ and $\mathbf{t_i} = [t_{i1}, t_{i2}, \ldots t_{im}]^T \in \mathbf{R}^m$, standard SLFNs with L hidden neurons and activation function g(x) are mathematically modeled as:

$$\sum_{i=1}^{L} \boldsymbol{\beta}_\mathbf{i}\, \mathrm{g}(\mathbf{w_i} \cdot \mathbf{x_j} + bi) = \mathrm{h}\left(\mathbf{x_j}\right) \boldsymbol{\beta},\ j = 1, 2, \ldots N \tag{1}$$

where $\mathbf{w_i} = [w_{i1}, w_{i2}, \ldots w_{in}]^T$ is the weight vector connecting the ith hidden neuron and the input neurons, $\boldsymbol{\beta} = [\boldsymbol{\beta}_\mathbf{1}, \boldsymbol{\beta}_\mathbf{2}, \ldots \boldsymbol{\beta}_\mathbf{L}]^T$ is the weight vector connecting the ith hidden neuron and the output neurons, and bi is the deviation of the ith hidden neuron. $\mathbf{w_i} \cdot \mathbf{x_j}$ is the inner product of $\mathbf{w_i}$ and $\mathbf{x_j}$.

Aiming to minimize the training errors and maximize the marginal distance between classes, the goal of ELM is:

$$\text{Minimize: } \frac{1}{2}\|\boldsymbol{\beta}\|^2 + \frac{1}{2}C\sum\nolimits_{i=1}^{N}\|\boldsymbol{\varepsilon}_\mathbf{i}\|^2 \tag{2}$$

$$\text{Subject to: } \mathrm{h}(\mathbf{x_i})\boldsymbol{\beta} - \mathbf{t_i} = \boldsymbol{\varepsilon}_\mathbf{i},\ i = 1, 2, \ldots, N \tag{3}$$

where C is a regularization parameter to represent the trade-off between the minimization of training errors and the maximization of the marginal distance. Here $\boldsymbol{\varepsilon}_\mathbf{i}$ is the training error vector of the m output nodes with respect to the training sample $\mathbf{x_i}$. According to KKT theorem [20], the solution is as follows:

$$\text{when } N < L: \ \boldsymbol{\beta} = \mathbf{H}^\mathbf{T}\left(\frac{\mathbf{I}}{C} + \mathbf{H}\mathbf{H}^\mathbf{T}\right)^{-1}\mathbf{T} \tag{4}$$

$$\text{when } N > L: \ \boldsymbol{\beta} = \left(\frac{\mathbf{I}}{C} + \mathbf{H}^\mathbf{T}\mathbf{H}\right)^{-1}\mathbf{H}^\mathbf{T}\mathbf{T} \tag{5}$$

where $\mathbf{H} = [\mathrm{h}\left(\mathbf{x_i}\right), \ldots, \mathrm{h}\left(\mathbf{x_N}\right)]^T$ is the hidden layer output matrix, and $\mathbf{T} = [\mathbf{t_1}, \ldots, \mathbf{t_N}]^T$ is the target vector.

2.2 Weighted extreme learning machine

To maximize the marginal distance and to minimize the weighted cumulative error with respect to each sample, the optimization problem of the imbalanced learning can be written mathematically as:

$$\text{Minimize: } \frac{1}{2}\|\boldsymbol{\beta}\|^2 + \frac{1}{2}C\mathbf{W}\sum\nolimits_{i=1}^{N}\|\boldsymbol{\varepsilon}_\mathbf{i}\|^2 \quad (6)$$

$$\text{Subject to: } \mathrm{h}(\mathbf{x_i})\boldsymbol{\beta} - \mathbf{t_i} = \boldsymbol{\varepsilon}_\mathbf{i}, i = 1, 2, \ldots, N \quad (7)$$

where **W** is a weighted matrix, and it is determined by the data sets. If a sample comes from the minority class, we can receive the large weight. According to KKT theorem [20], the equivalent dual optimization problem can be converted into Lagrange equation as:

$$\mathrm{L}(\boldsymbol{\beta}, \boldsymbol{\varepsilon}, \boldsymbol{\alpha}) = \frac{1}{2}||\boldsymbol{\beta}||^2 + \frac{1}{2}C\mathbf{W}\sum\nolimits_{i=1}^{N}||\boldsymbol{\varepsilon}_\mathbf{i}||^2 - \sum\nolimits_{i=1}^{N}\alpha_i(\mathrm{h}(\mathbf{x_i})\boldsymbol{\beta} - \mathbf{t_i} + \boldsymbol{\varepsilon}_\mathbf{i}) \quad (8)$$

where $\alpha_i \in R^m$ is the Lagrange operator. Then by making the partial derivatives with respect to variables all equal to zero, we can get the solution of the Lagrange equation as:

$$\begin{cases} \frac{\partial L(\boldsymbol{\beta},\boldsymbol{\varepsilon},\boldsymbol{\alpha})}{\partial \boldsymbol{\beta}} = 0 \\ \frac{\partial L(\boldsymbol{\beta},\boldsymbol{\varepsilon},\boldsymbol{\alpha})}{\partial \boldsymbol{\varepsilon}} = 0 \\ \frac{\partial L(\boldsymbol{\beta},\boldsymbol{\varepsilon},\boldsymbol{\alpha})}{\partial \alpha} = 0 \end{cases} \rightarrow \begin{cases} \boldsymbol{\beta} = \mathbf{H^T}\boldsymbol{\alpha} \\ \boldsymbol{\alpha} = -C\mathbf{W}\boldsymbol{\varepsilon}^\mathbf{T} \\ \mathbf{H}\boldsymbol{\beta} - \mathbf{T} - \boldsymbol{\varepsilon} = 0 \end{cases} \quad (9)$$

Similar to unweighted ELM, the solution of weighted ELM also has two version according to the size of N and L.

$$\text{when } N < L: \boldsymbol{\beta} = \mathbf{H^T}\left(\frac{\mathbf{I}}{C} + \mathbf{WHH^T}\right)^{-\mathbf{1}}\mathbf{WT} \quad (10)$$

$$\text{when } N > L: \boldsymbol{\beta} = \left(\frac{\mathbf{I}}{C} + \mathbf{H^TWH}\right)^{-\mathbf{1}}\mathbf{H^TWT} \quad (11)$$

In binary classification problem, given a new sample **x**, the output function of the ELM classifier can be written as $\mathrm{f}(\mathbf{x}) = \mathrm{signh}(\mathbf{x})\,\boldsymbol{\beta}$. In multiclass problem, class label is expanded into a binary label vector of length m, and the output function vector $\mathrm{f}(\mathbf{x}) = [f_1(\mathbf{x}), \ldots f_m(\mathbf{x})]^T$ corresponds to each label. The predicted label of classification result is $\mathrm{label}(\mathbf{x}) = \mathrm{argmax} f_i(\mathbf{x}), i = 1, 2, \ldots, m$.

3 Decay-weighted extreme learning machine

3.1 Weight setting in previous work

The core of weighted ELM is the calculating of weight matrix **W**. For $\mathbf{W} = \mathrm{diag}(W_{11}, W_{22}, \ldots W_{NN})$, and Deng et al. [18] proposed a setting method to deal with the influence generated by dispersed pots:

$$W_{ii} = \begin{cases} 1 & \left|\varepsilon_i/\hat{s}\right| \le c1 \\ \frac{c2-\left|\varepsilon_i/\hat{s}\right|}{c2-c1} & c1 \le \left|\varepsilon_i/\hat{s}\right| \le c2 \\ 10^{-4} & \left|\varepsilon_i/\hat{s}\right| > c2 \end{cases} \quad (12)$$

where $\hat{s}$ is the standard deviation of ε_i; $\varepsilon_i = -\frac{\alpha_i}{C}$ is the sample error of original ELM; $c1 = 2.5$; $c2 = 3$.

Zong et al. [19] applied weighted ELM to the research of imbalanced data, while the weights are different from the former:

$$\mathbf{W1}: \quad W_{ii} = 1/N(t_i)$$

$$\mathbf{W2}: \begin{cases} W_{ii} = 0.618/N(t_i) & N(t_i) > AVG(N(t_i)) \\ W_{ii} = 1/N(t_i) & N(t_i) \le AVG(N(t_i)) \end{cases} \quad (13)$$

where $N(t_i)$ is the number of samples that belong to class label t_i. W1 is determined by the number of samples, and the weight sums of each class $S(t_i) = W_{ii}N(t_i)$ are equal to 1. W2 sets $S(t_i)$ as golden partition coefficient 0.618 to further reduce the weights of the majority class data.

The original ELM algorithm essentially aims to reduce the training error. If the misclassification costs of each sample are equal, it is easy to cause boundary expansion of the majority class. The worst condition is that all the samples of minority class are divided into majority class by mistake.

Therefore, Zong et al. [19] use weight to balance the misclassification cost and compress the classification boundaries of the majority class. The weighted ELM achieves balanced classification by balancing the sum of misclassification costs. In the matrix W1, due to the equality of the weight sum of each class, single sample in majority class will have smaller weight value than minority class. As a result, the boundary of minority class will expand, and more samples will be classified as the minority class. However, there will be also many majority samples misclassified as minority class. In the matrix W2, the weight of majority class is further decreased, and the boundary of minority class will expand further. In this situation, the number of misclassified samples will greatly increase.

However, this weight setting model (weight sum S fixed at 1 or 0.618) lacks flexibility. The equal misclassification cost does not guarantee the optimal classification boundary. In original ELM, the ratio of sum misclassification cost of each class is equal to the ratio of the sample's quantity. To a certain extent, this ratio also reflects the sample distribution in each category. The simple setting of W1 and W2 simply increased the misclassification cost of minority class, which may result in weak antinoise performance and over expansion of class boundary. Therefore, the classification accuracy decreased.

In this paper, we consider that the weight sums should be more flexible, assigned as different value according to the size of each class, so that an optimal boundary can be generated.

3.2 Movement of boundary with different weight sum

This section analyzes the influence of weight sum on the imbalanced classification. Without loss of generality, consider a binary classification problem with majority samples labeled as negative class and minority samples labeled as positive class. The misclassification cost of the weighted ELM can be written as:

$$\begin{aligned} \text{Cost} &= \frac{1}{2}\|\boldsymbol{\beta}\|^2 + \frac{1}{2}C\mathbf{W}\sum\nolimits_{i=1}^{N}\|\boldsymbol{\varepsilon}_{\mathbf{i}}\|^2 \\ &= \frac{1}{2}\|\boldsymbol{\beta}\|^2 + \frac{1}{2}C\sum\nolimits_{i=1}^{N}\mathbf{W_{ii}}\|\boldsymbol{\varepsilon}_{\mathbf{i}}\|^2 \\ &= \frac{1}{2}\|\boldsymbol{\beta}\|^2 + \frac{1}{2}C\left(\sum_{t_i=t_+}\frac{S(t_+)}{N(t_+)}\|\boldsymbol{\varepsilon}_{\mathbf{i}}\|^2 + \sum_{t_i=t_-}\frac{S(t_-)}{N(t_-)}\|\boldsymbol{\varepsilon}_{\mathbf{i}}\|^2\right) \end{aligned} \tag{14}$$

where t_+ represents positive label of minority class, and t_- represents negative label of majority class. In order to minimize the cost, when the value $S(t_-) : S(t_+)$ decreases, the classifier prefers to set a sample as positive label and vice versa.

In Fig. 1, the number of the majority samples is 8 times the number of minority samples. Obviously, in the figure (a), when the proportion of weight sum is set as $S(t_-) : S(t_+) =$ 8:1, the weighted ELM becomes no different to the original ELM. Here the boundary was pushed toward the minority sample to a large distance. This is the usual phenomenon of imbalanced classification.

As can be seen in figure (b) to (e), with the increase in the proportion of weight sum, the classification boundaries also continue to move and extend toward the majority samples. When the weight sums of these two classes become equal (W1 weight setting), the boundary has been remarkably close to the majority class. Intuitively, the W1 weight is unable to make sure that the boundary is in the reasonable position and the W2 weight leads to the overexpansion of boundary. However, as can be seen in figure (c), when the proportion of weight sum is adjusted properly, the boundary position would be more appropriate.

3.3 Decay-weight setting

In this paper, we think that if the weights are adjusted in a simple way, the weighted ELM can generate more reasonable classification boundaries when dealing with imbalanced data. Based on the analysis in Sect. 3.2, in the binary classification problem, the weight sum of majority class should not be less than minority class, in case of boundary over expansion, while the ratio of the two class should not exceed the ratio of the number of samples to set more weight on minority samples:

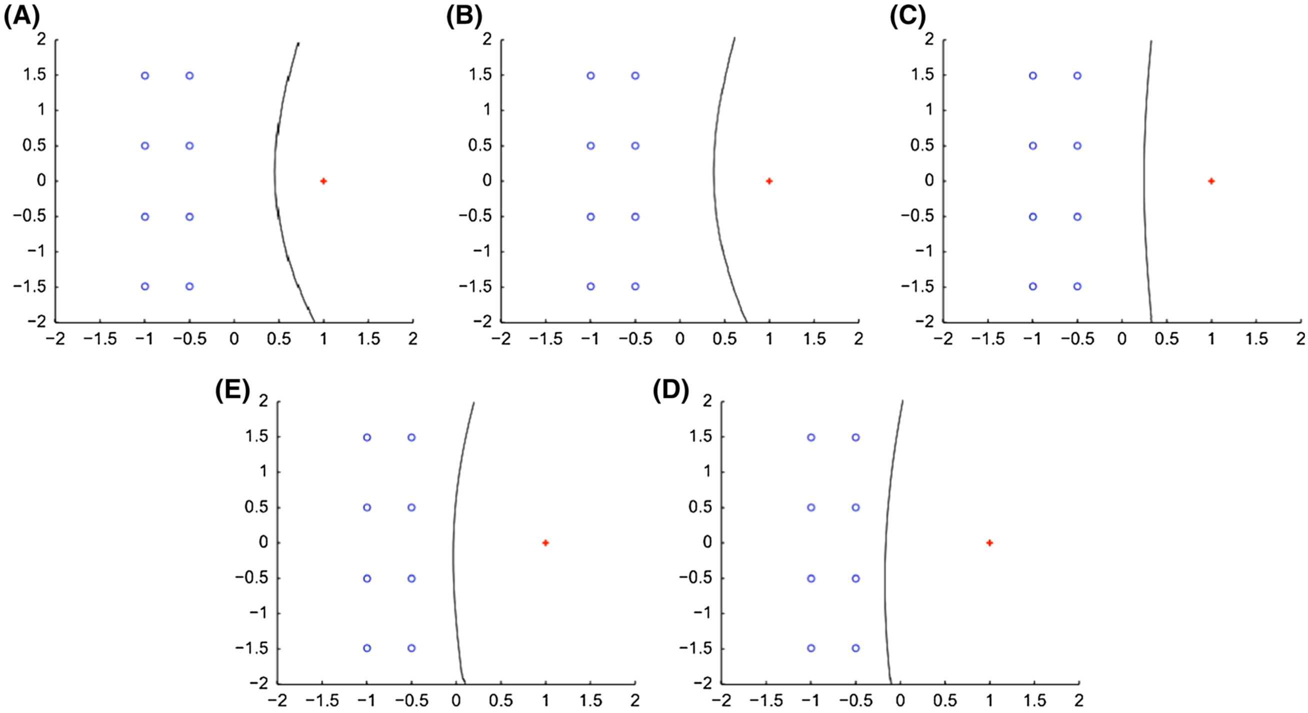

Fig. 1 The decision boundary produced by weighted ELM with different weight sum setting on the synthetically generated data set: The *blue circle* represents majority class, the *red cross* represents minority class, and the parameter C is fixed at 1: **a** S(*blue*):S(*red*) = 8:1 (the same as original ELM); **b** S(*blue*):S(*red*) = 2:1; **c** S(*blue*):S(*red*) = 1.5:1; **d** S(*blue*):S(*r*) = 1:1 (W1 setting); **e** S(*blue*):S(*red*) = 0.618:1 (W2 setting) (color figure online)

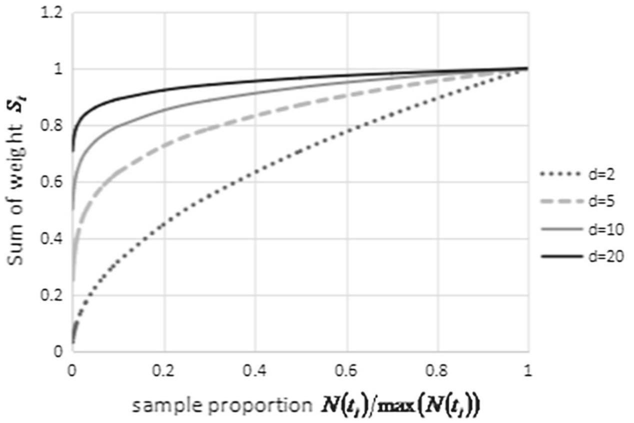

Fig. 2 The relationship between the sum of weight and sample proportion

$$1 \le S(t_-) : S(t_+) \le N(t_-) : N(t_+) \tag{15}$$

Extended to multiclass problems, the weight sum of any two classes should also follow this principle. Therefore, the weight sum value $S(t_i)$ of each class should be positively correlated with the number of samples in each class. When the number of samples is reduced, the weight sum is also reduced, but the reducing speed is slower than the situation of unweighted ELM.

This weight is called decay-weight, so that the new method is called as decay-weighted ELM (DW-ELM). The decay-weight is set as follows:

$$\mathbf{DW}: \quad W_{ii} = \frac{\sqrt[d]{N(t_i)/\max(N(t_i))}}{N(t_i)} \tag{16}$$

With the decrease in the sample number $N(t_i)$, the weight sum $S_i = \sqrt[d]{N(t_i)/\max(N(t_i))}$ decays. As can be seen in Fig. 2, the slope of the curve represents the velocity of decay. Decaying velocity is slow at the beginning. When the proportion $N(t_i)/\max(N(t_i))$ is too small, decaying velocity will be fast. The decaying parameter d is related to decaying effect. The larger the decaying parameter d, the more important the minority class. The relationship between the decaying parameter d and the sum of weight is shown in Fig. 2. When $d = 1$, the weight of each sample will be the same and the DW-ELM will translate to original ELM. Therefore, by adjusting the decaying parameter, the decaying velocity could float between original ELM and W1-ELM.

4 Experimental results

4.1 Experimental environment

The experiment was implemented in MATLAB R2013b on a 3.40 GHz machine with 4GB of memory. In this paper, the experiment used six data sets from UCI [21], including 3 binary classification data and 3 multiclass classification data. The experimental data sets are shown in Table 1.

The unbalance rate in Table 1 is calculated as follows. In binary classification data, the unbalance rate is $R = N(t_+)/N(t_-)$. In multiclass classification data, the unbalance rate is $R = \min(N(t_i))/\max(N(t_i))$, which means the ratio of the minimum sample size to the maximum sample size.

4.2 Comparison in unbalanced classification performance

In the experiment, different methods are taken in comparison. The classification results of different models are taken into comparison including the original ELM, W1-ELM, W2-ELM and DW-ELM. Besides, under-sampling [16] and over-sampling approaches [17] are also taken into comparison, before the original ELM completes the learning task on the resampling data.

In the experiment, all the activation functions in the hidden nodes of ELM are sigmoid function. The regularization parameter C is chosen from $\{2^{-20}, 2^{-19}, \ldots 2^{19}, 2^{20}\}$, and the decaying parameter d is chosen from $\{2, 3, \ldots, 20\}$. The number of hidden nodes is set as 1000, because when there are a large number of hidden nodes, the classifier is insensitive to the number of hidden nodes. In the experiment result, the comparison in G-mean is shown in Table 2, and the comparison in accuracy is shown in Table 3.

From these tables, we can conclude that weighted optimization method performs much better than the resampling method, especially on the Adult, Statlog shuttle and Satimage data set. In most instances, the DW-ELM achieved a higher G-mean value compared with the others. In the binary classification, the results of DW-ELM is near to the W1-ELM, and the DW-ELM is better than the W1-ELM except that on the Adult data set. While in the multiclass classification, the effect of DW-ELM is much better than the others.

In most cases, the optimization scheme of the unbalanced problem is to improve the recognition rate of the minority class by reducing the recognition rate of the majority class, so that the accuracy is reduced in general. However, the accuracy of DW-ELM is higher than the original ELM, which means that the DW-ELM not only can improve the correct classification of minority classes, but also to keep the correct classification of majority classes at the same level as ELM. At the same time, the accuracy of DW-ELM is much higher than W1-ELM in the multiclass classification, which shows that DW-ELM can be more effective to distinguish between different types of imbalance, and improving the classification performance.

In addition, the greatest feature of ELM is the fast speed. The training speed of weighted ELM and the original ELM is compared in this paper, and the result is shown in Fig. 3. Because the weight optimization method only adds the pro-

Table 1 Data sets information

Data sets	Characteristic number	Class number	Training data	Testing data	Unbalance rate
Adult	123	2	4781	27,780	0.3306
Banana	2	2	400	4900	0.8605
Wisconsin	9	2	546	137	0.5381
Statlog shuttle	9	7	8000	50,000	0.0726
Satimage	36	6	4435	2000	0.3871
USPS	256	10	7291	2007	0.4733

Table 2 The value of G-mean (%)

Data sets	Original ELM	Under-sampling + ELM	Over-sampling + ELM	W1-ELM	W2-ELM	DW-ELM
Adult	73.54	79.42	76.86	**81.67**	80.42	81.22
Banana	86.98	87.43	88.37	89.13	89.04	**89.33**
Wisconsin	94.85	96.34	96.32	97.07	96.97	**97.18**
Statlog shuttle	92.44	93.48	92.84	94.14	93.93	**96.1**
Satimage	86.92	87.16	88.07	89.56	89.03	**90.62**
USPS	94.53	95.78	95.39	96.39	96.28	**98.64**

The bold values mean the best performance of learning method in each data set

Table 3 The value of accuracy (%)

Data sets	Original ELM	Under-sampling + ELM	Over-sampling + ELM	W1-ELM	W2-ELM	DW-ELM
Adult	84.66	82.03	81.56	83.41	82.1	**84.79**
Banana	89.71	87.82	88.46	89.84	89.8	**90**
Wisconsin	**97.66**	96.49	95.67	97.65	93.37	97.22
Statlog shuttle	94.35	92.67	92.59	94.1	93.09	**95.29**
Satimage	90.45	89.54	89.72	90.4	90	**91.35**
USPS	97.11	95.68	94.87	96.96	96.76	**97.9**

The bold values mean the best performance of learning method in each data set

cess of calculating the weight matrix, a diagonal matrix, the training time DW-ELM is similar to the original ELM.

In order to analysis the promotion both in the G-mean and accuracy, the experiment takes the Banana data set as an example. Figure 4 shows how the class boundary distributes, respectively, with 3 different weight settings.

The figure (a) clearly shows that W1-ELM balanced weight sum of each class, and the minority class can get wider distribution boundary. However, many of the majority class samples is also divided into the minority class, so that the G-mean value increased but the accuracy value decreased. In figure (b), W2-ELM set much less weight on majority class with golden section number, so that the boundary of minority class gets further more expansion. Even the outlier sample is also correctly classified, but the price is that too many majority class samples around the outlier were misclassified. This leads to the decrease in G-mean values rather than increase. At the same time, the accuracy is also greatly reduced. It can be seen that both W1 and W2 may cause the boundary to expand excessively, resulting in the weak antinoise ability.

In figure (c), based on DW setting, the weight of minority class is decreased to a certain extent, which properly compresses the boundary. Some minority samples with good differentiation are still classified correctly and given a large boundary range, while the outlier sample is abandoned, leading to much more majority samples being recognized correctly. Therefore, the decaying weight setting is able to improve the recognition rate of minority class to improve the G-mean value, at the same time, to ensure the correct classification of the majority class, keeping a good overall accuracy.

4.3 Performance sensitivity on parameters

Experiments on binary problem data set Adult and multiclass problem data set Satimage analyzed the influence of differ-

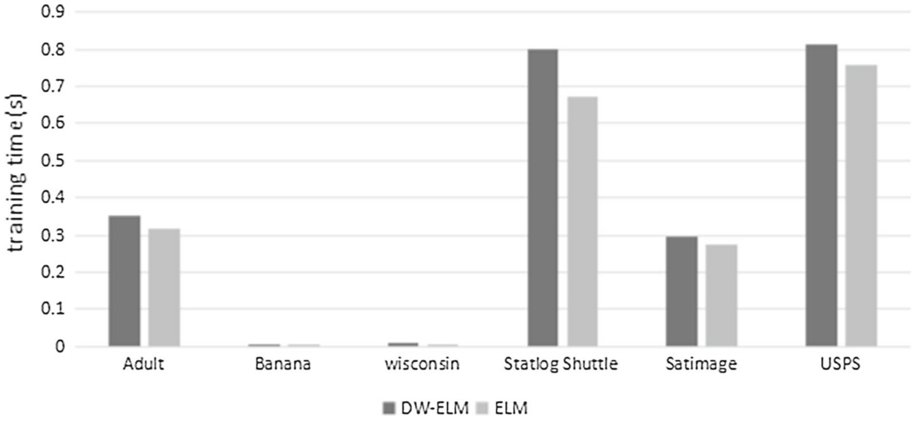

Fig. 3 Training efficiency comparison

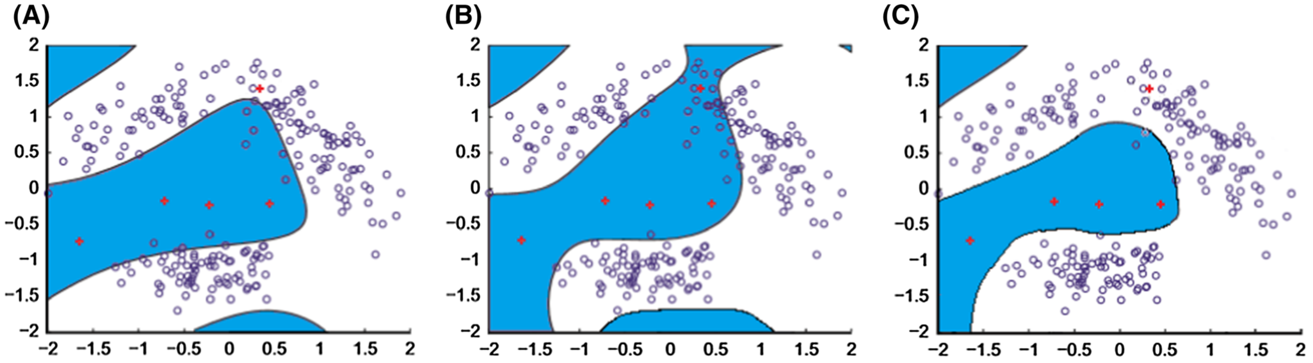

Fig. 4 Class boundary distribution on Banana data set in different weight setting: The *blue circle* represents majority class, and the *red cross* represents minority class. **a** W1-ELM, **b** W2-ELM, **c** DW-ELM (color figure online)

ent parameters on the effect of DW-ELM learning. Figures 5 and 6 show the effects of different regularization coefficients and the number of hidden nodes. As can be seen in these figures, when the number of hidden nodes increases, the accuracy and G-mean will stay in a stable level, so that this parameter is usually assigned to a high value. However, the performance of DW-ELM is sensitive to the value of regularization coefficients C. Therefore, when training the classifier, regularization coefficients C should be optimally chose from a series of values.

The decaying coefficient also has much influence on the DW-ELM. The effect of decaying coefficient on DW-ELM is shown in Fig. 7. The low value of decaying coefficient leads to low weight value on minority class, so that more minority class samples will be classified incorrectly and the G-mean value will be relatively low. With the increasing of decaying coefficient, the accuracy and G-mean increases and finally stays stable. However, there is a best value for decaying coefficient to make the accuracy and G-mean reach the peak. Because of the sensitivity to decaying coefficient, the DW-ELM should select optimal value through repeated test.

In conclusion, as can be seen from the experiment results, on the two classification data sets, DW-ELM makes less significant improvement upon the DW-ELM. However, on the multiclass classification data sets, the new method gets much better results both in the G-mean and accuracy. So the new method has more advantages in solving the unbalanced multiclass classification problems.

5 Application in traffic congestion prediction

5.1 Traffic congestion prediction system

Urban Transportation Assessment and Forecast System analyzes the traffic congestion of transportation network in a city of southwest China. This system shows the evaluation results of the real-time traffic states on the GIS map with different colors, on the foundation of the floating cars' GPS information including license plate basic information, speed, direction and position. Based on history information, this system forecasts the future traffic conditions based on history information to give suggestion for traffic control.

As can be seen in Fig. 8, traffic congestion prediction is an important part of core function whose data source consists of: road sections information (containing road grades, the number of lanes, the number of neighborhood lanes) and floating car data. We preprocess the floating car data and

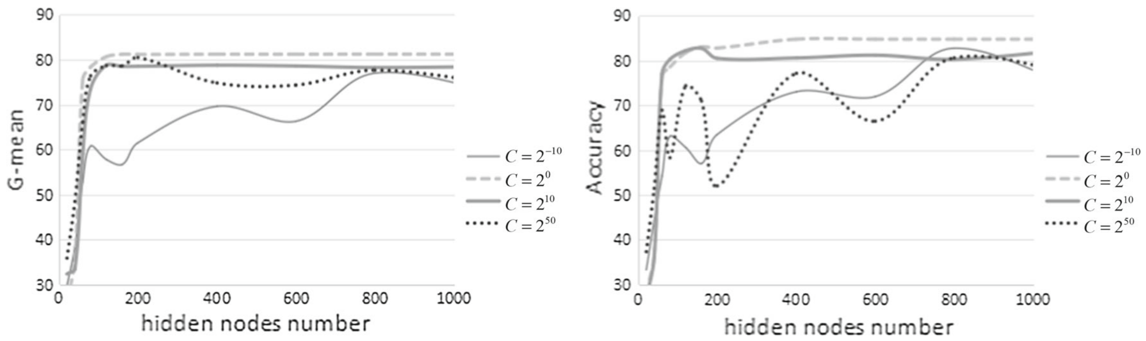

Fig. 5 Effects of different regularization coefficients and the number of hidden nodes on Adult data sets

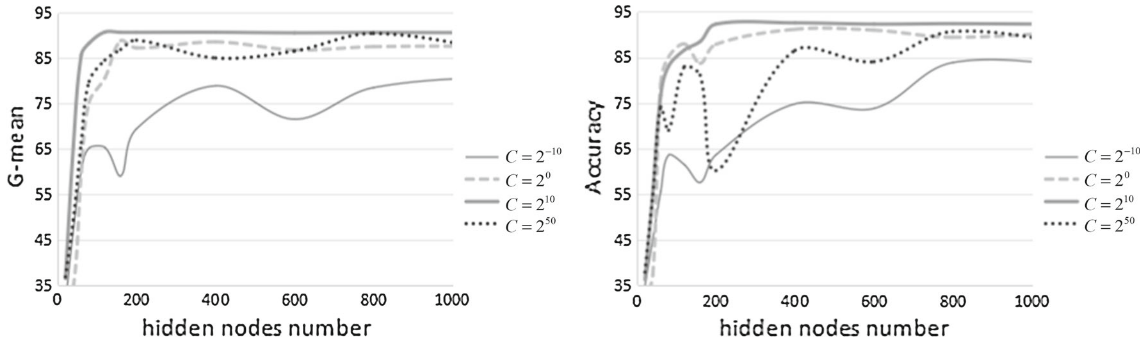

Fig. 6 Effects of different regularization coefficients and the number of hidden nodes on Satimage data sets

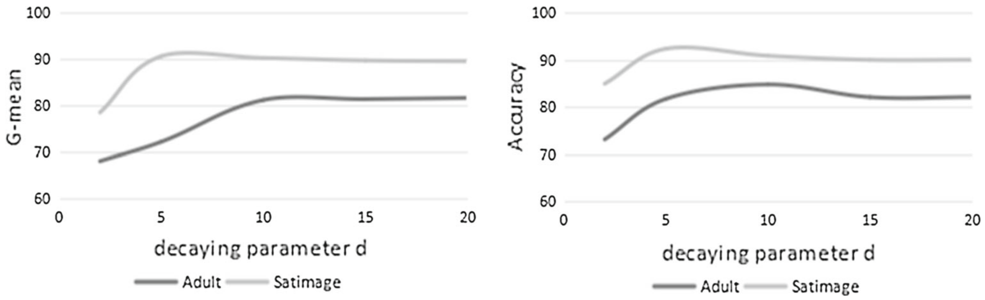

Fig. 7 Influence of decaying coefficient on G-mean and accuracy

match the effective floating car speed information to the every road section. Then, the eigenvalues of road section traffic flow are be able to calculated.

5.2 Unbalanced distribution in traffic congestion

According to the floating car data and the GPS position, the speed information of the floating car can be matched to the surrounding road. Then the average speed of each road section in a certain period can be calculated, and the road congestion index is based on it.

Traffic congestion index is a quantitative index of road congestion, and the value is from 0 to 100, which provides a unified evaluation value for all roads. In the case of road level determination, the congestion value and the speed are negatively correlated, which means the higher the speed, the smaller the congestion value. The calculation function of traffic congestion index is:

$$f(x) = 100 - \left(\frac{1}{1 + e^{-wx}} - \frac{1}{2}\right) \times 200 \tag{17}$$

Parameter w is used to fine-tune the different road grades. In different road grades, same average speeds can achieve different congestion value. The congestion prediction is also implemented, respectively, in different road grades. Previously, the prediction model is based on the balance data. However, the speed distribution is obviously unbalanced in each kind of road (Fig. 9), so that the prediction result can hardly get correct prediction in some key point such as rare smooth condition or congestion.

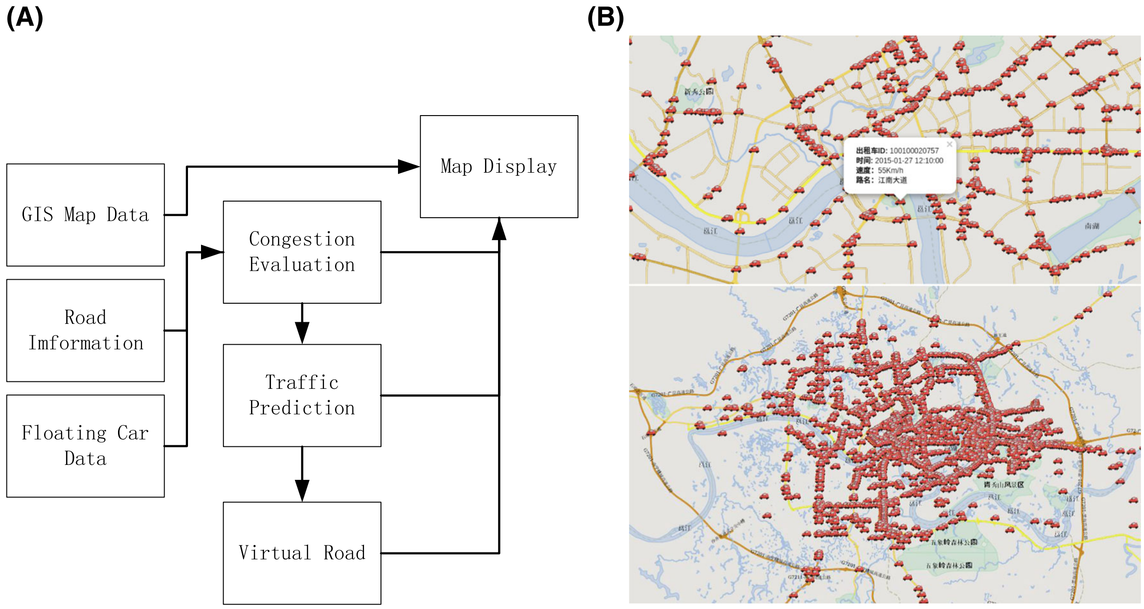

Fig. 8 Urban Transportation Assessment and Forecast system **a** system structure, **b** floating car distribution on the map

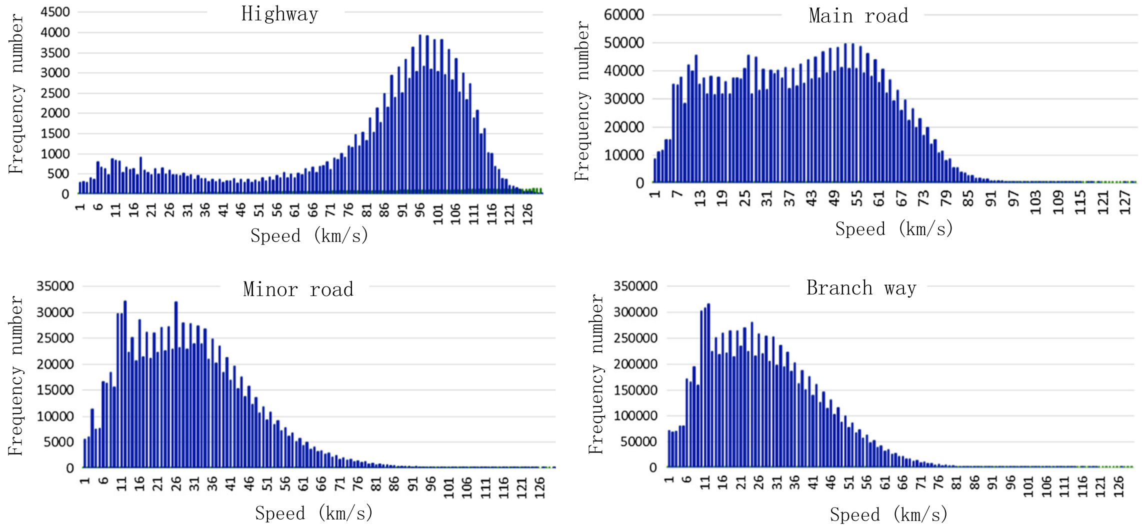

Fig. 9 The speed distribution of floating cars in different road grade

5.3 Decay-weighted ELM in traffic congestion

In order to optimize the prediction model, the Decay-Weighted ELM was used in this application. In this section, the experiment used the traffic data in March 6, 2015 to test the feasibility and performance of DW-ELM. Traffic congestion prediction system takes the road sections as the individual samples. Specifically, a road section demonstrates a portion of a road in a single direction. Its traffic congestion prediction originates from two sources.

The first source is the essential information of the road section from the transportation department. The extracted information includes number of lanes, number of entrance and exit, number of traffic lights, and road grades.

The second source is the real-time speed information of the road section from the floating car data. In the interval ΔT

Table 4 Prediction result on realistic traffic data (%)

	Original ELM	W1-ELM	W2-ELM	DW-ELM
G-mean	82.97	87.16	87.10	**87.79**
Accuracy	91.76	90.84	89.80	**92.08**

The bold values mean the best performance of learning method in each data set

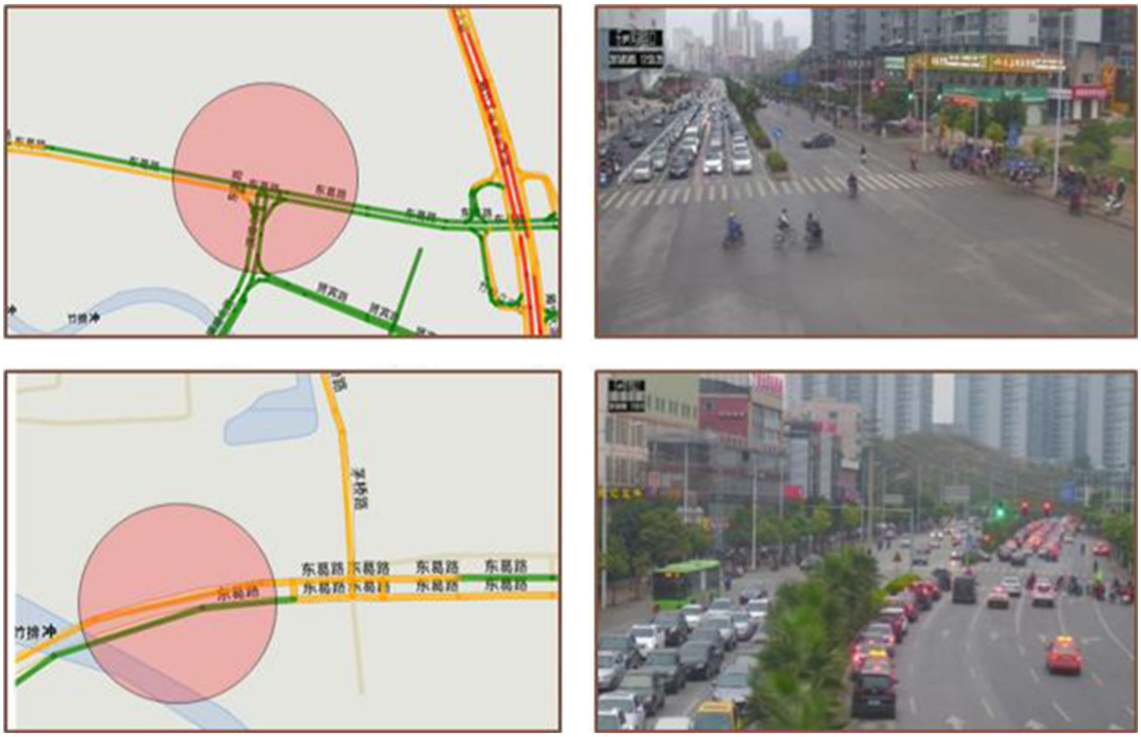

Fig. 10 Comparison between prediction and the real condition after 5 min.

defined by $\Delta T = 5$ min, the data are calculated and matched to the corresponding road section and are transformed to the some kinds of eigenvalues including average speed, average stopping time and current time.

From all kinds of eigenvalues above, 12-dimensional traffic congestion eigenvalue can be calculated. The data are grouped in interval for 5 min and matched to the corresponding road section. Finally, we collect 1,515,446 samples, and each sample will be used to predict the congestion condition after 5 min. The data set was randomly divided into 4 groups for cross-validation, each group took turns as the test set, and the other 3 groups were the training set. In order to carry out the accuracy and G-mean, the congestion level was divided in to 10 levels. If the congestion values of a prediction and the corresponding real data stay in the same level, this prediction can be considered to be correct.

For comparison, we tested the original ELM, the W1-ELM, W2-ELM and the DW-ELM. Table 4 shows that the prediction model trained by DW-ELM had the highest average G-mean and accuracy at 87.79 and 92.09%.

The trained model was used in the Urban Transportation Assessment and Forecast System. Figure 10 displays the real-time traffic condition prediction. In the map, Green represents smooth traffic, yellow shows average condition, and red means the road is congested. As can be seen in the image taken by surveillance cameras, the traffic prediction accurately reflects the road traffic congestion at that time.

6 Conclusions

This paper improved the original ELM by combining weights and ELM and optimized the former weight model. The proposed Decay-weighted ELM sets the weight based on the sample number of each class, but the weight sum values of each class are not necessarily equal. When the number of samples is reduced, the weight sum is also reduced. By adjusting the decaying velocity, classifier could achieve more appropriate boundary position. The experiment results show that the Decay-weighted ELM obtains the better effects in solving the imbalance classification tasks, particularly in multiclass tasks. In the application of urban traffic congestion prediction, the DW-ELM method significantly promoted the prediction performance. However, there are two parameters (regularization parameter C and decaying parameter d) need to adjust through repeated test, which makes the classifier construction process in application more complex. Therefore, the future work is to research how to simplify parameter settings and to find a method to set decaying parameter d directly.

Acknowledgements This work was supported by National Nature Science Foundation of P. R. China (No. 61272357, 61300074) and National Key Research and Development Program of China (No. 2016YFB0700502, 2016YFB1001404).

References

1. Huang, G.B., Zhu, Q.Y., Siew, C.K.: Extreme learning machine: theory and applications. Neurocomputing **70**(1–3), 489–501 (2006)
2. Huang, G.B., Chen, L., Siew, C.K.: Universal approximation using incremental constructive feedforward networks with random hidden nodes. IEEE Trans. Neural Netw. **17**(4), 879–892 (2006)
3. Huang, G.B., Chen, L.: Convex incremental extreme learning machine. Neurocomputing **70**(16–18), 3056–3062 (2007)
4. Huang, G.B., Chen, L.: Enhanced random search based incremental extreme learning machine. Neurocomputing **71**(16–18), 3460–3468 (2008)
5. Huang, G.B., Ding, X., Zhou, H.: Optimization method based extreme learning machine for classification. Neurocomputing **74**(1–3), 155–163 (2010)
6. Huang, G.B., Zhou, H., Ding, X., et al.: Extreme learning machine for regression and multiclass classification. IEEE Trans. Syst. Man. Cybern. B Cybern. A Publ. IEEE Syst. Man Cybern. Soc. **42**(42), 513–529 (2012)
7. Zhong, H., Miao, C., Shen, Z., et al.: Comparing the learning effectiveness of BP, ELM, I-ELM, and SVM for corporate credit ratings. Neurocomputing **128**(5), 285–295 (2014)
8. Huang, G., Song, S., Gupta, J.N., et al.: Semi-supervised and unsupervised extreme learning machines. IEEE Trans. Cybern. **44**(12), 2405–2417 (2014)
9. Wang, P., Wang, D., Feng, W.: Online semi-supervised extreme learning machine based on manifold regularization[J]. J. Shanghai Jiaotong Univ. (Sci.) **49**(08), 1153–1158 (2015)
10. Wang, W., Zhang, R.: Improved Convex Incremental Extreme Learning Machine Based on Enhanced Random Search. Unifying Electrical Engineering and Electronics Engineering, pp. 2033–2040. Springer, New York (2014)
11. Hai-Feng, K.E., Ying, J.: Real-time license character recognition technology based on R-ELM. J. Zhejiang Univ. **48**(2), 1209–1216 (2014)
12. Stolfo, S. J., Fan, W., Lee, W., et al.: Cost-based modeling for fraud and intrusion detection: results from the JAM project. In: DARPA Information Survivability Conference and Exposition, 2000. DISCEX ’00. Proceedings. IEEE Xplore, vol. 2, pp. 130–144 (2000)
13. Strack, B., Deshazo, J.P., Gennings, C., et al.: Impact of HbA1c measurement on hospital readmission rates: analysis of 70,000 clinical database patient records.[J]. Biomed. Res. Int. **2014**(6), 781670 (2014)
14. He, H., Garcia, E.A.: Learning from imbalanced data. IEEE Trans. Knowl. Data Eng. **21**(9), 1263–1284 (2009)
15. Akbani, R., Kwek, S., Japkowicz, N.: Applying support vector machines to imbalanced datasets. Lecture Notes in Computer Science **3201**, 39–50 (2004)
16. Liu, X.Y., Wu, J., Zhou, Z.H.: Exploratory undersampling for class-imbalance learning. IEEE Trans. Syst. Cybern. B Cybern. A Publ. IEEE Syst. Man Cybern. Soc. **39**(2), 539–550 (2009)
17. Chawla, N.V., Lazarevic, A., Hall, L.O., et al.: SMOTEBoost: improving prediction of the minority class in boosting. Lecture Notes in Computer Science **2838**, 107–119 (2003)
18. Deng, W., Zheng, Q., Regularized, Chen L.: Learning, extreme, machine[C], computational intelligence and data mining, : CIDM ’09. IEEE Symposium on. IEEE Xplore **2009**, 389–395 (2009)
19. Zong, W., Huang, G.B., Chen, Y.: Weighted extreme learning machine for imbalance learning. Neurocomputing **101**(3), 229–242 (2013)
20. Fletcher, R.: Practical Methods of Optimization: Constrained Optimization. Practical Methods of Optimization, pp. 71–94. John Wiley, Hoboken (1981)
21. Lichman, M.: UCI Machine Learning Repository [http://archive.ics.uci.edu/ml]. Irvine, CA: University of California, School of Information and Computer Science (2013)

Qing Shen is currently working as the Ph.D. candidate at the Department of Computer Science and Technology, School of Computer and Communication Engineering, University of Science and Technology Beijing, China. His main research interests include machine learning, human-computer interaction and intelligent transportation system.

Xiaojuan Ban received the Ph.D. degree from University of Science and Technology Beijing, China, in 2003. She is senior member of China Computer Federation, and currently works as an Professor in the Department of Computer Science and Technology, School of Computer and Communication Engineering, University of Science and Technology Beijing, China. Her current research interests include machine learning, computational intelligence, artificial life, human-computer interaction and design of intelligent system.

Ruoyi Liu is currently working toward the Master’s degree at the Department of Computer Science and Technology, School of Computer and Communication Engineering, University of Science and Technology Beijing, China. Her research interests include machine learning and computational intelligence.

Yu Wang is currently working as the Ph.D. candidate at the Department of Computer Science and Technology, School of Computer and Communication Engineering, University of Science and Technology Beijing, China. He is also working as the researcher in the North Electronic Instrument Institute, China. His main research covers machine vision and machine learning technology in national defense.