Memetic Comp. (2019) 11:27–34
https://doi.org/10.1007/s12293-017-0236-3

REGULAR RESEARCH PAPER

An improved weighted extreme learning machine for imbalanced data classification

Chengbo Lu[1] · **Haifeng Ke**[2] · **Gaoyan Zhang**[2] · **Ying Mei**[1] · **Huihui Xu**[3]

Received: 2 January 2017 / Accepted: 20 June 2017 / Published online: 29 June 2017

Abstract This paper proposes an improved weighted extreme learning machine (IW-ELM) for imbalanced data classification. By incorporating voting method into weighted extreme learning machine (weighted ELM), three major steps are involved in the proposed method: training weighted ELM classifiers, eliminating unusable classifies to determine proper classifiers for voting, and finally determining the classification result based on majority voting. Simulations on many real world imbalanced datasets with various imbalance ratios have demonstrated that the proposed algorithm outperforms weighted ELM and other related classification algorithms.

This work is supported by Natural Science Foundation of China under Grant No. 61373057, First-class Disciplines of Zhejiang Province—Mathematics (Lishui University) and Scientific Research Foundation of Zhejiang Provincial Education Department under Grant No. Y201432787, Y201432200.

✉ Haifeng Ke
kehf@zucc.edu.cn

Chengbo Lu
lu.chengbo@aliyun.com

Gaoyan Zhang
zhanggy@zucc.edu.cn

Ying Mei
mymeiying@tom.com

Huihui Xu
hxu@pacific.edu

[1] School of Engineering, Lishui University, Lishui 323000, China

[2] School of Computer and Computing Science, Zhejiang University City College, Hangzhou 310015, China

[3] School of Engineering and Computer Science, University of the Pacific, Stockton, CA 95211, USA

Keywords Imbalanced datasets · Classification · Weighted extreme learning machine · Optimal weights · Majority voting

1 Introduction

The problem of classification for imbalanced data is one of the top ten studied areas in the data mining and machine learning field today [1]. Imbalanced data classification typically refers to a problem with the classes not equally represented. Applications of imbalanced data classification include a wide variety of "real-world" problems such as detecting fraudulent transactions of credit cards or characterizing the recurrence of breast cancer in patients. Most of classic imbalance studies focus on two-class (binary) classification since multi-class can be decomposed into multiple two-class. Commonly in a two-class classification problem, the majority of data are labeled as the "negative" class while the minority are the "positive" class. Because most of the standard classification learning algorithms [2–4] are designed to tackle balanced datasets, the small "positive" class is often underrepresented when these algorithms are being applied directly on imbalanced datasets. An excellent accuracy may be achieved due to correct classification for almost all samples, but the accuracy only reflects the underlying class distribution and therefore could be a misleading performance measure. Moreover, the positive classes usually have higher interest or importance than the negative class and oftentimes imply higher misclassification cost.

Many techniques have been developed recently to combat imbalanced data classification in machine learning. These techniques can be categorized as either data resampling methods or algorithmic approaches. Data resampling

methods aim to balancing data before applying standard classification learning algorithms. Two well-known strategies can be utilized—oversampling or undersampling. Synthetic minority over-sampling technique (SMOTE) [5] is an example of oversampling, in which new data samples are generated between the minority data and their selected neighbors. Tomek links method [6] and Wilson's editing [7] are undersampling methods, in which a fraction of the majority data are removed. Balancing data by resampling is helpful in reducing misclassification error, however, the effectiveness is not consistent but problem dependent [8]. On the other hand, new machine learning classification algorithms or modification of existing ones are also developed, such as one-class classifiers [9] and cost-sensitive learning [10]. As a popular algorithmic approach, cost-sensitive classifier incorporates cost sensitivity into machine learning process, i.e., a different misclassification cost is assigned for each particular example [11]. However, cases may arise in the real world applications when the misclassification cost matrix is difficult to generate.

Zong et al. [12] proposed a competitive cost-sensitive learning approach to automatically generate the misclassification cost matrix in accordance with class distribution, known as weighted extreme learning machine (weighted ELM). A weighting that is inversely proportional to the number of samples is assigned, which strengthens the impact of the positive class in imbalanced datasets. While maintaining the advantages from the original unweighted ELM, i.e., convenient implementation and easy application on multi-class data classification, weighted ELM demonstrates better performance on imbalanced datasets compared to unweighted ELM (i.e., ELM). However, the weighting schemes presented in Zong et al. induce a certain amount of misclassified samples from the negative class in compromise for a better accuracy in the positive class. Therefore, other ELM based approaches have been proposed. For instance, Zhang and Ji [13] developed fuzzy ELM (FELM): a fuzzy matrix is created to change the distributions of penalty factors. However, a unified design strategy for generating the fuzzy matrix is unpresented. In [14], Li et al. developed a Boosting weighted ELM, in which weighted ELM is embedded into a modified AdaBoost framework in order to obtain proper weights. But the computational complexity of algorithm is increased significantly due to iterations. Lin et al. [15] combined resampling technique with ELM; however, its performance is largely dependent on the adopted resampling technique.

In this paper, an improved weighted ELM (IW-ELM) algorithm is proposed, in which a voting based weighting scheme is introduced when assigning appropriate weights adaptively for imbalanced data classification. This paper is organized as follows. Section 2 briefly reviews the fundamental principles of weighted ELM and its issues. Section 3 elaborates the proposed algorithm. Experiment results are presented in Sect. 4. Section 5 compares the computational complexity of IW-ELM with other related classification algorithms using real world imbalanced datasets. Section 6 draws a conclusion.

2 Related work

The proposed algorithm is developed based on ELM and weighted ELM. The basic principles of ELM and weighted ELM are reviewed in Sects. 2.1 and 2.2 respectively. Section 2.2 also presents the problems related to weighted ELM when classifying imbalanced datasets.

2.1 Unweighted ELM

Least square based learning algorithm named ELM [16] was originally proposed for single layer feedforward networks (SLFNs), where input weights of a SLFN are randomly generated, and output weights are trained with batch learning technique of least squares. It has been proved that SLFNs with randomly hidden neurons and tunable output weights have universal approximation and excellent generalization performance [17]. More importantly, ELM outperforms most existed learning algorithms in training speed [18–22] and it has been widely used in applications of face recognition [23], image processing and classification[24], electricity price classification [25], energy commodity futures index forecast [26], location fingerprinting technique [27], protein sequence classification [28], and location classification [29].

The basics of unweighted ELM are briefly outlined as follows:

For any input data $\boldsymbol{x} \in R^n$, the output of a standard SLFN with L hidden nodes can be written as

$$H(\boldsymbol{x}) = \sum_{i=1}^{L} \boldsymbol{\beta}_i G(\boldsymbol{w}_i, b_i, \boldsymbol{x}),\ \boldsymbol{w}_i \in R^n,\ \boldsymbol{\beta}_i \in R^m,$$

where $\boldsymbol{\beta}_i = [\beta_{i1}, \beta_{i2}, \ldots, \beta_{im}]^T$ is the weight vector connecting the ith hidden node to the output nodes, and $G(\boldsymbol{w}_i, b_i, \boldsymbol{x})$ is the output of the ith hidden node.

Consider a set of training pairs with N inputs $\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_N$ (here $\boldsymbol{x}_i \in R^n$), and N target output $\boldsymbol{t}_1, \boldsymbol{t}_2, \ldots, \boldsymbol{t}_N$ (here $\boldsymbol{t}_i \in R^m$), respectively. The mathematical model of a SLFN is

$$\boldsymbol{H}\boldsymbol{\beta} = \boldsymbol{T} \tag{1}$$

with

$$\boldsymbol{H} = \begin{pmatrix} G(\boldsymbol{w}_1, b_1, \boldsymbol{x}_1) & \cdots & G(\boldsymbol{w}_L, b_L, \boldsymbol{x}_1) \\ \vdots & \ddots & \vdots \\ G(\boldsymbol{w}_1, b_1, \boldsymbol{x}_N) & \cdots & G(\boldsymbol{w}_L, b_L, \boldsymbol{x}_N) \end{pmatrix}, \tag{2}$$

Similar to least squares support vector machine (LS-SVM), the output can be formulated by finding the solution for an optimization problem:

$$\text{Minimize } \frac{1}{2}\|\boldsymbol{\beta}\|^2 + \frac{C}{2}\|\boldsymbol{\varepsilon}_i\|^2, \tag{3}$$

$$\text{subject to } \boldsymbol{\varepsilon}_i = H(\boldsymbol{x}_i)\boldsymbol{\beta} - \boldsymbol{t}_i,\ i = 1, 2, \ldots, N, \tag{4}$$

where $\boldsymbol{\varepsilon}_i$ is the training error vector of the m output nodes with respect to the training sample and C is a positive real regularization parameter.

By solving optimization problem (3), the least square solution can be determined as:

$$\begin{aligned}\text{when } N < L:\ \boldsymbol{\beta} &= \tilde{\boldsymbol{H}}^{+}\boldsymbol{T} = \boldsymbol{H}^T\left(\frac{1}{C}\boldsymbol{I} + \boldsymbol{H}\boldsymbol{H}^T\right)^{-1}\boldsymbol{T},\\ \text{when } N \geq L:\ \boldsymbol{\beta} &= \tilde{\boldsymbol{H}}^{+}\boldsymbol{T} = \left(\frac{1}{C}\boldsymbol{I} + \boldsymbol{H}^T\boldsymbol{H}\right)^{-1}\boldsymbol{H}^T\boldsymbol{T}.\end{aligned} \tag{5}$$

Similar to other conventional learning algorithms, the performance of unweighted ELM is highly dependent on class distribution. It performs well with balanced datasets, but imbalanced classification may be problematic. Intuitively, the negative class tends to push the separating boundary towards the positive class side to obtain better classification accuracy for itself (see [12] for more detailed analyses). Weighted ELM algorithm [12] was developed to overcome this problem.

2.2 Weighted ELM

As a cost-sensitive learning algorithm, weighted ELM assigns different weights for each example to minimize misclassification of positive samples and associated cost errors. That is, in weighted ELM, the goal is to maximize the marginal distances:

$$\text{Minimize } \frac{1}{2}\|\boldsymbol{\beta}\|^2 + \frac{C}{2}\boldsymbol{W}\|\boldsymbol{\varepsilon}_i\|^2, \tag{6}$$

$$\text{subject to } \boldsymbol{\varepsilon}_i = H(\boldsymbol{x}_i)\boldsymbol{\beta} - \boldsymbol{t}_i,\ i = 1, 2, \ldots, N, \tag{7}$$

where $\boldsymbol{W}$ is a misclassification cost matrix in accordance with class distribution.

Similar to (5), solutions of $\boldsymbol{\beta}$ can be obtained:

$$\begin{aligned}\text{when } N < L:\ \boldsymbol{\beta} &= \boldsymbol{H}^T\left(\frac{1}{C}\boldsymbol{I} + \boldsymbol{W}\boldsymbol{H}\boldsymbol{H}^T\right)^{-1}\boldsymbol{W}\boldsymbol{T},\\ \text{when } N \geq L:\ \boldsymbol{\beta} &= \left(\frac{1}{C}\boldsymbol{I} + \boldsymbol{H}^T\boldsymbol{W}\boldsymbol{H}\right)^{-1}\boldsymbol{H}^T\boldsymbol{W}\boldsymbol{T}.\end{aligned} \tag{8}$$

In [12], two cost matrices $\boldsymbol{W_1}$ and $\boldsymbol{W_2}$, were proposed.

$$\text{Weighting scheme } \boldsymbol{W_1}: W_{i,i} = \frac{1}{\#(t_i)},\ i = 1, 2, \ldots, m, \tag{9}$$

where $\#(t_i)$ is the number of samples in class t_i.

The imbalanced datasets reach a cardinal balance when scheme $\boldsymbol{W_1}$ is applied.

In comparison, another weighting scheme

$$\boldsymbol{W_2}: \begin{cases} W_{i,i} = \frac{1}{\#(t_i)}, & \text{if } t_i > \text{AVG},\\ W_{i,i} = \frac{0.618}{\#(t_i)}, & \text{if } t_i \leq \text{AVG}, \end{cases}$$

where the golden ratio—0.618—is used and the AVG indicates the averaged sample size per class. Actually $\boldsymbol{W_2}$ is a trade-off between unweighted ELM and weighted ELM $\boldsymbol{W_1}$.

After applying weighting scheme $\boldsymbol{W_1}$ or $\boldsymbol{W_2}$ into unweighted ELM, weighted ELM reduces the misclassification error associated with the small positive class. In another word, the weighting scheme pulls the boundary back towards the negative class side so that additional positive class samples can be correctly classified.

However, assigning higher weights to the positive class simply implies lowering the weights of the negative class. In fact, if the weights of the positive class are too high, then the classification accuracy in the negative class reduces. However, if the weights of the positive class are not high enough, then the classification accuracy in the positive class may be unsatisfactory. As noted in [12], a certain amount of samples from the negative class are misclassified in compromise for a better accuracy in the positive class. Therefore, the key point is to, prior to applying weighted ELM, determine an optimal weight or the most accurate position in the negative class side where the boundary should be pushed towards. However, this is generally not possible because the optimal weight is problem dependent. As shown in Table 1, different weights are evaluated when applying weighted ELM algorithm. Out of 46 tested datasets, only 11 happen to have the optimal weights as $\boldsymbol{W_1}$ or $\boldsymbol{W_2}$.

3 Description of IW-ELM

Since it is not practical to determine an optimal weight when designing a weighted ELM classifier in advance, in IW-ELM, we propose to use a series of "appropriate" weights in place of a single $\boldsymbol{W_1}$ or $\boldsymbol{W_2}$ for imbalanced data classification. For convenience, only binary classification problems are considered. Assume t_1 is the positive class and t_2 is the negative class, $a = \frac{1}{\#(t_1)}$, $b_i =$

Table 1 Performance results of binary problems with imbalance ratio ∈ (0.0078, 0.8605)

G-mean value (%)	Unweighted ELM	Weighted ELM W_1 or W_2	Optimal weighted ELM		VW-ELM	IW- ELM
	Testing result	Testing result	Optimal weight	Testing result	Testing result	Testing result
abalone19	47.52	77.19	$\widetilde{W}_8$	85.85	81.21	85.68
yeast6	70.77	88.29	$\widetilde{W}_7$	93.4	90.94	93.93
yeast5	81.04	95.39	$\widetilde{W}_3$	96.1	96.56	97.95
yeast-1-2-8-9_vs_7	59.23	75.83	$\widetilde{W}_5$	82.15	77.93	79.65
yeast4	65.52	87.29	$\widetilde{W}_{11}$	90.68	86.56	89.84
shuttle-c2-vs-c4	93.54	100	$\widetilde{W}_2$	100	99.19	100
glass5	90.81	96.60	$\widetilde{W}_3$	98.77	97.86	97.77
yeast-2_vs_8	72.83	76.01	$\widetilde{W}_8$	93.33	85.93	92.66
yeast-1-4-5-8_vs_7	61.07	67.10	$\widetilde{W}_8$	71.68	69.61	72.23
glass-0-1-6_vs_5	92.41	98.70	$\widetilde{W}_2$	100	100	100
abalone9-18	75.29	88.72	$\widetilde{W}_3$	96.31	92.54	94.65
page-blocks1	97.78	98.98	$\widetilde{W}_6$	98.94	100	100
glass4	85.72	91.46	$\widetilde{W}_7$	98.77	94.79	98.35
ecoli4	91.96	97.83	$\widetilde{W}_3$	98.43	98.78	98.26
yeast-1_vs_7	65.58	77.26	$\widetilde{W}_8$	84.1	80.74	82.69
shuttle-c0-vs-c4	100	100	$\widetilde{W}_4$	100	100	100
glass2	79.49	83.34	$\widetilde{W}_{10}$	82.16	86.72	84.07
glass-0-1-6_vs_2	67.78	83.77	$\widetilde{W}_8$	81.06	85.65	80.07
vowel0	100	100	$\widetilde{W}_2$	100	100	100
yeast-0-5-6-7-9_vs_4	64.49	81.05	$\widetilde{W}_{10}$	85.39	83.56	87.57
yeast-2_vs_4	86.25	91.56	$\widetilde{W}_4$	96.72	89.32	94.56
page-blocks0	89.92	93.40	$\widetilde{W}_{11}$	97.09	93.98	96.74
ecoli3	77.38	90.17	$\widetilde{W}_3$	93.21	90.78	91.68
yeast3	80.75	93.25	$\widetilde{W}_8$	93.91	95.41	91.04
glass6	94.96	95.90	$\widetilde{W}_5$	98.64	97.08	98.36
segment0	99.24	99.75	$\widetilde{W}_9$	99.87	98.53	100
ecoli2	91.17	94.51	$\widetilde{W}_9$	97.33	96.06	96.75
newthyroid1	98.24	99.72	$\widetilde{W}_9$	100	100	100
newthyroid2	95.55	100	$\widetilde{W}_7$	100	100	100
ecoli1	87.88	90.69	$\widetilde{W}_3$	93.03	89.22	91.7
glass-0-1-2-3_vs_4-5-6	90.67	94.68	$\widetilde{W}_7$	96.92	95.91	97.43
vehicle0	98.57	99.32	$\widetilde{W}_3$	99.61	99.67	99.21
adult	73.86	81.67	$\widetilde{W}_9$	88.72	84.57	86.47
vehicle3	78.15	85.13	$\widetilde{W}_6$	84.92	85.67	86.63
vehicle1	79.29	85.30	$\widetilde{W}_7$	87.7	85.35	86.86
vehicle2	98.43	99.12	$\widetilde{W}_4$	100	100	100
haberman	49.16	65.11	$\widetilde{W}_6$	64.95	67.91	64.72
yeast1	63.26	72.57	$\widetilde{W}_7$	76.27	73.76	75.39
Leukemia(Gene Sel)	100	100	$\widetilde{W}_6$	100	100	100
glass0	79.61	82.62	$\widetilde{W}_3$	91.24	85.66	89.59
Iris0	100	100	$\widetilde{W}_6$	100	100	100
pima	70.10	74.74	$\widetilde{W}_4$	85.25	79.16	83.22
wisconsin	96.32	97.07	$\widetilde{W}_5$	98.57	97.49	98.98

Table 1 continued

G-mean value (%)	Unweighted ELM	Weighted ELM $\boldsymbol{W_1}$ or $\boldsymbol{W_2}$	Optimal weighted ELM		VW-ELM	IW- ELM
	Testing result	Testing result	Optimal weight	Testing result	Testing result	Testing result
glass1	78.36	79.32	$\widetilde{\boldsymbol{W}}_2$	81.83	80.33	85.75
Colon(Gene Sel)	85.28	85.28	$\widetilde{\boldsymbol{W}}_9$	90.32	88.32	89.15
banana	88.98	89.13	$\widetilde{\boldsymbol{W}}_9$	92.34	91.34	94.66

Data sets are listed from low to high ratio

$\frac{1}{\#(t_1)+\{\#(t_2)-\#(t_1)\}\times\frac{1}{k-1}\times(i+\frac{k-1}{2})}$, $(i = 0, 1, \ldots, k-1)$. The proposed method is implemented in three major steps.

Step 1 k different weight matrices are first selected:

$$\tilde{\boldsymbol{W}}_i = \boldsymbol{W}_i^+ + \boldsymbol{W}_i^-, \tag{10}$$

where $\boldsymbol{W}_i^+ = \text{diag}\left(\underbrace{0, 0, \cdots, 0}_{\#(t_2)}, \underbrace{a, a, \cdots, a}_{\#(t_1)}\right)$, $\boldsymbol{W}_i^- = \text{diag}\left(\underbrace{b_i, b_i, \cdots, b_i}_{\#(t_2)}, \underbrace{0, 0, \cdots, 0}_{\#(t_1)}\right)$, $i = 0, 1, \ldots, k-1$.

Note that when i is equal to $\frac{k-1}{2}$, $\tilde{\boldsymbol{W}}_i$ becomes the weight matrix $\boldsymbol{W_1}$ in Eq. (9).

Step 2 k weighted ELM classifiers are trained corresponding to the selected k weight matrices in Step 1.

It is expected that some of the classifiers are not usable—they may induce significant false-negative errors or false-positive errors. In order to eliminate those unusable classifiers, for each classifier, the number of positive classification determined is recorded. Then $\lceil \frac{k}{4} \rceil$ classifiers corresponding to the biggest $\lceil \frac{k}{4} \rceil$ numbers are rejected since they are very likely biased towards the positive class ($\lceil\ \rceil$ is rounding up function). Similarly, based on the number of negative classification recorded, $\lceil \frac{k}{4} \rceil$ classifiers corresponding to the biggest $\lceil \frac{k}{4} \rceil$ negative classification numbers are rejected as well because of the high possibility of misclassifying too many positive samples.

Step 3 Upon removing $\lceil \frac{k}{2} \rceil$ unusable classifiers, an ensemble composed of remaining classifiers is generated and majority voting is used to classify samples.

Note that the proposed algorithm and voting based extreme learning machine (V-ELM) proposed by Cao et al. in [30] are significantly different even though both are voting based techniques. V-ELM is an algorithm targeted towards balanced datasets, in which classifiers are first trained with unweighted ELM algorithm then the results are averaged by majority voting on classifying samples. V-LEM needs to be modified for imbalanced data classification. One intuitive technique is to combine weighted ELM into V-ELM. We name this method VW-ELM, in which the results are averaged from weighted ELM classifiers (scheme $\boldsymbol{W_1}$ or $\boldsymbol{W_2}$). It is known that the classifiers in an effective ensemble need to be "good and different". That is to say, every single classifier in the ensemble should have a certain level accuracy for classification while maintaining some dissimilarity from others. Although the single weighted ELM classifier in VW-ELM possibly performs better, the similarity of some classifiers makes VW-ELM unsatisfactory.

4 Performance evaluation

In this section, the performance of the proposed IW-ELM is compared with unweighted ELM, weighted ELM $\boldsymbol{W_1}$, weighted ELM $\boldsymbol{W_2}$, and VW-ELM using fivefold cross-validation. The results of unweighted ELM are quoted from Table 3 and Table 4 in [12] directly. For weighted ELM $\boldsymbol{W_1}$ and weighted ELM $\boldsymbol{W_2}$, the one with better performance is chosen from Table 3 and Table 4 in [12]. All experiments are conducted in MATLAB (R2014, 64bit) on a Window 10 OS with Intel Core i7-2620M 2.70GHz GHZ CPU and 12 GB RAM.

4.1 Data specification

For comparison purposes, the proposed IW-ELM are evaluated using the same 46 binary classification datasets as listed in Table 1 in [12], with the attributes normalized into $[-1, 1]$ and various imbalance ratios ranging from 0.0078 to 0.8605. Note that the imbalance ratio (IR) of binary classes is defined as:

$$\text{IR} = \frac{\#(t_1)}{\#(t_2)}. \tag{11}$$

4.2 Parameter settings

For convenience, the proposed algorithm is tested on Sigmoid additive node with a Sigmoid additive activation function defined as $G(\boldsymbol{a}, b, \boldsymbol{x}) = \frac{1}{1+\exp(-\boldsymbol{a}\boldsymbol{x}+b)}$. There are two parameters to tune: the trade-off constant C and the number of hidden nodes L. A grid search of C on

Table 2 The values of trade-off constant C and the number of hidden nodes L corresponding to the results in Table 1

Data sets	Unweighted ELM	Weighted ELM W_1 or W_2	Optimal weighted ELM	VW-ELM	IW- ELM
abalone19	(2^{42},990)	(2^{6},150)	(2^{10},10)	(2^{6},150)	(2^{30},60)
yeast6	(2^{44},350)	(2^{14},90)	(2^{10},10)	(2^{34},20)	(2^{40},310)
yeast5	(2^{36},90)	(2^{30},100)	(2^{20},10)	(2^{30},100)	(2^{10},710)
yeast-1-2-8-9_vs_7	(2^{42},880)	(2^{42},20)	(2^{20},110)	(2^{42},20)	(2^{0},310)
yeast4	(2^{42},960)	(2^{6},20)	(2^{20},60)	(2^{6},20)	(2^{10},110)
shuttle-c2-vs-c4	(2^{40},20)	(2^{31},10)	(2^{0},10)	(2^{31},10)	(2^{10},10)
glass5	(2^{20},90)	(2^{10},40)	(2^{20},60)	(2^{10},110)	(2^{20},50)
yeast-2_vs_8	(2^{0},290)	(2^{8},60)	(2^{40},60)	(2^{1},960)	(2^{20},10)
yeast-1-4-5-8_vs_7	(2^{44},970)	(2^{10},120)	(2^{30},10)	(2^{10},120)	(2^{10},60)
glass-0-1-6_vs_5	(2^{18},660)	(2^{6},960)	(2^{50},60)	(2^{6},960)	(2^{20},110)
abalone9-18	(2^{40},150)	(2^{32},20)	(2^{30},60)	(2^{16},70)	(2^{0},360)
page-blocks1	(2^{10},440)	(2^{30},30)	(2^{30},30)	(2^{25},40)	(2^{35},30)
glass4	(2^{34},30)	(2^{10},140)	(2^{0},10)	(2^{12},120)	(2^{30},160)
ecoli4	(2^{22},60)	(2^{6},180)	(2^{10},60)	(2^{6},180)	(2^{32},70)
yeast-1_vs_7	(2^{40},960)	(2^{16},550)	(2^{30},10)	(2^{16},550)	(2^{10},10)
shuttle-c0-vs-c4	(2^{14},10)	(2^{38},10)	(2^{30},10)	(2^{38},10)	(2^{50},10)
glass2	(2^{28},110)	(2^{24},130)	(2^{50},110)	(2^{22},140)	(2^{10},60)
glass-0-1-6_vs_2	(2^{34},150)	(2^{14},380)	(2^{30},110)	(2^{14},380)	(2^{50},60)
vowel0	(2^{28},110)	(2^{50},120)	(2^{40},110)	(2^{28},110)	(2^{10},410)
yeast-0-5-6-7-9_vs_4	(2^{32},390)	(2^{-2},150)	(2^{10},10)	(2^{-2},150)	(2^{20},60)
yeast-2_vs_4	(2^{36},280)	(2^{26},940)	(2^{10},10)	(2^{26},940)	(2^{50},60)
page-blocks0	(2^{34},830)	(2^{24},820)	(2^{50},60)	(2^{22},750)	(2^{40},310)
ecoli3	(2^{44},70)	(2^{46},10)	(2^{20},110)	(2^{46},10)	(2^{12},100)
yeast3	(2^{40},100)	(2^{16},700)	(2^{40},110)	(2^{16},700)	(2^{20},40)
glass6	(2^{46},450)	(2^{26},190)	(2^{20},10)	(2^{44},30)	(2^{34},30)
segment0	(2^{8},720)	(2^{18},870)	(2^{20},60)	(2^{18},870)	(2^{10},460)
ecoli2	(2^{36},60)	(2^{30},20)	(2^{10},70)	(2^{28},40)	(2^{50},60)
newthyroid1	(2^{18},180)	(2^{20},20)	(2^{10},60)	(2^{18},30)	(2^{16},160)
newthyroid2	(2^{18},40)	(2^{12},480)	(2^{0},10)	(2^{16},290)	(2^{10},160)
ecoli1	(2^{16},140)	(2^{4},320)	(2^{0},160)	(2^{4},320)	(2^{10},10)
glass-0-1-2-3_vs_4-5-6	(2^{8},140)	(2^{10},290)	(2^{0},10)	(2^{10},290)	(2^{20},160)
vehicle0	(2^{10},460)	(2^{16},850)	(2^{20},310)	(2^{16},850)	(2^{22},470)
adult	(2^{10},580)	(2^{4},840)	(2^{8},500)	(2^{10},550)	(2^{16},800)
vehicle3	(2^{8},450)	(2^{14},710)	(2^{14},710)	(2^{14},710)	(2^{50},210)
vehicle1	(2^{8},570)	(2^{14},450)	(2^{30},110)	(2^{14},450)	(2^{10},410)
vehicle2	(2^{12},600)	(2^{16},800)	(2^{30},160)	(2^{16},800)	(2^{30},210)
haberman	(2^{44},910)	(2^{34},10)	(2^{24},10)	(2^{34},130)	(2^{40},10)
yeast1	(2^{44},300)	(2^{26},120)	(2^{10},60)	(2^{26},120)	(2^{20},60)
Leukemia (Gene Sel)	(2^{10},100)	(2^{10},100)	(2^{10},100)	(2^{20},80)	(2^{8},120)
glass0	(2^{14},950)	(2^{22},710)	(2^{10},110)	(2^{22},800)	(2^{26},170)
Iris0	(2^{-2},10)	(2^{2},10)	(2^{2},10)	(2^{2},15)	(2^{2},30)
pima	(2^{32},30)	(2^{14},20)	(2^{24},50)	(2^{18},40)	(2^{16},10)
wisconsin	(2^{34},50)	(2^{34},60)	(2^{20},60)	(2^{34},60)	(2^{2},440)
glass1	(2^{16},440)	(2^{22},240)	(2^{16},120)	(2^{22},900)	(2^{50},60)
Colon(Gene Sel)	(2^{-12},940)	(2^{-6},320)	(2^{-8},800)	(2^{-6},700)	(2^{-14},900)
banana	(2^{24},50)	(2^{34},50)	(2^{20},60)	(2^{28},100)	(2^{34},80)

$\{2^{-18}, 2^{-16}, \ldots, 2^{48}, 2^{50}\}$ and L on $\{10, 20, \ldots, 990, 1000\}$ is conducted in order to find the optimal result. The value of k in Eq. (10) is defined as 11, thus the number of remaining classifiers is 5. In order to make a fair comparison, the number of classifiers used in VW-ELM is also 5. C and L corresponding to the optimal result in our experiments are shown in Table 2.

4.3 Evaluation metrics for imbalanced datasets

Due to the possible presence of significant class imbalance, i.e., negative samples outnumber positive samples by a large margin, accuracy alone is not a reliable performance measure for imbalanced data classification. Other frequently used ones include F-measure, receiver operating characteristic (ROC) curve and geometric mean of the true rates (G-mean), etc. To compare the proposed method with weighted ELM presented in [12], G-mean is used in this paper. For binary classification problem, G-mean value is defined as the square root of (positive class accuracy × negative class accuracy):

$$\text{G-mean value} = \sqrt{\frac{TP}{TP+FN} \times \frac{TN}{TN+FP}}, \tag{12}$$

where TP, TN, FP, FN stands for true positive, true negative, false positive and false negative, respectively.

4.4 Experiment results

4.4.1 IW-ELM versus unweighted ELM

From Table 1, it can be seen that the proposed IW-ELM algorithm outperforms unweighted ELM. For problems with an imbalance ratio close to 1, i.e., the datasets are relatively balanced, weighted ELM does not have a significant advantage over unweighted ELM while IW-ELM still performs better than either one of them with higher G-mean values.

4.4.2 IW-ELM versus weighted ($\boldsymbol{W_1}$, $\boldsymbol{W_1}$)

As mentioned before, there are only 11 out of 46 datasets happen to have the optimal weights as $\boldsymbol{W_1}$ or $\boldsymbol{W_2}$. Even for these 11 datasets, the results of IW-ELM and weighted ELM are still comparable because of the ensemble used in IW-ELM.

4.4.3 IW-ELM versus VW-ELM

As seen in Table 1, VW-ELM generally performs better than weighted ELM because of the voting strategy; and IW-ELM has even higher successful classification rates over VW-ELM for most of the imbalanced problems tested. This is due to the fact that, in IW-ELM, unusable classifiers are eliminated from the ensemble and remaining classifiers are more different compared to the ones in VW-ELM.

5 Computational complexity

When using IW-ELM, there is a compromise between the high accuracy achieved and the computational complexity. Use ELM as a guideline, weighted-ELM has approximately the same computational complexity, while the complexities of VW-ELM and the proposed IW-ELM are higher due to the classifier training process. Since $\frac{k}{2}$ unweighted classifiers are trained in VW-ELM and k weighted classifies in IW-ELM, the execution time of IW-ELM is increased by k times compared to the original ELM.

6 Conclusions

In this paper, an improved algorithm designed for solving binary imbalanced classification problems—IW-ELM—has been proposed. There are three major steps in the process: training k weighted ELM classifiers, removing $\lceil \frac{k}{2} \rceil$ unusable classifiers, and determining the final result based on majority voting of remaining classifiers. The performance of IW-ELM (k set to be 11) is verified using 46 imbalanced datasets with various imbalance ratios ranging from 0.0078 to 0.8605. Simulation results demonstrated that IW-ELM achieves higher accuracy compared to other ELM based algorithms.

References

1. Monika I Trivedi, Mridush M (2016) Credit card fraud detection. Int J Adv Res Comput Commun Eng 5(1):39–50
2. Fan B, Lu X, Li HX (2016) Probabilistic inference-based least squares support vector machine for modeling under noisy environment. IEEE Trans Syst Man Cybern Syst 46(12):1703–1710
3. Wu DD, Zheng L, Olson DL (2014) A decision support approach for online stock forum sentiment analysis. IEEE Trans Syst Man Cybern Syst 44(8):1077–1087
4. Jiao L, Denoeux T, Pan Q (2016) A hybrid belief rule-based classification system based on uncertain training data and expert knowledge. IEEE Trans Syst Man Cybern Syst 46(12):1711–1723
5. Chawla NV, Bowyer KW, Hall LO et al (2002) SMOTE: synthetic minority over-sampling technique. J Mach Learn Res 16:321–357
6. Tomek I (1976) Two modifications of CNN. IEEE Trans Syst Man Cybern 6:769–772
7. Zquez F, Nchez JS, Pla F (2005) A stochastic approach to wilson's editing algorithm. In: Second Iberian conference on pattern recognition and image analysis volume part II, vol 3523, pp 35–42
8. Estabrooks A, Japkowicz N (2001) A mixture-of-experts framework for learning from imbalanced data sets. In: 4th international conference, IDA 2001: advances in intelligent data analysis, volume 2189 of the series. Lecture notes in computer science, pp 34–43

9. Maldonado S, Montecinos C (2014) Robust classification of imbalanced data using one-class and two-class SVM-based multi classifiers. Intell Data Anal 18(1):95–112
10. Huang Y (2015) Cost-sensitive incremental classification under the map reduce framework for mining imbalanced massive data streams. J Discrete Math Sci Cryptogr 18(1–2):177–194
11. López V, Fernández A, García S et al (2013) An insight into classification with imbalanced data: empirical results and current trends on using data intrinsic characteristics. Inf Sci 250:113–141
12. Zong W, Huang GB, Chen Y (2013) Weighted extreme learning machine for imbalance learning. Neurocomputing 101:229–242
13. Zhang WB, Ji HB (2013) Fuzzy extreme learning machine for classification. Electron Lett 49:448–450
14. Li K, Kong X, Lu Z, Liu W, Yin J (2014) Boosting weighted ELM for imbalanced learning. Neurocomputing 128:15–21
15. Lin SJ, Chang C, Hsu MF (2013) Multiple extreme learning machines for a two-class imbalance corporate life cycle prediction. Knowl Based Syst 39(3):214–223
16. Huang GB, Zhu QY, Siew CK (2006) Extreme learning machine: theory and applications. Neurocomputing 70(1–3):489–501
17. Zhang R, Lan Y, Huang GB, Xu ZB (2012) Universal approximation of extreme learning machine with adaptive growth of hidden nodes. IEEE Trans Neural Netw Learn Syst 23:365–371
18. Cao J, Hao J, Lai X et al (2016) Ensemble extreme learning machine and sparse representation classification. J Franklin Inst 353(17):4526–4541
19. Li X, Mao W, Jiang W et al (2016) Extreme learning machine via free sparse transfer representation optimization. Memetic Comput 8(2):85–95
20. Cheng X, Liu H, Xu X et al (2016) Denoising deep extreme learning machine for sparse representation. Memetic Comput. doi:10.1007/s12293-016-0185-2
21. Zhang N, Ding S (2016) Unsupervised and semi-supervised extreme learning machine with wavelet kernel for high dimensional data. Memetic Comput. doi:10.1007/s12293-016-0198-x
22. Lu H, Du B, Liu J et al (2017) A kernel extreme learning machine algorithm based on improved particle swam optimization. Memetic Comput 9(2):121–128
23. Jin Y, Cao J, Wang Y et al (2016) Ensemble based extreme learning machine for cross-modality face matching. Multimed Tools Appl 75(19):11831–11846
24. Cao J, Zhang K, Luo M (2016) Extreme learning machine and adaptive sparse representation for image classification. Neural Netw 81:91–102
25. Shrivastava NA, Panigrahi BK, Lim M-H (2016) Electricity price classification using extreme learning machines. Neural Comput Appl 27(1):9–18
26. Das SP, Padhy S (2016) Unsupervised extreme learning machine and support vector regression hybrid model for predicting energy commodity futures index. Memetic Comput. doi:10.1007/s12293-016-0191-4
27. Dwiyasa F, Lim M-H, Ong YS et al (2017) Extreme learning machine for indoor location fingerprinting. Multidimens Syst Signal Process 28:867. doi:10.1007/s11045-016-0409-0
28. Cao J, Luo X (2014) Protein sequence classification with improved extreme learning machine algorithms. Biomed Res Int 2014(1):103054
29. Dwiyasa F, Lim M-H (2015) Extreme learning machine for active RFID location classification. In: Proceedings of the 18th Asia pacific symposium on intelligent and evolutionary systems, volume 2 of the series proceedings in adaptation, learning and optimization, pp 657–670
30. Cao J, Lin Z, Huang GB et al (2012) Voting based extreme learning machine. Inf Sci 185:66–77