Information Sciences 563 (2021) 358–374

Contents lists available at ScienceDirect

Information Sciences

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

The improved AdaBoost algorithms for imbalanced data classification

Wenyang Wang [a,b,*], Dongchu Sun [c,d]

[a] School of Maritime Economics and Management, Dalian Maritime University, Dalian, Liaoning 116026, China
[b] Collaborative Innovation Center for Transport Studies, Dalian Maritime University, Dalian, Liaoning 116026, China
[c] Department of Statistics, University of Nebraska-Lincoln, Lincoln, Nebraska 68583, USA
[d] School of Statistics, Faculty of Economics and Management, East China Normal University, Shanghai 200241, China

ARTICLE INFO

Article history:
Received 22 June 2019
Received in revised form 16 September 2020
Accepted 19 March 2021
Available online 26 March 2021

Keywords:
AdaBoost
Imbalanced data problem
Weighted parameters adjustment

ABSTRACT

Class imbalance is one of the most popular and important issues in the domain of classification. The AdaBoost algorithm is an effective solution for classification, but it still needs improvement in the imbalanced data problem. This paper proposes a method to improve the AdaBoost algorithm using the new weighted vote parameters for the weak classifiers. Our proposed weighted vote parameters are determined not only by the global error rate but also by the classification accuracy rate of the positive class, which is our primary interest. The imbalanced index of the data is also a factor in constructing our algorithms. Our proposed algorithms outperform the traditional ones, especially regarding the evaluation criterion of $F-1$ *Measure*. Theoretic proofs of the advantages of our proposed algorithms are presented. Two kinds of simulated datasets and four real datasets are applied in the experiment as the specific support to our proposed algorithms.

1. Introduction

Class imbalance is one of the most severe and challenging problems in machine learning, especially in classification. The class imbalanced situation frequently appears in many fields such as facial detection [36,16], financial fraud detection [6,26,5], network intrusion detection [27], software defect prediction [4,30], and oil exploration [19,14,13].

A binary imbalanced data can be described as follows: Let $\boldsymbol{S}=\{(\boldsymbol{x}_1,y_1),(\boldsymbol{x}_2,y_2),\ldots,(\boldsymbol{x}_n,y_n)\}$ be the training data for a classification problem, where $\boldsymbol{x}_i\in\boldsymbol{X}\subseteq\boldsymbol{R}^l$, $\boldsymbol{X}$ is a subset of l-dimensional vector space, and the response $y_i\in\{-1,1\}$indicates two classes, $i=1,\ldots,n$. Define $\boldsymbol{S}^+=\{(\boldsymbol{x}_i,y_i)\in\boldsymbol{S}:y_i=1,i=1,\ldots,n\}$ is the positive or minority class and $\boldsymbol{S}^-=\{(\boldsymbol{x}_i,y_i)\in\boldsymbol{S}:y_i=-1,i=1,\ldots,n\}$ is the negative or majority class. The class types, {minority, majority} and {positive, negative} are used to describe $\{\boldsymbol{S}^+,\boldsymbol{S}^-\}$.

Definition 1. Let $|\boldsymbol{A}|$ denote the number of the elements in a set $\boldsymbol{A}$. Define $N_p=|\boldsymbol{S}^+|$ and $N_n=|\boldsymbol{S}^-|$ are the numbers of samples in the positive class and negative class, respectively. The imbalanced index of data is defined as $b=N_n/N_p$.

* Corresponding author.
E-mail addresses: ww424@mail.missouri.edu (W. Wang), dsun9@unl.edu (D. Sun).

https://doi.org/10.1016/j.ins.2021.03.042
0020-0255/

If $N_n > N_p$, this imbalanced condition in the different classes is called the imbalanced data problem. b is a positive number larger than 1 in imbalanced data. There is no explicit definition of the imbalanced data only based on b since other factors can affect the classification performance such as the dimensions of data, the overlap situation, and the total sample size [31]. Typically, when the accuracy rate of $\boldsymbol{S}^+$ is significantly lower than that of $\boldsymbol{S}^-$ because of the skewed structure of the data, we treat this as the imbalanced data problem. Six significant factors are known to cause an imbalanced data problem [25]:

(a) Overlap between classes;
(b) Borderline instances;
(c) Areas with small disjuncts;
(d) Noisy data;
(e) Low density and lack of information in the training data;
(f) Possible difference in the distributions of the training and the test data.

These factors are grouped into three principal issues[20]: (1) class overlap ((a) and (b)), (2) small disjuncts ((c) and (d)), and (3) data shift ((e) and (f)). Fig. 1 illustrates these three situations. Each of the sub-figures in Fig. 1 includes 30 black points of the negative class and 20 blue points of the positive class indicating $b = 1.5$ and the bias of the data is not severe. Overlap, as shown in Fig. 1 (1), involves two classes that are not completely separated especially when almost all the positive class points located on the borderline of the negative class. Fig. 1 (2) shows a small disjunct problem occurring when several small clusters of the positive points are located inside of the negative class region. As can be seen in Fig. 1 (3), the data shift means the distributions of the training and test data are different. Even when b is small such as 1.5 in Fig. 1, the classification accuracy of $\boldsymbol{S}^+$ would be degraded when class overlap, small disjuncts or data shift happens. Under the balanced situation, $N_n \approx N_p$, many algorithms are available for classification such as linear classifiers, Support Vector Machines, decision trees, random forest and so on [2]. For the imbalanced situation, although the aforementioned algorithms still can obtain an excellent global accuracy rate, the classification accuracy rate of $\boldsymbol{S}^+$ could be low. A worse result is that the classifier might identify the positive samples as noise and ignore them in the training process. If $\boldsymbol{S}^+$ is our particular interest, we need some other specific algorithms for the classification. Such as in a cancer diagnostic where the cancer cases are quite rare compared with the healthy populations. People desire an optimal classifier which gives the maximum accuracy rate of the patients' category without sacrificing the global accuracy rate much. Only the binary imbalanced class problem is studied in this paper. Multiple-class issues can be solved based on the one-versus-all scheme [29].

Data level and algorithm level methods are two typical approaches [35,18] to solve the imbalanced data problem. The former is a data pre-processing method [3,33], where resampling is utilized frequently. The basic idea of the data level method is to delete the instances in $\boldsymbol{S}^-$ or increase the instances in $\boldsymbol{S}^+$ to change the data sizes of the two classes and relieve the imbalanced situation before the training process. Although the data-level approach is simple, the random under-sampling or over-sampling makes the pre-processing dataset different from the raw data. Due to this limitation, the data level method is used less often. On the other hand, the algorithm level method focuses on training the classifier directly with the original data. The most common ones are Cost-Sensitive Learning [8] and Ensemble Schemes including boosting [34,7] and bagging [12]. Cost-Sensitive Learning assumes that the misclassified costs have prior information to enhance the impact of the minority class. The limitation is that the prior information regarding the costs is hard to find. Also, if the minority class samples are sparse, Cost-Sensitive Learning may not construct an appropriate decision boundary[24]. Fig. 2 summarizes the popular classifiers for balanced data and imbalanced data.

The AdaBoost (the abbreviation of Adaptive Boosting), formulated by Yoav Freund and Robert Schapire [10], is a popular boosting algorithm in Ensemble Schemes. It trains a series of weak classifiers iteratively based on the weighted data. The outputs of all weak classifiers are combined into a weighted summation that presents the final boosted classifier. A review of the AdaBoost algorithm is stated below.

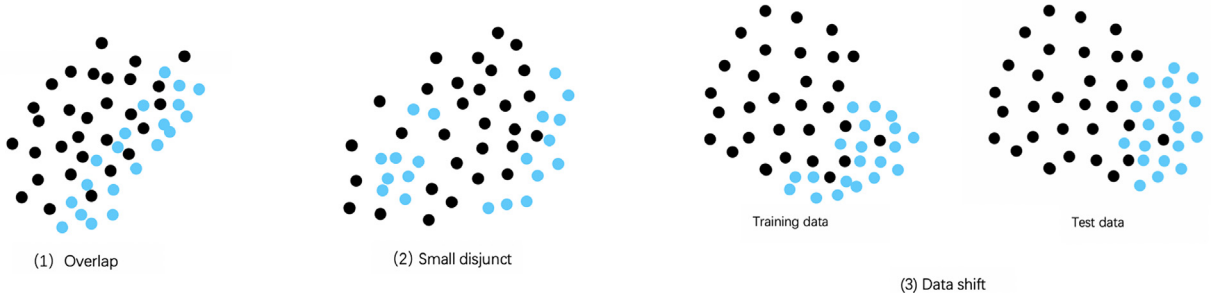

Fig. 1. Principal Issues Leading to Imbalanced Data Problem.

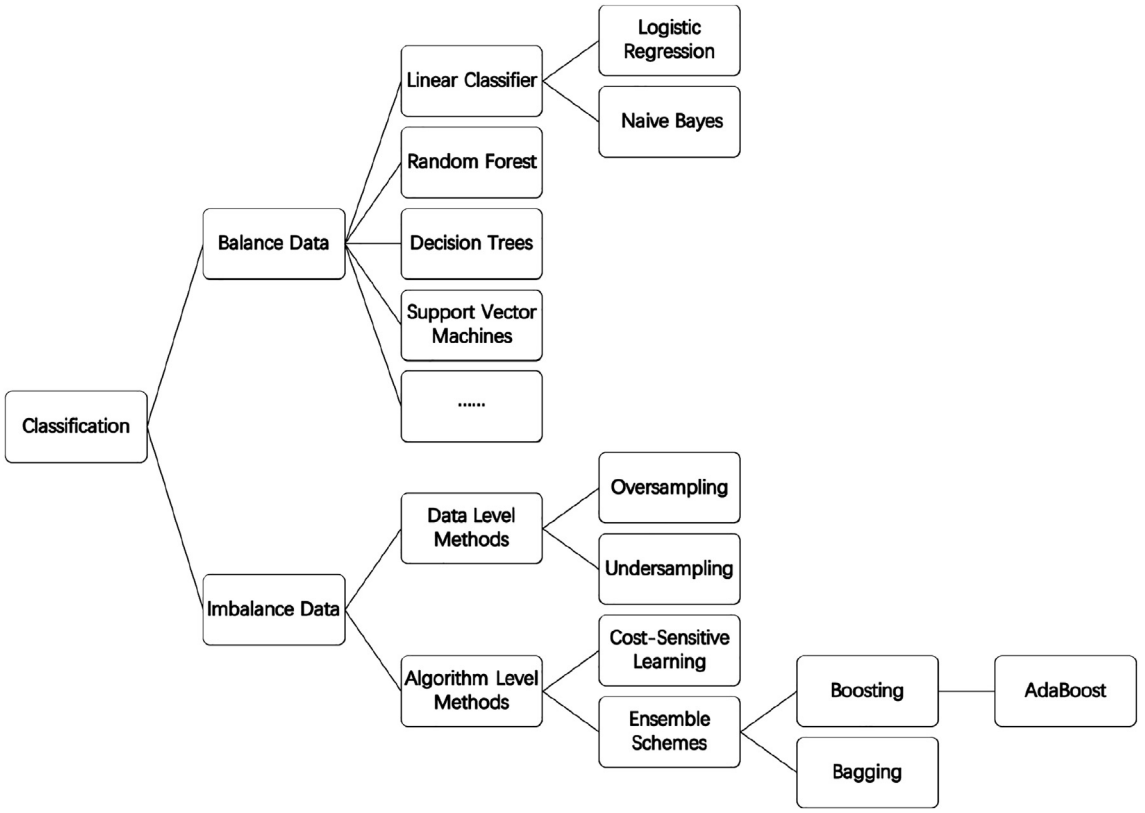

Fig. 2. Classifiers Summary.

Algorithm 1. The AdaBoost

Input. The training data: $(\boldsymbol{x}_1, y_1), (\boldsymbol{x}_2, y_2), \ldots, (\boldsymbol{x}_n, y_n), \boldsymbol{x}_i \in \boldsymbol{X} \subseteq \boldsymbol{R}^l$ and $y_i \in \{-1, 1\}, i = 1, \ldots, n$.Choose the number of weak classifiers T.

0.Let $t = 1$ and initialize the sample weight $D_t(i) = 1/n, i = 1, \ldots, n$;

1. Find the weak classifier $h_t(\boldsymbol{x}) \in \{-1, 1\}$ minimizing the weighted error rate $\epsilon_t = \sum_{i:h_t(\boldsymbol{x}_i) \neq y_i} D_t(i)$;

2. Calculate the weak classifier weight $\alpha_1^t = \frac{1}{2} log\{(1 - \epsilon_t)/\epsilon_t\}$;

3. Update $D_{t+1}(i) = D_t(i) exp\{-\alpha_1^t y_i h_t(\boldsymbol{x}_i)\}, i = 1, \ldots, n$, renormalize it as $D_{t+1}(i) = D_{t+1}(i)/\sum_{j=1}^{n} D_{t+1}(j)$;

4. If $t = T$, stop the iteration, else let $t = t + 1$ and return to Step **1**.

Output. The final strong classifier is $C_1(\boldsymbol{x}) = sign\{H_1(\boldsymbol{x}) - M_1\}$, where M_1 is the threshold, and $H_1(\boldsymbol{x}) = \sum_{t=1}^{T} \alpha_1^t h_t(\boldsymbol{x})$.

A weak classifier $h_t(\boldsymbol{x})$ presents a map from $\boldsymbol{X}$ to $\{-1, 1\}$. When $h_t^*(\boldsymbol{x})$ is any map from $\boldsymbol{X}$ to $\boldsymbol{R}$, we can use $h_t(\boldsymbol{x}) = sign(h_t^*(\boldsymbol{x}))$ to transfer $h_t^*(\boldsymbol{x})$ to a weak classifier. When $h_t^*(\boldsymbol{x}_i)$ is zero, randomly set $\boldsymbol{x}_i$ as the positive or negative class. α^t is the corresponding weighted vote parameter of the weak classifiers. Note that α_1^t in Step **2** is chosen based on the method of minimizing the exponential loss function [28], which implies that the more accurate a weak classifier is, the larger vote weight it has in constructing the final strong classifier. Step **3** presents the weight of the samples, $D_t(i)$. If Sample $(\boldsymbol{x}_i, y_i)$ is misclassified by $h_t(\boldsymbol{x})$, then $D_{t+1}(i) > D_t(i)$ in $h_{t+1}(\boldsymbol{x})$. On the other hand, if Sample $(\boldsymbol{x}_i, y_i)$ is classified correctly by $h_t(\boldsymbol{x})$, then $D_{t+1}(i) < D_t(i)$ in $h_{t+1}(\boldsymbol{x})$ [11]. So in AdaBoost, every weak classifier would get updated by focusing on the misclassified samples in the previous iteration to improve the global classification accuracy. By highlighting the misclassified samples and combining all the weak classifiers to build a strong one, AdaBoost is regarded as a popular and powerful algorithm in classification.

If $\boldsymbol{S}^+$ is our primary interest, the AdaBoost algorithm should be improved because it uses the same $D_t(i)$ for all samples in both $\boldsymbol{S}^+$ and $\boldsymbol{S}^-$. Also, it chooses the α_1^t only based on the global error rate ϵ_t. In recent years, several improved AdaBoost algorithms have been proposed for the class imbalanced problem. Typically, they include adjusting $D_t(i)$ or α_1^t. AdaCost [9] and

Cost-Sensitive AdaBoost [31,37,32] adjusted $D_t(i)$ by adding a higher misclassification costto the minority class. They assimilated the Cost-Sensitive Learning into the AdaBoost but these methods have a similar limitation as of the Cost-Sensitive Learning. Adjusting α_1^t is a mighty improvement without apparent shortcomings. AD AdaBoost [22] proposed a new α^t to improve the AdaBoost algorithm and received a convincing result in the object detection problem. In this paper, we first propose an Enhanced AdaBoost algorithm which is based on the AD AdaBoost but designed for the class imbalanced problem by adjusting α_1^t. When the imbalanced index b is large, the Enhanced AdaBoost can obtain an excellent classification result. However, b is not the only factor to define the imbalanced data. Recall Fig. 1, when b is small, we still could face an imbalanced problem. So we propose the Reinforced AdaBoost, which is another improved AdaBoost algorithm. The Reinforced AdaBoost can perform better when the imbalanced index b is relatively small but needs an additional iteration compared to the Enhanced AdaBoost. Similar to AD AdaBoost, our improved α^t includes not only the global error rate but also the classification accuracy rate of $\boldsymbol{S}^+$. Also, the imbalanced index b is considered in our proposed algorithms. The final classifier with our proposed weight parameters α^t can focus on $\boldsymbol{S}^+$ to increase the criterion of $F-1$ *Measure*, which is the evaluation criterion we are most interested in, without losing the global accuracy rate.

The remainder of this paper is organized as follows: In Section 2, we describe our proposed Enhanced AdaBoost algorithm. Section 3 gives the Reinforced AdaBoost, another improved AdaBoost designed for the imbalanced data problem with small b. Section 4 shows the applications of our proposed algorithms and compares them with several other traditional algorithms. Discussion and comments are given in Section 5.

2. The Enhanced AdaBoost Algorithm

2.1. The Enhanced AdaBoost

We follow the notations, $D_t(i)$ and ϵ_t in **Algorithm 1**, b in **Definition 1**.

Definition 2. Define $K_t = \sum_{i:y_i=1} D_t(i), P_t = \sum_{i:y_i=1, h_t(\boldsymbol{x}_i)=1} D_t(i)$, and $\gamma_t = P_t/K_t$.

For every $h_t(\boldsymbol{x}), K_t$ is the summation of weights of all samples in $\boldsymbol{S}^+, P_t$ means the summation of weights of the correctly classified samples in $\boldsymbol{S}^+, \gamma_t$ is a measure of the classification ability of $h_t(\boldsymbol{x})$ in $\boldsymbol{S}^+$. Note that $\gamma_t \in [0,1], b \in (1,\infty)$, and $\epsilon_t \in [0, 0.5)$ since it is assumed that all the classifiers have to be better than a random guess. The summation of sample weights under different conditions of y_i and $h_t(\boldsymbol{x}_i)$ are presented in Table 1. The global error rate ϵ_t is the summation of the false positive rate $A^t_{1,-1}$ and the false negative rate $A^t_{-1,1}, \epsilon_t = A^t_{1,-1} + A^t_{-1,1}$. $K_t = A^t_{1,\cdot}$ and $P_t = A^t_{1,1}$ are based on **Definition 2**. The proportion of $A^t_{1,1}$ in $A^t_{1,\cdot}$ is presented by $\gamma_t = P_t/K_t$. Note that in the t^{th} step of the iterations, the number of correctly classified samples in $\boldsymbol{S}^+$ is $N_p\gamma_t$. Now we propose the Enhanced AdaBoost with $D_t(i), \epsilon_t, \gamma_t$, and b defined above.

Algorithm 2. The Enhanced AdaBoost

Input. The training data: $(\boldsymbol{x}_1, y_1), (\boldsymbol{x}_2, y_2), \ldots, (\boldsymbol{x}_n, y_n), \boldsymbol{x}_i \in \boldsymbol{X} \subseteq \boldsymbol{R}^l$ and $y_i \in \{-1, 1\}, i = 1, \ldots, n$.Choose the number of weak classifiers T, the parameters β and k. $b = N_n/N_p$ is determined by thetraining data.

0. Let $t = 1$ and initialize the sample weight $D_t(i) = 1/n, i = 1, \ldots, n$;

1. Find the weak classifier $h_t(\boldsymbol{x}) \in \{-1, 1\}$ minimizing the weighted error rate $\epsilon_t = \sum_{i:h_t(\boldsymbol{x}_i)\neq y_i} D_t(i)$.

Let $\gamma_t = \frac{A_{1,1}}{A_{1,\cdot}}$, if $\gamma_t > \frac{1}{2}, \epsilon_t$should satisfy $\epsilon_t < \frac{1}{2}\{1 - (2\gamma_t - 1)/(b+1)\}$, else repeat this step;

2. Calculate the weak classifier weight

$$\alpha_2^t = \frac{1}{2} log\{(1-\epsilon_t)/\epsilon_t\} + k exp\{\beta(2\gamma_t - 1)\}, \tag{1}$$

where k and β are two parameters, $k > 0$;

3. Update $D_{t+1}(i) = D_t(i) exp\{-\alpha_1^t y_i h_t(\boldsymbol{x}_i)\}, i = 1, \ldots, n$, renormalize it as $D_{t+1}(i) = D_{t+1}(i)/\sum_{j=1}^n D_{t+1}(j)$;

4. If $t = T$, stop the iteration, else let $t = t + 1$ and return to Step **1**.

Output. The final strong classifier is $C_2(\boldsymbol{x}) = sign\{H_2(\boldsymbol{x}) - M_2\}$, where M_2 is the threshold, and $H_2(\boldsymbol{x}) = \sum_{t=1}^{T} \alpha_2^t h_t(\boldsymbol{x})$.

2.2. Properties of the Enhanced AdaBoost

In (1), α_2^t includes two important parameters ϵ_t and γ_t, which means that α_2^t considers not only the global error rate but also the classification ability of $\boldsymbol{S}^+$ [22]. In Step **1** of **Algorithm 2**, b plays an important role and constructs a restriction for the

Table 1
Summation of Sample Weights in Different Conditions of y_i and $h_t(\boldsymbol{x}_i)$.

	$h_t(\boldsymbol{x}_i) = 1$	$h_t(\boldsymbol{x}_i) = -1$	Row Total
$y_i = 1$	$A^t_{1,1} = \sum_{i:y_i=1, h_t(\boldsymbol{x}_i)=1} D_t(i)$	$A^t_{1,-1} = \sum_{i:y_i=1, h_t(\boldsymbol{x}_i)=-1} D_t(i)$	$A^t_{1,\cdot} = \sum_{i:y_i=1} D_t(i)$
$y_i = -1$	$A^t_{-1,1} = \sum_{i:y_i=-1, h_t(\boldsymbol{x}_i)=1} D_t(i)$	$A^t_{-1,-1} = \sum_{i:y_i=-1, h_t(\boldsymbol{x}_i)=-1} D_t(i)$	$A^t_{-1,\cdot} = \sum_{i:y_i=-1} D_t(i)$
Column Total	$A^t_{\cdot,1} = \sum_{i:h_t(\boldsymbol{x}_i)=1} D_t(i)$	$A^t_{\cdot,-1} = \sum_{i:h_t(\boldsymbol{x}_i)=-1} D_t(i)$	1

global error rate ϵ_t when $\gamma_t > \frac{1}{2}$. This restriction is weak when b is large and should be satisfied easily in a severe imbalanced data problem.

Fact 1. Define

$$H_a(\boldsymbol{x}) = k\sum_{t=1}^{T} h_t(\boldsymbol{x}) exp[\beta(2\gamma_t - 1)], \tag{2}$$

$H_1(\boldsymbol{x})$ and $H_2(\boldsymbol{x})$ in **Algorithm 1** and **2** satisfy $H_2(\boldsymbol{x}) = H_1(\boldsymbol{x}) + H_a(\boldsymbol{x})$.

Note that $C_2(\boldsymbol{x})$ in **Algorithm 2** is a monotone function of $H_2(\boldsymbol{x})$. Improving the performance of $C_2(\boldsymbol{x})$ is equivalent to adjusting $H_2(\boldsymbol{x})$. The idea is to improve the classification accuracy rate by making the value of $H_2(\boldsymbol{x})$ for $\boldsymbol{S}^+$ increase and the value of $H_2(\boldsymbol{x})$ for $\boldsymbol{S}^-$ decrease compared to $H_1(\boldsymbol{x})$, respectively. Fig. 3 is a frequency histogram of $H_1(\boldsymbol{X})$ and $H_2(\boldsymbol{X})$ based on the simulated Gaussian data part in the numeric study section with $b = 10, \beta = 0.5$, and $k = 0.0009$. In Fig. 3, $H_2(\boldsymbol{x})$ of $\boldsymbol{S}^+$ and $\boldsymbol{S}^-$ moves further away than the $H_1(\boldsymbol{x})$ does, which means the Enhanced AdaBoost would result in more accurate classification.

Fact 2. In (2), $H_a(\boldsymbol{x})$ satisfies

$$\sum_{i:y_i=1} H_a(\boldsymbol{x}_i) = N_p k \sum_{t=1}^{T} (2\gamma_t - 1)\, exp\{\beta(2\gamma_t - 1)\},$$

$$\sum_{i:y_i=-1} H_a(\boldsymbol{x}_i) = N_p k \sum_{t=1}^{T} \{(2\gamma_t - 1) + (2\epsilon_t - 1)(b+1)\} exp\{\beta(2\gamma_t - 1)\}.$$

Note that for $\boldsymbol{S}^+$ and $\boldsymbol{S}^-$, we have

$$\begin{aligned}\sum_{i:y_i=1} h_t(\boldsymbol{x}_i) &= N_p\gamma_t \cdot 1 + N_p(1-\gamma_t)\cdot(-1)\\ &= N_p(2\gamma_t - 1),\end{aligned}$$

$$\begin{aligned}\sum_{i:y_i=-1} h_t(\boldsymbol{x}_i) &= \{N_n - [(N_n + N_p)\epsilon_t - (N_p - N_p\gamma_t)]\}\cdot(-1) + [(N_p + N_n)\epsilon_t - (N_p - N_p\gamma_t)]\cdot 1\\ &= 2\{(N_p + N_n)\epsilon_t - (N_p - N_p\gamma_t)\} - N_n.\end{aligned}$$

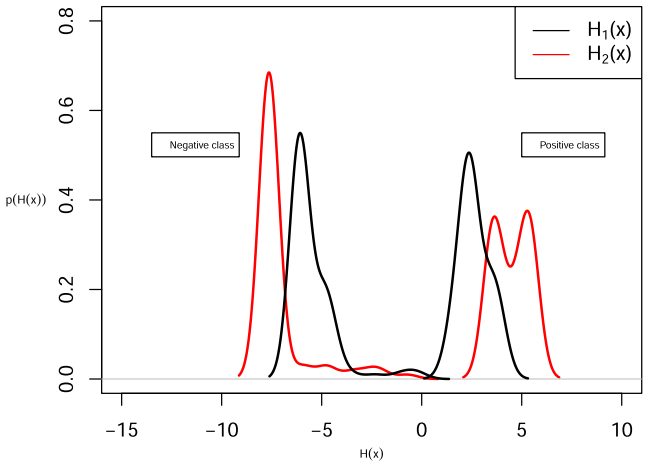

Fig. 3. Distribution of H(x) in **Algorithm 1** and **2**.

Fact 2 follows from **Fact 1**. We consider three situations of γ_t : (a) $\gamma_t \in [0, \frac{1}{2})$, (b) $\gamma_t = \frac{1}{2}$, and (c) $\gamma_t \in (\frac{1}{2}, 1]$. It is often that γ_t is larger than 0.5 but in some steps, especially in the last few steps of iterations, γ_t might be smaller than 0.5 if the distribution of $\boldsymbol{S}^+$ is sparse or the small disjunct situation is serious.

Theorem 1. *(a) For* $\gamma_t \in [0, \frac{1}{2})$,

$$\sum_{i:y_i=-1} H_a(\boldsymbol{x}_i) < \sum_{i:y_i=1} H_a(\boldsymbol{x}_i) < 0, \tag{3}$$

(b) For $\gamma_t = \frac{1}{2}$,

$$\sum\nolimits_{i:y_i=-1} H_a(\boldsymbol{x}_i) < \sum\nolimits_{i:y_i=1} H_a(\boldsymbol{x}_i) = 0, \tag{4}$$

(c) For $\gamma_t \in (\frac{1}{2}, 1]$, *assume that*

$$\epsilon_t < 0.5(1 - \frac{2\gamma_t - 1}{b+1}) < 0.5, \tag{5}$$

we have

$$\sum\nolimits_{i:y_i=-1} H_a(\boldsymbol{x}_i) < 0 < \sum\nolimits_{i:y_i=1} H_a(\boldsymbol{x}_i). \tag{6}$$

Proof. See Appendix, A1. □

In all three situations of **Theorem 1**, $\sum_{i:y_i=-1} H_a(\boldsymbol{x}_i) < \sum_{i:y_i=1} H_a(\boldsymbol{x}_i)$ always holds to improve the ability of distinguishing $\boldsymbol{S}^+$ and $\boldsymbol{S}^-$, but *(c)* does a better job than *(a)* and *(b)*. In *(c)*, $H_2(\boldsymbol{x})$ of $\boldsymbol{S}^+$ and $\boldsymbol{S}^-$ move toward opposite directions compared to $H_1(\boldsymbol{x})$ because of adding $H_a(\boldsymbol{x})$. However, in *(a)* and *(b)*, $H_2(\boldsymbol{x})$ moves to the negative direction compared to $H_1(\boldsymbol{x}), H_2(\boldsymbol{x})$ of $\boldsymbol{S}^-$ has a larger movement than that of $\boldsymbol{S}^+$. Fortunately, *(c)* happens more frequently in the training process. Combine all $h_t(\boldsymbol{x})$ with α_2^t as the weights to construct the final classifier $C_2(\boldsymbol{x})$. $\boldsymbol{S}^+$ and $\boldsymbol{S}^-$ would be separated toward the opposite directions. Fig. 3 gives a refined interpretation of this. It indicates that $H_2(\boldsymbol{x})$ has a further movement than $H_1(\boldsymbol{x})$ in the right direction. The two curves of $H_1(\boldsymbol{x})$ intersect around zero. If we choose the intersection point as the threshold and the curves beyond the threshold point in each class cause the misclassification. But the two curves of $H_2(\boldsymbol{x})$ are separated perfectly. Choose any point between the gap of the two curves as the threshold, we could obtain a 100% accurate classifier.

2.3. Choices of β and k

The Enhanced AdaBoost requires pre-specification of k and $\beta, k > 0$. $H_a(\boldsymbol{x})$ consisting of k and β contributes to $H_2(\boldsymbol{x})$ to improve the classification performance compared to $H_1(\boldsymbol{x})$. The parameters k and β are important to ensure that the Enhanced AdaBoost can not only increase the classification accuracy rate for $\boldsymbol{S}^+$ but also keep the global error rate low.

In the Enhanced AdaBoost, we need α_2^t to be an increasing function of γ_t. So that under the same ϵ_t, the weak classifier will have a larger vote weight if it has a stronger classification ability for $\boldsymbol{S}^+$. Express this condition as

$$\frac{\partial}{\partial \gamma_t} \alpha_2^t > 0, \tag{7}$$

which implies

$$\beta > 0. \tag{8}$$

We generally choose $\beta = 0.5$.

The parameter k is a positive parameter, which decides the adjusting part in α_2^t. A large k can make the stretching effect more obvious. But the classification ability of $\boldsymbol{S}^-$ would decrease by a lot if we increase k unadvisedly. A larger k could make the first term of (1) less important. So it is important to determine a proper value of k. Define $Z_t = \sum_{j=1}^n D_{t+1}(j), t = 1, \ldots, T-1$. Schapire [29] proved that for AdaBoost, the global training error satisfies:

$$\frac{1}{n} |\{i : H(\boldsymbol{x}_i) \neq y_i\}| \leqslant \prod_{t=1}^{T-1} Z_t, \tag{9}$$

where $|\cdot|$ means the number of elements in the set, $i = 1, .., n$.

For $t = 1, \ldots, T-1$, the upper boundary of the global training error should decrease to receive a more accurate final classifier when a new weak classifier is added, which means that we need $Z_t < 1$ [22]. In the Enhanced AdaBoost, Z_t is [15]:

$$Z_t = (1 - \epsilon_t) exp\{-\alpha_2^t\} + \epsilon_t exp\{\alpha_2^t\}. \tag{10}$$

By solving the inequality $Z_t < 1$,

$$0 < k < \frac{0.5\,log\left(\frac{1-\epsilon_t}{\epsilon_t}\right)}{exp\{\beta(2\gamma_t - 1)\}}. \tag{11}$$

The upper boundary of k in (11) is a decreasing function of ϵ_t. Recall (5), in the most frequent case, we rewrite (11) as:

$$0 < k \leqslant \frac{0.5\,log\left(\frac{1+\frac{2\gamma_t-1}{b+1}}{1-\frac{2\gamma_t-1}{b+1}}\right)}{exp\{\beta(2\gamma_t - 1)\}} < \frac{0.5\,log\left(\frac{1-\epsilon_t}{\epsilon_t}\right)}{exp\{\beta(2\gamma_t - 1)\}},$$

where $\gamma_t \in (\frac{1}{2}, 1]$.

We recommend to set $\beta = 0.5$. b is specified when the training data is given. We only consider $\gamma_t \in (\frac{1}{2}, 1]$ for computational simplicity. Given $\beta = 0.5, b = (2, 5, 10, 20, 50, 100)$, and $\gamma_t = (0.505, 0.6, 0.7, 0.8, 0.9, 1.0)$, the upper boundaries of k can be obtained in Table 2. Here we keep all the numbers rounded to the lowest value with four decimal places. For example, in the first part of the simulated Gaussian data study, $b = 10$. We choose $\beta = 0.5, k = 0.0009$ based on Table 2 for the new weak classifier weights α_2^t. This choice can ensure (9) holds and the global error rate remains low in our proposed algorithm.

3. The Reinforced AdaBoost Algorithm

The Enhanced AdaBoost has the restriction of ϵ_t in (5) including b and γ_t. The restriction would be weak when the value of b is large so that Step **1** in **Algorithm 2** does not need to be repeated. But b is not the only factor to define imbalanced data. Assume we have a dataset with $b = 2$. In some particular training iteration of the Enhanced AdaBoost, $\gamma_t = 0.99$, the restriction of ϵ_t in (5) is $0 \leqslant \epsilon_t < 0.3367$, which is a small subinterval of $0 \leqslant \epsilon_t < 0.5$. This situation often happens for a small b and Step **1** in **Algorithm 2** would be repeated to seek for a stronger $h_t(\boldsymbol{x})$ which satisfies the restriction. The worst situation is that under small b, there is no such $h_t(\boldsymbol{x})$ minimizing ϵ_t and satisfying the restriction of ϵ_t in (5). So we need to propose a new α^t without any restriction to obtain the same further separation of $H_t(\boldsymbol{x})$ for $\boldsymbol{S}^+$ and $\boldsymbol{S}^-$ as α_2^t does. Note that if $0.5(1 - \frac{2\gamma_t-1}{b+1}) \leqslant \epsilon_t < 0.5, \gamma_t$ has to be in the interval of $(0.5, 1]$. Now we propose another improved AdaBoost algorithm by considering a new modified parameter α_3^t called the Reinforced AdaBoost.

Algorithm 3. The Reinforced AdaBoost

Input. The training data: $(\boldsymbol{x}_1, y_1), (\boldsymbol{x}_2, y_2), \ldots, (\boldsymbol{x}_n, y_n), \boldsymbol{x}_i \in \boldsymbol{X} \subseteq \boldsymbol{R}^l$ and $y_i \in \{-1, 1\}, i = 1, \ldots, n$. Choose the number of weak classifiers T, the parameters β and k. $b = N_n/N_p$ is determined by the training data.

0. Let $t = 1$ and initialize the sample weight $D_t(i) = 1/n, i = 1, \ldots, n$;

1. Find the weak classifier $h_t \in \{-1, 1\}$ minimizing the weighted error rate $\epsilon_t = \sum_{i:h_t(\boldsymbol{x}_i)\neq y_i} D_t(i)$, let

$$\gamma_t = \frac{A_{1,1}^t}{A_{1,\cdot}^t};$$

2.1. When (1) $0 \leqslant \gamma_t \leqslant \frac{1}{2}$ or (2) $\frac{1}{2} < \gamma_t \leqslant 1$ and $\epsilon_t < \frac{1}{2}\{1 - (2\gamma_t - 1)/(b + 1)\}$, calculate the weak classifier weight

$$\alpha^t = \alpha_2^t = \frac{1}{2} log\left(\frac{1-\epsilon_t}{\epsilon_t}\right) + k\,exp\{\beta(2\gamma_t - 1)\};$$

2.2. When $\frac{1}{2} < \gamma_t \leqslant 1$ and $\frac{1}{2}\{1 - (2\gamma_t - 1)/(b + 1)\} \leqslant \epsilon_t$, calculate the weak classifier weight

$$\alpha^t = \alpha_3^t = \frac{1}{2} log\left(\frac{1-\epsilon_t}{\epsilon_t}\right)\left\{\frac{exp(\gamma_t - 0.5) + 0.5}{0.5 - \epsilon_t} + \frac{0.5(b+1)}{\gamma_t - 0.5}\right\}; \tag{12}$$

3. Update $D_{t+1}(i) = D_t(i) exp\{-\alpha_1^t y_i h_t(\boldsymbol{x}_i)\}, i = 1, \ldots, n$, renormalize it as $D_{t+1}(i) = D_{t+1}(i)/\sum_{j=1}^n D_{t+1}(j)$;

4. If $t = T$, stop the iteration, else let $t = t + 1$ and return to Step **1**.

Output. The final strong classifier is $C_3(\boldsymbol{x}) = sign\{H_3(\boldsymbol{x}) - M_3\}$, where $H_3(\boldsymbol{x}) = \sum_{t=1}^T \alpha^t h_t(\boldsymbol{x}), M_3$ is the threshold.

Here $A_{1,1}^t$ and $A_{1,\cdot}^t$ follow from Table 1. Define $H_3'(\boldsymbol{x}) = \sum_{t=1}^{T'} \alpha_3^t h_t(\boldsymbol{x}), T'$ is the number of weak classifiers that have the weighted vote parameters of α_3^t. We can obtain **Fact 3** based on **Fact 1** and **2**.

Table 2
Upper Boundary of k.

$\gamma_t \backslash b$	2	5	10	20	50	100
0.505	0.0033	0.0016	0.0009	0.0004	0.0001	0.0001
0.6	0.0604	0.0301	0.0164	0.0086	0.0035	0.0017
0.7	0.1098	0.0546	0.0297	0.0155	0.0064	0.0032
0.8	0.1501	0.0743	0.0404	0.0211	0.0087	0.0044
0.9	0.1831	0.0899	0.0488	0.0255	0.0105	0.0053
1.0	0.2102	0.1020	0.0552	0.0289	0.0118	0.0060

Fact 3. We have the equations:

$$\begin{aligned}
\textstyle\sum_{i:y_i=1} H_1(\boldsymbol{x}_i) &= (\gamma_t - 0.5)\, log\left(\tfrac{1-\epsilon_t}{\epsilon_t}\right) N_p,\\
\textstyle\sum_{i:y_i=-1} H_1(\boldsymbol{x}_i) &= \{(\gamma_t - 0.5) + (\epsilon_t - 0.5)(b+1)\}\, log\left(\tfrac{1-\epsilon_t}{\epsilon_t}\right) N_p,\\
\textstyle\sum_{i:y_i=1} H_3'(\boldsymbol{x}_i) &= (\gamma_t - 0.5)\left\{\tfrac{exp(\gamma_t-0.5)+0.5}{0.5-\epsilon_t} + \tfrac{0.5(b+1)}{\gamma_t-0.5}\right\} log\left(\tfrac{1-\epsilon_t}{\epsilon_t}\right) N_p,\\
\textstyle\sum_{i:y_i=-1} H_3'(\boldsymbol{x}_i) &= \{(\gamma_t - 0.5) + (\epsilon_t - 0.5)(b+1)\}\left\{\tfrac{exp(\gamma_t-0.5)+0.5}{0.5-\epsilon_t} + \tfrac{0.5(b+1)}{\gamma_t-0.5}\right\} log\left(\tfrac{1-\epsilon_t}{\epsilon_t}\right) N_p.
\end{aligned}$$

Theorem 2. *Assume that*

$$\gamma_t \in (\frac{1}{2}, 1] \text{ and } 0.5(1 - \frac{2\gamma_t - 1}{b+1}) \leqslant \epsilon_t < 0.5 \tag{13}$$

we have

$$\textstyle\sum_{i:y_i=1} H_3'(\boldsymbol{x}_i) > \sum_{i:y_i=1} H_1(\boldsymbol{x}_i), \tag{14}$$

$$\textstyle\sum_{i:y_i=-1} H_3'(\boldsymbol{x}_i) \leqslant \sum_{i:y_i=-1} H_1(\boldsymbol{x}_i). \tag{15}$$

Proof. See Appendix, A2. □

Theorem 2 indicates that under $\frac{1}{2}\{1 - (2\gamma_t - 1)/(b+1)\} \leqslant \epsilon_t$ with our proposed $\alpha_3^t, H(\boldsymbol{x})$ of $\boldsymbol{S}^+$ and $\boldsymbol{S}^-$ still can obtain a further separation in positive and negative directions than the AdaBoost does, respectively. (14) is a strict inequation, which means the positive class can obtain a strict increase in the classification result. The Reinforced AdaBoost eliminates the restriction of global error in the Enhanced AdaBoost and is a comprehensive solution to improve the AdaBoost algorithm for the imbalanced data problem. But the additional iteration in the definition of α^t in Step **2** of the Reinforced AdaBoost makes it more complex compared with the Enhanced AdaBoost. So when b is small, the Reinforced AdaBoost is the preferential choice. Otherwise, the Enhanced AdaBoost is powerful enough for the imbalanced data classification.

4. Numerical studies

In this performance evaluation, we apply several algorithms including our two proposed ones, the Enhanced AdaBoost and the Reinforced AdaBoost, into two kinds of simulated datasets and four real datasets. Because the goal of our paper is to improve the AdaBoost algorithm, we choose three kinds of original AdaBoost algorithms with different weak classifiers as the comparisons. On the other hand, Support Vector Machines (SVMs) is a popular classifier. Especially, SVMs with Radial Basis Function (RBF) kernel has the congenital advantage in Gaussian data classification. The RBF kernel is defined as

$$K_{RBF}(\boldsymbol{x}, \boldsymbol{x}') = exp(-\gamma||\boldsymbol{x} - \boldsymbol{x}'||^2),$$

where γ defines how far the influence of a single training example reaches. So SVMs with RBF kernel is under our consideration. Table 3 summarizes all the algorithms we use. Evaluation criteria play a crucial role in assessing the performance of a classifier. Before we give the evaluation criteria, the definition of the Confusion Matrix for the binary classifier is needed. A Confusion Matrix [17] contains the basic information about actual and predicted classification done by a classifier. The common evaluation criteria based on the Confusion Matrix are:(see Table 4)

- *Global Accuracy* $= TP + TN$;
- *Global Error Rate* $= 1 - Global\ Accuracy = 1 - (TP + TN)$;
- *Sensitivity* $= Recall = TP/P$;
- *Specificity* $= TN/N$;
- *Precision* $= TP/P'$.

Table 3
Algorithms in Study.

Name	Description	Parameters
SVMs	Standard single Support Vector Machines with RBF kernel	γ_{best} is obtained by grid searching in $2^{[-20:20]}$
Ada-DT	AdaBoost using Decision Tree as weak classifiers	
Ada-LSVMs	AdaBoost using linear kernel SVMs as weak classifiers	
Ada-RSVMs	AdaBoost using RBF kernel SVMs as weak classifiers	γ_{best} is obtained by grid searching in $2^{[-20:20]}$
En-Ada	Enhanced AdaBoost using Decision Tree as weak classifiers	
Re-Ada	Reinforced AdaBoost using Decision Tree as weak classifiers	

Table 4
Confusion Matrix.

		Prediction		
		Positive	Negative	Row Total
Truth	Positive	*TP* = True Positive rate	*FN* = False Negative rate	*P*
	Negative	*FP* = False Positive rate	*TN* = True Negative rate	*N*
	Column Total	P'	N'	1

In the imbalanced data problem, our goal is to keep the global error low and improve the classification accuracy rate of $\boldsymbol{S}^+$. *Sensitivity* (*Recall*) and *Precision* are more important for accessing the classification performance of $\boldsymbol{S}^+$. $F-1$ *Measure* [21] integrates *Sensitivity* and *Precision* as an average, defined as:

- $F-1$ *Measure* $= \frac{2\times Precision\times Sensitivity}{Precision+Sensitivity}$.

$F-1$ *Measure* represents a harmonic mean between *Sensitivity* and *Precision*. The harmonic mean of two numbers is closer to the small one. So a high $F-1$ *Measure* can ensure *Sensitivity* and *Precision* are both high.

The Receiver Operating Characteristic (ROC) curve is created by plotting the true positive rate (TP) on the Y-axis against the false positive rate (FP) on the X-axis at various threshold settings. The area under the curve (*AUC*) is equal to the probability that a classifier will rank a randomly chosen positive instance higher than a randomly chosen negative one. *AUC* provides a single measure of a classifier's performance based on the ROC curve. Large *AUC* means the classifier performs well.

We choose five from all seven criteria above. *Global Accuracy* and *AUC* are used to evaluate the global classification ability. *Sensitivity* and $F-1$ *Measure* are used as the evaluation criteria to compare the classification ability for $\boldsymbol{S}^+$ in different algorithms. *Specificity* is used to assess the classification ability of $\boldsymbol{S}^-$. All the results of the criteria in this section are kept rounded to four decimal places.

4.1. Simulation study based on gaussian data

The simulated training dataset is:

$$\boldsymbol{X}_{ij} = \begin{pmatrix} X_{ij1} \\ X_{ij2} \end{pmatrix} \overset{i.i.d.}{\sim} \boldsymbol{N}_2(\boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i), \tag{16}$$

where $i = -1, 1$ and $j = 1, \ldots, n_i$.

$$\boldsymbol{\mu}_{-1} = \begin{pmatrix} 7 \\ 8 \end{pmatrix}, \boldsymbol{\mu}_1 = \begin{pmatrix} 13 \\ 15 \end{pmatrix}, \boldsymbol{\Sigma}_{-1} = \begin{pmatrix} 10 & 3 \\ 3 & 8 \end{pmatrix}, \boldsymbol{\Sigma}_1 = \begin{pmatrix} 1 & 0 \\ 0 & 2 \end{pmatrix}. \tag{17}$$

All the variables in $\boldsymbol{X}_{-1j}$ and $\boldsymbol{X}_{1j}$ have the class labels -1 (Majority) and 1 (Minority), respectively. We consider two cases based on $b = 10$ and $b = 5$.

4.1.1. $b = 10$

We set the training sample sizes of the positive and negative class to be $(n_{-1}, n_1) = (500, 50)$. The test dataset has the same distributions as the training data, but the test data size is $(n^*_{-1}, n^*_1) = (100, 10)$. The scatter plots of this simulated data are given in Fig. 4.

For each simulation of the data, we conduct all the algorithms in Table 3 to compare the criteria of *Sensitivity*, *Specificity*, $F-1$ *Measure*, *AUC* and *Global Accuracy*. We repeat the experiments 100 times for every algorithm to reduce the randomness impact of the data simulation. Table 5 displays the mean values and standard deviation values (shown in brackets) of 100 repeated results. The optimum results are labeled in bold from Tables 5–14.

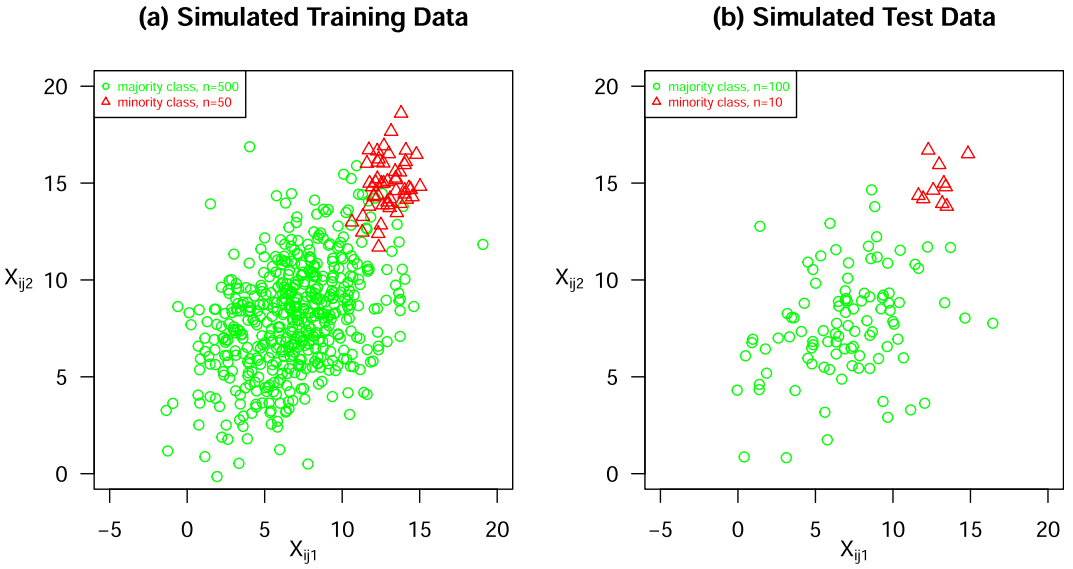

Fig. 4. Simulated Gaussian Data with $b = 10$.

Table 5
The Classification Results for Simulated Gaussian Data ($b = 10$).

Algorithms	Evaluation Measures				
	Sensitivity	Specificity	F-1 Measure	AUC	Global Accuracy
SVMs	0.8931 (0.1051)	**0.9903 (0.0109)**	0.8968 (0.0761)	0.9417 (0.0527)	**0.9815 (0.0135)**
Ada-DT	0.8485 (0.1188)	0.9873 (0.0116)	0.8575 (0.0890)	0.9717 (0.0599)	0.9747 (0.0154)
Ada-LSVMs	0.8778 (0.1145)	0.9739 (0.0210)	0.8233 (0.0524)	0.9259 (0.0497)	0.9651 (0.0129)
Ada-RSVMs	0.9822 (0.1024)	0.9693 (0.0092)	0.8636 (0.0352)	**0.9801 (0.0082)**	0.9712 (0.0085)
En-Ada	0.9833 (0.0157)	0.8834 (0.0779)	0.9153 (0.0900)	0.9259 (0.0541)	0.9185 (0.0376)
Re-Ada	**0.9840 (0.0574)**	0.8902 (0.0685)	**0.9159 (0.0767)**	0.9262 (0.0576)	0.9213 (0.0568)

The Reinforced AdaBoost has the largest $F-1$ *Measure* and *Sensitivity*. Notably, $F-1$ *Measure* obtains an obvious improvement in both of our proposed algorithms compared to others. So our proposed algorithms outperform others regarding the classification performance of the positive class.

The SVMs with RBF kernel has the largest *Specificity* and *Global Accuracy* and the AdaBoost with RBF kernel SVMs has the largest *AUC*. The phenomenon of SVMs performing well makes sense since as a reliable classifier, SVMs can obtain an excellent result for the classification if we are interested in the global accuracy. Also, the data we use is Gaussian data, all the algorithms related to SVMs with RBF kernel should perform well instinctively. Our proposed algorithms are both AdaBoost with the decision tree as the weak classifiers and they show better classification results in the positive class also obtain persuasive results from the global perspective.

The Reinforced AdaBoost has a similar result to the Enhanced AdaBoost. The reason is that $b = 10$ is a large value and the theoretical improvement in the Reinforced AdaBoost is not obvious compared to the Enhanced AdaBoost. We use the same simulated data but change $b = 5$ and repeat the experiment in the next case.

4.1.2. $b = 5$

Now the training data and test data are kept in the same distribution as the previous case but $(n_{-1}, n_1) = (500, 100)$ for the training data, and $(n^*_{-1}, n^*_1) = (100, 20)$ for the test data. The compared algorithms and criteria are the same as the last experiment. The results of 100 repeated experiments for the situation of $b = 5$ are shown in Table 6.

All the results in Table 6 are similar to the results in Table 5. But the criteria differences between the Reinforced AdaBoost and Enhanced AdaBoost are more obvious in this study. This result supports the claim that the Reinforced AdaBoost outperforms the Enhanced AdaBoost when b is relatively small.

4.2. Simulation study based on uniform data

The simulated training dataset is:

Table 6
The Classification Results for Simulated Gaussian Data ($b = 5$).

Algorithms	Evaluation Measures				
	Sensitivity	Specificity	F-1 Measure	AUC	Global Accuracy
SVMs	0.9510 (0.0447)	**0.9824 (0.0131)**	0.9232 (0.0364)	0.9667 (0.0227)	**0.9771 (0.0127)**
Ada-DT	0.9005 (0.0695)	0.9805 (0.0141)	0.9011 (0.0507)	0.9405 (0.0356)	0.9672 (0.0167)
Ada-LSVMs	0.9663 (0.0425)	0.9621 (0.0267)	0.9003 (0.0520)	0.9642 (0.0291)	0.9628 (0.0207)
Ada-RSVMs	0.9736 (0.0168)	0.9672 (0.0181)	0.9226 (0.0396)	**0.9804 (0.0127)**	0.9716 (0.0155)
En-Ada	0.9810 (0.0534)	0.9111 (0.0567)	0.9341 (0.0213)	0.9402 (0.0867)	0.9402 (0.0789)
Re-Ada	**0.9830 (0.0123)**	0.9154 (0.0345)	**0.9401 (0.0364)**	0.9451 (0.0234)	0.9433 (0.0364)

$$\boldsymbol{X}_{ij} = \begin{pmatrix} X_{ij1} \\ X_{ij2} \end{pmatrix} \overset{i.i.d.}{\sim} \textbf{Uniform}_2(\boldsymbol{a}_i, \boldsymbol{b}_i), \tag{18}$$

where $i = -1, 1$ and $j = 1, \ldots, n_i$.

$$\boldsymbol{a}_{-1} = \begin{pmatrix} 1 \\ 5 \end{pmatrix}, \boldsymbol{b}_{-1} = \begin{pmatrix} 10 \\ 12 \end{pmatrix}, \boldsymbol{a}_1 = \begin{pmatrix} 4 \\ 12 \end{pmatrix}, \boldsymbol{b}_1 = \begin{pmatrix} 6 \\ 13 \end{pmatrix}. \tag{19}$$

All the variables in $\boldsymbol{X}_{-1\,j}$ and $\boldsymbol{X}_{1\,j}$ have the class labels -1 (Majority) and 1 (Minority), respectively. Four cases are considered based on $b = 20, b = 10, b = 5$, and $b = 3$.

4.2.1. $b = 20$

We set $b = 20$, the training sample sizes of the positive and negative class are $(n_{-1}, n_1) = (1000, 50)$.
The test dataset is:

$$\boldsymbol{X}^*_{ij} = \begin{pmatrix} X^*_{ij1} \\ X^*_{ij2} \end{pmatrix} \overset{i.i.d.}{\sim} \textbf{Uniform}_2(\boldsymbol{a}_i, \boldsymbol{b}_i) + \boldsymbol{\epsilon}_{ij}, \tag{20}$$

where $i = -1, 1, j = 1, \ldots, n^*_i$, and

$$\boldsymbol{\epsilon}_{ij} = \begin{pmatrix} \epsilon_{ij1} \\ \epsilon_{ij2} \end{pmatrix} \overset{i.i.d.}{\sim} \boldsymbol{N}_2(\boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i). \tag{21}$$

Where

$$\boldsymbol{\mu}_{-1} = \begin{pmatrix} 0 \\ 0 \end{pmatrix}, \boldsymbol{\mu}_1 = \begin{pmatrix} 0 \\ 0 \end{pmatrix}, \boldsymbol{\Sigma}_{-1} = \begin{pmatrix} 0.1 & 0 \\ 0 & 0.1 \end{pmatrix}, \boldsymbol{\Sigma}_1 = \begin{pmatrix} 0.05 & 0 \\ 0 & 0.05 \end{pmatrix}. \tag{22}$$

The test data size is $(n^*_{-1}, n^*_1) = (200, 10)$. The scatter plots of this simulated data are given in Fig. 5.

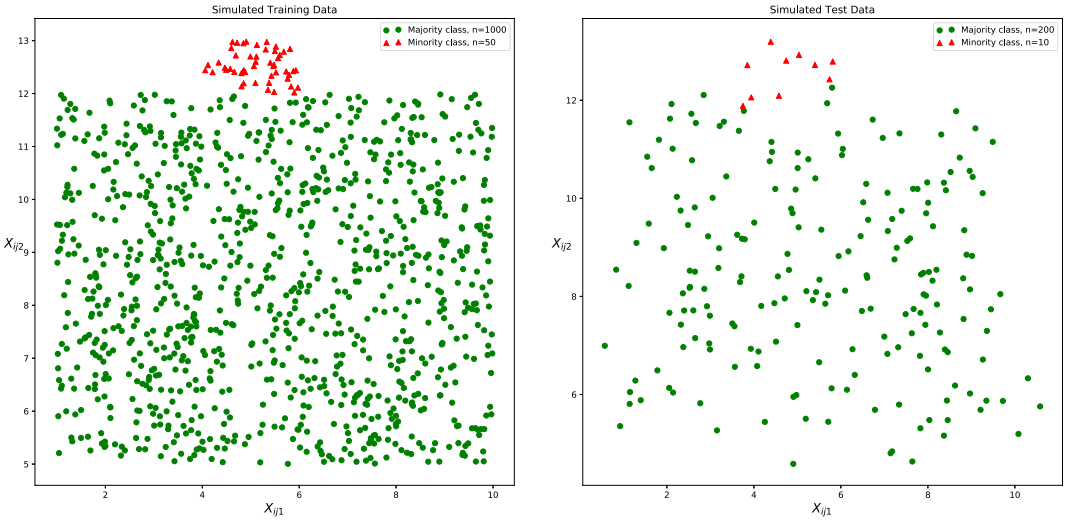

Fig. 5. Simulated Uniform Data with $b = 20$.

Except for the different sample sizes, the test data has an additional Gaussian error term compared to the training data. The training data points are located in two disjoint squares without overlap between the two classes, but the test data would have the overlap because of the error term. In Fig. 1 (3), this situation is referred to as a Data Shift problem, in the sense that the distributions of the training data and test data are different. This situation often occurs when a classifier is trained based on standard and clean training data. However, the test data comes from the real practice that would have randomness and is not the same as the training data.

Table 7 contains the classification results for this simulated Uniform data. The results are still similar to Table 5 and Table 6, but the $F-1$ *Measure* obtains a more significant improvement in the Reinforced AdaBoost and the Enhanced AdaBoost compared to other algorithms. Our proposed AdaBoost algorithms improve the classification ability for positive class apparently without losing the global classification ability.

4.2.2. $b = 10$

In the next three cases, $b = 10, 5$, and 3, the training data and test data are kept in the same distribution as the previous case but $(n_{-1}, n_1) = (1000, 100), (1000, 200)$, and $(1000, 333)$ for the training data; $(n^*_{-1}, n^*_1) = (200, 20), (200, 40)$, and $(200, 66)$ for the test data. The compared algorithms and criteria are the same as Table 3. The results of 100 repeated experiments for the situation of $b = 10$ are shown in Table 8. Almost all the classification results are better than the previous one since the imbalanced situation is weak. The differences between our two proposed algorithms are significant.

4.2.3. $b = 5$

Table 9 shows the results of 100 repeated experiments for the case of $b = 5$. All the classification results are better than the cases of $b = 20$ and 10. The differences between our two proposed algorithms are very significant since $b = 5$ is a relatively small number.

4.2.4. $b = 3$

In the case of $b = 3$, Table 10 shows the results of 100 repeated experiments. The Reinforced AdaBoost obtains the best classification results but the performances of all algorithms are very similar. $b = 3$ is too small to define an imbalanced data problem for this two-dimensional Uniform dataset. In this case, we do not need a specific algorithm to conduct the classification since most classic algorithms perform well.

4.3. Real data studies

Four binary real datasets are studied and their imbalanced index b are $15.13, 11.59, 5.14$, and 3.36. Since all the datasets are imbalanced, each dataset is randomly split in half as the training part and half as the test part to avoid the situation that there are no or too few numbers of positive samples in the training or test data.

4.3.1. Seismic-bumps data study

The Seismic-bumps data with $b = 15.13$ is obtained from the University of California, Irvine, Machine Learning Repository [23]. This dataset is about the seismic hazard in mining activity. Mining activity is always connected with the occurrence of dangers which are called mining hazards. Seismic hazard is a special and dangerous case in mining hazards. The Seismic-bumps dataset is intensely imbalanced with only 170 samples of the positive class among a total of 2584 samples and it contains 18 explanatory variables including seismic hazard assessment obtained by different methods, seismic energy recorded by different types of equipment, the number of seismic bumps in different energy range and so on. The hazardous state is the negative class defined as that there is a high energy seismic bump occurring in the next shift, which is indicated by $y = -1$. The non–hazardous state is the positive class defined as that there is no high energy seismic bump occurring in the next shift, which is indicated by $y = 1$.

Table 11 displays the Seismic-bumps data experiment. It shows the great performance of our proposed algorithms for the classification of this dataset. Except for the *specificity*, all evaluation criteria of the Reinforced AdaBoost and the Enhanced AdaBoost perform better than other algorithms, especially for the $F-1$ *Measure*. Another interesting fact is the results based on the Enhanced AdaBoost and the Reinforced AdaBoost are the same. Because $b = 15.13$ is a considerable number in this

Table 7
The Classification Results for Simulated Uniform Data ($b = 20$).

Algorithms	Evaluation Measures				
	Sensitivity	Specificity	F-1 Measure	AUC	Global Accuracy
SVMs	0.9198 (0.0906)	**0.9941 (0.0053)**	0.9027 (0.0623)	0.9570 (0.0449)	**0.9906 (0.0060)**
Ada-DT	0.9030 (0.0888)	0.9829 (0.0082)	0.8068 (0.0743)	0.9429 (0.0444)	0.9791 (0.0087)
Ada-LSVMs	0.9594 (0.0827)	0.9369 (0.0397)	0.6309 (0.1448)	0.9482 (0.0345)	0.9380 (0.0357)
Ada-RSVMs	0.9386 (0.0824)	0.9913 (0.0068)	0.8901 (0.0687)	**0.9650 (0.0410)**	0.9888 (0.0073)
En-Ada	0.9801 (0.0431)	0.9256 (0.0234)	0.9404 (0.0245)	0.9446 (0.0345)	0.9424 (0.0253)
Re-Ada	**0.9812 (0.0634)**	0.9353 (0.0353)	**0.9465 (0.0245)**	0.9364 (0.0465)	0.9467 (0.0253)

Table 8
The Classification Results for Simulated Uniform Data ($b = 10$).

Algorithms	Evaluation Measures				
	Sensitivity	Specificity	F-1 Measure	AUC	Global Accuracy
SVMs	0.9201 (0.0852)	**0.9977 (0.0067)**	0.9176 (0.0782)	0.9579 (0.0592)	**0.9960 (0.0103)**
Ada-DT	0.9088 (0.0734)	0.9855 (0.0177)	0.8444 (0.0324)	0.9502 (0.0397)	0.9800 (0.0104)
Ada-LSVMs	0.9622 (0.0841)	0.9393 (0.0288)	0.7783 (0.1332)	0.9544 (0.0651)	0.9544 (0.0117)
Ada-RSVMs	0.9466 (0.0455)	0.9941 (0.0110)	0.8876 (0.0558)	**0.9711 (0.0511)**	0.9903 (0.0107)
En-Ada	0.9898 (0.0842)	0.9400 (0.0311)	0.9471 (0.0119)	0.9403 (0.0418)	0.9503 (0.0331)
Re-Ada	**0.9924 (0.0338)**	0.9618 (0.0737)	**0.9595 (0.0188)**	0.9614 (0.0388)	0.9615 (0.0564)

Table 9
The Classification Results for Simulated Uniform Data ($b = 5$).

Algorithms	Evaluation Measures				
	Sensitivity	Specificity	F-1 Measure	AUC	Global Accuracy
SVMs	0.9366 (0.0952)	**0.9982 (0.0133)**	0.9331 (0.0472)	0.9693 (0.0452)	**0.9977 (0.0339)**
Ada-DT	0.9211 (0.0522)	0.9900 (0.0312)	0.8798 (0.0564)	0.9662 (0.0423)	0.9903 (0.0459)
Ada-LSVMs	0.9781 (0.0524)	0.9551 (0.0441)	0.8711 (0.1102)	0.9691 (0.0496)	0.9710 (0.0293)
Ada-RSVMs	0.9617 (0.0337)	0.9977 (0.0144)	0.9109 (0.0405)	**0.9821 (0.0449)**	0.9960 (0.0217)
En-Ada	0.9923 (0.0719)	0.9419 (0.0144)	0.9619 (0.0322)	0.9698 (0.0552)	0.9688 (0.0441)
Re-Ada	**0.9957 (0.1031)**	0.9722 (0.0441)	**0.9837 (0.0152)**	0.9908 (0.0351)	0.9894 (0.0614)

Table 10
The Classification Results for Simulated Uniform Data ($b = 3$).

Algorithms	Evaluation Measures				
	Sensitivity	Specificity	F-1 Measure	AUC	Global Accuracy
SVMs	0.9618 (0.0352)	**0.9992 (0.0077)**	0.9698 (0.0025)	0.9877 (0.0110)	**0.9998 (0.1739)**
Ada-DT	0.9594 (0.0037)	0.9974 (0.0166)	0.9577 (0.0319)	0.9823 (0.0055)	0.9973 (0.0037)
Ada-LSVMs	0.9890 (0.0144)	0.9817 (0.0119)	0.9388 (0.0771)	0.9871 (0.0411)	0.9933 (0.0037)
Ada-RSVMs	0.9917 (0.0411)	0.9988 (0.0083)	0.9688 (0.0172)	0.9916 (0.0073)	0.9985 (0.0072)
En-Ada	0.9987 (0.0133)	0.9954 (0.0017)	0.9851 (0.0052)	0.9974 (0.0121)	**0.9988 (0.0074)**
Re-Ada	**0.9989 (0.1031)**	0.9954 (0.0389)	**0.9855 (0.0558)**	**0.9981 (0.0652)**	**0.9988 (0.0434)**

Table 11
The Classification Results for Seismic-bumps Data ($b = 15.13$).

Algorithms	Evaluation Measures				
	Sensitivity	Specificity	F-1 Measure	AUC	Global Accuracy
SVMs	0.0000	**1.0000**	0.0000	0.5	0.9320
Ada-DT	0.1047	0.9873	0.1636	0.5460	0.9273
Ada-LSVMs	0.0000	0.9992	0.0000	0.4996	0.9312
Ada-RSVMs	0.6279	0.6675	0.2030	0.6477	0.6648
En-Ada	**0.9211**	0.9755	**0.9601**	**0.9101**	**0.9533**
Re-Ada	**0.9211**	0.9755	**0.9601**	**0.9101**	**0.9533**

dataset and γ_t should be relatively small in this high-dimensional data, the improvement part in the Reinforced AdaBoost compared to the Enhanced AdaBoost does not work.

4.3.2. Glass-2 data study

The Glass-2 dataset is obtained from the Knowledge Extraction Based on Evolutionary Learning (KEEL) Dataset Repository [1]. This dataset is originally from the USA Forensic Science Service and intensely imbalanced with only 17 samples of the positive class among a total of 214 samples, which means $b = 11.59$. The Glass-2 dataset has 8 explanatory variables including the refractive index, the different chemical elements content and so on. The positive class is labeled as the glass is non-float processed and it is indicated by $y = 1$. The negative class is defined as the glass is made by other processed methods and it is indicated by $y = -1$.

Table 12 displays the Glass-2 data experiment. It shows similar results as the Seismic-bumps data results. Except for the *specificity*, all evaluation measures of our proposed algorithms perform better than other algorithms, especially for the $F-1$ *Measure*. Compared to the Enhanced AdaBoost, the Reinforced AdaBoost has a slight improvement.

Table 12
The Classification Results for Glass-2 Data ($b = 11.59$).

Algorithms	Evaluation Measures				
	Sensitivity	Specificity	F-1 Measure	AUC	Global Accuracy
SVMs	0.0000	**1.0000**	0.0000	0.5	0.8972
Ada-DT	0.0909	0.9896	0.1538	0.5402	0.8972
Ada-LSVMs	0.0000	**1.0000**	0.0000	0.5000	0.8970
Ada-RSVMs	0.1818	0.9063	0.1818	0.5440	0.8318
En-Ada	0.8962	0.8327	0.8554	0.8948	0.8617
Re-Ada	**0.9056**	0.9087	**0.9074**	**0.9047**	**0.9072**

4.3.3. New-thyroid-1 data study

The New-thyroid-1 dataset is also obtained from the KEEL Dataset Repository [1]. This dataset is original from the Garavan Institute in Sydney, Australia. It includes the information of thyroid patients and was rearranged by KEEL to be a binary imbalanced dataset. The New-thyroid-1 dataset is imbalanced with 35 positive samples among a total of 215 samples, which indicates the imbalanced index is $b = 5.14$. This dataset contains 5 explanatory variables including the levels of different hormones like Thyroxin, Triiodothyronine and so on. The positive class is labeled as Hyperthyroidism, which is indicated by $y = 1$. The negative class is defined as non–Hyperthyroidism, which is indicated by $y = -1$.

Table 13 displays the results of the New-thyroid-1 data experiment. It shows the satisfactory performance of our proposed algorithms compared to others for this data classification problem. All the evaluation criteria of the Reinforced AdaBoost and the Enhanced AdaBoost perform better than other algorithms. Because $b = 5.143$ is a relatively small number in this dataset, the improvement part in the Reinforced AdaBoost compared to the Enhanced AdaBoost works.

4.3.4. Ecoli-1 data study

The Ecoli-1 dataset is also obtained from the KEEL Dataset Repository [1]. This dataset originally comes from the Institute of Molecular and Cellular Biology at Osaka University. It includes the information of the cellular localization sites of proteins and is rearranged by KEEL as a binary imbalanced dataset. The Ecoli-1 dataset is imbalanced with 77 positive samples among a total of 336 samples, which indicates the imbalanced index is $b = 3.36$. This dataset contains 7 explanatory variables including the signal sequence recognition scores by different methods, the presence of charge on N-terminus and so on. The positive class is labeled as the inner membrane without a signal sequence, which is indicated by $y = 1$. The negative class is defined as other situations, which is indicated by $y = -1$.

Table 14 displays the results of the Ecoli-1 data experiment. It shows the Reinforced AdaBoost has the largest $F-1$ *Measure* and *Global Accuracy*. Compared to the Enhanced AdaBoost, the relatively small value of $b = 3.36$ makes the improvement part in Reinforced AdaBoost play a significant role. The differences between our two proposed algorithms are more obvious in this case.

Table 13
The Classification Results for New-thyroid-1 Data ($b = 5.14$).

Algorithms	Evaluation Measures				
	Sensitivity	Specificity	F-1 Measure	AUC	Global Accuracy
SVMs	0.2632	**1.0000**	0.4167	0.6316	0.8704
Ada-DT	0.7895	0.9888	0.8571	0.8891	0.9537
Ada-LSVMs	0.8421	0.9888	0.8889	0.9154	0.9630
Ada-RSVMs	0.8421	0.9663	0.8421	0.9042	0.9444
En-Ada	0.9881	0.9495	0.9673	0.9888	0.9680
Re-Ada	**0.9888**	**1.000**	**0.9944**	**0.9944**	**0.9943**

Table 14
The Classification Results for Ecoli-1 Data ($b = 3.36$).

Algorithms	Evaluation Measures				
	Sensitivity	Specificity	F-1 Measure	AUC	Global Accuracy
SVMs	0.8889	0.9242	0.8205	0.9066	0.9167
Ada-DT	0.8611	0.9091	0.7848	0.8851	0.8988
Ada-LSVMs	0.0000	**1.0000**	0.0000	0.5000	0.7857
Ada-RSVMs	**0.9444**	0.8636	0.7727	0.9040	0.8810
En-Ada	0.9322	0.9437	0.9383	**0.9280**	0.9379
Re-Ada	0.9205	0.9706	**0.9457**	0.8371	**0.9441**

Based on the numeric studies, our two proposed algorithms outperform the traditional ones in imbalanced data classification. With large imbalanced index b, our proposed algorithms show excellent performance. The difference between the Enhanced AdaBoost and the Reinforced AdaBoost is more obvious in the case of the small value of b and they would have the same performance in a large b case such as the Seismic-bumps data study.

5. Discussion and comments

Class imbalanced problem is a critical issue in many fields and raises considerable hardship for classification. To address the challenge of the imbalanced class problem, the Enhanced AdaBoost and Reinforced AdaBoost are proposed in this paper to improve the Adaboost algorithm. The innovation point is an adjustment of the weighted vote parameters of weaker classifiers α^t, which includes the global error rate and the positive class accuracy rate. In addition, our algorithms consider the imbalanced index b, to improve the performance of classification in the imbalanced class problem.

Numerical studies of two kinds of simulated datasets and four real datasets compare the proposed algorithms and four other traditional algorithms. Our algorithms outperform in the positive class, especially regarding $F-1$ *Measure*, and do not lose the global accuracy rate. When b is large, the Enhanced AdaBoost is a powerful algorithm for the imbalanced data classification. But if b is relatively small, the Reinforced AdaBoost is recommended to be used for the imbalanced data problem to obtain a better result.

CRediT authorship contribution statement

Wenyang Wang: Methodology, Software, Formal analysis, Visualization, Writing - original draft. **Dongchu Sun:** Conceptualization, Formal analysis, Resources, Writing - review & editing, Supervision.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

The project was partially supported by the 111 Project of China (No. B14019) and the National Natural Science Foundation of China (No. 11671146).

The authors gratefully acknowledge the very constructive comments of the editor and the anonymous referees.

Appendix A. Proofs

A.1. Proof of Theorem 1.

It follows from **Fact 2** that

$$\begin{aligned}\sum_{i:y_i=-1}H_a(\boldsymbol{x}_i) &= N_p k\sum_{t=1}^{T}\{(2\gamma_t-1)+(2\epsilon_t-1)(b+1)\}exp\{\beta(2\gamma_t-1)\}\\ &= \sum_{i:y_i=1}H_a(\boldsymbol{x}_i)+N_p k(b+1)\sum_{t=1}^{T}(2\epsilon_t-1)\,exp\{\beta(2\gamma_t-1)\}.\end{aligned}$$

$N_p k(b+1)\sum_{t=1}^{T}(2\epsilon_t-1)\,exp\{\beta(2\gamma_t-1)\}<0$ because of $0\leqslant\epsilon_t<\frac{1}{2}$, which means $\sum_{i:y_i=-1}H_a(\boldsymbol{x}_i)<\sum_{i:y_i=1}H_a(\boldsymbol{x}_i)$ always holds.

(a) When $\gamma_t\in[0,\frac{1}{2})$, it is obvious that $\sum_{i:y_i=1}H_a(\boldsymbol{x}_i)<0$, this implies (3);

(b) When $\gamma_t=\frac{1}{2}$, (4) follows from $\sum_{i:y_i=1}H_a(\boldsymbol{x}_i)=0$;

(c) When $\gamma_t\in(\frac{1}{2},1]$, under (5),

$$\begin{aligned}\sum_{i:y_i=1}H_a(\boldsymbol{x}_i) &= N_p k\sum_{t=1}^{T}(2\gamma_t-1)\,exp\{\beta(2\gamma_t-1)\}>0,\\ \sum_{i:y_i=-1}H_a(\boldsymbol{x}_i) &= N_p k\sum_{t=1}^{T}\{(2\gamma_t-1)+(2\epsilon_t-1)(b+1)\}exp\{\beta(2\gamma_t-1)\}<0.\end{aligned}$$

These imply (6).

A.2. Proof of Theorem 2.

Under assumption (13),

$$\begin{aligned}\sum_{i:y_i=1}H_3'(\boldsymbol{x}_i)-\sum_{i:y_i=1}H_1(\boldsymbol{x}_i)>0 &\iff \frac{exp(\gamma_t-0.5)+0.5}{0.5-\epsilon_t}+\frac{0.5(b+1)}{\gamma_t-0.5}-1>0\\ &\iff (b+1)\frac{exp(\gamma_t-0.5)+1}{\gamma_t-0.5}>1.\end{aligned}\tag{23}$$

(23) always holds, so that (14) gets proved.

$$\begin{aligned}\sum_{i:y_i=-1}H_3'(\boldsymbol{x}_i)-\sum_{i:y_i=-1}H_1(\boldsymbol{x}_i)\leqslant 0 \iff & \{(\gamma_t-0.5)+(\epsilon_t-0.5)(b+1)\}\left\{\tfrac{exp(\gamma_t-0.5)+0.5}{0.5-\epsilon_t}+\tfrac{0.5(b+1)}{\gamma_t-0.5}-1\right\}\leqslant 0\\ \iff & \left\{(b+1)-\tfrac{0.5(b+1)^2}{\gamma_t-0.5}\right\}(0.5-\epsilon_t)^2-\{(\gamma_t-0.5)+(b+1)exp(\gamma_t-0.5)\}(0.5-\epsilon_t)\\ & +(\gamma_t-0.5)exp(\gamma_t-0.5)+0.5(\gamma_t-0.5)\leqslant 0\\ \iff & 0.5-\epsilon_t\leqslant\tfrac{\gamma_t-0.5}{b+1}\\ \iff & 0.5\{1-(2\gamma_t-1)/(b+1)\}\leqslant\epsilon_t.\end{aligned}\tag{24}$$

(24) always holds under assumption (13), so that (15) gets proved.

Appendix B. Supplementary data

Supplementary data associated with this article can be found, in the online version, athttps://doi.org/10.1016/j.ins.2021.03.042.

References

[1] Jesús Alcalá-Fdez, Alberto Fernández, Julián Luengo, Joaquín Derrac, Salvador García, Luciano Sánchez, Francisco Herrera, Keel data-mining software tool: data set repository, integration of algorithms and experimental analysis framework, J. Multiple-Valued Logic Soft Comput. 17 (2011).
[2] Ethem Alpaydin, Introduction to Machine Learning, MIT Press, 2009.
[3] Gustavo EAPA Batista, Ronaldo C. Prati, Maria Carolina Monard, A study of the behavior of several methods for balancing machine learning training data. ACM SIGKDD Explor. Newslett. 6(1) (2004) 20–29. ISSN 1931–0145.https://doi.org/10.1145/1007730.1007735.
[4] Kwabena Ebo Bennin, Jacky Keung, Passakorn Phannachitta, Akito Monden, Solomon Mensah, Mahakil: diversity based oversampling approach to alleviate the class imbalance issue in software defect prediction, IEEE Trans. Software Eng. 44 (6) (2017) 534–550.
[5] Yiyang Bian, Min Cheng, Chen Yang, Yuan Yuan, Qing Li, J. Leon Zhao, Liang Liang, Financial fraud detection: a new ensemble learning approach for imbalanced data, in: PACIS, 2016, pp. 315.
[6] Philip K. Chan, Salvatore J. Stolfo, Toward scalable learning with non-uniform class and cost distributions: a case study in credit card fraud detection, in: KDD, vol. 1998, 1998, pp. 164–168.
[7] Nitesh V. Chawla, Aleksandar Lazarevic, Lawrence O. Hall, Kevin W. Bowyer, Smoteboost: improving prediction of the minority class in boosting, in: European conference on principles of data mining and knowledge discovery, Springer, 2003, pp. 107–119.
[8] Fanyong Cheng, Jing Zhang, Cuihong Wen, Cost-sensitive large margin distribution machine for classification of imbalanced data, Pattern Recogn. Lett. 80 (2016) 107–112. ISSN 0167–8655.https://doi.org/10.1016/j.patrec.2016.06.009.
[9] Wei Fan, Salvatore J. Stolfo, Junxin Zhang, Philip K. Chan, Adacost: misclassification cost-sensitive boosting, in: Icml, 1999, pp. 97–105.
[10] Yoav Freund, Robert E Schapire, A decision-theoretic generalization of on-line learning and an application to boosting, J. Comput. Syst. Sci. 55 (1) (1997) 119–139.
[11] Yoav Freund, Robert Schapire, Naoki Abe, A short introduction to boosting, J.-Japanese Soc. Artif. Intell. 14 (771–780) (1999) 1612.
[12] Mikel Galar, Alberto Fernandez, Edurne Barrenechea, Humberto Bustince, Francisco Herrera, Ordering-based pruning for improving the performance of ensembles of classifiers in the framework of imbalanced datasets, Inf. Sci. 354 (2016) 178–196. ISSN 0020–0255.https://doi.org/10.1016/j.ins.2016.02.056.
[13] Xia Geng, Yu-Quan Zhu, Zhi Yang, A novel classification method for class-imbalanced data and its application in microrna recognition, Int. J. Bioautomation 22 (2) (2018).
[14] Guo Haixiang, Li Yijing, Li Yanan, Liu Xiao, Li Jinling, Bpso-adaboost-knn ensemble learning algorithm for multi-class imbalanced data classification, Eng. Appl. Artif. Intell. 49 (2016) 176–193.
[15] Peter Harrington, Machine Learning in Action, Manning Publications Co., 2012.
[16] Chen Huang, Yining Li, Change Loy Chen, Xiaoou Tang, Deep imbalanced learning for face recognition and attribute prediction, IEEE Trans. Pattern Anal. Mach. Intell. (2019).
[17] R. Kohavi, F. Provost, Glossary of terms: special issue on applications of machine learning and the knowledge discovery process. 1998 (cited 2016).
[18] Sotiris Kotsiantis, Dimitris Kanellopoulos, Panayiotis Pintelas, et al., Handling imbalanced datasets: a review, GESTS Int. Trans. Comput. Sci. Eng. 30 (2006).
[19] Miroslav Kubat, Robert C. Holte, Stan Matwin, Machine learning for the detection of oil spills in satellite radar images, Mach. Learn. 30(2–3) (1998) 195–215. ISSN 1573–0565.https://doi.org/10.1023/A:1007452223027.
[20] Wonji Lee, Chi-Hyuck Jun, Jong-Seok Lee, Instance categorization by support vector machines to adjust weights in adaboost for imbalanced data classification, Inf. Sci. 381 (2017) 92–103, https://doi.org/10.1016/j.ins.2016.11.014.
[21] David Lewis, William Gale, Training text classifiers by uncertainty sampling, 1994.
[22] Chuang Li, Xiao-Qing Ding, Wu. You-Shou, Revised adaboost algorithm – ad adaboost, Jisuanji Xuebao/Chin. J. Comput. 30 (1) (2007) 103–109.
[23] M. Lichman, UCI machine learning repository, 2013. http://archive.ics.uci.edu/ml/datasets/seismic-bumps.
[24] Victoria Lopez, Alberto Fernandez, Jose G. Moreno-Torres, Francisco Herrera, Analysis of preprocessing vs. cost-sensitive learning for imbalanced classification. open problems on intrinsic data characteristics, Expert Syst. Appl. 39(7) (2012) 6585–6608. ISSN 0957–4174.https://doi.org/10.1016/j.eswa.2011.12.043.
[25] Victoria Lopez, Alberto Fernandez, Salvador Garcia, Vasile Palade, Francisco Herrera, An insight into classification with imbalanced data: empirical results and current trends on using data intrinsic characteristics, Inf. Sci. 250 (2013) 113–141.
[26] Sara Makki, Zainab Assaghir, Yehia Taher, Rafiqul Haque, Mohand-Said Hacid, Hassan Zeineddine, An experimental study with imbalanced classification approaches for credit card fraud detection, IEEE Access 7 (2019) 93010–93022.
[27] Md Ochiuddin Miah, Sakib Shahriar Khan, Swakkhar Shatabda, Dewan Md Farid, Improving detection accuracy for imbalanced network intrusion classification using cluster-based under-sampling with random forests, in: 2019 1st International Conference on Advances in Science, Engineering and Robotics Technology (ICASERT), IEEE, 2019, pp. 1–5.
[28] Raul Rojas, Adaboost and the super bowl of classifiers a tutorial introduction to adaptive boosting, Freie University, Berlin, Tech. Rep, 2009.
[29] Robert E. Schapire, Yoram Singer, Improved boosting algorithms using confidence-rated predictions, Mach. Learn. 37 (3) (1999) 297–336.
[30] Qinbao Song, Yuchen Guo, Martin Shepperd, A comprehensive investigation of the role of imbalanced learning for software defect prediction, IEEE Trans. Software Eng. 45 (12) (2018) 1253–1269.
[31] Yanmin Sun, Mohamed S. Kamel, Andrew K.C. Wong, Yang Wang, Cost-sensitive boosting for classification of imbalanced data, Pattern Recogn. 40(12) (2007) 3358–3378. ISSN 0031–3203.https://doi.org/10.1016/j.patcog.2007.04.009.

[32] Xinmin Tao, Qing Li, Wenjie Guo, Chao Ren, Chenxi Li, Rui Liu, Junrong Zou, Self-adaptive cost weights-based support vector machine cost-sensitive ensemble for imbalanced data classification, Inf. Sci. 487 (2019) 31–56.
[33] Jason Van Hulse, Taghi M. Khoshgoftaar, Amri Napolitano, Experimental perspectives on learning from imbalanced data, in: Proceedings of the 24th International Conference on Machine Learning, ICML '07, New York, NY, USA, 2007, ACM, pp. 935–942. ISBN 978-1-59593-793-3.https://doi.org/10.1145/1273496.1273614.
[34] Paul Viola, Michael Jones, Fast and robust classification using asymmetric adaboost and a detector cascade. vol. 14, 2002, pp. 1311–1318.
[35] Gary M. Weiss, Mining with rarity: a unifying framework, ACM Sigkdd Explorations Newslett. 6(1) (2004) 7–19. ISSN 1931–0145.https://doi.org/10.1145/1007730.1007734.
[36] Songqing Yue. Imbalanced malware images classification: a cnn based approach. arXiv preprint arXiv:1708.08042, 2017.
[37] Bin Zhou, Tuo Wang, Mingqi Luo, Shijuan Pan, An online tracking method via improved cost-sensitive adaboost, in: 2017 Eighth International Conference on Intelligent Control and Information Processing (ICICIP), IEEE, 2017, pp. 49–54.