Hindawi Publishing Corporation
Journal of Electrical and Computer Engineering
Volume 2015, Article ID 835357, 17 pages
http://dx.doi.org/10.1155/2015/835357

*Research Article*

# Analysis of Generalization Ability for Different AdaBoost Variants Based on Classification and Regression Trees

**Shuqiong Wu[1] and Hiroshi Nagahashi[2]**

[1] *Department of Computational Intelligence and Systems Science, Tokyo Institute of Technology, 4259 Nagatsuta-cho, Midori-ku, Yokohama, Kanagawa 226-8503, Japan*
[2] *Imaging Science and Engineering Laboratory, Tokyo Institute of Technology, 4259 Nagatsuta-cho, Midori-ku, Yokohama, Kanagawa 226-8503, Japan*

Correspondence should be addressed to Shuqiong Wu; qq19850916@gmail.com

Received 13 November 2014; Accepted 21 January 2015

Academic Editor: Sos Agaian

As a machine learning method, AdaBoost is widely applied to data classification and object detection because of its robustness and efficiency. AdaBoost constructs a global and optimal combination of weak classifiers based on a sample reweighting. It is known that this kind of combination improves the classification performance tremendously. As the popularity of AdaBoost increases, many variants have been proposed to improve the performance of AdaBoost. Then, a lot of comparison and review studies for AdaBoost variants have also been published. Some researchers compared different AdaBoost variants by experiments in their own fields, and others reviewed various AdaBoost variants by basically introducing these algorithms. However, there is a lack of mathematical analysis of the generalization abilities for different AdaBoost variants. In this paper, we analyze the generalization abilities of six AdaBoost variants in terms of classification margins. The six compared variants are Real AdaBoost, Gentle AdaBoost, Modest AdaBoost, Parameterized AdaBoost, Margin-pruning Boost, and Penalized AdaBoost. Finally, we use experiments to verify our analyses.

## 1. Introduction

In the last two decades, AdaBoost and its variants were used in various fields, such as face detection [1, 2], hands detection [3, 4], and human detection [5]. AdaBoost was first introduced to the machine learning literature by Freund and Schapire [6]; and it has obtained tremendous success in classification. AdaBoost is proved efficient in increasing the classification margins of training data [7]. To describe AdaBoost mathematically, Schapire and Singer proposed Real AdaBoost, which is a generalized version of AdaBoost [8]. Real AdaBoost calculates its weak hypotheses by directly optimizing the upper bounds of training errors. Therefore, it converges faster than AdaBoost in the training [9]. To improve the training speed of Real AdaBoost, Wu and Nagahashi devised Parameterized AdaBoost which utilizes a new weight adjustment policy [10]. In 2000, Friedman et al. used additive logistic models to explain AdaBoost and proposed Gentle AdaBoost which computes weak hypotheses by minimizing the least square errors [11]. Friedman et al. also proved that Gentle AdaBoost is more robust than AdaBoost and Real AdaBoost [11]. For reducing the generalization error of Gentle AdaBoost, A. Vezhnevets and V. Vezhnevets suggested Modest AdaBoost which highlights weak classifiers that work well on difficult-to-classify instances [12]. Modest AdaBoost achieves better generalization errors than Gentle AdaBoost in some data sets [13]. However, its performance is unstable because the accuracy drops occasionally. For the same purpose, Wu and Nagahashi devised Margin-pruning Boost [14] and Penalized AdaBoost [15]. Margin-pruning Boost applies a weight reinitialization approach to reduce the influence from noise-like data, while Penalized AdaBoost improves Margin-pruning Boost by introducing an adaptive weight resetting policy. Moreover, it utilizes a margin distribution to penalize the misclassification of small-margin instances. Freund created BrownBoost to reduce the influence of outliers in training [16]. LPBoost was introduced to optimize the minimal margin of training data by using linear programming [17]. However, a comparison showed that LPBoost overall performs worse than AdaBoost [18]. Similarly to Margin-pruning Boost, MadaBoost and SmoothBoost were devised to increase the robustness against malicious noise data [19, 20]. Some AdaBoost variants such as AdaCost, AdaC1, AdaC2, AdaC3, CSB0, CSB1, CSB2, and RareBoost assign weights to positive training instances and negative training instances differently to obtain a better performance on imbalanced data sets [21–24]. While others trade off the integrity of training data for a faster training [25–28]. In 2004, AdaTree was proposed to speed up the training process. It utilizes the same way as in AdaBoost to select weak classifiers but combines them in a nonlinear manner [29]. Filterboost and Regularized AdaBoost were proposed to solve overfitting problem [30, 31]. Filterboost is based on a new logistic regression technique whereas Regularized AdaBoost requires validation subsets to identify and correct the overfitting iteratively. FloatBoost and FM-AdaBoost filter the less effective weak classifiers so that they can outperform AdaBoost when they have the same number of weak classifiers as AdaBoost [32, 33]. Nevertheless, they require more training cycles than AdaBoost. Nowadays, many novel AdaBoost variants were devised to improve the generalization ability such as SoftBoost, Interactive Boosting, ReweightBoost, Soft-LPBoost, and RobustBoost [34–39]. SoftBoost maximizes a soft margin instead of the hard margin used in AdaBoost [34, 36]. Interactive Boosting gives weights to both features and training instances [37]. ReweightBoost builds a tree structure by reusing the selected weak classifiers. Nevertheless, it can only use stump decision trees as its weak classifiers [38]. Soft-LPBoost combines SoftBoost with LPBoost [39]. While RobustBoost is an extension of BrownBoost [35]. All the five AdaBoost variants can achieve better generalization errors than AdaBoost. However, they suffered from a lot of complicated calculations which may lead to a longer training time. In the last few years, A novel approach called SemiBoost has been developed rapidly. It combines supervised learning with semisupervised learning by using both the labelled and unlabelled training instances [40]. The purpose of SemiBoost is to increase the generalization ability when the labelled training instances are insufficient [41]. In addition, AdaBoost.M1, Conservative.2 AdaBoost, and Aggressive AdaBoost are AdaBoost variants proposed for multiple classification problems [42].

With the proposals of many AdaBoost variants, a lot of surveys and comparison studies based on these variants were published. Miao and Heaton compared AdaBoost with Random Forest in ecosystem classification problems and showed that AdaBoost overall outperforms Random Forest [43]. Another comparison of AdaBoost and neural networks proved that AdaBoost ensemble of trees performs better than an individual neural network in experiments using cross validations [44]. Research in [45] applied AdaBoost and SVM to the Synthetic Aperture Radar Automatic Target Recognition Systems and found that AdaBoost is more robust than SVM. Ferreira briefly introduced many boosting algorithms and labelled them as "supervised learning" or "semisupervised learning" [46]. Seiffert et al. compared resampling boost algorithms with reweighting boost algorithms on imbalanced data sets and summarized that boosting by resampling generally outperforms boosting by reweighting [47]. A comparison of LPBoost and AdaBoost based on the experimental results of UCI repository was also conducted in [18]. Hegazy and Denzler evaluated AdaBoost and SoftBoost in generic object recognition and concluded that AdaBoost is more suitable for low-noise data sets while SoftBoost is more suitable for high-noise data sets [48]. Another study which compares AdaBoost with AdaTree was accomplished by Drauschke and Forstner [49]. Its experimental results showed that AdaTree usually performs better than AdaBoost but is prone to overfitting due to its tree-like structure. Jurić-Kavelj and Petrović evaluated three AdaBoost variants (Real, Gentle, and Modest AdaBoost) based on experiments in leg detection and found that Modest AdaBoost can not reduce the error rate as the number of iterations increases [50]. Sun et al. compared Discrete, Real, and Gentle AdaBoost by analyzing the experimental results in license plate detection and explained that Gentle AdaBoost achieves better performance than the other two methods [51]. Comparison in [52] focused on comparing weak classifiers of AdaBoost constructed by Bayes net, naive Bayes, and decision trees, and it showed that decision trees are the best. A review systematically introduced AdaBoost variants proposed during 1999 to 2012. Nevertheless, there is a lack of comparison between different AdaBoost variants [53].

In general, the above surveys and comparison studies either introduce the basic ideas of AdaBoost variants or compare different AdaBoost variants by experiments in a specific research. Differently from these studies, we compare the generalization abilities of six AdaBoost variants (Real AdaBoost, Gentle AdaBoost, Modest AdaBoost, Parameterized AdaBoost, Margin-pruning Boost, and Penalized AdaBoost) by analyzing the classification margins. The remainder of this paper is organized as follows. Section 2 explains the materials and methods. Section 3 shows experimental results. Section 4 draws a conclusion.

## 2. Materials and Methods

This section describes the training data and weak classifiers used in our research. It also explains the basic ideas of the six compared AdaBoost variants and their generalization abilities in terms of the classification margins.

*2.1. Training Data and Weak Classifiers.* Here we give a brief introduction of the training data. Given $S = \{(x_1, y_1), (x_2, y_2), \dots, (x_N, y_N)\}$ is a training set, where $N$ is the number of instances. We let $y_i$ be 1 if $x_i$ is positive, or −1 if $x_i$ is negative. In this paper, we only discuss binary classification problems. We use CART as weak classifiers [54]. As shown in Figure 1, a CART is a decision tree whose leaves output the classification results and inner nodes split the tree to minimize its error rate. In Figure 1, $X_i = [x_{i,1}, x_{i,2}, \dots, x_{i,m}]$ is the feature vector of instance $x_i$ [54].

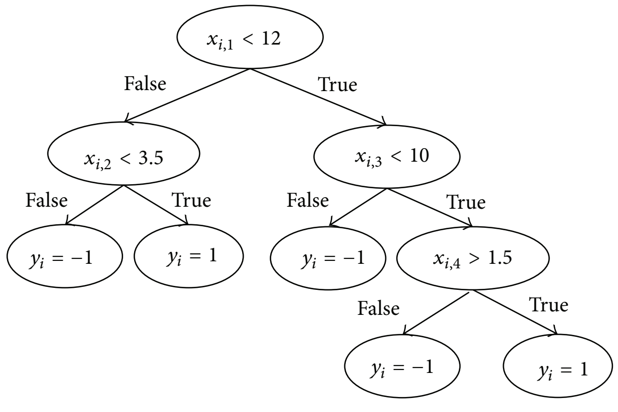


Figure 1: An example of classification and regression tree.

### 2.2. Different AdaBoost Variants

*2.2.1. AdaBoost and Real AdaBoost.* AdaBoost is a machine learning method. At each round, it increases the weights of misclassified instances and decreases the weights of correctly classified instances. The weight adjustment policy is most important because AdaBoost can focus on difficult-to-classify instances. A generalized version of AdaBoost which is called Real AdaBoost is described as follows.

*Algorithm 1* (Real AdaBoost). (1) Set the initial weights $w_{i,1} = 1/N$, where $i = 1, 2, \ldots, N$.

(2) Do the following tasks for $t = 1, 2, \ldots, T$.

(a) Train a weak classifier based on the weighted instances to divide the training set $S$ into $C$ partitions. Each leaf of the CART represents a partition. For any partition $S_t^j$ where $j \in \{1, 2, \ldots, C\}$, calculate $W_{t+}^j$ and $W_{t-}^j$ as follows:

$$W_{t+}^j = \sum_{i:x_i \in S_t^j \wedge y_i = 1} w_{i,t}. \qquad W_{t-}^j = \sum_{i:x_i \in S_t^j \wedge y_i = -1} w_{i,t}. \tag{1}$$

(b) Compute the weak hypothesis for each partition $S_t^j$:

$$f_t^j(x) = 0.5 \times \ln\left(\frac{W_{t+}^j}{W_{t-}^j}\right). \tag{2}$$

For any training instance $x_i$, let its weak hypothesis $f_t(x_i)$ be $f_t^j(x)$, where $j$ is the index of partition that $x_i$ falls into.

(c) Do weight updating by

$$w_{i,t+1} = \frac{w_{i,t} \exp\left[-y_i f_t(x_i)\right]}{Z_t}, \qquad Z_t = \sum_i w_{i,t} \exp\left[-y_i f_t(x_i)\right]. \tag{3}$$

(3) Let $F_T(x_i) = \sum_{t=1}^{T} f_t(x_i)$ and output the strong classifier $H(x_i) = \text{sign}[F_T(x_i)]$.

In AdaBoost, the values of weak hypotheses are +1 or −1. But in Real AdaBoost, they are real numbers. The sign of each weak hypothesis $f_t^j(x)$ stands for the class of the weighted majority of instances falls into the partition $S_t^j$, and the absolute value of $f_t^j(x)$ represents a predication confidence.

*2.2.2. Gentle AdaBoost.* Gentle AdaBoost utilizes the same weight adjustment policy as Real AdaBoost does. However, it computes the weak hypotheses in a different way. Next we introduce Gentle AdaBoost as follows.

*Algorithm 2* (Gentle AdaBoost). (1) Set the initial weights $w_{i,1} = 1/N$, where $i = 1, 2, \ldots, N$.

(2) Do the following tasks for $t = 1, 2, \ldots, T$.

(a) Train a weak classifier based on the weighted instances and then calculate $W_{t+}^j$ and $W_{t-}^j$ for each partition $S_t^j$ the same as in Step (2)(a) of Algorithm 1.

(b) Compute the weak hypothesis for each partition $S_t^j$:

$$f_t^j(x) = \frac{\left(W_{t+}^j - W_{t-}^j\right)}{\left(W_{t+}^j + W_{t-}^j\right)}. \tag{4}$$

For any training instance $x_i$, its weak hypothesis $f_t(x_i)$ equals $f_t^j(x)$, where $j$ is the index of partition that $x_i$ belongs to.

(c) Update instances' weights by (3).

(3) Let $F_T(x_i) = \sum_{t=1}^{T} f_t(x_i)$ and output the strong classifier $H(x_i) = \text{sign}[F_T(x_i)]$.

In Algorithm 2, the weak hypotheses are calculated by optimizing the weighted least square errors. Thus, it is more robust and stable than AdaBoost [11].

*2.2.3. Modest AdaBoost.* Modest AdaBoost was proposed to suppress the generalization error of Gentle AdaBoost. It is explained by Algorithm 3.

*Algorithm 3* (Modest AdaBoost). (1) Set the initial weights $w_{i,1} = 1/N$, where $i = 1, 2, \ldots, N$.

(2) Do the following tasks for $t = 1, 2, \ldots, T$.

(a) Train a weak classifier based on the weighted instances and then calculate $W_{t+}^j$ and $W_{t-}^j$ for each partition $S_t^j$ the same as in Step (2)(a) of Algorithm 1.

(b) Calculate an inverted weight $\overline{w}_{i,t}$ by

$$\overline{w}_{i,t} = \frac{(1 - w_{i,t})}{M_t}, \tag{5}$$

where $M_t = \sum_i (1 - w_{i,t})$. Then compute $\overline{W}_{t+}^j$ and $\overline{W}_{t-}^j$ for each partition $S_t^j$ as

$$\overline{W}_{t+}^j = \sum_{i:x_i \in S_t^j \wedge y_i = 1} \overline{w}_{i,t}, \qquad \overline{W}_{t-}^j = \sum_{i:x_i \in S_t^j \wedge y_i = -1} \overline{w}_{i,t}. \tag{6}$$

(c) Compute the weak hypothesis for every partition $S_t^j$:

$$f_t^j(x) = W_{t+}^j\left(1 - \overline{W}_{t+}^j\right) - W_{t-}^j\left(1 - \overline{W}_{t-}^j\right). \tag{7}$$

For an instance $x_i$, its weak hypothesis $f_t(x_i)$ equals $f_t^j(x)$, where $j$ is the index of partition that $x_i$ belongs to.

(c) Update instances' weights by (3).

(3) Let $F_T(x_i) = \sum_{t=1}^{T} f_t(x_i)$ and output the strong classifier $H(x_i) = \text{sign}[F_T(x_i)]$.

Modest AdaBoost uses an "inverted" distribution to decrease the contribution of weak hypotheses that only work well on instances with small weights [12].

*2.2.4. Parameterized AdaBoost.* Parameterized AdaBoost is shown by Algorithm 4. Its purpose is to speed up the training process of Real AdaBoost.

*Algorithm 4* (Parameterized AdaBoost). (1) Set the initial conditions in this step by letting $w_{i,1} = 1/N$ and $F_0(x_i) = 0$, where $i = 1, 2, \ldots, N$.

(2) Do the following tasks for $t = 1, 2, \ldots, T$.

(a) Train a weak classifier based on the weighted instances and then calculate $W_{t+}^j$ and $W_{t-}^j$ for each partition $S_t^j$ the same as in Step (2)(a) of Algorithm 1.

(b) Calculate the weak hypotheses by (2).

(c) Let $F_t(x_i) = F_{t-1}(x_i) + f_t(x_i)$ and $\alpha \in (0, 1)$. Then update the weights of instances by

$$w_{i,t+1} = \frac{\exp\left[-y_i F_t(x_i) - \alpha\left|y_i F_t(x_i)\right|\right]}{Z_t}, \tag{8}$$

$$Z_t = \sum_i \exp\left[-y_i F_t(x_i) - \alpha\left|y_i F_t(x_i)\right|\right]. \tag{9}$$

(3) Output the strong classifier $H(x_i) = \text{sign}[F_T(x_i)]$.

For tuning parameter $\alpha$, the training and generalization errors on a data set Gamma Telescope which includes nearly 20000 instances were measured for $\alpha = \{0.1, 0.3, 0.5, 0.7, 0.9\}$. The results showed that $\alpha = 0.5$ has the best overall performance, so $\alpha$ is set to be 0.5 [10].

The difference between Real and Parameterized AdaBoost is the weight updating policy. In Step (2)(c), Parameterized AdaBoost adds a parameter and an absolute item to emphasize the instances whose margins are near 0 [10].

*2.2.5. Margin-Pruning Boost.* Margin-pruning Boost was designed to decrease the influence of noise-like instances. Next we describe this approach as follows.

*Algorithm 5* (Margin-pruning Boost). (1) Set the initial weights $w_{i,1} = 1/N$, where $i = 1, 2, \ldots, N$.

(2) Do the following tasks for $t = 1, 2, \ldots, T$.

(a) Train a weak classifier based on the weighted instances and then calculate $W_{t+}^j$ and $W_{t-}^j$ for each partition $S_t^j$ the same as in Step (2)(a) of Algorithm 1.

(b) Compute the weak hypothesis for each partition $S_t^j$ by (4).

(c) Update the weights of instances as

$$w_{i,t+1} = \exp\left[-y_i \sum_t f_t(x_i)\right]. \tag{10}$$

(d) Set a threshold $Q_{t+1}$ by the following equation:

$$Q_{t+1} = \max_i \{w_{i,t+1}\} - \frac{\max_i \{w_{i,t+1}\} - \min_i \{w_{i,t+1}\}}{50}. \tag{11}$$

For any instance $x_i$, if $w_{i,t+1} > Q_{t+1}$, reset $w_{i,t+1} = 1$ and $\sum_t f_t(x_i) = 0$. Compute $Z_t = \sum_i w_{i,t+1}$; then do normalization by letting $w_{i,t+1} = w_{i,t+1}/Z_t$.

(3) Let $F_T(x_i) = \sum_{t=1}^{T} f_t(x_i)$ and output the strong classifier $H(x_i) = \text{sign}[F_T(x_i)]$.

Margin-pruning Boost restrains the weight increase of potential noise instances by resetting their weights and summed weak hypotheses [14]. Here the resetting of the sum of weak hypotheses can keep the weights of noise-like instances small. On the other hand, the resetting means that the combination of these weak hypotheses (the current strong hypothesis) for these noise-like instances is reset to be 0 because it can not correctly classify these instances.

*2.2.6. Penalized AdaBoost.* Penalized AdaBoost is an extension of Margin-pruning Boost. It introduces a margin distribution to penalize the misclassification of small-margin instances. Before introducing Penalized AdaBoost, we first explain the classification margins. The classification margin of an instance $x_i$ shows the difference between prediction confidence of weak hypotheses providing correct classification and that of weak hypotheses leading to misclassification [7]. It is in the range $[-1, 1]$, and the instance $x_i$ is correctly classified if and only if its margin is positive [7]. Therefore, the margin of instance $x_i$ is defined as [10]

$$\text{Margin}_T(x_i) = \frac{\sum_{t=1}^{T} y_i f_t(x_i)}{\sum_{t=1}^{T} \left|f_t(x_i)\right|}, \tag{12}$$

where $f_t(x_i)$ denotes the weak hypothesis for instance $x_i$ at round $t$ and $T$ is the number of total iterations. Next we explain Penalized AdaBoost by the following algorithm.

*Algorithm 6* (Penalized AdaBoost). (1) Set the initial weights $w_{i,1} = 1/N$, where $i = 1, 2, \ldots, N$.

(2) Do the following tasks for $t = 1, 2, \ldots, T$.

(a) Train a weak classifier based on the weighted instances and then calculate $W_{t+}^j$ and $W_{t-}^j$ for each partition $S_t^j$ the same as in Step (2)(a) of Algorithm 1.

(b) Calculate a margin feedback factor $m_{i,t}$ as shown in

$$m_{i,t} = \frac{\exp\left(-\text{Margin}_{t-1}(x_i)\right)}{D_t}, \tag{13}$$

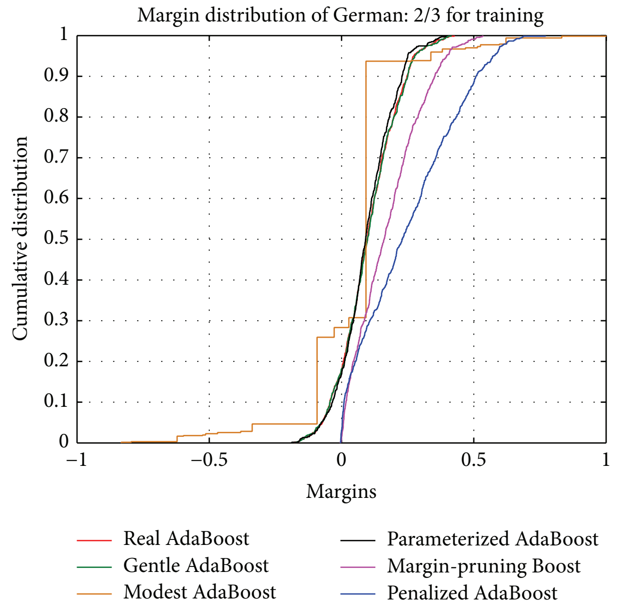


FIGURE 2: Margin distributions using CART-1 at iteration 200.

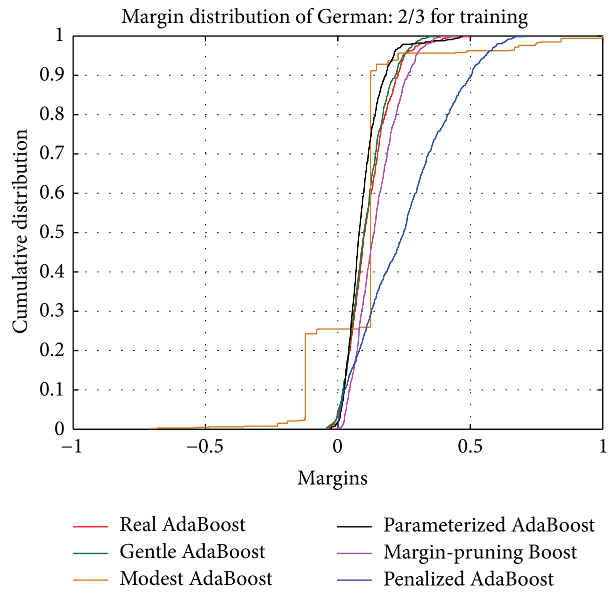


FIGURE 4: Margin distributions using CART-2 at iteration 200.

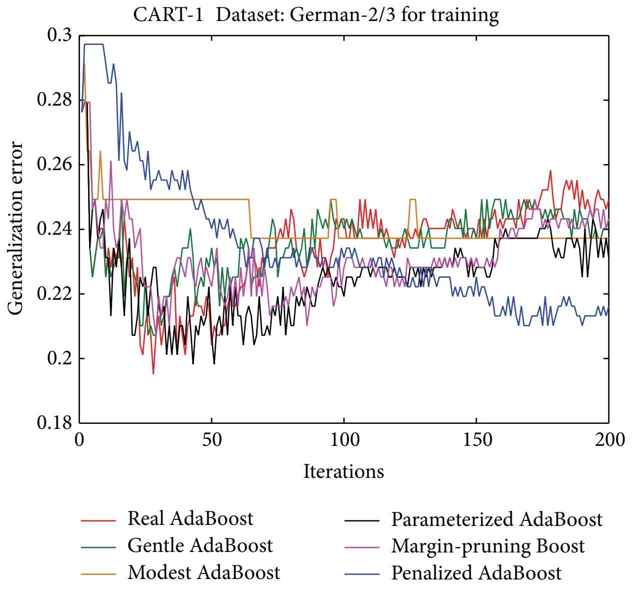


FIGURE 3: Generalization errors using CART-1.

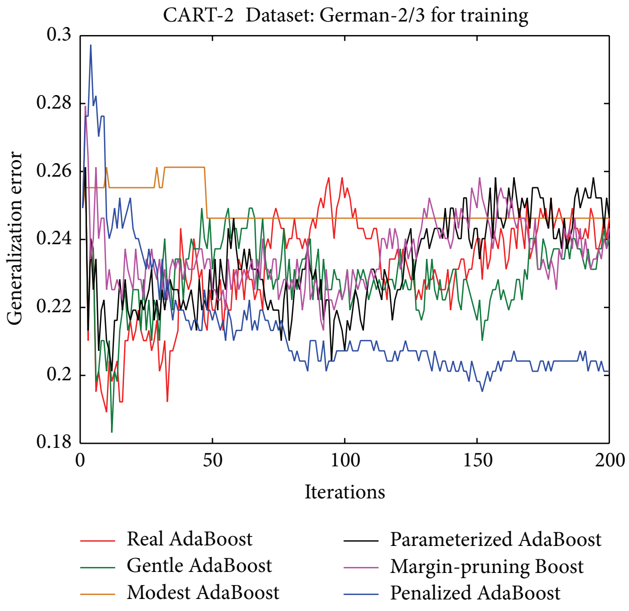


FIGURE 5: Generalization errors using CART-2.

where $D_t$ equals $\sum_i \exp(-\text{Margin}_{t-1}(x_i))$. Then compute $M_{t+}^j$ and $M_{t-}^j$ of every partition $S_t^j$ by

$$M_{t+}^j = \sum_{i:x_i \in S_t^j \wedge y_i = 1} m_{i,t},$$
$$M_{t-}^j = \sum_{i:x_i \in S_t^j \wedge y_i = -1} m_{i,t}. \tag{14}$$

(c) Compute the weak hypothesis for each $S_t^j$ as

$$f_t^j(x) = \begin{cases} \left(W_{t+}^j - W_{t-}^j\right)\left(1 - M_{t-}^j\right), & \text{if } W_{t+}^j > W_{t-}^j \\ \left(W_{t+}^j - W_{t-}^j\right)\left(1 - M_{t+}^j\right), & \text{otherwise.} \end{cases} \tag{15}$$

For each instance $x_i$, set its weak hypothesis $f_t(x_i)$ to be $f_t^j(x)$, where $j$ is the index of partition which $x_i$ falls into.

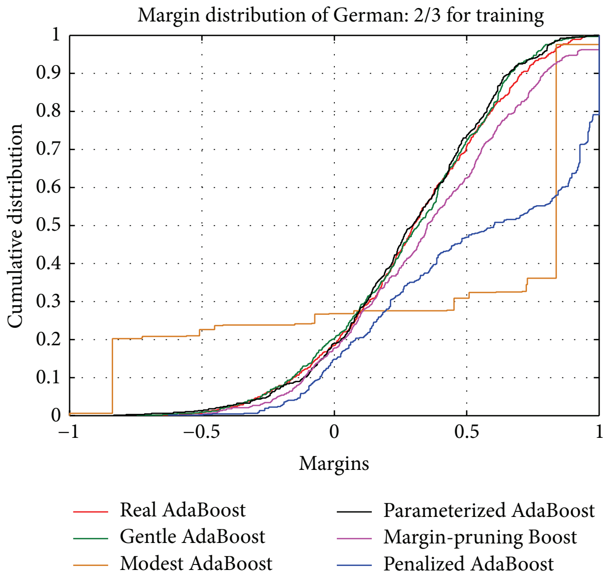


FIGURE 6: Margin distributions using CART-3 at iteration 10.

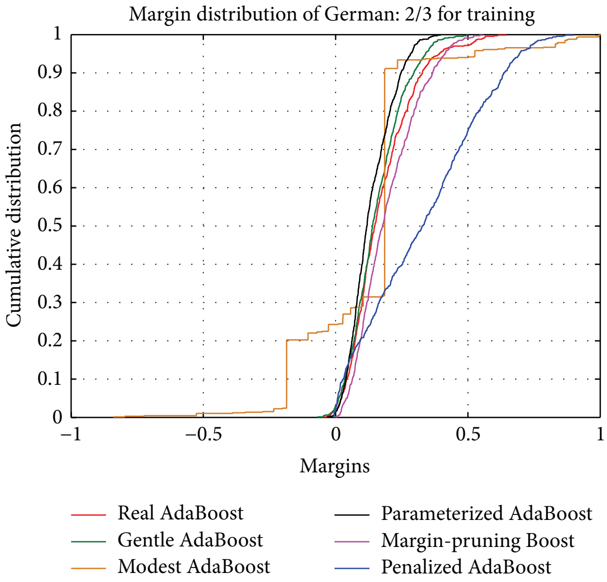


FIGURE 7: Margin distributions using CART-3 at iteration 100.

(d) Update all instance weights by

$$w_{i,t+1} = \exp\left[-y_i \sum_t f_t(x_i)\right]. \quad (16)$$

(e) For each instance $x_i$, if $w_{i,t+1} > Q_{t+1}$ and $\text{Margin}_t(x_i) < 0$, reinitialize instance $x_i$ as follows:

$$w_{i,t+1} = 1, \qquad \sum_t f_t(x_i) = 0. \quad (17)$$

$$Q_{t+1} = \max_i \{w_{i,t+1}\} - \frac{\max_i \{w_{i,t+1}\} - \min_i \{w_{i,t+1}\}}{\beta}. \quad (18)$$

Then normalize $w_{i,t+1}$ by letting $\sum_i w_{i,t+1} = 1$.

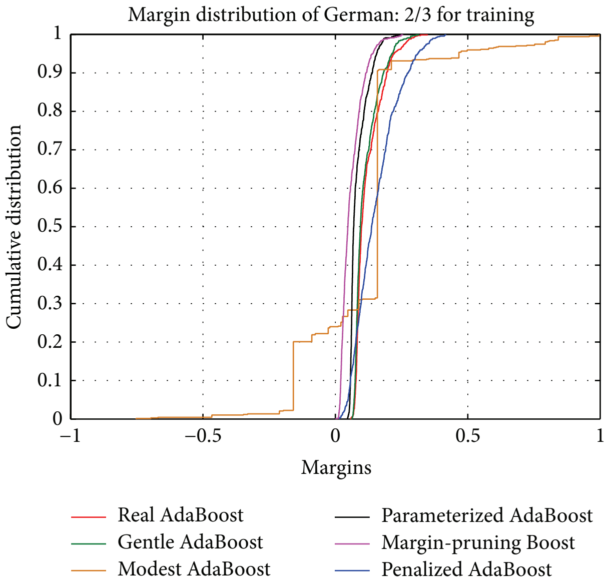


FIGURE 8: Margin distributions using CART-3 at iteration 1000.

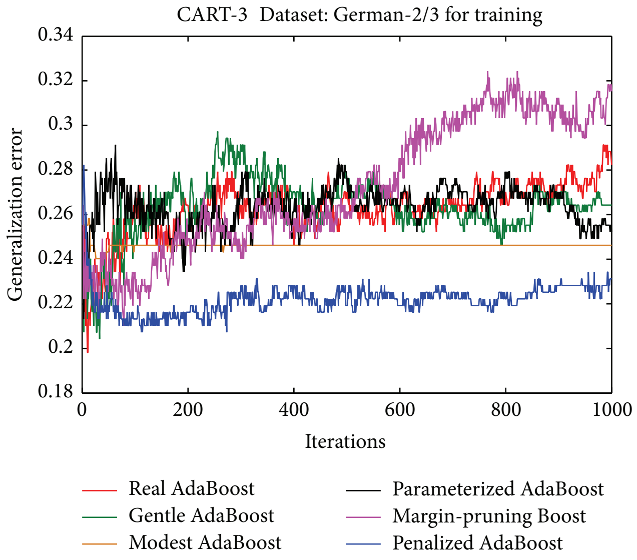
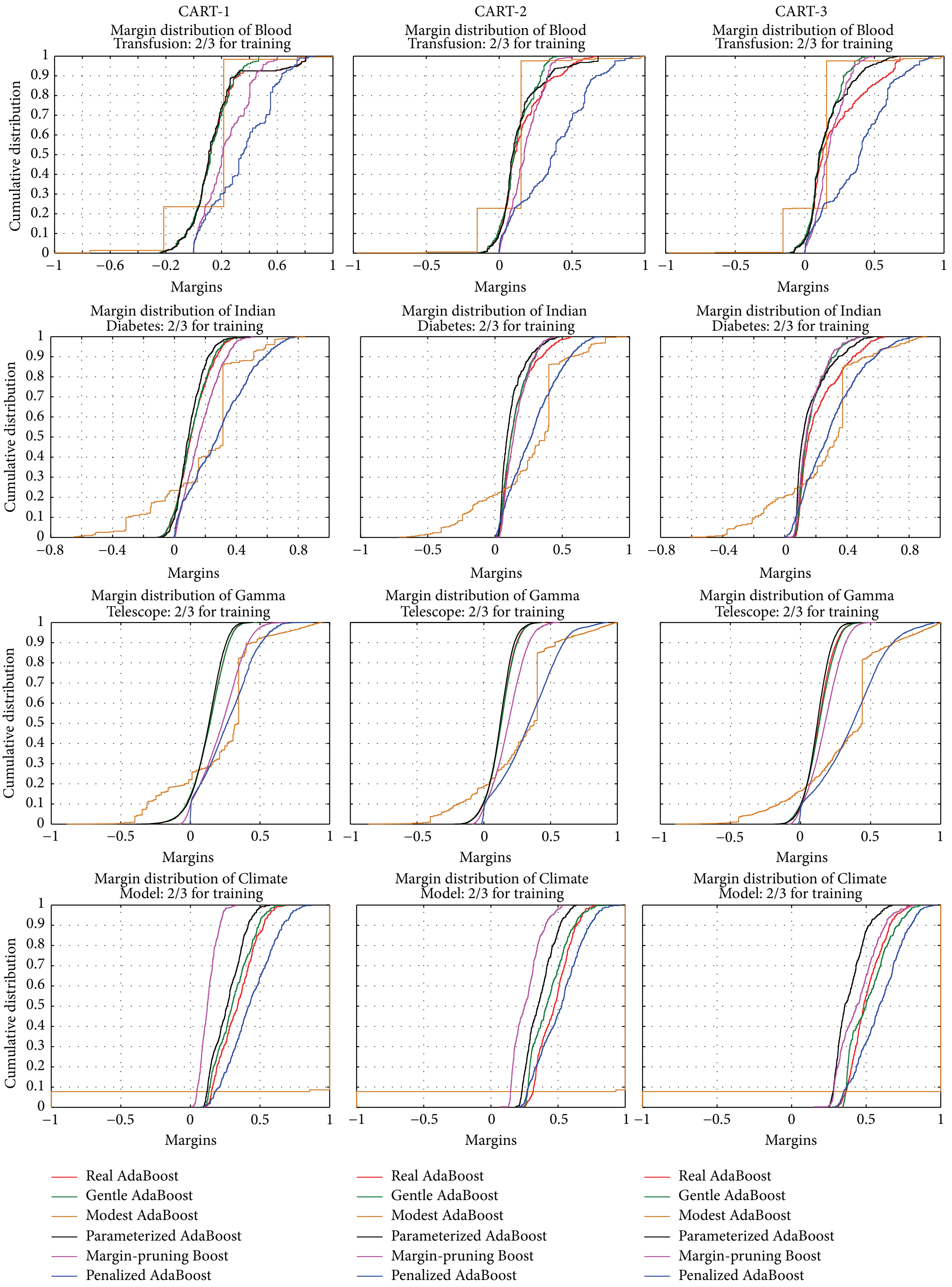


FIGURE 9: Generalization errors using CART-3.

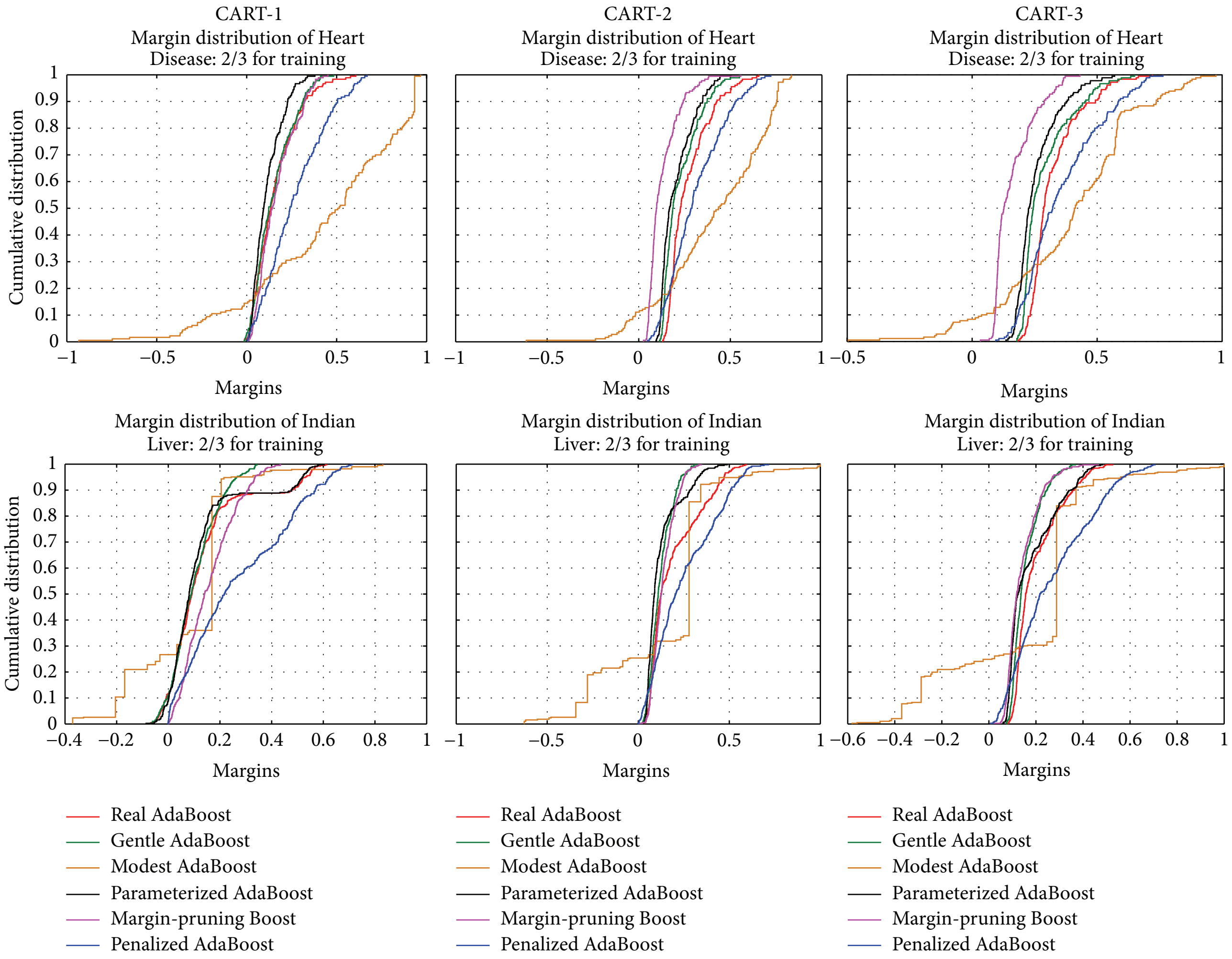
(3) Set $F_T(x) = \sum_{t=1}^{T} f_t(x)$ and output the strong classifier $H(x) = \text{sign}[F_T(x)]$.

Penalized AdaBoost calculates its weak hypotheses by introducing a margin feedback factor in Steps (2)(b) and (2)(c). Moreover, it improves the thresholding of Margin-pruning Boost in Step (2)(e). The parameter $\beta$ in (18) is tuned by the following steps: first the classification performances on 5 data sets with different values of $\beta$ ($\beta = 10$; 30; 50; 70; 90) are evaluated, and then the value with the best performance



(a)

Figure 10: Continued.



(b)

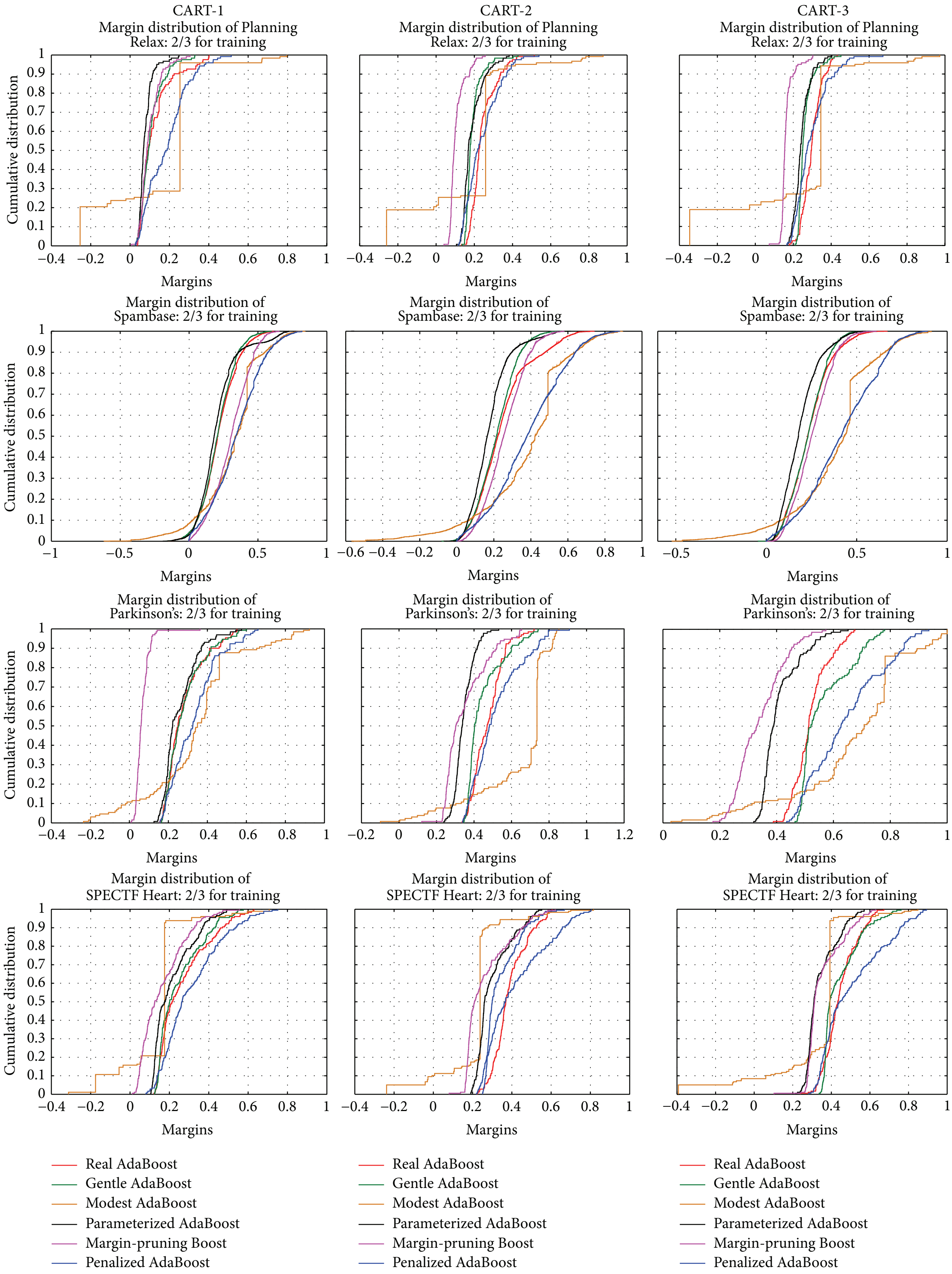
Figure 10: Margin distributions at iteration 200.

is assigned to $\beta$ [15]. Then we will analyze the generalization abilities of the six variants in the next section.

*2.3. Generalization Ability Analysis.* In this section, we analyze the generalization abilities of the six AdaBoost variants by comparing their weak hypotheses and weight updating policies.

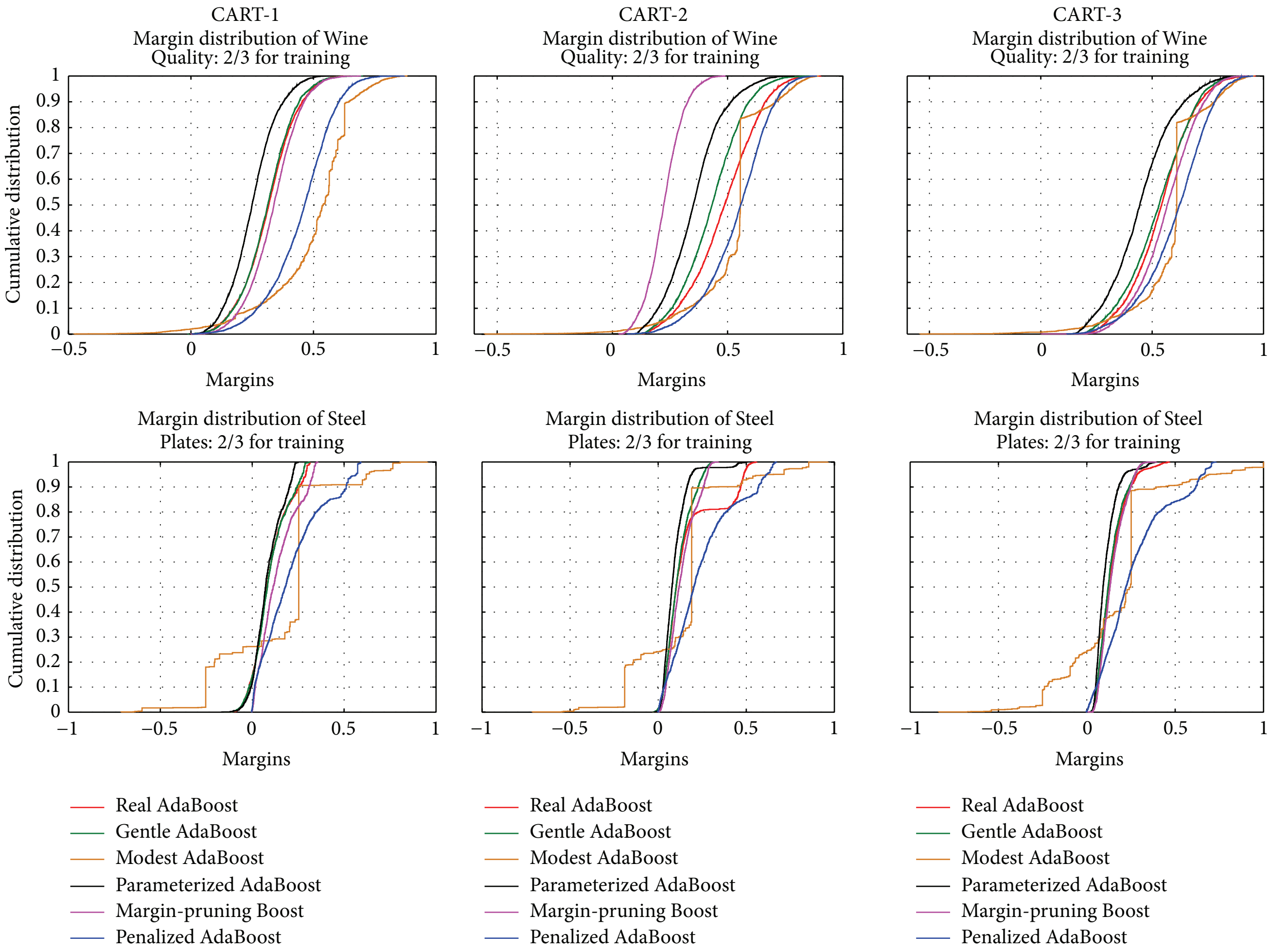
*2.3.1. Real and Gentle AdaBoost.* The difference between Real and Gentle AdaBoost is how they calculate their weak hypotheses. Real AdaBoost computes the weak hypothesis by minimizing the upper bound of training error in each loop [8]. Gentle AdaBoost calculates its weak hypothesis by optimizing the weighted least square error iteratively [11]. Real AdaBoost tries to decrease the training error whereas Gentle AdaBoost aims at reducing the variance of its weak hypotheses. Thus, in most cases, Real AdaBoost converges faster than Gentle AdaBoost in training, but Gentle AdaBoost is more stable than Real AdaBoost with respect to the generalization error.

*2.3.2. Real and Parameterized AdaBoost.* Comparing Step (2)(c) in Real AdaBoost with that in Parameterized AdaBoost, we find that the weight updating policies of the two variants are different. From (12), we know that the training error converges to 0 if and only if the margins of all training instances are increased to be positive. In Step (2)(c) of Real AdaBoost, instances with small margins obtain more weights so that they are more likely correctly classified in future iterations. However, as Real AdaBoost focuses on instances with small margins, it may lead to an increase of instances whose margins are near 0; that is, these instances change the sign of their margins back and forth in the boosting process. Here we call them "swinging instances." These swinging instances slow the convergence of training error. To decrease the number of swinging instances, Parameterized AdaBoost introduces



(a)

Figure 11: Continued.



(b)

Figure 11: Margin distributions at iteration 200.

parameter $\alpha$ to give them larger weights. From (8), we can see that $\alpha\times|y_iF_t(x_i)|$ reduces more weight for instances whose margins are far from 0 and less weight for swinging instances. Therefore, swinging instances obtain more attention than nonswinging ones. Parameterized AdaBoost aims at reducing the number of swinging instances by trying to correctly classify them in the early training phase. Thus, Parameterized AdaBoost can converge faster than Real AdaBoost in training. With respect to the generalization error, Parameterized AdaBoost is more prone to overfitting than Real AdaBoost especially when the CART used as weak classifiers has more inner nodes. The reason is that it focuses more on swinging instances at the cost of focusing less on instances with minimal margins. However, it can perform similarly to or slightly better than Real AdaBoost if it uses simple weak classifiers such as stump decision trees because stump trees are more resistant to overfitting than CART with many inner nodes.

*2.3.3. Gentle and Modest AdaBoost.* Modest AdaBoost utilizes an inverted weight distribution to highlight weak hypotheses that can correctly classify instances with small margins. Nevertheless, the performance of Modest AdaBoost is not stable. Here we give an example to explain the reason. We suppose $W^j_{t+} > W^j_{t-}$; if $W^j_{t+}(1-\overline{W}^j_{t+})$ is also larger than $W^j_{t-}(1-\overline{W}^j_{t-})$, the factor $(1-\overline{W}^j_{t+})$ assigns higher prediction confidence to weak hypotheses that correctly classify small-margin instances. At the same time, the factor $(1-\overline{W}^j_{t-})$ reduces the prediction confidence for weak hypotheses misclassifying small-margin instances. In this case, Modest AdaBoost outperforms Gentle AdaBoost. However, if $W^j_{t+}(1-\overline{W}^j_{t+})$ is smaller than $W^j_{t-}(1-\overline{W}^j_{t-})$ in the case $W^j_{t+} > W^j_{t-}$, the sign of the weak hypothesis will be negative. This means the factor $(1-\overline{W}^j_{t-})$ reduces the prediction confidence for weak hypotheses that correctly classify small-margin instances. Meanwhile, $(1-\overline{W}^j_{t+})$ increases the prediction confidence of weak hypotheses misclassifying small-margin instances. In this case, Modest AdaBoost performs far worse than Gentle AdaBoost.

Table 1: Test results at iteration 200 using CART-1.

| Data sets: 25 | RAB | GAB | MAB | PAAB | MPB | PAB |
|---|---|---|---|---|---|---|
| Australian | 0.1536 | 0.1435 | 0.1435 | 0.1449 | 0.1391 | **0.1304** |
| Blood Transfusion | 0.2392 | 0.2312 | 0.2352 | 0.2392 | **0.2218** | 0.2298 |
| Banana | 0.2732 | 0.2713 | 0.2804 | 0.2672 | **0.2526** | 0.2809 |
| Banknote | 0.0029 | 0.0022 | 0.0372 | **0.0015** | 0.0058 | 0.0080 |
| Breast Cancer | 0.0281 | **0.0246** | 0.0422 | 0.0299 | 0.0669 | 0.0281 |
| Climate model | 0.0704 | **0.0593** | 0.0963 | 0.0759 | 0.0704 | 0.0630 |
| Gamma Telescope | 0.1525 | 0.1546 | 0.2297 | 0.1489 | **0.1454** | 0.1548 |
| German | 0.2560 | 0.2580 | 0.2950 | 0.2490 | 0.2470 | **0.2450** |
| Heart Disease | 0.2259 | 0.2333 | 0.1704 | 0.2185 | 0.2000 | **0.1630** |
| Hepatitis | 0.2200 | 0.2130 | 0.2515 | 0.2065 | 0.2129 | **0.1869** |
| Indian Diabetes | 0.2448 | 0.2578 | 0.2383 | 0.2539 | 0.2318 | **0.2253** |
| Indian Liver | 0.3074 | 0.3005 | 0.3230 | 0.3092 | **0.2850** | 0.3126 |
| Image Segment | 0.0061 | **0.0045** | 0.0076 | 0.0061 | 0.0106 | 0.0061 |
| Parkinson's | **0.0872** | **0.0872** | 0.1692 | 0.0923 | 0.0974 | 0.0923 |
| Planning Relax | 0.4286 | 0.4012 | 0.3519 | 0.4448 | 0.4175 | **0.3186** |
| Ring Norm | 0.0269 | **0.0268** | 0.0470 | 0.0293 | 0.0277 | 0.0509 |
| Sonar | **0.1445** | 0.1589 | 0.1637 | 0.1540 | 0.2117 | 0.1734 |
| Spambase | 0.0580 | 0.0546 | 0.0895 | 0.0554 | **0.0528** | 0.0617 |
| SPECTF Heart | 0.2097 | 0.2285 | 0.2472 | 0.2172 | 0.2584 | **0.1985** |
| Splice | 0.0675 | 0.0675 | 0.0919 | 0.0685 | **0.0612** | 0.0639 |
| Steel Plates | **0.2277** | 0.2324 | 0.2751 | 0.2344 | 0.2318 | 0.2421 |
| Twonorm | 0.0303 | 0.0305 | 0.0315 | 0.0324 | **0.0292** | 0.0304 |
| Waveform | 0.0993 | 0.0975 | 0.0962 | 0.0996 | 0.0917 | **0.0853** |
| WBPC | 0.2475 | 0.2727 | 0.3030 | 0.2879 | 0.3081 | **0.2323** |
| Wine Quality | 0.0055 | 0.0054 | 0.0240 | 0.0062 | **0.0052** | 0.0054 |
| Sum | 3.8128 | 3.8170 | 4.2405 | 3.8727 | 3.8820 | **3.5887** |
| VS.GAB | −0.0042 | 0.0000 | 0.4235 | 0.0557 | 0.0650 | **−0.2283** |
| No.Best | 3 | 5 | 0 | 1 | 8 | **9** |
| No.To.GAB | 11 | — | 5 | 9 | **15** | 13 |

*2.3.4. Gentle and Margin-Pruning Boost.* At each round of Gentle AdaBoost, the weights of misclassified instances are increased whereas the weights of correctly classified instances are decreased. This will lead to the phenomenon that the weights of difficult-to-classify instances are increased too large. If these instances are noise data or outliers, the performance of the final strong classifier will be degraded. To solve this problem, Margin-pruning Boost utilizes a threshold to filter instances whose weights are too large and then resets their weights to be 1.

Margin-pruning Boost reduces the influence from noise-like instances effectively in the early training phase by restraining the weight increase of filtered instances [14]. However, as the number of iterations increases, the weights of instances filtered by thresholding become smaller and smaller. In the late training phase, the weights of these filtered instances are probably reduced smaller than 1. In that case, resetting their weights to be 1 actually increases the influence of these instances. Thus, the performance of Margin-pruning Boost drops when the number of loops increases.

*2.3.5. Margin-Pruning Boost and Penalized AdaBoost.* Penalized AdaBoost is an improvement of Margin-pruning Boost. First it introduces a margin feedback factor to assign higher prediction confidence to weak hypotheses that can correctly classify small-margin instances. From (13), (14), and (15), we can see that $(1 - M_{t-}^{j})$ and $(1 - M_{t+}^{j})$ are proportional to the margins, and they are computed from the sum of margin feedback factors of misclassified instances. This means that misclassifying small-margin instances will lead to small $(1 - M_{t-}^{j})$ and $(1 - M_{t+}^{j})$. Therefore, the prediction confidence of weak hypotheses which misclassify small-margin instances will be degraded. Compared with Gentle AdaBoost and Margin pruning Boost, Penalized AdaBoost can stand out more competent weak hypotheses. Therefore, it is more robust than the other two variants. Modest AdaBoost highlights more competent weak hypotheses in some cases but downplays these weak hypotheses in other cases. By contrast, Penalized AdaBoost attaches importance to these more competent weak hypotheses under any circumstance. Thus it is more stable than Modest AdaBoost.

Table 2: Test results at iteration 500 using CART-1.

| Data sets: 25 | RAB | GAB | MAB | PAAB | MPB | PAB |
|---|---|---|---|---|---|---|
| Australian | 0.1681 | 0.1638 | 0.1435 | 0.1638 | 0.1377 | **0.1333** |
| Blood Transfusion | 0.2446 | 0.2540 | 0.2352 | 0.2500 | 0.2272 | **0.2258** |
| Banana | 0.2734 | 0.2725 | 0.2804 | 0.2662 | **0.2547** | 0.2738 |
| Banknote | 0.0029 | 0.0036 | 0.0372 | **0.0022** | 0.0058 | 0.0058 |
| Breast Cancer | 0.0281 | **0.0246** | 0.0369 | 0.0264 | 0.0756 | **0.0246** |
| Climate Model | 0.0704 | 0.0648 | 0.0963 | 0.0741 | 0.0722 | **0.0593** |
| Gamma Telescope | 0.1522 | 0.1519 | 0.2292 | 0.1479 | **0.1424** | 0.1473 |
| German | 0.2700 | 0.2730 | 0.2950 | 0.2710 | 0.2490 | **0.2430** |
| Heart Disease | 0.2481 | 0.2481 | **0.1704** | 0.2333 | 0.2000 | 0.1852 |
| Hepatitis | 0.2262 | 0.2130 | 0.2515 | 0.2132 | 0.2129 | **0.2128** |
| Indian Diabetes | 0.2487 | 0.2526 | 0.2383 | 0.2565 | 0.2409 | **0.2266** |
| Indian Liver | 0.3005 | 0.3074 | 0.3558 | 0.2971 | **0.2953** | 0.3057 |
| Image Segment | 0.0045 | 0.0045 | 0.0045 | **0.0030** | 0.0061 | **0.0030** |
| Parkinson’s | **0.0821** | 0.0974 | 0.1641 | 0.0923 | 0.0974 | 0.0872 |
| Planning Relax | 0.4342 | 0.4397 | 0.3519 | 0.4064 | 0.4175 | **0.3462** |
| Ring Norm | 0.0264 | **0.0249** | 0.0309 | 0.0280 | 0.0251 | 0.0318 |
| Sonar | **0.1446** | 0.1638 | 0.1589 | 0.1684 | 0.2548 | 0.1734 |
| Spambase | 0.0593 | 0.0574 | 0.0895 | 0.0578 | **0.0517** | 0.0572 |
| SPECTF Heart | 0.2097 | 0.2285 | 0.2584 | 0.2060 | 0.2772 | **0.2022** |
| Splice | 0.0726 | 0.0746 | 0.0919 | 0.0752 | 0.0622 | **0.0605** |
| Steel Plates | 0.2318 | 0.2293 | 0.2777 | 0.2339 | **0.2267** | 0.2334 |
| Twonorm | 0.0291 | 0.0296 | 0.0286 | 0.0326 | 0.0296 | **0.0276** |
| Waveform | 0.0990 | 0.0984 | 0.0962 | 0.1038 | 0.0950 | **0.0881** |
| WBPC | 0.2475 | 0.2828 | 0.3030 | 0.2424 | 0.3081 | **0.2374** |
| Wine Quality | 0.0060 | 0.0054 | 0.0231 | 0.0054 | 0.0060 | **0.0051** |
| Sum | 3.8800 | 3.9656 | 4.2484 | 3.8569 | 3.9711 | **3.5963** |
| VS.GAB | −0.0856 | 0.0000 | 0.2828 | −0.1087 | 0.0055 | **−0.3693** |
| No.Best | 2 | 2 | 1 | 2 | 5 | **15** |
| No.To.GAB | 14 | — | 9 | 14 | 16 | **20** |

Furthermore, Penalized AdaBoost solves the problem of Margin-pruning Boost by utilizing a more adaptive thresholding method. Penalized AdaBoost also uses the thresholding to filter the large-weight instances similarly to Margin-pruning Boost. However, it only resets the weights of filtered instances with negative margins. This technique guarantees that the reset weights are always smaller than the original ones. For these noise-like instances, Penalized AdaBoost does not completely exclude them because they are not definitely noise. Nevertheless, Penalized AdaBoost keeps their weights small to reduce their influence on the final strong classifier. Thereby, it has better generalization ability than Margin-pruning Boost.

*2.4. Margin Distribution Comparison.* In this section, we compare the generalization abilities of the six different variants by analyzing their margin distributions. Li and Shen showed that reducing the minimal margin of training data plays little role in improving the generalization ability [18]. However, enlarging the whole margin distribution to obtain a balance between the training error and complexity is crucial to the generalization ability [18]. Here we use three kinds of CART as weak classifiers to evaluate the six AdaBoost variants. The three kinds of CART are CART-1 (CART with one inner node), CART-2 (CART with two inner nodes), and CART-3 (CART with three inner nodes). To get the cumulative margin distributions, for each data set, we use 2/3 of its data to train the final strong classifiers. Figure 2 shows the cumulative margin distributions based on CART-1 in data set German at iteration 200. Figure 3 shows the generalization errors of the same data set with respect to Figure 2. In Figure 2, Penalized AdaBoost enlarges the whole margin distribution more than the other variants so that it achieves the best generalization error in Figure 3. Real AdaBoost, Gentle AdaBoost, Parameterized AdaBoost, and Margin-pruning Boost perform similarly on the margins. So their generalization errors are also similar when the number of iterations reaches to 200. The margin curve of Modest AdaBoost in Figure 2 is not smooth. That may explain why its generalization error in Figure 3 is not changed gradually.

Margin distributions based on CART-2 in the same data set at iteration 200 are shown in Figure 4. Figure 5 shows the generalization errors with respect to Figure 4. From Figures 4 and 5, we notice that the generalization abilities of the

Table 3: Test results at iteration 800 using CART-1.

| Data sets: 25 | RAB | GAB | MAB | PAAB | MPB | PAB |
|---|---|---|---|---|---|---|
| Australian | 0.1754 | 0.1696 | 0.1435 | 0.1899 | 0.1362 | **0.1348** |
| Blood Transfusion | 0.2487 | 0.2473 | 0.2379 | 0.2527 | 0.2272 | **0.2218** |
| Banana | 0.2740 | 0.2753 | 0.2804 | 0.2687 | **0.2530** | 0.2709 |
| Banknote | 0.0036 | 0.0044 | 0.0372 | **0.0022** | 0.0058 | 0.0036 |
| Breast Cancer | **0.0246** | 0.0281 | 0.0369 | **0.0246** | 0.0774 | **0.0246** |
| Climate Model | 0.0759 | 0.0685 | 0.0963 | 0.0722 | 0.0722 | **0.0611** |
| Gamma Telescope | 0.1530 | 0.1531 | 0.2290 | 0.1485 | **0.1425** | 0.1448 |
| German | 0.2800 | 0.2770 | 0.2950 | 0.2780 | 0.2550 | **0.2430** |
| Heart Disease | 0.2519 | 0.2333 | **0.1704** | 0.2519 | 0.2000 | 0.2111 |
| Hepatitis | 0.2197 | 0.2195 | 0.2646 | 0.2068 | 0.2129 | **0.2000** |
| Indian Diabetes | 0.2513 | 0.2513 | 0.2396 | 0.2591 | 0.2396 | **0.2305** |
| Indian Liver | 0.2971 | 0.3005 | 0.3592 | 0.2988 | 0.3022 | **0.2936** |
| Image Segment | 0.0045 | 0.0045 | 0.0045 | **0.0030** | 0.0061 | **0.0030** |
| Parkinson’s | 0.0872 | 0.1026 | 0.1641 | **0.0821** | 0.0974 | 0.0923 |
| Planning Relax | 0.4669 | 0.4287 | **0.3519** | 0.4230 | 0.4175 | 0.3737 |
| Ring Norm | 0.0258 | **0.0235** | 0.0273 | 0.0288 | 0.0262 | 0.0288 |
| Sonar | **0.1493** | 0.1542 | 0.1638 | 0.1732 | 0.2500 | 0.1685 |
| Spambase | 0.0593 | 0.0563 | 0.0880 | 0.0572 | **0.0517** | 0.0561 |
| SPECTF Heart | 0.2172 | 0.2247 | 0.2697 | **0.2060** | 0.2734 | 0.2135 |
| Splice | 0.0766 | 0.0772 | 0.0919 | 0.0772 | 0.0642 | **0.0598** |
| Steel Plates | 0.2303 | 0.2303 | 0.2736 | 0.2349 | **0.2262** | 0.2318 |
| Twonorm | 0.0301 | 0.0300 | 0.0286 | 0.0315 | 0.0309 | **0.0269** |
| Waveform | 0.0987 | 0.0981 | 0.0962 | 0.1065 | 0.0956 | **0.0902** |
| WBPC | 0.2424 | 0.2828 | 0.2980 | 0.2475 | 0.3081 | **0.2273** |
| Wine Quality | 0.0057 | 0.0058 | 0.0235 | 0.0055 | 0.0072 | **0.0046** |
| Sum | 3.9492 | 3.9466 | 4.2711 | 3.9298 | 3.9785 | **3.6163** |
| VS.GAB | 0.0026 | 0.0000 | 0.3245 | −0.0168 | 0.0319 | **−0.3303** |
| No.Best | 2 | 1 | 2 | 5 | 4 | **14** |
| No.To.GAB | 14 | — | 8 | 13 | 14 | **22** |

six variants are consistent with their performance on the margins. We also evaluate margin distributions using CART-3 on the data set German. Margin curves at iterations 10, 100, and 1000 are shown by Figures 6, 7, and 8, respectively. Comparing the three figures, we find that the margins are enlarged gradually as the number of iterations increases. Furthermore, we notice that Margin-pruning Boost outperforms Gentle AdaBoost in Figures 6 and 7. However it performs worse than Gentle AdaBoost in Figure 8. This demonstrates that the performance of Margin-pruning Boost drops as the number of iterations increases. Differently from Margin-pruning Boost, Penalized AdaBoost outperforms others at most cases. Thus it is most robust and stable. Figure 9 shows the generalization errors of the six variants which use CART-3 as their weak classifiers. In Figure 9, Margin-pruning Boost obtains lower generalization errors than Gentle AdaBoost before iteration 500. Unfortunately, it leads to severe overfitting after iteration 500.

Figures 10 and 11 show margin distributions of other data sets. From these margin curves, we can conclude that Penalized AdaBoost generally outperforms the other five variants on enlarging the whole margin distributions. Real and Gentle AdaBoost perform very similarly, and Parameterized AdaBoost is slightly worse than Real AdaBoost when it uses CART-2 and CART-3. Margin-pruning Boost is better than Gentle AdaBoost if the number of iterations is small, While the margin curves of Modest AdaBoost are not smooth, they may lead to an unstable performance on generalization errors.

## 3. Experiments

In this section, we compare the six AdaBoost variants using 25 binary classification data sets from UCI [55]. For every data set, we used Matlab AdaBoost Toolbox [54] and 3-fold cross validation. First we measure the generalization errors (estimated by the classification error on the test set) of the six variants based on CART-1. Table 1 summarizes the results of the six variants using CART-1 at iteration 200. Tables 2 and 3 show their generalization errors using CART-1 at iterations 500 and 800, respectively. We also compare the generalization errors of the six variants based on CART-2 and CART-3. Table 4 shows the comparison results using CART-2, and Table 5 compares the six variants using CART-3.

Table 4: Test results at iteration 200 using CART-2.

| Data sets: 25 | RAB | GAB | MAB | PAAB | MPB | PAB |
|---|---|---|---|---|---|---|
| Australian | 0.1449 | 0.1478 | 0.1420 | 0.1580 | 0.1420 | **0.1246** |
| Blood Transfusion | 0.2581 | 0.2540 | 0.2366 | 0.2608 | 0.2419 | **0.2204** |
| Banana | 0.1021 | **0.1002** | 0.2430 | 0.1006 | 0.1045 | 0.1479 |
| Banknote | 0.0051 | **0.0022** | 0.0058 | **0.0022** | 0.0044 | 0.0029 |
| Breast Cancer | 0.0264 | **0.0229** | 0.0387 | 0.0299 | 0.0316 | 0.0246 |
| Climate Model | 0.0574 | 0.0574 | 0.0944 | 0.0593 | 0.1019 | **0.0519** |
| Gamma Telescope | 0.1301 | 0.1317 | 0.2006 | **0.1295** | 0.1320 | 0.1426 |
| German | 0.2710 | 0.2690 | 0.2890 | 0.2850 | 0.2580 | **0.2380** |
| Heart Disease | 0.2667 | **0.2000** | 0.2074 | 0.2556 | 0.2889 | 0.2037 |
| Hepatitis | 0.1934 | 0.2001 | 0.2320 | **0.1675** | 0.2709 | 0.1934 |
| Indian Diabetes | 0.2591 | 0.2682 | **0.2357** | 0.2839 | 0.2500 | 0.2370 |
| Indian Liver | 0.2953 | 0.2815 | 0.3592 | 0.3005 | **0.2694** | 0.3057 |
| Image Segment | 0.0091 | 0.0061 | **0.0045** | **0.0045** | 0.0091 | **0.0045** |
| Parkinson's | 0.0667 | **0.0564** | 0.1385 | 0.0667 | 0.0974 | 0.0769 |
| Planning Relax | 0.4117 | 0.3791 | **0.3352** | 0.3731 | 0.4392 | 0.3573 |
| Ring Norm | 0.0288 | 0.0284 | 0.0312 | 0.0311 | **0.0276** | 0.0343 |
| Sonar | 0.1396 | 0.1397 | 0.1493 | 0.1443 | **0.1395** | 0.1491 |
| Spambase | 0.0511 | 0.0506 | 0.0798 | 0.0550 | **0.0476** | 0.0546 |
| SPECTF Heart | **0.2060** | 0.2135 | 0.2472 | **0.2060** | 0.2434 | 0.2135 |
| Splice | 0.0431 | 0.0408 | **0.0368** | 0.0481 | 0.0408 | 0.0381 |
| Steel Plates | 0.2102 | 0.2020 | 0.2694 | 0.2184 | **0.1999** | 0.2123 |
| Twonorm | 0.0300 | 0.0303 | 0.0297 | 0.0342 | 0.0288 | **0.0270** |
| Waveform | 0.0908 | 0.1008 | 0.0920 | 0.1032 | 0.0935 | **0.0860** |
| WBPC | 0.2424 | 0.2424 | 0.3030 | 0.2374 | 0.2525 | **0.2222** |
| Wine Quality | 0.0051 | **0.0038** | 0.0149 | 0.0042 | 0.0060 | 0.0045 |
| Sum | 3.5442 | 3.4289 | 4.0159 | 3.5590 | 3.7208 | **3.3730** |
| VS.GAB | 0.1153 | 0.0000 | 0.5870 | 0.1301 | 0.2919 | **−0.0559** |
| No.Best | 1 | 6 | 4 | 5 | 5 | **8** |
| No.To.GAB | 10 | — | 8 | 7 | 12 | **13** |

In Tables 1, 2, 3, 4, and 5, RAB, GAB, MAB, PAAB, MPB, and PAB denote Real AdaBoost, Gentle AdaBoost, Modest AdaBoost, Parameterized AdaBoost, Margin-pruning Boost, and Penalized AdaBoost separately. The row of VS.GAB shows the residues that the sum of generalization errors of other variants subtracts that of Gentle AdaBoost. Here the bold values show the best performance and No.Best means the number of best generalization errors. No.To.GAB denotes the number of data sets in which Gentle AdaBoost is outperformed by others. From Tables 1, 2, and 3, we can conclude that Real, Gentle, and Parameterized AdaBoost perform similarly when using CART-1 as weak classifiers. Modest AdaBoost performs worse than other variants at most cases. Moreover, its error rates are rarely changed even when the number of loops increases. Comparing No.Best of Margin-pruning Boost in Tables 1, 2, and 3, we find that its performance drops as the number of iterations increases. We can also see that Penalized AdaBoost generally outperforms other variants from VS.GAB, No.Best, and No.To.GAB in Tables 1, 2, and 3. Comparing Tables 1, 4, and 5, we can conclude that increasing the inner nodes of CART is important to reduce the generalization errors. In Tables 4 and 5, we can see that Gentle AdaBoost is slightly better than Real AdaBoost. However, the performance of Parameterized AdaBoost and Margin-pruning Boost drops sharply. This means the two variants are more suitable for CART-1. On the other hand, Modest AdaBoost using CART-2 or CART-3 performs better than that using CART-1. This suggests that Modest AdaBoost is suitable for CART with more inner nodes. From all tables, we notice that the performance of Gentle and Penalized AdaBoost is not degraded neither by the number of inner nodes in CART nor by the number of iterations. Nevertheless, Penalized AdaBoost shows stronger robustness when compared with Gentle AdaBoost.

## 4. Conclusion

This paper analyzes the generalization abilities of six AdaBoost variants mathematically. The novel contributions of our work are listed as follows.

(1) There are many comparison studies of AdaBoost variants. However, we compare three new proposed variants (Parameterized AdaBoost, Margin-pruning

Table 5: Test results at iteration 200 using CART-3.

| Data sets: 25 | RAB | GAB | MAB | PAAB | MPB | PAB |
|---|---|---|---|---|---|---|
| Australian | 0.1478 | 0.1536 | 0.1333 | 0.1594 | 0.1522 | **0.1203** |
| Blood Transfusion | 0.2863 | 0.2715 | 0.2392 | 0.2984 | 0.2527 | **0.2245** |
| Banana | 0.1040 | 0.1028 | 0.2142 | 0.1047 | **0.1002** | 0.1132 |
| Banknote | 0.0036 | 0.0022 | 0.0029 | **0.0015** | 0.0051 | 0.0022 |
| Breast Cancer | 0.0334 | 0.0281 | **0.0246** | 0.0281 | 0.0281 | 0.0281 |
| Climate Model | 0.0611 | 0.0519 | 0.0944 | 0.0593 | 0.0722 | **0.0500** |
| Gamma Telescope | 0.1300 | 0.1305 | 0.1770 | 0.1316 | **0.1278** | 0.1360 |
| German | 0.2870 | 0.2660 | 0.2780 | 0.2920 | 0.2750 | **0.2440** |
| Heart Disease | 0.2074 | 0.2185 | 0.2074 | 0.2074 | 0.2593 | **0.2000** |
| Hepatitis | **0.1932** | 0.1933 | 0.2252 | 0.1998 | 0.1934 | 0.2127 |
| Indian Diabetes | 0.2513 | 0.2552 | 0.2357 | 0.2695 | 0.2643 | **0.2344** |
| Indian Liver | **0.2781** | **0.2781** | 0.3437 | 0.3092 | 0.2867 | 0.2971 |
| Image Segment | **0.0015** | **0.0015** | 0.0030 | 0.0030 | 0.0061 | 0.0030 |
| Parkinson's | 0.0718 | **0.0615** | 0.1026 | 0.0821 | 0.0769 | 0.0769 |
| Planning Relax | 0.3514 | 0.3844 | 0.3518 | **0.3459** | 0.3899 | 0.3682 |
| Ring Norm | 0.0277 | **0.0266** | 0.0296 | 0.0295 | 0.0296 | 0.0309 |
| Sonar | **0.1202** | 0.1395 | 0.1637 | 0.1636 | 0.1347 | 0.1349 |
| Spambase | 0.0528 | **0.0454** | 0.0728 | 0.0519 | 0.0476 | 0.0500 |
| SPECTF Heart | 0.2210 | 0.2022 | 0.2472 | **0.1948** | 0.2285 | **0.1948** |
| Splice | 0.0321 | 0.0361 | **0.0294** | 0.0428 | 0.0428 | 0.0308 |
| Steel Plates | 0.2030 | 0.2020 | 0.2653 | 0.2138 | 0.2035 | **0.1999** |
| Twonorm | 0.0319 | 0.0301 | 0.0288 | 0.0299 | 0.0304 | **0.0265** |
| Waveform | 0.1029 | 0.0972 | 0.0890 | 0.1053 | 0.1038 | **0.0872** |
| WBPC | 0.2323 | **0.2172** | 0.3030 | 0.2525 | 0.2273 | **0.2172** |
| Wine Quality | 0.0051 | **0.0038** | 0.0125 | 0.0048 | 0.0042 | **0.0038** |
| Sum | 3.4369 | 3.3992 | 3.8743 | 3.5808 | 3.5423 | **3.2866** |
| VS.GAB | 0.0377 | 0.0000 | 0.4751 | 0.1816 | 0.1431 | **−0.1126** |
| No.Best | 4 | 7 | 2 | 3 | 2 | **12** |
| No.To.GAB | 10 | — | 9 | 6 | 6 | **17** |

Boost, and Penalized AdaBoost) with three traditional variants (Real, Gentle, and Modest AdaBoost). This kind of comparison is new in the machine learning studies.

(2) Differently from conventional comparison works that draw conclusion from experimental results, we analyze the generalization abilities of the six variants by comparing their classification margins.

(3) We design experiments to verify our analyses. The experimental results are consistent with our analyses.

In general, the analyses and comparison in this paper are useful for researchers who want to improve the classification performance by switching to a new AdaBoost variant. In our current research, we focus on two classification problems. In our future work, we want to extend our analyses to multiclassification problems. In addition, we will compare more kinds of weak classifiers such as SVM and ANN to find out which kind of weak classifiers is suitable for which AdaBoost variant.

## Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

## References

[1] J.-M. Guo, C.-C. Lin, M.-F. Wu, C.-H. Chang, and H. Lee, "Complexity reduced face detection using probability-based face mask prefiltering and pixel-based hierarchical-feature adaboosting," *IEEE Signal Processing Letters*, vol. 18, no. 8, pp. 447–450, 2011.

[2] P. Viola and M. Jones, "Rapid object detection using a boosted cascade of simple features," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR '01)*, pp. I-511–I-518, December 2001.

[3] M. Kölsch and M. Turk, "Robust hand detection," in *Proceedings of the 6th IEEE International Conference on Automatic Face and Gesture Recognition (FGR '04)*, pp. 614–619, May 2004.

[4] J. M. Guo, Y. F. Liu, C. H. Chang, and H. S. Nguyen, "Improved hand tracking system," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 22, no. 5, pp. 693–701, 2012.

[5] J. Xu, Q. Wu, J. Zhang, and Z. Tang, "Fast and accurate human detection using a cascade of boosted MS-LBP features," *IEEE Signal Processing Letters*, vol. 19, no. 10, pp. 676–679, 2012.

[6] Y. Freund and R. E. Schapire, "A short introduction to boosting," *Journal of Japanese Society for Artificial Intelligence*, vol. 14, no. 5, pp. 771–780, 1999.

[7] R. E. Schapire, Y. Freund, P. Bartlett, and W. S. Lee, "Boosting the margin: a new explanation for the effectiveness of voting methods," *The Annals of Statistics*, vol. 26, no. 5, pp. 1651–1686, 1998.

[8] R. E. Schapire and Y. Singer, "Improved boosting algorithms using confidence-rated predictions," *Machine Learning*, vol. 37, no. 3, pp. 297–336, 1999.

[9] Z. L. Fu, "Analysis and improvement on real AdaBoost algorithm," *Journal of University of Electronic Science and Technology of China*, vol. 45, no. 10, pp. 1747–1755, 2012.

[10] S. Wu and H. Nagahashi, "Parameterized adaboost: introducing a parameter to speed up the training of real adaboost," *IEEE Signal Processing Letters*, vol. 21, no. 6, pp. 687–691, 2014.

[11] J. Friedman, T. Hastie, and R. Tibshirani, "Additive logistic regression: a statistical view of boosting," *The Annals of Statistics*, vol. 28, no. 2, pp. 337–407, 2000.

[12] A. Vezhnevets and V. Vezhnevets, "Modest AdaBoost: teaching AdaBoost to generalize better," *Graphicon*, vol. 12, no. 5, pp. 987–997, 2005.

[13] Y. Z. J. Thongkam, G. Xu, and F. Huang, "Breast cancer survivability via adaboost algorithms," in *Proceedings of the 2nd Australasian Workshop on Health Data and Knowledge Management (HDKM '08)*, vol. 80, pp. 55–64, New South Wales, Australia, January 2008.

[14] S. Wu and H. Nagahashi, "A new method for solving overfitting problem of gentle AdaBoost," in *Proceedings of the 5th International Conference on Graphic and Image Processing*, vol. 9069 of *Proceedings of SPIE*, pp. 1–6, Hong Kong, China, January 2014.

[15] S. Wu and H. Nagahashi, "Penalized AdaBoost: improving the generalization error of gentle AdaBoost through a margin distribution," *IEICE Transactions on Information and Systems*. Submitted.

[16] Y. Freund, "An adaptive version of the boost by majority algorithm," in *Proceedings of the 12th Annual Conference on Computational Learning Theory*, pp. 102–113, 2000.

[17] A. Demiriz, K. P. Bennett, and J. Shawe-Taylor, "Linear programming boosting via column generation," *Machine Learning*, vol. 46, no. 1–3, pp. 225–254, 2002.

[18] H. Li and C. Shen, "Boosting the minimum margin: LP boost vs. ada boost," in *Proceedings of the Digital Image Computing: Techniques and Applications (DICTA '08)*, pp. 533–539, December 2008.

[19] C. Domingo and O. Watanabe, "Madaboost: a modification of adaboost," in *Proceedings of the 13th Annual Conference on Computational Learning Theory*, pp. 180–189, Palo Alto, Calif, USA, July 2000.

[20] R. A. Servedio, "Smooth boosting and learning with malicious noise," *Journal of Machine Learning Research*, vol. 4, no. 4, pp. 633–648, 2004.

[21] W. Fan, S. J. Stolfo, J. Zhang, and P. K. Chan, "AdaCost: misclassification cost-sensitive boosting," in *Proceedings of the 16th International Conference on Machine Learning*, pp. 97–105, 1999.

[22] K. M. Ting, "A comparative study of cost-sensitive boosting algorithms," in *Proceedings of 17th International Conference on Machine Learning*, pp. 983–990, 2000.

[23] M. V. Joshi, V. Kumar, and R. C. Agarwal, "Evaluating boosting algorithms to classify rare classes: comparison and improvements," in *Proceedings of the 1st IEEE International Conference on Data Mining (ICDM '01)*, pp. 257–264, December 2001.

[24] Y. Sun, M. S. Kamel, A. K. C. Wong, and Y. Wang, "Cost-sensitive boosting for classification of imbalanced data," *Pattern Recognition*, vol. 40, no. 12, pp. 3358–3378, 2007.

[25] D. P. Young and J. M. Ferryman, "Faster learning via optimised adaboost," in *Proceedings of the IEEE Conference on Advanced Video and Signal Based Surveillance (AVSS '05)*, pp. 400–405, September 2005.

[26] M. Seyedhosseini, A. R. C. Paiva, and T. Tasdizen, "Fast AdaBoost training using weighted novelty selection," in *Proceedings of the International Joint Conference on Neural Network (IJCNN '11)*, pp. 1245–1250, August 2011.

[27] B. Paul, G. Athithan, and M. N. Murty, "Speeding up AdaBoost classifier with random projection," in *Proceedings of the 7th International Conference on Advances in Pattern Recognition (ICAPR '09)*, pp. 251–254, February 2009.

[28] C. Sun, J. Hu, and K.-M. Lam, "Feature subset selection for efficient AdaBoost training," in *Proceedings of the 12th IEEE International Conference on Multimedia and Expo (ICME '11)*, pp. 1–6, IEEE, Barcelona, Spain, July 2011.

[29] E. Grossmann, "AdaTree: boosting a weak classifier into a decision tree," in *Proceedings of the Conference on Computer Vision and Pattern Recognition Workshop (CVPR '04)*, pp. 105–112, Washington, DC, USA, 2004.

[30] J. K. Bradley and R. E. Schapire, "FilterBoost: regression and classification on large datasets," in *Proceedings of the 21st Annual Conference on Neural Information Processing Systems (NIPS '07)*, pp. 185–192, December 2007.

[31] Y. Sun, J. Li, and W. Hager, "Two new regularized AdaBoost algorithms," in *Proceedings of the International Conference on Machine Learning and Applications (ICMLA '04)*, pp. 41–48, December 2004.

[32] S. Z. Li and Z. Zhang, "FloatBoost learning and statistical face detection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 26, no. 9, pp. 1112–1123, 2004.

[33] Y. Zhang and P. He, "A revised AdaBoost algorithm: FM-AdaBoost," in *Proceedings of the International Conference on Computer Application and System Modeling (ICCASM '10)*, pp. V11-277–V11-281, October 2010.

[34] G. Rätsch, T. Onoda, and K.-R. Müller, "Soft margins for AdaBoost," *Machine Learning*, vol. 42, no. 3, pp. 287–320, 2001.

[35] Y. Freund, "A more robust boosting algorithm," http://arxiv.org/abs/0905.2138.

[36] M. K. Warmuth, K. Glocer, and G. Rätsch, "Boosting algorithms for maximizing the soft margin," in *Proceedings of the 21st Annual Conference on Neural Information Processing Systems (NIPS '07)*, pp. 1–8, December 2007.

[37] Y. Lu, Q. Tian, and T. Huang, "Interactive boosting for image classification," in *Proceedings of the 7th International Conference on Multiple Classifier Systems*, pp. 180–189, 2007.

[38] J. J. Rodríguez and J. Maudes, "Boosting recombined weak classifiers," *Pattern Recognition Letters*, vol. 29, no. 8, pp. 1049–1059, 2008.

[39] M. Warmuth, K. Glocer, and S. Vishwanathan, "Entropy regularized LPBoost," in *Proceedings of the 19th International Conference on Algorithmic Learning Theory*, pp. 256–271, 2008.

[40] P. K. Mallapragada, R. Jin, A. K. Jain, and Y. Liu, "SemiBoost: boosting for semi-supervised learning," *IEEE Transactions on*

*Pattern Analysis and Machine Intelligence*, vol. 31, no. 11, pp. 2000–2014, 2009.

[41] K. Chen and S. Wang, "Semi-supervised learning via regularized boosting working on multiple semi-supervised assumptions," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 33, no. 1, pp. 129–143, 2011.

[42] L. I. Kuncheva, "Error bounds for aggressive and conservative adaboost," in *Proceedings of the 4th International Workshop on Multiple Classifier Systems*, pp. 25–34, 2003.

[43] X. Miao and J. S. Heaton, "A comparison of random forest and AdaBoost tree in ecosystem classification in east Mojave Desert," in *Proceedings of the 18th International Conference on Geoinformatics*, pp. 1–6, Beijing, China, June 2010.

[44] E. Alfaro, N. García, M. Gámez, and D. Elizondo, "Bankruptcy forecasting: an empirical comparison of AdaBoost and neural networks," *Decision Support Systems*, vol. 45, no. 1, pp. 110–122, 2008.

[45] Y. Wang, P. Han, X. Lu, R. Wu, and J. Huang, "The performance comparison of Adaboost and SVM applied to SAR ATR," in *Proceedings of the CIE International Conference on Radar (ICR '06)*, pp. 1–4, October 2006.

[46] A. Ferreira, "Survey on boosting algorithms for supervised and semisupervised learning," Tech. Rep., Instituto Superior de Engenharia de Lisboa, 2007.

[47] C. Seiffert, T. M. Khoshgoftaar, J. Van Hulse, and A. Napolitano, "Resampling or reweighting: a comparison of boosting implementations," in *Proceedings of the 20th IEEE International Conference on Tools with Artificial Intelligence (ICTAI '08)*, pp. 445–451, November 2008.

[48] D. Hegazy and J. Denzler, "Performance comparison and evaluation of adaboost and softboost algorithms on generic object recognition," *Proceedings of World Academy of Science: Engineering & Technolog*, vol. 47, pp. 70–74, 2008.

[49] M. Drauschke and W. Forstner, *Comparison of AdaBoost and ADTboost for Feature Subset Selection*, INSTICC PRESS, 2008.

[50] S. Jurić-Kavelj and I. Petrović, "Experimental comparison of AdaBoost algorithms applied on leg detection with different range sensor setups," in *Proceedings of the 19th International Workshop on Robotics in Alpe-Adria-Danube Region (RAAD '10)*, pp. 267–272, June 2010.

[51] J. Sun, D. Cui, D. Gu, H. Cai, and G. Liu, "Empirical analysis of AdaBoost algorithms on license plate detection," in *Proceedings of the IEEE International Conference on Mechatronics and Automation (ICMA '09)*, pp. 3497–3502, August 2009.

[52] P. Natesan, P. Balasubramanie, and G. Gowrison, "Performance comparison of AdaBoost based weak classifiers in network intrusion detection," *Journal of Information Systems and Communication*, vol. 3, no. 1, pp. 295–299, 2012.

[53] A. J. Ferreira and M. T. Figueiredo, "Boosting algorithms: a review of methods, theory, and applications," *Journal of Information Systems and Communication*, vol. 3, no. 1, pp. 35–85, 2012.

[54] A. Vezhnevets and V. Vezhnevets, "GML AdaBoostMatlab Toolbox Manual," http://graphics.cs.msu.ru/ru/science/research/machinelearning/adaboosttoolbox.

[55] K. Bache and M. Lichman, *UCI Machine Learning Repository*, University of California, School of Information and Computer Science, Irvine, Calif, USA, 2013, http://archive.ics.uci.edu/ml/.

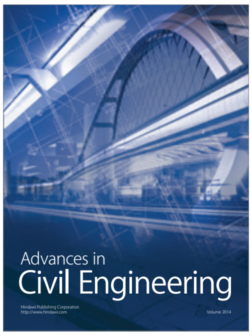
Journal of
Engineering

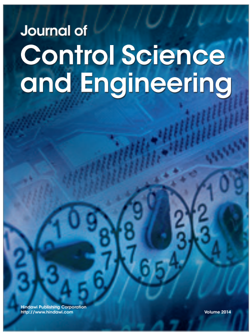
The Scientific World Journal

International Journal of
Rotating Machinery

Journal of
Sensors

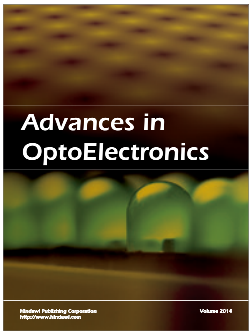
International Journal of
Distributed Sensor Networks


Advances in
Civil Engineering


Journal of
Control Science and Engineering

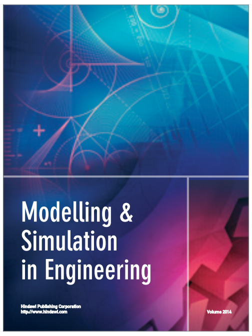
Journal of
Robotics

Journal of
Electrical and Computer Engineering

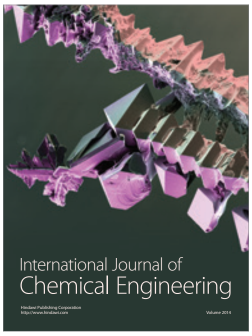

Advances in
OptoElectronics

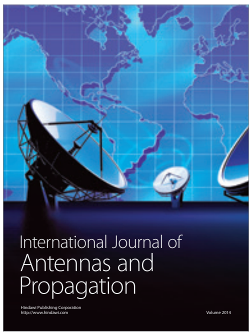
VLSI Design

International Journal of
Navigation and Observation

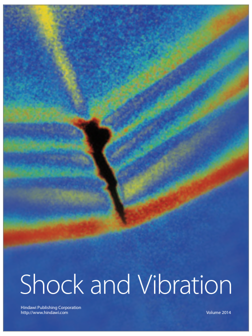

Modelling & Simulation in Engineering

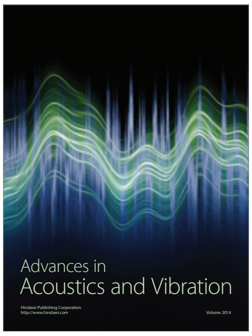
International Journal of
Aerospace Engineering


International Journal of
Chemical Engineering


International Journal of
Antennas and Propagation

Active and Passive Electronic Components


Shock and Vibration


Advances in
Acoustics and Vibration