Neurocomputing 156 (2015) 186–192

Contents lists available at ScienceDirect

Neurocomputing

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

A selective boosting technique for pattern classification

De Z. Li [a], Wilson Wang [b,*], Fathy Ismail [c]

[a] Department of Mechanical and Mechatronics Engineering, University of Waterloo, Waterloo, ON, Canada N2L 3G1
[b] Department of Mechanical Engineering, Lakehead University, Thunder Bay, ON, Canada P7B 5E1
[c] Department of Mechanical and Mechatronics Engineering, University of Waterloo, Waterloo, ON, Canada N2L 3G1

ARTICLE INFO

Article history:
Received 3 August 2014
Received in revised form
19 December 2014
Accepted 21 December 2014
Available online 14 January 2015

Keywords:
AdaBoost algorithm
Ensemble learning
Pattern classification
Overfitting problem

ABSTRACT

The classical AdaBoost algorithm is an ensemble of weak learners, and can be used to construct a strong classifier. A weak learner is incorporated into the ensemble at each step, and the classification of the derived ensemble is improved by properly adjusting the weight of each weak learner. The classical AdaBoost algorithm has some limitations; for example, it is sensitive to noisy data, which may impede the generalization capability of the derived classifier and lead to overfitting problem. A selective boosting, sBoost, technique is proposed in this paper to tackle these problems. The proposed sBoost classifier focuses on the generalized classification performance rather than those hard-to-learn samples, and the penalties of hard-to-learn samples are mitigated to the degree associated with their noise level. An error correction method is suggested to detect potential clean samples and prevent them from misclassification to further alleviate the overfitting problem. The effectiveness of the developed sBoost technique is tested by a series of simulation tests. Test results show that the developed sBoost technique can improve the classification accuracy and prevent the overfitting problem effectively.

© 2015 Elsevier B.V. All rights reserved.

1. Introduction

AdaBoost, short for Adaptive Boosting, is a machine learning algorithm, which can be used to improve data training performance [1]. AdaBoost is adaptive because subsequent constructed classifiers are modified to help discern samples misclassified by the previous classifiers. It can emphasize misclassified samples by increasing their weights in training data distribution, and then classify them by using appropriate weak learners. By integrating a weak learner at each step, the ensemble of weak learners can better learn how to classify the samples and improve classification accuracy. A weak learner can be any classifier that has moderate classification accuracy. The classical AdaBoost has merits such as solid theoretical basis and ease of implementation, and has been used in many applications such as medical diagnoses [2,3], image retrieval/classification [4,5], and network intrusion detection [6]. However, the classical AdaBoost algorithm has some limitations such as being sensitive to noisy samples, overfitting and slow convergence [7,8].

Many research efforts have been made to improve the AdaBoost [9–11]. To speed up convergence, a Real AdaBoost algorithm was proposed in [1] to calculate the probability and confidence limits to estimate the degree to which a sample belongs to a specific class. A LogitBoost method [12] employed an additive logistic regression model as cost function to improve the convergence. However, these methods also suffer from the overfitting problem. Overfitting occurs as the number of weak learners increases when noisy samples are present [7]. The classifier does not have an efficient mechanism to learn noisy samples, which causes some potential clean samples to be misclassified. Ratsch et al. [13] proposed a boosting technique with soft margins to deal with noisy samples by computing the aggregated weights of each sample and reducing the penalties for hard-to-learn samples; some hard-to-learn samples, however, may not be noise affected but clean ones, and their penalty should not be reduced. Cao et al. [14] applied the noise detection method to identify noisy samples in the training data and assign them a different class label from their original ones in the binary classification problem. This technique is highly dependent on the noise detection mechanism; less effective noise detection method may lead to poor classifier performance. Gao et al. [15] applied a weighted *K*-nearest-neighbor algorithm in a transformed space to identify and discard suspicious noisy samples; but some of these discarded samples may be useful to improve the classification accuracy.

The classical AdaBoost algorithms [1,9–12] focus on the classification of those large weight samples, which may lead to severe misclassification of small weight samples and result in overfitting problems. The overfitting problem could be addressed using some

* Corresponding author. Tel.: +807 766 7174.
E-mail addresses: d45li@uwaterloo.ca (D.Z. Li), Wilson.Wang@Lakeheadu.ca (W. Wang), fmismail@uwaterloo.ca (F. Ismail).

http://dx.doi.org/10.1016/j.neucom.2014.12.063
0925-2312/© 2015 Elsevier B.V. All rights reserved.

methods such as [13–15] to some extent; however they cannot effectively process large weight samples based on data characteristics, which may result in low classification accuracy. To effectively reduce the overfitting problem, a new selective Boosting technique, sBoost for short, is proposed in this work to tackle the overfitting problem effectively and improve classification accuracy, under low noise levels (e.g., from no noise to 10% noise). In the proposed sBoost technique, the penalties of samples with large weights are adjusted adaptively to improve the classification performance. For those large weight samples caused by noise, their penalties are mitigated according to the noise degree to reduce the overfitting problem. Some clean samples may carry large weight because of the limitation of weak learners or inconsistency in combining weak learners. Their weights are maintained to prevent misclassification. This proposed sBoost technique is new in the following aspects: (a) a sample weight regulator is proposed to adaptively regulate samples' penalties based on their accumulated weights and noise degrees to reduce overfitting; and (b) an error correction method is proposed to detect potential clean samples in the test data, predict their class label, and prevent them from being misclassified by a classifier suffering overfitting.

The remainder of this paper is organized as follows. The proposed sBoost technique is discussed in Section 2. In Section 3, the effectiveness of the proposed sBoost technique is examined by some benchmark classification problems. Some concluding remarks are drawn in Section 4.

2. The proposed sBoost technique

Before the discussion of the proposed sBoost technique, some description is given first for the classical AdaBoost algorithm.

2.1. Classical AdaBoost algorithm

Consider a binary classification problem with training data samples $(x_1, y_1), (x_2, y_2), \ldots, (x_N, y_N)$, where the class label is $y_i \in \{-1, +1\}$ and N is the number of samples in the training data set. If T is the maximum number of weak learners, the AdaBoost algorithm takes the following processes:

(a) Initialize the distribution of training data set $L_1(i) = 1/N$, $i = 1, 2, \ldots, N$.
(b) Train the weak learner $h_t \in \{-1, +1\}$ using training data set with distribution L_t.
(c) Compute the sum of weights of misclassified samples $\varphi_t = \sum_{i=1}^{N} L_t(i)[[h_t(x_i) \neq y_i]]$. The logic operator is defined as $[[\Gamma]] = \begin{cases} 1 & \Gamma \neq 0 \\ 0 & \Gamma = 0 \end{cases}$.
(d) Calculate the weight of the weak learner h_t, $\alpha_t = (1/2)\ln(1-\varphi_t/\varphi_t)$.
(e) Update the weights of the training samples, $L_{t+1}(i) = (L_t(i)\exp(-\alpha_t h_t(x_i) y_i)/J_t)$, where $J_t = \sum_{i=1}^{N} L_t(i)\exp(-y_i \alpha_t h_t(x_i))$.
(f) Repeat step (b) to step (e) with $t = 1: T$.
(g) Normalize the weight of weak learner h_t, $\alpha'_t = (\alpha_t / \sum_{t=1}^{T} \alpha_t)$.

If $f(x_i) = \sum_{t=1}^{T} \alpha'_t h_t(x_i)$, and $H(x_i) = sign[f(x_i)]$ is the predicted class label, the margin of a sample (x_i, y_i) can be given as

$$\hat{\rho}_t(x_i) = \frac{y_i \sum_{t=1}^{T} \alpha_t h_t(x_i)}{\sum_{t=1}^{T} \alpha_t} \tag{1}$$

Theorem 1. *(Schapire et al. [16]): Given the weighted training error $\varphi_t = \sum_{i=1}^{N} L_t(i)[[h_t(x_i) \neq y_i]]$, for any δ, the probability that $\hat{\rho}_t(x_i) \leq \delta$ can be given as,*

$$\mathbf{P}[\hat{\rho}_t(x_i) \leq \delta] \leq 2^T \prod_{t=1}^{T} \sqrt{\varphi_t^{1-\delta}(1-\varphi_t)^{1+\delta}} \tag{2}$$

When $\delta = 0$, $\mathbf{P}[\hat{\rho}_t(x_i) \leq 0]$ represents the probability that a sample is misclassified, which should be minimized.

2.2. Sample weight regulator in the proposed sBoost technique

The accumulated weight of a sample is an effective indicator, which determines the degree this sample contributes to potential overfitting. Thus it is used to properly adjust the penalty of this sample in order to reduce overfitting.

Let β_t be the weight of the weak learner h_t in sBoost, the margin can be defined in the same way as in Eq. (1)

$$\rho_t(x_i) = \frac{y_i \sum_{t=1}^{T} \beta_t h_t(x_i)}{\sum_{t=1}^{T} \beta_t} \tag{3}$$

The accumulated weight of sample i is computed as

$$\kappa_t(i) = \frac{\exp(\rho_t(x_i))}{\sum_{i=1}^{N} \exp(\rho_t(x_i))} \tag{4}$$

$$\xi_t(i) = \sum_{j=1}^{t} (\beta_t \kappa_t(i)) \tag{5}$$

$$\psi_t(i) = \frac{\xi_t(i)}{\max\{\xi_t(i)\}} \beta_t \tag{6}$$

Since some noisy samples may cause overfitting, they need to be processed properly in the training process. In the proposed sample weight regulator, the k-nearest-neighbor method is adopted to identify noisy samples and evaluate their noise degrees. Given K nearest samples of a target sample in the training data set, if p samples of these K samples have the same class label as this target sample $(0 \leq p \leq K)$, the noise degree of this sample will be estimated by

$$\theta_i = 1 - \frac{p}{K} \tag{7}$$

where the noise degree is independent of time step t.

The proposed sample weight regulator is given by

$$\varpi_t(i) = \psi_t(i)\theta_i = \frac{\xi_t(i)}{\max\{\xi_t(i)\}} \beta_t \theta_i \tag{8}$$

It can be proven that $0 \leq \varpi_t(i) \leq \beta_t$. Accordingly, the update of the i^{th} sample weight at step t can be formulated as:

$$\begin{aligned} L_{t+1}(i) &= \frac{L_t(i)\exp(-\beta_t h_t(x_i) y_i - \varpi_t(i))}{J'_t} \\ &= \frac{1}{N} \frac{\exp\left\{-\sum_{t=1}^{T} [y_i \beta_t h_t(x_i) + \varpi_t(i)]\right\}}{\prod_{t=1}^{T} J'_t} \end{aligned} \tag{9}$$

where J'_t is the cost function of sBoost.

By using the sample weight regulator $\varpi_t(i)$, the noisy samples will be processed according to their noise degree. For each sample, the maximum penalty reduction is determined by its accumulated weight, which will be lessened by the sample's noise degree. Consequently, the less noisy samples will not have over-reduced penalties to decrease their chance of being correctly classified, and the penalties of noisy samples can be adjusted to reduce overfitting.

Given $0<\exp(-\varpi_t(i))\le 1$, the cost function associated with the sample weight regulator is suggested as:

$$\begin{aligned} J_t' &= \sum_{i=1}^{N} L_t(i)\exp(-y_i\beta_t h_t(x_i)-\varpi_t(i)) \\ &= \left(\sum_{y_i h_t(x_i)=1} L_t(i)\exp(-\varpi_t(i))\right)\exp(-\beta_t) \\ &\quad + \left(\sum_{y_i h_t(x_i)=-1} L_t(i)\exp(-\varpi_t(i))\right)\exp(\beta_t) \\ &= \varepsilon_t\exp(-\beta_t)+\tau_t\exp(\beta_t) \end{aligned} \tag{10}$$

where $\varepsilon_t = \sum_{y_i h_t(x_i)=1} L_t(i)\exp(-\varpi_t(i))$ and $\tau_t = \sum_{y_i h_t(x_i)=-1} L_t(i)\exp(-\varpi_t(i))$. It can be seen that $0<\varepsilon_t<1$, and $0<\tau_t<1$. In addition, $\sum_{i=1}^{N} L_{t+1}(x_i)=1$ can be derived from Eqs. (9) and (10).

Taking derivative of Eq. (10) with respect to β_t and letting it equal zero, the optimal weight of weak classifier h_t will be

$$\beta_t = \frac{1}{2}\ln\left(\frac{\varepsilon_t}{\tau_t}\right) \tag{11}$$

Inserting Eq. (11) into Eq. (10), we get

$$\begin{aligned} J_t' &= \varepsilon_t\exp(-\beta_t)+\tau_t\exp(\beta_t) \\ &= \varepsilon_t\exp\left(-\frac{1}{2}\ln\left(\frac{\varepsilon_t}{\tau_t}\right)\right)+\tau_t\exp\left(\frac{1}{2}\ln\left(\frac{\varepsilon_t}{\tau_t}\right)\right) \\ &= 2\sqrt{\varepsilon_t\tau_t} \end{aligned} \tag{12}$$

Let

$$\begin{aligned} \rho_t'(x_i) &= \frac{\sum_{t=1}^{T}\left[y_i\beta_t h_t(x_i)+\varpi_t(i)\right]}{\sum_{t=1}^{T}\beta_t} \\ &= \frac{y_i\sum_{t=1}^{T}\beta_t h_t(x_i)}{\sum_{t=1}^{T}\beta_t}+\frac{\sum_{t=1}^{T}\varpi_t(i)}{\sum_{t=1}^{T}\beta_t} \end{aligned} \tag{13}$$

Since $\mathbf{P}[\rho_t(x_i)\le 0]$ represents the probability that a sample is misclassified, the minimization of $\mathbf{P}[\rho_t(x_i)\le 0]$ is equivalent to

$$\mathbf{P}[\rho_t(x_i)\le 0] = \mathbf{P}\left[\rho_t'(x_i)\le \frac{\sum_{t=1}^{T}\varpi_t(i)}{\sum_{t=1}^{T}\beta_t}\right] \le \mathbf{P}[\rho_t'(x_i)\le\gamma] \tag{14}$$

where

$$\gamma = \sup_i\left(\frac{\sum_{t=1}^{T}\varpi_t(i)}{\sum_{t=1}^{T}\beta_t}\right) \tag{15}$$

The probability of $\rho_t'(x_i)\le\gamma$ can be derived as

$$\sum_{t=1}^{T}\left[y_i\beta_t h_t(x_i)+\varpi_t(i)\right]\le\gamma\sum_{t=1}^{T}\beta_t \tag{16}$$

$$\exp\left(\gamma\sum_{t=1}^{T}\beta_t-\sum_{t=1}^{T}\left[y_i\beta_t h_t(x_i)+\varpi_t(i)\right]\right)\ge 1 \tag{17}$$

$$\begin{aligned} \mathbf{P}[\rho'(x_i)\le\gamma] &\le \mathbf{E}\left[\exp\left(\gamma\sum_{t=1}^{T}\beta_t-\sum_{t=1}^{T}\left[y_i\beta_t h_t(x_i)+\varpi_t(i)\right]\right)\right] \\ &= \frac{1}{N}\exp\left(\gamma\sum_{t=1}^{T}\beta_t\right)\sum_{i=1}^{N}\exp\left(-\sum_{t=1}^{T}\left[y_i\beta_t h_t(x_i)+\varpi_t(i)\right]\right) \\ &= \exp\left(\gamma\sum_{t=1}^{T}\beta_t\right)\left(\prod_{t=1}^{T}J_t'\right)\sum_{i=1}^{N}L_{t+1}(x_i) \\ &= \exp\left\{\gamma\sum_{t=1}^{T}\left[\frac{1}{2}\ln\left(\frac{\varepsilon_t}{\tau_t}\right)\right]\right\}\left[\prod_{t=1}^{T}\left(2\sqrt{\varepsilon_t\tau_t}\right)\right] \\ &= 2^T\prod_{t=1}^{T}\left(\sqrt{\left(\frac{\varepsilon_t}{\tau_t}\right)^r\varepsilon_t\tau_t}\right) \end{aligned} \tag{18}$$

Since the condition $0\le\varpi_t(i)\le\beta_t$ is satisfied, it can be derived that $0\le(\sum_{t=1}^{T}\varpi_t(i)/\sum_{t=1}^{T}\beta_t)\le 1$, and $0\le\gamma\le 1$. From Eq. (18), it is seen that $0<\sqrt{(\varepsilon_t/\tau_t)^r\varepsilon_t\tau_t}<1$, when $0<\varepsilon_t<\tau_t<1$. The larger the T is, the smaller the value $\prod_{t=1}^{T}\left(\sqrt{\left(\frac{\varepsilon_t}{\tau_t}\right)^r\varepsilon_t\tau_t}\right)$ becomes. From Eqs. (14) and (18), it is easy to derive that $\mathbf{P}[\rho_t(x_i)\le 0]\le 2^T\prod_{t=1}^{T}\left(\sqrt{\left(\frac{\varepsilon_t}{\tau_t}\right)^r\varepsilon_t\tau_t}\right)$. Consequently, the more the weak learners are incorporated into the ensemble (or the larger the T is), the lower the probability that a sample is misclassified (or smaller the $\mathbf{P}[\rho_t(x_i)\le 0]$ becomes).

2.3. Error correction method

The sample weight regulator is used in the proposed sBoost to alleviate the penalties of noisy hard-to-classify samples, so as to reduce overfitting. However, the overfitting problem may still occur due to the inconsistency of training data and test data. To address this problem, an error correction method is suggested in the proposed sBoost to detect potential clean samples in the test data, and correctly classify them. To implement the proposed error correction method, each dimension of the training data $x(i)$ is normalized over [0, 1]. Let $\hat{x}$ denote the original data; U^d and L^d denote the respective maximum and minimum values of the original training data at dimension d, then

$$x_i^d = \frac{\hat{x}_i^d - L^d}{U^d - L^d}, i = 1,2,\ldots,N \tag{19}$$

where N is the number of training data, and d is the index of dimension. Each test sample $x'(i)$ is also processed using the same way as the training data.

$$x_j'^d = \frac{\hat{x}_j'^d - L^d}{U^d - L^d}, j = 1,2,\ldots,N_t \tag{20}$$

where N_t is the number of test data.

The error correction method can be represented as

$$K_t(j) = \sum_{i\in S_j}(y_i/q), j = 1,2,\ldots,N_t \tag{21}$$

where $y_i\in\{-1,1\}$ is the class label of the training data. S_j represents a set that contains q samples in the training data that are nearest neighbors to the test sample y_j' measured by Euclidean distance. It is seen that $|K_t(j)|\le 1$. The error correction method can be summarized as follows:

(I) If $K_t(j)=1$, the test sample x_j' is considered as a clean sample with class label $y_j' = 1$.
(II) If $K_t(j)=-1$, the test sample x_j' is considered as a clean sample with class label $y_j' = -1$.
(III) If $|K_t(j)|<1$, the class label of the test sample x_j' is determined by boosting using the sample weight regulator.

Potential clean samples (i.e., $|K_t(j)|=1$) in the test data set are detected using Eq. (21) and assigned predicted class label according to (I) and (II). Other test samples are classified using boosting

with the proposed sample weight regulator. By using this error correction method, the potential clean samples will not be misclassified by the classifier and overfitting can be further reduced to improve the classification accuracy.

2.4. Implementation of the sBoost technique

The developed sBoost technique involves the boosting and the error correction. The proposed boosting strategy with sample weight regulator will be implemented using the following steps:

(a) Initialize the distribution of training data set $L_1(i)=1/N$, $i=1, 2, \ldots, N$.
(b) Train the weak learner $h_t \in \{-1, +1\}$ using training data set with distribution L_t.
(c) Calculate $\varepsilon_t = \sum_{y_i h_t(x_i)=1} L_t(i)\exp(-\varpi_t(i))$; $\tau_t = \sum y_i h_t(x_i) = -1 L_t(i)\exp(-\varpi_t(i))$; and the weight of the weak learner h_t, $\beta_t = (1/2)\ln(\varepsilon_t/\tau_t)$.
(d) If $0<\tau_t<\varepsilon_t<1$ is not satisfied, $L_{t+1}(i)=1/N$, $i=1, 2, \ldots, N$; otherwise, update the weights of the training samples, $L_{t+1}(i) = (L_t(i)\exp(-\beta_t h_t(x_i) y_i - \varpi_t(i))/J_t')$, where $J_t' = \sum_{i=} 1^N L_t(i)\exp(-y_i\beta_t h_t(x_i) - \varpi_t(i))$.
(e) Repeat step (b) to step (d) with $t=1: T$.
(f) Formulate the ensemble classifier as $H(x_i) = sign\left[\sum_{t=1}^{T}\beta_t' h_t(x_i)\right]$, where $\beta_t' = (\beta_t/\sum_{t=1}^{T}\beta_t)$ is the normalized the weights of weak learner h_t.

After boosting with sample weight regulator, the error correction method is utilized to further improve the classification accuracy. Given a test sample x_j', the error correction is undertaken in the following steps:

(a) Determine if x_j' is a potential clean sample using Eq. (21). If it is, its class label is predicted using rules (I)–(II).
(b) If $|K_t(j)|<1$, apply rule (III). Predict the class label of x_j' by $H(x_j') = sign\left[\sum_{t=1}^{T}\beta_t' h_t(x_j')\right]$.

3. Performance evaluation and application

The effectiveness of the proposed sBoost technique is verified in this section by simulation tests using 12 benchmark data sets from KEEL database [17]. The dataset information is summarized in Table 1. These data sets range in size from 106 to 19,020, as well as in dimension (features) from 5 to 60. To make a comparison, the classical AdaBoost algorithm, AB for short, the weight decay AdaBoost algorithm, WAB for short [13], the noise detection based AdaBoost algorithm, ND for short [14], and sBoost only using sample weight regulator, sBoost-1, are employed. The sBoost using both sample weight regulator and error correction method is denoted as sBoost-2. The AB, WAB, and ND are used to evaluate the effect of sample weight regulator in sBoost; whereas the sBoost-1 is used to evaluate the effectiveness of the suggested error correction mechanism.

Table 1
Data set information.

Data name	No. of samples	No. of features
Phoneme	5404	5
Appendicitis	106	7
Pima	768	8
Breast	277	9
Tic-tac-toe	958	9
Magic	19,020	10
Heart	270	13
Australian	690	14
German	1000	20
Twonorm	7400	20
Wdbc	569	30
Sonar	208	60

In each test, the maximum number of weak learners is fixed at 200 for the five classifiers: AB, WAB, ND and sBoost-1 and sBoost-2. The decision tree C4.5 is used as the weak learner. Artificial noise is added to each of the benchmark data sets. The 2-fold cross-validation approach is conducted 5 times (i.e., 5×2 cross-validations) for each benchmark data set. In each 2-fold cross-validation, the data are randomly split into two data sets for two runs: one set is used for training and the other set for testing as the first run. Then the two data sets are switched for training and testing as the second run. Noise level $E\%$ (e.g., 0–10%) in a data set is realized by randomly selecting $E\%$ of total samples in the training data set and changing their class labels [18]. The parameters of the sBoost technique are given as $K=5$ and $q=5$.

To illustrate the advantage of the proposed sBoost technique over the other related classifiers, the differences of successful classification rate (SR), that is, SR(ND) – SR(WAB), SR(sBoost-1) – SR(WAB) and SR(sBoost-2) – SR(WAB) for the 11 benchmark data sets are shown in Figs. 1–3. The SR of the WAB is the minuend of the subtraction. The indices $1\sim11$ in the horizontal axes represent the data sets of Phoneme, Appendicitis, Pima, Breast, Magic, Heart, Australian, German, Twonorm, Wdbc, Sonar, respectively. Since the values of SR(sBoost-2) – SR(WAB) of the Tic-tac-toe data set fall in the range of [0.023, 0.038], much larger than those in other cases, they are not shown in Figs. 1–3 for proper view scales. The difference values of SR(AB) with respect to other classifiers (i.e., WAB, ND, sBoost-1 and sBoost-2) are also much larger than other

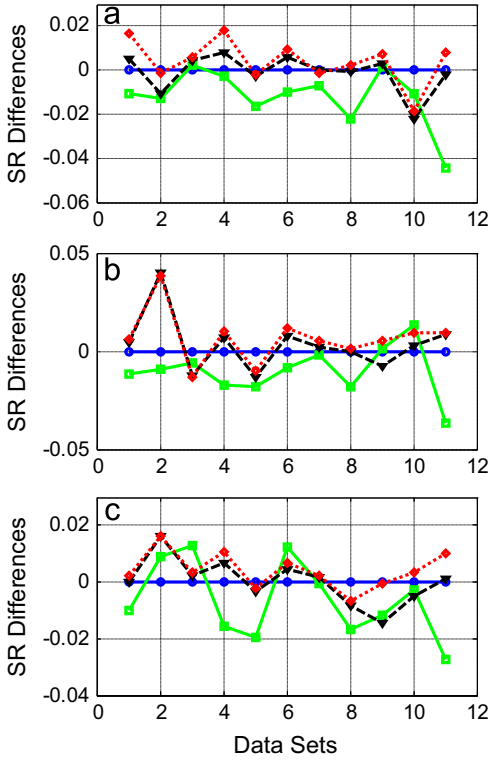


Fig. 1. The differences of SR using 50 weak learners with different noise levels: (a) 0%, (b) 5% and (c) 10%. The blue-circle-solid line represents SR(WAB) – SR(WAB); the green-square-solid line represents SR(ND) – SR(WAB); the black-triangle-dashed line represents SR(sBoost-1) – SR(WAB); the red-diamond-dotted line represents SR(sBoost-2) – SR(WAB). (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

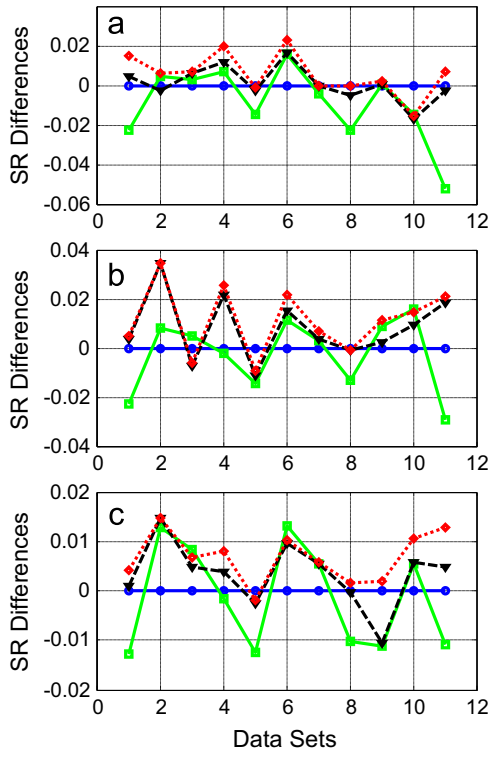


Fig. 2. The differences of SR using 100 weak learners with different noise levels: (a) 0%, (b) 5% and (c) 10%. The blue-circle-solid line represents SR(WAB) – SR(WAB); the green-square-solid line represents SR(ND) – SR(WAB); the black-triangle-dashed line represents SR(sBoost-1) – SR(WAB); the red-diamond-dotted line represents SR(sBoost-2) – SR(WAB). (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

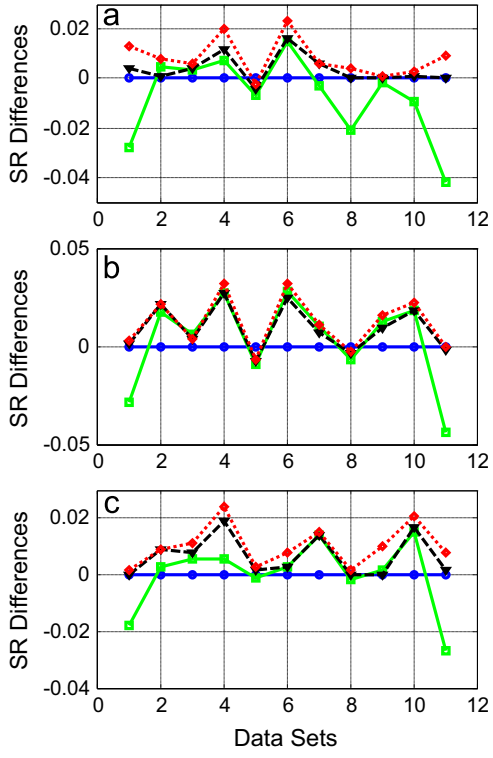


Fig. 3. The differences of SR using 200 weak learners with different noise levels: (a) 0%, (b) 5% and (c) 10%. The blue-circle-solid line represents SR(WAB) – SR(WAB); the green-square-solid line represents SR(ND) – SR(WAB); the black-triangle-dashed line represents SR(sBoost-1) – SR(WAB); the red-diamond-dotted line represents SR(sBoost-2) – SR(WAB). (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

cases (e.g., SR(sBoost-2) – SR(AB) $\in$ [0.047,0.127]), correspondingly these values are not shown in Figs. 1–3 either.

From Figs. 1–3, it is seen that sBoost-1 and sBoost-2 outperform WAB and ND in overall performance of these 11 data sets. Their superior performance over WAB is clear as more weak learners are incorporated into the ensemble, because the sample weight regulator in sBoost-1 and sBoost-2 can effectively adjust noisy sample weights to reduce overfitting. The sBoost-1 and sBoost-2 are superior to ND due to its efficient noise-degree-based adaptive weight regulation. On the other hand, the sBoost-2 outperforms sBoost-1 in most test scenarios, because it detects and processes clean samples more effectively. In Fig. 1(c), it is seen that the WAB performs better than the proposed sBoost techniques in the fifth, eighth, and ninth data sets, whereas ND can provide superior performance in the third and sixth data sets. The sBoost techniques reduce the penalty of noise samples, so as to improve their generalization capability. This may result in misclassification of some noisy samples, and in turn lead to poor classification performance in the premature stage (e.g., 50 weak learners are used in this case). Nevertheless, the proposed sBoost techniques surpass both WAB and ND as more weak learners are incorporated into the ensemble as shown in Fig. 3(c) (e.g., 200 weak learners are used in this case), because the sBoost techniques have better generalization capability than WAB and ND. When the noise level is high, the sample misclassification is high; since the weights (or penalties) of noise samples accumulate fast, the overfitting problem is prone to occurring. The weights (or penalties) of noise samples accumulate as the number of weak learners increases. Thus the more weak learners are used; it is more likely the overfitting occurs.

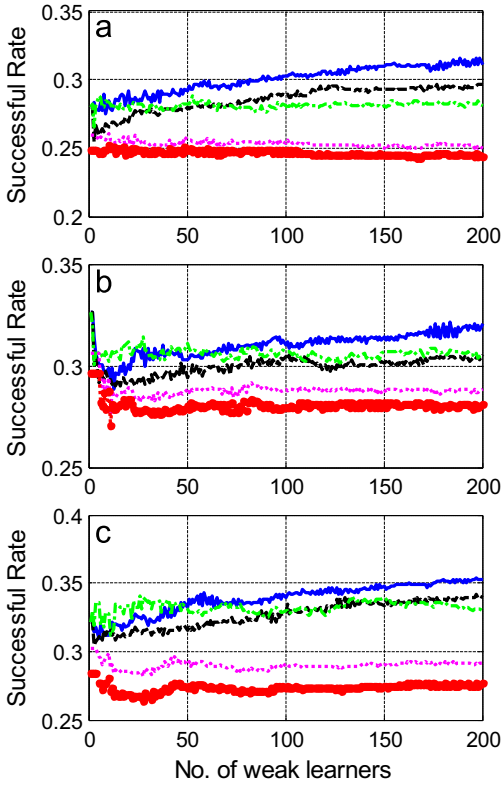


Fig. 4. The averaged test error rates of breast cancer data with different noise levels: (a) 0%, (b) 5% and (c) 10%. The blue-solid line represents AB; the black-dashed line represents WAB; the green-dash-dotted line represents ND; the magenta-dotted line represents sBoost-1; the red-square-dotted line represents sBoost-2. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

Fig. 4 shows the averaged test error rates of the real-world breast cancer data set in terms of five boosting techniques. It is seen that both WAB and ND generate low averaged test error rates than AB as the number of weak learner increases, because of their capacity for addressing overfitting problem (i.e., the test error increases as the number of weak learner increases). Nevertheless, their averaged test rates still increase a bit because the overfitting is not sufficiently reduced. The sBoost techniques outperform both WAB and ND, and no obvious overfitting is detected, because their sample weight regulation can effectively reduce the overfitting problem and improve the classification accuracy. The sBoost-2 overtakes sBoost-1 because the potential clean samples can be correctly classified. At low noise level, less noisy samples are present, which will result in less sample misclassification. The weights (or penalties) of noisy samples accumulate slowly, and the overfitting problem can be reduced effectively using the proposed sample weight regulation. When the noise level increases, the rate of sample misclassification also increases. Therefore, the weights of noisy samples increase fast, and the overfitting problem could be reduced less effectively.

Table 2 summarizes the averaged test error rates of the AB, WAB, ND, support vector machine (SVM), sBoost-1 and sBoost-2 over 12 data sets. It is seen that the WAB and ND outperforms AB because of their strategy to process hard-to-learn samples. sBoost-1 outperforms AB, WAB and ND at all three noise levels, because it can effectively adjust sample weights to reduce overfitting. The error correction method in sBoost-2 can further improve classification accuracy compared with sBoost-1, because it can correct some misclassified clean samples due to the overfitting problem. The sBoost-2 outperforms SVM because of the advanced boosting mechanism with both sample weight regulation and error correction.

Next, the statistical tools, Wilcoxon signed-rank test [19] and sign test [20], will be employed to evaluate the pairwise statistical significant difference among these six classifiers, AB, WAB, ND, SVM, sBoost-1 and sBoost-2. The comparisons are illustrated in Tables 3–5 as a matrix of the pairwise statistical tests. The numbers of wins, ties and loses are represented as $w/t/l$. p_s is the p-value of the sign test and p_w denotes the p-value of Wilcoxon signed-rank test. Symbol '$\surd$' is used to indicate that the two classifiers are significantly different at 95% confidence; where as '□' indicates that the two classifiers are significantly different at 90% confidence.

From Tables 3–5, it is seen that WAB, ND, SVM, sBoost-1 and sBoost-2 outperform AB at the three noise levels. Although sBoost-1 has no clear difference over WAB at 0% noise level, its performance is much better than WAB as the noise level increases, due to its efficient noise manipulation mechanism. The ND performs better than WAB at 5% and 10% noise level, however, sBoost techniques outperforms ND in all three noise levels because of its enhanced noisy sample processing strategy. On the other hand, the sBoost-2 outperforms the sBoost-1 at all three noise levels due to its effective clean sample correction approach. It is also seen from Tables 3–5 that sBoost-2 generates less mean error than SVM. It overtakes SVM in more benchmark data sets in terms of th-

Table 3
Statistical significant comparisons of six boosting techniques with 0% noise over 12 data sets.

Methods		AB	WAB	ND	SVM	sBoost-1	sBoost-2
Mean		0.1899	0.1770	0.1915	0.1931	0.1758	0.1715
	$w/t/l$		12/0/0	7/0/5	8/0/4	12/0/0	12/0/0
AB	p_s		0.0005 $\surd$	0.7744	0.3877	0.0005 $\surd$	0.0005 $\surd$
	p_w		0.0005 $\surd$	0.9248	0.9097	0.0005 $\surd$	0.0005 $\surd$
	$w/t/l$			4/0/8	7/0/5	7/3/2	10/0/2
WAB	p_s			0.3877	0.7744	0.1797	0.0386 $\surd$
	p_w			0.2407	0.7188	0.2852	0.0503 □
	$w/t/l$				7/0/5	9/0/3	11/0/1
ND	p_s				0.7744	0.1460	0.0063 $\surd$
	p_w				0.7910	0.0664 □	0.0063 $\surd$
	$w/t/l$					7/0/5	8/0/4
SVM	p_s					.7744	0.3877
	p_w					.3911	0.1699
	$w/t/l$						11/1/0
sBoost-1	p_s						0.0010 $\surd$
	p_w						0.0010 $\surd$

Table 4
Statistical significant comparisons of six boosting techniques with 5% noise over 12 data sets.

Methods		AB	WAB	ND	SVM	sBoost-1	sBoost-2
Mean		0.2366	0.2234	0.2235	0.2279	0.2153	0.2100
	$w/t/l$		12/0/0	9/0/3	8/0/4	11/0/1	11/0/1
AB	p_s		0.0005 $\surd$	0.1460	0.3877	0.0063 $\surd$	0.0063 $\surd$
	p_w		0.0005 $\surd$	0.1294	0.6221	0.0010 $\surd$	0.0010 $\surd$
	$w/t/l$			7/0/5	7/0/5	8/0/4	9/1/2
WAB	p_s			0.7744	0.7744	0.3877	0.0654 □
	p_w			0.9268	1.0000	0.0732 □	0.0107 $\surd$
	$w/t/l$				6/1/5	6/2/4	11/0/1
ND	p_s				1.0000	0.7539	0.0063 $\surd$
	p_w				0.8984	0.3359	0.0015 $\surd$
	$w/t/l$					6/0/6	8/0/4
SVM	p_s					1.0000	0.3877
	p_w					0.6353	0.3496
	$w/t/l$						10/2/0
sBoost-1	p_s						0.0020 $\surd$
	p_w						0.0020 $\surd$

Table 2
Averaged test errors of AB, WAB, ND, SVM, sBoost-1, and sBoost-2 using 5×2 cross-validation and 200 weak learners.

Noise	0%						5%						10%					
Data Name	*AB*	*WAB*	*ND*	*SVM*	*Sb1*	*Sb2*	*AB*	*WAB*	*ND*	*SVM*	*Sb1*	*Sb2*	*AB*	*WAB*	*ND*	*SVM*	*Sb1*	*Sb2*
Phoneme	0.193	0.192	0.220	0.250	0.188	**0.179**	0.220	0.215	0.243	0.278	0.214	**0.212**	0.260	0.253	0.271	0.292	0.253	**0.251**
Appendicitis	0.192	0.189	0.184	**0.145**	0.188	0.181	0.245	0.232	0.214	**0.160**	0.211	0.211	0.294	0.292	0.289	**0.181**	0.283	0.283
Pima	0.258	0.249	0.232	**0.230**	0.245	0.243	0.295	0.286	0.280	**0.257**	0.282	0.282	0.320	0.310	**0.284**	0.291	0.302	0.299
Breast	0.299	0.292	0.285	0.290	0.280	**0.272**	0.330	0.320	0.293	0.298	0.293	**0.288**	0.347	0.344	0.338	0.322	0.325	**0.320**
Tic-tac-toe	0.245	**0.166**	0.278	0.331	0.192	0.191	0.269	0.221	0.252	0.348	0.222	**0.191**	0.304	**0.242**	0.325	0.379	0.258	0.257
Magic	0.172	**0.165**	0.172	0.208	0.169	0.167	0.211	**0.206**	0.216	0.240	0.214	0.213	0.245	0.242	0.243	0.263	0.240	**0.239**
Heart	0.207	0.199	0.178	**0.169**	0.183	0.176	0.276	0.259	0.231	**0.202**	0.234	0.227	0.324	0.307	0.296	**0.236**	0.304	0.299
Australian	0.163	0.149	0.152	0.145	**0.143**	**0.143**	0.206	0.195	0.185	0.185	0.188	**0.184**	0.239	0.225	0.211	0.212	0.211	**0.210**
German	0.260	0.254	0.275	0.252	0.254	**0.250**	0.293	**0.281**	0.288	0.287	0.285	0.284	0.312	0.308	0.310	0.309	0.308	**0.306**
Twonorm	0.032	0.031	0.033	**0.024**	0.031	0.030	0.109	0.101	0.088	**0.075**	0.092	0.085	0.160	0.150	0.148	**0.119**	0.150	0.140
Wdbc	0.050	0.034	0.043	0.044	0.033	**0.031**	0.116	0.112	0.094	0.092	0.094	**0.090**	0.172	0.161	0.146	0.146	0.144	**0.140**
Sonar	0.208	0.204	0.246	0.229	0.204	**0.195**	0.269	**0.253**	0.297	0.313	0.255	**0.253**	0.317	0.313	0.340	0.322	0.311	**0.305**

Note: sB1 represents sBoost-1, and sB2 represents sBoost-2.

Table 5
Statistical significant comparisons of six boosting techniques with 10% noise over 12 data sets.

Methods		AB	WAB	ND	SVM	sBoost-1	sBoost-2
Mean		0.2745	0.2622	0.2668	0.2560	0.2574	0.2541
	w/t/l		12/0/0	9/0/3	8/0/4	12/0/0	12/0/0
AB	p_s		0.0005 √	0.1460	0.3877	0.0005 √	0.0005 √
	p_w		0.0005 √	0.1455	0.2334	0.0005 √	0.0005 √
	w/t/l			7/0/5	7/0/5	8/3/1	11/0/1
WAB	p_s			0.7744	0.7744	0.0391 √	0.0063 √
	p_w			0.8057	0.5186	0.0703 □	0.0176 √
	w/t/l				6/1/5	8/1/3	10/0/2
ND	p_s				1.0000	0.2266	0.0386 √
	p_w				.5918	0.1533	0.0210 √
	w/t/l					7/0/5	8/0/4
SVM	p_s					0.7744	0.3877
	p_w					0.9858	0.6636
	w/t/l						11/1/0
sBoost-1	p_s						0.0010 √
	p_w						0.0010 √

ree noise levels, 0%, 5% and 10%. This is because sBoost-2 can effectively reduce the overfitting problem of boosting and correctly classify potentially clean samples, in order to improve classification accuracy.

4. Conclusion

A selective boosting technique, sBoost, was developed in this work for more efficient pattern classification under low noise levels (up to 10%). The sBoost adaptively adjusts the effect of noisy samples on the data distribution using the proposed sample weight regulator, so as to reduce the overfitting problem. An error correction method was suggested in sBoost to detect and process potential clean samples in the test data, so as to further improve the classification accuracy. The effectiveness of the developed sBoost classifier has been verified by simulations using a series of benchmark data sets that are commonly used in this research field. Test results showed that the developed sBoost is an efficient classification tool. It can properly categorize the patterns and reduce the overfitting problem. In the future work, the proposed sBoost technique will be tested and improved using higher noise levels, for example up to 50%. It would state that this work mainly focuses on the boosting with weak base classifiers such as C4.5. The boosting techniques with strong base classifiers such as SVM will be studied in future research.

Acknowledgment

This work is supported in part by Natural Sciences and Engineering Research Council of Canada (NSERC) and eMech Systems Inc.

References

[1] R. Schapire, Y. Singer, Improved boosting algorithms using confidence-rated predictions, Mach. Learn. 37 (1999) 297–336.
[2] A. Takemura, A. Shimizu, K. Hamamoto, Discrimination of breast tumors in ultrasonic images using an ensemble classifier based on the AdaBoost algorithm with feature selection, IEEE Trans. Med. Imaging 29 (3) (2010) 598–609.
[3] J.H. Morra, Z. Tu, L.G. Apostolova, A.E. Green, A.W. Toga, P.M. Thompson, Comparison of AdaBoost and support vector machines for detecting Alzheimer's disease through automated hippocampal segmentation, IEEE Trans. Med. Imaging 29 (1) (2010) 30–43.
[4] M. Wang, X. Hua, R. Hong, J. Tang, G. Qi, Y. Song, Unified video annotation via multigraph learning, IEEE Trans. Circuits Syst. Video Technol. 19 (5) (2009) 733–746.
[5] Y. Gao, M. Wang, D. Tao, R. Ji, Q. Dai, 3-D object retrieval and recognition with hypergraph analysis, IEEE Trans. Image Process. 21 (9) (2012) 4290–4303.
[6] W. Hu, W. Hu, S. Maybank, AdaBoost-based algorithm for network intrusion detection, IEEE Trans. Syst. Man Cybern. Part B: Cybern. 38 (2) (2008) 577–583.
[7] T. Dietterich, An experimental comparison of three methods for constructing ensembles of decision trees: bagging, boosting, and randomization, Mach. Learn. 40 (2) (2000) 139–157.
[8] G.C. Cawley, N.L.C. Talbot, On over-fitting in model selection and subsequent selection bias in performance evaluation, J. Mach. Learn. Res. 11 (2010) 2079–2107.
[9] H. Masnadi-Shirazi, N. Vasconcelos, Cost-sensitive boosting, IEEE Trans. Pattern Anal. Mach. Intell. 33 (2) (2011) 294–309.
[10] L. Landesa-Vazquez, J.L. Alba-Castro, Double-base asymmetric AdaBoost, Neurocomputing 118 (2013) 101–114.
[11] Y. Gao, G. Ji, Z. Yang, J. Pan, A dynamic AdaBoost algorithm with adaptive changes of loss function, IEEE Trans. Syst. Man Cybern. – Part C: Appl. Rev. 42 (6) (2012) 1828–1841.
[12] J. Friedman, T. Hastie, R. Tibshirani, Additive logistic regression: a statistical view of boosting, Ann. Stat. 28 (2) (2000) 337–407.
[13] G. Ratsch, T. Onoda, K.R. Muller, Soft margins for AdaBoost, Mach. Learn. 42 (2001) 287–320.
[14] J. Cao, S. Kwong, R. Wang, A noise-detection based AdaBoost algorithm for mislabeled data, Pattern Recognit. 45 (2012) 4451–4465.
[15] Y. Gao, F. Gao, Edited AdaBoost by weighted kNN, , Neurocomput. 73 (16–18) (2010) 3079–3088.
[16] R. Schapire, Y. Freund, P. Bartlett, W. Lee, Boosting the margin: a new explanation for the effectiveness of voting methods, Ann. Stat. 26 (5) (1998) 1651–1686.
[17] J. Alcala-Fdez, L. Sanchez, S. Garcia, M.N.D. Jesus, S. Ventura, J.M. Garrell, J. Otero, C. Romero, J. Bacardit, V.M. Rivas, J.C. Fernandez, F. Herrera, KEEL: a software tool to assess evolutionary algorithms to data mining problems, Soft Comput. 13 (3) (2009) 307–318.
[18] P. Melville, N. Shah, L. Mihalkova, R.J. Mooney, Experiments on ensembles with missing noisy data, in: Proceedings of 5th International Workshop on Multiple Classifier Systems, 2004, pp. 293–302.
[19] R. Lowry, Concepts and Applications of Inferential Statistics, ⟨http://vassarstats.net/textbook/⟩, (retrieved 19 February 2014).
[20] J. Demsar, Statistical comparisons of classifiers over multiple data sets, J. Mach. Learn. Res. 7 (2006) 1–30.



De Z. Li received his B.Sc. degree in electrical engineering from Shandong University, Jinan, China, in 2008, and M.Sc. degree in control engineering from Lakehead University, Thunder Bay, ON, Canada in 2010.

From 2010 to 2011, he was a Research Associate at Lakehead University. He is currently a Ph.D. candidate with the Department of Mechanical & Mechatronics Engineering at University of Waterloo. His research interests include machinery health condition monitoring, signal processing, machine learning, mechatronic systems, linear/nonlinear system control and artificial intelligence.



Wilson Wang received his M. Eng. in industrial engineering from the University of Toronto, Toronto, ON, Canada, in 1998 and the Ph.D. in mechatronics engineering from the University of Waterloo, Waterloo, ON, Canada, in 2002.

From 2002 to 2004, he was a Senior Scientist with Mechworks Systems Inc. He joined the faculty of Lakehead University, Thunder Bay, ON, Canada, in 2004, where he is currently a Professor with the Department of Mechanical Engineering. His research interests include signal processing, artificial intelligence, machinery condition monitoring, intelligent control, and mechatronics.



Fathy Ismail received the B.Sc. and M.Sc. degrees in mechanical and production engineering, in 1970 and 1974, respectively, from Alexandria University, Egypt, and the Ph.D. degree from McMaster University, Hamilton, Ontario, Canada, in 1983.

He joined the University of Waterloo, Waterloo, Ontario, Canada, in 1983, and is currently a professor in the Department of Mechanical and Mechatronics Engineering. He has served as the chair of the department and the associate dean of the faculty of engineering for graduate studies. His research interests include machining dynamics, high-speed machining, modeling structures from modal analysis testing, and machinery health condition monitoring and diagnosis.