LPBoost with Strong Classifiers*

Yu K. Fang†
School of Computer Science and Engineering,
University of Electronic Science and Technology of China,
Chengdu 610054, Sichuan, P. R. China
E-mail: liusha.fang@gmail.com or fangyuke@uestc.edu.cn

Yan .Fu, Chong J. Sun, Jun. L Zhou
School of Computer Science and Engineering,
University of Electronic Science and Technology of China,
Chengdu 610054, Sichuan, P. R. China
E-mail: fuyan@uestc.edu.cn; sunchongjing2005@163.com; jlzhou@uestc.edu.cn

Abstract

The goal of boosting algorithm is to maximize the minimum margin on sample set. Based on minimax theory, the goal can be converted into minimize the maximum edge. This idea motivates LPBoost and its variants (including TotalBoost, SoftBoost, ERLPBoost) which solve the optimization problem by linear programming. These algorithms ignore the strong classifier and just minimize the maximum edge of weak classifiers so that all the edges of weak classifier are at most γ .This paper shows that the edge of strong classifier may be higher than the maximum edge of weak classifiers and proposes a novel boosting algorithm which introduced strong classifier into the optimization problem and constrained the edges of both weak and strong classifiers no more than γ . Furthermore, we justified the reasonability of introducing strong classifier using minimax theory.

We compared our algorithm with other approaches including AdaBoost, LPBoost, TotalBoost, SoftBoost, and ERLPBoost on the UCI benchmark dataset. In simulation studies we show that our algorithm converges faster than SoftBoost and ERLPBoost. In a benchmark comparison we illustrate the competiveness of our approach from the aspect of time consuming, and generalization error.

Keywords: boosting, strong classifier, soft margin, minimax theory, linear programming

1. Introduction

Boosting algorithms have shown considerable success in many fields, such as OCR (optical character recognition), face recognition, ranking or recommendation, text classification, natural language process. Boosting algorithms originated from PAC[1, 2] (Probably Approximately Correct) learning theory.

[†] Corresponding address: School of Computer Science and Engineering, University of Electronic Science and Technology of China, No.2006, XiYuan Road, Chengdu, 611731, Sichuan, China. Tel.:+86-028-61830375

Published by Atlantis Press

Kearns and Valiant (1989) postulated the boosting conjecture in the framework of PAC learning. In this method, a weak classifier (with success probability just a bit over 50 %) can be boosted into a strong one (strong classifier) in the sense that the training error of the new one would go to zero with a polynomial-time run time.

The AdaBoost algorithm, proposed by Freund and Schapire[3, 4, 5], is an efficient stage wise-optimum method, which boosts a series of simple and weak learners into one strong learner. The corrective update of sample distribution in AdaBoost can be view as a solution to minimize a relative entropy of current sample distribution versus uniform distribution, and this optimum problem is subject to some linear constraints that the edge of the last hypothesis is zero [6, 7]. One of the important properties of AdaBoost is that it has a decent iteration bound and approximately maximizes the margin of the examples [8].Similar algorithms including LogitBoost [9], AdaBoost$_v$*[10], all of which can be viewed as "corrective" family of boosting algorithms that enforce only a single constraint at each iteration [6] (the edge of the hypothesis must be at most γ, where γ is adapted).

However, a natural idea is to constrain the edges of all past hypotheses to be at most γ and otherwise minimize the relative entropy to the initial distribution. Basing on this idea, such algorithms (i.e. LPBoost[11, 12], TotalBoost[13]) were proposed and are called totally corrective in the sense that they optimize their weight based on all past hypotheses. Moreover LPBoost and TotalBoost are provable maximizing the margin with linear program. Nevertheless, unlike LPBoost, in which the upper bound γ on the edge is chosen to be as small as possible in each iteration, TotalBoost uses entropic regularization. Also, the γ decreased more moderately in TotalBoost.

Maximizing the hard margin is a provably approach for low generalization error [3] when the data is linearly separable. However, in case of inseparable data, maximizing the soft margin is a more robust and efficient choice. The soft margin maximization can be implemented via linear program with capping constraints for some small hard examples. Based on this idea, there are lots of boosting algorithms including AdaBoost with soft-margin [8], MadaBoost [14], v-arc [15, 16], SmoothBoost [17, 18], SoftBoost[19], corrective ERLPBoost[20], ERLPBoost[21]. This line of research culminated in SoftBoost and ERLPBoost, which both require $O(\frac{\ln N}{\delta^2})$ iteration bound within δ accuracy converging to the maximum minimum soft margin. More accurately, SoftBoost minimizes the relative entropy to the initial distribution subject to some linear constraints on the edges of all weak hypotheses in the past, and ERLPBoost added a factor $1/\eta$ of the relative entropy to the initial distribution and made a trade-off between maximizing the soft margin and minimizing the relative entropy which solves the main problem in SoftBoost: the generalization error decreases slowly in early iterations.

However, in total corrective family of algorithms, all of them update the sample distribution with weak classifiers ignoring the strong classifier. At each iteration in total corrective boosting algorithms, they just constrain the edges of existing weak classifiers to be at most γ even though the edge of strong classifier is larger than γ .It can be shown that the strong classifier edge is possibly larger than the maximum edge of all weak classifiers. Thus, a natural algorithm emerged: simply add edge constraint of strong classifier into edge-restraint conditions of ERLPBoost, which make the constraints stricter. Based on this, we proposed the StrongLPBoost which introduces the constraint of strong hypothesis to improve the convergence rate.

Our new algorithm is most similar to ERLPBoost because their goals are both to optimize the soft margin with all past hypotheses on condition of minimizing relative entropy. The most important difference is that we use tighter constraints. An important result of our work is to show that this strategy may help to increase the convergence speed.

The paper is organized as follows: in Section 2 we introduce the relevant notations, basic concepts and LPBoost. Section 3 deeply discusses the 4 problems existed in LPBoost and gives solution correspondingly before describing the detailed algorithms StrongLPBoost in Section 4.Finally, Section 5 contains our experimental evaluation of StrongLPBoost and its competitors. And the paper concludes with an outlook and discussion in Section 6.

2. Preliminaries

To conclude this section, we like to point the reader to Table 1 which summarizes our notations. And then some basic notations and relevant concepts about LPBoost will be presented in this section. Firstly, we will introduce two definitions: edge and margin, which are of a provable dual problem.

The edge γ_h of a weak classifier $h(x_m)$ on dataset $\mathrm{X} = \{(x_m, y_m);\ 1 \le m \le M\}$ is denoted by:

$\gamma_h = \sum_{m=1}^{M} d_m y_m h(x_m)$, where M is sample dataset size, d_m is the weight of a sample (x_m, y_m), $y_m \in \{-1,1\}$ and $h(x_m) \in \{-1,1\}$ .Similarly, error rate ε_h of h can be defined as: $\varepsilon_h = \sum_{m=1}^{M} d_m (y_m \neq h(x_m))$, then γ_h is a affine transformation of ε_h on $h(x_n)$: $\varepsilon_h(d) = \frac{1}{2} - \frac{1}{2}\gamma_h$.

Table 1. Notation for margins, edges, and LPBoost

Symbol	Description
X	Domain of examples
M	Number of examples e in X
(x_m, y_m)	m^{th} example
$y_m \in \{-1,1\}$	m^{th} label
$\overline{d}$	Distribution on examples
$\overline{w}$	Distribution on hypotheses
t	Iteration number
T	Final number of iterations
$u_{i,j} = y_i h^j(x_i)$	Convenient notation for combining labels and hypotheses
$U = \{u_{i,j}\}$, $i = 1:M$, $j = 1:T$	Error matrix
$u_{\bullet,j}, u_{i,\bullet}$	j^{th} row and i^{th} column of matrix U
$h^t(x_m) \in \{-1,1\}$	Prediction of hypothesis t on example x_m
ρ	Margin value
γ	Edge value
ε	Precision parameter
$\phi \in [0, \propto]^M$	Slack variable for soft margin problem
ν	Capping parameter
$H^t(x_n) \in [-1,1]$	Prediction of strong hypothesis t
η	Entropy regularized parameter
$sign$	Signal function

If $\gamma_h = 1$,the weak classifier $h(x_m)$ has zero error, and in the case of a random classifier, $\gamma_h = 0$ .Generally speaking, the higher the γ_h , the more useful the classifier is.

The final strong classifier of boosting algorithm is formed as: $f_w(x_m) = \sum_{t=1}^{T} w_t h^t(x_m)$, where T is the number of the weak classifier, and w_t corresponds to the weight of weak classifier h^t . We let ρ_m denote the margin of a sample (x_m, y_m) to f_w , and $\rho_m = y_m f_w(x_m)$ .As for the data set X , its margin is the minimum margin of set. Generally speaking, the margin represents for the generalization ability of a classifier. Obviously, more training data gives better generalization, and maximizing the margin can improve the ability to generalize.

$$\begin{array}{c|cccc|c}
 & h_1 & \dots & h_n & \overline{d} \\
x_1 & u_{1,1} & \dots & u_{1,n} & d_1 \\
\dots & \dots & \dots & \dots & \dots \\
x_m & u_{m,1} & \dots & u_{1,n} & d_n \\
\overline{w} & \mathrm{w}_1 & \dots & \mathrm{w}_n &
\end{array}$$

Fig. 1. Error matrix

It's noteworthy that edges are linear in the distribution over samples and margins are linear in the distribution over the current set of hypotheses. This optimization problem of maximizing margin can be converted into linear programming [11, 12].According to Refs. 11 we give a brief introduction of LPBoost. Given a fixed ensemble H and training set X , the error matrix U (as shown in Fig.1) contains entries $u_{i,j} = h_j(x_i)*y_i$ such that $u_{ij} = 1$ if $h_j(x_i)=y_i$ and $u_{ij} = -1$ if $h_j(x_i) \neq y_i$. In terms of U, the margin on sample i corresponds to the dot product: $\rho_i = \sum_{j=1}^{t} w_j u_{i,j} = \overline{w} \cdot \overline{u_{i\bullet}}$ .And the margin of a set of samples is denoted as: $\gamma_S = \min_i \overline{w} \cdot \overline{u_{i\bullet}}$, i.e. the minimum margin of all the samples. The goal is to find a weight vector $\overline{w}$ that obtains the largest possible margin subject to the constraints $w_j \geq 0$, $\sum_j w_j = 1$ .This is a maxi-min problem where we choose $\overline{w}$ to maximize γ_S subject to $w_j \geq 0$, $\sum_j w_j = 1$ .Fortunately, this problem can be turned into a linear programming problem as seen in (1).

$$\max_{w} \min_{i\in\{1,\dots,m\}} \overline{w}*u_{i\cdot}$$
$$s.t.\ w_j \ge 0, \sum_j w_j = 1 \qquad \text{or}$$

$$\max_{w} \rho$$
$$s.t.\ \sum_{j=1}^{t} w_j.u_{i,j} \ge \rho,\ for\ i = 1,\dots,m \qquad (1)$$
$$w_j \ge 0, \sum_j w_j = 1$$

The task and objective of LPBoost have been presented in detail. For the issue (1), its dual problem can be proven as listed in (2) via Von-Neumann's MiniMax theory [23]. Additionally, the objective values of (1) and (2) satisfy equation (3).

$$\min_{\overline{d}} \max_{j\in\{1,2,\dots,t\}} u_{\cdot j} * \overline{d}$$
$$s.t.\ d_i \ge 0, \sum_i d_i = 1 \qquad \text{or}$$

$$\min_{d} \gamma$$
$$s.t.\ \sum_{i=1}^{m} u_{i,j} d_i \le \gamma,\ for\ \ j = 1,\dots,t \qquad (2)$$
$$d_i \ge 0, \sum_i d_i = 1$$

$$\rho^* = \max_{w} \min_{i} \overline{w}*u_{i\cdot} = \ \gamma^* = \min_{\overline{d}} \max_{j} u_{\cdot j}.\overline{d} \qquad (3)$$

Following (2), another boosting approach can be deduced, alternatively, the distribution over samples can be computed by linear programming, which maximizes the margin over all the base hypotheses. In dual problem (2), the goal is that finding $(\overline{d},\gamma)$ to minimize γ subject to the constraints $\sum_i d_i u_{i,j} \le \gamma,\ j = 1,\dots,$ t and $\sum_i d_i = 1, d_i \ge 0$. Note that these notations or parameters have good natural explanation: $\sum_i d_i u_{i,j}$ means a score of weak classifier h_j on set X .Thus, LPBoost tried to find a distribution of samples to minimize the edge of best weak classifier, which increases the weights of misclassification samples and decreases the weights of accurate classification samples.

The SoftBoost and ERLPBoost represent for latest research in the variants of LPBoost. In order to ignore the bad effect of noise or difficult samples, SoftBoost adds the slack variant ζ_i for each x_i of these samples and maximizes the "soft margin": $\rho - \frac{v}{m}\sum_{i=1}^{m}\zeta_i$ to form the new primal problem as shown (4). Note that the relationship between capping and the hinge loss has long been exploited by the SVM community [24, 25].

Moreover, the relative entropy is introduced to ERLPBoost for updating the sample distribution smoothly and continually. The relative entropy is denoted as follow:

$$\Delta(\overline{d^t}, \overline{d^0}) := \sum_n d_n^t \ln\frac{d_n^t}{d_n^0},$$

where d^0, d^t are distributions of samples in the different iterations. Then the dual problem is listed as (5).

$$\max_{w,\rho,v} \rho - \frac{v}{m}\sum_{i=1}^{m}\zeta_i$$
$$s.t.\ \ w.u_i \ge \rho - \zeta_i, i = 1,\dots,m; \qquad (4)$$
$$\zeta_i \ge 0;$$
$$w_j \ge 0, \sum_j w_j = 1;$$

$$\min_{d_t,\gamma} \gamma + \eta.\Delta(\overline{d^t}, \overline{d^0})$$
$$s.t.\ \sum_{i=1}^{m} u_{i,j} d_i \le \gamma,\ for\ \ 1 \le j \le t; \qquad (5)$$
$$0 \le d_j \le \frac{v}{m}, \sum_j d_j = 1;$$

3. Proposition of StrongLPBoost

3.1. *Strong classifier and LPBoost*

Following the dual problems (2) and (5), we can see that only one weak classifier with maximum edge was chose to minimize the edge when updating the distribution $\overline{d}$. Their convergence rates can be improved by tightening the constraints of the optimization problem. The edge of strong classifier may be larger than the maximum bound of the weak classifiers. Thus, we can convert the strong classifier into a new weak one to add to the edge constraints of ERLPBoost and make the constraints stricter. In this way, the convergence speed can be accelerated.

We employ formal methods to describe it in detail. Referring to the above notations and definitions, the new notations defined as follows:

$$\gamma_{\min} = \min_j (\sum_{i=1}^{m} h_j(x_i) y_i d_i)\ \ for\ \ j = 1\ to\ t;$$

$\gamma_{\max} = \max_j (\sum_{i=1}^{m} h_j(x_i) y_i d_i) \ for \ j = 1 \, to \, t$;

$H(x_i) = \sum_{j=1}^{t} h_j(x_i) w_j$;

$H'(x_i) = \begin{cases} 1 & if \ H(x_i) > 0 \\ -1 & if \ H(x_i) \le 0 \end{cases}$;

Obviously, all the edges of the hypotheses satisfy inequation (6), but how about $H(x_i)$ and $H'(x_i)$?

$$\gamma_{\min} \le \sum_{i=1}^{m} h_j(x_i) y_i d_i \le \gamma_{\max}, \ for \ j = 1,...,m \qquad (6)$$

For the $H(x_i)$, we deduce like (7):

$$\begin{aligned} \sum_{i=1}^{m} H(x_i) d_i y_i &= \sum_{i=1}^{m}\sum_{j=1}^{t} h_j(x_i) w_j d_i y_i \\ &= \sum_{j=1}^{t} w_j \sum_{i=1}^{m} h_j(x_i) d_i y_i \end{aligned} \qquad (7)$$

Incorporating with (6), we can obtain (8):

$$\gamma_{\min} \le \sum_{i=1}^{m} H(x_i) d_i y_i \le \gamma_{\max} \qquad (8)$$

From inequation (8), we can find that the edge of strong classifier $H(x_i)$ is lower than the maximum edge of the weak classifiers. Using this logic, the error of final strong classifier is higher than minimum error of the weak ones. However, this conclusion conflicts with LPBoost and runs counter to the boosting theory. Thus, what's wrong with the above conclusion and deduction? Close inspection shows that the final strong classifier should not be $H(x_i)$ but $H'(x_i)$ .Alternatively, $H'(x_i)$ maybe not satisfy the inequation (9). Once the edge of strong classifier was found to may be higher than the weak, a natural algorithm emerged: simply adding the edge constraint of strong classifier to the LPBoost, which makes constrains tighter. And then, the convergence rate is improved further. The experiment section gives painstaking experimental verification.

$$\gamma_{\min} \le \sum_{i=1}^{m} H^{'}(x_i) d_i y_i \le \gamma_{\max} \qquad (9)$$

3.2. *Strong classifier and minimax theory*

This section shows the necessity of introducing the strong classifier from the point of minimax theory. The goal of boosting algorithm is to maximize the margin over sample set[25,] and this maximizing problem(left side of the equation (10)) can be converted into the minimax problem(right side of the equation (10)) according to the equation (3). In more specific terms, the minimax problem can be solved by two steps: the first is to find the weak classifier with maximum edge, and then, adjusting the weight $\overline{d}$ over samples to minimize the edge of classifier j (where $j = \arg\max_j u_{\cdot j}.\overline{d}$). All of the existing variants of LPBoost (including SoftBoost, ERLPBoost) basing on this idea find a weak classifier with maximum edge to minimize its edge.

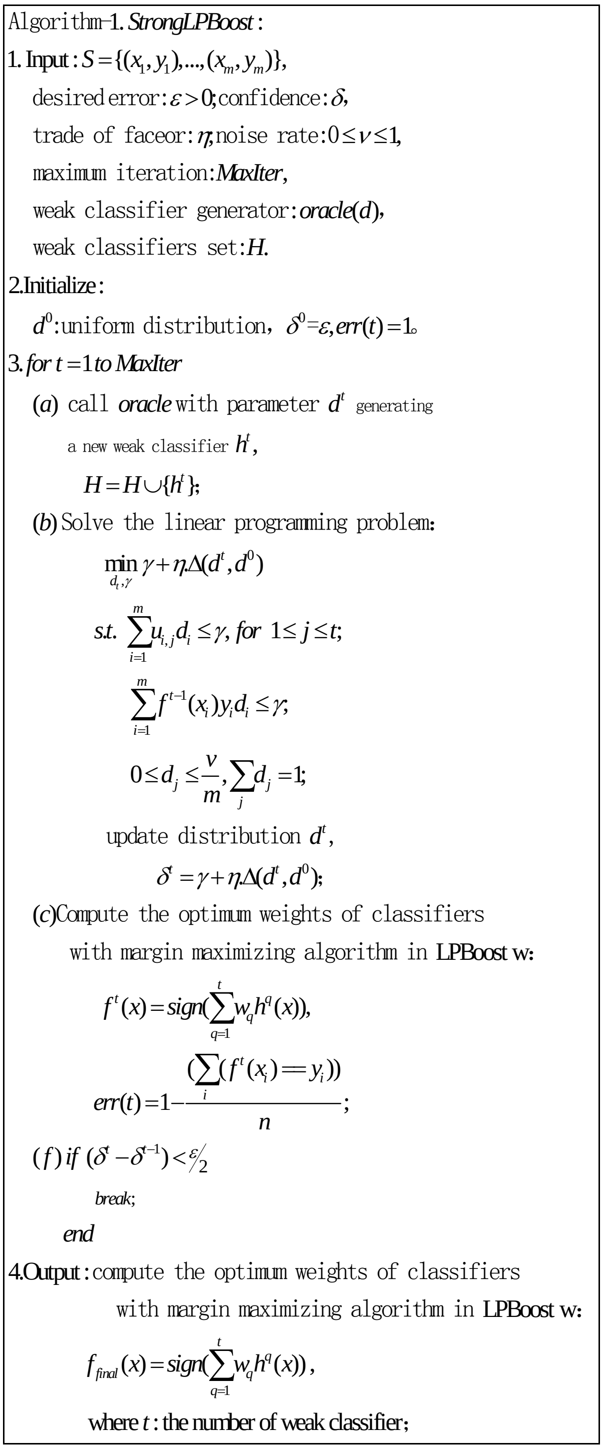
$$\max_{w} \min_{d} \overline{w} * U * \overline{d} = \min_{d} \max_{w} \overline{w} * U * \overline{d} \qquad (10)$$

We expand minimax equation (3) to equation (10) which can be proved to still hold via Von-Neumann's MiniMax theory [21, 26].Different with (3), equation (10) employ weighted mixed strategy rather than pure strategy. Observe that when $\overline{d}$ in the left side and $\overline{w}$ in the right side are values of base vector, we arrive at the equation (3). Obviously, equation (10) is better than (3) because the mixed strategy is more practical than pure strategy. In terms of (10), its left side is weighted combination of margin over examples, and the right side is weighted combination of edge over hypotheses. In this respect, strong classifier can be representative for combination of weak hypotheses. Consequently, it is reasonable to introduce the strong classifier into the equation (3). Specifically, when strong classifier H is converted into a new weak classifier h' , and add the h' into the cost matrix and solve the optimum problem (5).

4. StrongLPBoost

In the minimax problem that motivates the main algorithm of this paper, StrongLPBoost, a constraint of strong classifier is added to the constraint of linear programming (5). The modified linear programming problem is defined as minimizing problem (11) after appending the edge constraints of $H'(x_i)$ to the dual problem. The pseudo-code is shown in Fig.2. StrongLPBoost minimizes the relative entropy to initial distribution of samples when maximizing the soft margin.

At each iteration, weak hypothesis h^t is generated via calling oracle with parameter d^{t-1} , and then, new d^t and w_t can be obtained by solving the (11) and (5) respectively. The problem (11) only devotes to the update of sample distribution and (5) is employed to get



```
Algorithm-1. $StrongLPBoost$:
1. Input: $S=\{(x_1,y_1),...,(x_m,y_m)\}$,
   desired error: $\varepsilon>0$; confidence: $\delta$,
   trade of faceor: $\eta$; noise rate: $0\le\nu\le1$,
   maximum iteration: $MaxIter$,
   weak classifier generator: $oracle(d)$,
   weak classifiers set: $H$.
2.Initialize:
   $d^0$: uniform distribution, $\delta^0=\varepsilon, err(t)=1$。
3. $for\ t=1\ to\ MaxIter$
   $(a)$ call $oracle$ with parameter $d^t$ generating
        a new weak classifier $h^t$,
         $H=H\cup\{h^t\}$;
   $(b)$ Solve the linear programming problem:
           $\min_{d_t,\gamma}\gamma+\eta.\Delta(d^t,d^0)$
          $s.t.\ \sum_{i=1}^{m}u_{i,j}d_i\le\gamma,\ for\ 1\le j\le t;$
              $\sum_{i=1}^{m}f^{t-1}(x_i)y_id_i\le\gamma;$
              $0\le d_j\le\frac{\nu}{m},\sum_j d_j=1;$
          update distribution $d^t$,
              $\delta^t=\gamma+\eta.\Delta(d^t,d^0)$;
   $(c)$ Compute the optimum weights of classifiers
        with margin maximizing algorithm in LPBoost w:
          $f^t(x)=sign(\sum_{q=1}^{t}w_qh^q(x))$,
          $err(t)=1-\frac{(\sum_i(f^t(x_i)==y_i))}{n}$;
   $(f)\ if\ (\delta^t-\delta^{t-1})<\varepsilon/2$
          $break$;
     $end$
4.Output: compute the optimum weights of classifiers
           with margin maximizing algorithm in LPBoost w:
          $f_{final}(x)=sign(\sum_{q=1}^{t}w_qh^q(x))$,
          where $t$: the number of weak classifier;
```

Fig. 2. StrongLPBoost algorithm pseudo-code.

the weights of hypotheses. Thus, we can get the current strong classifier: $f^t(x)=\sum_{q=1}^{t}w_qh^q(x)$.

$$\begin{aligned}&\min_{d_t,\gamma}\ \gamma+\eta\cdot\Delta(\overline{d^t},\overline{d^0})\\&s.t.\ \sum_{i=1}^{m}u_{i,j}d_i\le\gamma,\ for\ \ 1\le j\le t;\\&\sum_{i=1}^{m}H'_{t-1}(x_i)y_id_i\le\gamma;\\&0\le d_j\le\frac{\nu}{m},\sum_j d_j=1;\end{aligned}\tag{11}$$

On one hand, our iteration bound for StrongLPBoost is the same to the bound proven for ERLPBoost since this algorithm just does the work on making the constraints of ERLPBoost tighter. On the other hand, the tighter constraints make $\overline{d}$ faster to reach to the ideal distribution than ERLPBoost, that is to say it needs at most $O(\frac{1}{\varepsilon^2}\ln\frac{N}{\nu})$ iterations to reach the optimum soft margin with ε error rate, where ν is the number of noise.

5. Experiment

In order to evaluate the performance of our new algorithm, we made an extensive comparison among the original AdaBoost, LPBoost, SoftBoost and ERLPBoost using decision tree as the weak classifier algorithm.

5.1. *Experiment Setup*

As previously used in Refs.8,18,19, except for Spiral and Banana datasets, all of our experiments utilize data from 9 benchmark data sets derived from the UCI and DELVE benchmark repository: banana ,breast cancer, diabetes, german, heart, image segment, ringnorm, new-thyroid, twonorm, waveform, spiral. However, these datasets can not be used as experiment data before preprocessing them as follows:

(1) A random partition into two classes is necessary for the data set that is not used for binary classification originally.

(2)We remove the samples with missing value so that all the attributes of the samples have values.

(3)The symbolic or nominal attributes in samples are mapped into the number from 1 to N, here N is the number of attribute values.

Finally, the experiment data descriptions are shown as Table II. Basing on the two dimensional Spiral and Banana datasets (Seen in Fig.3), it's more convenient to observe the differences of several boosting algorithms over edge, margin and iteration bound.

All the weak classifiers in these boosting algorithms are single decision tree. On each training set 5-fold-cross validation is used to train and test model for every

Table 2. Dataset description in the experiments

dataset	Attribute number	Original attribute number	Sample size for each class	Sample size
banana	2	2	1000/1000	2000
breast cancer	10	2	357/212	569
heart	14	2	150//120	270
image segment	19	7	990/1320	2310
ringnorm	21	2	3700/3700	7400
flare sonar	11	3	323/646	969
splice	60	3	1535/1655	3290
new-thyroid	5	3	150/75	215
titanic	4	4	1316/885	2201
twonorm	21	2	3700/3700	7400
waveform	21	3	2000/2000	6000
spiral	3	2	900/900	1800
german	20	2	700//300	1000
Diabetes	8	2	500/268	768

dataset (Training: Test=60%:40%). This method will make this comparison more robust and the results more reliable. For the algorithm basing on soft margin, we set: $\varepsilon = 0.01$, $v = 0.05$ and $\eta = \frac{2}{\varepsilon}\ln\frac{N}{v}$.

5.2. *Accuracy*

Firstly, we evaluate the accuracy of StrongLPBoost comparing with other 4 boosting algorithms over 11 datasets. In Table III the average generalization performance (with standard deviation) over the 11 datasets with 5 models for every boosting algorithm are shown. For the purpose of more extensive comparison, we introduce other evaluation criterions (including recall, fscore, fp_rate, specificity, matthews[24,29,30]).It's difficult to list all the results of 5 algorithms, so we just show the results of AdaBoost, LPBoost, StrongLPBoost

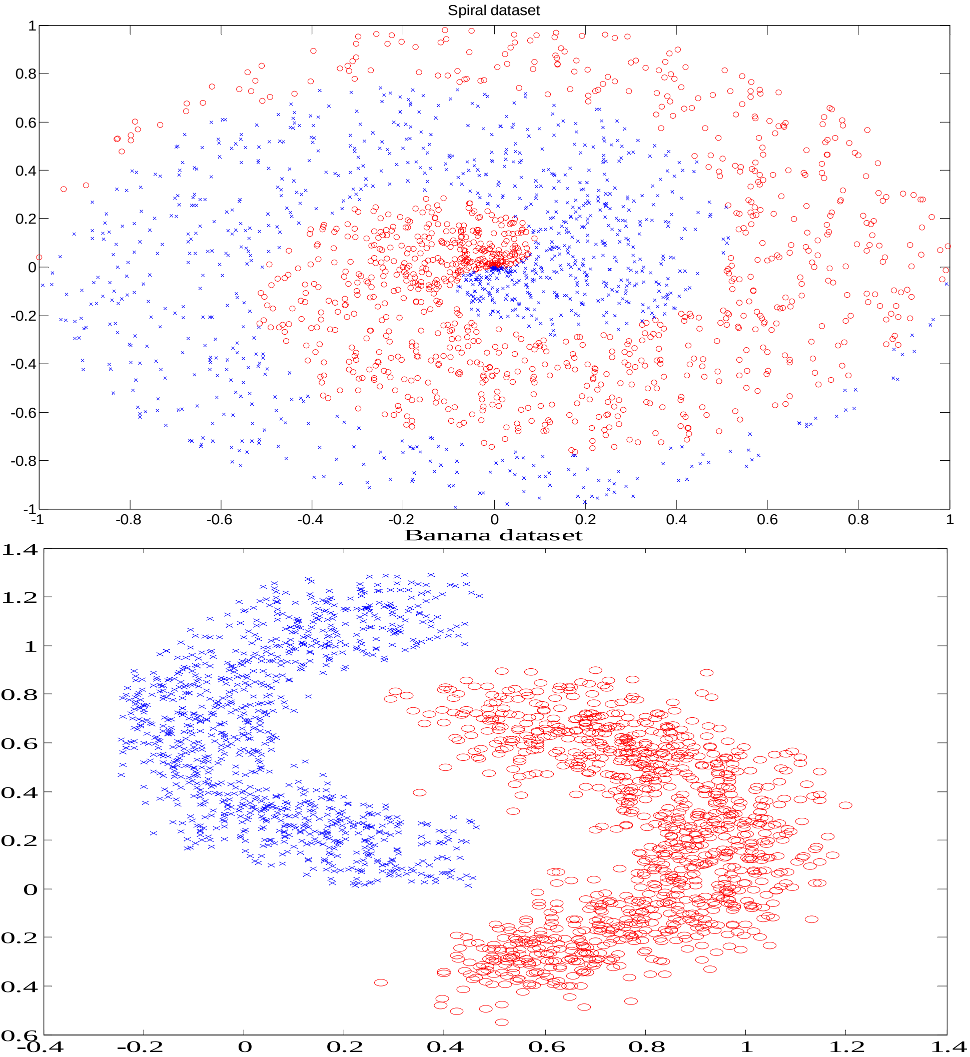


Fig. 3. Dataset of Spiral and Banana

Table 3. Accuracy of 5 algorithms over 11 datasets: the mean and variance of 5-fold-cross validation

Data set	AdaBoost	LPBoost	SoftBoost	ERLPBoost	**StrongLPBoost**
banana	1	1	1	1	**1**
breast cancer	0.9571±0. 0005	0.9607±0. 0009	0.9446±0. 0008	0.9732±0. 0007	**0.9643**±0. 0007
heart	0.8259±0. 0023	0.7667±0. 0049	0.7889±0. 0013	0.763±0. 0006	**0.7907**±**0. 0007**
image segment	0.9230±0. 0003	0.9572±0. 0001	0.8865±0. 0045	0.9752±0. 001	**0.9804**±0. 0002
ringnorm	0.745±0. 0016	0.6495±0. 0006	0.693±0. 0087	0.638±0. 0019	**0.8825**±0. 0007
new-thyroid	0.8837±0. 003	0.907±0. 006	0.8977±0. 0031	0.9023±0. 0046	**0.9395**±0. 0019
twonorm	0.9455±0. 0001	0.9480±0. 0001	0.8920±0. 0062	0.947±0. 0013	**0.9480**±0. 0003
waveform	0.9259±0	0.9275±0. 0001	0.92±0. 0002	0.9250±0. 0002	**0.9270**±0. 0002
german	0.742±0. 0001	0.733±0. 0009	0.7360±0. 0004	0.7360±0. 0006	**0.75**±0. 0006
diabetes	0.7519±0. 0004	0.7299±0. 0009	0.7143±0. 0026	0.7091±0. 0013	**0.7495**±**0. 0012**
spiral	0.8094±0. 0001	0.8094±0. 0005	0.7819±0. 002	0.8075±0. 0007	**0.8169**±0. 0002

(Note that:Bold marking the statistics of the StrongLPBoost and underscores marking the weaker results comparing with the other algorithms)

Table 4. Six evaluation criterions of 3 algorithms over 11 datasets: the average value of 5-fold-cross validation

	accuracy			recall			fscore			fp_rate			specificity			matthews		
Data set	ab	lb	**sb**	ab	lb	**sb**	ab	lb	**sb**	ab	lb	**sb**	ab	lb	**sb**	ab	lb	sb
banana	1	1	**1**	1	1	**1**	1	1	**1**	1	1	**1**	1	1	**1**	1	1	**1**
breast cancer	0.957	0.961	**0.964**	0.955	0.963	**0.967**	0.966	0.967	**0.972**	0.04	0.04	**0.041**	0.961	0.956	**0.959**	0.958	0.959	**0.963**
heart	0.826	0.767	**0.791**	0.826	0.808	**0.812**	0.846	0.7875	**0.82**	0.167	0.284	**0.257**	0.833	0.716	**0.726**	0.83	0.76	**0.81**
image segment	0.923	0.957	**0.98**	0.921	0.978	**0.981**	0.933	0.978	**0.983**	0.072	0.028	**0.0195**	0.928	0.972	**0.981**	0.928	0.975	**0.98**
ringnorm	0.745	0.65	**0.883**	0.705	0.604	**0.879**	0.783	0.733	**0.887**	0.163	0.174	**0.114**	0.837	0.826	**0.886**	0.771	0.715	**0.883**
new-thyroid	0.884	0.907	**0.94**	0.881	0.915	**0.949**	0.92	0.937	**0.957**	0.089	0.083	**0.083**	0.911	0.918	**0.918**	0.896	0.916	**0.933**
twonorm	0.946	0.948	**0.948**	0.95	0.947	**0.943**	0.944	0.948	**0.947**	0.0.058	0.043	**0.047**	0.942	0.957	**0.953**	0.946	0.948	**0.948**
waveform	0.926	0.928	**0.927**	0.931	0.922	**0.93**	0.931	0.93	**0.93**	0.0712	0.0665	**0.076**	0.929	0.934	**0.924**	0.93	0.928	**0.927**
german	0.742	0.733	**0.75**	0.76	0.793	**0.802**	0.834	0.815	**0.827**	0.359	0.437	**0.398**	0.641	0.563	**0.603**	0.7005	0.6778	**0.702**
diabetes	0.752	0.73	**0.75**	0.781	0.796	**0.78**	0.819	0.792	**0.813**	0.921	0.374	**0.410**	0.679	0.626	**0.743**	0.73	0.711	**0.724**
spiral	0.809	0.809	**0.817**	0.814	0.809	**0.819**	0.811	0.813	**0.819**	0.192	0.187	**0.182**	0.808	0.813	**0.818**	0.811	0.811	**0.818**

(Note that bold marking the statistics of the StrongLPBoost and underscores marking the weaker results comparing with the other algorithms;ab=AdaBoost,lb=LPBoost,sb=StrongLPBoost)

(As seen in Table 4).Note that except for the heart and diabetes datasets, the performance of StrongLPBoost is better than other boosting algorithms in almost all cases. For the two datasets, even though StrongLPBoost perform not as good as to AdaBoost, experimental results still show the competiveness compared with other variants of LPBoost.

5.3. *Margin and iteration bound*

Next, we compare the four maximizing margin algorithms from the aspects of the weak classifier relevance, margin, accuracy and iteration number. In order to make it easier to compare StrongLPBoost with SoftBoost and ERLPBoost, we use the Banana dataset in this experiment similar with the work by[18, 19]. Note that the reason we leave out AdaBoost is that it is not based on margin maximizing theory. Fig. 4 and Fig. 5 are the experimental result of 4 algorithms over Banana dataset at a time. Fig. 4 shows the margin value along with iteration number. The result shows that StrongLPBoost has the best convergence rate; ERLPBoost and LPBoost have similar convergence speed and the convergence of SoftBoost is worse when compared with other 3 boosting algorithms. Fig. 5 describes the accuracy boosting with the iteration about four algorithms. From the two figures, it can be seen that StrongLPBoost has fast convergence speed to close to the optimum soft margin with a small quantity of weak classifiers.

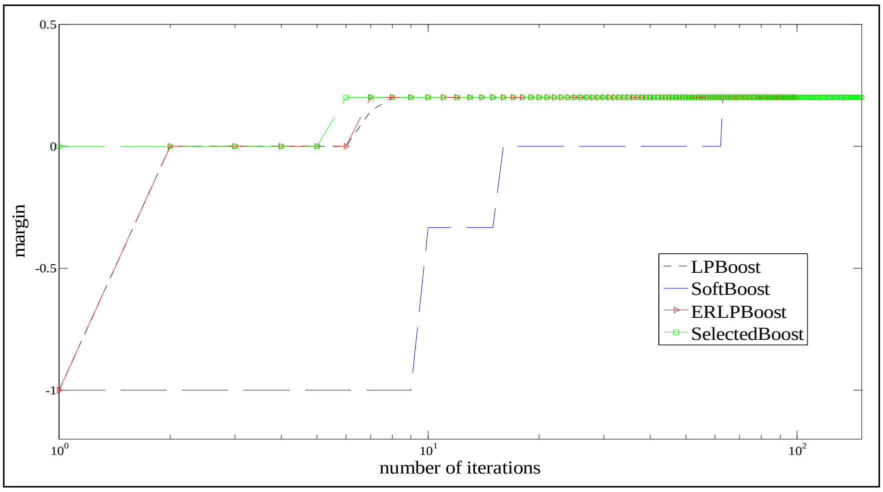


Fig. 4 Soft margin of LPBoost，SoftBoost，ERLPBoost and StrongLPBoost over Banana dataset

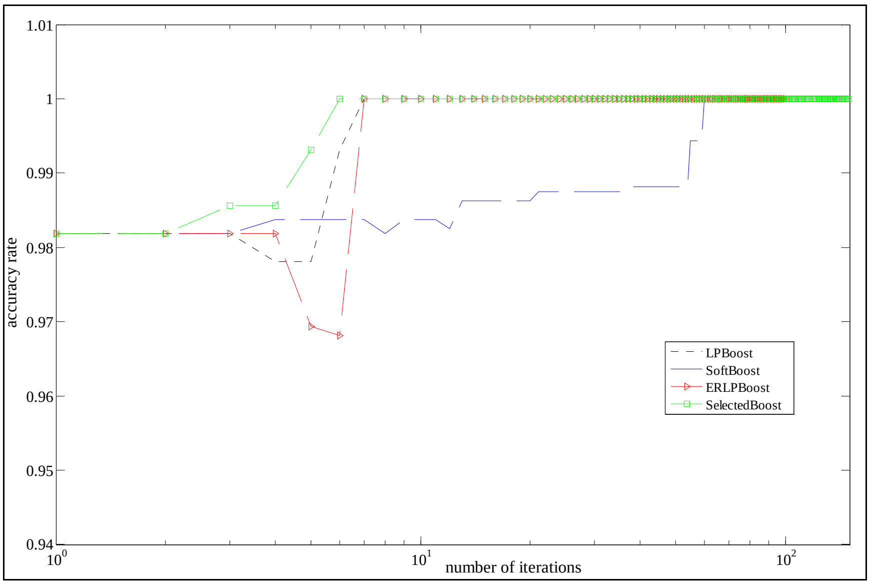


Fig. 5 .Accuracy rates of the LPBoost, SoftBoost, ERLPBoost and StrongLPBoost over Banana dataset

5.4. *Strong classifier edge constraint and convergence*

In this subsection, we show that the edge constraint of strong classifier has an impact on convergence. Here,

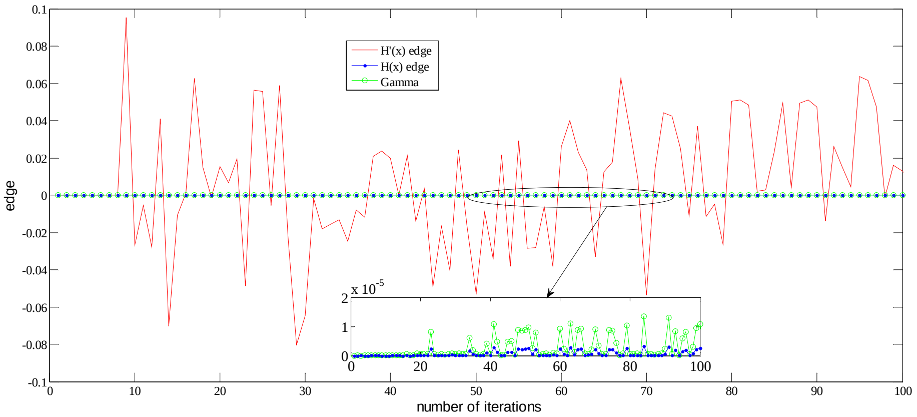


Fig. 6. Edges of $H^{'}(x)$, $H(x)$ and γ before adding the strong classifier $H^{'}(x)$

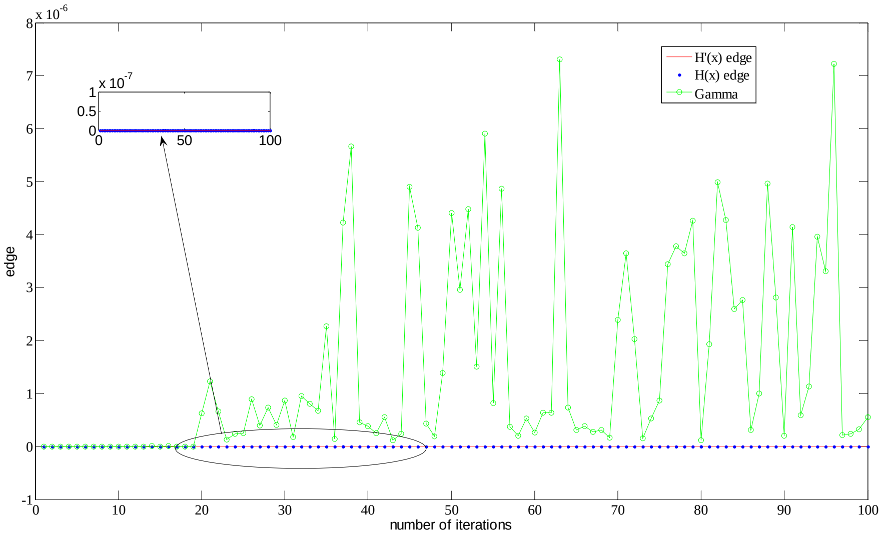


Fig. 7. Edges of $H^{'}(x)$, $H(x)$ and γ after adding the strong classifier $H^{'}(x)$

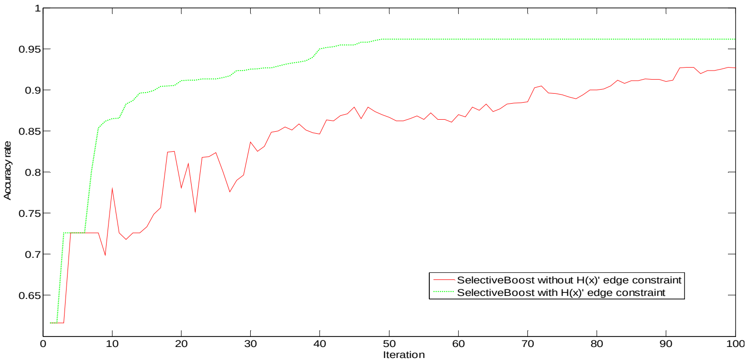


Fig. 8. Accuracy convergence rate before and after adding the strong classifier $H^{'}(x)$

the experimental data is based on spiral dataset (The reason why not using banana dataset is that four boosting algorithms are so easy to converge to the optimum of soft margin that it's difficult to observe the convergence change after adding strong classifier edge constraint.). Fig. 6 shows the edges of $H^{'}(x)$, $H(x)$ and γ along with iteration before adding the edge constraint of $H^{'}(x)$ .Red curve represents for the edge of $H^{'}(x)$,however, the edge of $H(x)$ and γ approach zero(10^{-6}) after solving the dual problem(5) at each iteration. Similar with our conclusion in section 3, the edge of $H(x)$ (blue curve) is always lower than γ , and moreover, the edge of $H^{'}(x)$ may be lower and higher than γ . When the edge constraints of $H^{'}(x)$ are added, the three edges are shown as Fig. 7, they all approach zero. Then, we can see that $H^{'}(x)$ takes effect. The convergence changes after adding $H^{'}(x)$'s edge constraint are shown in Fig. 8.

5.5. *Strong classifier and weak classifier*

Finally, this experiment shows how the constraint of strong classifier influences the weak classifier. In order to simplify the experiment, 800 samples in banana dataset are used. Fig. 9 shows the 15 weak classifiers in the 20 iterations generated by SoftBoost. There are amounts of relevance and redundancy during the 20 classifiers. On the contrary, there are just seven weak

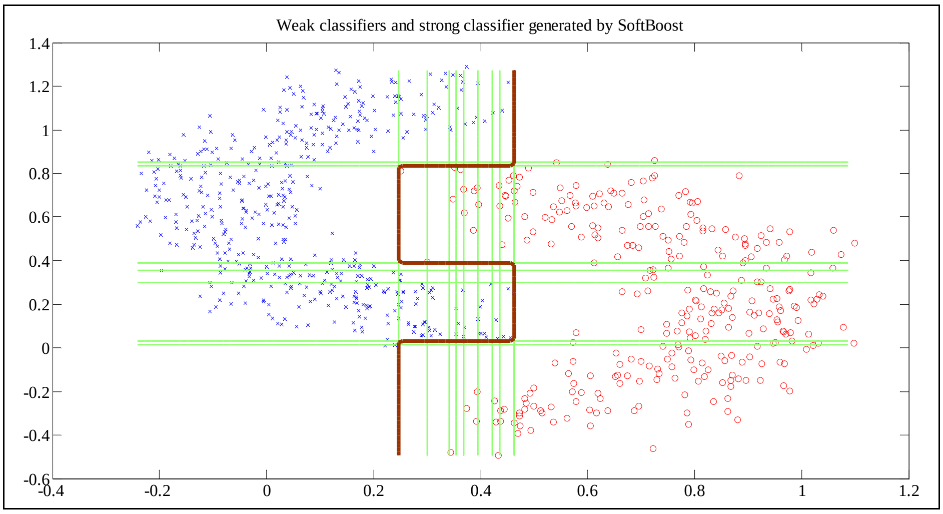


Fig. 9. Edges of $H^{'}(x)$, $H(x)$ and γ before adding the strong classifier $H^{'}(x)$

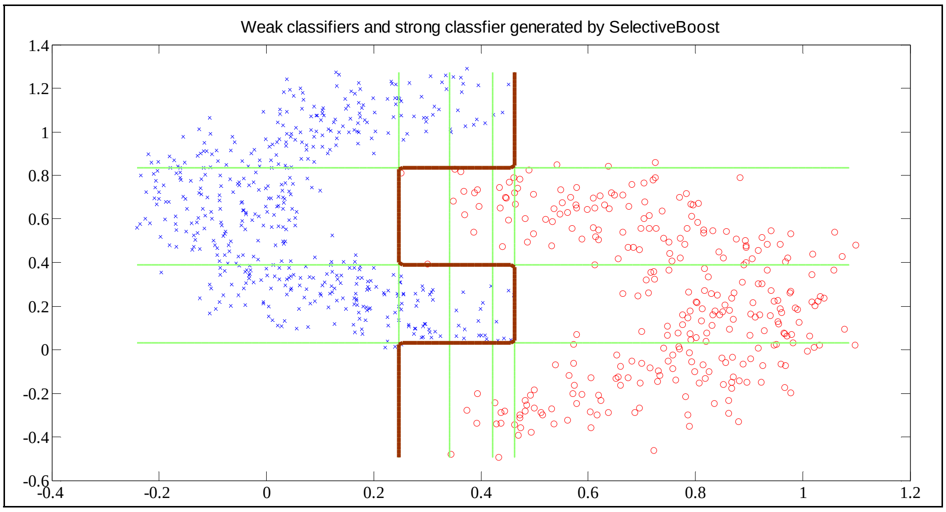


Fig. 10. Weak classifiers (thin) and strong classifier (thick) generated by SoftBoost over Banana dataset

classifiers generated by StrongLPBoost as shown in Fig. 10.

6. Conclusion

In this paper, we firstly review the research progress of boosting algorithm, and analysis the LPBoost and its variants from the point of minimax theory. The existing algorithms based on LPBoost originated from the minimax of pure strategy. The new distribution of samples is computed via solving the problem of minimizing the edge of weak classifier with maximum edge. And then, we expand the minimax from pure strategy to mixed strategy because the mixed strategy is more practical compared with pure strategy. According to the minimax of mixed strategy and ERLPBoost, we proposed a new boosting algorithm of simply adding the edge constraint of strong classifier to the problem of minimizing the maximum edge. Finally, we evaluate the StrongLPBoost with the experiments over the benchmark data sets and the experimental results show that the new algorithm of this paper has the higher convergence rate and accuracy compared with the popular boosting algorithms.

Our future work will concentrate on a continuing improvement of selection on weak classifiers for noisy real world applications, in addition, a further analysis of relation between strong classifier edge and margin convergence. Moreover, it is interesting to see how the techniques established in this work can be applied to find the support samples.

Acknowledgment

This work was supported in part by the National Natural Science Foundation of China under Grant No. 60903073,60973120,61003231; Programs for Science and Technology Development of Sichuan Province under Grant No. 2008GZ0009.

References

1. M. J. Kearns and U. Vazirani. An Introduction to Computational Learning Theory. MIT Press, Cambridge, MA, 1994.
2. L. Valiant. “A theory of the learnable”. Communications of ACM, 27(11):1134–1142, 1984.
3. Y. Freund and R. Schapire. “A decision-theoretic generalization of on-line learning and an application to boosting”. *Journalof Computer and System Sciences*, 55(1):119–139, Aug 1997
4. R.E.Schapire and Y. Singer. “Improved boosting algorithms using confidence-rated predictions”. In Proceedings of the Eleventh Annual Conference on Computational Learning Theory, pages 80–91, 1998.
5. R.E. Schapire, Y. Freund, P.L. Bartlett, and W.S. Lee. Boosting the margin: A new explanation for the effectiveness of voting methods. *The Annals of Statistics*, 26(5):1651–1686, 1998.
6. Kivinen, J., Warmuth, M.K.: Boosting as entropy projection. In: Proc. 12th Annu. Conf. on Comput. Learning Theory, pp. 134–144. ACM Press, New York (1999)
7. Lafferty, J.: Additive models, boosting, and inference for generalized divergences. In: Proceedings of the 12th Annual Conference on Computional Learning Theory, pp. 125–133. ACM Press, New York (1999)
8. G. Rätsch, T. Onoda, and K.-R. M¨uller. Soft margins for AdaBoost. Machine *Learning*, 42(3):287–320, 2001.
9. Friedman, J., Hastie, T., & Tibshirani, R. (2000). Additive Logistic Regression: a statistical view of boosting.Annals of Statistics, 2, 337–374.
10. G. Rätsch and M. K.Warmuth. Efficient margin maximizing with boosting. *Journal of Machine Learning Research*, 6:2131–2152, December 2005.
11. A. Demiriz, K.P. Bennett, and J. Shawe-Taylor. Linear programming boosting via column generation. *Machine Learning*, 46(1-3):225–254, 2002.
12. A.J. Grove and D. Schuurmans. Boosting in the limit: Maximizing the margin of learned ensembles. In *Proceedings of the Fifteenth National Conference on Artifical Intelligence*, 1998.
13. M.K. Warmuth, J. Liao, and G. Rätsch. Totally corrective boosting algorithms that maximize the margin. In *Proc. ICML ’06*, pages 1001–1008. ACM Press, 2006.
14. C. Domingo and O. Watanabe. Madaboost: A modification of Adaboost. In *Proc. COLT ’00*, pages 180–189, 2000.
15. G. Rätsch, B. Schölkopf, A.J. Smola, S. Mika, T. Onoda, and K.-R. M¨uller. Robust ensemble learning. In A.J. Smola, P.L. Bartlett, B. Schölkopf, and D. Schuurmans, editors, *Advances in Large Margin Classifiers*, pages 207–219. MIT Press, Cambridge, MA, 2000.
16. G. Rätsch. *Robust Boosting via Convex Optimization: Theory and Applications*. PhD thesis, University of Potsdam, Germany, December 2001.
17. Rocco A. Servedio. Smooth boosting and learning with malicious noise. *Journal of Machine Learning Research*, 4:633–648, 2003.
18. Rocco A. Servedio. Smooth boosting and learning with malicious noise. In 14th Annual Conference on Computational Learning Theory, COLT 2001 and 5th European Conference on Computational Learning Theory, EuroCOLT 2001, Amsterdam, The Netherlands, July 2001, Proceedings, volume 2111 of Lecture Notes in Artificial Intelligence, pages 473-489. Springer, 2001.
19. Warmuth, M.K., Glocer, K., Rätsch, G.: Boosting algorithms for maximizing the soft margin. In: Platt, J., Koller, D., Singer, Y., Roweis, S. (eds.) Advances in

Neural Information Processing Systems 20. MIT Press, Cambridge (2007)

20. Shai Shalev-Shwartz and Yoram Singer. On the equivalence of weak learnability and linear separability: New relaxations and efficient boosting algorithms. In Proceedings of COLT 2008, 2008.
21. M. Warmuth, K. Glocer, and S.V.N. Vishwanathan. Entropy regularized lpboost. *In Algorithmic Learning Theory (ALT)*, 2008.
22. L. Breiman. Arcing the edge. Technical report, Statistics Department, U. C. Berkeley, 1997.
23. Von Neumann J. Zur theorie der gesellschaftsspiele (on the theory of parlor games). Math. Ann., 100:295–320, 1928.
24. B. Schölkopf, A.J. Smola, R.C. Williamson, and P.L. Bartlett. New support vector algorithms. *Neural Comput.*, 12(5):1207–1245, 2000.
25. C. Cortes and V. Vapnik. Support-vector networks. *Machine Learning*, 20(3):273–297, 1995.
26. Yoav Freund and Robert E. Schapire. Game theory, on-line prediction and boosting.In COLT '96: Proceedings of the ninth annual conference on Computational learning theory, pages 325-332, New York, NY, USA, 1996. ACM.
27. Stephen Boyd and Lieven Vandenberghe. Convex Optimization. Cambridge University Press, 2004.
28. Pierre B, Brunak S, Chauvin Y, Andersen C, Nielsen H: Assessing the accuracy of prediction algorithms for classification: an overview. *Bioinformatics* Vol. 16(5). 412-424,2000
29. Sivakumari, S. ; Praveena Priyadarsini, R. ; Amudha, P. Accuracy evaluation of C4.5 and Naive Bayes classifiers using attribute ranking method . *International journal of computational intelligence systems*, 2(1):60–68, 2009(3).
30. Junlin Zhou, Aleksandar Lazarevic, Kuo-Wei Hsu, etc unsupervised learning based distributed detection of global anomalies，IJITDM，2010，Vol.9,No.6, 935-957