HAL open science

# Learning boosted asymmetric classifiers for object detection

Xinwen Hou, Cheng-Lin Liu, Tieniu Tan

## To cite this version:

Xinwen Hou, Cheng-Lin Liu, Tieniu Tan. Learning boosted asymmetric classifiers for object detection. Computer Vision and Pattern Recognition, Jun 2006, New York / USA, United States. pp.330-338, 10.1109/CVPR.2006.166 . inria-00120424

**HAL Id: inria-00120424**

**https://inria.hal.science/inria-00120424v1**

Submitted on 14 Dec 2006

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

# Learning Boosted Asymmetric Classifiers for Object Detection

Xinwen Hou, Cheng-Lin Liu, and Tieniu Tan
National Laboratory of Pattern Recognition
Institute of Automation, Chinese Academy of Science
100080, Beijing, P. R. China
{*xwhou, liucl, tnt*}*@nlpr.ia.ac.cn*

## Abstract

*Object detection can be posted as those classification tasks where the rare positive patterns are to be distinguished from the enormous negative patterns. To avoid the danger of missing positive patterns, more attention should be payed on them. Therefore there should be different requirements for False Reject Rate (FRR) and False Accept Rate (FAR) , and learning a classifier should use an asymmetric factor to balance between FRR and FAR. In this paper, a normalized asymmetric classification error is proposed for the task of rejecting negative patterns. Minimizing it not only controls the ratio of FRR and FAR, but more importantly limits the upper-bound of FRR. The latter characteristic is advantageous for those tasks where there is a requirement for low FRR. Based on this normalized asymmetric classification error, we develop an asymmetric AdaBoost algorithm with variable asymmetric factor and apply it to the learning of cascade classifiers for face detection. Experiments demonstrate that the proposed method achieves less complex classifiers and better performance than some previous AdaBoost methods.*

## 1 Introduction

Boosting [13] is one of the most successful recent techniques in machine learning and pattern classification. Its underlying idea is combining simple rules to form an ensemble so that the performance of the final ensemble is improved. Kearns and Valiant [9] proved the astonishing fact that simple rules, each performing only slightly better than random guess, can be combined to form an arbitrarily precise ensemble. Schapire [18] was the first to provide a provably effective (polynomial time) Boosting algorithm, then Freund and Schapire proposed AdaBoost [5, 4] which is the first step towards more practical Boosting algorithms. It is often referred to as Discrete AdaBoost for the adoption of Boolean weak classifiers. Many extensions of AdaBoost have been made in the past few years. Real AdaBoost [19] adopts confidence-rated weak classifiers to lower the upper-bounds of the classification error. Friedman *et al.* proposed LogitBoost [6] from the Logistic loss function and Gentle AdaBoost from the adaptive Newton descending. Other extensions can be found in [16, 17, 12, 15].

Boosting strategies have been successfully used in various real-world applications [14, 1, 20, 2]. Perhaps the most impressive application is face detection by Viola and Jones [23], where a cascade of successively more complex classifiers are learned based on simple Harr-like features. During detection, only those sub-windows which are not rejected by the preceding classifier are processed by the following classifier. If any classifier rejects the sub-window, no further processing is performed. This cascade structure makes the detection speed extremely high at 15 frames per second for $384 \times 288$ images, while achieving high detection rate. Li *et al.* [10] applied a pyramid cascade structure for multi-view face detection with high performance. Huang *et al.* [8] contrived a nested cascade structure for multi-view face detection and pose estimation. They reported the highest detection speed about 30ms to process a $320 \times 240$ image. In these face detection applications, classifiers are learned according to minimizing classification error, but the thresholds are tuned to get very low FRR and moderate FAR for effectively rejecting non-face patterns. Since the final goal deviates from minimizing classification error, the features selected by AdaBoost are not optimal for detection tasks.

To cope with the detection problem, Viola and Jones [22] introduced an asymmetric classification error: false rejects cost $k$ times more than false accepts. They applied an asymmetric pre-weighting technique to the initial distribution of the samples, but unfortunately the initial asymmetric weights are immediately absorbed by the first weak classifier because the AdaBoost process is too greedy. Ma and Ding [11] applied cost-sensitive learning technique [21, 3] to face detection, where there is also an asymmetric pre-weighting to the initial distribution, with a modified weight updating rule to avoid the weight absorb phenomenon. This

Proceedings of the 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06)
0-7695-2597-0/06 $20.00 © 2006 IEEE

modified weight updating rule pays more attention on positive samples, no matter what they are correctly classified or not, but it needs careful choice of asymmetric factors: they are unknown in advance and very application dependent, and need extensive trails for the best performance. For example, for the harder classification problem in the last few layers of the detector cascade, the asymmetric factors are often set close to 1.

In order to overcome these problems, we propose using variable asymmetric factor to adapt the requirement for specific applications and automatically obtain the best performance. In this paper, we define another asymmetric classification error which is equivalent to Viola and Jones's, but has more distinctive meanings. Minimizing this asymmetric error not only controls the ratio of FRR and FAR, but also limits the upper-bound of FRR. Based on this definition, we develop an asymmetric AdaBoost algorithm suitable for the task of rejecting negative samples. Features are selected according to minimizing the asymmetric error with variable asymmetric factor, and linearly combined into strong classifiers to achieve very low FRR and moderate FAR.

The rest of the paper is organized as follows: Section 2 introduces the asymmetric AdaBoost including definition, algorithm, and cascade structure of face detector. Section 3 and 4 give the experiment results and conclusions respectively. Convergence of the asymmetric AdaBoost algorithm is proved in the Appendix.

## 2 Asymmetric AdaBoost

### 2.1 Normalized Asymmetric Classification Error

For convention, assume training data are $\{(x_i, y_i) : i = 1, \ldots, N\}$ where $x_i$ are the samples with labels $y_i = 1$ or $-1$. For object detection tasks, the aim is to detect occurrences of a specific target (positives) from clutter background (negatives), where the probability of the former is substantially smaller than that of the latter. In this case, learning a classifier should use an asymmetric classification error criterion to balance between FRR and FAR [7], *i.e.*, FRR costs more than FAR:

$$\varepsilon = aFRR + bFAR \tag{1}$$

where $a \geq b > 0$. In order to compare the performance under different choices of $a$ and $b$, $\varepsilon$ should be normalized, *e.g.*, $\varepsilon = (aFRR + bFAR)/(a + b)$ according to our intuition, but we will adopt the following normalization definition as explained in Eq. (19) in the Appendix

$$\varepsilon = \frac{aFRR + bFAR}{\sqrt{ab}} \tag{2}$$

It can be easily checked that $\varepsilon$ is invariant under the substitution $a' = a/(a + b)$ and $b' = b/(a + b)$. The optimal classifier is achieved by finding the minimum of the above asymmetric error in (2), suppose it is $\varepsilon_0$ and the corresponding optimal $FRR$ and $FAR$ are $FRR_0$ and $FAR_0$. Suppose the probabilities of positives and negatives are respectively $P(1)$ and $P(-1)$, we could tune the threshold of the classifier and get $FRR = 0$, $FAR = P(-1)$, then $\varepsilon = \sqrt{b/a}P(-1)$. This shows

$$\varepsilon_0 \leq \sqrt{b/a}P(-1) \tag{3}$$

From the definition (2),

$$\varepsilon_0 = \frac{aFRR_0 + bFAR_0}{\sqrt{ab}} \geq \sqrt{a/b}FRR_0 \tag{4}$$

we can get

$$FRR_0 \leq \frac{b}{a}P(-1) \tag{5}$$

Eq. (5) shows that minimizing the asymmetric error can also control the upper bound of the optimal $FRR$. This characteristic is advantageous for object detection tasks where there is a requirement for low FRR.

Viola and Jones's asymmetric error [22] requires FRR cost $k$ times more than FAR, *i.e.*, $\varepsilon = FRR\sqrt{k} + FAR/\sqrt{k}$, which is a special case of our definition (2). It can be elegantly incorporated into the exponential upper bound of AdaBoost as $e^{y_i ln\sqrt{k}}$, results in pre-weighting over the initial distribution. But this approach is not effective because the initial asymmetric weights are immediately balanced and absorbed by the first weak classifier. An alternative is applying the asymmetric multiplier $e^{\frac{1}{M}y_i ln\sqrt{k}}$ in each round for a total $M$ round precess. Ma and Ding [11] also applied a pre-weighting technique to the initial distribution, $\frac{2c}{c+1}$ for positives and $\frac{2}{c+2}$ for negatives with $c > 1$. In order to avoid the weight absorb phenomenon, they modified the weight updating rule [3] to pay more attention on positives, whether they are correctly classified or not. Another problem is the choice of asymmetric factor $k$ and $c$: they are very application dependent and need extensive trials for the best performance. In order to overcome these problems, we propose using variable asymmetric factor to adapt the requirement for specific applications and automatically obtain the best performance.

### 2.2 Asymmetric AdaBoost Algorithm

Instead of changing the distribution by directly assigning unequal weights for positive and negative samples as in [22], we assign different values $a$ and $-b$ for positives and negatives samples. This can also indirectly changing the weights through exponential $e^{-ah(x_i)\alpha} =$

$e^{(1-a)h(x_i)\alpha}e^{-y_i h(x_i)\alpha}$ for $y_i = 1$ or $e^{bh(x_i)\alpha} = e^{-(1-b)h(x_i)\alpha}e^{-y_i h(x_i)\alpha}$ for $y_i = -1$, while it is easy for theoretical analysis. Because the normalized asymmetric classification error in definition (2) is invariant under the substitution $a' = a/(a+b)$ and $b' = b/(a+b)$, we will assume $b = 1 - a$ later in this paper, and develop an asymmetric AdaBoost algorithm as shown in Figure 1.

---

**Algorithm: Asymmetric AdaBoost**

**Input:** N labelled samples $(x_1, y_1), \ldots, (x_N, y_N)$ where $y_i \in \{a, -b\}$ for positive and negative samples. Weak classifier set $\{h\}$. $a \geq b > 0$, $a + b = 1$.

**Initialize:** $w_i^1 = \frac{1}{2n}, \frac{1}{2m}$ for $y_i = a, -b$ respectively, where $n$ and $m$ are the number of positive and negative samples respectively.

For $t = 1, \ldots, T$:

1. Normalize the weights, $w_i^t \leftarrow \frac{w_i^t}{\sum_{k=1}^{N} w_i^t}$
2. For each classifier $h$, compute the normalized margin $\gamma = \frac{1}{a}\sum_{i=1}^{N} w_i^t y_i h(x_i)$, tune $a$ to maximize $|\gamma|$
3. Choose the classifier $h_t$, with the maximal $|\gamma_t|$, and corresponding $a_t$
4. Compute the coefficient $\alpha_t = \frac{1}{2a_t}\ln\frac{1+\gamma_t}{1-\gamma_t}$
5. Update the weights, $w_i^{t+1} = w_i^t e^{-y_i h(x_i)\alpha_t}$, for $y_i = a_t, -b_t$

**Output:** The final strong classifier is:

$$H(x) = \begin{cases} 1 & \sum_{t=1}^{T} h_t(x_i)\alpha_t \geq \ln\frac{1-a}{a} \\ -1 & otherwise \end{cases}$$

---

**Figure 1. Asymmetric AdaBoost Algorithm.**

The following theorem justifies the convergence of the asymmetric AdaBoost algorithm.

**Theorem 1** *Using the notation of Figure 1, assume each $h_t$ has range $[-1, +1]$, then the normalized asymmetric classification error of H on training set is upper bounded by*

$$C\prod_{t=1}^{T}\sqrt{1-\gamma_t^2} \tag{6}$$

*Assume each $h_t$ has range $\{-1, +1\}$, then the normalized asymmetric classification error of H on training set is upper bounded by*

$$C\prod_{t=1}^{T} 2(\sqrt{A_t B_t} + \sqrt{C_t D_t}) \tag{7}$$

*where $C$ is a constant.*

Explanation of $\gamma_t, A_t, B_t, C_t, D_t$ and details of the proofs are in the Appendix. As shown in the Appendix, the asymmetric AdaBoost is a generalization of AdaBoost, and the later is the special case of the former in case of $a = b = a_t = b_t = \frac{1}{2}$.

## 2.3 Cascade Face Detector

Our face detector follows the same flowchart as in [23]. The input image is resized by a scaling factor $s$ iteratively and scanned by a $24 \times 24$ rectangular sub-window with a search step $\Delta$. Four types of Harr-like features are extracted from the sub-window as shown in Figure 2, and form a feature set of 45,396. The feature value is defined as the difference of the sum of the pixels in the white rectangle and the sum of the pixels in the gray rectangle. Weak classifiers are defined by thresholding the feature values. Strong classifiers are trained by our asymmetric AdaBoost algorithm to achieve very low FRR and moderate FAR. The successive strong classifiers form the final cascade face detector as shown in Figure 3. Only those sub-windows which pass through the former layer are fed into the next layer. A sub-window passing through all layers is accepted as a face pattern, while it is rejected at any layer results in immediate discard and needs no further process. This mechanism results in very high detection speed. At last, the overlapping face patterns detected in the image will be merged for the final result.

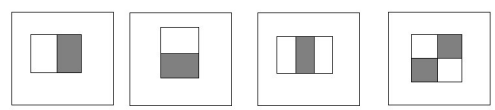

**Figure 2. Four types of Harr-like features**

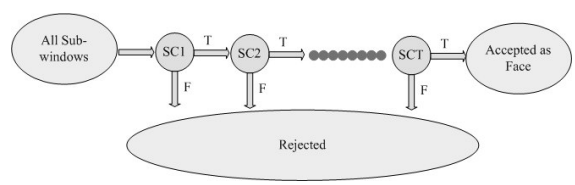


**Figure 3. The cascade structure of face detector. A series of $T$ strong classifiers (SC) are applied to each sub-window.**

# 3 Experiments

A total of 1,650 frontal face samples are collected from various source. Three images are generated from each sample by randomly rotated between $[-10°, +10°]$, shifted between $[-2, +2]$ pixels, and scaled between $[0.8, 1.2]$. Then these images are vertically mirrored and form a training set of 9,900 face samples. 9,900 non-face samples are randomly extracted from 18,042 images containing no faces.

All the samples are resized to $24 \times 24$ gray-scale images and normalized by their variances to reduce the influence of illumination.

For the training of cascade face detector by asymmetric AdaBoost algorithm, we set $a = 0.9$, $\frac{1}{2} \leq a_t \leq a$ in all layers. We use weak classifier $h_t \in \{-1, +1\}$ to speed up the optimization process of $a_t$ in step 2 in Figure 1. This can be done by directly computing the optimal parameters and bounds through Eq. (28), so the asymmetric factor $a_t$ are selected adaptively. As a result, there are 17 layers and total 2,174 features, achieving a detection rate of 98.3% and false accept rate of $1.5 \times 10^{-6}$ on the training set. By setting the scaling factor $s = 1.2$ and search step $\Delta = 1$, our face detector can scan a $320 \times 240$ image within 40 ms on PIV2.0 PC.

The MIT+CMU test set is used for evaluating the performance, which consists of 130 images containing 507 frontal faces. Some detection results are shown in Figure 4. By tuning the threshold of the final layer, the ROC curve of our detector is shown in Figure 5. For comparison, Table 1 lists the detection rates for various numbers of false positives for our detector as well as Viola's [23, 22] and Ma's [11] published previously, but we will not compare to those state-of-art results [8] where Real or Gentle AdaBoost is used. From Table 1, it can be seen that our detector performs better that other AdaBoost-based methods. The main reason is that in the learning process, we not only consider the requirement of asymmetric weights for positives and negatives, but also automatically selected the most appropriate asymmetric factor $a_t$ for each weak classifier. Besides, our detector has few features compared to Viola's 32 layers with 4,297 features and Ma's 20 layer with about 3,000 features.

**Figure 4. Some detection results on MIT+CMU test set.**

## 4 Conclusions

For object detection, disease diagnosis and many other applications, there are asymmetric costs for false positives and false negatives. Common classifier designing methods will not get satisfied results in these cases. Such type of asymmetric classification problem can be efficiently solved by minimizing asymmetric classification error and cascade structure of classifiers. According to minimizing the asymmetric classification error, classifiers in each layer achieve very low FRR and moderate FAR, while moderate FAR could be used to immediately reject part of negatives. With the cascade structure, the whole cascade will get low FRR and extremely low FAR, while achieving high precision and speed. In this paper, we propose a normalized asymmetric classification error, which can compare the performances under different choices of asymmetric factors. Minimizing it not only controls the ratio of FRR and FAR, but more importantly limits the upper-bound of FRR. The latter characteristic is coincide with the requirement for detection tasks where there is a need on such low upper-bound. Based on this normalized asymmetric classification error, we develop an asymmetric AdaBoost algorithm and apply it to the learning of cascade classifiers for face detection. This algorithm can adaptively choose the asymmetric factor for best performance in real applications. Comparative experiments demonstrate that the proposed method achieves less complex classifiers and better performance than previous symmetric and asymmetric AdaBoost methods.

## 5 Acknowledgements

This work is supported by the National Basic Research Program of China (Grant No. 2004CB318100).

## Appendix

Proofs of Theorem 1:

Let $y_i^t = a_t, -b_t$ for positive and negative samples, combining the weight normalizing step in Figure 1, the weight updating step can be expressed as

$$w_i^{t+1} = \frac{w_i^t e^{-y_i^t h_t(x_i)\alpha_t}}{Z_t} \tag{8}$$

| False positives | 10 | 31 | 50 | 65 | 78 | 95 | 110 | 167 |
|---|---|---|---|---|---|---|---|---|
| Our method | 90.5% | 91.9% | 93.1% | 93.9% | 94.1% | 94.5% | 94.7% | 95.3% |
| Ma-Ding | 90.1% | 91.3% | 92.5% | 93.1% | 93.3% | 93.5% | - | 94.1% |
| Viola-Jones(Asym) | - | 88.5% | 91.5% | 91.9% | 92.1% | 92.9% | 93.1% | 93.8% |
| Viola-Jones | 78.3% | 85.2% | 88.8% | 89.8% | 90.1% | 90.8% | 91.1% | 91.8% |

**Table 1. Detection rates for various numbers of false positives on the MIT+CMU test set.**

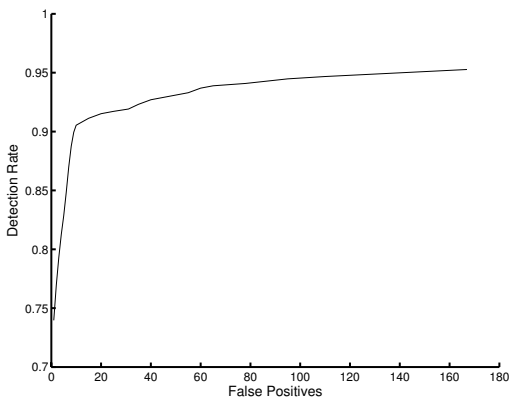


**Figure 5. Detection rate versus false positives of our detector on MIT+CMU test set.**

where $Z_t = \sum_{i=1}^{N} w_i^t e^{-y_i^t h_t(x_i)\alpha_t}$. By induction, we have

$$\sum_{i=1}^{N} w_i^1 e^{-\sum_{t=1}^{T} y_i^t h_t(x_i)\alpha_t} = \prod_{t=1}^{T} Z_t \tag{9}$$

We can assume $\alpha_t$ always no less that 0, or we can use $-h_t$ instead. For positive samples $y_i^t = a_t \geq \frac{1}{2}$ (because $a_t \geq b_t$)

$$\begin{aligned} -\sum_{t=1}^{T} a_t h_t(x_i)\alpha_t &= -\sum_{t=1}^{T} a_t[h_t(x_i)-1]\alpha_t - \sum_{t=1}^{T} a_t\alpha_t \\ &\geq -\sum_{t=1}^{T} \frac{1}{2}[h_t(x_i)-1]\alpha_t - \sum_{t=1}^{T} a_t\alpha_t \\ &= -\sum_{t=1}^{T} \frac{1}{2} h_t(x_i)\alpha_t + \sum_{t=1}^{T}(\frac{1}{2}-a_t)\alpha_t \end{aligned} \tag{10}$$

For negative samples $y_i^t = -b_t \geq -\frac{1}{2}$

$$\begin{aligned} \sum_{t=1}^{T} b_t h_t(x_i)\alpha_t &= \sum_{t=1}^{T} b_t[h_t(x_i)-1]\alpha_t + \sum_{t=1}^{T} b_t\alpha_t \\ &\geq \sum_{t=1}^{T} \frac{1}{2}[h_t(x_i)-1]\alpha_t + \sum_{t=1}^{T} b_t\alpha_t \\ &= \sum_{t=1}^{T} \frac{1}{2} h_t(x_i)\alpha_t + \sum_{t=1}^{T}(\frac{1}{2}-a_t)\alpha_t \end{aligned} \tag{11}$$

Suppose the final strong classifier is

$$H(x) = \begin{cases} 1 & \sum_{t=1}^{T} h_t(x_i)\alpha_t \geq K \\ -1 & \sum_{t=1}^{T} h_t(x_i)\alpha_t < K \end{cases} \tag{12}$$

Then for falsely rejected samples,

$$e^{-\sum_{t=1}^{T} a_t h_t(x_i)\alpha_t} > e^{-\frac{1}{2}K + \sum_{t=1}^{T}(\frac{1}{2}-a_t)\alpha_t} \tag{13}$$

Since it is independent of individual samples, we can assume

$$e^{-\frac{1}{2}K + \sum_{t=1}^{T}(\frac{1}{2}-a_t)\alpha_t} = aZ \tag{14}$$

For false accepted samples,

$$e^{\sum_{t=1}^{T} b_t h_t(x_i)\alpha_t} \geq e^{\frac{1}{2}K + \sum_{t=1}^{T}(\frac{1}{2}-a_t)\alpha_t} = (1-a)Z \tag{15}$$

From Eq. (14) and (15), we can get

$$K = \ln \frac{1-a}{a} \tag{16}$$

$$Z = \frac{e^{\sum_{t=1}^{T}(\frac{1}{2}-a_t)\alpha_t}}{\sqrt{a(1-a)}} \tag{17}$$

So

$$\begin{aligned} \sum_{i=1}^{N} w_i^1 e^{-\sum_{t=1}^{T} y_i^t h_t(x_i)\alpha_t} &\geq Z[aFRR + (1-a)FAR] \\ &= e^{\sum_{t=1}^{T}(\frac{1}{2}-a_t)\alpha_t} \frac{[aFRR + (1-a)FAR]}{\sqrt{a(1-a)}} \end{aligned} \tag{18}$$

In order to compare the performance under different choices of $a$, we define the normalized asymmetric classification error as

$$\varepsilon = \frac{aFRR + (1-a)FAR}{\sqrt{a(1-a)}} \tag{19}$$

For $h_t \in [-1, +1]$,

$$\begin{aligned} Z_t &= \sum_{i=1}^{N} w_i^t e^{-y_i^t h_t(x_i)\alpha_t} \\ &\leq \sum_{i=1}^{N} w_i^t [e^{-a_t\alpha_t}\frac{a_t + y_i^t h_t(x_i)}{2a_t} + e^{a_t\alpha_t}\frac{a_t - y_i^t h_t(x_i)}{2a_t}] \\ &= \sum_{i=1}^{N} w_i^t [e^{-a_t\alpha_t}\frac{1 + y_i^t h_t(x_i)/a_t}{2} + e^{a_t\alpha_t}\frac{1 - y_i^t h_t(x_i)/a_t}{2}] \end{aligned}$$

$$(20)$$

Let

$$\gamma_t = \frac{\sum_{i=1}^{N} w_i^t y_i^t h_t(x_i)}{a_t} \tag{21}$$

be the normalized margin, we have

$$Z_t \leq e^{-a_t \alpha_t} \frac{1+\gamma_t}{2} + e^{a_t \alpha_t} \frac{1-\gamma_t}{2} \tag{22}$$

Minimizing the right hand side, we get

$$Z_t \leq \sqrt{1-\gamma_t^2} \tag{23}$$

$$\alpha_t = \frac{1}{2a_t} \ln \frac{1+\gamma_t}{1-\gamma_t} \tag{24}$$

So the training error

$$\varepsilon \leq e^{\sum_{t=1}^{T}(a_t - \frac{1}{2})\alpha_t} \prod_{t=1}^{T} \sqrt{1-\gamma_t^2} \tag{25}$$

Since the weak classifiers selected become more and more trivial in the training process in practice, $\alpha_t$ will tend to 0, so $e^{\sum_{t=1}^{T}(a_t - \frac{1}{2})\alpha_t}$ will be bounded by a constant $C$, so

$$\varepsilon \leq C \prod_{t=1}^{T} \sqrt{1-\gamma_t^2} \tag{26}$$

When $a = b = a_t = b_t = \frac{1}{2}$, it can be seen that the asymmetric AdaBoost is really the ordinary AdaBoost.

For $h_t \in \{-1, +1\}$, the upper bound can be further investigated

$$\begin{aligned} Z_t &= \sum_{i=1}^{N} w_i^t e^{-y_i^t h_t(x_i)\alpha_t} \\ &= A_t e^{-a_t \alpha_t} + B_t e^{a_t \alpha_t} + C_t e^{b_t \alpha_t} + D_t e^{-b_t \alpha_t} \end{aligned} \tag{27}$$

where $A_t = P(y_i^t = a_t, h_t(x_i) = 1)$, $B_t = P(y_i^t = a_t, h_t(x_i) = -1)$, $C_t = P(y_i^t = -b_t, h_t(x_i) = 1)$ and $D_t = P(y_i^t = -b_t, h_t(x_i) = -1)$. Minimizing the right hand side of equation (27), we have

$$\begin{aligned} Z_t &\leq 2(\sqrt{A_t B_t} + \sqrt{C_t D_t}) \\ \alpha_t &= \frac{1}{2} \ln \frac{A_t D_t}{B_t C_t} \\ a_t &= \frac{\ln A_t / B_t}{2\alpha_t} \end{aligned} \tag{28}$$

So the training error

$$\varepsilon \leq C \prod_{t=1}^{T} 2(\sqrt{A_t B_t} + \sqrt{C_t D_t}) \tag{29}$$

The upper bound of $Z_t$ of AdaBoost in [5] is $2\sqrt{(A_t + D_t)(B_t + C_t)}$. Since $2(\sqrt{A_t B_t} + \sqrt{C_t D_t}) \leq 2\sqrt{(A_t + D_t)(B_t + C_t)}$, the asymmetric AdaBoost can get smaller training error bound.

## References


[1] M. Dettling and P. Bühlmann. How to use Boosting for tumor classification with gene expression data. *Bioinformatics*, 19:1061–1069, 2003.

[2] H. Drucker, R. E. Schapire, and P. Simard. Boosting performance in neural networks. *International Journal of Pattern Recognition and Artifical Intelligence*, 7:705–719, 1993.

[3] W. Fan, S. J. Stolfo, J. Zhang, and P. K. Chan. Adacost: Misclassification cost-sensitive Boosting.

[4] Y. Freund and R. E. Schapire. Experiments with a new Boosting algorithm. pages 148–156, 1996.

[5] Y. Freund and R. E. Schapire. A decision-theoretic generalization of on-line learning and an application to Boosting. *International Journal of Computer and System Sciences*, 5:119–139, 1997.

[6] J. Friedman, T. Hastie, and R. Tibshirani. Additive Logistic regression: A statistical view of Boosting. *Annals of Statistics*, 28:337–407, 2000.

[7] K. Grigoris and S. T. John. Optimizing classifers for imbalanced training sets. *Neural Information Processing Systems*, pages 253–259, 1998.

[8] C. Huang, Z. H. Ai, B. Wu, and S. Lao. Boosting nested cascade detector for multi-view face detection. pages 415–418, 2004.

[9] M. Kearns and L. G. Valiant. Cryptographic limitations on learning Boolean formulae and finite automata. *Journal of the ACM*, 41:67–95, 1994.

[10] S. Z. Li, L. Zhu, Q. Z. Zhang, A. Blake, J. H. Zhang, and H. Shum. Statistical learning of multi-view face detection. pages 67–81, 2002.

[11] Y. Ma and X. Q. Ding. Robust real-time face detection based on cost-sensitive AdaBoost method. pages 465–478, 2003.

[12] R. Meir, R. El-Yaniv, and S. Ben-David. Localized Boosting. pages 190–199, 2000.

[13] R. Meir and G. Rätsch. An introduction to Boosting and Leveraging. In *Advanced Lectures on Maching Learning*, volume LNAI 2600, pages 118–183. Springer-Verlag, 2003.

[14] T. Onoda, G. Rätsch, and K. R. Müller. A non-intrusive monitoring system for household electric appliances with inverters. 2000.

[15] N. Oza and S. Russell. Experimental comparisions of online and batch versions of Bagging and Boosting. pages 359–364, 2001.

[16] G. Rätsch and M. K. Warmuth. Marginal Boosting.

[17] G. Rätsch, M. K. Warmuth, S. Mika, T. Onoda, S. Lemm, and K. R. Müller. Barrier Boosting. pages 170–179, 2000.

[18] R. E. Schapire. The strength of weak learnability. *Machine Learning*, 5:197–227, 1990.

[19] R. E. Schapire and Y. Singer. Improved Boosting algorithms using confidence-rated predictions. *Machine Learning*, 37:297–336, 1999.

[20] H. Schwenk and Y. Bengio. Boosting neural networks. *Neural Computation*, 12:1869–1887, 2000.

[21] K. M. Ting and Z. Zheng. Boosting trees for cost-sensitive classifications.

[22] P. Viola and M. Jones. Fast and robust classification using asymmetric Adaboost and a detector cascade. *Neural Information Processing Systems*, pages 1311–1318, 2001.

[23] P. Viola and M. Jones. Rapid object detection using a boosted cascade of simple features. pages 1063–6919, 2001.




0-7695-2597-0/06 $20.00 © 2006 IEEE