# Multi-class AdaBoost with Hypothesis Margin

Xiaobo Jin Xinwen Hou Cheng-Lin Liu
National Laboratory of Pattern Recognition, Institute of Automation, Chinese Academy of Sciences
95 Zhongguancun East Road, Beijing 100190, P. R. China
{xbjin,xwhou,liucl}@nlpr.ia.ac.cn

## Abstract

*Most AdaBoost algorithms for multi-class problems have to decompose the multi-class classification into multiple binary problems, like the Adaboost.MH and the LogitBoost. This paper proposes a new multi-class AdaBoost algorithm based on hypothesis margin, called AdaBoost.HM, which directly combines multi-class weak classifiers. The hypothesis margin maximizes the output about the positive class meanwhile minimizes the maximal outputs about the negative classes. We discuss the upper bound of the training error about AdaBoost.HM and a previous multi-class learning algorithm AdaBoost.M1. Our experiments using feedforward neural networks as weak learners show that the proposed AdaBoost.HM yields higher classification accuracies than the AdaBoost.M1 and the AdaBoost.MH, and meanwhile, AdaBoost.HM is computationally efficient in training.*

## 1. Introduction

AdaBoost algorithm [4, 9] obtains a strong learner by combing multiple weak learners for the binary classification problem. For multi-class classification problems, most AdaBoost algorithms reduce the multi-class problem into multiple binary problems, such as AdaBoost.MH (AdaBoost with Multi-class Hamming Loss), AdaBoost.MR (AdaBoost with Ranking Loss) and LogitBoost [6], or generally by ECOC (Error Correction Output Coding). For problems with very large number of classes, the time and memory needed to construct and store all resulting binary classifiers can be prohibitive.

How do we construct multi-class AdaBoost algorithms directly based on multi-class weak classifiers? Freund and Schapire [4] proposed AdaBoost.M1, which extends directly the two-class AdaBoost algorithm to the multi-class algorithm with multi-class classifiers as weak learners. It needs the performance of all weak classifiers such that the error rate $\epsilon$ of each weak classifier is less than $1/2$. Eibl [2, 3] and Zhu [10] loosed the above restrictions to $\epsilon \leq 1 - 1/L$ where $L$ is the number of the classes and proposed multi-class algorithms, which are variants of AdaBoost.M1 from different views called AdaBoost.M1W and SAMME (Stagewise Additive Modeling using a Multi-class Exponential loss function), respectively.

Although the stump classifier as weak learners has obtained success in many applications, could we get better performance if we use weak learners more powerful than the decision stump? Freund [5] implemented Adaboost.M2 with C4.5 and random nearest neighbor as weak learners. [8, 7] implemented AdaBoost.M2 with neural networks as weak learners. They pointed out that most of the performance improvement for an ensemble comes from the first few classifiers, so we could get perfect performance with fewer classifiers.

In this paper, we propose a new AdaBoost framework, called AdaBoost.HM, with the hypothesis margin originated from LVQ [1] directly based on the multi-class weak classifiers. It can be seen that it minimizes the maximal output of the negative classes instead of the output sum of those and the former has a tight upper bound of all output about the negative classes. In addition, we give the upper bound of the training error about AdaBoost.HM. The experiments with the neural networks [8, 7](especially, multi-layer perceptron) as the weak learners show that AdaBoost.HM yields better performance than AdaBoost.M1 and AdaBoost.MH, since AdaBoost.HM has a tighter upper bound of the training error.

The rest of this paper is organized as follows: section 2 discusses two typical multi-class AdaBoost algorithms including AdaBoost.M1 and AdaBoost.MH; section 3 proposes a new Adaboost algorithm with hypothesis margin (AdaBoost.HM); section 4 demonstrates the performance of AdaBoost.HM; the last section summaries the total paper.

1051-4651/10 $26.00 © 2010 IEEE
DOI 10.1109/ICPR.2010.25

## 2. Functional Gradient Descent

Let us consider a labeled data set $D = \{(\mathbf{x}_n, y_n)|n = 1, 2, \cdots, N\}$, where $\mathbf{x}_n \in \mathcal{R}^d$ and $y_n \in \mathcal{Y} = \{1, 2, \cdots, L\}$ ($L$ is the number of classes). Providing that $\mathbf{F}(\mathbf{x})$ is the linear combination of the vector classifiers:

$$\mathbf{F}(\mathbf{x}) = \sum_{t=1}^{T} \alpha_t \mathbf{f}_t(\mathbf{x}), \tag{1}$$

where $\mathbf{f}_t$ comes from some fixed classifier set $\mathcal{F} = \{\mathbf{f}(\mathbf{x}; \boldsymbol{\beta}) \in [-1, +1]^L | \boldsymbol{\beta} \in \mathcal{R}^m\}$ and $\alpha_t \in \mathcal{R}^+$ is the classifier weight. For a training set $D$, we want to find $\mathbf{F}$ minimizing $\phi(\mathbf{F})$, so we search for $\mathbf{f}$ with greatest inner product with $-\nabla\phi(\mathbf{F})$ instead of $\nabla\phi(\mathbf{F})$:

$$\boldsymbol{\beta}_t = \text{argmax}_{\boldsymbol{\beta}} \langle -\nabla\phi(\mathbf{F}_{t-1}), \mathbf{f}(\boldsymbol{\beta}) \rangle, \tag{2}$$

When $\langle -\nabla\phi(\mathbf{F}_{t-1}), \mathbf{f}(\boldsymbol{\beta}_t) \rangle \leq 0$, the algorithm will terminate since $\mathbf{f}_t = \mathbf{f}(\boldsymbol{\beta}_t)$ no longer points to the downhill direction. Finally, we may choose the optimal step size in the direction $\mathbf{f}_t$:

$$\alpha_t = \text{argmin}_{\alpha} \phi(\mathbf{F}_{t-1} + \alpha \mathbf{f}_t) \tag{3}$$

In the following, we will introduce two representative algorithms: AdaBoost.MH and AdaBoost.M1. The former reduces the multi-class problems into the multiple binary problems and the latter builds directly on the multi-class weak classifier.

### 2.1. AdaBoost.MH

AdaBoost.MH is a multi-class algorithm with Hamming loss, which can be regarded as the average exponential loss on $L$ binary classification problems. It minimizes the following loss function:

$$\phi(\mathbf{F}) = \frac{1}{N} \sum_{n=1}^{N} \sum_{l=1}^{L} \exp(-Y[y_n = l] \mathbf{F}^l(\mathbf{x}_n)), \tag{4}$$

where $\mathbf{F}(\mathbf{x}) \in \mathcal{R}^L$ is the combination of multiple classifiers, $Y[\pi]$ is $+1$ when $\pi$ is true, otherwise is $-1$.

We refer to the following theorem about the upper bound of the training error from [9] for comparisons[1]:

$$\frac{1}{N} \sum_{n=1}^{N} I[H(\mathbf{x}_n) \neq y_n] \leq L \prod_{t=1}^{T} Z_t. \tag{5}$$

where $Z_t = \sum_{n=1}^{N} \sum_{l=1}^{L} D_{t-1}(n, l) \exp(-\alpha_t Y[y_n = l] \mathbf{f}_t^l(\mathbf{x}_n))$, $I[\pi] = 1$ if $\pi$ is true, otherwise is 0, and $H(\mathbf{x}) = \text{argmax}_{l \in \mathcal{Y}} \sum_{t=0}^{T-1} \alpha_{t+1} \mathbf{f}_{t+1}^l(\mathbf{x})$.

[1] In [9], the author further gives more tighter upper bound $L/2 \prod_{t=1}^{T} Z_t$.

**Algorithm 1** AdaBoost with Hypothesis Margin

Given the example set $(\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), \cdots, (\mathbf{x}_N, y_N)$, $\mathbf{x}_n \in \mathcal{X}, y_n \in \mathcal{Y}$.
Initialize the distribution: $D_0(n) = 1/N$.
**for** $t = 0, 1, \cdots, T-1$ **do**
  Train weak Learner with the distribution $D_t : \mathbf{f}_{t+1} : \mathcal{X} \to [0, +1]^L$.
  Obtain $u_{t+1}(\mathbf{x}_n) = \mathbf{f}_{t+1}^{y_n}(\mathbf{x}_n) - \max_{y \neq y_n} \mathbf{f}_{t+1}^{y}(\mathbf{x}_n)$ and Choose $\alpha_{t+1} \in \mathbb{R}$.
  Update: $D_{t+1}(n) = \frac{D_t(n) \exp(-\alpha_{t+1} u_{t+1}(\mathbf{x}_n))}{Z_{t+1}}$ where $Z_{t+1}$ is a normalization factor.
**end for**
Output the final hypothesis: $H(\mathbf{x}) = \text{argmax}_{l \in Y} \sum_{t=0}^{T-1} \alpha_{t+1} \mathbf{f}_{t+1}^l(\mathbf{x})$

### 2.2. AdaBoost.M1

AdaBoost.M1 directly uses multi-class classifiers as the weak learners and minimizes the loss function :

$$\phi(\mathbf{F}) = \frac{1}{N} \sum_{n=1}^{N} \exp(-\sum_{l=1}^{L} Y[y_n = l] \mathbf{F}^l(\mathbf{x}_n)). \tag{6}$$

For comparisons, we give the upper bound of AdaBoost.M1 in the form different from [4]:

$$\frac{1}{N} \sum_{n=1}^{N} I[H(\mathbf{x}_n) \neq y_n] \leq \prod_{t=1}^{T} Z_t \leq \prod_{t=1}^{T} \sqrt{1 - r_t^2} \tag{7}$$

where $Z_t = \sum_{n=1}^{N} D_{t-1}(n) \exp(-\alpha_t \sum_{l=1}^{L} Y[y_n = l] \mathbf{f}_t^l(\mathbf{x}_n))$ and

$$r_t = \sum_{n=1}^{N} \sum_{l=1}^{L} D_{t-1}(n) \mathbf{f}_t^l(\mathbf{x}_n) Y[y_n = l]. \tag{8}$$

## 3. AdaBoost with Hypothesis Margin

Hypothesis margin [1] is an important concept in learning vector quantization. It is defined as:

$$\mathbf{F}^{y_n}(\mathbf{x}_n) - \max_{l \neq y_n} \mathbf{F}^l(\mathbf{x}_n). \tag{9}$$

We can optimize the following exponential loss:

$$\phi(\mathbf{F}) = \frac{1}{N} \sum_{n=1}^{N} \exp(-h(\mathbf{x}_n)), \tag{10}$$

where $h(\mathbf{x}_n) = \mathbf{F}^{y_n}(\mathbf{x}_n) - \max_{l \neq y_n} \mathbf{F}^l(\mathbf{x}_n)$. It is clear that the loss maximizes the output of the positive class. Since $\max_{l \neq y_n} \mathbf{F}^l(\mathbf{x}_n) < \sum_{l \neq y_n} \mathbf{F}^l(\mathbf{x}_n)$, the loss minimizes the maximum output of the negative classes induces a tighter bound about the output of all negative classes, instead of the output sum of those in AdaBoost.M1. Algorithm 1 shows the framework of AdaBoost.HM.

### 3.1. Upper Bound of Training Error

The example $\mathbf{x}_n$ will be misclassified when $h(\mathbf{x}_n) \leq 0$, then the following inequality holds (assume $\alpha_t > 0$):

$$\begin{aligned} \sum_t \alpha_t u_t(\mathbf{x}_n) &= \sum_t \alpha_t \mathbf{f}_t^{y_n}(\mathbf{x}_n) - \sum_t \alpha_t \max_{l \neq y_n} \mathbf{f}_t^l(\mathbf{x}_n) \\ &\leq \sum_t \alpha_t \mathbf{f}_t^{y_n}(\mathbf{x}_n) - \max_{l \neq y_n} \sum_t \alpha_t \mathbf{f}_t^l(\mathbf{x}_n) \\ &= h(\mathbf{x}_n), \end{aligned} \tag{11}$$

Thus

$$I[h(\mathbf{x}_n) \leq 0] \leq \exp(-\sum_t \alpha_t u_t(\mathbf{x}_n)). \quad (12)$$

Where $u_t(\mathbf{x}_n) = \mathbf{f}_t^{y_n}(\mathbf{x}_n) - \max_{l \neq y_n} \mathbf{f}_t^l(\mathbf{x}_n)$. By unraveling the update rule, we can conclude the following upper bound about the training error:

$$\frac{1}{N}\sum_{n=1}^{N} I[h(\mathbf{x}_n) \leq 0] \leq \prod_{t=1}^{T} Z_t, \quad (13)$$

where $Z_t = \sum_{n=1}^{N} D_{t-1}(n)\exp(-\alpha_t u_t(\mathbf{x}_n))$.

### 3.2. Select $\alpha_t$

Suppose $\mathbf{f}(\mathbf{x}) \in [0,1]^L$, otherwise we can re-scale it in interval $[0,1]$. Following [9], we can derive the upper bound of $Z_t(\alpha_t > 0)$:

$$\begin{aligned} Z_t &= \sum_{n=1}^{N} D_{t-1}(n) e^{-\alpha_t u_t(\mathbf{x}_n)} \\ &\leq \frac{1+r_t}{2} e^{-\alpha_t} + \frac{1-r_t}{2} e^{\alpha_t}, \end{aligned} \quad (14)$$

where $r_t = \sum_n D_{t-1}(n) u_t(\mathbf{x}_n)$. Next, we can analytically choose $\alpha_t$ in the right hand side of (14)

$$\alpha_t = \frac{1}{2}\ln\left(\frac{1+r_t}{1-r_t}\right). \quad (15)$$

Plugging into (14), this choice gives the upper bound:

$$Z_t \leq \sqrt{1-r_t^2}. \quad (16)$$

When $r_t > 0$, then the weight $\alpha_t > 0$. So we can try to find $\mathbf{f}_t$ to maximize $r_t$:

$$r_t = \sum_n D_t(n)(\mathbf{f}_t^{y_n}(\mathbf{x}_n) - \max_{l \neq y_n} \mathbf{f}_t^l(\mathbf{x}_n)). \quad (17)$$

### 3.3. Discussions

We have noted that the upper bounds about the training error of multiple binary AdaBoost algorithms such as AdaBoost.MR [9] ($(L-1)\prod_t Z_t$) and AdaBoost.MH ($L\prod_t Z_t$) are relevant to the number of the classes, which will make the upper bound nonsense with the increasing class number (since the training error is always less than 1). But the upper bounds of multi-class AdaBoost algorithm (AdaBoost.M1, AdaBoost.HM) is independent of $L$. Furthermore, AdaBoost.HM can obtain larger $r_t$ than AdaBoost.M1 by considering $\mathbf{f}_t^{y_n}(\mathbf{x}_n) - \max_{l \neq y_n} \mathbf{f}_t^l(\mathbf{x}_n) \geq \sum_{l=1}^{L} \mathbf{f}_t^l(\mathbf{x}_n) Y[y_n = l]$ (see (8) and (17)). So AdaBoost.HM can obtain tighter upper bound than AdaBoost.M1. Table 1 gives the comparisons on the loss functions, the output ranges of the weak learner and the upper bounds of the training error of the discussed algorithms.

**Table 1. Comparisons on three algorithms**

| Algorithm | Loss on $\mathbf{x}_n$ | $\mathbf{f}(\mathbf{x}_n)$ | Upper Bound |
|---|---|---|---|
| AdaBoost.MH | $\sum_{l=1}^{L}\exp(-Y[y_n = l]F^l(\mathbf{x}_n))$ | $\{-1,+1\}^L$ | $L\prod_t Z_t$ |
| AdaBoost.M1 | $\exp(-\sum_{l=1}^{L} Y[y_n = l]F^l(\mathbf{x}_n))$ | $\{0,+1\}^L$ | $\prod_t Z_t$ |
| AdaBoost.HM | $\exp(-[F^{y_n}(\mathbf{x}_n) - \max_{l \neq y_n} F^l(\mathbf{x}_n)])$ | $[0,+1]^L$ | $\prod_t Z_t$ |

When the training error rate of the weak learner is near zero such as MLP (multi-layer perceptron), AdaBoost may terminate after combining a few learners. Besides that, AdaBoost may degenerate with MLP as the weak learners. For example, suppose the outputs of MLP are normalized between 0 and 1, and the outputs for all classes on one sample are larger than 0.5 or less than 0.5, then AdaBoost.MH with one MLP as the weak learner may probably misclassify the example since it will produce the identical labels for each class. But MLP may classify the example correctly for it returns the label having maximum output. Unlike AdaBoot.MH, AdaBoost.HM provides that the weak learners output the hypothesis margins as the difference between the output of positive class and the maximal output of the negative class, and AdaBoost.HM with one MLP weak learner will give the same decisions as the single MLP.

## 4. Experiment Results

We compared the performance of the algorithms on 23 UCI datasets with MLP and decision tree (DT) as the weak learners. Results were averaged over ten 10-fold cross validation for 15 small datasets and 8 pre-partitioned large datasets. For the MLP, all data-sets were normalized by the maximum standard deviation. The initial learning rate $\tau$ was set to $\{0.01, 0.02, 0.05, 0.1, 0.5\}$ and the iteration time was set to 40. The number of hidden nodes $H$ was selected from $\{1,2,3,4,5\}$ for the 15 small datasets whiling for the 8 large datasets it was chosen from $\{100, 120, 150, 180, 200\}$. The training parameters and the model parameters were optimized in space $(\tau, H)$ by cross validation on the training set. After optimizing MLP, AdaBoost algorithms build the ensemble of ten MLPs. In addition, we set the number of the weak learners to 100 for the DT.

Table 2 shows the average accuracy and the average rank [2] on all datasets, where AdaBoost.M1W comes from [2, 10] and the highest accuracy of the algorithms

[2]The top rank (highest accuracy) on a dataset is scored 1, second rank scored 2, and so on.

**Table 2. Classification accuracy of AdaBoost and MLP algorithms (%). The last three rows show the average accuracy, the average rank and the sign-rank value.**

| DataSet | M1W +DT | MH +DT | MLP | M1 +MLP | MH +MLP | HM +MLP |
|---|---|---|---|---|---|---|
| Breast | **96.02** | **96.02** | 94.45 | 93.82 | 94.32 | 94.23 |
| Diabetes | 75.48 | 75.47 | **76.47** | 76.19 | 74.82 | 75.74 |
| Ionosphere | 90.91 | 90.43 | 90.09 | 89.80 | **90.94** | 90.82 |
| Pima | 75.79 | 75.79 | 76.09 | 76.01 | 74.23 | **76.55** |
| Sonar | 84.00 | 83.94 | 81.78 | 78.53 | 81.28 | **84.24** |
| Balance | 90.74 | 92.14 | 93.71 | 92.69 | **97.25** | 97.23 |
| Dermatology | 57.15 | **96.98** | 90.40 | 94.76 | 95.42 | 95.53 |
| Glass | 56.60 | **72.88** | 62.26 | 62.70 | 64.65 | 64.65 |
| Iris | 93.73 | 94.20 | 96.47 | 95.80 | 96.67 | **97.07** |
| Lymphography | 69.72 | **83.81** | 80.34 | 75.35 | 76.32 | 76.73 |
| Shuttle-small | 93.45 | **99.84** | 99.56 | 99.16 | 99.55 | 99.67 |
| Vehicle | 65.22 | 74.13 | 82.40 | 78.78 | 82.70 | **84.29** |
| Waveform-21 | 82.87 | 83.17 | 86.70 | 86.97 | 86.97 | **87.23** |
| Wine | 96.23 | 97.70 | **98.14** | 97.14 | 97.68 | 97.92 |
| Yeast | 49.18 | 58.46 | 59.33 | 52.15 | 57.29 | **60.04** |
| Isolet | 25.72 | 91.60 | 95.57 | 92.88 | 93.91 | **95.96** |
| Letter | 39.78 | 76.82 | 88.36 | 74.70 | 96.42 | **97.10** |
| Mnist | 72.03 | 87.65 | 96.86 | 92.83 | 96.01 | **97.86** |
| Optdigit | 81.91 | 93.82 | 96.10 | 96.22 | 93.71 | **96.72** |
| Pendigit | 67.09 | 91.85 | 92.14 | 96.63 | 96.68 | **96.83** |
| Satimage | 68.40 | 86.85 | 88.80 | 89.25 | 89.50 | **89.55** |
| Thyroid | 94.57 | **99.36** | 96.65 | 98.13 | 97.84 | 98.10 |
| Usps | 78.48 | 88.54 | 92.87 | 91.38 | 91.23 | **93.82** |
| Mean Accuracy | 73.99 | 86.58 | 87.19 | 85.97 | 88.08 | **89.05** |
| Mean Rank | 5.30 | 3.57 | 3.09 | 3.98 | 3.30 | **1.76** |
| Sign-Rank | 0.0001 | 0.0264 | 0.0019 | 0.0001 | 0.0001 | / |

is highlighted in boldface. From Table 2, we find that AdaBoost.HM+ MLP achieves the highest average accuracy and shows great improvements in contrast with the single MLP on some datasets such as Letter and Pendigit. Although AdaBoost.MH + MLP shows similar performance as AdaBoost.HM+MLP, it is superior than the latter on most datasets. We can see that among six algorithms, AdaBoost.HM+MLP also obtains the highest rank (1.76), followed by MLP (3.09), AdaBoost.MH+MLP(3.30) and AdaBoost.MH+DT (3.57).

We further use sign-rank test to compare AdaBoost.HM with other algorithms in the statistical significance (the last row in Table 2) . It is found that AdaBoost.HM+MLP achieves the statistical significance in comparison with AdaBoost.MH+MLP ($p = 0.0001$) and AdaBoost.MH+DT ($p = 0.0264$).

We summarize the time complexity. Since there are only a few MLP learners in AdaBoost algorithms, we approximate the number of the weak learners to $O(1)$. It can be concluded that the training time is proportional to $dLN$ in AdaBoost.HM+MLP and AdaBoost.MH+DT, where $d$ is the dimension and $N$ is the sample number of the dataset. So AdaBoost.HM+MLP is scalable the same as AdaBoost.MH+DT for the large classes problem. But the test time of AdaBoost.HM+MLP is longer than AdaBoost.MH+DT.

## 5. Conclusions and Further Work

We propose a multi-class AdaBoost framework with hypothesis margin. It maximizes the output of the positive class meanwhile minimizes the maximum output of the negative classes. We also compare it with other two AdaBoost algorithms on loss functions and upper bounds. In experiments on 23 datasets, AdaBoost.HM with MLP as the weak learners is demonstrated to outperform the previous algorithms AdaBoost.M1 and AdaBoost.MH. In addition, AdaBoost.HM+MLP is scalable the same as AdaBoost.MH+DT. We also note that AdaBoost with the MLP may degenerate or improve little on some datasets. This phenomenon will be further investigated experimentally and theoretically.

## 6. Acknowledgements

This work was supported by the National Natural Science Foundation of China (NSFC) under grant no.60825301.

## References

[1] K. Crammer, R. Gilad-Bachrach, A. Navot, and N. Tishby. Margin analysis of the lvq algorithm. In *NIPS*, pages 462–469, 2002.

[2] G. Eibl and K. P. Pfeiffer. How to make adaboost.m1 work for weak base classifiers by changing only one line of the code. In *ECML*, pages 72–83, 2002.

[3] G. Eibl and K.-P. Pfeiffer. Multiclass boosting for weak classifiers. *The Journal of Machine Learning Research*, 6:189–210, 2005.

[4] Y. Freund and R. E. Schapire. A decision-theoretic generalization of on-line learning and an application to boosting. In *European Conference on Computational Learning Theory*, pages 23–37, 1995.

[5] Y. Freund and R. E. Schapire. Experiments with a new boosting algorithm. In *International Conference on Machine Learning*, pages 148–156, 1996.

[6] J. Friedman, T. Hastie, and R. Tibshirani. Additive logistic regression: a statistical view of boosting. *Stanford University Technical Report*, 1998.

[7] S. Holger and B. Yoshua. Boosting neural networks. *Neural Computing*, 12(8):1869–1887, 2000.

[8] D. Opitz and R. Maclin. Popular ensemble methods: An empirical study. *Journal of Artificial Intelligence Research*, 11:169–198, 1999.

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

[10] J. Zhu, S. Rosset, H. Zou, and T. Hastie. Multiclass adaboost. Technical report, Department of Statistics,University of Michigan, 2006.