数理解析研究所講究録
1489巻 2006年 57-63

より効率的なフィルタリング型ブースティング技法
An Efficient Smooth Boosting by Filtering

畑埜晃平[†]
Kohei Hatano

九州大学 システム情報科学研究院 情報理学部門
Department of Informatics, Kyushu University

Abstract

Boosting is a general method to construct a highly accurate classifier by combining "weakly" accurate ones. Smooth boosting algorithms are variants of boosting methods which handle only smooth distributions on the data. They are proved to be noise-tolerant and can be used in the "boosting by filtering" scheme, which is suitable for learning over huge data. However, current smooth boosting algorithms have rooms for improvements: A non-smooth boosting algorithm, InfoBoost can perform more efficiently than typical boosting algorithms by using an information-theoretic criterion for choosing hypotheses. In this paper, we propose a new smooth boosting algorithm with an information-theoretic criterion and we show that it inherits the advantages of two approaches, smooth boosting and InfoBoost.

1 Introduction

In recent years, huge data have become available due to the development of computers and the Internet. In knowledge discovery and machine learning tasks, size of such huge data can reach hundreds of gigabytes or more. So it is important to make knowledge discovery or machine learning algorithms scalable. Sampling is one of effective techniques to deal with large data. That is, instead of using whole the data, we can obtain a summary of the data by sampling randomly from it. There are many results on sampling techniques (see, e.g., [4]) and applications to data mining tasks such as decision tree learning [6], support vector machine [2], and boosting [4, 5].

Especially, boosting is simple and efficient learning method among machine learning algorithms. The basic idea of boosting is combining many slightly accurate hypotheses(which we call "weak" hypotheses) into a highly accurate one. Originally, boosting was invented under the boosting by filtering framework, where the booster can sample examples randomly from the whole data [16, 7]. The main advantage of the filtering framework is that the learner need to store less examples. In general, the learner have to store many example enough in order to evaluate its final hypothesis. On the other hand, in boosting by framework, the booster does not have to store all sampled examples, but have to keep examples only for learning weak hypotheses, which is much smaller than those for the final hypotheses. So the boosting by filtering framework seems to fit learning tasks over huge data. However, early boosting algorithms [16, 7] which work in the filtering framework were not practical because they were not adaptive, i.e., they need the prior knowledge on the accuracy of weak hypotheses.

Madaboost, a modification of AdaBoost [8], is the first adaptive boosting algorithm which works in the filtering framework [5]. Combining with adaptive sampling methods [4], Madaboost is shown to be more efficient than AdaBoost over huge data, while keeping the prediction accuracy. By its nature of updating scheme, MadaBoost is categorized as one of "smooth" boosting algorithms [18, 9], where the name, smooth boosting, comes from the fact that these boosting algorithms only deal with smooth distributions over data (In contrast, for example, AdaBoost might construct exponentially skew dis-

[2]Email: hatano@i.kyushu-u.ac.jp

tributions over data). Smoothness of distributions enables boosting algorithms to sample data efficiently. Also, smooth boosting algorithms have theoretical guarantees for noise tolerance in the various noisy learning settings, such as statistical query model [5], malicious noise model [18] and agnostic boosting [9].

However it seems that there is still room for improvements on smooth boosting. A non-smooth boosting algorithm, InfoBoost [1] (which is a special form of real AdaBoost [17]), performs more efficiently than other boosting algorithms in the boosting by subsampling framework, where only a bunch of data is given in advance. More precisely, given hypotheses with error $1/2 - \gamma/2$ $(0 < \gamma < 1)$, typical boosting algorithms take $O(1/\gamma^2)$ iterations to learn sufficiently accurate hypothesis. On the other hand, InfoBoost learns in from $O(1/\gamma)$ to $O(1/\gamma^2)$ iterations by taking advantage of the situation when weak hypotheses have low false positive error [10, 11]. So InfoBoost can be more efficient at most by $O(1/\gamma)$ times.

The main difference between InfoBoost and other boosting algorithms such as AdaBoost or MadaBoost is the criterion for choosing weak hypotheses. Typical boosting algorithms are designed to choose hypotheses whose errors are minimum with respect to given distributions. In contrast, InfoBoost uses an information-theoretic criterion to choose weak hypotheses. The criterion was previously proposed by Kearns and Mansour [12], and also applied to boosting algorithms using decision trees [12] and branching programs [14]. But, so far, no smooth algorithm is known to have such the nice property of InfoBoost.

In this paper, we modify one of smooth boosting algorithms, AdaFlat [9], as it is quite similar to MadaBoost and yet simple to analyze. Our modification is derived in a similar way that InfoBoost is given. As a result, we propose a new smooth boosting algorithm with yet another information-theoretic criterion. Our preliminary experiments show that our modification, which we call MadaFlat (Modification of AdaFlat), outperforms MadaBoost in the filtering framework.

2 Preliminaries

2.1 Learning Model

We adapt the PAC learning model [19]. Let $\mathcal{X}$ be an *instance space* and let $\mathcal{Y} = \{-1, +1\}$ be a set of labels. We assume an unknown *target function* $f : \mathcal{X} \to \mathcal{Y}$. Further we assume that f is contained in a known class $\mathcal{F}$ of functions from $\mathcal{X}$ to $\mathcal{Y}$. Let D be an unknown distribution over $\mathcal{X}$. The learner has an access to the *example oracle* $\mathrm{EX}(f, D)$. When given a call from the learner, $\mathrm{EX}(f, D)$ returns an *example* $(\boldsymbol{x}, f(\boldsymbol{x}))$ where each $\boldsymbol{x}$ is drawn randomly according to D. Let $\mathcal{H}$ be a hypothesis space, or a set of functions from $\mathcal{X}$ to $\mathcal{Y}$. We assume that $\mathcal{H} \supset \mathcal{F}$. For any distribution D over $\mathcal{X}$, *error* of hypothesis $h \in \mathcal{H}$ is defined as $\mathrm{err}_D(h) \stackrel{\mathrm{def}}{=} \Pr_D\{h(\boldsymbol{x}) \neq f(\boldsymbol{x})\}$. Let S be a *sample*, a set of examples $((\boldsymbol{x}_1, f(\boldsymbol{x}_1)), \ldots, (\boldsymbol{x}_m, f(\boldsymbol{x}_m)))$. For any sample S, *training error* of hypothesis $h \in \mathcal{H}$ is defined as $\widehat{\mathrm{err}}_S(h) \stackrel{\mathrm{def}}{=} |\{(\boldsymbol{x}_i, f(\boldsymbol{x}_i) \in S \mid h(\boldsymbol{x}_i) \neq f(\boldsymbol{x}_i)\}|/|S|$.

Now we define PAC learnability as follows.

Definition 1 (Strong learning). A learning algorithm A is a *strong learner* for $\mathcal{F}$ if and only if, for any $f \in \mathcal{F}$ and any distribution D, given ε, δ $(0 < \varepsilon, \delta < 1)$, a hypothesis space $\mathcal{H}$, and access to the example oracle $\mathrm{EX}(f, D)$ as inputs, A outputs a hypothesis $h \in \mathcal{H}$ such that $\mathrm{err}_D(h) = \Pr_D\{h(x) \neq f(x)\} \leq \varepsilon$ with probability at least $1 - \delta$.

On the other hand, an apparently weaker notion of learning was proposed [16].

Definition 2 (Weak learning). A learning algorithm A is a *weak leaner* for $\mathcal{F}$ if and only if, for any $f \in \mathcal{F}$, given a hypothesis space $\mathcal{H}$, and access to the example oracle $\mathrm{EX}(f, D)$ as inputs, A outputs a hypothesis $h \in \mathcal{H}$ such that $\mathrm{err}_D(h) \leq 1/2 - \gamma/2$ for a fixed γ $(0 < \gamma < 1)$.

2.2 Boosting Approach

Schapire proved that strong and weak PAC learnability are equivalent to each other for the first time [16]. Especially the technique to construct

a strong learner by using a weak learner is called "boosting". Basic idea of boosting is the following: First, the booster trains a weak learner with respect to different distributions $D_1, \ldots, D_T$ over the domain $\mathcal{X}$, and gets different "weak" hypotheses $h_1, \ldots, h_T$ such that $\mathrm{err}_{D_t}(h_t) \leq 1/2 - \gamma_t/2$ for each $t = 1, \ldots, T$. Then the booster combines weak hypotheses $h_1, \ldots, h_T$ into a final hypotheses h_{final} satisfying $\mathrm{err}_D(h_{final}) \leq \varepsilon$.

Subsampling versus Filtering We consider two frameworks of boosting, *boosting by subsampling* and *boosting by filtering*. In the subsampling framework, the booster is given a sample $S = ((\boldsymbol{x}_1, f(\boldsymbol{x}_1)), \ldots, (\boldsymbol{x}_m, f(\boldsymbol{x}_m)))$ in advance. The booster constructs the final hypothesis whose training $\widehat{\mathrm{err}}_S(h_{final}) \leq \varepsilon$ by training the weak learner over the given sample S. Then the generalization error is estimated by using arguments on VC-dimensions or margin (E.g., see [15]). For example, for typical boosting algorithms, $\mathrm{err}_D(h_{final}) \leq \widehat{\mathrm{err}}_S(h_{final}) + \tilde{O}(\sqrt{T \log |W|/m})$ [1] with high probability, where T is the size of the final hypotheses, i.e., the number of weak hypotheses combined in h_{final}.

In the filtering framework, on the other hand, the booster deal with the whole instance space $\mathcal{X}$ through $\mathrm{EX}(f, D)$. By using statistics obtained from calls of $\mathrm{EX}(f, D)$, the booster tries to minimizes $\mathrm{err}_D(h_{final})$ directly. There are two advantages of the boosting by filtering over the boosting by subsampling. First of all, the space complexity is reduced. Roughly speaking, the booster needs to store $\tilde{O}(T \log |W|)$ examples in the subsampling framework, whereas, the booster only need to store $\tilde{O}(\log |W|)$ examples in the filtering framework. Second, the booster does not have to determine the size of sample S a priori. There advantages are preferable for learning over huge data.

Smooth Boosting Smooth boosting algorithms only deal such distributions $D_1, \ldots, D_t$ that are "smooth" with respect to the original distribution D. We define the following measure of smoothness.

[1] In the $\tilde{O}(g(n))$ notation, we neglect $poly(\log(n))$ terms.

Definition 3. Let D and D' be any distributions over $\mathcal{X}$. We say that D' is λ-smooth with respect to D if $\sup_{\boldsymbol{x} \in \mathcal{X}} D'(\boldsymbol{x})/D(\boldsymbol{x}) \leq \lambda$.

The smoothness parameter λ has crucial roles in robustness of boosting algorithms [5, 18, 9]. Also, it affects the efficiency of sampling methods. For example, by rejection sampling, we use $1/\lambda$ calls of $EX(f, D)$ on average to simulate a call of $\mathrm{EX}(f, D')$ for a distribution D' that is λ-smooth w. r. t. D.

Our Assumption and Technical Goal In the rest of the paper, we assume the *weak hypothesis assumption on* W as follows: The learner is given a finite set W of hypotheses such that for any distribution D' over $\mathcal{X}$, there exists a hypothesis $h \in W$ satisfying $\mathrm{err}_{D'}(h) \leq 1/2 - \gamma/2$ $(0 < \gamma < 1)$.

Now our technical goal is to construct an efficient smooth boosting algorithm which works in both the subsampling and the filtering framework.

3 Boosting by Subsampling

In this section, we propose a modification of AdaFlat in the subsampling framework. Let

$$\ell(x) = \begin{cases} 1, & x \geq 0 \\ x - 1, & -1 < x < 0 \\ 0, & x \leq -1. \end{cases}$$

The description of our modification is given in Figure 3 Given a sample $S = ((\boldsymbol{x}_1, f(\boldsymbol{x}_1)), \ldots, (\boldsymbol{x}_m, f(\boldsymbol{x}_m)))$, and a combined hypothesis $H_t = \sum_{j=1}^{t} \alpha_j [h_j(\boldsymbol{x})] h_j(\boldsymbol{x})$ at iteration t, *pseudo gain* of hypothesis h is given as follows:

$$\begin{aligned} \Delta_t(h) &= \frac{m_{h,+}}{m} \mu_{h,t}[+1]^2 \gamma_{h,t}[+1]^2 \\ &\quad + \frac{m_{h,-}}{m} \mu_{h,t}[-1]^2 \gamma_{h,t}[-1]^2. \end{aligned}$$

where

$$\mu_{h,t}[\pm 1] = \frac{1}{m_{h,\pm}} \sum_{i : h(\boldsymbol{x}_i) = \pm 1} \ell(-f(\boldsymbol{x}_i) H_t(\boldsymbol{x}_i)),$$

$$\gamma_{h,t}[\pm 1] = \frac{\sum_{i : h(\boldsymbol{x}_i) = \pm 1} f(\boldsymbol{x}_i) h(\boldsymbol{x}_i) D_t(i)}{\sum_{i : h(\boldsymbol{x}_i) = \pm 1} D_t(i)},$$

$$D_t(\boldsymbol{x}_i) = \frac{\ell(-f(\boldsymbol{x}_i) H_t(\boldsymbol{x}_i))}{\sum_{i=1}^{m} \ell(-f(\boldsymbol{x}_i) H_t(\boldsymbol{x}_i))},$$

and $m_{h,\pm} = |\{i : h(\boldsymbol{x}_i) = \pm 1\}|$ (In the case when $m_{h,\pm}$ is zero, we assume that $\mu_{h,t}[\pm 1] = \gamma_{h,t}[\pm 1] = 0$). At each iteration t, our algorithm MadaFlat chooses hypothesis h that maximizes its

MadaFlat

Given: $S = ((\boldsymbol{x}_1, f(\boldsymbol{x}_1)), ..., (\boldsymbol{x}_m, f(\boldsymbol{x}_m)))$, and ε $(0 < \varepsilon < 1)$
begin
1. $D_1(i) \leftarrow 1/m$; $(i = 1, ..., m)$ $H_1(\boldsymbol{x}) \leftarrow 0$; $t \leftarrow 1$;
2. **while** $\widehat{\mathrm{err}}_S(h_{final}) > \varepsilon$ **do**
   a) $h_t \leftarrow \arg\max_{h \in W} \Delta_t(h)$;
   b) $\alpha_t[+1] \leftarrow \mu_t[+1]\gamma_t[+1]$;
      $\alpha_t[-1] \leftarrow \mu_t[-1]\gamma_t[-1]$;
   c) $H_{t+1}(\boldsymbol{x}) \leftarrow H_t(\boldsymbol{x}) + \alpha_t[h_t(\boldsymbol{x})]h_t(\boldsymbol{x})$;
   d) Define the next distribution D_{t+1} as
$$D_{t+1}(i) = \frac{\ell(-f(\boldsymbol{x}_i)H_{t+1}(\boldsymbol{x}_i))}{\sum_{i=1}^m \ell(-f(\boldsymbol{x}_i)H_{t+1}(\boldsymbol{x}_i))};$$
   e) $t \leftarrow t + 1$;
   end-while
3. Output the final hypothesis defined by
$$h_{final}(\boldsymbol{x}) = \mathrm{sign}\left(H_{t+1}(\boldsymbol{x})\right).$$
end.

Figure 1: MadaFlat

pseudo gain $\Delta_t(h)$. For simplicity, we denote $\Delta_t = \Delta_t(h_t)$, $\mu_t[\pm 1] = \mu_{h_t,t}[\pm 1]$ and $\gamma_t = \gamma_{h_t,t}$. Further, let us define $\mu_t = \sum_{i=1}^m \ell(-f(\boldsymbol{x}_i)H_t(\boldsymbol{x}_i))/m$. For each h_t, let $\gamma_t = \sum_{i=1}^m f(\boldsymbol{x}_i)h_t(\boldsymbol{x}_i)D_t(i)$. Note that $\mathrm{err}_{D_t}(h_t) = 1/2 - \gamma_t/2$.

First, we show that the smoothness of distributions D_t.

Proposition 1. During the execution of MadaFlat, each distribution D_t $(t \geq 1)$ is $1/\varepsilon$-smooth with respect to D_1, the uniform distribution over S.

Next, we prove the time complexity of MadaFlat.

Theorem 2. Assume the weak hypothesis assumption on W. Then, MadaFlat outputs a final hypothesis h_{final} satisfying $\widehat{\mathrm{err}}_S(h_{final}) \leq \varepsilon$ within $T = O\left(1/\varepsilon^2\gamma^2\right)$ iterations.

4 Boosting by Filtering

In this section, we propose MadaFlat$_{filt}$ in the filtering framework. Let

$$D_t(\boldsymbol{x}) = \frac{D(\boldsymbol{x})\ell(-f(\boldsymbol{x})H_t(\boldsymbol{x}))}{\sum_{\boldsymbol{x}\in\mathcal{X}} D(\boldsymbol{x})\ell(-f(\boldsymbol{x})H_t(\boldsymbol{x}))}.$$

We define $\Delta_t(h) = p_h\mu_{h,t}[+1]^2\gamma_{h,t}[+1]^2 + (1 - p_h)\mu_{h,t}[-1]^2\gamma_{h,t}[-1]^2$, where $p_h = \Pr_D\{h(\boldsymbol{x}) = +1\}$,

$$\mu_{h,t}[\pm 1] = \frac{\sum_{\boldsymbol{x}:h(\boldsymbol{x})=\pm 1} D(\boldsymbol{x})\ell(-f(\boldsymbol{x})H_t(\boldsymbol{x}))}{\sum_{\boldsymbol{x}:h(\boldsymbol{x})=\pm 1} D(\boldsymbol{x})},$$

MadaFlat$_{filt}(\varepsilon, \delta, \mathrm{EX}(f, D))$

begin
1.Let $H_1(x) = 0$; $t \leftarrow 1$; $\delta_1 \leftarrow \delta/4$;
2.**while** $\hat{\mu}_t \geq \frac{4\varepsilon}{5}$ **do**
   a) $(h_t, S_t) \leftarrow$ HSelect$(1/2, \delta_t)$;
   b) $(\hat{\mu}_t, \hat{\mu}_t[+1], \hat{\mu}_t[-1], \hat{\gamma}_t[+1], \hat{\gamma}_t[-1])$
      $\leftarrow$ empirical estimates over S_t;
   c) $\alpha_t[+1] \leftarrow \hat{\mu}_t[+1]\hat{\gamma}_t[+1]$; $\alpha_t[-1] \leftarrow \hat{\mu}_t[-1]\hat{\gamma}_t[-1]$;
   d) $H_{t+1}(\boldsymbol{x}) \leftarrow H_t(\boldsymbol{x}) + \alpha_t[h_t(\boldsymbol{x})]h_t(\boldsymbol{x})$;
   e) $t \leftarrow t + 1$; $\delta_t \leftarrow \delta/(2t(t+1))$;
   end-while
3. Output the final hypothesis defined by
$$h_{final}(\boldsymbol{x}) = \mathrm{sign}\left(H_{T+1}(\boldsymbol{x})\right);$$
end.

HSelect(ε, δ)
begin
  $m \leftarrow 0$; $S \leftarrow \emptyset$; $i \leftarrow 1$; $\Delta_g \leftarrow 1$; $\delta' \leftarrow \delta/(2|W|)$;
  repeat
    $(\boldsymbol{x}, f(\boldsymbol{x})) \leftarrow \mathrm{EX}(f, D)$;
    $S \leftarrow S \cup (\boldsymbol{x}, f(\boldsymbol{x}))$; $m \leftarrow m + 1$;
    if $m = \left\lceil \frac{c_1 \ln \frac{b_1}{\delta'}}{\Delta_g} \right\rceil$ **then**
      if $\exists h \in W$, $\hat{\Delta}(h, S) \geq \Delta_g$ **then** return h and S;
      else $\Delta_g \leftarrow \Delta_g/2$; $i \leftarrow i + 1$; $\delta \leftarrow \delta/(i(i+1)|W|)$;
    end-if
  end-repeat
end.

Figure 2: MadaFlat$_{filt}$

and

$$\gamma_{h,t}[\pm 1] = \frac{\sum_{\boldsymbol{x}:h(\boldsymbol{x})=\pm 1} -f(\boldsymbol{x})h(\boldsymbol{x})D(\boldsymbol{x})\ell(-f(\boldsymbol{x})H_t(\boldsymbol{x}))}{\sum_{\boldsymbol{x}:h(\boldsymbol{x})=\pm 1} D(\boldsymbol{x})\ell(-f(\boldsymbol{x})H_t(\boldsymbol{x}))},$$

respectively. Also let

$$\mu_t = \sum_{\boldsymbol{x}\in\mathcal{X}} D(\boldsymbol{x})\ell(-f(\boldsymbol{x})H_t(\boldsymbol{x}))$$

and $\gamma_{h,t} = p_h\gamma_{h,t}[+1] + (1 - p_h)\gamma_{h,t}[-1]$. As in the previous section, we use the similar notation: $\Delta_t = \Delta_t(h_t)$, $p_t = p_{h_t}$, and $\gamma_t[\pm 1] = \gamma_{h,t}[\pm 1]$. We denote $\hat{a}$ as the empirical estimate of the parameter a given a sample S_t. The description of MadaFlat$_{filt}$ is given in Figure 2.

First, we prove the following lemma.

Lemma 1. Let $\hat{\Delta}_t = \hat{p}_t\hat{\mu}_t[+1]^2\hat{\gamma}_t[+1]^2 + (1 - \hat{p}_t)\hat{\mu}_t[-1]^2\hat{\gamma}_t[-1]^2$ be the empirical estimate of Δ_t given S_t. Then it holds for any ε $(0 < \varepsilon < 1)$ that

$$\Pr_{D^m}\{\hat{\Delta}_t \geq (1 + \varepsilon)\Delta_t\} \leq b_1 e^{-\frac{\varepsilon^2\Delta_t m}{c_1}}, \qquad (1)$$

and

$$\Pr_{D^m}\{\widehat{\Delta}_t \leq (1-\varepsilon)\Delta_t\} \leq b_1 e^{-\frac{\varepsilon^2 \Delta_t m}{c_2}}, \qquad (2)$$

where $b_1 \leq 8$, $c_1 \leq 600$, and $c_2 \leq 64$.

Then we show the property of HSelect.

Lemma 2. Fix any step t in MadaFlat. Let $\Delta_* = \max_{h \in W} \Delta_t(h)$. Then, the following statements hold. (i) With probability at least $1-\delta$, HSelect(ε, δ) outputs a hypothesis $h \in W$ such that $\Delta_t(h) > (1-\varepsilon)\Delta_*$. (ii)The number of calls of $EX(f, D)$ is

$$O\left(\frac{\log\frac{1}{\delta} + \log|W| + \log\log\frac{1}{\Delta_*}}{\varepsilon^2 \Delta_*}\right).$$

Finally we obtain the following theorem.

Theorem 3. With probability at least $1-\delta$,

(i) MadaFlat$_{filt}$ outputs the final hypothesis h_{final} such that $\mathrm{err}_D(h_{final}) \leq \varepsilon$,

(ii) MadaFlat$_{filt}$ terminates in $T = O\left(1/\varepsilon^2\gamma^2\right)$ iterations, and

(iii) the number of calls of EX(f, D) is

$$O\left(\frac{\log\frac{1}{\delta} + \log\frac{1}{\varepsilon\gamma} + \log|W| + \log\log\frac{1}{\varepsilon\gamma}}{\varepsilon^2\gamma^4}\right).$$

5 Experimental Results

In this section, we show some experimental results on both artificial and real data sets. Our experiments consists of two parts.

In the first part, we compare MadaFlat, AdaBoost, InfoBoost, and MadaBoost. in the subsampling framework.

For real data, we use some datasets from UCI machine learning repository [3]. Also, we prepare artificial data in order to examine behavior of boosting algorithms in details. To do so we use r-of-k function as the target function. An r-of-k function f over boolean domain $\{-1,+1\}^N$ consists of k relevant variables and $f(\boldsymbol{x}) = +1$ if at least r of the k relevant variables takes $+1$, otherwise $f(\boldsymbol{x}) = -1$.

dataset	Ada.	InfoB.	Mada.	MadaF.
kr-vs-kp	**0.052**	0.056	**0.052**	0.061
hypothyroid	0.025	0.025	0.025	**0.024**
sick-euthoroid	0.060	**0.055**	0.060	0.057
spambase	0.23	0.23	0.23	0.23
10of70	0.19	0.062	0.19	**0.045**
20of70	0.074	0.060	0.074	**0.042**
30of70	0.073	0.067	0.073	**0.051**

Table 1: Test errors of boosting algorithms in the subsampling framework.

Note that 1-of-k function and $k/2$-of-k function correspond to k-disjunction and k-majority, respectively. In [11], it is shown that Info Boost can learn r-of-k functions in $O(rk)$ steps whereas AdaBoost needs $O(k^2)$ steps when boolean literals are used as weak hypotheses. For $r = 1, 3, 5$ and $k = 10$, we fix a r-of-k function over 100 boolean variables as the target function, and we generate 10,000 random examples labeled by each r-of-k function, where the random examples are drawn so that positive and negative examples are equally likely. The size of data we use varies from about 3,000 to 10,000.

For each dataset, we prepare decision stumps and the constant hypothesis $+1$ (i.e. the hypothesis that always answers $+1$) as weak hypotheses. In each dataset, each record have numeric attributes or binary attributes. For each numeric attribute, we construct a decision stump with a threshold, which predicts $+1$ or -1 depending on whether the value of the attribute is below the threshold or not. The threshold is chosen so that the training error of the decision stump is minimized. For each binary attribute, we prepare the decision stump which answers the value of the attribute.

We evaluate the boosting algorithms by cross validation. We split each data randomly 10 times, where each example is put into a training set with probability 0.7 and a test set with with probability 0.3. For each training set, we run the boosting algorithms in 100 steps and evaluate their final hypotheses on the test data. The results are summarized in Table 5.

As shown in Table 5, performance of MadaFlatappears to be comparable to those of others on real datasets. Also, MadaFlat is significantly better on artificial datasets, as well as InfoBoost.

In the second part, we compare MadaBoost and MadaFlatin the filtering framework. Basic settings of our experiments in the filtering framework are the same as those in the subsampling framework, except the following: First of all, in order to obtain large datasets, as is done in [4], we inflate the datasets by preparing 100 copies of each record in the data and changing their order randomly. Consequently, the size of the inflated data varies from 300,000 to 1,000,000. Second, instead of running each algorithm in 100 steps, we run them until they sample 10,0000 examples. More precisely, we run MadaFlatwith HSelect(ε, δ), where parameter $\varepsilon = 0.5$ and $\delta = 0.1$ are fixed. Also, we run MadaBoost with geometric AdaSelect whose parameters are $s = 2$, $\varepsilon = 0.5$ and $\delta = 0.1$.

Third, note that we use heuristics for Lemma 1. Although Lemma 1 gives a theoretical guarantee to approximate the pseudo gain accurately enough, it is too rough to use in practice. By using the central limit theorem, it is not hard to show that $\hat{\Delta}$ is asymptotically distributed from $N(\Delta, \sigma^2)$, where $\sigma \leq 5\Delta/m$. This analysis implies that it is safe to replace the condition on m in HSelect with

$$m = \left\lceil \frac{10\left(\ln \frac{1}{\sqrt{5}\delta'} - \frac{1}{2}\ln\ln \frac{1}{\sqrt{5}\delta'}\right)}{\Delta_g} \right\rceil .$$

In the following experiments, we use this improved heuristics.

Finally, in addition, we apply MadaBoost and MadaFlatfor text categorization tasks on a collection of Reuters news (Reuters-21578 [2]). We use the modified Apte("ModApte") split which contains about 10,000 news documents labeled with topics. We choose two major topics ("earnings" and "acquisitions") and for each of two topics, we let boosting algorithms classify whether a news document belongs to the topic or not. As weak hypotheses, we prepare about 30,000 decision stumps corresponding to words. This experiment is done in the same setting of previous ones, except that we do not inflate this dataset. The results are summarized in Table 5.

[2]http://www.daviddlewis.com/resources/testcollections /reuters21578.

dataset	MadaBoost	MadaFlat
kr-vs-kp	0.061	0.061
hypothyroid	**0.021**	0.026
sick-euthoroid	0.056	0.056
spambase	0.24	0.24
10of70	0.41	**0.31**
20of70	0.38	**0.30**
30of70	**0.29**	0.30

Table 2: Test errors of boosting algorithms in the filtering framework.

As indicated in Table 5 and Figure 3, MadaFlatoften outperforms MadaBoost.

6 Summary and Future Work

In this paper, we propose a modification of AdaFlat that uses an information-theoretic criterion for choosing hypotheses. Our preliminary experiments show that our modification appears to outperform MadaBoost in the filtering framework. As future work, advantages and noise torelance of MadaFlat are yet to be investigated theoretically. Also, we plan to conduct experiments over much larger data.

Acknowledgments

This work is supported in part by the 21st century COE program at Graduate School of Information Science and Electrical Engineering in Kyushu University.

References

[1] J. A. Aslam. Improving algorithms for boosting. In *Proc. 13th Annu. Conference on Comput. Learning Theory*, pages 200–207, 2000.

[2] Jose L. Balcazar, Yang Dai, and Osamu Watanabe. Provably fast training algorithms for support vector machines. In *Proceedings of*

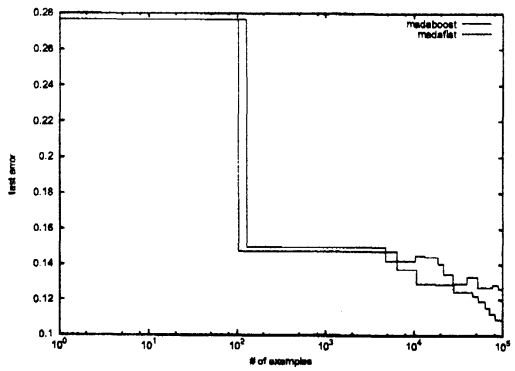

Figure 3: Test error of boosinting algorithms for Reuters-21578 data. The test errors are averaged over topics. The lower line corresponds to the test error of MadaFlat.

IEEE International Conference on Data Mining (ICDM'01), pages 43–50, 2001.

[3] C.L. Blake D.J. Newman, S. Hettich and C.J. Merz. UCI repository of machine learning databases, 1998.

[4] C. Domingo, R. Gavaldà, and O. Watanabe. Adaptive sampling methods for scaling up knowledge discovery algorithms. *Data Mining and Knowledge Discovery*, 6(2):131–152, 2002.

[5] C. Domingo and O. Watanabe. MadaBoost: A modification of AdaBoost. In *Proceedings of 13th Annual Conference on Computational Learning Theory*, pages 180–189, 2000.

[6] P. Domingos and G. Hulten. Mining high-speed data streams. In *Proceedings of the Sixth ACM International Conference on Knowledge Discovery and Data Mining*, pages 71–80, 2000.

[7] Y. Freund. Boosting a weak learning algorithm by majority. *Information and Computation*, 121(2):256–285, 1995.

[8] Y. Freund and R. E. Schapire:. A decision-theoretic generalization of on-line learning and an application to boosting. *Journal of Computer and System Sciences*, 55(1):119–139, 1997.

[9] D. Gavinsky. Optimally-smooth adaptive boosting and application to agnostic learning. *Journal of Machine Learning Research*, 2003.

[10] K. Hatano and M. K. Warmuth. Boosting versus covering. In *Advances in Neural Information Processing Systems 16*, 2003.

[11] K. Hatano and O. Watanabe. Learning r-of-k functions by boosting. In *Proceedings of the 15th International Conference on Algorithmic Learning Theory*, pages 114–126, 2004.

[12] M. Kearns and Y. Mansour. On the boosting ability of top-down decision tree learning algorithms. *Journal of Computer and System Sciences*, 58(1):109–128, 1999.

[13] Michael J. Kearns, Robert E. Schapire, and Linda Sellie. Toward efficient agnostic learning. In *COLT*, pages 341–352, 1992.

[14] Yishay Mansour and David A. McAllester. Boosting using branching programs. *Journal of Computer and System Sciences*, 64(1):103–112, 2002.

[15] R. Meir and G. Rätsch. An introduction to boosting and leveraging. In *Advanced lectures on machine learning*, pages 118–183. Springer-Verlag New York, Inc, 2003.

[16] Robert E. Schapire. The strength of weak learnability. *Machine Learning*, 5(2):197–227, 1990.

[17] Robert E. Schapire and Yoram Singer. Improved boosting algorithms using confidence-rated predictions. *Machine Learning*, 37(3):297–336, 1999.

[18] R. A. Servedio. Smooth boosting and learning with malicious noise. In *14th Annual Conference on Computational Learning Theory*, pages 473–489, 2001.

[19] L. G. Valiant. A theory of the learnable. *Communications of the ACM*, 27(11):1134–1142, 1984.