

Empirical Bernstein stopping

Volodymyr Mnih, Csaba Szepesvari, Jean-Yves Audibert

▶ **To cite this version:**

Volodymyr Mnih, Csaba Szepesvari, Jean-Yves Audibert. Empirical Bernstein stopping. ICML '08 Proceedings of the 25th international conference on Machine learning, Jul 2008, Helsinki, Finland. pp.672-679, 10.1145/1390156.1390241 . hal-00834983

HAL Id: hal-00834983

https://enpc.hal.science/hal-00834983

Submitted on 18 Jun 2013

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.

Empirical Bernstein Stopping

Volodymyr Mnih MNIH@CS.UALBERTA.CA
Csaba Szepesvári SZEPESVA@CS.UALBERTA.CA
Department of Computing Science, University of Alberta, Edmonton, AB T6G 2E8 Canada

Jean-Yves Audibert AUDIBERT@CERTIS.ENPC.FR
Certis - Ecole des Ponts, 6 avenue Blaise Pascal, Cité Descartes, 77455 Marne-la-Vallée France
Willow - ENS / INRIA, 45 rue d'Ulm, 75005 Paris, France

Abstract

Sampling is a popular way of scaling up machine learning algorithms to large datasets. The question often is how many samples are needed. Adaptive stopping algorithms monitor the performance in an online fashion and they can stop early, saving valuable resources. We consider problems where probabilistic guarantees are desired and demonstrate how recently-introduced empirical Bernstein bounds can be used to design stopping rules that are efficient. We provide upper bounds on the sample complexity of the new rules, as well as empirical results on model selection and boosting in the filtering setting.

1. Introduction

Being able to handle large datasets and streaming data is crucial to scaling up machine learning algorithms to many-real world settings. When making even a single pass through the data is prohibitive, sampling may offer a good solution. In order for the resulting algorithms to be theoretically sound, sampling techniques that come with probabilistic guarantees are desirable. For example, when estimating the error of a classifier on a large dataset one may want to sample until the estimated error is within some small number ϵ of the true error with probability at least $1 - \delta$. The key problem is one of *stopping* or determining the required number of samples. Taking too many samples will result in inefficient algorithms, while taking too few may not be enough to achieve the desired guarantees.

Appearing in *Proceedings of the 25th International Conference on Machine Learning*, Helsinki, Finland, 2008. Copyright 2008 by the author(s)/owner(s).

Finite sample bounds, such as Hoeffding's inequality (Hoeffding, 1963), are the key technique used by recent, non-parametric stopping algorithms with probabilistic guarantees. While these stopping algorithms have proved to be effective for scaling up machine learning algorithms (Bradley & Schapire, 2008), (Domingos & Hulten, 2001), they can be significantly improved by incorporating variance information in a principled manner. We show how to employ the recently introduced empirical Bernstein bounds (Audibert et al., 2007a) to improve stopping algorithms and provide sample complexity bounds and empirical results to demonstrate the effect of incorporating variance information.

Before proceeding, we identify two classes of stopping problems that will be examined. The first class concerns problems where some unknown quantities have to be measured either up to some prespecified level of accuracy or to support making a binary decision. Examples in this class include stopping with a fixed relative or absolute accuracy, with applications in hypothesis testing such as deciding on the sign of the mean, independence tests, and change detection. In problems belonging to the second group, the task is to pick the best option from a finite pool while measuring their performance using samples. Some notable examples include various versions of bandit problems, Hoeffding Races (Maron & Moore, 1993), and the general framework for scaling up learning algorithms proposed by Domingos (2001).

The paper is organized as follows. In Section 2 we examine Hoeffding's inequality and introduce the empirical Bernstein bound. In Section 3, we introduce a new stopping algorithm for stopping with a predefined relative accuracy and show that it is more efficient than previous algorithms. Section 4 demonstrates how a simple application of the empirical Bernstein bound can result in substantial improvements for problems

from the second class. Conclusions and future work directions are presented in Section 5.

2. Hoeffding Bounds vs. Empirical Bernstein Bounds

Let $X_1, \ldots, X_t$ real-valued *i.i.d.* random variables with range R and, mean μ, and let $\overline{X}_t = 1/t \sum_{i=1}^{t} X_i$. Hoeffding's inequality (Hoeffding, 1963) states that with probability at least $1 - \delta$

$$|\overline{X}_t - \mu| \leq R\sqrt{\frac{\log(2/\delta)}{2t}}.$$

Due to its generality, Hoeffding's inequality has been widely used in online learning scenarios. A drawback of the bound is that it scales linearly with the range R and does not scale with the variance of X_i. If a bound on the variance is known, Bernstein's inequality can be used instead, which can yield significant improvements when the variance bound is small relative to the range. Since useful a priori bounds on the variance are rarely available, this approach is not practical.

An approach that is more suitable to online scenarios is to apply Bernstein's inequality to the sum of $X_1, \ldots, X_t$, as well as the sum of the squares to obtain a single bound on the mean of $X_1, \ldots, X_t$. The resulting bound, which we will refer to as the *empirical Bernstein bound* (Audibert et al., 2007a) states that with probability at least $1 - \delta$

$$|\overline{X}_t - \mu| \leq \overline{\sigma}_t\sqrt{\frac{2\log(3/\delta)}{t}} + \frac{3R\log(3/\delta)}{t},$$

where $\overline{\sigma}_t$ is the empirical standard deviation of $X_1, \ldots, X_t$: $\overline{\sigma}_t^2 = \frac{1}{t}\sum_{i=1}^{t}(X_i - \overline{X}_t)^2$. The term involving the range R decreases at the rate of t^{-1} and quickly becomes negligible when the variance is large, while the square root term depends on $\overline{\sigma}_t$ instead of R. Hence, when $\overline{\sigma}_t \ll R$ the empirical Bernstein bound quickly becomes much tighter than Hoeffding's inequality.

3. Stopping Rules

Let $X_1, X_2, \ldots$ be i.i.d. random variables with mean μ and variance σ^2. We will refer to an algorithm as a stopping rule if at time t it observes X_t and based on past observations decides whether to stop or continue sampling. If a stopping rule $\mathcal{S}$ returns $\hat{\mu}$ that satisfies

$$\mathbb{P}\left[|\hat{\mu} - \mu| \leq \epsilon|\mu|\right] \geq 1 - \delta, \tag{1}$$

then $\mathcal{S}$ is a (ϵ, δ)-stopping rule and $\hat{\mu}$ is an (ϵ, δ)-approximation of μ. In this section, we develop an (ϵ, δ)-stopping rule for bounded X_i.

Algorithms proposed for this problem include the Nonmonotonic Adaptive Sampling (NAS) algorithm, shown as Algorithm 1, due to Domingo et al. (2000a). The general idea is to first use Hoeffding's inequality to construct a sequence $\{\alpha_t\}$ such that the event $\mathcal{E} = \left\{|\overline{X}_t - \mu| \leq \alpha_t,\, t \in \mathbb{N}^+\right\}$ occurs with probability at least $1 - \delta$, and then use this sequence to design a stopping criterion that stops only if $\left|\overline{X}_t - \mu\right| \leq \epsilon|\mu|$ given that $\mathcal{E}$ holds.

Algorithm 1 Algorithm NAS

```
t ← 0
repeat
  t ← t + 1
  Obtain X_t
  α ← sqrt((1/2t) log(t(t+1)/δ))
until |X̄_t| < α(1 + 1/ε)
return X̄_t
```

Domingo et al. (2000a) argue that if T is the number of samples after which NAS stops, and $|\mu| > 0$, then there exists a constant C such that with probability at least $1 - \delta$

$$T < C \cdot \frac{R^2}{\epsilon^2\mu^2}\Big(\log\frac{2}{\delta} + \log\frac{R}{\epsilon\mu}\Big). \tag{2}$$

The assumption that $|\mu| > 0$ is necessary for guaranteeing that the algorithm will indeed stop, and will be assumed for the rest of the section.

Dagum et al. (2000) introduced the $\mathcal{AA}$ algorithm for the case of bounded and nonnegative X_i. The $\mathcal{AA}$ algorithm is a three step procedure. In the first step, an $(\max(\sqrt{\epsilon}, 1/2), \delta/3)$-approximation of μ, $\tilde{\mu}$, is obtained. In the second step $\tilde{\mu}$ is used to determine the number of samples necessary to produce an estimate $\tilde{\sigma}^2$ of σ^2 such that $\max(\tilde{\sigma}^2, \epsilon\tilde{\mu})$ is a high probability upper bound on $\max(\sigma^2, \epsilon\mu)/2$. In the last step, $c\max(\tilde{\sigma}^2, \epsilon\tilde{\mu})\frac{\log(1/\delta)}{\epsilon^2\tilde{\mu}^2}$ samples are drawn and their average is returned as $\hat{\mu}$, where c is a universal constant.

Dagum et al. (2000) prove that $\hat{\mu}$ is indeed an (ϵ, δ)-approximation of μ and that that if T is the number of samples taken by $\mathcal{AA}$, then there exists a constant C such that with probability at least $1 - \delta$

$$T \leq C \cdot \max(\sigma^2, \epsilon\mu) \cdot \frac{1}{\epsilon^2\mu^2}\log\frac{2}{\delta}. \tag{3}$$

In addition, Dagum et al. prove that if T is the number of samples taken by any (ϵ, δ)-stopping rule, then there exists a constant C' such that with probability at least $1 - \delta$

$$T \geq C' \cdot \max(\sigma^2, \epsilon\mu) \cdot \frac{1}{\epsilon^2\mu^2}\log\frac{2}{\delta}.$$

Hence, for bounded X_i, the $\mathcal{AA}$ algorithm requires a number of samples that is at most a multiplicative constant larger than that required by any other (ϵ, δ)-stopping rule. In this sense the algorithm achieves "optimal" efficiency, up to a multiplicative constant.

While the $\mathcal{AA}$ algorithm is able to take advantage of variance, it requires the random variables to be nonnegative. Any trivial extension of the $\mathcal{AA}$ algorithm to the case of signed random variables seems unlikely since the rule heavily relies on the monotonicity of partial sums that is present in the nonnegative case. On the other hand, Equation (2) suggests that the NAS algorithm is not able to take advantage of variance.

As the first demonstration of how the empirical Bernstein bound can be used to design improved stopping algorithms, we propose a new algorithm, EBStop, which uses empirical Bernstein Bounds to achieve nearly the same scaling properties as the $\mathcal{AA}$ algorithm and, like the NAS algorithm, only requires the random variables to be bounded.

3.1. EBStop

Similarly to the NAS algorithm, EBStop relies on a sequence $\{c_t\}$ with the property that the event $\mathcal{E} = \{\left|\overline{X}_t - \mu\right| \leq c_t, t \in \mathbb{N}^+\}$ occurs with probability at least $1-\delta$. Let d_t be a positive sequence satisfying $\sum_{t=1}^{\infty} d_t \leq \delta$ and let

$$c_t = \overline{\sigma}_t \sqrt{\frac{2\log(3/d_t)}{t}} + \frac{3R\log(3/d_t)}{t}.$$

Since $\{d_t\}$ sums to at most δ and $(\overline{X}_t - c_t, \overline{X}_t + c_t)$ is a $1-d_t$ confidence interval for μ obtained from the empirical Bernstein bound, by a union bound argument, the event $\mathcal{E}$ indeed occurs with probability at least $1-\delta$. In our work, we use $d_t = c/t^p$ and $c = \delta(p-1)/p$.

The pseudocode for EBStop is shown as Algorithm 2, but the general idea is as follows. After drawing t samples, we set LB to $\max(0, \max_{1\leq s\leq t} |\overline{X}_s| - c_s)$ and UB to $\min_{1\leq s\leq t}(|\overline{X}_s| + c_s)$. EBStop terminates as soon as $(1+\epsilon)\text{LB} \geq (1-\epsilon)\text{UB}$ and returns $\hat{\mu} = \text{sgn}(\overline{X}_t)\cdot 1/2\cdot [(1+\epsilon)\text{LB} + (1-\epsilon)\ \text{UB}]$.

To see why $\hat{\mu}$ is an (ϵ, δ)-approximation, suppose the stopping condition has been satisfied and event $\mathcal{E}$ holds. Then

$$|\hat{\mu}| \leq 1/2 \cdot [(1+\epsilon)\text{LB} + (1+\epsilon)\text{LB}] \leq (1+\epsilon)|\mu|,$$

and similarly $(1-\epsilon)|\mu| \leq |\hat{\mu}|$. From the definition of LB, it also follows that $|\overline{X}_t| > c_t \geq |\overline{X}_t - \mu|$ which implies that $\text{sgn}(\hat{\mu}) = \text{sgn}(\mu)$. Since event $\mathcal{E}$ holds with probability at least $1-\delta$, $\hat{\mu}$ is indeed an (ϵ, δ)-approximation of μ.

Algorithm 2 Algorithm EBStop

```
LB ← 0
UB ← ∞
t ← 1
Obtain X_1
while (1 + ε)LB < (1 − ε)UB do
  t ← t + 1
  Obtain X_t
  LB ← max(LB, |X̄_t| − c_t)
  UB ← min(UB, |X̄_t| + c_t)
end while
return sgn(X̄_t) · 1/2· [(1 + ε)LB +(1 − ε) UB]
```

While we omit the proof due to space constraints[1], we note that if X is a random variable distributed with range R, and if T is the number of samples taken by EBStop on X, then there exists a constant C such that with probability at least $1-\delta$

$$T < C \cdot \max\left(\frac{\sigma^2}{\epsilon^2\mu^2}, \frac{R}{\epsilon|\mu|}\right)\left(\log\frac{1}{\delta} + \log\frac{R}{\epsilon|\mu|}\right). \quad (4)$$

This bound is very similar to the upper bound for the stopping time of the $\mathcal{AA}$ algorithm, with the only difference being the extra $\log\frac{R}{\epsilon|\mu|}$ term. This term comes from constructing a confidence interval at each t and is not an artifact of our proof techniques. However, this extra term can be reduced to $\log\log\frac{1}{\epsilon|\mu|}$ by applying a geometric grid as we will see in the next section. Since EBStop does not require the variables to be non-negative, we can say that EBStop combines the best properties of NAS and $\mathcal{AA}$ algorithms for signed random variables.

3.2. Improving EBStop

While EBStop has the desired scaling properties, we make two simple improvements in order to make it more efficient in practice.

The first improvement is based on the idea that if the algorithm is not close to stopping, there is no point in checking the stopping condition at every point. We incorporate this idea into EBStop by adopting a *geometric* sampling schedule, also used by Domingo and Watanabe (2000a). Instead of testing the stopping criterion after each sample, we perform the kth test after $\lceil\beta^k\rceil$ samples have been taken for some $\beta > 1$. Under this sampling strategy, when EBStop constructs a $1-d$ confidence interval after t samples, d is of the order $1/(\log_\beta t)^p$, which is much larger than $1/t^p$. Since

[1] A version of the paper containing the proofs will be made available as a technical report.

this results in tighter confidence intervals, LB and UB will approach each other faster and the stopping condition will be satisfied after fewer samples.

While geometric sampling can often reduce the number of required samples, it can also lead to taking roughly β times too many samples, because testing is only done at the ends of intervals. Nevertheless, the following result due to Audibert et al. (2007b) can be used to test the stopping condition after each sample without sacrificing the advantages of geometric sampling. Let $t_1 \leq t_2$ for $t_1, t_2 \in \mathbb{N}$ and let $\alpha \geq t_2/t_1$. Then with probability at least $1-d$, for all $t \in \{t_1, \ldots, t_2\}$

$$|\overline{X}_t - \mu| \leq \overline{\sigma}_t \sqrt{2\alpha \log(3/d)/t} + 3\alpha \log(3/d)/t. \quad (5)$$

We use Equation (5) with $t_1 = \lfloor \beta^k \rfloor + 1$, $t_2 = \lfloor \beta^{k+1} \rfloor$, and $d = c/k^p$ to construct c_t for each $t \in \{t_1, \ldots, t_2\}$. This allows us to test the stopping condition after each sample, and use d that is on the order of $1/(\log_\beta t)^{p\alpha}$ after t samples. A variant of EBStop that incorporates these two improvements is shown as Algorithm 3.

Algorithm 3 Algorithm EBGStop

```
LB ← 0
UB ← ∞
t ← 1
k ← 0
Obtain X_1
while (1 + ε)LB < (1 − ε)UB do
  t ← t + 1
  Obtain X_t
  if t > floor(β^k) then
    k ← k + 1
    α ← floor(β^k)/floor(β^{k−1})
    x ← −α log d_k/3
  end if
  c_t ← σ̄_t sqrt(2x/t) + 3Rx/t
  LB ← max(LB, |X̄_t| − c_t)
  UB ← min(UB, |X̄_t| + c_t)
end while
return sgn(X̄_t) · 1/2· [(1 + ε)LB +(1 − ε) UB]
```

One can show that as the result of adding geometric sampling to EBStop reduces the $\log \frac{1}{\epsilon|\mu|}$ term in inequality (4) to $\log\log \frac{1}{\epsilon|\mu|}$. It should be noted that from the arguments of Dagum et al. (2000), no stopping rule can achieve a better bound than (3) for the case of bounded non-negative random variables. Hence, EBGStop is very close to being "optimal" in this sense. Where it would loose (for non-negative random variables) to $\mathcal{AA}$ is when ϵ, μ and δ are such that $\log(R/(\epsilon\mu))$ becomes significantly larger than $1/\delta$. We do not expect to see this happening in practice for not too large values of δ. For example, for $\delta = 0.05$, $R = 1$, the condition is $\epsilon\mu < e^{-20}$.

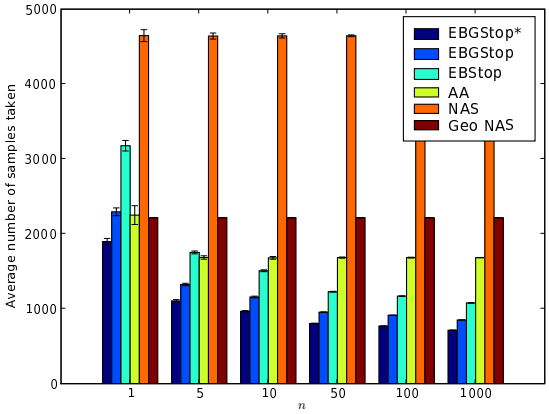


Figure 1. Comparison of stopping rules on averages of uniform random variables with varying variances. Error bars are at one standard deviation.

3.3. Results: Synthetic Data

In this section we evaluate the stopping rules $\mathcal{AA}$, NAS, geometric NAS, EBStop, and EBGStop on the problem of estimating means of various random variables. To make the comparison fair, the geometric version of the NAS algorithm and EBGStop both grew intervals by a factor of 1.5, as this value worked best in previous experiments (Domingo & Watanabe, 2000b). We also used $d_t = (t(t+1))^{-1}$ in EBStop and EBGStop since this is the sequence implicitly used in NAS for constructing confidence intervals at time t. Since this put EBGStop at a slight disadvantage, we also include results for EBGStop, denoted by EBGStop*, with our default choice of $d_t = c/(\log_\beta t)^p$, $p = 1.1$, and $\beta = 1.1$. In all the experiments we used $\epsilon = \delta = 0.1$. We use only non-negative valued random variables as they allow comparison to $\mathcal{AA}$. Finally, we only compare the number of samples taken because none of the algorithms produced any estimates with relative error greater than ϵ in any of our experiments.

The first set of experiments was meant to test how well the various stopping rules are able to exploit the variance when it is small. Let the average of n uniform $[a, b]$ random variables be denoted by $U(a, b, n)$. Note that the expected value and variance of $U(a, b, n)$ are $(a + b)/2$ and $(b - a)^2/(12n)$, respectively. For this comparison we fixed a to 0, b to 1, and varied n to control the variance for a fixed mean. Figure 1 shows the results of running each stopping rule 100 times on $U(0, 1, n)$ random variables for $n = 1, 5, 10, 50, 100, 1000$. Not surprisingly, NAS and geometric NAS fail to make use of the variance

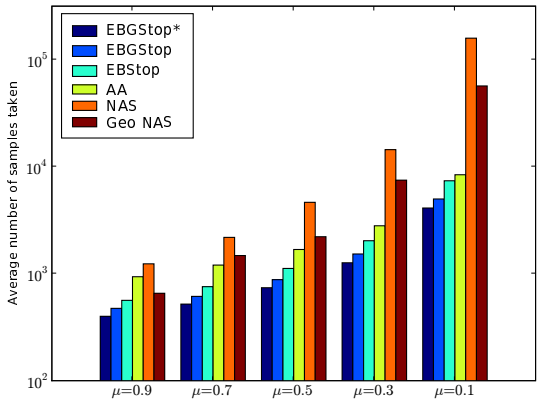


Figure 2. Comparison of stopping rules on averages of uniform random variables with varying means. The number of samples is shown in log scale.

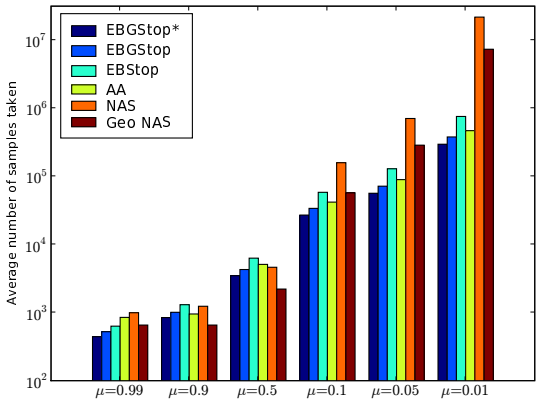


Figure 3. Comparison of stopping rules on Bernoulli random variables. The number of samples is shown in log scale.

and take roughly the same numbers of samples for all values of n. Variants of EBStop improve when the variance decreases, with EBGStop* performing especially well, beating all the other algorithms for all the scenarios tested. $\mathcal{AA}$ initially improves with the decreasing variance, but the effect is not as large as with EBGStop* because of the multi-phase structure of $\mathcal{AA}$.

In the second set of experiments we fix n at 10 and $b - a$ at 0.2, keeping the variance fixed, and vary the mean. The variance is small enough that EBStop, its variants, and $\mathcal{AA}$ should take a number of samples in the order of $R/(\epsilon\mu)$. The results are presented in Figure 2 and suggest that both variants of NAS require $1/\mu$ times more samples than the variance-adaptive methods. Note that Figure 2 shows the number of samples taken by each method in log scale.

It may be surprising that in both experiments the $\mathcal{AA}$ algorithm did not outperform EBStop and EBGStop even though $\mathcal{AA}$ offers better guarantees on sample complexity. We believe that EBStop is able make better use of the data because it uses all samples in its stopping criterion, while $\mathcal{AA}$ wastes some samples on intermediate quantities. However, this difference should be reflected in the hidden constants. As discussed earlier, for really small values of μ and ϵ the $\mathcal{AA}$ algorithm should stop earlier than EBStop.

Finally, we include a comparison of the stopping rules on Bernoulli random variables. Since Bernoulli random variables have maximal variance of all bounded random variables, the advantage of variance estimation should be diminished. However, inequality (4) suggests that in the case of Bernoulli random variables EBStop requires $O(1/(\epsilon^2\mu))$ samples since $\sigma^2 = \mu(1-\mu)$. Similarly, inequality (2) suggests that NAS requires $O(1/(\epsilon^2\mu^2))$ samples.

Figure 3 shows the results of running each stopping rule on Bernoulli random variables with means 0.99, 0.9, 0.5, 0.1, 0.05, and 0.01, averaged over 100 runs. As in the previous set of experiments, the variance-adaptive methods seem to require $1/\mu$ times fewer samples to stop. It should also be noted that the geometric version of the NAS algorithm does outperform EBStop for some intermediate values of μ, where the variance is the largest. However the performance difference is not large, and so we think the price paid for the unboundedly better performance of EBStop for small or large values of μ is not large.

3.4. Results: FilterBoost

Boosting by filtering (Bradley & Schapire, 2008) is a framework for scaling up boosting algorithms to large or streaming datasets. Instead of working with the entire training set, all steps, such as finding a weak learner that has classification accuracy of at least 0.5, are done through sampling that employs stopping algorithms. Bradley and Schapire showed that such an approach can lead to a drastic speedup over a batch boosting algorithm.

We evaluated the suitability of EBGStop and both variants of the NAS algorithm for the boosting by filtering setting by plugging them into the FilterBoost algorithm (Bradley & Schapire, 2008). The $\mathcal{AA}$ algorithm was not included because it cannot deal with signed random variables.

Following Bradley and Schapire, the Adult and Covertype datasets from the UCI machine learning repository (Asuncion & Newman, 2007) were used. The covertype dataset was converted into a binary classi-

fication problem by taking "Lodgepole Pine" as one class and merging the other classes. In setting up boosting we followed the procedure of Domingo and Watanabe (2000b) who also considered the use of stopping rules in the same context. Accordingly, we used decisions stumps as weak learners and we discretized all continuous attributes by binning their values into five equal bins. The results for the Adult dataset were averaged over 10 runs on the training set, while 10-fold cross-validation was used for the Covertype dataset.

As shown in Figure 4, EBGStop required fewer samples and offered lower variance in stopping times than either variant of the NAS algorithm on both datasets. At the same time, the resulting classification accuracies were within 0.2% of each other on the Adult dataset and within 0.04% of each other on the Covertype dataset.

4. Racing Algorithms

In this section we demonstrate how a general stopping algorithm that makes use of finite sample deviation bounds can be improved with the use of empirical Bernstein bounds. We consider the Hoeffding races algorithm (Maron & Moore, 1993) since it is representative of the class of general stopping algorithms.

Racing algorithms aim to reduce the computational burden of performing tasks such model selection using a hold-out set by discarding poor models quickly (Maron & Moore, 1993; Ortiz & Kaelbling, 2000). The context of racing algorithms is the one of multi-armed bandit problems. Formally, consider M options. When option m is chosen the t^{th} time, it gives a random value $X_{m,t}$ from an unknown distribution ν_m. The samples $\{X_{m,t}\}_{t\geq 1}$ are independent of each other. Let $\mu_m = \int x\nu_m(dx)$ be the mean reward obtained of option m. The goal is to find the options with the highest mean reward.

Let $\delta > 0$ be the confidence level parameter and N be the maximal amount of time allowed for deciding which option leads to the best expected reward. A racing algorithm either terminates when it runs out of time (i.e. at the end of the N-th round) or when it can say that with probability at least $1-\delta$, it has found the best option, i.e. an option m^* with $\mu_{m^*} = \max_{m\in\{1,\dots,M\}} \mu_m$.

The Hoeffding race is an algorithm based on discarding options which are likely to have smaller mean than the optimal one until only one option remains. Precisely, for each time step and each distribution, $\delta/(MN)$ confidence intervals are constructed for the mean. Options with upper confidence smaller than the lower confidence bound of another option are discarded. The algorithm samples one by one all the options that have not been discarded yet.

We assume that the rewards have a bounded range R. If $\overline{X}_{m,t}$ denotes the sample mean for option m after seeing t samples of this option then according to Hoeffding's inequality, a $\delta/(MN)$ confidence interval for the mean of option m is

$$\left[\overline{X}_{m,t} - R\sqrt{\frac{\log(2MN/\delta)}{2t}}, \overline{X}_{m,t} + R\sqrt{\frac{\log(2MN/\delta)}{2t}}\right]$$

The Hoeffding race has been introduced and studied in (Maron & Moore, 1993; Maron & Moore, 1997) in a slightly different viewpoint since there the target was to find an option with mean at most ϵ below the optimal mean $\max_{m\in\{1,\dots,M\}} \mu_m$, where ϵ is a given positive parameter. The same problem was also studied by (Even-Dar et al., 2002, Theorem 3) in the infinite horizon setting.

By substituting Hoeffding's inequality with the empirical Bernstein bound we obtain a new algorithm, which we will refer to as the empirical Bernstein race.

4.1. Analysis of Racing Algorithms

For the analysis we are interested in the expected number of samples taken by the Hoeffding race and the empirical Bernstein race. Due to space limitations, we omit the proofs of the following theorems.

Let $\Delta_m = \mu_{m^*} - \mu_m$, where option m^* still denotes an optimal option: $\mu_{m^*} = \max_{m\in\{1,\dots,M\}} \mu_m$. Let $\lceil u \rceil$ denote the smallest integer larger or equal to u, and let $\lfloor u \rfloor$ denote the largest integer smaller or equal to u.

Theorem 1 (Hoeffding Race)**.** *Let* $n_H(m) = \lceil \frac{8R^2\log(2MN/\delta)}{\Delta_m^2} \rceil$. *Without loss of generality, assume that* $n_H(1) \leq n_H(2) \leq \ldots \leq n_H(m)$. *The number of samples, T, taken by the Hoeffding race is bounded by*

$$2\sum_{\mu_m<\mu_{m*}} n_H(m).$$

The probability that no optimal option is returned is bounded by δ. *If the algorithm runs out of time, then with probability at least* $1-\delta$, *(i) the number of discarded options is at least d, where d is the largest integer such that* $2\sum_{m=1}^{d} n_H(m) \leq N$, *and (ii) the non-discarded options satisfy*

$$\mu_m \geq \mu_{m^*} - 4R\sqrt{\frac{\log(2MN/\delta)}{2\lfloor N/M \rfloor}}.$$

We recall that the principle of the empirical Bernstein race algorithm is the same as the Hoeffding's one. We

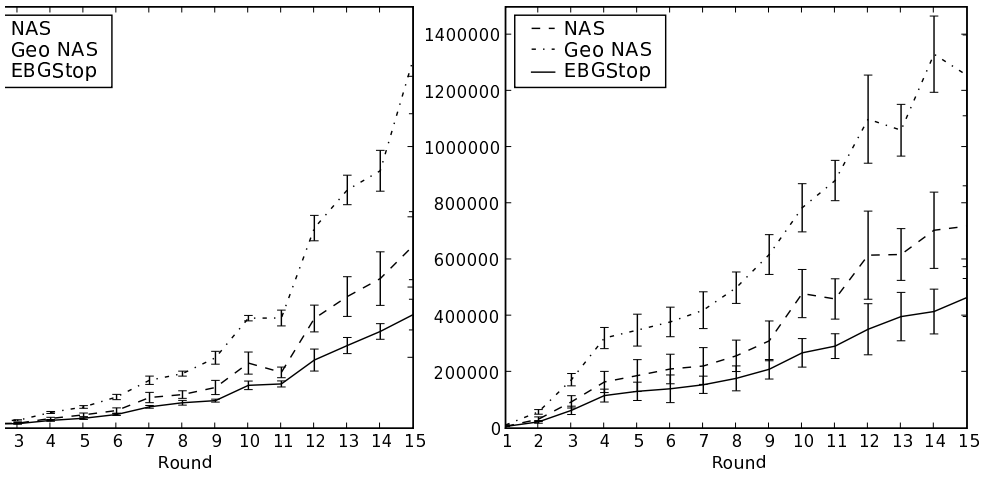


Figure 4. Comparison of the number of samples required by different stopping rules in FilterBoost. Parameters ϵ, δ were set to 0.1 for both methods while τ was set to 0.25. Error bars are at 1 standard deviation. a) Results on the Adult dataset b) Results on the covertype dataset.

sample one by one all the distributions that has not been discarded yet. The algorithm discards an option as soon as the upper bound on its mean reward is smaller than at least one of the lower bound on the mean of any other option.

Theorem 2 (Empirical Bernstein Race)**.** *Let σ_m denote the standard deviation of ν_m. Introduce $\Sigma_m = \sigma_{m^*} + \sigma_m$ and*

$$n(m) = \left\lceil \frac{8\Sigma_m^2 + 18R\Delta_m}{\Delta_m^2} \log(4MN/\delta) \right\rceil.$$

Without loss of generality, assume that $n(1) \leq n(2) \leq \ldots \leq n(m)$. The number of samples taken by the empirical Bernstein race is bounded by

$$2 \sum_{\mu_m < \mu_{m^*}} n(m).$$

The probability that no optimal option is returned is bounded by δ. If the algorithm runs out of time, then with probability at least $1-\delta$, (i) the number of discarded options is at least d, where d is the largest integer such that $2\sum_{m=1}^{d} n_H(m) \leq N$, and (ii) the non-discarded options satisfy

$$\mu_m \geq \mu_{m^*} - \Sigma_m \sqrt{\frac{8\log(4MN/\delta)}{\lfloor N/M \rfloor}} - \frac{9R\log(4MN/\delta)}{\lfloor N/M \rfloor}.$$

As can be seen from the bounds, the result of incorporating the variance estimates is similar to what was observed in Section 3: The dependence of the number of required samples on R^2 is reduced to a dependence on R and the variance. Similar results can be expected when applying the empirical Bernstein bound to other situations.

4.2. Results

Following the procedure of Maron and Moore (1997), we evaluated the Hoeffding and empirical Bernstein races on the task of selecting the best k for k-nearest neighbor regression and classification through leave-one-out cross-validation.[2] Three datasets of different types were used for the comparison. The SARCOS data presents a regression problem which involves predicting the torques at 7 joints of a robot arm based on the positions, velocities and accelerations at those joints. We only considered predicting the torque at the first joint. The Covertype2 dataset consists of 50,000 points sampled from the Covertype dataset from Section 3.4 and is a binary classification task. The Local dataset presents a regression problem that was created by sampling 10,000 points from a noisy piecewise-linear function defined on the unit interval and having a range of 1.

The value of the range parameter R was set to 1 for the Covertype2 and Local datasets. For the SARCOS dataset, R was set to the range of the target values in the dataset. This differs from the approach of setting R separately for each option to several times the standard deviation in the samples observed, suggested by Maron and Moore (1997). We do not follow this approach because it invalidates the use of Hoeffding's inequality.

[2]Since leave-one-out cross-validation creates dependencies between the samples, the analysis does not apply to this case. However, our experiments gave similar results when we used a separate hold-out set. We decided to present results for leave-one-out cross-validation to facilitate comparison with the original papers.

Table 1. Percentage of work saved / number of options left after termination.

Data set	Hoeffding	EB
SARCOS	0.0% / 11	44.9% / 4
Covertype2	14.9% / 8	29.3% / 5
Local	6.0% / 9	33.1% / 6

All methods were given the options $k = 2^0, 2^1, 2^2, 2^3, \ldots, 2^{10}$ to begin with. The results are presented in Table 1. The table shows the percentage of work saved by each method (1− number of samples taken by method / MN), as well as the number of options remaining after termination.

The empirical Bernstein racing algorithm, which is denoted by EB, significantly outperforms the Hoeffding racing algorithm on all three datasets. The advantage of incorporating variance estimates is the smallest on the Covertype2 classification dataset. This is expected because the samples come from Bernoulli distributions which have the largest possible variance for a bounded random variable. The advantage of variance estimation is the largest on the SARCOS dataset, where R is much larger than the variance. While one may argue that the Hoeffding racing algorithm would do much better if R was set to a smaller value based on the standard deviation, the empirical Bernstein algorithm would also benefit. However, tweaking R this way is merely an unprincipled way of incorporating variance estimates into a racing algorithm.

5. Conclusions and Future Work

We showed how variance information can be exploited in stopping problems in a principled manner. Most notably, we presented a near-optimal stopping rule for relative error estimation on bounded random variables, significantly extending the results of Domingo and Watanabe, and Dagum et al.. We also provided empirical and theoretical results on the effect that can be expected from incorporating variance estimates into existing stopping algorithms.

One interesting question that should be addressed is if the bound achieved by the $\mathcal{AA}$ algorithm in the non-negative case, which is known to be optimal, can be achieved without the non-negativity condition.

Acknowledgements

This work was supported in part by Agence Nationale de la Recherche project "Modèles Graphiques et Applications", the Alberta Ingenuity Fund, iCore, the Computer and Automation Research Institute of the Hungarian Academy of Sciences, and NSERC.

References

Asuncion, A., & Newman, D. (2007). UCI machine learning repository.

Audibert, J. Y., Munos, R., & Szepesvári, C. (2007a). Tuning bandit algorithms in stochastic environments. *ALT* (pp. 150–165).

Audibert, J.-Y., Munos, R., & Szepesvári, C. (2007b). *Variance estimates and exploration function in multi-armed bandit* (Technical Report 07-31). Certis - Ecole des Ponts. `http://certis.enpc.fr/~audibert/RR0731.pdf`.

Bradley, J. K., & Schapire, R. (2008). Filterboost: Regression and classification on large datasets. *NIPS-20* (pp. 185–192).

Dagum, P., Karp, R., Luby, M., & Ross, S. (2000). An optimal algorithm for Monte Carlo estimation. *SIAM Journal on Computing*, *29*, 1484–1496.

Domingo, C., & Watanabe, O. (2000a). MadaBoost: A modification of AdaBoost. *COLT'00* (pp. 180–189).

Domingo, C., & Watanabe, O. (2000b). Scaling up a boosting-based learner via adaptive sampling. *Pacific-Asia Conference on Knowledge Discovery and Data Mining* (pp. 317–328).

Domingos, P., & Hulten, G. (2001). A general method for scaling up machine learning algorithms and its application to clustering. *ICML* (pp. 106–113).

Even-Dar, E., Mannor, S., & Mansour, Y. (2002). PAC bounds for multi-armed bandit and Markov decision processes. *COLT'02* (pp. 255–270).

Hoeffding, W. (1963). Probability inequalities for sums of bounded random variables. *Journal of the American Statistical Association*, *58*, 13–30.

Maron, O., & Moore, A. (1993). Hoeffding races: Accelerating model selection search for classification and function approximation. *NIPS 6* (pp. 59–66).

Maron, O., & Moore, A. W. (1997). The racing algorithm: Model selection for lazy learners. *Artificial Intelligence Review*, *11*, 193–225.

Ortiz, L. E., & Kaelbling, L. P. (2000). Sampling methods for action selection in influence diagrams. *AAAI/IAAI* (pp. 378–385).