Proceedings of Machine Learning Research vol 237:1–19, 2024 35th International Conference on Algorithmic Learning Theory

The Impossibility of Parallelizing Boosting

Amin Karbasi AMIN.KARBASI@YALE.EDU
Yale University
Kasper Green Larsen LARSEN@CS.AU.DK
Aarhus University

Editors: Claire Vernade and Daniel Hsu

Abstract

The aim of boosting is to convert a sequence of weak learners into a strong learner. At their heart, these methods are fully sequential. In this paper, we investigate the possibility of parallelizing boosting. Our main contribution is a strong negative result, implying that significant parallelization of boosting requires an exponential blow-up in the total computing resources needed for training.

Keywords: Boosting, Generalization, Weak-to-Strong Learning, Parallelization

1. Introduction

Boosting is one of the most successful ideas in machine learning, allowing one to ”boost” the performance of a base learning algorithm with rather poor accuracy into a highly accurate classifier, with recent applications in adversarial training (Abernethy et al., 2021), reinforcement learning (Brukhim et al., 2021), and federated learning (Hamer et al., 2020; Shen et al., 2022), among many others. The classic boosting algorithm, known as AdaBoost (Freund and Schapire, 1997), achieves this by iteratively training classifers on the training data set. After each iteration, the data set is reweighed and a new classifier is trained using a weighted loss function. The weights intuitively guide the attention of the base learning algorithm towards training samples that the previous classifiers struggle with. After a sufficiently large number of iterations, the produced classifiers are combined by taking a weighted majority vote among their predictions.

Both the classic AdaBoost algorithm, as well as more modern gradient boosters (Friedman, 2001; Ke et al., 2017; Chen and Guestrin, 2016), all have this highly sequential behaviour, where the algorithm runs in multiple iterations that adjust the learning problem based on previously trained classifiers/regressors. Indeed, the best performance of gradient boosters on benchmark data sets is often obtained after hundreds, or even thousands of iterations (Natekin and Knoll, 2013). This may be appropriate when the base learning algorithm has a small training time. However, it prevents the use of boosting in combination with e.g. medium-sized neural networks as the base learning algorithm, or if one wishes to use all available training data in a large data set. Here the sequential nature of boosting algorithms is particularly critical, as it is not possible to simply distribute the training task to many machines. This shortcoming of boosting algorithms was also highlighted in the survey by Natekin and Knoll (2013) when discussing drawbacks of gradient boosting.

In light of the above concerns, it would have a huge practical impact if a highly parallel boosting algorithm could have been developed. Unfortunately, our main result shows that parallelizing boosting cannot be done without an *exponential* increase in the total work needed for training!

Weak to Strong Learning. To formalize the above claim that boosting cannot be parallelized, we need to introduce the theoretical framework in which we prove our impossibility result.

Boosting was introduced to address a theoretical question by Kearns (1988); Kearns and Valiant (1994), asking whether a so-called *weak learner* can always be converted to a *strong learner*. In the following, we define these notions formally. First, let $\mathcal{X}$ be an input domain and $c : \mathcal{X} \to \{-1, 1\}$ an unknown concept that we want to learn.

A γ-weak learner for a concept $c : \mathcal{X} \to \{-1, 1\}$, is an algorithm that given some constant number of samples m_0 labeled by c from *any* unknown distribution $\mathcal{D}$ over $\mathcal{X}$, with constant probability returns a hypothesis $h : \mathcal{X} \to \{-1, 1\}$ such that $\mathcal{L}_{\mathcal{D}}(h) := \Pr_{x\sim\mathcal{D}}[h(x) \neq c(x)] \leq 1/2 - \gamma$. It thus has an accuracy that is γ better than guessing. We say that the weak learner has a γ *advantage*. We remark that whether or not the number of samples m_0 is allowed to depend on γ is irrelevant for our results and they apply in all circumstances.

A strong learner on the other hand, is a learning algorithm such that for any $0 < \varepsilon, \delta < 1$, there is some number of samples $m(\varepsilon, \delta)$, such that when given $m(\varepsilon, \delta)$ samples labeled by c from any unknown distribution $\mathcal{D}$ over $\mathcal{X}$, with probability at least $1 - \delta$, it returns a hypothesis $h : \mathcal{X} \to \{-1, 1\}$ such that $\mathcal{L}_{\mathcal{D}}(h) \leq \varepsilon$. A strong learner can thus obtain arbitrarily high accuracy when given enough training data. We refer to $m(\varepsilon, \delta)$ as the sample complexity of the strong learner.

Kearns and Valiant thus asked whether the rather poor accuracy of a weak learner can always be exploited to obtain a classifier with arbitrarily high accuracy, i.e. a strong learner. This question was answered affirmatively by Schapire (1990), and AdaBoost is one example of such a weak-to-strong learning algorithm. Concretely, the tightest known bounds on the sample complexity of AdaBoost states that the hypothesis h it produces on a training set $S \sim \mathcal{D}^m$ of m samples, when using a γ-weak learner $\mathcal{W}$ as the base learning algorithm, satisfies

$$\mathcal{L}_{\mathcal{D}}(h) = O\left(\frac{d \ln(m) \ln(m/d) + \ln(1/\delta)}{\gamma^2 m}\right) \tag{1}$$

with probability at least $1 - \delta$ over S, see e.g. Shalev-Shwartz and Ben-David (2014). Here d denotes the VC-dimension of the hypothesis set $\mathcal{H}$ that the weak learner $\mathcal{W}$ outputs from. It is clear from this formula that the accuracy can be made arbitrarily high when given enough training data. In particular, for any ε, δ, if we ignore log-factors, then the sample complexity $m(\varepsilon, \delta)$ grows as $O((d + \ln(1/\delta))/\gamma^2\varepsilon)$. Furthermore, this sample complexity has recently been proved optimal for any weak-to-strong learning algorithm, see Larsen and Ritzert (2022).

In weak-to-strong learning algorithms $\mathcal{A}$, such as AdaBoost, a γ-weak learner $\mathcal{W}$ is typically used by feeding it a distribution $\mathcal{D}$ over the training data S (e.g. the weighing over the data set in each iteration of AdaBoost). It is then assumed that $\mathcal{W}$ returns a hypothesis h with error at most $1/2 - \gamma$ under $\mathcal{D}$. Note that this seems quite different from the above definition of a γ-weak learner, where the weak learner receives at least m_0 samples from an unknown distribution $\mathcal{D}$. However, the definition of a γ-weak learner actually implies that $\mathcal{W}$ can compute a hypothesis h with error at most $1/2 - \gamma$ for a query distribution $\mathcal{D}$ specified by $\mathcal{A}$. This is because $\mathcal{W}$ is given access to the training data S and the distribution $\mathcal{D}$ and thus it can just repeatedly sample m_0 samples from $\mathcal{D}$, compute a hypothesis from the samples, and compute its exact error probability under $\mathcal{D}$. If the error probability exceeds $1/2-\gamma$, $\mathcal{W}$ can just repeat with a fresh sample. Since $\mathcal{W}$ is a weak learner, this terminates in an expected constant number of tries. Here one critically exploits that a weak learner has advantage γ under *any* distribution when given at least m_0 samples. Henceforth, we thus think of a weak learner $\mathcal{W}$ as a procedure that we can query with a training set S, the labels $c(S)$ and a distribution $\mathcal{D}$ over S. It then returns a hypothesis h with $\mathcal{L}_{\mathcal{D}} \leq 1/2-\gamma$. A weak-to-strong learning algorithm $\mathcal{A}$ is then given query access to a γ-weak learner as well as the training set S and the labels $c(S)$. After querying $\mathcal{W}$ sufficiently many times to obtain hypotheses $h_1,\dots,h_k$, it must output a hypothesis $h : \mathcal{X} \to \{-1,1\}$. Note that we make no assumption that the output h is a voting classifier.

Parallel Weak to Strong Learning. We are now ready to define what we formally mean by parallelizing boosting. We say that a weak-to-strong learning algorithm $\mathcal{A}$ has *parallel complexity* (p,t) if it only invokes its weak learner $\mathcal{W}$ in p rounds. In each round, it may query the weak learner with up to t different distributions $\mathcal{D}_1,\dots,\mathcal{D}_t$. The weak learner then returns a hypothesis h_i for each $\mathcal{D}_i$ such that $\mathcal{L}_{\mathcal{D}_i}(h_i) \leq 1/2-\gamma$. The queries made in any round may only depend on $S, c(S)$ as well as the hypotheses obtained from $\mathcal{W}$ in previous rounds. We let $\mathcal{A}_{S,c(S),\mathcal{W}}$ denote the hypothesis $\mathcal{A}$ reports on input $S, c(S), \mathcal{W}$ when the p rounds are over.

In this terminology, it is known that AdaBoost needs $\Theta(\gamma^{-2}\ln m)$ iterations to obtain the accuracy claimed in (1). It thus has parallel complexity $(\Theta(\gamma^{-2}\ln m), 1)$. With parallel complexity defined, we are finally ready to present our main result, ruling out parallel boosting

Theorem 1 *There is a universal constant $a>0$ such that for any weak-to-strong learner $\mathcal{A}$, any m, any $0<\gamma<a$ and any VC-dimension d, there exists a domain $\mathcal{X}$, a concept $c : \mathcal{X} \to \{-1,1\}$ and a γ-weak learner $\mathcal{W}$ for c, such that $\mathcal{W}$ uses a hypothesis set of VC-dimension d and either: $p \geq \exp(\Omega(d))$ or $t \geq \min\{\exp(\Omega(d/\gamma^2)), \exp(\exp(\Omega(d)))\}$ or $\mathcal{L}_{\mathcal{D}}(\mathcal{A}_{S,c(S),\mathcal{W}}) \geq \exp(-O(p\max\{\gamma, \ln(tp)\gamma^2/d\}))$ in expectation over S and any random choices of $\mathcal{A}$. Here $\mathcal{X}, c$ and $\mathcal{W}$ depends on m, γ and d.*

Let us discuss the implications of the theorem in detail. First, let us consider the simplest case where one wants only a constant number of parallel rounds $p = O(1)$. Then the lower bound states that either $t \geq \exp(\Omega(d/\gamma^2)), t \geq \exp(\exp(\Omega(d)))$ or the error probability is at least $\exp(-O(\max\{\gamma, \ln(t)\gamma^2/d\}))$. To make this error probability comparable to AdaBoost (1) requires $t = \exp(\Omega(d\gamma^{-2}\ln m))$. Thus there is no way around either an exponential dependency on $d\gamma^{-2}$ or a double-exponential dependency on d.

If one is willing to use a super-constant number of rounds p, then there are a couple of possibilities for obtaining an error probability comparable to (1). First, one could have either $p = \exp(\Omega(d)), t = \exp(\Omega(d\gamma^{-2}))$ or $t = \exp(\exp(\Omega(d)))$. These bounds are all exponential in either the VC-dimension or γ^{-2} or both. Finally, there is the possibility of making the error $e^{-O(p\gamma)}$ small. To make it comparable to (1) requires $p = \Omega(\gamma^{-1}\ln m)$. This is only a γ factor less than AdaBoost. Thus there is unfortunately not much hope for parallelizing boosting, and certainly not to a near-constant number of rounds. In all circumstances, if the number of rounds is significantly less than $\gamma^{-1}\ln m$, then it requires an exponential number of queries t per round.

Finally, let us remark that it is sometimes stated that the unknown concept c belongs to some concept class $\mathbb{C}$. Here we implicitly assume that it belongs to the class of all concepts that may be γ-weak learned using the hypothesis set of the weak learner. Alon et al. (2021) proved that this concept class has VC-dimension at most $O_d(\gamma^{-2+2/(d+1)})$ when the hypothesis set of the γ-weak learner has VC-dimension d. Here $O_d(\cdot)$ hides factors depending only on d.

A Parallel Boosting Algorithm. To demonstrate the tightness of our lower bound, we also present a single-round boosting algorithm

Theorem 2 *There is a weak-to-strong learner $\mathcal{A}$, such that for any concept c, any γ-weak learner $\mathcal{W}$ for c using a hypothesis set of VC-dimension d, and any distribution $\mathcal{D}$, when given m samples $S \sim \mathcal{D}^m$, $\mathcal{A}$ has parallel complexity $(1, \exp(O(d\gamma^{-2}\ln m)))$ and with probability $1-\delta$ over S, it outputs a hypothesis h with*

$$\mathcal{L}_{\mathcal{D}}(h) = O\left(\frac{d\ln(m)\ln(m/d) + \ln(1/\delta)}{\gamma^2 m}\right).$$

The generalization error of our single-round algorithm thus matches that of AdaBoost (1), although making exponentially many queries to the weak learner. As shown by our lower bound, this is inevitable.

Let us also remark that using techniques in the two works by Larsen and Ritzert (2022); Larsen (2022), we can also remove the two logarithmic factors $\ln(m)\ln(m/d)$ from the upper bound. As the logarithmic factors are not essential to our contribution, we merely comment here that the logarithmic factors can be removed by creating a logarithmic number of bootstrap sub-samples of the training data, running our algorithm in parallel on all sub-samples, and outputting a majority vote among the resulting classifiers. The resulting algorithm is then an optimal weak-to-strong learner by the previously mentioned sample complexity lower bound of Larsen and Ritzert (2022).

Finally, let us comment that there are previous boosting algorithms, based on branching programs, that invoke a weak learner in parallel, see Mansour and McAllester (2002); Kalai and Servedio (2005); Long and Servedio (2005), however none of these works use $o(\gamma^{-2}\ln m)$ rounds of boosting.

Previous Lower Bounds for Parallelizing Boosting. Let us conclude by discussing related work by Long and Servedio (2013). In their work, they also study the parallel complexity of boosting, however under somewhat different assumptions. Concretely, they prove a lower bound showing that any weak-to-strong learner must have parallel complexity (p, t) satisfying $p = \Omega(\gamma^{-2}\ln m)$, regardless of the number of calls per round t. This strengthens a previous result by Freund (1995) and is quantitatively a stronger lower bound than ours. However, they also model the problem in a way that makes their result weaker than ours. First, they make no assumption on the complexity/VC-dimension of the hypothesis set used by the weak-learner. On close inspection of their construction, their input domain $\mathcal{X}$ is the full cube $\{-1,1\}^k$ with $k = \Theta(\gamma^{-2}\ln m)$ and they have one hypothesis h_i for each coordinate i, making the prediction $h_i(x) = x_i$. The VC-dimension of this hypothesis set is $\lfloor \lg_2 k \rfloor = \Theta(\ln(1/\gamma) + \ln\ln m)$, i.e. growing with m.

Secondly, and just as crucially, their lower bound assumes that the query distributions fed to the weak learner are obtained by ”filtering”. Concretely, this means that the weight/probability mass put on each sample point $x \in S$ is determined solely from the vector of predictions made by previously obtained hypotheses on x as well as the label $c(x)$ (see their Definition 2). In our lower bound, we make no assumption on how query distributions are chosen other than being computable from the hypotheses seen so far and the training data. This is a crucial difference, and in fact, our 1-round boosting algorithm explicitly queries for distributions that are *not* defined solely from labels and predictions while also using the bounded VC-dimension to define these distributions, thereby circumventing their lower bound.

2. Impossibility of Parallelization

In this section, we prove our main result, Theorem 1, stating that boosting can not be parallelized significantly without a major reduction in accuracy. We start by presenting the basic setup for our proof, then the high level ideas leading to the lower bound, followed by the formal details.

Basic Setup. Let $\mathcal{A}$ be an arbitrary, possibly randomized, weak-to-strong learning algorithm with parallel complexity (p, t). For any d, m and γ, we aim to design a γ-weak learner $\mathcal{W}$ for a concept $c : \mathcal{X} \to \{-1, 1\}$, using a hypothesis set $\mathcal{H}$ of VC-dimension d, such that $\mathcal{A}$ run with $\mathcal{W}$ as its weak learner must have a generalization error at least that stated in Theorem 1. We use the simple input domain $\mathcal{X} = \{x_1, \dots, x_{2m}\}$ and the hard data distribution $\mathcal{D}$ is the uniform distribution over $\mathcal{X}$.

Observe that if $p \geq \exp(\Omega(d)), t \geq \exp(\Omega(d/\gamma^2))$ or $t \geq \exp(\exp(\Omega(d)))$, we have nothing left to prove, so assume not. Let $\nu = e^{-O(p \max\{\gamma, \ln(tp)\gamma^2/d\})}$ for short. Our goal is to show that there is a concept c and weak learner $\mathcal{W}$ such that

$$\mathbb{E}_{S,\mathcal{A}}[\mathcal{L}_{\mathcal{D}}(\mathcal{A}_{S,c(S),\mathcal{W}})] \geq \nu. \tag{2}$$

Here we use $\mathbb{E}_{S,\mathcal{A}}$ to denote that the expectation is over both the random choice of S and randomness of $\mathcal{A}$.

Handling Randomness. To prove (2), we consider a *random* choice of concept c and weak learner $\mathcal{W}$, but a *deterministic* $\mathcal{A}$. Assume that we can design a distribution over c and $\mathcal{W}$ (independent of $\mathcal{A}$), such that any deterministic $\mathcal{A}$ satisfies

$$\mathbb{E}_{c,\mathcal{W}}[\mathbb{E}_S[\mathcal{L}_{\mathcal{D}}(\mathcal{A}_{S,c(S),\mathcal{W}})]] \geq \nu. \tag{3}$$

We claim (3) implies (2). To see this, let $\mathcal{A}$ be an arbitrary randomized weak-to-strong learning algorithm. Then by fixing the random choices of $\mathcal{A}, c$ and $\mathcal{W}$ (Yao's principle/linearity of expectation), there is a deterministic $\mathcal{A}^\star, c^\star$ and $\mathcal{W}^*$ with

$$\begin{aligned} \nu &\leq \mathbb{E}_{c,\mathcal{W}}[\mathbb{E}_S[\mathcal{L}_{\mathcal{D}}(\mathcal{A}^\star_{S,c(S),\mathcal{W}})]] \\ &\leq \mathbb{E}_{c,\mathcal{W}}[\mathbb{E}_{S,\mathcal{A}}[\mathcal{L}_{\mathcal{D}}(\mathcal{A}_{S,c(S),\mathcal{W}})]] \\ &\leq \mathbb{E}_{S,\mathcal{A}}[\mathcal{L}_{\mathcal{D}}(\mathcal{A}_{S,c^\star(S),\mathcal{W}^\star})]. \end{aligned}$$

Thus we henceforth focus on proving (3).

Hard Concept Distribution. We next describe our hard distribution over a concept c and corresponding γ-weak learner $\mathcal{W}$. The distribution over c is simply the uniform distribution, i.e. every $x \in \mathcal{X}$ is mapped independently to either -1 or 1 with equal probability. This is a natural hard distribution, as under this distribution, the training data $S, c(S)$ only provides information about labels of points in the training data.

Now observe that our data distribution is uniform over $\mathcal{X} = \{x_1, \dots, x_{2m}\}$ and $S \sim \mathcal{D}^m$ consists of m samples. Thus there are at least m samples that are not part of the training data. Without querying the weak learner $\mathcal{W}$, an algorithm $\mathcal{A}$ has no knowledge of these labels and can at best try to guess them with success probability $1/2$, leading to a constant error probability as a fresh sample $x \sim \mathcal{D}$ falls outside the training data with probability at least $1/2$.

Hard Weak Learner Distribution. In light of the above, we aim to design a γ-weak learner $\mathcal{W}$ such that its replies reveal as little as possible information about the labels outside the training data.

For this, we conceptually think of its hypothesis set $\mathcal{H}$ as consisting of p groups $\mathcal{H}_1, \dots, \mathcal{H}_p$. For each of these groups, starting with $\mathcal{H}_1$, we have a random subset X_i of $\mathcal{X}$. We let X_1 be a uniform random $2m/\beta$ sized subset of $\mathcal{X}$ and let X_i be a uniform random $2m\beta^{-i}$ sized subset of X_{i-1} for a parameter $\beta > 1$ to be determined. We then form $\mathcal{H}_i$ by drawing $2^{d/2}$ random hypotheses as follows: For each hypothesis $h \in \mathcal{H}_i$, we let $h(x) = c(x)$ for $x \notin X_i$ and we let $h(x)$ be uniform random and independently chosen for $x \in X_i$. We then add another $2^{d/2}$ random hypotheses to $\mathcal{H}_i$. These are simply chosen such that they return a uniform random label on every element $x \in \mathcal{X}$. Finally, we also add c to the hypothesis set $\mathcal{H}$. Since we assume $p \leq \exp(ad)$ for a sufficiently small constant a, we have that the hypothesis set has at most 2^d hypotheses and thus has VC-dimension at most d.

The weak learner $\mathcal{W}$ does the following upon being queried for a distribution $\mathcal{D}'$ over $\mathcal{X}$: It searches through the hypothesis sets $\mathcal{H}_i$, starting with $i = 1$, and returns the first hypothesis h it sees with $\mathcal{L}_{\mathcal{D}'}(h) \leq 1/2 - \gamma$. If no such h exists in any $\mathcal{H}_i$, it simply returns c.

Intuition. Let us discuss the main ideas in the design of the above (random) weak learner $\mathcal{W}$. We think of the hypothesis sets $\mathcal{H}_i$ and corresponding subsets X_i as being responsible for one parallel round each, where $\mathcal{H}_1, X_1$ is responsible for the first round.

If we consider the very first round, then observe that X_1 is a random β^{-1}-fraction of $\mathcal{X}$. Since this subset is unknown to a deterministic $\mathcal{A}$, if it queries $\mathcal{W}$ with a distribution $\mathcal{D}'$ that is somewhat uniform over $\mathcal{X}$, then about a $1 - \beta^{-1}$ fraction of the mass is on points $x \in \mathcal{X}$ with $x \notin X_1$. Each of the first $2^{d/2}$ hypotheses h in $\mathcal{H}_1$ has $h(x) = c(x)$ on such points, and thus immediately have an expected advantage of $1 - \beta^{-1}$. If $1 - \beta^{-1} \approx \gamma$, then there is an overwhelming probability that $\mathcal{W}$ can answer all queries made in the first round using hypotheses from $\mathcal{H}_1$.

Now assume that after i rounds, it holds that $\mathcal{W}$ has only returned hypotheses from $\mathcal{H}_1, \dots, \mathcal{H}_i$. Then for round $i + 1$, even if $\mathcal{A}$ learns all hypotheses in $\mathcal{H}_1, \dots, \mathcal{H}_i$, the subset X_{i+1} is again an unknown β^{-1}-fraction of X_i. It follows that the weak learner can again answer any queries made by $\mathcal{A}$ using only $\mathcal{H}_{i+1}$. Continuing this argument implies that after all p rounds have been completed, the weak learner $\mathcal{W}$ never returned the hypothesis c. Now even if $\mathcal{A}$ sees all of $\mathcal{H}_1, \dots, \mathcal{H}_p$, it still has no knowledge of the labels assigned by c

to points in X_p. Thus its error probability must be proportional to $|X_p \setminus S|/(2m) = \Omega(\beta^{-p})$. For $1 - \beta^{-1} \approx \gamma$, this is $\exp(-O(p\gamma))$ as required in (3).

Finally, let us remark that if $\mathcal{A}$ queries $\mathcal{W}$ with a distribution $\mathcal{D}'$ that is concentrated on a few entries, then the above argument breaks down, i.e. it suddenly becomes rather likely that less than a $1-\beta^{-1}$ fraction of the mass is on points $x \in \mathcal{X}$ with $x \notin X_i$. In the extreme case, $\mathcal{A}$ could simply query $\mathcal{W}$ with every singleton distribution (all mass on one point). This is the reason for including the extra $2^{d/2}$ hypotheses in $\mathcal{H}_i$ that are simply uniform random. These may be used whenever the support of $\mathcal{D}'$ is concentrated on $O(d\gamma^{-2})$ entries.

Formal Proof. Let $\mathcal{A}$ be an arbitrary deterministic weak-to-strong learning algorithm. Let

$$\eta = \mathbb{E}_{c,\mathcal{W}}[\mathbb{E}_S[\mathcal{L}_{\mathcal{D}}(\mathcal{A}_{S,c(S),\mathcal{W}})]].$$

We must prove $\eta \geq \nu$ to establish (3) and thus Theorem 1.

Now let E denote the event that throughout its execution on a random $S \sim \mathcal{D}^m$, $\mathcal{A}$ is never returned the hypothesis c by the weak learner. First we show that the expected error probability η is related to $\Pr[E]$.

Lemma 3 *It must be the case that*

$$(\Pr[E] - 1/2)2m\beta^{-p} \quad \leq \quad 2 + \lg m + 2m\eta \lg(e\beta^{-p}/\eta).$$

Intuitively, if we can show that $\Pr[E]$ is significantly larger than $1/2$, then the term $2m\eta\lg(e\beta^{-p}/\eta)$ must be proportional to $m\beta^{-p}$, thereby providing a lower bound on η. That $\Pr[E]$ is large is precisely the contents of the next lemma.

Lemma 4 *There are constants $a, a' > 0$, such that if $0 < \gamma < a$ and*

$$\beta = 1 + \max\{32\gamma, a' \ln(tp)\gamma^2/d\},$$

then $\Pr[E] \geq 49/50$.

Let us combine the two lemmas to establish (3). Let β be as in Lemma 4. From Lemma 3 and Lemma 4, we see that

$$(24/50)2m\beta^{-p} \leq 2 + \lg m + 2m\eta\lg(e\beta^{-p}/\eta).$$

Ignoring the $2 + \lg m$ term, this implies

$$\eta = \Omega(\beta^{-p}/\lg(e\beta^{-p}/\eta)),$$

which is equivalent to

$$\begin{aligned}
\eta &= \Omega(\beta^{-p}) \\
&= \Omega\left(\left(1 + \max\{32\gamma, a'\ln(tp)\gamma^2/d\}\right)^{-p}\right) \\
&\geq \exp(-O(p\max\{\gamma, \ln(tp)\gamma^2/d\})) \\
&= \nu.
\end{aligned}$$

This establishes (3) and thus Theorem 1.

Let us also remark that for the $2 + \lg m$ term to be relevant, we must have $m\beta^{-p} = O(\lg m)$, which is anyways when the lower bound $\nu = \exp(-O(p \max\{\gamma, \ln(tp)\gamma^2/d\}))$ drops to $1/m^{O(1)}$ and is dominated by the general $\Omega(d/(m\gamma^2))$ lower bound for weak-to-strong learning in previous work by Larsen and Ritzert (2022).

What remains is thus to prove Lemma 3 and Lemma 4. We start by proving Lemma 3.

Relating $\Pr[E]$ and η. The intuition in the proof of Lemma 3 is that whenever E occurs, the algorithm $\mathcal{A}$ has no knowledge of c inside $X_p \setminus S$, hence it can only guess these labels, resulting in an error probability of $\Omega(|X_p|/m) = \Omega(\beta^{-p})$.

To formalize this, we use an information theoretic argument. First, we show that even when revealing $S, c(S), \mathcal{H}_1, \dots, \mathcal{H}_p, X_1, \dots, X_p$, the random concept c still has a lot of randomness. This randomness is measured in terms of Shannon entropy $H(\cdot)$. Concretely, we show that

$$H(c \mid S, c(S), \mathcal{H}_1, \dots, \mathcal{H}_p, X_1, \dots, X_p) \geq m\beta^{-p}. \tag{4}$$

Next, we conversely show that if η is small and E is likely, then the entropy has to be small.

Lemma 5 *The conditional entropy of c is no more than*

$$H(c \mid S, c(S), \mathcal{H}_1, \dots, \mathcal{H}_p, X_1, \dots, X_p) \quad \leq \quad 2 + \lg m + (1 - \Pr[E])2m\beta^{-p} + 2m\eta \lg(e\beta^{-p}/\eta).$$

Before proving (4) and Lemma 5, we show that they together imply Lemma 3. Combining the two, we see that

$$m\beta^{-p} \leq 2 + \lg m + (1 - \Pr[E])2m\beta^{-p} + 2m\eta \lg(e\beta^{-p}/\eta).$$

Rearranging terms immediately implies Lemma 3.

We thus prove (4) and Lemma 5. For (4), we have

$$H(c \mid S, c(S), \mathcal{H}_1, \dots, \mathcal{H}_p, X_1, \dots, X_p) \quad \geq \quad \mathbb{E}[|\mathcal{H}_p \cap (\mathcal{X} \setminus S)|].$$

To see this, note that the conditional entropy of c, is the expectation over drawing $S, c(S)$ and $, \mathcal{H}_1, \dots, \mathcal{H}_p, X_1, \dots, X_p$, of the entropy of c conditioned on the outcome. But conditioned on the outcomes of these random variables, c is still uniform random inside $X_p \cap (\mathcal{X} \setminus S)$ and thus its conditional entropy is at least $|X_p \cap (\mathcal{X} \setminus S)|$.

We have $\mathbb{E}[|X_p \cap (\mathcal{X} \setminus S)|] \geq m\beta^{-p}$ from which (4) follows.

Next, we prove Lemma 5.

Proof (of Lemma 5) We first observe that

$$\mathbb{E}_{c,\mathcal{W},S}[\mathcal{L}_{\mathcal{D}}(\mathcal{A}_{S,c(S),\mathcal{W}}) \mid E] \leq \eta / \Pr[E].$$

Now to prove that the conditional entropy of c is small, we give a so-called encoding argument. By Shannon's source coding theorem, the conditional entropy of c is no more than the expected length of a prefix free encoding of c. Here the encoding is allowed to make use of the random variables $S, c(S), \mathcal{H}_1, \dots, \mathcal{H}_p, X_1, \dots, X_p$. We thus describe such a prefix free encoding and analyse its expected length.

Our encoding procedure is as follows when given $c, S, c(S), \mathcal{H}_1, \dots, \mathcal{H}_p, X_1, \dots, X_p$.

1. First, we check whether the event E occurs on the outcome of the random variables $c, S, c(S), \mathcal{H}_1, \dots, \mathcal{H}_p, X_1, \dots, X_p$. This can be checked since $\mathcal{W}$ can be simulated from $\mathcal{H}_1, \dots, \mathcal{H}_p$ as it always returns the first hypothesis that has γ advantage on a query distribution $\mathcal{D}$. Furthermore, $\mathcal{A}$ is deterministic, hence we can also simulate $\mathcal{A}$ from $S, c(S)$ and the replies by $\mathcal{W}$. Therefore, we can let the first bit of the encoding be 1 if E occurs and 0 otherwise.

2. If E did not occur, we finish the encoding by appending $|X_p|$ bits specifying $c(X_p)$. This takes $2m\beta^{-p}$ bits.

3. If E occurs, we simulate $\mathcal{A}$ with $\mathcal{W}, c(S), S$ to produce the hypothesis $h = \mathcal{A}_{S,c(S),\mathcal{W}}$. We then evaluate h on X_p and let Z denote the set of $x \in X_p$ for which $h(x) \neq c(x)$. Let $\Delta = |Z|$. Our encoding now specifies Δ by appending $\lg(2m) = 1 + \lg m$ bits. Next, we specify Z as a Δ-sized subset of X_p, costing $\lg \binom{|X_p|}{\Delta} = \lg \binom{2m\beta^{-p}}{\Delta} \leq \Delta \lg(2em\beta^{-p}/\Delta)$ bits.

We next describe how to decode c from the above encoding as well as $S, c(S)$ and $\mathcal{H}_1, \dots, \mathcal{H}_p, X_1, \dots, X_p$.

1. We start by checking the first bit of the encoding. If this is a 0-bit, we read the remaining $2m\beta^{-p}$ bits to recover $c(X_p)$. Finally, all remaining $c(x)$ can be reconstructed from the first hypothesis in $\mathcal{H}_p$.

2. If the first bit is 1, we read the second part of the encoding to reconstruct the set Z. We then simulate $\mathcal{A}$ and $\mathcal{W}$ to obtain $h = \mathcal{A}_{S,c(S),\mathcal{W}}$. Next, we evaluate h on every $x \in X_p$ and correct the mistakes using Z. Again, we can also recover $c(x)$ for every $x \notin X_p$ from $\mathcal{H}_p$.

The above procedures thus give a prefix free encoding of c conditioned on $S, c(S)$ and $\mathcal{H}_1, \dots, \mathcal{H}_p, X_1, \dots, X_p$. We now analyse its expected length to derive an upper bound on the conditional entropy of c. We have that the expected length of the encoding is

$$1 + (1 - \Pr[E])2m\beta^{-p} + \Pr[E](1 + \lg m + \mathbb{E}[\Delta \lg(2em\beta^{-p}/\Delta) \mid E])$$

Note that $x \lg(a/x) = \lg(e)(x \ln(a/x)) = \lg(e)(x \ln(a) - x \ln(x))$ has derivative $\lg(e)(\ln(a) - (\ln(x) + 1))$ and second derivative $-\lg(e)/x$. Thus it is a concave function of x and $\mathbb{E}[\Delta \lg(2em\beta^{-p}/\Delta) \mid E] \leq \mathbb{E}[\Delta \mid E] \lg(2em\beta^{-p}/\mathbb{E}[\Delta \mid E])$. This is an increasing function of $\mathbb{E}[\Delta \mid E]$ (note that $\Delta \leq 2m\beta^{-p}$) and thus we conclude

$$\begin{aligned} H(c \mid S, c(S), \mathcal{H}_1, \dots, \mathcal{H}_p, X_1, \dots, X_p) &\leq \\ 2 + \lg m + (1 - \Pr[E])2m\beta^{-p} + \Pr[E](2m\eta/\Pr[E]) \lg(2em \Pr[E]\beta^{-p}/(2m\eta)) &\leq \\ 2 + \lg m + (1 - \Pr[E])2m\beta^{-p} + 2m\eta \lg(e\beta^{-p}/\eta). \end{aligned}$$

■

The Event E is Likely. What remains is to prove Lemma 4, i.e. show that $\Pr[E] \geq 49/50$ for the choice of β in Lemma 4. For this, let E_i denote the event that in the first i rounds of querying the weak-learner $\mathcal{W}$, it holds that $\mathcal{W}$ only returns hypotheses from $\mathcal{H}_1, \dots, \mathcal{H}_i$.

We aim to show that $\Pr[E_i \mid \cap_{j=1}^{i-1} E_j]$ is large. For this, notice that conditioned on $\cap_{j=1}^{i-1} E_j$ and any outcome of $S, c(S), X_1, \dots, X_{i-1}, \mathcal{H}_1, \dots, \mathcal{H}_{i-1}$, all the t queries made by $\mathcal{A}$ in round i are fixed. This is because we can simulate the replies of the weak learner $\mathcal{W}$ during the first $i-1$ rounds from this information alone, as it always manages to return a hypothesis from $\mathcal{H}_1, \dots, \mathcal{H}_{i-1}$ when we condition on $\cap_{j=1}^{i-1} E_j$. Let $\mathcal{Q}_i(S, c(S), X_1, \dots, X_{i-1}, \mathcal{H}_1, \dots, \mathcal{H}_{i-1})$ denote the set of queries made by $\mathcal{A}$ in round i on $S, c(S), X_1, \dots, X_{i-1}, \mathcal{H}_1, \dots, \mathcal{H}_{i-1}$. Thus $\mathcal{Q}_i$ is a set of distributions.

Now observe that conditioned on any outcome of $S, c(S), X_1, \dots, X_{i-1}, \mathcal{H}_1, \dots, \mathcal{H}_{i-1}$, the set X_i is a uniform random $\beta^{-1}|X_{i-1}|$-sized subset of X_{i-1}. Thus for a query distribution $\mathcal{D}' \in \mathcal{Q}_i$, we can bound the probability that there is no $h \in \mathcal{H}_1, \dots, \mathcal{H}_i$ with advantage γ for $\mathcal{D}'$.

We split the proof in two parts.

Well-Spread Queries. In the first case, there is no set of $\rho = ad\gamma^{-2}$ elements receiving at least $1/4$ of the probability mass under $\mathcal{D}'$, where $a > 0$ is a sufficiently small constant.

We say that X_i *avoids* $\mathcal{D}'$ if $\Pr_{x \sim \mathcal{D}'}[x \notin X_i] \geq 2\gamma$. We claim that if X_i avoids $\mathcal{D}'$, then there is an $h \in \mathcal{H}_i$ with $\mathcal{L}_{\mathcal{D}'}(h) \leq 1/2 - \gamma$ with probability $1 - \exp(-\exp(\Omega(d)))$. We start by proving this.

Assume X_i avoids $\mathcal{D}'$ and let h be one of the $2^{d/2}$ random hypotheses from $\mathcal{H}_i$ that equal c for all $x \notin X_i$. Since $h(x)$ is uniform in $\{-1, 1\}$ for all $x \in X_i$, it holds with probability $1/2$ that $\sum_{x \in X_i} c(x)h(x) \geq 0$. When this happens, we have $\mathbb{E}_{x \sim \mathcal{D}'}[h(x)c(x)] \geq \sum_{x \notin X_i} \mathcal{D}'(x)c(x)h(x) = \sum_{x \notin X_i} \mathcal{D}'(x) = \Pr_{x \sim \mathcal{D}'}[x \notin X_i] \geq 2\gamma$. This implies that $\mathcal{L}_{\mathcal{D}'}(h) \leq 1/2 - \gamma$ and thus h is a valid response to the query $\mathcal{D}'$. Since this holds with probability $1/2$ for each of the $2^{d/2}$ random hypotheses, it holds with probability $1 - 2^{-2^{d/2}} = 1 - \exp(-\exp(\Omega(d)))$ that there is a hypothesis $h \in \mathcal{H}_i$ that is a valid response to $\mathcal{D}'$ for $\mathcal{W}$. We thus need to bound the probability that X_i avoids $\mathcal{D}'$.

Consider first the largest ρ elements under $\mathcal{D}'$. Denote this set Y. We know that $\mathcal{D}'$ puts at least $3/4$ probability mass outside Y and that no element outside has mass more than $1/\rho$.

Let $Z = \mathcal{X} \setminus (X_i \cup Y)$. Our goal is to show that with high probability (over X_i), we have $\Pr_{x \sim \mathcal{D}'}[x \in Z] \geq 2\gamma$, which implies that X_i avoids $\mathcal{D}'$.

Define an indicator I_x for every $x \in \mathcal{X} \setminus Y$, taking the value 1 if $x \notin X_i$ and 0 otherwise. We want to show that $\Pr_{X_i}[\Pr_{x \sim \mathcal{D}'}[x \in Z] < 2\gamma] = \Pr_{X_i}[\sum_{x \in \mathcal{X} \setminus Y} I_x \mathcal{D}'(x) < 2\gamma]$ is small. We start by showing that $\mu = \mathbb{E}_{X_i}[\sum_{x \in \mathcal{X} \setminus Y} I_x \mathcal{D}'(x)]$ is large. Here we notice that if $x \notin X_{i-1}$, then $I_x = 1$ with probability 1. For $x \in X_i$, we have $\Pr[I_x = 1] = 1 - |X_i|/|X_{i-1}| = 1 - \beta^{-1}$. We thus have $\mu \geq (1 - \beta^{-1}) \sum_{x \in \mathcal{X} \setminus Y} \mathcal{D}'(x) \geq (3/4)(1 - \beta^{-1})$. Since Lemma 4 constrains $\beta \geq 1 + 32\gamma$, we have $2\gamma \leq (\beta - 1)/16 = (\beta^{-1}\beta - \beta^{-1})/(16\beta^{-1}) = (1 - \beta^{-1})/(16\beta^{-1}) \leq (1 - \beta^{-1})/16$. We thus need to bound the probability that the sum drops to less than half of its expectation.

Now consider the population $\{\mathcal{D}'(x) : x \in X_{i-1} \setminus Y\}$ and add it to it $|Y \cap X_{i-1}|$ 0's. Then $\sum_{x \in X_{i-1} \setminus Y} I_x \mathcal{D}'(x)$ is distributed as the sum of $\Gamma = |X_{i-1}| - |X_i|$ samples $R_1, \dots, R_\Gamma$ without replacement from this population. Furthermore, every element in the population is bounded by $1/\rho$ in value. In the appendix, Section A, we show how to use a version of the Chernoff bound for sampling without replacement to bound $\Pr[\sum_i R_i < \mu/2]$ by $\exp(-\rho\mu/8)$.

Since we defined $\rho = ad\gamma^{-2}$, this probability is at most $\exp\left(-\Omega(d(1-\beta^{-1})\gamma^{-2})\right)$.

A union bound over all t queries $\mathcal{D}'$ in $\mathcal{Q}_i$ implies that X_i avoids them all with probability at least $1 - t\exp\left(-\Omega(d(1-\beta^{-1})\gamma^{-2})\right)$. When this happens, another union bound implies that there is a valid response to all such $\mathcal{D}'$ with probability $1 - t\exp(-\exp(\Omega(d)))$.

Concentrated Queries. Next, consider a $\mathcal{D}'$ with some set Y of $\rho = ad\gamma^{-2}$ elements receiving at least $1/4$ probability mass. For such $\mathcal{D}'$, we consider the $2^{d/2}$ random $h \in \mathcal{H}_i$ that return a uniform random value for every $x \in \mathcal{X}$. We want to show that each of these is reasonably likely to have the desired advantage of γ on $\mathcal{D}'$. For this, first define the following function of a vector $w \in \mathbb{R}^n$ and value $t > 0$:

$$F(w,t) = \sum_{i=1}^{\lfloor t^2 \rfloor} |w_{(i)}| + t\left(\sum_{j=\lfloor t^2 \rfloor+1}^{n} w_{(j)}^2\right)^{1/2}.$$

Here $w_{(i)}$ denotes the i'th largest entry of w in absolute value.

With this definition, The following theorem by Montgomery-Smith shows that it is rather likely that any such hypothesis h provides the desired advantage

Theorem 6 (Montgomery-Smith (1990)) *There exists universal constants $a_1, a_2 > 0$, such that the following holds: For any vector $w \in \mathbb{R}^n$, if x has uniform random and independent entries in $\{-1,1\}$, then for all $t > 0$: $\Pr\left[\langle w, x\rangle > a_1 F(w,t)\right] \geq a_2^{-1}\exp(-a_2 t^2)$.*

Now let w be the vector in $\mathbb{R}^\rho$ with entries containing the values $\mathcal{D}'(x)c(x)$ for $x \in Y$. For the random h, we invoke Theorem 6 with $t = a_3\sqrt{d}$ for a small constant $a_3 > 0$. Assume first that the largest $\lfloor t^2 \rfloor$ entries of w sum to more than $1/8$. Then for $\gamma < 1/16$, we immediately get from Theorem 6 that $\sum_{x \in Y} \mathcal{D}'(x)c(x)h(x) \geq 2\gamma$ with probability at least $\exp(-O(t)) = \exp(-O(a_3^2 d))$. If on the other hand the largest entries sum to less than $1/8$, then the remaining sum to at least $1/8$. Since $\mathbb{R}^\rho$ has ρ coordinates, it follows by Cauchy-Schwartz that $\sqrt{\sum_{j=\lfloor t^2 \rfloor+1}^{\rho} w_{(j)}^2} \geq 1/(8\sqrt{\rho})$ and thus from Theorem 6, we get $\sum_{x\in Y} \mathcal{D}'(x)c(x)h(x) = \Omega(t/\sqrt{\rho}) = \Omega(a_3\sqrt{d}/\sqrt{ad\gamma^{-2}}) \geq 2\gamma$ with probability $\exp(-O(t)) = \exp(-O(a_3^2 d))$. Here we use that $a > 0$ is sufficiently small compared to a_3. Finally note that $\sum_{x \notin Y} \mathcal{D}'(x)h(x)c(x)$ is at least 0 with probability $1/2$ and is independent of $\sum_{x\in Y} \mathcal{D}'(x)h(x)c(x)$.

The probability that there is no valid response to $\mathcal{D}'$ in $\mathcal{H}_i$ is thus at most $(1 - \exp(-O(a_3^2 d)))^{2^{d/2}} \leq \exp(-\exp(\Omega(d)))$ for a_3 sufficiently small. A union bound over all such $\mathcal{D}'$ in $\mathcal{H}_i$ finally shows that $\mathcal{H}_i$ has a valid response h to each of them with probability at least $1 - t\exp(-\exp(\Omega(d)))$.

Summary. We finally conclude from the above that

$$\Pr[E_i \mid \cap_{j=1}^{i-1} E_j] \quad \geq \quad 1 - t(\exp(-\exp(\Omega(d))) + \exp(-\Omega(d(1-\beta^{-1})\gamma^{-2}))).$$

which gives us

$$\Pr[E] \geq \left(1 - t(e^{-\exp(\Omega(d))} + e^{-\Omega(d(1-\beta^{-1})\gamma^{-2})})\right)^p \geq 1 - tp(e^{-\exp(\Omega(d))} + e^{-\Omega(d(1-\beta^{-1})\gamma^{-2})}).$$

Lemma 4 sets $\beta \geq 1+a' \ln(tp)\gamma^2/d$ for a large constant a'. We have $1-\beta^{-1} = (\beta-1)/\beta$. This is increasing in β and thus is at least $(a' \ln(tp)\gamma^2/d)/(1+a' \ln(tp)\gamma^2/d)$. Since we assume $tp = \exp(O(d\gamma^{-2}))$, we have $a' \ln(tp)\gamma^2/d \leq 1$ for sufficiently small constant in the O-notation (depending on a'). Thus $1-\beta^{-1} \geq (a'/2)\ln(tp)\gamma^2/d$. This finally implies $e^{-\Omega(d(1-\gamma^{-1})\gamma^{-2})} \leq 1/(100tp)$ for big enough a'. By assumption, we also have $tp \leq \exp(\exp(O(d)))$, implying $tpe^{-\exp(\Omega(d))} \leq 1/100$ and thus $\Pr[E] \geq 49/50$ as desired. This concludes the proof of Lemma 4 and thereby also the proof of Theorem 1.

3. Single-Round Boosting

In this section, we demonstrate a single-round parallel boosting algorithm with good generalization performance. Our algorithm initially asks a number of queries in parallel to a γ-weak learner. Following that, it has several sequential rounds in which it decides how to combine the obtained hypotheses. We comment that this is a single-round algorithm according to our definition of parallel complexity. Moreover, invoking the weak learner effectively corresponds to training a model, whereas the later sequential steps of our algorithm use only inference. In practical setups where training is significantly more expensive than inference, this may still be a major speed up. Furthermore, our algorithm demonstrates the near-tightness of our lower bound.

The goal of our algorithm is to produce a voting classifier with large margins on all training samples. Concretely, given a γ-weak learner $\mathcal{W}$ and m samples $S \sim \mathcal{D}^m$, our goal is to produce a voting classifier $f(x) = \text{sign}(g(x))$ with $g(x) = (1/k)\sum_{i=1}^{k} h_i(x)$ such that for every training sample $(x, c(x)) \in S$, it holds that $c(x)g(x) \geq \gamma/16$. Generalization of this voting classifier then follows from generalization bounds for voting classifiers with large margins, see e.g. Breiman (1999); Gao and Zhou (2013); Bartlett et al. (1998). We note that we assume that the weak learner $\mathcal{W}$ *always* returns a hypothesis with error at most $1/2-\gamma$ under the distribution $\mathcal{D}$ it is queried with.

Algorithm. For a large enough constant $a > 0$, our algorithm simply queries the weak learner $\mathcal{W}$ for every distribution $\mathcal{D}_T$ with $T \subseteq S$, $|T| = ad\gamma^{-2}$, where T is a multiset. Here $\mathcal{D}_T$ is the uniform distribution over T, putting a mass of $t/|T|$ on an element that occurs t times in the multiset. These queries can clearly be performed in parallel and there are no more than $m^{ad\gamma^{-2}} = \exp(O(d\gamma^{-2}\ln m))$ such queries.

Let $h_T \in \mathcal{H}$ denote the hypothesis returned by $\mathcal{W}$ on the query $\mathcal{D}_T$. Having obtained the set of all h_T, we form g by running a version of AdaBoost shown in Algorithm 1.

We invoke Algorithm 1 with the training data set S as well as a number of rounds $K = 16\gamma^{-2}\ln m$.

We make a few remarks regarding the algorithm. First note that it is possible to check whether $\mathcal{L}_{\mathcal{D}_k}(h_k) > 1/2-\gamma/4$ because $\mathcal{D}_k$ is supported only on the training data, and thus the error probability of h_k under $\mathcal{D}_k$ can be computed exactly. Furthermore, note that it is not a given that h_k has $\mathcal{L}_{\mathcal{D}_k}(h_k) \leq 1/2-\gamma/4$ since h_k is obtained by invoking $\mathcal{W}$ on a distribution $\mathcal{D}_T \neq \mathcal{D}_k$.

The reader familiar with AdaBoost will also notice that the weight changes we make, i.e. the chosen w, is consistent with AdaBoost's weight updates if the hypothesis h_k was correct on precisely a $1/2+\gamma/4$ fraction of the samples when weighted according to $\mathcal{D}_k$.

Algorithm 1: Sampled Boosting

```
Input: Training set $S = \{(x_1, c(x_1)), \dots, (x_m, c(x_m))\}$,
  number of rounds $K$
Result: A voting classifier $g$
$\mathcal{D}_1 \leftarrow (\frac{1}{m}, \dots \frac{1}{m})$
$w \leftarrow \frac{1}{2}\ln(\frac{1/2+\gamma/4}{1/2-\gamma/4})$
$n \leftarrow ad\gamma^{-2}$
for every multiset $T \subseteq S$ with $|T| = n$ in parallel do
    Query weak learner $\mathcal{W}$ on $\mathcal{D}_T$ to obtain $h_T$
end
for $k = 1, \dots, K$ do
    Draw $n$ samples $T_k \sim \mathcal{D}_k^n$
    Let $h_k \leftarrow h_{T_k}$
    while $\mathcal{L}_{\mathcal{D}_k}(h_k) > 1/2 - \gamma/4$ do
        Re-draw $n$ samples $T_k \sim \mathcal{D}_k^n$
        Let $h_k \leftarrow h_{T_k}$
    end
    for $i \in \{1, \dots, m\}$ do
        $\mathcal{D}_{k+1}(i) \leftarrow \mathcal{D}_k(i)\exp(-wc(x_i)h_k(x_i))$
    end
    $Z_k \leftarrow \sum_{i=1}^{m} \mathcal{D}_k(i)\exp(-wc(x_i)h_k(x_i))$
    $\mathcal{D}_{k+1} \leftarrow \mathcal{D}_{k+1}/Z_k$
end
return $g(x) = \frac{1}{K}\sum_{i=1}^{K} h_i(x)$
```

Our algorithm is thus a variant of AdaBoost with uniform weight updates and weighing of hypotheses in the output g.

Analysis. We analyse our algorithm in two steps. First, we show that if the algorithm terminates, then the resulting voting classifier $g = (1/K)\sum_{i=1}^{K} h_i(x)$ has all margins at least $\gamma/16$. Next, we show that the algorithm always terminates. These two properties are captured in the following lemmas

Lemma 7 *If Algorithm 1 terminates, then the resulting voting classifier g satisfies $c(x)g(x) \geq \gamma/16$ for all $(x, c(x)) \in S$.*

Lemma 8 *Algorithm 1 always terminates.*

The main observation for the proof of Lemma 7 is that whenever our algorithm terminates, every hypothesis h_k that it uses, has $\mathcal{L}_{\mathcal{D}_k}(h_k) \leq 1/2 - \gamma/4$. Furthermore, we use the same weight adjustment w in all iterations and weigh the h_k equally in the output g. A previous analysis by Grønlund et al. (2019) of AdaBoost with identical weights has also shown that this results in margins of $\Omega(\gamma)$.

Proof (of Lemma 7) Consider the exponential loss

$$\sum_{i=1}^{m} \exp\left(-wc(x_i)\sum_{k=1}^{K} h_k(x_i)\right).$$

We compare this to the final weights $\mathcal{D}_{K+1}$. Here we note that

$$\begin{aligned} 1 &= \sum_{i=1}^{m} \mathcal{D}_{K+1}(i) \\ &= \sum_{i=1}^{m} \frac{\mathcal{D}_K(i)\exp(-wc(x_i)h_K(x_i))}{Z_k} \\ &= \frac{1}{m}\sum_{i=1}^{m} \frac{\exp(-wc(x_i)\sum_{k=1}^{K} h_k(x_i))}{\prod_{k=1}^{K} Z_k}. \end{aligned}$$

From this, we observe that

$$\sum_{i=1}^{m} \exp\left(-wc(x_i)\sum_{k=1}^{K} h_k(x_i)\right) = m\prod_{k=1}^{K} Z_k.$$

Next, let $er_k = \Pr_{x\sim\mathcal{D}_k}[h_k(x) \neq c(x)]$ and recall $er_k \leq 1/2 - \gamma/4$. Observe that

$$\begin{aligned} Z_k &= \sum_{i=1}^{m} \mathcal{D}_k(i)\exp(-wc(x_i)h_k(x_i)) \\ &= \sum_{i:h_k(x_i)\neq c(x_i)} \mathcal{D}_k(i)e^{w} + \sum_{i:h_k(x_i)=c(x_i)} \mathcal{D}_k(i)e^{-w} \\ &= er_k e^{w} + (1-er_k)e^{-w} \\ &\leq (1/2-\gamma/4)e^{w} + (1/2+\gamma/4)e^{-w} \\ &= 2\sqrt{(1/2-\gamma/4)(1/2+\gamma/4)} \\ &= \sqrt{1-\gamma^2/4}. \end{aligned}$$

Since $K = 16\gamma^{-2}\ln m$, we have $\prod_{k=1}^{K} Z_k \leq (1-\gamma^2/4)^{K/2} \leq \exp(-\gamma^2 K/8) \leq 1/m^2$. We therefore have

$$\sum_{i=1}^{m} \exp\left(-wc(x_i)\sum_{k=1}^{K} h_k(x_i)\right) \leq 1/m.$$

By non-negativity of the exponential function, this implies that $\exp(-wc(x_i)\sum_{k=1}^{K} h_k(x_i)) \leq 1/m$ for all i. Raising both sides of the inequality to the power $1/(Kw)$ gives $\exp(-c(x_i)g(x_i)) \leq 1/m^{1/Kw}$. Taking inverses and log gives $c(x_i)g(x_i) \geq \ln(m)/(Kw)$. Note that

$$\begin{aligned} w &= \frac{1}{2}\ln((1/2+\gamma/4)/(1/2-\gamma/4)) \\ &\leq \frac{1}{2}\ln(1+2\gamma) \\ &\leq \gamma. \end{aligned}$$

Hence we conclude $c(x_i)g(x_i) \geq \ln(m)/(Kw) \geq \gamma/16$. ■

To prove termination (Lemma 8), we first need to recall the notion of an ε-approximation. For a concept $c : \mathcal{X} \to \{-1, 1\}$, a hypothesis set $\mathcal{H}$ and a distribution $\mathcal{D}$ over $\mathcal{X}$, a set of samples T is an ε-approximation for $(c, \mathcal{D}, \mathcal{H})$ if for all $h \in \mathcal{H}$, it holds that

$$|\Pr_{x \sim \mathcal{D}}[h(x) \neq c(x)] - |\{x \in T : h(x) \neq c(x)\}|/|T|| \leq \varepsilon.$$

The following classic result shows that a small random sample $T \sim \mathcal{D}^n$ is an ε-approximation with good probability.

Theorem 9 (Li et al. (2001); Talagrand (1994); Vapnik and Chervonenkis (1971)) *There is a universal constant $b > 0$, such that for any $0 < \varepsilon, \delta < 1$, $\mathcal{H} \subseteq \mathcal{X} \to \{-1, 1\}$ of VC-dimension d and distribution $\mathcal{D}$ over $\mathcal{X}$, it holds with probability at least $1 - \delta$ over a set $T \sim \mathcal{D}^n$ that T is an ε-approximation for $(c, \mathcal{D}, \mathcal{H})$ provided that $n \geq b((d + \ln(1/\delta))\varepsilon^{-2})$.*

With this, we are ready to prove that Algorithm 1 terminates.

Proof (of Lemma 8) We need to show that the while-loop always terminates. So consider some distribution $\mathcal{D}_k$ in round k of Algorithm 1. We claim the while-loop terminates if the sample T_k is a $\gamma/2$-approximation for $(c, \mathcal{D}_k, \mathcal{H})$. To see this, note that if it is a $\gamma/2$-approximation, then

$$\left| \Pr_{x \sim \mathcal{D}_k}[h(x) \neq c(x)] - \frac{|\{x \in T_k : h(x) \neq c(x)\}|}{|T_k|} \right| \leq \gamma/2.$$

Furthermore, we have that $h_k = h_{T_k}$ was obtained from the weak learner $\mathcal{W}$ using the distribution $\mathcal{D}_{T_k}$. Thus

$$1/2 - \gamma \leq \Pr_{x \sim \mathcal{D}_{T_k}}[h_k(x) \neq c(x)] = |\{x \in T_k : h_k(x) \neq c(x)\}|/|T_k|.$$

It follows that $\Pr_{x \sim \mathcal{D}_k}[h_k(x) \neq c(x)] \leq (1/2 - \gamma) + \gamma/2 = 1/2 - \gamma/2$. The conclusion now follows from Theorem 9 which says that we have a $\gamma/2$-approximation with constant probability in every iteration of the while-loop, provided that $n \geq b(d+2)\gamma^{-2}/4$, which is indeed satisfied for our choice of $n = ad\gamma^{-2}$ for a large enough. ■

Generalization Performance. We finally conclude by observing that our algorithm always produces a voting classifier $f = \text{sign}(g(x))$ where $c(x)g(x) \geq \gamma/16$ for all $(x, c(x)) \in S$ (Lemma 7 and Lemma 8). We now invoke Breiman's min-margin bound

Theorem 10 (Breiman (1999)) *Let $c : \mathcal{X} \to \{-1, 1\}$ be an unknown concept, $\mathcal{H} \subseteq \mathcal{X} \to \{-1, 1\}$ a hypothesis set of VC-dimension d and $\mathcal{D}$ an arbitrary distribution over $\mathcal{X}$. There is a universal constant $a > 0$ such that with probability at least $1 - \delta$ over a set of m samples $S \sim \mathcal{D}^m$, it holds for every voting classifier $f(x) = \text{sign}(g(x))$ with $c(x)g(x) \geq \gamma$ for all $(x, c(x)) \in S$ that*

$$\mathcal{L}_{\mathcal{D}}(f) \leq a \cdot \frac{d \ln(m) \ln(m/d) + \ln(1/\delta)}{\gamma^2 m}.$$

The generalization of our single-round boosting algorithm follows immediately from Theorem 10 and the observations above.

4. Conclusion

In this work, we established strong barriers for the possibilities of parallelizing boosting. Concretely, to parallelize below $O(\gamma^{-1} \ln m)$ rounds incurs an exponential blow-up in the number of invocations of a weak learner.

The classic algorithms, such as AdaBoost, use $O(\gamma^{-2} \ln m)$ rounds. Thus it is conceivable that boosting can be somewhat parallelized. We leave this as an exciting direction for future research.

We also complemented our lower bound by a near-optimal single-round boosting algorithm. A slightly unsatisfactory aspect of our single-round algorithm is that performs sequential work in $O(\gamma^{-2} \ln m)$ rounds after having invoked the weak learner/trained a number of hypotheses. It would be interesting to obtain a truly parallel algorithm also when accounting for post-processing of trained models.

References

Jacob Abernethy, Pranjal Awasthi, and Satyen Kale. A multiclass boosting framework for achieving fast and provable adversarial robustness. *arXiv preprint arXiv:2103.01276*, 2021.

Noga Alon, Alon Gonen, Elad Hazan, and Shay Moran. Boosting simple learners. In *STOC*, pages 481–489. ACM, 2021.

Peter Bartlett, Yoav Freund, Wee Sun Lee, and Robert E. Schapire. Boosting the margin: a new explanation for the effectiveness of voting methods. *The Annals of Statistics*, 26 (5):1651 – 1686, 1998.

Leo Breiman. Prediction games and arcing algorithms. *Neural computation*, 11(7):1493–1517, 1999.

Nataly Brukhim, Elad Hazan, and Karan Singh. A boosting approach to reinforcement learning. *arXiv preprint arXiv:2108.09767*, 2021.

Tianqi Chen and Carlos Guestrin. Xgboost: A scalable tree boosting system. In *KDD*, pages 785–794. ACM, 2016. ISBN 978-1-4503-4232-2.

Y. Freund. Boosting a weak learning algorithm by majority. *Information and Computation*, 121(2):256–285, 1995. ISSN 0890-5401.

Yoav Freund and Robert 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.

Jerome H. Friedman. Greedy function approximation: A gradient boosting machine. *The Annals of Statistics*, 29(5):1189 – 1232, 2001.

Wei Gao and Zhi-Hua Zhou. On the doubt about margin explanation of boosting. *Artif. Intell.*, 203:1–18, 2013.

Allan Grønlund, Lior Kamma, Kasper Green Larsen, Alexander Mathiasen, and Jelani Nelson. Margin-based generalization lower bounds for boosted classifiers. *Advances in Neural Information Processing Systems*, 32, 2019.

Jenny Hamer, Mehryar Mohri, and Ananda Theertha Suresh. FedBoost: A communication-efficient algorithm for federated learning. In *Proceedings of the 37th International Conference on Machine Learning*, Proceedings of Machine Learning Research, 2020.

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

Adam Tauman Kalai and Rocco A. Servedio. Boosting in the presence of noise. *J. Comput. Syst. Sci.*, 71(3):266–290, 2005.

Guolin Ke, Qi Meng, Thomas Finley, Taifeng Wang, Wei Chen, Weidong Ma, Qiwei Ye, and Tie-Yan Liu. Lightgbm: A highly efficient gradient boosting decision tree. In *NIPS*, 2017.

Michael Kearns. Learning boolean formulae or finite automata is as hard as factoring. *Technical Report TR-14-88 Harvard University Aikem Computation Laboratory*, 1988.

Michael Kearns and Leslie Valiant. Cryptographic limitations on learning boolean formulae and finite automata. *Journal of the ACM (JACM)*, 41(1):67–95, 1994.

Kasper Green Larsen. Bagging is an optimal PAC learner. *arXiv preprint*, arXiv/2212.02264, 2022.

Kasper Green Larsen and Martin Ritzert. Optimal weak to strong learning. *Advances in Neural Information Processing Systems (NeurIPS 2022)*, 2022. To appear.

Y. Li, P.M. Long, and A. Srinivasan. Improved bounds on the sample complexity of learning. *Journal of Computer and System Sciences*, 62:516 – 527, 2001.

Philip M. Long and Rocco A. Servedio. Martingale boosting. In Peter Auer and Ron Meir, editors, *Learning Theory, 18th Annual Conference on Learning Theory, COLT 2005, Bertinoro, Italy, June 27-30, 2005, Proceedings*, volume 3559 of *Lecture Notes in Computer Science*, pages 79–94. Springer, 2005.

Philip M. Long and Rocco A. Servedio. Algorithms and hardness results for parallel large margin learning. *J. Mach. Learn. Res.*, 14(1):3105–3128, jan 2013. ISSN 1532-4435.

Yishay Mansour and David A. McAllester. Boosting using branching programs. *J. Comput. Syst. Sci.*, 64(1):103–112, 2002.

S. J. Montgomery-Smith. The distribution of rademacher sums. *Proceedings of the American Mathematical Society*, 109(2):517–522, 1990.

Alexey Natekin and Alois Knoll. Gradient boosting machines, a tutorial. *Frontiers in Neurorobotics*, 7, 2013. ISSN 1662-5218.

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

Shai Shalev-Shwartz and Shai Ben-David. *Understanding machine learning: From theory to algorithms*. Cambridge university press, 2014.

Zebang Shen, Hamed Hassani, Satyen Kale, and Amin Karbasi. Federated functional gradient boosting. In *International Conference on Artificial Intelligence and Statistics*, pages 7814–7840. PMLR, 2022.

M. Talagrand. Sharper Bounds for Gaussian and Empirical Processes. *The Annals of Probability*, 22(1):28 – 76, 1994.

V. N. Vapnik and A. Ya. Chervonenkis. On the uniform convergence of relative frequencies of events to their probabilities. *Theory of Probability and its Applications*, 16(2):264–280, 1971.

Appendix A. Sampling Without Replacement

We prove the following lemma for sampling without replacement from a finite population

Lemma 11 *Let $Y_1, \dots, Y_n$ be samples without replacement from a finite population of real values between 0 and $1/\rho$ for $\rho > 0$. Then for any $0 < \delta < 1$ and any $\mu \leq \mathbb{E}[\sum_i Y_i]$, it holds that*

$$\Pr[\sum_i Y_i \leq (1-\delta)\mu] \leq \exp(-\rho\delta^2\mu/2)$$

and for any $\gamma \geq \mathbb{E}[\sum_i Y_i]$ it holds that

$$\Pr[\sum_i Y_i \geq (1+\delta)\mu] \leq \exp(-\rho\delta^2\mu/3).$$

Proof We start by rescaling all elements in the population by a factor ρ. This changes $\mathbb{E}[\sum_i Y_i]$ by a factor ρ and guarantees that all values are between 0 and 1. We thus need to show that the two probabilities are bounded by $\exp(-\delta^2\mu/2)$ and $\exp(-\delta^2\mu/3)$ for these rescaled variables. Notice that these two are precisely the standard Chernoff bounds for sums of independent 0/1 random variables. The proof of the Chernoff bounds follows from proving that

$$\mathbb{E}[\exp(t\sum_i Y_i)] \leq \prod_i \left(1 + \mathbb{E}[Y_i](e^t - 1)\right). \tag{5}$$

for any $t \in \mathbb{R}$. We thus prove (5) for our rescaled variables and refer the reader to standard proofs of the Chernoff bound from thereon.

Now let $X_1, \dots, X_n$ be independent and uniform random samples with replacement from the population. Then $\mathbb{E}[\sum_i Y_i] = \mathbb{E}[\sum_i X_i]$. Since $x \to \exp(tx)$ is a convex and continous function for any $t \in \mathbb{R}$, it follows from Hoeffding (1963) that $\mathbb{E}[\exp(t\sum_i Y_i)] \leq \mathbb{E}[\exp(t\sum_i X_i)] = \prod_i \mathbb{E}[\exp(tX_i)]$. Again using convexity of $\exp(tx)$, we get that for any $x \in [0,1]$, it holds that $\exp(tx) \leq (1-x)\exp(t0) + x\exp(t1)$. Since we rescaled the population, every X_i takes values between 0 and 1. Hence $\mathbb{E}[\exp(tX_i)] \leq \mathbb{E}[(1-X_i) + X_i\exp(t)] = 1 + \mathbb{E}[X_i](e^t - 1) = 1 + \mathbb{E}[Y_i](e^t - 1)$. ■