Bagging is an Optimal PAC Learner (Extended Abstract)[*]

Kasper Green Larsen
Aarhus University
larsen@cs.au.dk

Abstract

Determining the optimal sample complexity of PAC learning in the realizable setting was a central open problem in learning theory for decades. Finally, seminal work by Hanneke gave an algorithm with a provably optimal sample complexity. His algorithm is based on a careful and structured sub-sampling of the training data and then returning a majority vote among hypotheses trained on each of the sub-samples. While being a very exciting theoretical result, it has not had much impact in practice, in part due to inefficiency, since it constructs a polynomial number of sub-samples of the training data, each of linear size.

In this work, we prove the surprising result that the practical and classic heuristic *bagging* (a.k.a. bootstrap aggregation), due to Breiman, is in fact also an optimal PAC learner. Bagging pre-dates Hanneke's algorithm by twenty years and is taught in most undergraduate machine learning courses. Moreover, we show that it only requires a logarithmic number of sub-samples to reach optimality.

1 Introduction

PAC learning, or probably approximately correct learning [Valiant, 1984], is the most classic theoretical model for studying classification problems in supervised learning. For binary classification in the *realizable* setting, the goal is to design a learning algorithm that with probability $1 - \delta$ over a random training data set, outputs a hypothesis that mispredicts the label of a new random sample with probability at most ε. More formally, one assumes that samples come from an input domain $\mathcal{X}$ and that there is an unknown *concept* $c : \mathcal{X} \to \{-1, 1\}$ that we are trying to learn. The realizable setting means that c belongs to a predefined concept class $\mathbb{C} \subseteq \mathcal{X} \to \{-1, 1\}$ and that the correct label of any $x \in \mathcal{X}$ is always $c(x)$.

For the above learning task, a learning algorithm $\mathcal{A}$ receives a training data set $\mathbf{S}$ of m i.i.d. samples $(\mathbf{x}_1, c(\mathbf{x}_1)), \dots, (\mathbf{x}_m, c(\mathbf{x}_m))$ where each $\mathbf{x}_i$ is drawn independently from an *unknown* data distribution $\mathcal{D}$ over $\mathcal{X}$. From this data set, the learning algorithm must output a hypothesis $h_{\mathbf{S}} : \mathcal{X} \to \{-1, 1\}$. The algorithm $\mathcal{A}$ is a PAC learner, if for any distribution $\mathcal{D}$ and any concept $c \in \mathbb{C}$, it holds that if $\mathcal{A}$ is given enough i.i.d. training samples $\mathbf{S}$, then with probability at least $1 - \delta$, the hypothesis $h_{\mathbf{S}}$ that it outputs satisfies $\mathcal{L}_{\mathcal{D}}(h_{\mathbf{S}}) = \Pr_{\mathbf{x} \sim \mathcal{D}}[h(\mathbf{x}) \neq c(\mathbf{x})] \leq \varepsilon$. We remark that the algorithm $\mathcal{A}$ knows the concept class $\mathbb{C}$, but not the data distribution $\mathcal{D}$. Determining the minimum number of samples $\mathcal{M}(\varepsilon, \delta)$, as a function of ε, δ and the VC-dimension [Vapnik and Chervonenkis, 1971] d of $\mathbb{C}$ (see Section 2 for a formal definition of VC-dimension) needed for this learning task, is one of the fundamental problems in PAC learning.

The most natural learning algorithm for the above is *empirical risk minimization* (ERM). Here a learning algorithm simply outputs an arbitrary hypothesis/concept $h_{\mathbf{S}} \in \mathbb{C}$ that correctly predicts the labels of the training data, i.e. it has $h_{\mathbf{S}}(\mathbf{x}_i) = c(\mathbf{x}_i)$ for all $(\mathbf{x}_i, c(\mathbf{x}_i)) \in \mathbf{S}$. Clearly such a hypothesis exists since $c \in \mathbb{C}$. Such a learning algorithm is referred to as a *proper* learner as it outputs a hypothesis/concept from the concept class $\mathbb{C}$. ERM is known to obtain a sample complexity of $O(\varepsilon^{-1}(d \lg(1/\varepsilon) + \lg(1/\delta)))$ [Vapnik, 1982; Blumer *et al.*, 1989]. Moreover, it can be shown that this analysis cannot be tightened, i.e. there are distributions $\mathcal{D}$ and concept classes $\mathbb{C}$ where any proper learner needs $\Omega(\varepsilon^{-1}(d \lg(1/\varepsilon) + \lg(1/\delta)))$ samples [Auer and Ortner, 2004]. However, a PAC learning algorithm is not necessarily required to output a hypothesis $h \in \mathbb{C}$. That better strategies might exist may seem counter-intuitive at first, since we are promised that the unknown concept c lies in $\mathbb{C}$. Nonetheless, the strongest known lower bounds for arbitrary PAC learning algorithms only show that $\Omega(\varepsilon^{-1}(d + \lg(1/\delta)))$ samples are necessary [Blumer *et al.*, 1989; Ehrenfeucht *et al.*, 1989]. This leaves a gap of a factor $\lg(1/\varepsilon)$ between ERM and the lower bound for arbitrary algorithms.

Despite its centrality, closing this gap remained a big open problem for more than thirty years. Finally, in 2016, Hanneke [2016] built on ideas by Simon [2015] and presented an algorithm with an asymptotically optimal sample complexity of $\mathcal{M}(\varepsilon, \delta) = O(\varepsilon^{-1}(d + \lg(1/\delta)))$. His algorithm is based on constructing a number of subsets $\mathbf{S}_i \subset \mathbf{S}$ of the training data $\mathbf{S}$ with carefully designed overlaps between the $\mathbf{S}_i$'s (see Section 2). He then runs ERM on each $\mathbf{S}_i$ to obtain hypotheses $h_{\mathbf{S}_i} \in \mathbb{C}$ and finally outputs the hypothesis $f_{\mathbf{S}}$ taking the majority vote $f_{\mathbf{S}}(x) = \operatorname{sign}(\sum_i h_{\mathbf{S}_i}(x))$ among the $h_{\mathbf{S}_i}$'s.

[*] Original paper [Larsen, 2023] appeared at COLT'23.

While being a major theoretical breakthrough, Hanneke's algorithm has unfortunately not had any significant practical impact. One explanation is that it requires a rather large number of sub-samples $\mathbf{S}_i$. Concretely, with optimal $m = \Theta(\varepsilon^{-1}(d + \lg(1/\delta)))$ samples, it requires $m^{\lg_4 3} \approx m^{0.79}$ sub-samples of linear size $|\mathbf{S}_i| = \Omega(m)$, resulting in a somewhat slow learning algorithm.

Our Contribution. In this work, we present an alternative optimal PAC learner in the realizable setting. Surprisingly, our algorithm is not new, but actually pre-dates Hanneke's algorithm by twenty years. Concretely, we show that the heuristic known as bagging (bootstrap aggregation) by Breiman [1996], also gives an optimal PAC learner. Bagging, and its slightly more involved extension known as *random forest* [Breiman, 2001], have proved very efficient in practice and are classic topics in introduction to machine learning courses.

In bagging, for t iterations, we sample a subset $\mathbf{S}_i$ of n independent and uniform samples with replacement from $\mathbf{S}$. We then run ERM on each $\mathbf{S}_i$ to produce hypotheses $h_{\mathbf{S}_i} \in \mathbb{C}$ and finally output the hypothesis $f_{\mathbf{S}_1,\dots,\mathbf{S}_t}$ taking the majority vote $f_{\mathbf{S}_1,\dots,\mathbf{S}_t}(x) = \operatorname{sign}(\sum_i h_{\mathbf{S}_i}(x))$ among the $h_{\mathbf{S}_i}$'s. The sub-samples $\mathbf{S}_i$ are referred to as *bootstrap samples*.

While being similar to Hanneke's algorithm, it is simpler to construct the subsets $\mathbf{S}_i$, and also, we show that it suffices with just $t = O(\lg(m/\delta))$ bootstrap samples of size n for any $0.02m \le n \le m$, which should be compared to Hanneke's algorithm requiring $m^{0.79}$ subsets (where m is optimal $\Theta(\varepsilon^{-1}(d+\lg(1/\delta)))$). For ease of notation, let $\mathcal{D}_c$ denote the distribution of a pair $(\mathbf{x}, c(\mathbf{x}))$ with $\mathbf{x} \sim \mathcal{D}$. Our result is then formalized in the following theorem

Theorem 1. *There is a universal constant $a > 0$ such that for every $0 < \delta < 1$, every distribution $\mathcal{D}$ over an input domain $\mathcal{X}$, every concept class $\mathbb{C} \subseteq \mathcal{X} \to \{-1, 1\}$ of VC-dimension d and every $c \in \mathbb{C}$, if $t \ge 18\ln(2m/\delta)$ and $0.02m \le n \le m$, then it holds with probability at least $1 - \delta$ over the random choice of a training set $\boldsymbol{S} \sim \mathcal{D}_c^m$ and t bootstrap samples $\boldsymbol{S}_1, \dots, \boldsymbol{S}_t \subset \boldsymbol{S}$ of size n, that the hypothesis $f_{\boldsymbol{S}_1,\dots,\boldsymbol{S}_t}$ produced by bagging satisfies*

$$\mathcal{L}_{\mathcal{D}}(f_{\boldsymbol{S}_1,\dots,\boldsymbol{S}_t}) = \Pr_{\boldsymbol{x}\sim\mathcal{D}}[f_{\boldsymbol{S}_1,\dots,\boldsymbol{S}_t}(\boldsymbol{x}) \ne c(\boldsymbol{x})] \le a \cdot \frac{d + \ln(1/\delta)}{m}.$$

Solving for m such that $\varepsilon \le \mathcal{L}_{\mathcal{D}}(g)$ gives a sample complexity of $O(\varepsilon^{-1}(d+\lg(1/\delta))$ as claimed. We remark that the constants 18 and 0.02 can be reduced at the cost of increasing the unspecified constant a.

In addition to providing an alternative and simpler algorithm for optimal PAC learning, we also believe there is much value in providing further theoretical justification for the wide practical success of bagging.

In Section 2, we first present Hanneke's optimal PAC learner and highlight the main ideas in his analysis. We then proceed to present a high-level overview of our proof of Theorem 1, which re-uses some of the ideas from Hanneke's proof.

2 Proof Overview

In this section, we first present Hanneke's PAC learning algorithm and discuss the main ideas in his analysis. We then proceed to give a high-level overview of the keys ideas in our proof that bagging is also an optimal PAC learning algorithm. For completeness, we start by recalling the definition of Vapnik-Chervonenkis dimension [Vapnik and Chervonenkis, 1971], or VC-dimension for short.

Algorithm 1: *Sub-Sample*(U, V)

```
Input: Two sets of training samples $U, V$.
Result: Collection consisting of sub-samples.
if $|U| < 4$ then
    return $\{U \cup V\}$.
else
    Partition $U$ into 4 disjoint sets $U_0, U_1, U_2, U_3$ of $|U|/4$ samples each.
    return
    $\bigcup_{i=1}^{3}$ Sub-Sample$(U_0, V \cup (\bigcup_{j\in\{1,2,3\}\setminus\{i\}} U_j))$.
```

A concept class $\mathbb{C} \subseteq \mathcal{X} \to \{-1, 1\}$ has VC-dimension d, where d is the largest integer such that there exists a set of d samples $x_1, \dots, x_d \in \mathcal{X}$ for which any labeling of the d samples can be realized by a concept $c \in \mathbb{C}$. That is, $|\{(c(x_1), \dots, c(x_d)) : c \in \mathbb{C}\}| = 2^d$. Throughout the paper, we assume $d \ge 1$ which is always true when $\mathbb{C}$ contains at least two distinct concepts. Also, as the reader may have observed, we consistently use **bold** face letters to denote random variables.

Hanneke's Algorithm and Analysis. As mentioned in Section 1, Hanneke's algorithm constructs a carefully selected collection of sub-samples of the training set $\mathbf{S}$. These sub-samples are constructed by invoking Algorithm 1 as *Sub-Sample*$(\mathbf{S}, \emptyset)$.

For simplicity, we have presented the *Sub-Sample* algorithm assuming that m is a power of 4.

Since $|U|$ is reduced by a factor 4 in each recursive call and there are 3 such calls, we get that the total number of sub-samples produced is $3^{\lg_4 m} = m^{\lg_4 3} \approx m^{0.79}$.

To analyse the hypothesis produced by invoking *Sub-Sample*$(\mathbf{S}, \emptyset)$, running ERM on each produced sub-sample $\mathbf{S}_i$ to produce hypotheses $h_{\mathbf{S}_i}$ and taking a majority vote $f_{\mathbf{S}}(x) = \operatorname{sign}(\sum_i h_{\mathbf{S}_i}(x))$, we zoom in on the recursive invocations of *Sub-Sample*$(\mathbf{U}, \mathbf{V})$. Such an invocation produces a number of sub-samples $\mathbf{S}_1, \dots, \mathbf{S}_t$ of $\mathbf{U} \cup \mathbf{V} \subseteq \mathbf{S}$. Letting $h_{\mathbf{S}_1}, \dots, h_{\mathbf{S}_t}$ denote the hypotheses obtained by running ERM on these sub-samples and $f_{(\mathbf{U},\mathbf{V})}(x) = \operatorname{sign}(\sum_i h_{\mathbf{S}_i}(x))$ the majority vote among them, we prove by induction (with the base being the leaves of the recursion) that with probability at least $1 - \delta$ over $\mathbf{U}$, it holds that $\mathcal{L}_{\mathcal{D}}(f_{(\mathbf{U},\mathbf{V})}) \le a \cdot (d + \ln(1/\delta))/|\mathbf{U}|$ for a universal constant $a > 0$. If we can complete this inductive step, then the conclusion follows by examining the root invocation *Sub-Sample*$(\mathbf{S}, \emptyset)$.

The base case in the inductive proof is simply when $a \cdot (d + \ln(1/\delta))/|\mathbf{U}| > 1$. Here the conclusion follows trivially as we always have $\mathcal{L}_{\mathcal{D}}(f_{(\mathbf{U},\mathbf{V})}) \le 1$. For the inductive step, let $f_{1,(\mathbf{U},\mathbf{V})}, f_{2,(\mathbf{U},\mathbf{V})}$ and $f_{3,(\mathbf{U},\mathbf{V})}$ denote the majority voters produced by the three recursive calls *Sub-Sample*$(\mathbf{U}_0, \mathbf{V} \cup \mathbf{U}_2 \cup \mathbf{U}_3)$, *Sub-Sample*$(\mathbf{U}_0, \mathbf{V} \cup \mathbf{U}_1 \cup \mathbf{U}_3)$ and *Sub-Sample*$(\mathbf{U}_0, \mathbf{V} \cup \mathbf{U}_1 \cup \mathbf{U}_2)$. Each of these have

$\mathcal{L}_{\mathcal{D}}(f_{i,(\mathbf{U},\mathbf{V})}) \leq a\cdot(d+\ln(1/\delta))/|\mathbf{U}_i| = 4a\cdot(d+\ln(1/\delta))/|\mathbf{U}|$ by the induction hypothesis (except with some probability δ which we ignore here for simplicity). For short, we say that a hypothesis h errs on x if $h(x) \neq c(x)$. The crux of the argument is now to show that it is very unlikely that $f_{i,(\mathbf{U},\mathbf{V})}$ errs on a sample $\mathbf{x} \sim \mathcal{D}$ at the same time as a hypothesis $h_{\mathbf{S}'}$ trained on a sub-sample $\mathbf{S}'$ produced by a recursive call $j \neq i$ also errs on $\mathbf{x}$.

For this step, let us wlog. consider the majority vote $f_{1,(\mathbf{U},\mathbf{V})}$ (over hypotheses produced by ERM on *Sub-Sample*$(\mathbf{U}_0, \mathbf{V} \cup \mathbf{U}_2 \cup \mathbf{U}_3)$). Intuitively, if $\mathcal{L}_{\mathcal{D}}(f_{1,(\mathbf{U},\mathbf{V})}) \ll a\cdot(d+\ln(1/\delta))/|\mathbf{U}|$ then the hypotheses in $f_{1,(\mathbf{U},\mathbf{V})}$ contribute little to $\mathcal{L}_{\mathcal{D}}(f_{(\mathbf{U},\mathbf{V})})$. So assume instead $\mathcal{L}_{\mathcal{D}}(f_{1,(\mathbf{U},\mathbf{V})}) \approx 4a\cdot(d+\ln(1/\delta))/|\mathbf{U}|$. Consider now some hypothesis $h_{\mathbf{S}'}$ obtained by running ERM on a sub-sample $\mathbf{S}'$ produced by the recursive call *Sub-Sample*$(\mathbf{U}_0, \mathbf{V} \cup \mathbf{U}_1 \cup \mathbf{U}_3)$. The key observation and property of the *Sub-Sample* algorithm, is that *all* hypotheses in the majority vote $f_{1,(\mathbf{U},\mathbf{V})}$ have been trained on sub-samples that *exclude* all of $\mathbf{U}_1$. This means that the samples $\mathbf{U}_1$ are independent of $f_{1,(\mathbf{U},\mathbf{V})}$. When $\mathcal{L}_{\mathcal{D}}(f_{1,(\mathbf{U},\mathbf{V})}) \approx 4a\cdot(d+\ln(1/\delta))/|\mathbf{U}|$, we will now see about $|\mathbf{U}_1|4a\cdot(d+\ln(1/\delta))/|\mathbf{U}| = a\cdot(d+\ln(1/\delta))$ samples $(\mathbf{x}, c(\mathbf{x}))$ in $\mathbf{U}_1$ for which $f_{1,(\mathbf{U},\mathbf{V})}(\mathbf{x}) \neq c(\mathbf{x})$. The observation is that, conditioned on $f_{1,(\mathbf{U},\mathbf{V})}$, these samples are i.i.d. from the conditional distribution $\mathcal{D}(\cdot \mid f_{1,(\mathbf{U},\mathbf{V})} \text{ errs})$. The second key observation is that $h_{\mathbf{S}'}$ is obtained by ERM on a sub-sample $\mathbf{S}'$ that *includes* all of $\mathbf{U}_1$ (we add $\mathbf{U}_1$ to $\mathbf{V}$ in both of the other recursive calls). Moreover, since we are in the realizable setting, we have $h_{\mathbf{S}'}(\mathbf{x}) = c(\mathbf{x})$ for every $\mathbf{x} \in \mathbf{U}_1$. In particular, this holds for all the samples where $f_{1,(\mathbf{U},\mathbf{V})}(\mathbf{x}) \neq c(\mathbf{x})$. The classic sample complexity bounds for proper PAC learning in the realizable setting then implies that $h_{\mathbf{S}'}$ has $\mathcal{L}_{\mathcal{D}(\cdot|f_{1,(\mathbf{U},\mathbf{V})} \text{ errs})}(h_{\mathbf{S}'}) = O((d + \ln(1/\delta))/(a\cdot(d+\ln(1/\delta)))) \leq 1/200$ for a sufficiently large. Note that this is under the conditional distribution $\mathcal{D}(\cdot \mid f_{1,(\mathbf{U},\mathbf{V})} \text{ errs})$. That is, $h_{\mathbf{S}'}$ rarely errs when $f_{1,(\mathbf{U},\mathbf{V})}$ errs. We thus get that $\Pr[f_{1,(\mathbf{U},\mathbf{V})}(\mathbf{x}) \neq c(\mathbf{x}) \wedge h_{\mathbf{S}'}(\mathbf{x}) \neq c(\mathbf{x})] = \Pr[f_{1,(\mathbf{U},\mathbf{V})}(\mathbf{x}) \neq c(\mathbf{x})] \cdot \Pr[h_{\mathbf{S}'}(\mathbf{x}) \neq c(\mathbf{x}) \mid f_{1,(\mathbf{U},\mathbf{V})}(\mathbf{x}) \neq c(\mathbf{x})] \leq (a/50)\cdot(d+\ln(1/\delta))/|\mathbf{U}|$. Since this holds for every $f_{i,(\mathbf{U},\mathbf{V})}$ and $h_{\mathbf{S}'}$ from a recursive call $j \neq i$, we can now argue that $\mathcal{L}_{\mathcal{D}}(f_{(\mathbf{U},\mathbf{V})}) \ll a\cdot(d+\ln(1/\delta))/|\mathbf{U}|$. To see this, note first that for $f_{(\mathbf{U},\mathbf{V})}$ to err on an $x \in \mathcal{X}$, it must be the case that at least one $f_{i,(\mathbf{U},\mathbf{V})}$ also errs. Even in this case, since one recursive call only contributes a third of the hypotheses in the majority vote $f_{(\mathbf{U},\mathbf{V})}$, there must be many hypotheses $h_{\mathbf{S}'}$ (at least a $1/2 - 1/3 = 1/6$ fraction of all hypotheses) trained from sub-samples $\mathbf{S}'$ produced by the recursive calls $j \neq i$ that also err on x. But we have just argued that it is very unlikely that both $f_{i,(\mathbf{U},\mathbf{V})}$ and such an $h_{\mathbf{S}'}$ err at the same time. Formalizing this intuition completes the inductive proof.

Bagging Analysis. We now turn to presenting the key ideas in our proof of Theorem 1, i.e. that bagging is an optimal PAC learner. Along the way, we also discuss the issues we encounter towards establishing the result. Recall that in bagging with a training set $\mathbf{S} \sim \mathcal{D}_c^m$, we randomly sub-sample t bootstrap samples $\mathbf{S}_1, \dots, \mathbf{S}_t \subset \mathbf{S}$ each consisting of n i.i.d. samples with replacement from $\mathbf{S}$. We then run ERM on each $\mathbf{S}_i$ to produce hypotheses $h_{\mathbf{S}_1}, \dots, h_{\mathbf{S}_t}$ and finally return the majority vote $f_{\mathbf{S}_1,\dots,\mathbf{S}_t}(x) = \operatorname{sign}(\sum_i h_{\mathbf{S}_i}(x))$. It will be convenient for us to think of $f_{\mathbf{S}_1,\dots,\mathbf{S}_t}$ in a slightly different way. Concretely, we instead let $f_{\mathbf{S}_1,\dots,\mathbf{S}_t}(x) = (1/t)\sum_i h_{\mathbf{S}_i}(x)$ be a *voting classifier*. Then $\operatorname{sign}(f_{\mathbf{S}_1,\dots,\mathbf{S}_t}(x)) \neq c(x)$ if and only if $f_{\mathbf{S}_1,\dots,\mathbf{S}_t}(x)c(x) \leq 0$. We thus seek to bound $\mathcal{L}_{\mathcal{D}}(f_{\mathbf{S}_1,\dots,\mathbf{S}_t}) = \Pr_{\mathbf{x}\sim\mathcal{D}}[f_{\mathbf{S}_1,\dots,\mathbf{S}_t}(\mathbf{x})c(\mathbf{x}) \leq 0]$. The motivation for thinking about $f_{\mathbf{S}_1,\dots,\mathbf{S}_t}$ as a voting classifier, is that it allows us to re-use some of the ideas that appear in proving generalization bounds for AdaBoost [Freund and Schapire, 1997] and other voting classifiers.

We first observe that similarly to Hanneke's sub-sampling, if we look at just two hypotheses $h_{\mathbf{S}_i}$ and $h_{\mathbf{S}_j}$ with $i \neq j$, then $h_{\mathbf{S}_i}$ is trained on a bootstrap sample $\mathbf{S}_i$ leaving out a rather large portion of $\mathbf{S}$. Furthermore, $h_{\mathbf{S}_j}$ will be trained on most of these left-out samples and thus one could again argue that it is unlikely that $h_{\mathbf{S}_i}$ and $h_{\mathbf{S}_j}$ err at the same time. Unfortunately, this line of argument fails when we start combining a non-constant number of hypotheses. In particular, with high probability over the bootstrap samples, the union of any set of ℓ bootstrap samples contains all but an $\exp(-\Omega(\ell))$-fraction of $\mathbf{S}$. This leaves very few samples that are independent of the hypotheses trained on such ℓ bootstrap samples. Trying to repeat Hanneke's argument unfortunately requires $\Omega(m)$ independent samples towards the last steps of an inductive proof. In a nutshell, what saves Hanneke's construction is that a third all sub-samples together still leave out a quarter of the training data. For bagging, such a property is just not true if we have more than a constant number of bootstrap samples.

Abandoning the hope of directly applying Hanneke's line of reasoning, we instead start by relating the performance of bagging to that of a particular voting classifier that is deterministically determined from a training set S, i.e. we get rid of the bootstrap samples. To formalize this, we first introduce some notation. From a training set S of m samples $(x_1, c(x_1)), \dots, (x_m, c(x_m))$ and a vector of n not necessarily distinct integers $I = (i_1, \dots, i_n) \in [m]^n$, let $S(I)$ denote the bootstrap sample $(x_{i_1}, c(x_{i_1})), \dots, (x_{i_n}, c(x_{i_n}))$. Then a random bootstrap sample $\mathbf{S}_i$ from S has the same distribution as if we draw $\mathbf{I}$ uniformly from $[m]^n$ and let $\mathbf{S}_i = S(\mathbf{I})$. Also, let $h_{S(I)} \in \mathbb{C}$ denote the hypothesis resulting from running ERM on $S(I)$. Finally, for a list of t vectors $B = (I_1, \dots, I_t) \in [m]^{n \times t}$, we let $f_{S,B} = (1/t)\sum_{i=1}^{t} h_{S(I_i)}$ denote the voting classifier produced by bagging with bootstrap samples $S(I_1), \dots, S(I_t)$. Using the notation $\mathbf{B} \sim [m]^{n\times t}$ to denote a uniform random $\mathbf{B}$ from $[m]^{n\times t}$ we thus have that the hypothesis produced by bagging on S has the same distribution as $f_{S,\mathbf{B}}$.

Now consider the following voting classifier

$$g_S(x) := \frac{1}{m^n} \sum_{I\in[m]^n} h_{S(I)}(x).$$

That is, g_S is the voting classifier averaging the predictions over all m^n possible bootstrap samples of S. Of course one would never compute g_S. Nonetheless, the performance of the random $f_{S,\mathbf{B}}$ with $\mathbf{B} \sim [m]^{n\times t}$ is closely related to that of g_S. To see this, we introduce the notion of *margins* which are typically used in the study of generalization of voting classifiers, see e.g. the works [Bartlett *et al.*, 1998; Gao and Zhou, 2013; Larsen and Ritzert, 2022]. For a sample $x \in \mathcal{X}$ and voting classifier $f(x) = (1/t)\sum_{i=1}^{t} h_i(x)$, we say that f has margin $f(x)c(x)$ on the sample x. Since each $h_i(x)$ is in $\{-1, 1\}$, we have the margin is a number between -1 and 1. Intuitively, 1 represents that all the hypotheses h_i agree on the label of x and those predictions are correct. In general, a margin of γ implies that an α-fraction of the hypotheses h_i are correct, where $\alpha - (1-\alpha) = \gamma \Rightarrow \alpha = 1/2 + \gamma/2$. For a margin $0 \leq \gamma \leq 1$, define $\mathcal{L}^{\gamma}_{\mathcal{D}}(f) = \Pr_{\mathbf{x}\sim\mathcal{D}}[f(\mathbf{x})c(\mathbf{x}) \leq \gamma]$. That is, $\mathcal{L}^{\gamma}_{\mathcal{D}}(f)$ is the probability over a random sample $\mathbf{x}$ from $\mathcal{D}$ that f has margin at most γ on $\mathbf{x}$. We have $\mathcal{L}_{\mathcal{D}}(f) = \mathcal{L}^{0}_{\mathcal{D}}(f)$. With margins defined, we show that for every training set $S = \{(x_i, c(x_i))\}_{i=1}^{m}$, if $t = \Omega(\ln(m/\delta))$, then with probability $1-\delta$ over $\mathbf{B} \sim [m]^{n\times t}$, we have

$$\mathcal{L}_{\mathcal{D}}(f_{S,\mathbf{B}}) \leq \mathcal{L}^{1/3}_{\mathcal{D}}(g_S) + 1/m. \tag{1}$$

What this gives us, is that it suffices to understand how often the voting classifier that averages over all possible bootstrap samples has margin at most $1/3$. To see why (1) is true, notice that every hypothesis $h_{S(\mathbf{I}_i)}$ in $f_{S,\mathbf{B}} = (1/t)\sum_{i=1}^{t} h_{S(\mathbf{I}_i)}$ is uniform random among the hypotheses averaged by g_S. Hence for any $x \in \mathcal{X}$ where g_S has margin more than $1/3$, we have $\mathbb{E}_{\mathbf{I}_i\sim[m]^n}[h_{S(\mathbf{I}_i)}(x)c(x)] > 1/3$. A Chernoff bound and independence of the bootstrap samples implies that $\Pr_{\mathbf{B}\sim[m]^{n\times t}}[f_{S,\mathbf{B}}(x)c(x) \leq 0] \leq \exp(-\Omega(t)) \leq \delta/m$. Using that this holds for every x with $g_S(x)c(x) > 1/3$ establishes (1).

Our next step is to show that $\mathcal{L}^{1/3}_{\mathcal{D}}(g_{\mathbf{S}})$ is small with high probability over $\mathbf{S} \sim \mathcal{D}^m_c$. For this, our key idea is to create groups of bootstrap samples $\mathbf{S}(I)$ with $I \in [m]^n$. These groups have a structure similar to those produced by Hanneke's *Sub-Sample* procedure. We remark that these groups are only for the sake of analysis and are not part of the bagging algorithm.

For simplicity, let us for now assume that bagging produced samples without replacement instead of with replacement. To indicate this, we slightly abuse notation and let $\binom{m}{n}$ denote all vectors $I \in [m]^n$ where all entries are distinct. Also, let us assume that n precisely equals the number of samples in each sub-sample created by Hanneke's *Sub-Sample* (technically, this is $n = m - \sum_{i=0}^{\lg_4 m-1} 4^i$). For a set $S = (x_1, c(x_1)), \ldots, (x_m, c(x_m))$, let $\mathcal{I}$ denote the collection of all vectors $I \in \binom{m}{n}$ such that $S(I)$ is one of the sub-samples produced by *Sub-Sample*$(S, \emptyset)$. Note that $\mathcal{I}$ only depends on m, not on S itself. We now define *buckets* $\mathcal{C}_i$ of vectors $I \in \binom{m}{n}$ (i.e. of vectors corresponding to bootstrap samples). For every permutation π of the indices $1, \ldots, m$, we create a bucket $\mathcal{C}_\pi$. We add a vector $I = (i_1, \ldots, i_n) \in \binom{m}{n}$ to $\mathcal{C}_\pi$ if and only if $\pi(I) = (\pi(i_1), \ldots, \pi(i_n))$ is in $\mathcal{I}$.

With these buckets defined, we now make several crucial observations. First, for any bucket $\mathcal{C}_\pi$, if $\mathbf{S} \sim \mathcal{D}^m_c$, then the joint distribution of the bootstrap samples $\mathbf{S}(I)$ with $I \in \mathcal{C}_\pi$ is precisely the same as the joint distribution of the sub-samples produced by *Sub-Sample*$(\mathbf{S}, \emptyset)$. This holds since permuting the samples in $\mathbf{S}$ does not change their distribution. Hence for any bucket, Hanneke's analysis shows that the majority of hypotheses $h_{\mathbf{S}(I)}$ with $I \in \mathcal{C}_\pi$ rarely errs. More precisely, if we let $f_{\mathbf{S},\pi} = (1/|\mathcal{C}_\pi|)\sum_{I\in\mathcal{C}_\pi} h_{\mathbf{S}(I)}$ then $\mathcal{L}_{\mathcal{D}}(f_{\mathbf{S},\pi}) = O((d+\ln(1/\delta))/m)$ with probability $1-\delta$ over $\mathbf{S}$. Here we need something slightly stronger, namely that $\mathcal{L}^{5/6}_{\mathcal{D}}(f_{\mathbf{S},\pi}) = O((d+\ln(1/\delta))/m)$. Assume for now that this holds.

Next observe that if $x \in \mathcal{X}$ has $g_S(x)c(x) \leq 1/3$ for a training set S, then at least one third of the hypotheses $h_{S(I)}$ with $I \in \binom{m}{n}$ err on x, i.e. $h_{S(\mathbf{I})}$ for $\mathbf{I}$ uniform in $\binom{m}{n}$ errs on x with probability at least $1/3$ (here we assume that g_S averages over $h_{S(I)}$ with $I \in \binom{m}{n}$, i.e. sampling without replacement instead of with). Symmetry now implies that every $I \in \binom{m}{n}$ is included in equally many buckets and all buckets contain equally many vectors I. This observation implies that the uniform $\mathbf{I} \sim \binom{m}{n}$ has the same distribution as if we first sample a uniform random bucket $\mathbf{C}$ and then sample a uniform random $\mathbf{I}$ from $\mathbf{C}$. But then if $h_{S(\mathbf{I})}$ errs on x with probability at least $1/3$, it must be the case that a constant fraction of the buckets $\mathcal{C}_\pi$ have $f_{S,\pi}(x)c(x) \leq 5/6$. This intuitively gives us that with high probability over $\mathbf{S}$, $\mathcal{L}^{1/3}_{\mathcal{D}}(g_{\mathbf{S}})$ can only be a constant factor larger than $\mathcal{L}^{5/6}_{\mathcal{D}}(f_{\mathbf{S},\pi})$. But $\mathcal{L}^{5/6}_{\mathcal{D}}(f_{\mathbf{S},\pi})$ is $O((d+\ln(1/\delta))/m)$ with high probability, establishing the same thing for $\mathcal{L}^{1/3}_{\mathcal{D}}(g_{\mathbf{S}})$ as desired.

The above are the main ideas in the proof, though making the steps completely formal requires some care, see the full version [Larsen, 2023] for details.

3 Conclusion

In this work, we have shown that the classic bagging heuristic is an optimal PAC learner in the realizable setting if we sample just a logarithmic number of bootstrap samples.

Let us remark a small downside of bagging compared to Hanneke's algorithm. In his work, the number of sub-samples is independent of δ, whereas our result for bagging requires $\Omega(\ln(m/\delta))$ sub-samples. Indeed, as bagging is a randomized algorithm, there will be some non-zero contribution to the failure probability due to the number of sub-samples (with some non-zero probability, all sub-samples are the same). We find it an interesting open problem whether the analysis can be tightened to yield a better dependency on δ in the number of sub-samples needed.

Finally, let us comment on the unspecified constant $a > 0$ in Theorem 1. In our proof, we did not attempt to minimize a and indeed it is rather ridiculous. With some care, one could certainly shave several orders of magnitude, but at the end of it, we still rely on Hanneke's proof which also does not have small constants. We believe that bagging actually provides quite good constants, but this would require a different proof strategy. At least our reduction to understanding $\mathcal{L}^{1/3}_{\mathcal{D}}(g_S)$ is very tight and incurs almost no loss in the constant a. A starting point for improvements would be to find an alternative proof that $\mathcal{L}^{1/3}_{\mathcal{D}}(g_{\mathbf{S}})$ is small with high probability over $\mathbf{S}$.

Acknowledgments

Supported by Independent Research Fund Denmark (DFF) Sapere Aude Research Leader grant No 9064-00068B.

References

[Auer and Ortner, 2004] Peter Auer and Ronald Ortner. A new pac bound for intersection-closed concept classes. In *Learning Theory*, pages 408–414. Springer Berlin Heidelberg, 2004.

[Bartlett *et al.*, 1998] 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.

[Blumer *et al.*, 1989] Anselm Blumer, Andrzej Ehrenfeucht, David Haussler, and Manfred K Warmuth. Learnability and the vapnik-chervonenkis dimension. *Journal of the ACM (JACM)*, 36(4):929–965, 1989.

[Breiman, 1996] Leo Breiman. Bagging predictors. *Machine Learning*, 24(2):123–140, 1996.

[Breiman, 2001] Leo Breiman. Random forests. *Machine Learning*, 45(1):5–32, 2001.

[Ehrenfeucht *et al.*, 1989] Andrzej Ehrenfeucht, David Haussler, Michael Kearns, and Leslie Valiant. A general lower bound on the number of examples needed for learning. *Information and Computation*, 82(3):247–261, 1989.

[Freund and Schapire, 1997] 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.

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

[Hanneke, 2016] Steve Hanneke. The optimal sample complexity of pac learning. *The Journal of Machine Learning Research*, 17(1):1319–1333, 2016.

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

[Larsen, 2023] Kasper Green Larsen. Bagging is an optimal PAC learner. In *COLT*, volume 195 of *Proceedings of Machine Learning Research*, pages 450–468. PMLR, 2023.

[Simon, 2015] Hans U Simon. An almost optimal pac algorithm. In *Conference on Learning Theory*, pages 1552–1563. PMLR, 2015.

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

[Vapnik and Chervonenkis, 1971] 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.

[Vapnik, 1982] Vladimir Vapnik. *Estimation of Dependences Based on Empirical Data: Springer Series in Statistics (Springer Series in Statistics)*. Springer-Verlag, Berlin, Heidelberg, 1982.