The Optimal Sample Complexity of PAC Learning

Steve Hanneke STEVE.HANNEKE@GMAIL.COM

Abstract

This work establishes a new upper bound on the number of samples sufficient for PAC learning in the realizable case. The bound matches known lower bounds up to numerical constant factors. This solves a long-standing open problem on the sample complexity of PAC learning. The technique and analysis build on a recent breakthrough by Hans Simon.

1. Introduction

Probably approximately correct learning (or *PAC* learning; Valiant, 1984) is a classic criterion for supervised learning, which has been the focus of much research in the past three decades. The objective in PAC learning is to produce a classifier that, with probability at least $1-\delta$, has error rate at most ε. To qualify as a PAC learning algorithm, it must satisfy this guarantee for all possible target concepts in a given family, under all possible data distributions. To achieve this objective, the learning algorithm is supplied with a number m of i.i.d. training samples (data points), along with the corresponding correct classifications. One of the central questions in the study of PAC learning is determining the minimum number $\mathcal{M}(\varepsilon,\delta)$ of training samples necessary and sufficient such that there exists a PAC learning algorithm requiring at most $\mathcal{M}(\varepsilon,\delta)$ samples (for any given ε and δ). This quantity $\mathcal{M}(\varepsilon,\delta)$ is known as the *sample complexity*.

Determining the sample complexity of PAC learning is a long-standing open problem. There have been upper and lower bounds established for decades, but they differ by a logarithmic factor. It has been widely believed that this logarithmic factor can be removed for certain well-designed learning algorithms, and attempting to prove this has been the subject of much effort. Simon (2015) has very recently made an enormous leap forward toward resolving this issue. That work proposed an algorithm that classifies points based on a majority vote among classifiers trained on independent data sets. Simon proves that this algorithm achieves a sample complexity that reduces the logarithmic factor in the upper bound down to a very slowly-growing function. However, that work does not quite completely resolve the gap, so that determining the optimal sample complexity remains open.

The present work resolves this problem by completely eliminating the logarithmic factor. The algorithm achieving this new bound is also based on a majority vote of classifiers. However, unlike Simon's algorithm, here the voting classifiers are trained on data subsets specified by a recursive algorithm, with substantial overlaps among the data subsets the classifiers are trained on.

2. Notation

We begin by introducing some basic notation essential to the discussion. Fix a nonempty set $\mathcal{X}$, called the *instance space*; we suppose $\mathcal{X}$ is equipped with a σ-algebra, defining the measurable subsets of $\mathcal{X}$. Also denote $\mathcal{Y} = \{-1, +1\}$, called the *label space*. A *classifier* is any measurable function $h : \mathcal{X} \to \mathcal{Y}$. Fix a nonempty set $\mathbb{C}$ of classifiers, called the *concept space*. To focus the discussion on nontrivial cases,[1] we suppose $|\mathbb{C}| \geq 3$; other than this, the results in this article will be valid for *any* choice of $\mathbb{C}$.

In the learning problem, there is a probability measure $\mathcal{P}$ over $\mathcal{X}$, called the *data distribution*, and a sequence $X_1(\mathcal{P}), X_2(\mathcal{P}), \ldots$ of independent $\mathcal{P}$-distributed random variables, called the *unlabeled data*; for $m \in \mathbb{N}$, also define $\mathbb{X}_{1:m}(\mathcal{P}) = \{X_1(\mathcal{P}), \ldots, X_m(\mathcal{P})\}$, and for completeness denote $\mathbb{X}_{1:0}(\mathcal{P}) = \{\}$. There is also a special element of $\mathbb{C}$, denoted $f^\star$, called the *target function*. For any sequence $S_x = \{x_1, \ldots, x_k\}$ in $\mathcal{X}$, denote by $(S_x, f^\star(S_x)) = \{(x_1, f^\star(x_1)), \ldots, (x_k, f^\star(x_k))\}$. For any probability measure P over $\mathcal{X}$, and any classifier h, denote by $\mathrm{er}_P(h; f^\star) = P(x : h(x) \neq f^\star(x))$. A *learning algorithm* $\mathcal{A}$ is a map,[2] mapping any sequence $\{(x_1, y_1), \ldots, (x_m, y_m)\}$ in $\mathcal{X} \times \mathcal{Y}$ (called a *data set*), of any length $m \in \mathbb{N} \cup \{0\}$, to a classifier $h : \mathcal{X} \to \mathcal{Y}$ (not necessarily in $\mathbb{C}$).

Definition 1 *For any $\varepsilon, \delta \in (0, 1)$, the* sample complexity of (ε, δ)-PAC learning*, denoted $\mathcal{M}(\varepsilon, \delta)$, is defined as the smallest $m \in \mathbb{N} \cup \{0\}$ for which there exists a learning algorithm $\mathcal{A}$ such that, for every possible data distribution $\mathcal{P}$, $\forall f^\star \in \mathbb{C}$, denoting $\hat{h} = \mathcal{A}((\mathbb{X}_{1:m}(\mathcal{P}), f^\star(\mathbb{X}_{1:m}(\mathcal{P}))))$,*

$$\mathbb{P}\left(\mathrm{er}_{\mathcal{P}}\left(\hat{h}; f^\star\right) \leq \varepsilon\right) \geq 1 - \delta.$$

If no such m exists, define $\mathcal{M}(\varepsilon, \delta) = \infty$.

The sample complexity is our primary object of study in this work. We require a few additional definitions before proceeding. Throughout, we use a natural extension of set notation to sequences: for any finite sequences $\{a_i\}_{i=1}^k, \{b_i\}_{i=1}^{k'}$, we denote by $\{a_i\}_{i=1}^k \cup \{b_i\}_{i=1}^{k'}$ the concatenated sequence $\{a_1, \ldots, a_k, b_1, \ldots, b_{k'}\}$. For any set A, we denote by $\{a_i\}_{i=1}^k \cap A$ the subsequence comprised of all a_i for which $a_i \in A$. Additionally, we write $b \in \{b_i\}_{i=1}^{k'}$ to indicate $\exists i \leq k'$ s.t. $b_i = b$, and we write $\{a_i\}_{i=1}^k \subseteq \{b_i\}_{i=1}^{k'}$ or $\{b_i\}_{i=1}^{k'} \supseteq \{a_i\}_{i=1}^k$ to express that $a_j \in \{b_i\}_{i=1}^{k'}$ for every $j \leq k$. We also denote $|\{a_i\}_{i=1}^k| = k$ (the length of the sequence). For any $k \in \mathbb{N} \cup \{0\}$ and any sequence $S = \{(x_1, y_1), \ldots, (x_k, y_k)\}$ of points in $\mathcal{X} \times \mathcal{Y}$, denote $\mathbb{C}[S] = \{h \in \mathbb{C} : \forall (x, y) \in S, h(x) = y\}$, referred to as the set of classifiers *consistent* with S.

Following Vapnik and Chervonenkis (1971), we say a sequence $\{x_1, \ldots, x_k\}$ of points in $\mathcal{X}$ is *shattered* by $\mathbb{C}$ if $\forall y_1, \ldots, y_k \in \mathcal{Y}$, $\exists h \in \mathbb{C}$ such that $\forall i \in \{1, \ldots, k\}$, $h(x_i) = y_i$: that is, there are 2^k distinct classifications of $\{x_1, \ldots, x_k\}$ realized by classifiers in $\mathbb{C}$. The Vapnik-Chervonenkis dimension (or *VC dimension*) of $\mathbb{C}$ is then defined as the largest integer k for which there exists a sequence $\{x_1, \ldots, x_k\}$ in $\mathcal{X}$ shattered by $\mathbb{C}$; if no such largest k exists, the VC dimension is said to be infinite. We denote by d the VC dimension of $\mathbb{C}$. This quantity is of fundamental importance in characterizing the sample complexity

1. The sample complexities for $|\mathbb{C}| = 1$ and $|\mathbb{C}| = 2$ are already quite well understood in the literature, the former having sample complexity 0, and the latter having sample complexity either 1 or $\Theta(\frac{1}{\varepsilon} \ln \frac{1}{\delta})$ (depending on whether the two classifiers are exact complements or not).
2. We also admit randomized algorithms $\mathcal{A}$, where the "internal randomness" of $\mathcal{A}$ is assumed to be independent of the data. Formally, there is a random variable R independent of $\{X_i(P)\}_{i,P}$ such that the value $\mathcal{A}(S)$ is determined by the input data S and the value of R.

of PAC learning. In particular, it is well known that the sample complexity is finite for any $\varepsilon, \delta \in (0,1)$ if and only if $d < \infty$ (Vapnik, 1982; Blumer, Ehrenfeucht, Haussler, and Warmuth, 1989; Ehrenfeucht, Haussler, Kearns, and Valiant, 1989). For simplicity of notation, for the remainder of this article we suppose $d < \infty$; furthermore, note that our assumption of $|\mathbb{C}| \geq 3$ implies $d \geq 1$.

We adopt a common variation on big-O asymptotic notation, used in much of the learning theory literature. Specifically, for functions $f, g : (0,1)^2 \to [0,\infty)$, we let $f(\varepsilon,\delta) = O(g(\varepsilon,\delta))$ denote the assertion that $\exists \varepsilon_0, \delta_0 \in (0,1)$ and $c_0 \in (0,\infty)$ such that, $\forall \varepsilon \in (0,\varepsilon_0)$, $\forall \delta \in (0,\delta_0)$, $f(\varepsilon,\delta) \leq c_0 g(\varepsilon,\delta)$; however, we also require that the values $\varepsilon_0, \delta_0, c_0$ in this definition be *numerical constants*, meaning that they are *independent of* $\mathbb{C}$ *and* $\mathcal{X}$. For instance, this means c_0 cannot depend on d. We equivalently write $f(\varepsilon,\delta) = \Omega(g(\varepsilon,\delta))$ to assert that $g(\varepsilon,\delta) = O(f(\varepsilon,\delta))$. Finally, we write $f(\varepsilon,\delta) = \Theta(g(\varepsilon,\delta))$ to assert that both $f(\varepsilon,\delta) = O(g(\varepsilon,\delta))$ and $f(\varepsilon,\delta) = \Omega(g(\varepsilon,\delta))$ hold. We also sometimes write $O(g(\varepsilon,\delta))$ in an expression, as a place-holder for some function $f(\varepsilon,\delta)$ satisfying $f(\varepsilon,\delta) = O(g(\varepsilon,\delta))$: for instance, the statement $N(\varepsilon,\delta) \leq d + O(\log(1/\delta))$ expresses that $\exists f(\varepsilon,\delta) = O(\log(1/\delta))$ for which $N(\varepsilon,\delta) \leq d + f(\varepsilon,\delta)$. Also, for any value $z \geq 0$, define $\mathrm{Log}(z) = \ln(\max\{z, e\})$ and similarly $\mathrm{Log}_2(z) = \log_2(\max\{z, 2\})$.

As is commonly required in the learning theory literature, we adopt the assumption that the events appearing in probability claims below are indeed measurable. For our purposes, this comes into effect only in the application of classic generalization bounds for sample-consistent classifiers (Lemma 4 below). See Blumer, Ehrenfeucht, Haussler, and Warmuth (1989) and van der Vaart and Wellner (1996) for discussion of conditions on $\mathbb{C}$ sufficient for this measurability assumption to hold.

3. Background

Our objective in this work is to establish *sharp* sample complexity bounds. As such, we should first review the known *lower bounds* on $\mathcal{M}(\varepsilon,\delta)$. A basic lower bound of $\frac{1-\varepsilon}{\varepsilon}\ln\left(\frac{1}{\delta}\right)$ was established by Blumer, Ehrenfeucht, Haussler, and Warmuth (1989) for $0 < \varepsilon < 1/2$ and $0 < \delta < 1$. A second lower bound of $\frac{d-1}{32\varepsilon}$ was supplied by Ehrenfeucht, Haussler, Kearns, and Valiant (1989), for $0 < \varepsilon \leq 1/8$ and $0 < \delta \leq 1/100$. Taken together, these results imply that, for any $\varepsilon \in (0, 1/8]$ and $\delta \in (0, 1/100]$,

$$\mathcal{M}(\varepsilon,\delta) \geq \max\left\{\frac{d-1}{32\varepsilon}, \frac{1-\varepsilon}{\varepsilon}\ln\left(\frac{1}{\delta}\right)\right\} = \Omega\left(\frac{1}{\varepsilon}\left(d + \mathrm{Log}\left(\frac{1}{\delta}\right)\right)\right). \tag{1}$$

This lower bound is complemented by classic *upper bounds* on the sample complexity. In particular, Vapnik (1982) and Blumer, Ehrenfeucht, Haussler, and Warmuth (1989) established an upper bound of

$$\mathcal{M}(\varepsilon,\delta) = O\left(\frac{1}{\varepsilon}\left(d\mathrm{Log}\left(\frac{1}{\varepsilon}\right) + \mathrm{Log}\left(\frac{1}{\delta}\right)\right)\right). \tag{2}$$

They proved that this sample complexity bound is in fact achieved by any algorithm that returns a classifier $h \in \mathbb{C}[(\mathbb{X}_{1:m}(\mathcal{P}), f^\star(\mathbb{X}_{1:m}(\mathcal{P})))]$, also known as a *sample-consistent learning algorithm* (or *empirical risk minimization* algorithm). A sometimes-better upper bound was established by Haussler, Littlestone, and Warmuth (1994):

$$\mathcal{M}(\varepsilon, \delta) = O\left(\frac{d}{\varepsilon}\mathrm{Log}\left(\frac{1}{\delta}\right)\right). \tag{3}$$

This bound is achieved by a modified variant of the *one-inclusion graph prediction algorithm*, a learning algorithm also proposed by Haussler, Littlestone, and Warmuth (1994), which has been conjectured to achieve the optimal sample complexity (Warmuth, 2004).

In very recent work, Simon (2015) produced a breakthrough insight. Specifically, by analyzing a learning algorithm based on a simple majority vote among classifiers consistent with distinct subsets of the training data, Simon (2015) established that, for any $K \in \mathbb{N}$,

$$\mathcal{M}(\varepsilon, \delta) = O\left(\frac{2^{2K}\sqrt{K}}{\varepsilon}\left(d \log^{(K)}\left(\frac{1}{\varepsilon}\right) + K + \mathrm{Log}\left(\frac{1}{\delta}\right)\right)\right), \tag{4}$$

where $\log^{(K)}(x)$ is the K-times iterated logarithm: $\log^{(0)}(x) = \max\{x, 1\}$ and $\log^{(K)}(x) = \max\{\log_2(\log^{(K-1)}(x)), 1\}$. In particular, one natural choice would be $K \approx \log^*\left(\frac{1}{\varepsilon}\right)$,[3] which (one can show) optimizes the asymptotic dependence on ε in the bound, yielding

$$\mathcal{M}(\varepsilon, \delta) = O\left(\frac{1}{\varepsilon}2^{O(\log^*(1/\varepsilon))}\left(d + \mathrm{Log}\left(\frac{1}{\delta}\right)\right)\right).$$

In general, the entire form of the bound (4) is optimized (up to numerical constant factors) by choosing $K = \max\left\{\log^*\left(\frac{1}{\varepsilon}\right) - \log^*\left(\frac{1}{d}\mathrm{Log}\left(\frac{1}{\delta}\right)\right) + 1, 1\right\}$. Note that, with either of these choices of K, there is a range of ε, δ, and d values for which the bound (4) is strictly smaller than both (2) and (3): for instance, for small ε, it suffices to have $\mathrm{Log}(1/\delta) \ll d\mathrm{Log}(1/\varepsilon)/(2^{2\log^*(1/\varepsilon)}\sqrt{\log^*(1/\varepsilon)})$ while $2^{2\log^*(1/\varepsilon)}\sqrt{\log^*(1/\varepsilon)} \ll \min\{\mathrm{Log}(1/\delta), d\}$. However, this bound still does not quite match the form of the lower bound (1).

There have also been many special-case analyses, studying restricted types of concept spaces $\mathbb{C}$ for which the above gaps can be closed (e.g., Auer and Ortner, 2007; Darnstädt, 2015; Hanneke, 2015). However, these special conditions do not include many of the most-commonly studied concept spaces, such as linear separators and multilayer neural networks. There have also been a variety of studies that, in addition to restricting to specific concept spaces $\mathbb{C}$, also introduce strong restrictions on the data distribution $\mathcal{P}$, and establish an upper bound of the same form as the lower bound (1) under these restrictions (e.g., Long, 2003; Giné and Koltchinskii, 2006; Bshouty, Li, and Long, 2009; Hanneke, 2009, 2015; Balcan and Long, 2013). However, there are many interesting classes $\mathbb{C}$ and distributions $\mathcal{P}$ for which these results do not imply any improvements over (2). Thus, in the present literature, there persists a gap between the lower bound (1) and the minimum of all of the known upper bounds (2), (3), and (4) applicable to the *general* case of an arbitrary concept space of a given VC dimension d (under arbitrary data distributions).

In the present work, we establish a new upper bound for a novel learning algorithm, which holds for *any* concept space $\mathbb{C}$, and which improves over all of the above general upper bounds in its joint dependence on ε, δ, and d. In particular, it is *optimal*, in the

3. The function $\log^*(x)$ is the iterated logarithm: the smallest $K \in \mathbb{N} \cup \{0\}$ for which $\log^{(K)}(x) \leq 1$. It is an extremely slowly growing function of x.

sense that it matches the lower bound (1) up to numerical constant factors. This work thus solves a long-standing open problem, by determining the precise form of the optimal sample complexity, up to numerical constant factors.

4. Main Result

This section presents the main contributions of this work: a novel learning algorithm, and a proof that it achieves the optimal sample complexity.

4.1 Sketch of the Approach

The general approach used here builds on an argument of Simon (2015), which itself has roots in the analysis of sample-consistent learning algorithms by Hanneke (2009, Section 2.9.1). The essential idea from Simon (2015) is that, if we have two classifiers, $\hat{h}$ and $\hat{g}$, the latter of which is an element of $\mathbb{C}$ consistent with an i.i.d. data set $\tilde{S}$ *independent* from $\hat{h}$, then we can analyze the probability that they *both* make a mistake on a random point by bounding the error rate of $\hat{h}$ under the distribution $\mathcal{P}$, and bounding the error rate of $\hat{g}$ under the *conditional* distribution given that $\hat{h}$ makes a mistake. In particular, it will either be the case that $\hat{h}$ itself has small error rate, or else (if $\hat{h}$ has error rate larger than our desired bound) with high probability, the number of points in $\tilde{S}$ contained in the error region of $\hat{h}$ will be at least some number $\propto \mathrm{er}_{\mathcal{P}}(\hat{h}; f^\star)|\tilde{S}|$; in the latter case, we can bound the conditional error rate of $\hat{g}$ in terms of the number of such points via a classic generalization bound for sample-consistent classifiers (Lemma 4 below). Multiplying this bound on the conditional error rate of $\hat{g}$ by the error rate of $\hat{h}$ results in a bound on the probability they both make a mistake. More specifically, this argument yields a bound of the following form: for an appropriate numerical constant $\tilde{c} \in (0, \infty)$, with probability at least $1-\delta$, $\forall \hat{g} \in \mathbb{C}[\tilde{S}]$,

$$\mathcal{P}\left(x : \hat{h}(x) = \hat{g}(x) \neq f^\star(x)\right) \leq \frac{\tilde{c}}{|\tilde{S}|}\left(d\mathrm{Log}\left(\frac{\mathrm{er}_{\mathcal{P}}(\hat{h}; f^\star)|\tilde{S}|}{d}\right) + \mathrm{Log}\left(\frac{1}{\delta}\right)\right).$$

The original analysis of Simon (2015) applied this reasoning repeatedly, in an inductive argument, thereby bounding the probability that K classifiers, each consistent with one of K independent training sets, all make a mistake on a random point. He then reasoned that the error rate of the majority vote of $2K-1$ such classifiers can be bounded by the sum of these bounds for all subsets of K of these classifiers, since the majority vote classifier agrees with at least K of the constituent classifiers.

In the present work, we also consider a simple majority vote of a number of classifiers, but we alter the way the data is split up, allowing significant overlaps among the subsamples. In particular, each classifier is trained on considerably more data this way. We construct these subsamples recursively, motivated by an inductive analysis of the sample complexity. At each stage, we have a working set S of i.i.d. data points, and another sequence T of data points, referred to as the *partially-constructed subsample*. As a terminal case, if $|S|$ is smaller than a certain cutoff size, we generate a subsample $S \cup T$, on which we will train a classifier $\hat{g} \in \mathbb{C}[S \cup T]$. Otherwise (for the nonterminal case), we use (roughly) a constant fraction of the points in S to form a subsequence S_0, and make three recursive calls to the algorithm, using S_0 as the working set in each call. By an inductive hypothesis, for each of these three recursive calls, with probability $1-\delta'$, the majority vote of the classifiers trained on subsamples generated by that call has error rate at most $\frac{c}{|S_0|}(d+\mathrm{Log}(1/\delta'))$, for an appropriate numerical constant c. These three majority vote classifiers, denoted h_1, h_2, h_3, will each play the role of $\hat{h}$ in the argument above.

With the remaining constant fraction of data points in S (i.e., those not used to form S_0), we divide them into three independent subsequences S_1, S_2, S_3. Then for each of the three recursive calls, we provide as its partially-constructed subsample (i.e., the "T" argument) a sequence $S_i \cup S_j \cup T$ with $i \neq j$; specifically, for the k^{th} recursive call ($k \in \{1,2,3\}$), we take $\{i,j\} = \{1,2,3\} \setminus \{k\}$. Since the argument T is retained within the partially-constructed subsample passed to each recursive call, a simple inductive argument reveals that, for each $i \in \{1,2,3\}$, $\forall k \in \{1,2,3\} \setminus \{i\}$, all of the classifiers $\hat{g}$ trained on subsamples generated in the k^{th} recursive call are contained in $\mathbb{C}[S_i]$. Furthermore, since S_i is not included in the argument to the i^{th} recursive call, h_i and S_i are independent. Thus, by the argument discussed above, applied with $\hat{h} = h_i$ and $\tilde{S} = S_i$, we have that with probability at least $1-\delta'$, for any $\hat{g}$ trained on a subsample generated in recursive calls $k \in \{1,2,3\} \setminus \{i\}$, the probability that *both* h_i *and* $\hat{g}$ make a mistake on a random point is at most $\frac{\tilde{c}}{|S_i|}\left(d\mathrm{Log}\left(\frac{\mathrm{er}_{\mathcal{P}}(h_i;f^\star)|S_i|}{d}\right) + \mathrm{Log}\left(\frac{1}{\delta'}\right)\right)$. Composing this with the aforementioned inductive hypothesis, recalling that $|S_i| \propto |S|$ and $|S_0| \propto |S|$, and simplifying by a bit of calculus, this is at most $\frac{c'}{|S|}\left(d\mathrm{Log}(c) + \mathrm{Log}\left(\frac{1}{\delta'}\right)\right)$, for an appropriate numerical constant c'. By choosing $\delta' \propto \delta$ appropriately, the union bound implies that, with probability at least $1-\delta$, this holds for all choices of $i \in \{1,2,3\}$. Furthermore, by choosing c sufficiently large, this bound is at most $\frac{c}{12|S|}\left(d + \mathrm{Log}\left(\frac{1}{\delta}\right)\right)$.

To complete the inductive argument, we then note that on any point x, the majority vote of all of the classifiers (from all three recursive calls) must agree with at least one of the three classifiers h_i, and must agree with at least $1/4$ of the classifiers $\hat{g}$ trained on subsamples generated in recursive calls $k \in \{1,2,3\} \setminus \{i\}$. Therefore, on any point x for which the majority vote makes a mistake, with probability at least $1/12$, a uniform random choice of $i \in \{1,2,3\}$, and of $\hat{g}$ from recursive calls $k \in \{1,2,3\} \setminus \{i\}$, results in h_i and $\hat{g}$ that both make a mistake on x. Applying this fact to a *random* point $X \sim \mathcal{P}$ (and invoking Fubini's theorem), this implies that the error rate of the majority vote is at most 12 times the average (over choices of i and $\hat{g}$) of the probabilities that h_i and $\hat{g}$ both make a mistake on X. Combined with the above bound, this is at most $\frac{c}{|S|}\left(d + \mathrm{Log}\left(\frac{1}{\delta}\right)\right)$. The formal details are provided below.

4.2 Formal Details

For any $k \in \mathbb{N} \cup \{0\}$, and any $S \in (\mathcal{X} \times \mathcal{Y})^k$ with $\mathbb{C}[S] \neq \emptyset$, let $L(S)$ denote an arbitrary classifier h in $\mathbb{C}[S]$, entirely determined by S: that is, $L(\cdot)$ is a fixed sample-consistent learning algorithm (i.e., empirical risk minimizer). For any $k \in \mathbb{N}$ and sequence of data sets $\{S_1,\ldots,S_k\}$, denote $L(\{S_1,\ldots,S_k\}) = \{L(S_1),\ldots,L(S_k)\}$. Also, for any values $y_1,\ldots,y_k \in \mathcal{Y}$, define the majority function: $\mathrm{Majority}(y_1,\ldots,y_k) = \mathrm{sign}\left(\sum_{i=1}^k y_i\right) = 2\mathbb{1}\left[\sum_{i=1}^k y_i \geq 0\right] - 1$. We also overload this notation, defining the *majority classifier* $\mathrm{Majority}(h_1,\ldots,h_k)(x) = \mathrm{Majority}(h_1(x),\ldots,h_k(x))$, for any classifiers $h_1,\ldots,h_k$.

Now consider the following recursive algorithm, which takes as input two finite data sets, S and T, satisfying $\mathbb{C}[S \cup T] \neq \emptyset$, and returns a *finite sequence of data sets* (referred to as *subsamples* of $S \cup T$). The classifier used to achieve the new sample complexity bound below is simply the majority vote of the classifiers obtained by applying L to these subsamples.

Algorithm: $\mathbb{A}(S;T)$
0. If $\lvert S\rvert \leq 3$
1. Return $\{S \cup T\}$
2. Let S_0 denote the first $\lvert S\rvert - 3\lfloor \lvert S\rvert/4\rfloor$ elements of S, S_1 the next $\lfloor \lvert S\rvert/4\rfloor$ elements, S_2 the next $\lfloor \lvert S\rvert/4\rfloor$ elements, and S_3 the remaining $\lfloor \lvert S\rvert/4\rfloor$ elements after that
3. Return $\mathbb{A}(S_0; S_2 \cup S_3 \cup T) \cup \mathbb{A}(S_0; S_1 \cup S_3 \cup T) \cup \mathbb{A}(S_0; S_1 \cup S_2 \cup T)$

Theorem 2

$$\mathcal{M}(\varepsilon, \delta) = O\left(\frac{1}{\varepsilon}\left(d + \mathrm{Log}\left(\frac{1}{\delta}\right)\right)\right).$$

In particular, a sample complexity of the form expressed on the right hand side is achieved by the algorithm that returns the classifier Majority$(L(\mathbb{A}(S;\emptyset)))$*, given any data set S.*

Combined with (1), this immediately implies the following corollary.

Corollary 3

$$\mathcal{M}(\varepsilon, \delta) = \Theta\left(\frac{1}{\varepsilon}\left(d + \mathrm{Log}\left(\frac{1}{\delta}\right)\right)\right).$$

The algorithm $\mathbb{A}$ is expressed above as a recursive method for constructing a sequence of subsamples, as this form is most suitable for the arguments in the proof below. However, it should be noted that one can equivalently describe these constructed subsamples *directly*, as the selection of which data points should be included in which subsamples can be expressed as a simple function of the indices. To illustrate this, consider the simplest case in which $S = \{(x_0, y_0), \ldots, (x_{m-1}, y_{m-1})\}$ with $m = 4^{\ell}$ for some $\ell \in \mathbb{N}$: that is, $\lvert S\rvert$ is a power of 4. In this case, let $\{T_0, \ldots, T_{n-1}\}$ denote the sequence of labeled data sets returned by $\mathbb{A}(S;\emptyset)$, and note that since each recursive call reduces $\lvert S\rvert$ by a factor of 4 while making 3 recursive calls, we have $n = 3^{\ell}$. First, note that (x_0, y_0) is contained in *every* subsample T_i. For the rest, consider any $i \in \{1, \ldots, m-1\}$ and $j \in \{0, \ldots, n-1\}$, and let us express i in its base-4 representation as $i = \sum_{t=0}^{\ell-1} i_t 4^t$, where each $i_t \in \{0, 1, 2, 3\}$, and express j in its base-3 representation as $j = \sum_{t=0}^{\ell-1} j_t 3^t$, where each $j_t \in \{0, 1, 2\}$. Then it holds that $(x_i, y_i) \in T_j$ if and only if the largest $t \in \{0, \ldots, \ell-1\}$ with $i_t \neq 0$ satisfies $i_t - 1 \neq j_t$. This kind of direct description of the subsamples is also possible when $\lvert S\rvert$ is not a power of 4, though a bit more complicated to express.

4.3 Proof of Theorem 2

The following classic result will be needed in the proof. A bound of this type is implied by a theorem of Vapnik (1982); the version stated here features slightly smaller constant factors, obtained by Blumer, Ehrenfeucht, Haussler, and Warmuth (1989).[4]

4. Specifically, it follows by combining their Theorem A2.1 and Proposition A2.1, setting the resulting expression equal to δ and solving for ε.

Lemma 4 *For any $\delta \in (0,1)$, $m \in \mathbb{N}$, $f^\star \in \mathbb{C}$, and any probability measure P over $\mathcal{X}$, letting $Z_1, \ldots, Z_m$ be independent P-distributed random variables, with probability at least $1-\delta$, every $h \in \mathbb{C}[\{(Z_i, f^\star(Z_i))\}_{i=1}^m]$ satisfies*

$$\mathrm{er}_P(h; f^\star) \leq \frac{2}{m}\left(d\mathrm{Log}_2\left(\frac{2em}{d}\right) + \mathrm{Log}_2\left(\frac{2}{\delta}\right)\right).$$

We are now ready for the proof of Theorem 2.

Proof of Theorem 2 Fix any $f^\star \in \mathbb{C}$ and probability measure $\mathcal{P}$ over $\mathcal{X}$, and for brevity, denote $\mathbb{S}_{1:m} = (\mathbb{X}_{1:m}(\mathcal{P}), f^\star(\mathbb{X}_{1:m}(\mathcal{P})))$, for each $m \in \mathbb{N}$. Also, for any classifier h, define $\mathrm{ER}(h) = \{x \in \mathcal{X} : h(x) \neq f^\star(x)\}$.

We begin by noting that, for any finite sequences S and T of points in $\mathcal{X}\times\mathcal{Y}$, a straightforward inductive argument reveals that all of the subsamples $\hat{S}$ in the sequence returned by $\mathbb{A}(S;T)$ satisfy $\hat{S} \subseteq S \cup T$ (since no additional data points are ever introduced in any step). Thus, if $f^\star \in \mathbb{C}[S]$ and $f^\star \in \mathbb{C}[T]$, then $f^\star \in \mathbb{C}[S] \cap \mathbb{C}[T] = \mathbb{C}[S \cup T] \subseteq \mathbb{C}[\hat{S}]$, so that $\mathbb{C}[\hat{S}] \neq \emptyset$. In particular, this means that, in this case, each of these subsamples $\hat{S}$ is a valid input to $L(\cdot)$, and thus $L(\mathbb{A}(S;T))$ is a well-defined sequence of classifiers. Furthermore, since the recursive calls all have T as a subsequence of their second arguments, and the terminal case (i.e., Step 1) includes this second argument in the constructed subsample, another straightforward inductive argument implies that every subsample $\hat{S}$ returned by $\mathbb{A}(S;T)$ satisfies $\hat{S} \supseteq T$. Thus, in the case that $f^\star \in \mathbb{C}[S]$ and $f^\star \in \mathbb{C}[T]$, by definition of L, we also have that every classifier h in the sequence $L(\mathbb{A}(S;T))$ satisfies $h \in \mathbb{C}[T]$.

Fix a numerical constant $c = 1800$. We will prove by induction that, for any $m' \in \mathbb{N}$, for every $\delta' \in (0,1)$, and every finite sequence T' of points in $\mathcal{X}\times\mathcal{Y}$ with $f^\star \in \mathbb{C}[T']$, with probability at least $1-\delta'$, the classifier $\hat{h}_{m',T'} = \mathrm{Majority}\left(L\left(\mathbb{A}(\mathbb{S}_{1:m'};T')\right)\right)$ satisfies

$$\mathrm{er}_{\mathcal{P}}(\hat{h}_{m',T'}; f^\star) \leq \frac{c}{m'+1}\left(d + \ln\left(\frac{18}{\delta'}\right)\right). \tag{5}$$

First, as a base case, consider any $m' \in \mathbb{N}$ with $m' \leq c\ln(18e) - 1$. In this case, fix any $\delta' \in (0,1)$ and any sequence T' with $f^\star \in \mathbb{C}[T']$. Also note that $f^\star \in \mathbb{C}[\mathbb{S}_{1:m'}]$. Thus, as discussed above, $\hat{h}_{m',T'}$ is a well-defined classifier. We then trivially have

$$\mathrm{er}_{\mathcal{P}}(\hat{h}_{m',T'}; f^\star) \leq 1 \leq \frac{c}{m'+1}\left(1 + \ln(18)\right) < \frac{c}{m'+1}\left(d + \ln\left(\frac{18}{\delta'}\right)\right),$$

so that (5) holds.

Now take as an inductive hypothesis that, for some $m \in \mathbb{N}$ with $m > c\ln(18e) - 1$, for every $m' \in \mathbb{N}$ with $m' < m$, we have that for every $\delta' \in (0,1)$ and every finite sequence T' in $\mathcal{X}\times\mathcal{Y}$ with $f^\star \in \mathbb{C}[T']$, with probability at least $1-\delta'$, (5) is satisfied. To complete the inductive proof, we aim to establish that this remains the case with $m' = m$ as well. Fix any $\delta \in (0,1)$ and any finite sequence T of points in $\mathcal{X}\times\mathcal{Y}$ with $f^\star \in \mathbb{C}[T]$. Note that $c\ln(18e) - 1 \geq 3$, so that (since $|\mathbb{S}_{1:m}| = m > c\ln(18e) - 1$) we have $|\mathbb{S}_{1:m}| \geq 4$, and hence the execution of $\mathbb{A}(\mathbb{S}_{1:m};T)$ returns in Step 3 (not Step 1). Let S_0, S_1, S_2, S_3 be as in the definition of $\mathbb{A}(S;T)$, with $S = \mathbb{S}_{1:m}$. Also denote $T_1 = S_2 \cup S_3 \cup T$, $T_2 = S_1 \cup S_3 \cup T$, $T_3 = S_1 \cup S_2 \cup T$, and for each $i \in \{1,2,3\}$, denote $h_i = \mathrm{Majority}\left(L\left(\mathbb{A}(S_0;T_i)\right)\right)$, corresponding to the majority votes of classifiers trained on the subsamples from each of the three recursive calls in the algorithm.

Note that $S_0 = \mathbb{S}_{1:(m-3\lfloor m/4\rfloor)}$. Furthermore, since $m \geq 4$, we have $1 \leq m-3\lfloor m/4\rfloor < m$. Also note that $f^\star \in \mathbb{C}[S_i]$ for each $i \in \{1,2,3\}$, which, together with the fact that $f^\star \in \mathbb{C}[T]$, implies $f^\star \in \mathbb{C}[T] \cap \bigcap_{j\in\{1,2,3\}\setminus\{i\}} \mathbb{C}[S_j] = \mathbb{C}[T_i]$ for each $i \in \{1,2,3\}$. Thus, since $f^\star \in \mathbb{C}[S_0]$ as well, for each $i \in \{1,2,3\}$, $L\left(\mathbb{A}(S_0;T_i)\right)$ is a well-defined sequence of classifiers (as discussed above), so that h_i is also well-defined. In particular, note that $h_i = \hat{h}_{(m-3\lfloor m/4\rfloor),T_i}$. Therefore, by the inductive hypothesis (applied under the conditional distribution given S_1, S_2, S_3, which are independent of S_0), combined with the law of total probability, for each $i \in \{1,2,3\}$, there is an event E_i of probability at least $1-\delta/9$, on which

$$\mathcal{P}(\mathrm{ER}(h_i)) \leq \frac{c}{|S_0|+1}\left(d + \ln\left(\frac{9\cdot 18}{\delta}\right)\right) \leq \frac{4c}{m}\left(d + \ln\left(\frac{9\cdot 18}{\delta}\right)\right). \tag{6}$$

Next, fix any $i \in \{1,2,3\}$, and denote by $\{(Z_{i,1}, f^\star(Z_{i,1})), \ldots, (Z_{i,N_i}, f^\star(Z_{i,N_i}))\} = S_i \cap (\mathrm{ER}(h_i)\times\mathcal{Y})$, where $N_i = |S_i \cap (\mathrm{ER}(h_i)\times\mathcal{Y})|$: that is, $\{(Z_{i,t}, f^\star(Z_{i,t}))\}_{t=1}^{N_i}$ is the subsequence of elements (x,y) in S_i for which $x \in \mathrm{ER}(h_i)$. Note that, since h_i and S_i are independent, $Z_{i,1},\ldots,Z_{i,N_i}$ are conditionally independent given h_i and N_i, each with conditional distribution $\mathcal{P}(\cdot|\mathrm{ER}(h_i))$ (if $N_i > 0$). Thus, applying Lemma 4 under the conditional distribution given h_i and N_i, combined with the law of total probability, we have that on an event E_i' of probability at least $1-\delta/9$, if $N_i > 0$, then every $h \in \mathbb{C}\left[\{(Z_{i,t}, f^\star(Z_{i,t}))\}_{t=1}^{N_i}\right]$ satisfies

$$\mathrm{er}_{\mathcal{P}(\cdot|\mathrm{ER}(h_i))}(h; f^\star) \leq \frac{2}{N_i}\left(d\mathrm{Log}_2\left(\frac{2eN_i}{d}\right) + \mathrm{Log}_2\left(\frac{18}{\delta}\right)\right).$$

Furthermore, as discussed above, each $j \in \{1,2,3\}\setminus\{i\}$ and $h \in L\left(\mathbb{A}(S_0;T_j)\right)$ have $h \in \mathbb{C}[T_j]$, and $T_j \supseteq S_i \supseteq \{(Z_{i,t}, f^\star(Z_{i,t}))\}_{t=1}^{N_i}$, so that $\mathbb{C}[T_j] \subseteq \mathbb{C}\left[\{(Z_{i,t}, f^\star(Z_{i,t}))\}_{t=1}^{N_i}\right]$. It follows that every $h \in \bigcup_{j\in\{1,2,3\}\setminus\{i\}} L\left(\mathbb{A}(S_0;T_j)\right)$ has $h \in \mathbb{C}\left[\{(Z_{i,t}, f^\star(Z_{i,t}))\}_{t=1}^{N_i}\right]$. Thus, on the event E_i', if $N_i > 0$, $\forall h \in \bigcup_{j\in\{1,2,3\}\setminus\{i\}} L(\mathbb{A}(S_0;T_j))$,

$$\begin{aligned}&\mathcal{P}(\mathrm{ER}(h_i)\cap\mathrm{ER}(h)) = \mathcal{P}(\mathrm{ER}(h_i))\mathcal{P}(\mathrm{ER}(h)|\mathrm{ER}(h_i))\\ &= \mathcal{P}(\mathrm{ER}(h_i))\mathrm{er}_{\mathcal{P}(\cdot|\mathrm{ER}(h_i))}(h;f^\star) \leq \mathcal{P}(\mathrm{ER}(h_i))\frac{2}{N_i}\left(d\mathrm{Log}_2\left(\frac{2eN_i}{d}\right) + \mathrm{Log}_2\left(\frac{18}{\delta}\right)\right).\end{aligned} \tag{7}$$

Additionally, since h_i and S_i are independent, by a Chernoff bound (applied under the conditional distribution given h_i) and the law of total probability, there is an event E_i'' of probability at least $1-\delta/9$, on which, if $\mathcal{P}(\mathrm{ER}(h_i)) \geq \frac{23}{\lfloor m/4\rfloor}\ln\left(\frac{9}{\delta}\right) \geq \frac{2(10/3)^2}{\lfloor m/4\rfloor}\ln\left(\frac{9}{\delta}\right)$, then

$$N_i \geq (7/10)\mathcal{P}(\mathrm{ER}(h_i))|S_i| = (7/10)\mathcal{P}(\mathrm{ER}(h_i))\lfloor m/4\rfloor.$$

In particular, on E_i'', if $\mathcal{P}(\mathrm{ER}(h_i)) \geq \frac{23}{\lfloor m/4\rfloor}\ln\left(\frac{9}{\delta}\right)$, then the above inequality implies $N_i > 0$.

Combining this with (6) and (7), and noting that $\mathrm{Log}_2(x) \leq \mathrm{Log}(x)/\ln(2)$ and $x \mapsto \frac{1}{x}\mathrm{Log}(c'x)$ is nonincreasing on $(0,\infty)$ (for any fixed $c' > 0$), we have that on $E_i \cap E_i' \cap E_i''$, if $\mathcal{P}(\mathrm{ER}(h_i)) \geq \frac{23}{\lfloor m/4 \rfloor} \ln\left(\frac{9}{\delta}\right)$, then every $h \in \bigcup_{j \in \{1,2,3\} \setminus \{i\}} L(\mathbb{A}(S_0; T_j))$ has

$$\begin{aligned}
\mathcal{P}(\mathrm{ER}(h_i) \cap \mathrm{ER}(h)) &\leq \frac{20}{7\ln(2)\lfloor m/4 \rfloor}\left(d\mathrm{Log}\left(\frac{2e(7/10)\mathcal{P}(\mathrm{ER}(h_i))\lfloor m/4 \rfloor}{d}\right) + \mathrm{Log}\left(\frac{18}{\delta}\right)\right) \\
&\leq \frac{20}{7\ln(2)\lfloor m/4 \rfloor}\left(d\mathrm{Log}\left(\frac{(7/5)ec\left(d + \ln\left(\frac{9\cdot 18}{\delta}\right)\right)}{d}\right) + \mathrm{Log}\left(\frac{18}{\delta}\right)\right) \\
&\leq \frac{20}{7\ln(2)\lfloor m/4 \rfloor}\left(d\mathrm{Log}\left((2/5)c\left((7/2)e + \frac{7e}{d}\ln\left(\frac{18}{\delta}\right)\right)\right) + \mathrm{Log}\left(\frac{18}{\delta}\right)\right) \\
&\leq \frac{20}{7\ln(2)\lfloor m/4 \rfloor}\left(d\ln((9/5)ec) + 8\ln\left(\frac{18}{\delta}\right)\right), \qquad (8)
\end{aligned}$$

where this last inequality is due to Lemma 5 in Appendix A. Since $m > c\ln(18e) - 1 > 3200$, we have $\lfloor m/4 \rfloor > (m-4)/4 > \frac{799}{800}\frac{m}{4} > \frac{799}{800}\frac{3200}{3201}\frac{m+1}{4}$. Plugging this relaxation into the above bound, combined with numerical calculation of the logarithmic factor (with c as defined above), we find that the expression in (8) is less than

$$\frac{150}{m+1}\left(d + \ln\left(\frac{18}{\delta}\right)\right).$$

Additionally, if $\mathcal{P}(\mathrm{ER}(h_i)) < \frac{23}{\lfloor m/4 \rfloor} \ln\left(\frac{9}{\delta}\right)$, then monotonicity of measures implies

$$\mathcal{P}(\mathrm{ER}(h_i) \cap \mathrm{ER}(h)) \leq \mathcal{P}(\mathrm{ER}(h_i)) < \frac{23}{\lfloor m/4 \rfloor}\ln\left(\frac{9}{\delta}\right) < \frac{150}{m+1}\left(d + \ln\left(\frac{18}{\delta}\right)\right),$$

again using the above lower bound on $\lfloor m/4 \rfloor$ for this last inequality. Thus, regardless of the value of $\mathcal{P}(\mathrm{ER}(h_i))$, on the event $E_i \cap E_i' \cap E_i''$, we have $\forall h \in \bigcup_{j \in \{1,2,3\} \setminus \{i\}} L(\mathbb{A}(S_0; T_j))$,

$$\mathcal{P}(\mathrm{ER}(h_i) \cap \mathrm{ER}(h)) < \frac{150}{m+1}\left(d + \ln\left(\frac{18}{\delta}\right)\right).$$

Now denote $h_{\mathrm{maj}} = \hat{h}_{m,T} = \mathrm{Majority}\left(L(\mathbb{A}(S; T))\right)$, again with $S = \mathbb{S}_{1:m}$. By definition of Majority$(\cdot)$, for any $x \in \mathcal{X}$, at least $1/2$ of the classifiers h in the sequence $L(\mathbb{A}(S;T))$ have $h(x) = h_{\mathrm{maj}}(x)$. From this fact, the strong form of the pigeonhole principle implies that at least one $i \in \{1,2,3\}$ has $h_i(x) = h_{\mathrm{maj}}(x)$ (i.e., the majority vote must agree with the majority of classifiers in at least one of the three subsequences of classifiers). Furthermore, since each $\mathbb{A}(S_0; T_j)$ (with $j \in \{1,2,3\}$) supplies an equal number of entries to the sequence $\mathbb{A}(S;T)$ (by a straightforward inductive argument), for each $i \in \{1,2,3\}$, at least $1/4$ of the classifiers h in $\cup_{j \in \{1,2,3\} \setminus \{i\}} L(\mathbb{A}(S_0; T_j))$ have $h(x) = h_{\mathrm{maj}}(x)$: that is, since $|L(\mathbb{A}(S_0; T_i))| = (1/3)|L(\mathbb{A}(S;T))|$, we must have at least $(1/6)|L(\mathbb{A}(S;T))| = (1/4)\left|\bigcup_{j \in \{1,2,3\} \setminus \{i\}} L(\mathbb{A}(S_0; T_j))\right|$ classifiers h in $\bigcup_{j \in \{1,2,3\} \setminus \{i\}} L(S_0; T_j)$ with $h(x) = h_{\mathrm{maj}}(x)$ in order to meet the total of at least $(1/2)|L(\mathbb{A}(S;T))|$ classifiers $h \in L(\mathbb{A}(S;T))$ with $h(x) = h_{\mathrm{maj}}(x)$. In particular, letting I be a random variable uniformly distributed on $\{1,2,3\}$ (independent of the data), and letting $\tilde{h}$ be a random variable conditionally (given I and S) uniformly distributed on the classifiers $\cup_{j \in \{1,2,3\} \setminus \{I\}} L(\mathbb{A}(S_0; T_j))$, this implies that for any fixed $x \in \mathrm{ER}(h_{\mathrm{maj}})$, with conditional (given S) probability at least $1/12$,

$h_I(x) = \tilde{h}(x) = h_{\mathrm{maj}}(x)$, so that $x \in \mathrm{ER}(h_I) \cap \mathrm{ER}(\tilde{h})$ as well. Thus, for a random variable $X \sim \mathcal{P}$ (independent of the data and I,$\tilde{h}$), the law of total probability and monotonicity of conditional expectations imply

$$\begin{aligned}
&\mathbb{E}\left[\mathcal{P}\left(\mathrm{ER}(h_I) \cap \mathrm{ER}(\tilde{h})\right)\middle| S\right] = \mathbb{E}\left[\mathbb{P}\left(X \in \mathrm{ER}(h_I) \cap \mathrm{ER}(\tilde{h})\middle| I, \tilde{h}, S\right)\middle| S\right] \\
&= \mathbb{E}\left[\mathbb{1}\left[X \in \mathrm{ER}(h_I) \cap \mathrm{ER}(\tilde{h})\right]\middle| S\right] = \mathbb{E}\left[\mathbb{P}\left(X \in \mathrm{ER}(h_I) \cap \mathrm{ER}(\tilde{h})\middle| S, X\right)\middle| S\right] \\
&\geq \mathbb{E}\left[\mathbb{P}\left(X \in \mathrm{ER}(h_I) \cap \mathrm{ER}(\tilde{h})\middle| S, X\right)\mathbb{1}\left[X \in \mathrm{ER}(h_{\mathrm{maj}})\right]\middle| S\right] \\
&\geq \mathbb{E}\left[(1/12)\mathbb{1}\left[X \in \mathrm{ER}(h_{\mathrm{maj}})\right]\middle| S\right] = (1/12)\mathrm{er}_{\mathcal{P}}(h_{\mathrm{maj}}; f^\star).
\end{aligned}$$

Thus, on the event $\bigcap_{i\in\{1,2,3\}} E_i \cap E_i' \cap E_i''$,

$$\begin{aligned}
\mathrm{er}_{\mathcal{P}}(h_{\mathrm{maj}}; f^\star) &\leq 12\mathbb{E}\left[\mathcal{P}\left(\mathrm{ER}(h_I) \cap \mathrm{ER}(\tilde{h})\right)\middle| S\right] \\
&\leq 12 \max_{i\in\{1,2,3\}} \max_{j\in\{1,2,3\}\setminus\{i\}} \max_{h\in L(\mathbb{A}(S_0;T_j))} \mathcal{P}(\mathrm{ER}(h_i) \cap \mathrm{ER}(h)) \\
&< \frac{1800}{m+1}\left(d + \ln\left(\frac{18}{\delta}\right)\right) = \frac{c}{m+1}\left(d + \ln\left(\frac{18}{\delta}\right)\right).
\end{aligned}$$

Furthermore, by the union bound, the event $\bigcap_{i\in\{1,2,3\}} E_i \cap E_i' \cap E_i''$ has probability at least $1-\delta$. Thus, since this argument holds for any $\delta \in (0,1)$ and any finite sequence T with $f^\star \in \mathbb{C}[T]$, we have succeeded in extending the inductive hypothesis to include $m' = m$.

By the principle of induction, we have established the claim that, $\forall m \in \mathbb{N}$, $\forall \delta \in (0,1)$, for every finite sequence T of points in $\mathcal{X} \times \mathcal{Y}$ with $f^\star \in \mathbb{C}[T]$, with probability at least $1-\delta$,

$$\mathrm{er}_{\mathcal{P}}(\hat{h}_{m,T}; f^\star) \leq \frac{c}{m+1}\left(d + \ln\left(\frac{18}{\delta}\right)\right). \tag{9}$$

To complete the proof, we simply take $T = \emptyset$ (the empty sequence), and note that, for any $\varepsilon, \delta \in (0,1)$, for any value $m \in \mathbb{N}$ of size at least

$$\left\lfloor \frac{c}{\varepsilon}\left(d + \ln\left(\frac{18}{\delta}\right)\right)\right\rfloor, \tag{10}$$

the right hand side of (9) is less than ε, so that Majority($L(\mathbb{A}(\cdot;\emptyset))$) achieves a sample complexity equal the expression in (10). In particular, this implies

$$\mathcal{M}(\varepsilon, \delta) \leq \frac{c}{\varepsilon}\left(d + \ln\left(\frac{18}{\delta}\right)\right) = O\left(\frac{1}{\varepsilon}\left(d + \mathrm{Log}\left(\frac{1}{\delta}\right)\right)\right).$$

■

5. Remarks

On the issue of computational complexity, we note that the construction of subsamples by $\mathbb{A}$ can be quite efficient. Since the branching factor is 3, while $|S_0|$ is reduced by roughly a factor of 4 with each recursive call, the total number of subsamples returned by $\mathbb{A}(S;\emptyset)$ is a sublinear function of $|S|$. Furthermore, with appropriate data structures, the operations within each node of the recursion tree can be performed in constant time. Indeed, as discussed above, one can directly determine which data points to include in each subsample via a simple function of the indices, so that construction of these subsamples truly is computationally easy.

The only remaining significant computational issue in the learning algorithm is then the efficiency of the sample-consistent base learner L. The existence of such an algorithm L, with running time polynomial in the size of the input sequence and d, has been the subject of much investigation for a variety of concept spaces $\mathbb{C}$ (e.g., Khachiyan, 1979; Karmarkar, 1984; Valiant, 1984; Pitt and Valiant, 1988; Helmbold, Sloan, and Warmuth, 1990). For instance, the commonly-used concept space of *linear separators* admits such an algorithm (where $L(S)$ may be expressed as a solution of a system of linear inequalities). One can easily extend Theorem 2 to admit base learners L that are *improper* (i.e., which may return classifiers not contained in $\mathbb{C}$), as long as they are guaranteed to return a sample-consistent classifier in *some* hypothesis space $\mathcal{H}$ of VC dimension $O(d)$. Furthermore, as discussed by Pitt and Valiant (1988) and Haussler, Kearns, Littlestone, and Warmuth (1991), there is a simple technique for efficiently converting *any* efficient PAC learning algorithm for $\mathbb{C}$, returning classifiers in $\mathcal{H}$, into an efficient algorithm L for finding (with probability $1-\delta'$) a classifier in $\mathcal{H}$ consistent with a given data set S with $\mathbb{C}[S] \neq \emptyset$. Additionally, though the analysis above takes L to be deterministic, this merely serves to simplify the notation in the proof, and it is straightforward to generalize the proof to allow randomized base learners L, including those that fail to return a sample-consistent classifier with some probability δ' taken sufficiently small (e.g., $\delta' = \delta/(2|\mathbb{A}(S;\emptyset)|)$). Composing these facts, we may conclude that, for any concept space $\mathbb{C}$ that is efficiently PAC learnable using a hypothesis space $\mathcal{H}$ of VC dimension $O(d)$, there exists an efficient PAC learning algorithm for $\mathbb{C}$ with optimal sample complexity (up to numerical constant factors).

We conclude by noting that the constant factors obtained in the above proof are quite large. Some small refinements are possible within the current approach: for instance, by choosing the S_i subsequences slightly larger (e.g., $(3/10)|S|$), or using a tighter form of the Chernoff bound when lower-bounding N_i. However, there are inherent limitations to the approach used here, so that reducing the constant factors by more than, say, one order of magnitude, may require significant changes to some part of the analysis, and perhaps the algorithm itself. For this reason, it seems the next step in the study of $\mathcal{M}(\varepsilon,\delta)$ should be to search for strategies yielding refined constant factors. In particular, Warmuth (2004) has conjectured that the one-inclusion graph prediction algorithm also achieves a sample complexity of the optimal form. This conjecture remains open at this time. The one-inclusion graph predictor is known to achieve the optimal sample complexity in the closely-related *prediction model* of learning (where the objective is to achieve *expected* error rate at most ε), with a numerical constant factor very close to optimal (Haussler, Littlestone, and Warmuth, 1994). It therefore seems likely that a (positive) resolution of Warmuth's one-inclusion graph conjecture may also lead to improvements in constant factors compared to the bound on $\mathcal{M}(\varepsilon,\delta)$ established in the present work.

Appendix A. A Technical Lemma

The following basic lemma is useful in the proof of Theorem 2.[5]

Lemma 5 *For any* $a, b, c_1 \in [1, \infty)$ *and* $c_2 \in [0, \infty)$,

$$a \ln\left(c_1\left(c_2 + \frac{b}{a}\right)\right) \leq a \ln\left(c_1(c_2 + e)\right) + \frac{1}{e}b.$$

Proof If $\frac{b}{a} \leq e$, then monotonicity of $\ln(\cdot)$ implies

$$a \ln\left(c_1\left(c_2 + \frac{b}{a}\right)\right) \leq a \ln(c_1(c_2 + e)) \leq a \ln(c_1(c_2 + e)) + \frac{1}{e}b.$$

On the other hand, if $\frac{b}{a} > e$, then

$$a \ln\left(c_1\left(c_2 + \frac{b}{a}\right)\right) \leq a \ln\left(c_1 \max\{c_2, 2\}\frac{b}{a}\right) = a \ln\left(c_1 \max\{c_2, 2\}\right) + a \ln\left(\frac{b}{a}\right).$$

The first term in the rightmost expression is at most $a \ln(c_1(c_2+2)) \leq a \ln(c_1(c_2+e))$. The second term in the rightmost expression can be rewritten as $b\frac{\ln(b/a)}{b/a}$. Since $x \mapsto \ln(x)/x$ is nonincreasing on (e, ∞), in the case $\frac{b}{a} > e$, this is at most $\frac{1}{e}b$. Together, we have that

$$a \ln\left(c_1\left(c_2 + \frac{b}{a}\right)\right) \leq a \ln(c_1(c_2 + e)) + \frac{1}{e}b$$

in this case as well. ■

Acknowledgments

I would like to express my sincere thanks to Hans Simon and Amit Daniely for helpful comments on a preliminary attempt at a solution.

References

P. Auer and R. Ortner. A new PAC bound for intersection-closed concept classes. *Machine Learning*, 66(2-3):151–163, 2007. 3

M.-F. Balcan and P. M. Long. Active and passive learning of linear separators under log-concave distributions. In *Proceedings of the* 26^{th} *Conference on Learning Theory*, 2013. 3

A. Blumer, A. Ehrenfeucht, D. Haussler, and M. K. Warmuth. Learnability and the Vapnik-Chervonenkis dimension. *Journal of the Association for Computing Machinery*, 36(4): 929–965, 1989. 2, 3, 3, 4.3

5. This lemma and proof also appear in a sibling paper (Hanneke, 2015).

N. H. Bshouty, Y. Li, and P. M. Long. Using the doubling dimension to analyze the generalization of learning algorithms. *Journal of Computer and System Sciences*, 75(6): 323–335, 2009. 3

M. Darnstädt. The optimal PAC bound for intersection-closed concept classes. *Information Processing Letters*, 115(4):458–461, 2015. 3

A. Ehrenfeucht, D. Haussler, M. Kearns, and L. G. Valiant. A general lower bound on the number of examples needed for learning. *Information and Computation*, 82:247–261, 1989. 2, 3

E. Giné and V. Koltchinskii. Concentration inequalities and asymptotic results for ratio type empirical processes. *The Annals of Probability*, 34(3):1143–1216, 2006. 3

S. Hanneke. *Theoretical Foundations of Active Learning*. PhD thesis, Machine Learning Department, School of Computer Science, Carnegie Mellon University, 2009. 3, 4.1

S. Hanneke. Refined error bounds for several learning algorithms. *arXiv:1512.07146*, 2015. 3, 5

D. Haussler, M. Kearns, N. Littlestone, and M. K. Warmuth. Equivalence of models of polynomial learnability. *Information and Computation*, 95(2):129–161, 1991. 5

D. Haussler, N. Littlestone, and M. K. Warmuth. Predicting $\{0, 1\}$-functions on randomly drawn points. *Information and Computation*, 115:248–292, 1994. 3, 3, 5

D. Helmbold, R. Sloan, and M. K. Warmuth. Learning nested differences of intersection-closed concept classes. *Machine Learning*, 5(2):165–196, 1990. 5

N. Karmarkar. A new polynomial-time algorithm for linear programming. *Combinatorica*, 4(4):373–395, 1984. 5

L. G. Khachiyan. A polynomial algorithm in linear programming. *Soviet Mathematics Doklady*, 20:191–194, 1979. 5

P. M. Long. An upper bound on the sample complexity of PAC learning halfspaces with respect to the uniform distribution. *Information Processing Letters*, 87(5):229–234, 2003. 3

L. Pitt and L. G. Valiant. Computational limitations on learning from examples. *Journal of the Association for Computing Machinery*, 35(4):965–984, 1988. 5

H. Simon. An almost optimal PAC algorithm. In *Proceedings of the* 28^{th} *Conference on Learning Theory*, 2015. 1, 3, 4.1

L. G. Valiant. A theory of the learnable. *Communications of the ACM*, 27(11):1134–1142, 1984. 1, 5

A. W. van der Vaart and J. A. Wellner. *Weak Convergence and Empirical Processes*. Springer, 1996. 2

V. Vapnik. *Estimation of Dependencies Based on Empirical Data*. Springer-Verlag, New York, 1982. 2, 3, 4.3

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

M. K. Warmuth. The optimal PAC algorithm. In *Proceedings of the* 17^{th} *Conference on Learning Theory*, 2004. 3, 5