The One-Inclusion Graph Algorithm is Near-Optimal for the Prediction Model of Learning[*]

Yi Li[†] Philip M. Long[‡] Aravind Srinivasan[§]

ABSTRACT

Haussler, Littlestone and Warmuth described a general-purpose algorithm for learning according to the prediction model, and proved an upper bound on the probability that their algorithm makes a mistake in terms of the number of examples seen and the Vapnik-Chervonenkis dimension of the concept class being learned. We show that their bound is within a factor of $1 + o(1)$ of the best possible such bound for any algorithm.

Key Words and Phrases: Computational learning, One-Inclusion Graph Algorithm, Prediction model, Sample complexity, VC-dimension.

I INTRODUCTION

In the prediction model [1], the algorithm is trying to learn a $\{0,1\}$-valued function f (called the *target*) from a known class $\mathcal{F}$. An adversary chooses the target function f and a probability distribution D over the domain of f, and elements $x_1, ..., x_{m+1}$ are independently

[*] A preliminary version of this work appears as part of the paper "Improved Bounds on the Sample Complexity of Learning", in the *Proc. ACM-SIAM Symposium on Discrete Algorithms*, 2000.

[†] Department of Computer Science, National University of Singapore, Singapore 117543, Republic of Singapore, liyi@comp.nus.edu.sg.

[‡] Department of Computer Science, National University of Singapore, Singapore 117543, Republic of Singapore, plong@comp.nus.edu.sg. Supported by National University of Singapore Academic Research Fund Grant RP960625.

[§] Bell Laboratories, Lucent Technologies, 600-700 Mountain Avenue, Murray Hill, NJ 07974-0636, USA, srin@research.bell-labs.com. Part of this work was done while this author was at the School of Computing of the National University of Singapore.

chosen according to D. The algorithm is given $(x_1, f(x_1)), ..., (x_m, f(x_m))$ and x_{m+1}, and must predict the value of $f(x_{m+1})$. Let $p(d, m)$ be the best-possible upper bound on the probability of a mistake in this model in terms of the VC-dimension d of $\mathcal{F}$ and the number m of examples; see Section II for a precise definition.

Haussler, Littlestone and Warmuth [1] showed that $\frac{d}{2(m+1)} \leq p(d, m) \leq \frac{d}{m+1}$. The upper bound on $p(d, m)$ follows from their *One-Inclusion Graph Algorithm.*[1] Here, we show that $p(d, m) \geq \frac{(1-o(1))d}{m}$ (the $o(1)$ term goes to 0 as m increases), matching the upper bound up to lower-order terms. In the case $d = 1$, the class $\mathcal{F}$ used in our argument consists of indicator functions for paths in trees, instead of the half-intervals used for proving the lower bound of [1]. More precisely, given a tree $T = (V, E)$, $\mathcal{F}$ is the set of functions $f : V \to \{0, 1\}$ that are indicator functions of root-to-leaf paths. As has become standard since [4], we first consider the case in which the target is chosen uniformly at random (we set the distribution over the domain to be uniform as well), and lower bound the probability that the Bayes-optimal algorithm makes a mistake in this setting. This implies the same lower bound for any algorithm for a randomly chosen target, which in turn implies the same lower bound for any algorithm for a worst-case target. We generalize to the case $d > 1$ in a manner similar to [1].

II PRELIMINARIES

Fix a countably infinite domain X. The VC-dimension of a set $\mathcal{F}$ of functions from X to $\{0, 1\}$, denoted by VCdim($\mathcal{F}$), is the largest d such that there is a sequence $x_1, ..., x_d$ of domain elements from X such that

$$\{(f(x_1), ..., f(x_d)) : f \in \mathcal{F}\} = \{0, 1\}^d.$$

In the prediction model [1], we have a known family $\mathcal{F}$ of functions mapping X to

[1] Suppose $p(\mathcal{F}, m)$ is the worst case probability of a mistake when the target is chosen from $\mathcal{F}$ (again, see Section II for a formal definition). Note that to prove $p(d, m) \geq \frac{d}{2(m+1)}$, it is sufficient to establish the *existence* of a class $\mathcal{F}$ of VC-dimension d for which $p(\mathcal{F}, m) \geq \frac{d}{2(m+1)}$. Vapnik and Chervonenkis [2] (see [3]) showed that for *any* set $\mathcal{F}$ of VC-dimension d, $p(\mathcal{F}, m) \geq (1 - 1/m)\frac{d-1}{2em}$. Haussler *et al.* [1] further showed that for any set $\mathcal{F}$ of VC-dimension d, $p(\mathcal{F}, m) \geq \frac{d-1}{2em}$.

$\{0,1\}$. For some unknown function $f \in \mathcal{F}$ and some unknown distribution D on X, we get m independent samples $x_1, x_2, \ldots, x_m$ chosen from D, and also get the values of $f(x_1), f(x_2), \ldots, f(x_m)$. We will refer to $x_1, x_2, \ldots, x_m$ and $f(x_1), f(x_2), \ldots, f(x_m)$ collectively as a "sample", and denote it by S. Then, given x_{m+1} drawn from D and independently of the previous samples, the learner's goal is to guess the value of $f(x_{m+1})$ correctly with as high a probability as possible. Recall that f and D are *unknown*. Given a learning algorithm $\mathcal{A}$, let $p'(\mathcal{A}, \mathcal{F}, m)$ denote the supremum, over all choices of f and D, of the probability that $\mathcal{A}$ *incorrectly* predicts $f(x_{m+1})$. (This probability is taken over the random choices of $x_1, x_2, \ldots, x_m, x_{m+1}$, as well as the internal coin-flips of $\mathcal{A}$, in case $\mathcal{A}$ is a randomized algorithm.) Define $p(\mathcal{F}, m) = \inf_{\mathcal{A}} p'(\mathcal{A}, \mathcal{F}, m)$; thus, $p(\mathcal{F}, m)$ is the worst-case error probability of the "best" learning algorithm for $\mathcal{F}$. Finally, let $p(d, m)$ be the supremum of $p(\mathcal{F}, m)$ over all choices of $\mathcal{F}$ whose VC-dimension is d. Thus, if $p(d, m) \geq q$, then for any $\epsilon > 0$, there is a family $\mathcal{F}$ of VC-dimension d such that for any learning algorithm, there is a choice for f and D so that the algorithm has an error probability of at least $q - \epsilon$ in predicting the value of $f(x_{m+1})$. To show that $p(d, m) \geq q$ for a desired q, we will in fact fix both $\mathcal{F}$ and D suitably; the existence of an appropriate $f \in \mathcal{F}$ is then shown by the probabilistic method.

The following correlational result involving a "balls and bins" experiment will be useful.

Lemma 1 ([5, 6]) *Suppose we throw m balls independently at random into n bins, each ball having an arbitrary distribution. Let B_i be the random variable denoting the number of balls in the ith bin. Then for any $t_1, \ldots, t_n$,* $\mathbf{Pr}\left(\bigwedge_{i=1}^{n} B_i \geq t_i\right) \leq \prod_{i=1}^{n} \mathbf{Pr}(B_i \geq t_i)$.

III THE LOWER BOUND

The heart of our analysis is the proof of the following theorem, which concerns the case in which the VC-dimension is 1. (We denote the logarithm to the base 2 by "log", and the logarithm to the base e by "ln".) We extend this result to the case $d > 1$ in Theorem 3.

Theorem 2

$$p(1, m) \geq \frac{1}{m}\left(1 - O\left(\frac{(\log\log m)^2}{\log m}\right)\right).$$

Proof: As in [4], we will fix D, and describe a distribution over the choice of f such that, for any algorithm $\mathcal{A}$, the probability, with respect to the choice of f as well as the random examples, that $\mathcal{A}$ makes a mistake is lower bounded as in Theorem 2. This will imply the existence of f for which the probability of making a mistake has the same lower bound with respect only to the random choice of examples.

The concept class $\mathcal{F}$ that we use is as follows. Given m, the number of examples seen so far, for certain integers $b = b(m)$ and $h = h(m)$, let $T = (V, E)$ be a rooted full b-ary tree of height h. Let $n = (b^{h+1} - 1)/(b - 1)$ be the number of nodes in T. Recall that the height of a node is defined to be the height of the subtree rooted at that node (thus, leaves are at height 0). Let $\mathcal{F}$ consist of indicator functions for all subsets of V corresponding to root-to-leaf paths in T. More formally, letting $\text{ancs}(v)$ denote the set of ancestors of a vertex v of T define $f_v : V \to \{0, 1\}$ by $f_v(w) = 1$ iff $w \in \text{ancs}(v)$. (Recall that $v \in \text{ancs}(v)$.) Then, $\mathcal{F} = \{f_v : v \text{ is a leaf of } T\}$. It is easy to check that $\text{VCdim}(\mathcal{F}) = 1$.

For our proof, we will find it convenient to prove a lower bound for an artificial learning model in which the learning algorithm is given information about the function to be learned in addition to a random sample of its behavior. Since an algorithm can ignore this extra information, lower bounds for the revised model imply lower bounds for the original prediction model.

Let D be the uniform distribution over V, and suppose the function f to be learned is chosen uniformly at random from $\mathcal{F}$. It will be useful to view this choice as being made via a random walk from the root to a leaf, by first choosing $\vec{r}$ uniformly at random from $\{1, ..., b\}^h$, and then each time we need to decide which of b children to take, checking the appropriate component of $\vec{r}$.

The additional information given to the algorithm depends on the random sample S it receives and the function f to be learned, as follows. Define $\mathrm{low}(S)$ to be the positive example in S that lies furthest down the path defining f if there are any positive examples, and to be the root otherwise (note that the root is contained in all root-to-leaf paths). In addition to a sample S, the algorithm receives all the components of $\vec{r}$ *except* the component used to tell which child of $\mathrm{low}(S)$ to take. If the lowest positive example is a leaf, no information is given (nor is it needed). Call this information $c(S, \vec{r})$.

If $\mathrm{low}(S)$ is not a leaf, the positive examples in S, together with $c(S, \vec{r})$, narrow the possibilities for f to one of b paths (see Figure 1). Negative examples falling on some of these paths can eliminate them as possibilities (see Figure 2).

As is well-known (see [7]), the probability of mistake is minimized by any algorithm that, given (i) a sample S, (ii) x_{m+1} and (iii) $c(S, \vec{r})$, outputs the prediction for $f(x_{m+1})$ that minimizes the *a posteriori* probability of a mistake, after conditioning on (i), (ii) and (iii). One such optimal algorithm (let us call it A) predicts 1 if and only if the conditional probability that $f(x_{m+1}) = 1$ is strictly greater than $1/2$. Thus, if there are at least two possibilities for the function f to be learned that are consistent with the information in S and $c(S, \vec{r})$, algorithm A predicts 1 for all elements on the path from the root to $\mathrm{low}(S)$, and 0 everywhere else. This is because, in this case, all possibilities for f remaining are equally likely, and the unknown portions are disjoint. So for any v not on the path from the root to $\mathrm{low}(S)$, the *a posteriori* probability that $f(v) = 1$ is at most $1/2$. Of course, if only one possibility remains for f, then A predicts $f(x_{m+1})$ correctly. Let $\mathrm{deter}(S, \vec{r})$ be the predicate that f is determined by S and $c(S, \vec{r})$, that is, that there is only one possibility for f.

Define

$$\lambda = 1 - \left(1 - \left(1 - \frac{h}{n - (h+1)}\right)^m\right)^{b-1}. \quad (1)$$

For some sample S, denote the positions and the values of the positive examples in S by $\mathrm{pos}(S)$. Fix an arbitrary value S^+ for $\mathrm{pos}(S)$, and an arbitrary value I for $c(S, \vec{r})$ from among those consistent with S^+. Define $\mathcal{E}_1$ to be the

Figure 1: An example of the possibilities for the function to be learned after viewing the positive examples in a sample S, together with the extra information $c(S, \vec{r})$. The edges on paths that could possibly be the target are drawn with dotted lines. The algorithm does not know which child of the lowest positive example is taken, but it knows which child is taken on every other step along the path.

Figure 2: Now the negative examples have been seen, and one that fell on a path that was consistent with the positive examples and $c(S, \vec{r})$ has eliminated that path as a possibility.

event "pos$(S) = S^+$", and $\mathcal{E}_2$ to be the event "$c(S,\vec{r}) = I$". Our first goal is to show that

$$\mathbf{Pr}(\text{mistake} \,|\, (\mathcal{E}_1 \wedge \mathcal{E}_2)) \geq \frac{\lambda \cdot \text{height}(\text{low}(S^+))}{n}. \tag{2}$$

Here, "mistake" is the event that the optimal algorithm A predicts $f(x_{m+1})$ incorrectly, and "low(S^+)" is the value of low(S) conditional on $\mathcal{E}_1$. (Note that low(S) depends only on the positive examples of S. So, formally, low(S^+) is the value of low(S) for any sample S for which pos$(S) = S^+$.)

If height(low(S^+)) $= 0$ (i.e., if low(S^+) is a leaf), then (2) is trivial, so suppose from now on that height(low(S^+)) > 0. Then,

$$\mathbf{Pr}(\text{mistake} \,|\, (\mathcal{E}_1 \wedge \mathcal{E}_2)) \geq$$
$$\tfrac{\text{height}(\text{low}(S^+))}{n} \cdot \mathbf{Pr}(\text{not deter}(S,\vec{r}) \,|\, (\mathcal{E}_1 \wedge \mathcal{E}_2))$$

since if f is not determined by S, A will incorrectly predict 0 for all positive examples on the path below $low(S^+)$. Let alive(S^+, I) be the set of b elements of $\mathcal{F}$ consistent with the information in $\mathcal{E}_1 \wedge \mathcal{E}_2$.

Say that a root-to-leaf path is "hit" if one of its vertices is a negative example. Note that, after conditioning on $\mathcal{E}_1 \wedge \mathcal{E}_2$, we can view S as being filled in by sampling the remaining examples independently at random uniformly from $V - f^{-1}(1)$. Thus,

$$\mathbf{Pr}(\text{not deter}(S,\vec{r}) \,|\, (\mathcal{E}_1 \wedge \mathcal{E}_2))$$
$$=$$
$$\mathbf{Pr}(\exists\, g \in \text{alive}(S^+, I) - \{f\} \text{ not hit} \,|\, (\mathcal{E}_1 \wedge \mathcal{E}_2))$$
$$=$$
$$1 - \mathbf{Pr}(\forall\, g \in \text{alive}(S^+, I) - \{f\} \text{ hit} \,|\, (\mathcal{E}_1 \wedge \mathcal{E}_2))$$
$$\geq 1 - \left(1 - \left(1 - \frac{\text{height}(\text{low}(S^+))}{n - (h+1)}\right)^m\right)^{b-1}$$

by Lemma 1 together with the fact that there are at most m negative examples. The fact that height(low(S^+)) $\leq h$ proves (2). Also, for each choice of I, it is easy to check that

$$\mathbf{Pr}(\text{mistake} \,|\, (\mathcal{E}_1 \wedge \mathcal{E}_2)) = \mathbf{Pr}(\text{mistake} \,|\, \mathcal{E}_1).$$

Thus,

$$\mathbf{Pr}(\text{mistake}) \geq \frac{\lambda \cdot \mathbf{E}(\text{height}(\text{low}(S)))}{n}. \tag{3}$$

Let $\exp(x) \doteq e^x$. Applying the approximations $\exp(-x/(1-x)) \leq 1-x \leq \exp(-x)$ (for $x < 1$) and some manipulations, we get

$$\begin{aligned}
&\mathbf{E}(\text{height}(\text{low}(S))) \\
&= \sum_{j=1}^{h} \mathbf{Pr}(\text{height}(\text{low}(S)) \geq j) \\
&\geq \sum_{j=1}^{h} \left(1 - \frac{j}{n}\right)^m \\
&\geq \sum_{j=1}^{h} \exp\left(\frac{-\frac{jm}{n}}{1 - \frac{j}{n}}\right) \\
&\geq \sum_{j=1}^{h} \exp\left(\frac{-\frac{jm}{n}}{1 - \frac{h}{n}}\right) \\
&= \frac{\left(1 - \exp\left(\frac{-\frac{hm}{n}}{1-\frac{h}{n}}\right)\right) \exp\left(\frac{-\frac{m}{n}}{1-\frac{h}{n}}\right)}{1 - \exp\left(\frac{-\frac{m}{n}}{1-\frac{h}{n}}\right)} \\
&\geq \frac{\left(1 - \exp\left(-\frac{hm}{n}\right)\right)\left(1 - \frac{m}{n-h}\right)}{1 - \left(1 - \frac{m/n}{1-h/n}\right)} \\
&= \frac{n-m-h}{m} \cdot (1 - \exp(-\frac{hm}{n})). \qquad (4)
\end{aligned}$$

Similarly,

$$\lambda \geq 1 - \exp\left(-(b-1)\exp\left(\frac{-hm}{n-(2h+1)}\right)\right). \qquad (5)$$

We recall the standard "$\Theta(\cdot)$" notation. A parameter x is $\Theta(g(m))$ for a given function g, iff there are positive constants c_1, c_2 and c_3 such that for all $m \geq c_3$, we have $c_1 g(m) \leq x \leq c_2 g(m)$. Now, suppose m is large enough and that we can define the integers b and h so that

$$h = \left\lfloor \frac{\ln m}{\ln((\ln m) \cdot \ln \ln m)} \right\rfloor - 1; \qquad (6)$$

$$b \geq (\ln m) \cdot \ln \ln m; \qquad (7)$$

$$n = \frac{b^{h+1} - 1}{b - 1} = \Theta(m \ln m / (\ln \ln m)^2); \qquad (8)$$

$$\ln \ln m - 1 \leq hm/n \leq \ln \ln m. \qquad (9)$$

We have the following two bounds: (i) The bound in (4) is at least $\frac{n}{m} \cdot \left(1 - O\left(\frac{(\log \log m)^2}{\log m}\right)\right)$. To see this, we first note from (9) and (8) that $\frac{n-m-h}{m} = \frac{n}{m} \cdot \left(1 - O\left(\frac{(\log \log m)^2}{\log m}\right)\right)$. Next, (9) shows that

$$\begin{aligned}
&\left(1 - \exp(-\frac{hm}{n})\right) \geq 1 - \exp(-(\ln \ln m - 1)) \\
&\geq 1 - e/\ln m,
\end{aligned}$$

lower-bounding (4) as desired.

(ii) The right-hand-side of (5) is at least $1 - O(\frac{1}{\log m})$. To see this, we note from (9) and (8) that

$$\begin{aligned}
t &\doteq \frac{hm}{n - (2h+1)} \\
&= \frac{hm}{n} \cdot (1 + \Theta(h/n))
\end{aligned}$$

$$= \frac{hm}{n} + \Theta((\log\log m)^2/m)$$
$$\le \ln\ln m + \Theta((\log\log m)^2/m).$$

Thus, $\exp(-t) \ge \frac{1}{\ln m}\cdot(1-\Theta((\log\log m)^2/m))$. This, combined with (7), yields the desired lower bound on the r.h.s. of (5).

Putting these two bounds together with (3) proves Theorem 2. Let h be as in (6). We now show how to set a suitable value for b (and hence for n) so that (7), (8) and (9) hold. Define $\Phi(x) = hm(x-1)/(x^{h+1}-1)$. Let y be the unique real such that $y \ge 2$ and $\Phi(y) = \ln\ln m$; we set $b = \lceil y \rceil$. For notational convenience, define $\alpha = \ln((\ln m)\cdot\ln\ln m)$. We note two facts:

- $(\ln m)/\alpha - 2 \le h \le (\ln m)/\alpha - 1$, and
- $((\ln m)\cdot\ln\ln m)^{(\ln m)/\alpha} = m$.

These two facts show that for any $\beta = \gamma(\ln m)\cdot\ln\ln m$ where $\gamma \ge 1$, we have

$$\frac{\gamma^{(\ln m)/\alpha-1}\cdot m}{(\ln m)\cdot\ln\ln m} - 1 \le \beta^{h+1} - 1 \le \gamma^{h+1}\cdot m. \quad (10)$$

This implies that if m is large enough, then $\Phi((\ln m)\cdot\ln\ln m) > \ln\ln m > \Phi(2(\ln m)\cdot\ln\ln m)$. (To see the first inequality here, substitute $\gamma = 1$ in the second inequality in (10); to see the second inequality here, substitute $\gamma = 2$ in the first inequality in (10).) So, since $\Phi(x)$ is a decreasing function for $x > 0$, we get

$$(\ln m)\cdot\ln\ln m < y < 2(\ln m)\cdot\ln\ln m. \quad (11)$$

The first inequality in (11) validates (7). Next, suppose $(\ln m)\cdot\ln\ln m \le z \le 2(\ln m)\cdot\ln\ln m$ for some z. We have

$$\frac{\Phi(z+1)}{\Phi(z)} = \frac{z}{z-1}\cdot\frac{1-z^{-(h+1)}}{(1+z^{-1})^{h+1}-z^{-(h+1)}}$$
$$= (1+\Theta(1/z))\cdot\frac{1-z^{-(h+1)}}{1+\Theta(hz^{-1})-z^{-(h+1)}};$$

the second equality follows since $z = \Theta((\ln m)\cdot\ln\ln m)$ and $h = \Theta((\ln m)/(\ln\ln m))$. (The fact $(1+z^{-1})^{h+1} = 1+\Theta(hz^{-1})$ can be verified as follows. First, we clearly have $(1+z^{-1})^{h+1} \ge 1+h/z$. Second, $(1+z^{-1})^{h+1} \le \exp((h+1)/z)$. Since $(h+1)/z \le 1$ if m is large enough, we can use the bound $e^x \leq 1 + 2x$ which holds for $x \in [0,1]$, to get that $(1+z^{-1})^{h+1} \leq 1 + 2(h+1)/z \leq 1 + 3h/z$ if m is large enough.) Thus we see that $\Phi(z+1)/\Phi(z) \geq 1 - O((\ln\ln m)^{-2})$. Now, (11) in conjunction with the facts that: (i) $\Phi(x)$ is a decreasing function for $x > 0$, (ii) $\Phi(y) = \ln\ln m$, and (iii) $b = \lceil y \rceil$, yields

$$(1 - O((\ln\ln m)^{-2}))\ln\ln m \leq \Phi(b) \leq \ln\ln m.$$

Since $n = hm/\Phi(b)$, we get that (8) and (9) hold if m is large enough. □

Via a more complicated proof, a similar lower bound on $p(1,m)$ can be shown for the above learning problem for a complete binary tree (i.e., $b = 2$), for an appropriate choice of $n = n(m)$.

Finally, we generalize Theorem 2 to the case $d > 1$ in a similar manner as in [1]. We recall a standard case of the Chernoff-Hoeffding large-deviation bounds; see [8] for more information. Suppose $Y_1, Y_2, \ldots, Y_\ell$ are *independent* random variables, each taking on values in $[0,1]$. Let $Y = \sum_i Y_i$, with $\mathbf{E}(Y) = \mu$. Then, for any $\epsilon \in [0,1]$, the Chernoff-Hoeffding bounds show that

$$\mathbf{Pr}(Y \geq \mu(1+\epsilon)) \leq \exp(-\mu\epsilon^2/3). \qquad (12)$$

Since we are interested in the asymptotics as m increases, we assume in Theorem 3 that $m \geq 8d$.

Theorem 3 *For $m \geq 8d$, $p(d,m) \geq \frac{d}{m} \cdot \left(1 - O\left(\frac{(\log\log(m/d))^2}{\log(m/d)}\right)\right)$.*

Proof Sketch: Define $m' = \lceil m/d + \sqrt{(3m/d)\cdot\ln\ln(m/d)}\rceil$. The concept class $\mathcal{F}_d$ that we use is as follows. Recall that our concept class of VC-dimension one involved a $b = b(m)$-ary tree with $n = n(m)$ nodes. We will now work with a forest $F = (V,E)$, which contains d pairwise vertex-disjoint trees $T_i = (V_i, E_i)$, $i = 1, 2, \ldots, d$. Each T_i is a rooted complete b-ary tree with $n = n(m')$ nodes and $b = b(m')$. As before, for each $v \in V$, let $f_v : V \to \{0,1\}$ be such that $f_v(w) = 1$ iff $w \in \mathrm{ancs}(v)$. (Note in particular that if v and w are from different trees T_i and T_j, then

$f_v(w) = 0$.) Our concept class $\mathcal{F}_d$ is

$$\left\{ \sum_{i=1}^{d} f_{u_i} :\ u_i \text{ is a leaf of } T_i, \text{ for } i = 1, 2, \ldots, d \right\};$$

it is not hard to verify that $\mathrm{VCdim}(\mathcal{F}) = d$.

The adversary's strategy is to pick leaves $y_1, y_2, \ldots, y_d$ using random walks independently from $T_1, T_2, \ldots, T_d$ respectively; the unknown function is then set to be $\sum_{i=1}^{d} f_{y_i}$. The adversary also sets the distribution D of the samples x_i, to be uniform on V. As before, the learner wishes to maximize the probability of correctly guessing the value of $\sum_{i=1}^{d} f_{y_i}(x_{m+1})$. Note that if x_{m+1} belongs to some tree T_i, then this value (to be guessed) is simply $f_{y_i}(x_{m+1})$. It is also easy to check that those samples among the first m samples that fell in *other* trees T_j, give no information to the learner. We are thus essentially reduced to our earlier setting of the "single tree" problem. Since $m \geq 8d$, (12) shows that with probability at least $1 - 1/\ln(m/d)$, the number of samples among the first m that landed in T_i, is at most m'. (Briefly, let random variable Y_j be 1 if the jth sample landed in T_i, and be 0 otherwise. $Y = \sum_{j=1}^{m} Y_j$ is the number of samples among the first m that landed in T_i; we have $\mathbf{E}(Y) = m/d$. Bound (12) shows that $\mathbf{Pr}(Y \geq m') \leq 1/\ln(m/d)$.) Thus, since T_i has $n(m')$ nodes, Theorem 2 shows that

$$\begin{aligned} & p(\mathcal{F}_d, m) \\ & \geq \frac{d}{m} \cdot (1 - \frac{1}{\ln \frac{m}{d}}) \cdot (1 - O(\frac{(\log\log(m/d))^2}{\log(m/d)})) \\ & \geq \frac{d}{m} \cdot (1 - O(\frac{(\log\log(m/d))^2}{\log(m/d)})). \end{aligned}$$

(A minor subtlety that we have glossed over is that the number of samples falling in T_i may have been less than m'–it may not have exactly equaled m'. But this is not a problem, since it can be seen that for any concept class $\mathcal{F}$, $p(\mathcal{F}, m)$ is non-increasing as a function of m; indeed, if more samples cannot help, the optimal learner will simply ignore such samples.)

□

Acknowledgments. We thank the editor Prof. Gábor Lugosi and the reviewers for their helpful comments, and for pointing out a flaw in an earlier version.

References

[1] D. Haussler, N. Littlestone, and M. K. Warmuth. Predicting $\{0, 1\}$-functions on randomly drawn points. *Information and Computation*, 115(2):129–161, 1994. Preliminary version in FOCS'88.

[2] Vapnik V. and Chervonenkis A. *Theory of Pattern Recognition.* Nauka, Moscow, 1974. (In Russian); German translation: *Theorie der Zeichenerkennung*, Akademie Verlag, Berlin, 1979.

[3] L. Devroye, L. Györfi, and G. Lugosi. *A Probabilistic Theory of Pattern Recognition.* Springer, 1996.

[4] 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(3):247–251, 1989.

[5] C. L. Mallows. An inequality involving multinomial probabilities. *Biometrika*, 55:422–424, 1968.

[6] D. Dubhashi and D. Ranjan. Balls and bins: A study in negative dependence. *Random Structures & Algorithms*, 13(2):99–124, Sept 1998.

[7] R. O. Duda and P. E. Hart. *Pattern Classification and Scene Analysis.* Wiley, 1973.

[8] R. Motwani and P. Raghavan. *Randomized Algorithms.* Cambridge University Press, 1995.

List of Figures