An Improved Online Reduction from PAC Learning to Mistake-Bounded Learning

Lucas Gretta
lucas_gretta@berkeley.edu
UT Austin

Eric Price
ecprice@cs.utexas.edu
UT Austin

Abstract

A basic result in learning theory is that mistake-bounded learnability implies PAC learnability. It was shown in [Lit89] that, if a problem can be learned with M mistakes, it can be (ε, δ)-PAC-learned from $O(\frac{1}{\varepsilon}(M+\log\frac{1}{\delta}))$ samples. However, this reduction needs to store either $O(\frac{1}{\varepsilon}\log\frac{1}{\delta})$ samples or $O(M)$ hypotheses. A different reduction, in [KLPV87], only needs to store $O(1)$ samples and hypotheses but was only shown to work with $O(\frac{1}{\varepsilon}(M \cdot \log\frac{M}{\delta}))$ samples.

We give a refined analysis of the KLPV reduction, showing that it only uses $O(\frac{M}{\varepsilon})$ samples with probability $1 - M^{-O(1)}$. This gives the optimal sample complexity with only $O(1)$ space overhead, for $\delta > M^{-O(1)}$.

1 Introduction

Two well-known learning models are mistake-bounded [Lit87] and Probably Approximately Correct (PAC) [Val84] learning. The mistake-bound model describes *online* learning algorithms, which are given a series of examples that the learner must classify as it receives. The learner is informed each time it makes an error, and must make a finite number M of these mistakes for any sequence of examples. The PAC model instead describes *batch* learning, where a learner is given a number of labeled examples drawn from some (typically unknown) distribution, and needs to output a hypothesis with at most ε error on this distribution with probability $1 - \delta$.

Classic results in learning theory show that a mistake-bounded learner can be converted into a PAC learner. The first step is to convert the mistake-bounded learner to be *strongly conservative*: to only update its internal state when making a mistake [Lit87]. Then Littlestone [Lit89], building off Haussler [Hau88], showed how to convert a strongly conservative M-mistake-bounded learner into an (ε, δ) PAC learner that uses $S = O(\frac{1}{\varepsilon}(M+\log\frac{1}{\delta}))$ samples. The Littlestone approach [Lit89] has two stages: run the mistake-bounded learner for $O(\frac{1}{\varepsilon}M)$ samples, recording the (at most M) hypotheses produced; then test each of these hypotheses on a fresh set of $O(\frac{1}{\varepsilon}\log\frac{M}{\delta})$ samples and return the highest-performing hypothesis.

The Littlestone algorithm gets the optimal sample complexity $O(\frac{1}{\varepsilon}(M + \log\frac{1}{\delta}))$, but has relatively high *space* complexity. As originally described, it must remember all M hypotheses to measure their performance; alternatively, one could pre-sample and store the second-stage samples in order to judge the first-stage hypotheses on the fly and only track the best hypothesis, but this requires storing $O(\frac{1}{\varepsilon}\log\frac{M}{\delta})$ samples.

A different algorithm, given by [KLPV87; Ang88], uses much less space at the cost of sample complexity. This "LongSurvivor" algorithm, presented in Algorithm 1, repeatedly samples examples from $\mathcal{D}$ and feeds these into a strongly conservative mistake-bound algorithm $\mathcal{A}$. If $\mathcal{A}$'s current hypothesis makes no mistakes for $O(\frac{1}{\varepsilon}\log\frac{M}{\delta})$ samples, it is immediately returned. A simple union bound shows that this uses at most $O(\frac{1}{\varepsilon}M\log\frac{M}{\delta})$ samples. This is a log factor more than the optimal sample complexity, but the reduction is space-efficient: samples are not reused, so the reduction just needs to store the current hypothesis and one $O(\log(\frac{1}{\varepsilon}\log\frac{M}{\delta}))$-bit counter.

Because of its simplicity, LongSurvivor has been the algorithm typically presented in graduate classes on learning theory; see, for instance, [Pag10; Bal11; Blu14; Kan17].

Our Results. We show that the LongSurvivor algorithm actually requires only $O\left(\frac{1}{\varepsilon}M\frac{\log\frac{M}{\delta}}{\log(M/\log\frac{1}{\delta})}\right)$ samples. This means, for $\delta > 1/M^{O(1)}$, that it has the optimal sample complexity in addition to its space efficiency.

Algorithm 1 LongSurvivor Algorithm

$\mathcal{A}()$ returns the initial hypothesis, and $\mathcal{A}(h, x)$ returns a new hypothesis given that h makes a mistake on x. $O_{\mathcal{D}}$ is the oracle that samples examples according to $\mathcal{D}$ and labels according to h^*.

```
procedure LongSurvivor($\mathcal{A}, \delta$)
    $h \leftarrow \mathcal{A}()$
    $k \leftarrow \left\lceil \frac{\log(\frac{M}{\delta})}{\varepsilon} \right\rceil$
    while True do
        for $i$ in $1, 2, \ldots, k$ do
            $(x, h^*(x)) \leftarrow O_{\mathcal{D}}$
            if $h(x) \neq h^*(x)$ then
                $h = \mathcal{A}(h, x)$
                break
        if $h$ has survived all $k$ iterations then
            return $h$
```

Algorithm	Sample Complexity	Hypotheses Stored	Samples Stored
Littlestone [Lit89]	$\frac{1}{\varepsilon}(M + \log\frac{1}{\delta})$	M	1
(modified version)	$\frac{1}{\varepsilon}(M + \log\frac{1}{\delta})$	1	$\frac{1}{\varepsilon}\log\frac{1}{\delta}$
[SG95]	$\frac{1}{\varepsilon}(M + \log\frac{1}{\delta})$	M	1
[HK21]	$\frac{1}{\varepsilon}(M + \log\frac{1}{\delta})$	1	$\frac{1}{\varepsilon}(M + \log\frac{1}{\delta})$
LongSurvivor [KLPV87]	$\frac{1}{\varepsilon}M\log\frac{M}{\delta}$	1	1
Theorem 1.1	$\frac{1}{\varepsilon}M\frac{\log\frac{M}{\delta}}{\log(M/\log\frac{1}{\delta})}$	1	1

Table 1: A summary of reductions from PAC learning to mistake-bounded learning. All values are up to constant factors. The modified Littlestone algorithm stores second-stage samples and judges hypotheses on the fly, as described on page 1.

Theorem 1.1. *Given an M-mistake-bounded learning algorithm $\mathcal{A}$ and $O\left(\frac{1}{\varepsilon}M\frac{\log\frac{M}{\delta}}{\log(M/\log\frac{1}{\delta})}\right)$ labeled samples from the distribution $\mathcal{D}$,* LongSurvivor *produces an ε-accurate hypothesis h with probability* $1 - 2\delta$.

Note that we use the typically implicit convention that $\log x$ actually denotes $\max(1, \log x)$ in big-O notation.

We complement our upper bound by showing that our analysis of LongSurvivor is tight up to constants:

Theorem 1.2. *For any M, ε, δ there exists a classification problem P and an M-mistake-bounded learner for P such that* LongSurvivor *needs $\Omega\left(\frac{1}{\varepsilon}M\frac{\log\frac{M}{\delta}}{\log(M/\log\frac{1}{\delta})}\right)$ samples to act as an (ε, δ)-PAC learner.*

Thus, LongSurvivor has optimal sample complexity for $\delta > M^{-\Theta(1)}$, suboptimal but a $\log M$ factor better sample complexity than the previous bound [KLPV87] for $M^{-\Theta(1)} > \delta > e^{-M^{.999}}$, and really does match the previous $\Theta(\frac{M}{\varepsilon}\log\frac{M}{\delta})$ bound for $\delta < e^{-M}$.

1.1 Related Work A simplified MBL-to-PAC reduction was recently given in [BHMZ20; HK21]: run the mistake-bounded algorithm on the samples cyclically until it stops making mistakes on the samples. This has the optimal sample complexity and is simpler than Littlestone [Lit89], but has even higher space complexity (storing all $O(\frac{1}{\varepsilon}(M + \log\frac{1}{\delta}))$ samples).

An extension of the [Lit89] reduction was given in [SG95], where the hypotheses are tested and accepted or rejected on the fly, in parallel to the production of new hypotheses. This attains the optimal sample complexity with improved constants and performance in simulation. However, it still requires tracking up to M hypotheses at once.

The generalization of the MBL-to-PAC problem was presented and solved in [CCG04]. For MBL-to-PAC, the algorithm given still requires tracking all M hypotheses.

Not only do all the above algorithms require more space than LongSurvivor, they also require more *time*. For example, the second stage of Littlestone [Lit89] measures M hypotheses against $O(\frac{1}{\varepsilon}\log\frac{M}{\delta})$ fresh samples, which requires $O(\frac{M}{\varepsilon}\log\frac{M}{\delta})$ evaluations. This matches the *previous* time bound for LongSurvivor, and so is an $O(\log\frac{M}{\log\frac{1}{\delta}})$ factor worse than our new bound. The other algorithms are no more than constant factors faster than [Lit89].

LongSurvivor is sometimes used as a subprocedure of other algorithms, e.g. [HRZ07], which details an algorithm to learn large margin halfspaces. LongSurvivor is used to verify the accuracy of generated coresets.

2 Notation

By Geometric$(p), p > 0$, we denote the random variable that takes value $k \in \mathbb{Z}^+$ with probability $p(1-p)^{k-1}$. We occasionally abuse notation by using Geometric(0) to refer to a random variable that is always greater than any integer.

We use the natural logarithm throughout. The notation $f \lesssim g$ means that $f \leq Cg$ for some universal positive constant C. We use the (usually implicit, but fairly standard in TCS) definition of big-O notation for multiple variables that $f = O(g)$ means that $f \leq Cg$ for some universal positive constant C, with the convention that inside big-O notation, $\log x$ actually denotes $\max(1, \log x)$. [Hence, for example, $O(\frac{1}{\varepsilon}M\log M)$ is meaningful regardless of whether $M \to \infty$ or $\varepsilon \to 0$ or both, and is $O(1/\varepsilon)$ not 0 for $M = 1$.]

When we refer to an (ε, δ)-PAC learning algorithm that uses m samples to solve a classification problem $h^* : A \to B$, we mean an algorithm which, when given m labeled samples drawn from A according to $\mathcal{D}$, outputs $\hat{h}$ such that

$$\mathrm{err}_{\mathcal{D}}\hat{h} := \Pr_{a\sim\mathcal{D}}[h^*(a) \neq \hat{h}(a)] \leq \varepsilon$$

with probability $1-\delta$.

When we refer to an M-mistake-bounded learning algorithm to solve a classification problem $h^* : A \to B$, we mean an algorithm which is repeatedly: given a value $a \in A$, outputs a guess $h(a)$, is told the correct answer $h^*(a)$. On any sequence of values, the algorithm errs at most M times.

3 Proof overview

We give a brief overview of the proof of Theorem 1.1 in this section, and go into the details in the next.

Standard analysis. The chance any specific hypothesis h_i with error $\mathrm{err}_{\mathcal{D}}h_i > \varepsilon$ is returned is at most $(1-\varepsilon)^k \leq \exp(-\varepsilon k) \leq \delta/M$, so a union bound gives a δ chance of failure. One of the first $M+1$ hypotheses will have zero error and always be returned if reached, so this algorithm uses at most $(M+1)k \leq O(\frac{M}{\varepsilon}\log\frac{M}{\delta})$ samples.

Bounding the number of samples. What we show is that the LongSurvivor algorithm only uses $O(M/\varepsilon)$ samples with high probability. Therefore with just $O(M/\varepsilon)$ samples we can run the LongSurvivor algorithm with $\delta' = \delta/2$. The algorithm only fails if it either outputs a bad hypothesis ($\delta/2$ chance) or needs more than $O(M/\varepsilon)$ samples (less than $\delta/2$ chance), giving a PAC learner.

Consider $\delta > M^{-9}$. There are two kinds of hypotheses:

1. *bad* hypotheses, which have accuracy below $(1-\varepsilon/20)$, and
2. *good* hypotheses, which have accuracy above $(1-\varepsilon/20)$.

Each bad hypothesis is expected to fail out in $20/\varepsilon \ll k$ samples, so we only expect to spend $O(M/\varepsilon)$ samples on bad hypotheses. Because the times spent are bounded in $[1, k]$, the Chernoff bound gives that the time spent on these hypotheses is within $k\sqrt{M\log\frac{2}{\delta}} \ll M/\varepsilon$ of its expectation, with probability $1-\delta$.

Good hypotheses could take longer, up to k samples on average. But each good hypothesis has a nontrivial chance of passing all k samples and being output by the algorithm; as a result, the algorithm is unlikely to consider many good hypotheses before finishing. More precisely, each good hypothesis has at least a $(1-\varepsilon/20)^k \gtrsim 1/(M/\delta)^{1/20} \gtrsim 1/\sqrt{M}$ chance of terminating the algorithm, for $\delta > M^{-9}$. Therefore, after $O(\sqrt{M}\log\frac{1}{\delta})$ good hypotheses, we will have terminated with probability $1-\delta$. This means the algorithm uses at most $O(k\sqrt{M}\log\frac{1}{\delta}) \ll M/\varepsilon$ samples on good hypotheses.

Overall, the above argument shows that, for any $\delta = M^{-c}$, $O(cM/\varepsilon)$ samples suffice with probability $1-\delta$. This matches Theorem 1.1 for $\delta > M^{-O(1)}$ but is worse for small δ. A more carefully tuned threshold than $1-\varepsilon/20$ gives Theorem 1.1 for general δ.

4 Analysis

To show our result formally, we first prove a property of independent geometric random variables:

Lemma 4.1. *Let $p_1, \ldots, p_M \in [0,1]$ with $p_M = 0$, and let $T_i \sim$ Geometric(p_i) independently for $i \in [M]$. Define i^* to be the minimum i with $T_i > k$, for $k \in \mathbb{Z}^+, k \geq \log M$. Then*

$$\sum_{i=1}^{i^*-1} T_i \lesssim \frac{Mk}{\max(1, \log \frac{M}{\log \frac{2}{\delta}})}$$

with probability $1-\delta$.

Proof. Since $T_i \leq k$ for all $i < i^*$, the left hand side is at most $i^* k \leq Mk$. So the bound is trivial when $M < \log \frac{2}{\delta}$, and we can assume $M \geq \log \frac{2}{\delta}$.

Define

$$\gamma_j := \Pr[i^* > j] = \prod_{i=1}^{j}(1-(1-p_i)^k),$$

so γ is decreasing with $\gamma_0 = 1$ and $\gamma_M = 0$. Let $m \leq M$ be the point where this crosses $\frac{\delta}{2}$, i.e., $\gamma_m > \frac{\delta}{2} \geq \gamma_{m+1}$. We have that $i^* \leq m+1$ with probability $1-\gamma_{m+1} \geq 1-\frac{\delta}{2}$. Therefore, with $1-\frac{\delta}{2}$ probability,

$$T := \sum_{i=1}^{m} \min(T_i, k) \geq \sum_{i=1}^{i^*-1} T_i. \tag{4.1}$$

Since m is fixed, and the $\min(T_i, k)$ are independent variables bounded in $[1, k]$, we can apply the additive Chernoff bound to T: with probability $1-\delta/2$,

$$T \leq \mathbb{E}[T] + k\sqrt{\frac{1}{2}M \log \frac{2}{\delta}}. \tag{4.2}$$

Bounding the expectation. Let $S \subseteq [m]$ contain the indices i with $p_i \leq \frac{\alpha \log M}{2k} \leq \frac{1}{2}$, for $\alpha \in (0,1)$. Every $i \notin S$ satisfies $\mathbb{E}[T_i] = 1/p_i \leq \frac{2k}{\alpha \log M}$, so

$$\sum_{i \notin S} \min(\mathbb{E}[T_i], k) \leq \frac{2Mk}{\alpha \log M}. \tag{4.3}$$

On the other hand, we bound $|S|$. For each $i \in S$, the chance that $T_i > k$ is

$$(1-p_i)^k \geq e^{-2p_i k} \geq e^{-\alpha \log M} = M^{-\alpha}.$$

Since

$$\frac{\delta}{2} \leq \gamma_m \leq \prod_{i \in S}(1-(1-p_i)^k) \leq (1-M^{-\alpha})^{|S|} \leq e^{-|S|M^{-\alpha}},$$

this means

$$|S| \leq M^{\alpha} \log \frac{2}{\delta}$$

and hence

$$\sum_{i \in S} \min(\mathbb{E}[T_i], k) \leq M^{\alpha} k \log \frac{2}{\delta} \tag{4.4}$$

Combining with (4.3) gives

$$\mathbb{E}[T] \leq \sum_{i=1}^{m} \min(\mathbb{E}[T_i], k) \leq \frac{2Mk}{\alpha \log M} + M^{\alpha} k \log \frac{2}{\delta}.$$

We now set $\alpha = \frac{1}{2}\frac{\log \frac{M}{\log \frac{2}{\delta}}}{\log M}$. This gives:

$$\mathbb{E}[T] \leq \frac{4Mk}{\log \frac{M}{\log \frac{2}{\delta}}} + \left(\frac{M}{\log \frac{2}{\delta}}\right)^{1/2} k \log \frac{2}{\delta}.$$

Plugging into (4.2), with $1 - \frac{\delta}{2}$ probability we have

$$T \leq \frac{4Mk}{\log \frac{M}{\log \frac{2}{\delta}}} + (1 + \frac{1}{\sqrt{2}})k\sqrt{M \log \frac{2}{\delta}}.$$

Since we assume $M \geq \log \frac{2}{\delta}$, we know that this second term has

$$k\sqrt{M \log \frac{2}{\delta}} = Mk \frac{1}{\sqrt{M/\log \frac{2}{\delta}}} \lesssim Mk \frac{1}{\log(M/\log \frac{2}{\delta})}.$$

Applying to (4.1) and a union bound, we have that with probability $1 - \delta$,

$$\sum_{i=1}^{i^*-1} T_i \leq T \lesssim \frac{Mk}{\log \frac{M}{\log \frac{2}{\delta}}}$$

as desired.

□

Correctness is the same as in prior work:

Lemma 4.2. LongSurvivor *returns a hypothesis h such that $\mathrm{err}_{\mathcal{D}} h \leq \varepsilon$ with probability $\geq 1 - \delta$.*

Proof. Given that we are on hypothesis h which has error $\mathrm{err}_{\mathcal{D}} h > \varepsilon$, the chance we incorrectly return an invalid hypothesis is

$$\begin{aligned}
\Pr[\text{LongSurvivor terminates on } h] &= (1 - \mathrm{err}_{\mathcal{D}} h)^k \\
&\leq (1 - \varepsilon)^k \\
&= (1 - \varepsilon)^{\frac{\log(\frac{M}{\delta})}{\varepsilon}} \\
&\leq \exp(-\log(\frac{M}{\delta})) \\
&\leq \frac{\delta}{M}
\end{aligned}$$

We have at most $M + 1$ hypotheses, and the $M + 1$th has zero error so we have $\leq M$ invalid hypotheses. Applying a union bound, we get a $\leq M \cdot \frac{\delta}{M} = \delta$ chance of failure. □

Theorem 4.1. *Given an M-mistake-bounded learning algorithm $\mathcal{A}$ and $O\left(\frac{1}{\varepsilon} M \frac{\log \frac{M}{\delta}}{\log(M/\log \frac{1}{\delta})}\right)$ labeled samples from the distribution $\mathcal{D}$,* LongSurvivor *produces an ε-accurate hypothesis h with probability $1 - 2\delta$.*

Proof. According to Lemma 4.2, we have a δ chance of outputting an insufficiently accurate hypothesis. Next we bound the probability LongSurvivor needs more than $O(\frac{1}{\varepsilon}M\frac{\log\frac{M}{\delta}}{\log(M/\log\frac{1}{\delta})})$ samples.

Note that LongSurvivor goes through at most $M+1$ hypotheses by our mistake bound. Denote the ith hypothesis by h_i, and let $k := \frac{\log\frac{M}{\delta}}{\varepsilon}$.

We would like to treat each hypothesis as a biased coin to apply Lemma 4.1. Unfortunately the number of tests of hypothesis h_i and hypothesis h_j are not independent as h_j can depend on h_i in some complicated way.

We examine a related process instead: running LongSurvivor with a modified termination criterion. If $\mathrm{err}_{\mathcal{D}}h_i \neq 0$, then we keep testing h_i until it fails, and continue the process with h_{i+1}. If $\mathrm{err}_{\mathcal{D}}h = 0$, then we run $k+1$ times and terminate. Call this process LongSurvivor+. LongSurvivor+ behaves exactly like LongSurvivor until LongSurvivor terminates. Running LongSurvivor+ generates a sequence of hypotheses, terminating in a hypothesis with zero error. Let P denote the sequence of hypothesis errors, that is, $P = (\mathrm{err}_{\mathcal{D}}h_1, \mathrm{err}_{\mathcal{D}}h_2, \dots)$, and let T_i denote the number of times we test h_i. In LongSurvivor+, *conditioned on* P, the T_i are independent random variables with $T_i \sim \mathrm{Geometric}(p_i)$ for $p_i > 0$ and $T_i = k+1$ if $p_i = 0$. This is because the mistake-bounded learning algorithm is strongly conservative, so the number of samples matching the current hypothesis do not affect the algorithm.

Now, define i^* as the minimum i such that $T_i > k$. Then the sum $k + \sum_{i=1}^{i^*-1} T_i$ is the number of samples used by LongSurvivor. But for any fixed value p of P we can apply Lemma 4.1, getting that

$$\Pr[\sum_{i=1}^{i^*-1} T_i \geq O(\frac{1}{\varepsilon}M\frac{\log\frac{M}{\delta}}{\log(M/\log\frac{1}{\delta})})|P = p] \leq \delta$$

which implies that

$$\Pr[k + \sum_{i=1}^{i^*-1} T_i \geq O(\frac{1}{\varepsilon}M\frac{\log\frac{M}{\delta}}{\log(M/\log\frac{1}{\delta})})] \leq \delta.$$

Therefore, LongSurvivor uses $O(\frac{1}{\varepsilon}M\frac{\log\frac{M}{\delta}}{\log(M/\log\frac{1}{\delta})})$ samples with probability at least $1-\delta$.

Union bounding with the δ probability of outputting a bad hypothesis, LongSurvivor succeeds with $O(\frac{1}{\varepsilon}M\frac{\log\frac{M}{\delta}}{\log(M/\log\frac{1}{\delta})})$ samples with probability $1-2\delta$. □

5 Lower bound

Now we show that our analysis is tight up to constants.

Theorem 5.1. *For any M, ε, δ there exists a classification problem P and an M-mistake-bounded learner for P such that* LongSurvivor *needs* $\Omega\left(\frac{1}{\varepsilon}M\frac{\log\frac{M}{\delta}}{\log(M/\log\frac{1}{\delta})}\right)$ *samples to act as an (ε,δ)-PAC learner.*

Proof. Note that we can choose the learning problem and learner in such a way that we get a sequence of hypotheses with any error values we desire, as long as the $M+1$th hypothesis has zero error.

Let $r := \frac{M}{\log\frac{1}{2\delta}}$, and $k := \frac{\log\frac{M}{\delta}}{\varepsilon}$.

Case $M < 2\log\frac{1}{\delta}$. Let the error of the 1st through Mth hypotheses be $\frac{2}{k}$.

The probability that we reject one of the first M hypotheses after testing it somewhere in $[k/8, k]$ times is (given that we reach that hypothesis)

$$(1-\frac{2}{k})^{k/8-1} - (1-\frac{2}{k})^k \geq 1 - \frac{1}{4} - \exp(-2) \geq \exp(-.5)$$

Therefore, the chance this happens for all of the first M hypotheses is $\geq \exp(-\frac{M}{2}) > \delta$. So we have at least a δ probability of using $\Theta(\frac{M}{\varepsilon}\log\frac{1}{\delta})$ samples. This matches the desired bound $\Theta(\frac{1}{\varepsilon}M\frac{\log\frac{M}{\delta}}{\log(M/\log\frac{1}{\delta})})$ for this regime of M.

Case $M \geq 2\log\frac{1}{\delta}$. Let the error of the 1st through Mth hypotheses be $\frac{\log r}{k}$.

The probability that we accept any one of the first M hypotheses, given that we reach that hypothesis, is

$$(1-\frac{\log r}{k})^k \leq \exp(-\log r) = \frac{1}{r}.$$

The probability that we don't accept any of the first $M-1$ hypotheses is lower bounded by

$$(1-\frac{1}{r})^M = (1-\frac{\log\frac{1}{2\delta}}{M})^{M-1} > 2\delta. \tag{5.5}$$

Suppose that we ignore for now the termination criterion of LongSurvivor. The probability that we use more than $\frac{k}{\log r}$ tests on a given hypothesis is approximately $1/e$ and certainly more than .3. Therefore a Chernoff bound gives that, with probability $1-e^{-\Omega(M)} > 1-\delta/2$, at least $M/4$ of the first $M-1$ hypotheses will need at least $\frac{k}{\log r}$ tests.

Combined with (5.5), there is at least a 1.5δ chance that LongSurvivor uses $\frac{M}{4}\frac{k}{\log r} = \Theta(\frac{1}{\varepsilon}M\frac{\log\frac{M}{\delta}}{\log(M/\log\frac{1}{\delta})})$ samples. □

6 Conclusion

We have presented a refined analysis of the LongSurvivor reduction which shows it performs better than previously known, using only $O(\frac{M}{\varepsilon})$ samples with high probability. Though not as data efficient as [Lit89] for exponentially small δ, the LongSurvivor simple reduction also has the advantage of a small memory footprint and time complexity compared to other reductions [SG95; Lit89]. In algorithms where the LongSurvivor reduction is used as a subprocess, e.g. [HRZ07], this improved analysis can be used to tighten bounds.

Acknowledgments

The authors were supported by NSF awards CCF-1751040 (CAREER) and the NSF AI Institute for Foundations of Machine Learning (IFML). They would like to thank Aravind Gollakota and Sushrut Karmalkar for helpful conversations.

References

[Val84] L. G. Valiant. "A Theory of the Learnable". In: *Commun. ACM* 27.11 (Nov. 1984), pp. 1134–1142.

[KLPV87] Michael Kearns, Ming Li, Leonard Pitt, and Leslie G. Valiant. "Recent Results on Boolean Concept Learning". In: *proceedings of the fourth international workshop on machine learning.* Ed. by Pat Langley. Morgan Kaufmann, 1987, pp. 337–352.

[Lit87] Nick Littlestone. "Learning Quickly When Irrelevant Attributes Abound: A New Linear-Threshold Algorithm". In: *Proceedings of the 28th Annual Symposium on Foundations of Computer Science.* SFCS '87. USA: IEEE Computer Society, 1987, pp. 68–77.

[Ang88] Dana Angluin. "Queries and Concept Learning". In: *Mach. Learn.* 2.4 (Apr. 1988), pp. 319–342.

[Hau88] D. Haussler. *Space Efficient Learning Algorithms.* Tech. rep. UCSC-CRL-88-2. Santa Cruz, CA: University of Calif. Computer Research Laboratory, 1988.

[Lit89] Nick Littlestone. "From On-Line to Batch Learning". In: *Proceedings of the Second Annual Workshop on Computational Learning Theory.* COLT '89. Santa Cruz, California, USA: Morgan Kaufmann Publishers Inc., 1989, pp. 269–284.

[SG95] Dale Schuurmans and Russell Greiner. "Sequential PAC Learning". In: *Proceedings of the Eighth Annual Conference on Computational Learning Theory.* COLT '95. Santa Cruz, California, USA: Association for Computing Machinery, 1995, pp. 377–384.

[CCG04] N. Cesa-Bianchi, A. Conconi, and C. Gentile. "On the generalization ability of on-line learning algorithms". In: *IEEE Transactions on Information Theory* 50.9 (2004), pp. 2050–2057.

[HRZ07] Sariel Har-Peled, Dan Roth, and Dav Zimak. "Maximum Margin Coresets for Active and Noise Tolerant Learning". In: *Proceedings of the 20th International Joint Conference on Artifical Intelligence.* IJCAI'07. Hyderabad, India: Morgan Kaufmann Publishers Inc., 2007, pp. 836–841.

[Pag10] David Page. *Lecture Notes on Theoretical Approaches to Machine Learning.* `http://pages.cs.wisc.edu/~dpage/cs760/MLlecturePAC.pdf`. CS 760, UW-Madison. 2010.

[Bal11] Nina Balcan. *Lecture Notes.* `http://www.cs.cmu.edu/~ninamf/ML11/lect0908.pdf`. 8803 Machine Learning Theory, Georgia Tech. 2011.

[Blu14] Avrim Blum. *Lecture Notes on Online Learning.* `http://www.cs.cmu.edu/~avrim/ML14/lect0829.pdf`. CS-598, CMU. 2014.

[Kan17] Varun Kanade. *Lecture Notes on Online Learning, Mistake Bounds, Perceptron Algorithm.* `https://www.cs.ox.ac.uk/people/varun.kanade/teaching/AML-HT2017/lectures/mistakebound-online.pdf`. Advanced Machine Learning, Oxford. 2017.

[BHMZ20] Olivier Bousquet, Steve Hanneke, Shay Moran, and Nikita Zhivotovskiy. "Proper learning, Helly number, and an optimal SVM bound". In: *Conference on Learning Theory.* PMLR. 2020, pp. 582–609.

[HK21] Steve Hanneke and Aryeh Kontorovich. "Stable Sample Compression Schemes: New Applications and an Optimal SVM Margin Bound". In: *Proceedings of the 32nd International Conference on Algorithmic Learning Theory.* Ed. by Vitaly Feldman, Katrina Ligett, and Sivan Sabato. Vol. 132. Proceedings of Machine Learning Research. PMLR, 16–19 Mar 2021, pp. 697–721.