Appears in
Proceedings of the Eighth ACM Conference on Computational Learning Theory (COLT-95),
Santa Cruz, CA, July 1995.

Sequential PAC Learning

Dale Schuurmans
Department of Computer Science
University of Toronto
Toronto, Ontario M5S 1A4, Canada
dale@cs.toronto.edu

Russell Greiner
Siemens Corporate Research
Princeton, NJ 08540, USA
greiner@scr.siemens.com

Abstract

We consider the use of "on-line" stopping rules to reduce the number of training examples needed to pac-learn. Rather than collect a large training sample that can be proved sufficient to eliminate all bad hypotheses *a priori*, the idea is instead to observe training examples one-at-a-time and decide "on-line" whether to stop and return a hypothesis, or continue training. The primary benefit of this approach is that we can detect when a hypothesizer has actually "converged," and halt training before the standard fixed-sample-size bounds. This paper presents a series of such *sequential* learning procedures for: distribution-*free* pac-learning, "mistake-bounded to pac" conversion, and distribution-*specific* pac-learning, respectively. We analyze the worst case *expected* training sample size of these procedures, and show that this is often smaller than existing fixed sample size bounds — while providing the *exact same* worst case pac-guarantees. We also provide lower bounds that show these reductions can at best involve constant (and possibly log) factors. However, empirical studies show that these sequential learning procedures actually use *many times* fewer training examples in practice.

1 Introduction

1.1 Model

We consider the standard problem of learning an accurate concept definition from examples: given a target concept $c : X \to \{0,1\}$ defined on a domain X, we are interested in observing a sequence of training examples $\langle\langle x_1, c(x_1)\rangle, ..., \langle x_t, c(x_t)\rangle\rangle$ and producing a hypothesis $h : X \to \{0,1\}$ that agrees with c on as much of the domain as possible. Here we are addressing the standard *batch* training protocol, where after a finite number of training examples the learner must produce a hypothesis h, which is then tested *ad infinitum* on subsequent test examples. We also adopt the standard (noise free) "i.i.d. random examples" model of the learning situation, which assumes domain objects are independently generated by a fixed domain distribution P on X and labelled according to a fixed target function $c: X \to \{0,1\}$. Thus, the *error* of a hypothesis h with respect to target concept c and a domain distribution P is given by $d_{\mathrm{P}}(h,c) \triangleq \mathrm{P}\{x : h(x) \neq c(x)\}$.

Given this model, we are interested in meeting the so-called *pac*(ϵ, δ)*-criterion*: producing a hypothesis h with error at most ϵ, with probability at least $1-\delta$. Of course, the difficulty of achieving this criterion depends on our prior knowledge of c and P. Here we will consider two distinct models of prior knowledge: the distribution-*free* model [Val84], where the target concept c is known to belong to some class C, but nothing is known about the domain distribution P; and the distribution-*specific* model [BI88a, Kul91], where the domain distribution P is *known*, but the target concept c is assumed only to belong to some class C. In either case, we consider what can be achieved in the "worst case" sense:

Definition 1 (Pac-learning problem) *A learner L* solves *the distribution-*specific *pac-learning problem $(C, \mathrm{P}, \epsilon, \delta)$ if, for* any *target concept $c \in C$, L returns a hypothesis h such that $d_{\mathrm{P}}(h,c) \leq \epsilon$, with probability at least $1-\delta$. A learner L* solves *the distribution-*free *pac-learning problem (C, ϵ, δ) if it solves $(C, \mathrm{P}, \epsilon, \delta)$ for all domain distributions* P.

In general, a *learner* L consists of a *stopping rule* $T_L(C, \epsilon, \delta) : (X \times \{0,1\})^\infty \to \mathbb{N}$ that maps training sequences to stopping times (where the event $\{T_L = t\}$ depends only on the first t examples), and a *hypothesizer* $H_L(C, \epsilon, \delta) : (X \times \{0,1\})^* \to \{0,1\}^X$ that maps finite sequences of training examples to hypotheses.

As well as designing *correct* pac-learning procedures, we are interested in developing *efficient* learning procedures, and determining the inherent *complexity* of pac-learning problems. (Note that our definitions deliberately separate the *correctness* of a learner from its *efficiency.*) Our primary focus is on the issue of *data*-efficiency rather than *computational*-efficiency.

1.2 Issue

Many algorithms have been developed for pac-learning various concept classes in the distribution-free model. Most of these procedures follow a simple (collect; find) fixed-sample-size strategy we call Procedure F (Figure 1). Ensuring the correctness of F is a simple matter of finding an appropriate sample size function $T_{\mathrm{F}}(C, \epsilon, \delta)$ that can be proved sufficient to eliminate every ϵ-bad hypothesis from C with probability at least $1 - \delta$. This is normally accomplished by using well-known results on the uniform convergence of families of frequency estimates to their true probabilities. *E.g.*, for *finite* concept classes $T_{finite}(C, \epsilon, \delta) = \frac{1}{\epsilon} \ln \frac{|C|}{\delta}$ random training examples are sufficient to ensure F pac(ϵ, δ)-learns C. For *infinite* concept classes, Blumer *et al.* [BEHW89] use the results of Vapnik and Chervonenkis [VC71] to show that for any (well behaved[1]) concept class C with $\textsc{vc}(C) = d$ $T_{BEHW}(C, \epsilon, \delta) = \max\left\{ \frac{8d}{\epsilon} \log_2 \frac{13}{\epsilon},\ \frac{4}{\epsilon} \log_2 \frac{2}{\delta} \right\}$ random examples are sufficient for Procedure F to solve (C, ϵ, δ).[2] In addition, Ehrenfeucht *et al.* [EHKV89] have shown that *no* learning procedure can observe fewer than $t_{EHKV}(C, \epsilon, \delta) = \max\left\{ \frac{d-1}{32\epsilon},\ \frac{1-\epsilon}{\epsilon} \ln \frac{1}{\delta} \right\}$ random training examples and still meet the pac(ϵ, δ)-criterion for every target concept $c \in C$ and domain distribution P. Therefore, Procedure F, using T_{BEHW} or T_{STAB}, pac(ϵ, δ)-learns concept classes with near-optimal data-efficiency (up to constants and a $\ln 1/\epsilon$ factor).

```
Procedure F $(C, \epsilon, \delta)$
COLLECT $T_{\mathrm{F}}(C, \epsilon, \delta)$ training examples, sufficient to eliminate every $\epsilon$-bad concept from $C$ with probability at least $1 - \delta$.
RETURN any $h \in C$ that correctly classifies every example.
```

Figure 1: Procedure F

However, despite these impressive results, pac-learning theory has arguably had little direct impact on the actual practice of machine learning. The problem is that the sufficient sample size bounds T_{BEHW} and T_{STAB} are far too large to be practical in most applications, even for reasonable choices of C, ϵ, and δ. This is a serious shortcoming in practice, where *training data*, not computation time, is often the critical resource. Common speculation (among practitioners) is that these large sample sizes inevitably follow from worst case guarantees — as this forces one to consider "pathological" domain distributions, when in fact much nicer distributions are "typically" encountered in practice. This motivates research that makes distributional assumptions in order to improve data-efficiency, *e.g.*, [BI88a, Bau90, AKA91, BW91].[3] However, there is a fundamental weakness in this line of reasoning: no-one has actually demonstrated that these "pathological" distributions really exist (for this would be tantamount to improving the lower bound result t_{EHKV}). Since the gap between T_{STAB} and t_{EHKV} is actually quite large (roughly a factor of $64 \ln(6/\epsilon)$), it is not clear that the worst case situation is really as bad as T_{STAB} suggests.

Approach: We consider an alternative view: perhaps the simplistic (collect; find) fixed-sample-size approach is not particularly data-efficient. This raises the question of whether alternative learning strategies might require fewer training examples. To this end, we investigate *sequential* learning procedures that observe training examples one-at-a-time, and *autonomously* decide when to stop training and return a hypothesis. The idea is to detect situations where an accurate hypothesis can be reliably returned even before the fixed-sample-size bounds have been reached. Our goal is to reduce the number of training examples observed, while still meeting the exact same pac-criterion as before: returning an ϵ-bad hypothesis with probability at most δ, in any situation permitted by our prior knowledge.

The first issue we must face is the fact that a sequential learner observes a *random*, rather than fixed, number of training examples. Thus, to compare the data-efficiency of our approach with previous techniques, we must compare a *distribution* of training sample sizes to a fixed number. There are a number of ways one could do this, but we focus on what is arguably the most natural measure: comparing the *average* (*i.e.*, expected) training sample size of a sequential learner with the *fixed* sample size demanded by previous approaches to solve the *same* pac-learning problem.

1.3 Results

In this paper we introduce a number of sequential pac-learning procedures, prove them to be correct pac-learners, derive upper bounds on their worst case expected data-efficiency, and derive lower bounds on the worst case expected data-complexity of pac-learning problems.

First, in Section 2 we consider the general problem of distribution-*free* pac-learning. Here we introduce a novel learning procedure S that works by keeping a list of hypotheses (produced by some consistent hypothesizer), testing each one "on-line" with a *sequential probability*

[1] Uniform convergence results assume the concept class C satisfies certain benign measurability restrictions. All concept classes we consider are assumed to be suitably "well behaved" in this manner.

[2] This result has since been improved by Shawe-Taylor *et al.* [STAB93] to $T_{STAB}(C, \epsilon, \delta) = \frac{1}{\epsilon(1-\sqrt{\epsilon})} \left(2d \ln \frac{6}{\epsilon} + \ln \frac{2}{\delta} \right)$.

[3] This is a different motivation from using distributional assumptions to reduce the *computational* complexity of pac-learning. *E.g.*, while μformulae *cannot* be efficiently pac-learned (unless standard cryptographic assumptions are false) [KV89], Schapire [Sch92] has demonstrated a poly-time learning procedure for (μformulae, uniform).

ratio test (`sprt`) [Wal47] to see whether any has sufficiently small error. We show (Theorem 1) that `S` correctly solves any pac-learning problem (C, ϵ, δ) for which $d = \text{VC}(C) < \infty$, $\epsilon > 0$, $\delta > 0$. An analysis of `S`'s data-efficiency (Theorem 2) shows that `S` never observes more than $\text{E}T_{\mathtt{S}}(C, \epsilon, \delta) \leq O(\frac{d}{\epsilon} \ln \frac{1}{\epsilon} + \frac{1}{\epsilon} \ln \frac{1}{\delta})$ training examples (on average), for any c in C and P. This bound actually beats T_{BEHW} and T_{STAB} for extremely small values of δ (Proposition 3). However, we note that `S`'s true data-efficiency is *decoupled* from any precise bounds we can prove about its performance, and empirical tests [SG95] show that `S` actually uses *many times* fewer training examples in practice. Finally, we prove (Theorem 4) that these results cannot be substantially improved upon, as any learner must always observe an average of at least $t_{avg}(C, \epsilon, \delta) \geq \Omega(\frac{d}{\epsilon})$ random training examples in order to correctly solve any pac-learning problem (C, ϵ, δ).

Next, in Section 2.1 we briefly consider the special case of *finite* concept classes. Here we show (Proposition 5) that a variant of Procedure `S` can perform "mistake bounded to pac" conversion while using strictly fewer training examples (on average) than the procedure proposed in [Lit89]. In fact, our procedure uses *substantially* fewer training examples in empirical tests.

Finally, in Section 3 we address the distribution-*specific* model of pac-learning. Here we introduce a variant of Procedure `S`, Procedure `Scov`, that correctly solves any pac-learning problem $(C, \text{P}, \epsilon, \delta)$ for which C has a finite "$\epsilon/2$-cover" under d_{P}. We show (Theorem 7) that `Scov` uses about 5 *times* fewer training examples (on average) than the fixed-sample-size procedure introduced in [BI88a]. However, a lower bound result (Theorem 8) shows that sequential learning does not increase the range of pac-learnable concept spaces.

1.4 Significance and related work

Overall, these results show how one can achieve the standard pac-learning guarantees, while significantly reducing the number of training examples required in practice. Although our theoretical bounds for distribution-free pac-learning are comparable to previous bounds, in practice Procedure `S` actually uses *many times* fewer training examples than previous fixed-sample-size approaches, while providing the *exact same* worst case pac-guarantees. Moreover, `S` introduces little additional computational overhead over `F`. Interestingly, the advantages of sequential learning become even more apparent when we consider distribution-specific pac-learning, where we can prove a substantial reduction in worst case expected data-efficiency over previous approaches.

While tighter analyses and more sophisticated procedures are certainly possible, nevertheless, we feel that these results open the way to exploring a much wider (and more interesting) range of learning algorithms in computational learning theory. Furthermore, the empirical performance of these sequential learners actually appears to be approaching near-"practical" levels (even while maintaining the theoretical guarantees), which we feel brings the theory closer to practical applications.

Related work: Many authors have sought to improve the data-efficiency of pac-learning procedures, but generally by incorporating additional assumptions about the domain distribution, *e.g.*, [Bau90, BW91, AKA91]. Our goal is to improve data-efficiency without making additional assumptions.

While work on *nonuniform* pac-learning [BI88b, LMR91, Koi94] resembles the present study by also using "on-line" stopping rules, it has a fundamentally different aim: Our goal is to obtain a *uniform* improvement in data-efficiency for *all* target concepts c in C, whereas nonuniform pac-learning *sacrifices* data-efficiency for certain target concepts (late in a preference ranking $C_1 \subset C_2 \subset ... = C$), in order to obtain an improvement for others (early in the ranking). The real goal of nonuniform pac-learning is to increase the *range* of pac-learnable concept classes (*e.g.*, to certain classes with *infinite* VCdimension), rather than improve data-efficiency on previously pac-learnable classes.[4]

It is also important to distinguish our approach from *on-line* learning, *e.g.*, [Lit89, LW89, HLL92]. On-line learning considers a "learning while doing" model which is fundamentally different from the "batch" paradigm considered here. We really are following the standard batch ("train then test") protocol introduced by [Val84] — the only difference is that we permit the size of the training sample to be under the learner's control rather than set by the designer *a priori*.

2 Distribution-*free* pac-learning

We first consider the problem of distribution-free pac-learning. Here we assume we have access to a "consistent" hypothesizer H, which produces concepts $h \in C$ that correctly classify every training example. Given such a hypothesizer, our basic strategy is to observe training examples, collect consistent hypotheses from H, and test these hypotheses against future training examples until one proves to have sufficiently small error. The main trick is to find an appropriate *stopping rule* that guarantees the pac-criterion, while observing as few training examples as possible.

Obvious approach: Perhaps the most obvious approach is the basic *repeated significance testing* strategy of nonuniform pac-learning: test a series of consistent hypotheses and accept the first one that correctly classifies sufficiently many consecutive training examples; see Procedure `R` in Figure 2. Although this is a plausible approach (which, in fact, works well in practice), it is hard to prove a reasonable bound on `R`'s expected train-

[4] To illustrate that these two issues really are orthogonal, note that one could easily incorporate a *sequential* approach to nonuniform pac-learning — using a sequential procedure (like `S`) for learning each sub-class $C_1 \subset C_2 \subset ... = C$ to obtain improved performance for each sub-class, in addition to the standard nonuniform advantages.

```
Procedure R ($C, \epsilon, \delta, H$)
OBTAIN an initial hypothesis $h_0$ from $H$. Fix a sequence
    $\{\delta_i\}_1^\infty$ such that $\sum \delta_i = \delta$.
SEQUENTIALLY observe training examples:
    RETURN current hypothesis $h_i$ if it correctly classifies
    $\frac{1}{\epsilon} \ln \frac{1}{\delta_i}$ consecutive training examples.
    REJECT hypothesis $h_i$ if it ever misclassifies a training
    example (and call $H$ to obtain $h_{i+1}$).
```

Figure 2: Procedure R

ing sample size. The problem is that R rejects "good enough" hypotheses with high probability, and yet takes a long time to do so (*i.e.*, R rejects hypotheses of error ϵ with probability $1-\delta$, but this takes $\frac{1}{\epsilon}$ expected time). Therefore, if H produces a series of "borderline" hypotheses, R will take a long time to terminate (expected time about $\frac{1}{\epsilon\delta}$, which is not very good). This prevents us from proving good bounds on R's data-efficiency — unless we incorporate additional assumptions about H, or somehow use the fact that H cannot produce an endless sequence of consistent hypotheses of ϵ error. However, it could simply be that R is not a particularly data-efficient approach. Rather than pursue a complicated analysis, we consider an alternative strategy which works better.

Improved approach: Here we propose a novel sequential learning strategy S, which is also based on repeated significance testing, but avoids the apparent inefficiency of R's "survival testing" approach; see Figure 3. Procedure S is based on two ideas: First, instead of discarding hypotheses after a single mistake, S *saves* hypotheses, and continues testing them until one proves to have small error. Second, S tests hypotheses by using a *sequential probability ratio test* (sprt) [Wal47] that decides *on-line* whether a hypothesis is sufficiently accurate; see Figure 4. Not only does S prove to be a correct pac-learning procedure, but we can also derive a reasonable upper bound on its expected sample size.

Theorem 1 (Correctness) *For any $\epsilon > 0$, $\delta > 0$, and any (well behaved) concept class C with* $\text{VC}(C) < \infty$*: using any consistent hypothesizer H for C, Procedure S meets the pac(ϵ,δ)-criterion for any $c \in C$ and* P.

Proof (Outline) First, to show S terminates with probability 1 (wp1) we note that *(i)* sprt eventually accepts any $\frac{\epsilon}{\kappa}$-good hypothesis wp1 (Lemma 9 in Appendix), and *(ii)* H eventually produces such a hypothesis wp1 (Lemma 12). Correctness then follows from the correctness of sprt [Wal47], and the fact that S accepts an ϵ-bad hypothesis with probability at most $\sum \delta_i = \delta$.(Note that this result generalizes to any class C that can be decomposed as $C = \cup_1^\infty C_i$, $\text{VC}(C_i) < \infty$, provided H guesses consistent concepts from earlier classes first.) □

```
Procedure S ($C, \epsilon, \delta, H$)
OBTAIN an initial hypothesis $h_0$ from $H$. Fix a sequence
    $\{\delta_i = \frac{6\delta}{\pi^2 i^2}\}_{i=1}^\infty$, and fix a constant $\kappa > 1$.
SEQUENTIALLY observe training examples:
    SUBJECT each hypothesis $h_i$ to a sprt by calling
        sprt($h_i(x) \neq c(x)$, $\frac{\epsilon}{\kappa}$, $\epsilon$, $\delta_i$, 0),
    which accepts $h_i$, if $\epsilon$-bad, with probability at most $\delta_i$.
    RETURN the first $h_i$ accepted by sprt.
    IF the current hypothesis $h_i$ ever makes a mistake, call
    $H$ to obtain an additional $h_{i+1}$ (begin testing $h_{i+1}$).
```

Figure 3: Procedure S

```
Procedure sprt ($\phi(x)$, $a$, $r$, $\delta_{acc}$, $\delta_{rej}$)
For Boolean random variable $\phi(x)$, test
$H_{acc}$: $\text{P}\{\phi(x) = 1\} \leq a$   vs.   $H_{rej}$: $\text{P}\{\phi(x) = 1\} \geq r$,  with:
- probability of incorrectly deciding $H_{acc}$ bounded by $\delta_{acc}$,
- probability of incorrectly deciding $H_{rej}$ bounded by $\delta_{rej}$.
SEQUENTIALLY observe the sum:
        $S_t(\boldsymbol{\phi}^t) = \sum_{\phi_i \in \boldsymbol{\phi}^t} \phi_i \ln \frac{a}{r} + (1 - \phi_i) \ln \frac{1-a}{1-r}$.
    RETURN "accept" if ever $S_t(\boldsymbol{\phi}^t) \geq \ln 1/\delta_{acc}$.
    RETURN "reject" if ever $S_t(\boldsymbol{\phi}^t) \leq \ln \delta_{rej}$.
```

Figure 4: Procedure sprt

Theorem 2 (Data efficiency) *For any $\epsilon > 0$, $\delta > 0$, and any (well behaved) concept class C with* $\text{VC}(C) = d < \infty$*: using any consistent hypothesizer H for C and any constant $\kappa > 1$, Procedure S uses an average training sample size of at most*

$$\text{E}T_{\mathtt{S}}(C, \epsilon, \delta) \leq \left(\frac{\kappa}{\kappa - 1 - \ln \kappa}\right) \frac{1}{\epsilon} \left([2.12\kappa d + 3] \ln \frac{14\kappa}{\epsilon} + \ln \frac{1}{\delta}\right).$$

Proof (Outline) Using the fact (again) that sprt accepts any $\frac{\epsilon}{\kappa}$-good hypothesis wp1, we bound S's stopping time by $T_{\mathtt{S}}(\epsilon, \delta) \leq T_H(\frac{\epsilon}{\kappa}) + T_{\mathtt{sprt}}(\frac{\epsilon}{\kappa}, \epsilon, \delta_{T_H})$, where $T_H(\frac{\epsilon}{\kappa})$ is the time for H to produce an $\frac{\epsilon}{\kappa}$-good hypothesis h_i, and $T_{\mathtt{sprt}}$ is the time to accept any such hypothesis once produced (using the bound $i \leq T_H$). Thus, $\text{E}T_{\mathtt{S}} \leq \text{E}T_H + \text{E}T_{\mathtt{sprt}}$. Lemma 11 shows that

$$\text{E}T_{\mathtt{sprt}}(\tfrac{\epsilon}{\kappa}, \epsilon, \delta_{T_H}) \quad \leq \quad \left(\frac{\kappa}{\kappa - 1 - \ln \kappa}\right) \frac{1}{\epsilon} \left(\ln \frac{1}{\delta_{T_H}} + 1\right),$$

and Lemma 13 shows

$$\text{E}T_H(\tfrac{\epsilon}{\kappa}) \quad \leq \quad \frac{1}{1 - \sqrt{\epsilon/\kappa}} \frac{\kappa}{\epsilon} \left(2d \ln \frac{6\kappa}{\epsilon} + \ln 2 + 1\right).$$

The only catch now is that $\text{E}T_{\mathtt{sprt}}$ contains a problematic $\text{E} \ln T_H$ term. However, this can be bounded by $\text{E} \ln T_H \leq \ln \text{E}T_H$, using Jensen's inequality and the fact that ln is concave; see *e.g.*, [Ash72]. The rest follows from algebraic manipulation. □

Although this is a crude bound, it is interesting to note that it *scales* the same as T_{BEHW} and T_{STAB}. Moreover, this bound actually *beats* T_{BEHW} and T_{STAB} for small values of δ — but this advantage is slight, and only holds for high reliability levels.

Proposition 3 (i) $\mathrm{E}T_{\mathsf{S}}(C,\epsilon,\delta) < T_{BEHW}(C,\epsilon,\delta)$ *for* $\kappa \geq 3.5$ *and sufficiently small* δ.
(ii) $\mathrm{E}T_{\mathsf{S}}(C,\epsilon,\delta) < T_{STAB}(C,\epsilon,\delta)$ *for* $\kappa \geq \frac{2}{\sqrt{\epsilon}} \ln \frac{2}{\sqrt{\epsilon}}$ *and sufficiently small* δ.

Although this theoretical advantage is slight, we expect S to perform *much* better in practice than any bounds we can prove about its performance; *n.b.*, this is not a possibility for fixed-sample-size approaches. In fact, this advantage is readily demonstrated in empirical case studies [SG95]. For example, we tested S on the pac-learning problem ($X = \mathbb{R}^{10}, C = \mathsf{halfspaces}, \epsilon = 0.01, \delta = 0.05$); fixing a uniform distribution on $[-1,1]^n$ and a particular target concept, setting $\kappa = 3.14619$, and supplying S with a consistent $\mathsf{halfspace}$ hypothesizer. After 100 trials we obtained the results in Table 1, which show that S used an average training sample size that was about 5 *times* smaller than T_{STAB}, and 27 *times* smaller than T_{BEHW} ! Moreover, this average was only 3 times larger than the empirical "rule of thumb" that $\frac{w}{\epsilon}$ training examples are needed to achieve ϵ error, for a concept class defined by w free weights [BH89]. Not only do these results scale up well for harder problems (Figure 5), they are also *robust* to changes in the target concept, domain distribution, and concept class (with the same VCdimension) [SG95]. One reason for this advantage is that S's data-efficiency is determined by the *specific case at hand*, not the worst case situation — or, worse yet, by what we can *prove* about the worst case situation. However, not only does S automatically take advantage of "easy" situations, it will also take advantage of the *true* worst case convergence properties of C (*i.e.*, if bad concepts are eliminated much sooner than proven bounds, then S automatically stops sooner). So, in effect, S's behavior implicitly exploits the *optimal* worst case bounds, despite our inability to prove exactly what these bounds really are.

Although S is far more efficient than previous fixed-sample-size approaches in practice, the following lower bound shows that sequential learning can at best offer a constant (or possibly log) improvement in the number of training examples needed to pac-learn. Therefore, no *new* concept classes become pac-learnable simply by adopting a sequential over a fixed-sample-size approach.

Theorem 4 (Data complexity) *For any* $0 < \epsilon \leq \frac{1}{8}$, $0 < \delta \leq \frac{1}{683}$, *any concept class* C *with* $\mathrm{VC}(C) = d \geq 2$*: any learner that always observes an average number of training examples less than*

$$t_{avg}(C,\epsilon,\delta) = \max\left\{\tfrac{d-1}{480\epsilon},\ \tfrac{1-\delta}{4\epsilon}\right\}$$

cannot meet the pac(ϵ,δ)*-criterion for all* $c \in C$ *and* P.

Proof (Outline of $t_{avg} \geq \frac{d-1}{480\epsilon}$.) Fix an arbitrary learner L with stopping rule T. The basic idea is to use Markov's inequality to show that if ET is too small relative to t_{EHKV} then L must fail the pac(ϵ,δ)-criterion for some $c' \in C$. This involves generalizing the proof of [EHKV89, Theorem 1] to handle the fact that T might not terminate at the same time for every $c \in C$.

For ($X = \mathbb{R}^{10}, C = \mathsf{halfspaces}, \epsilon = 0.01, \delta = 0.05$):

Sufficient:	T_{BEHW}	$=$	$91,030,$
Improved:	T_{STAB}	$=$	$15,981,$
Folklore:	T_{thumb}	$\approx$	$1,100,$
Necessary:	t_{EHKV}	$=$	$32.$
After 100 trials, Procedure S used:			
	avg T_{S}	$=$	$\mathbf{3,402,}$
	$\max T_{\mathsf{S}}$	$=$	$5,155,$
	$\min T_{\mathsf{S}}$	$=$	$2,267.$

Table 1: A direct comparison of training sample sizes for the pac-learning problem ($\mathbb{R}^{10}, \mathsf{halfspaces}, \epsilon = 0.01, \delta = 0.05$).

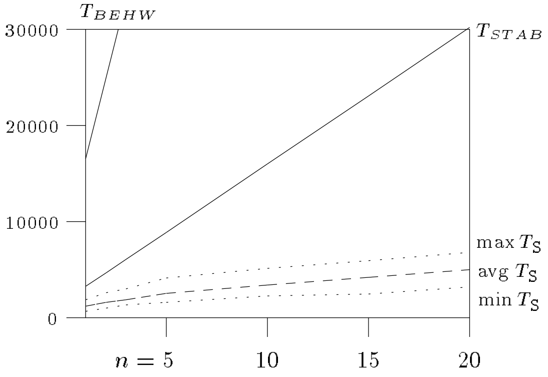

Figure 5: Scaling in input dimension n. Number of training examples observed for ($\mathbb{R}^n$, $\mathsf{halfspaces}$, $\epsilon = 0.01$, $\delta = 0.05$) with $n = 1, 2, 3, 5, 10, 15, 20$. (Results of 100 runs each.)

Following [EHKV89], we define a specific domain distribution P on a set of d objects $\{x_1, ..., x_d\}$ shattered by C: let $\mathrm{P}\{x_1\} = 1 - 8\epsilon$ and $\mathrm{P}\{x_i\} = \frac{8\epsilon}{d-1}$ for $2 \leq i \leq d$. Let the r.v. $U : X^\infty \to \mathbb{N}$ indicate the first time that half of the objects $\{x_2, ..., x_d\}$ appear in an observation sequence $\mathbf{x} \in X^\infty$. Let H^t denote L's hypothesis after t training examples.

(1) For any P, c and t we have the following inequality

$$\mathrm{P}\{d_{\mathrm{P}}(H^T,c) > \epsilon\} \quad \geq \quad \mathrm{P}\{\, d_{\mathrm{P}}(H^T,c) > \epsilon \,\big|\, Tc < U\,\} \times (\,\mathrm{P}\{Tc \leq t\} + \mathrm{P}\{U > t\} - 1\,).$$

We seek lower bounds on each of these terms.

(2) For any P, t, and $k > 1$, by Markov's inequality we know that if $\mathrm{E}T \leq \frac{t}{k}$ then $\mathrm{P}\{T \leq t\} \geq 1 - \frac{1}{k}$.

(3) Given P defined as above, [EHKV89, Lemma 3] shows that $\mathrm{P}\{U > \frac{d-1}{32\epsilon}\} \geq 1 - e^{-1/12} > \frac{1}{13}$.

(4) Finally, for any learner L it can be shown that, given P defined as above, there must be some $c' \in C$ for which $\mathrm{P}\{\, d_{\mathrm{P}}(H^T,c') > \epsilon \,\big|\, Tc' < U\,\} \geq \frac{1}{7}$. (This involves generalizing the proof of [EHKV89, Lemma 2]; see [Sch95] for complete details.)

Combining (1)–(4) shows that, for any $k > 1$, if $\mathrm{E}Tc \leq \frac{d-1}{32k\epsilon}$ for all $c \in C$, then there must be some $c' \in C$ for which $\mathrm{P}\{d_{\mathrm{P}}(H^T, c') > \epsilon\} \geq \frac{1}{7}\left[1 - \frac{1}{k} + \frac{1}{13} - 1\right] = \frac{1}{7}\left[\frac{1}{13} - \frac{1}{k}\right]$. Choosing $k = 15$ yields the result. □

2.1 "Mistake-bounded to pac" conversion

Before leaving the distribution-free model, we briefly consider the special case of finite concept classes, and obtain a somewhat stronger result in this case. Littlestone has observed that a concept from a finite class can always be learned while making a finite number of mistakes, in an on-line model where the learner produces a hypothesis after each example and tests it on the next [Lit88]. In later work [Lit89] he showed how a hypothesizer H with a small mistake bound could be converted into a data-efficient pac-learner. Littlestone develops a "two phase" conversion procedure Li that, given a hypothesizer H with mistake bound M, uses a fixed sample size of $T_{\mathtt{Li}} = \frac{4}{\epsilon}\left(M + 8\ln(M+2) + 12\ln\frac{2}{\delta} - \frac{1}{2}\right)$.

Here we consider a *sequential* approach to this problem. First, we note that S can be applied to "mistake-bounded to pac" conversion "as is." However, by modifying S to return a hypothesis once the mistake bound has been reached, setting $\kappa = 3.14619$ (so that $\kappa = \frac{\kappa}{\kappa - 1 - \ln\kappa}$), and testing each hypothesis h_i with $\delta_i = \frac{\delta}{M}$, we obtain a correct conversion procedure Smb that is provably more efficient than Li.

Proposition 5 *For any $\epsilon > 0$, $\delta > 0$, any finite concept class C: using any hypothesizer H with mistake bound M, Smb pac(ϵ, δ)-learns C with an average training sample size of at most*

$$\mathrm{E}T_{\mathit{Smb}} \quad \leq \quad \frac{3.14619}{\epsilon}\left(M + \ln M + \ln\frac{1}{\delta} + 1\right).$$

Proof (Sketch) As with S, we know Smb eventually accepts any $\frac{\epsilon}{\kappa}$-good hypothesis wp1. Thus, we can bound Smb's stopping time by $T_{\mathtt{Smb}} \leq T_H(\frac{\epsilon}{\kappa}) + T_{\mathtt{sprt}}(\frac{\epsilon}{\kappa}, \epsilon, \delta_M)$, where $T_{\mathtt{sprt}}$ is the time it takes to accept an $\frac{\epsilon}{\kappa}$-good hypothesis, and T_H is the time it takes for H to produce such a hypothesis. So $\mathrm{E}T_{\mathtt{Smb}} \leq \mathrm{E}T_H + \mathrm{E}T_{\mathtt{sprt}}$. Clearly, $\mathrm{E}T_H(\frac{\epsilon}{\kappa}) \leq \frac{\kappa M}{\epsilon}$ since the expected time for an $\frac{\epsilon}{\kappa}$-bad hypothesis to misclassify a training example is less than $\frac{\kappa}{\epsilon}$, and there can be at most M such hypotheses. Also, Lemma 11 shows that $\mathrm{E}T_{\mathtt{sprt}}(\frac{\epsilon}{\kappa}, \epsilon, \frac{\delta}{M}) \leq \left(\frac{\kappa}{\kappa - 1 - \ln\kappa}\right)\frac{1}{\epsilon}\left(\ln\frac{M}{\delta} + 1\right)$. The result then follows by choosing $\kappa = 3.14619$. □

This bound on $\mathrm{E}T_{\mathtt{Smb}}$ is uniformly smaller than $T_{\mathtt{Li}}$ by a small constant factor. However, as before, we expect Smb to perform much better in practice than any bounds we can prove about its performance. This too is readily demonstrated in empirical case studies. For example, we tested Smb on the pac-learning problem ($X = \{0,1\}^{30}, C = \mathsf{halfspaces}, \epsilon, \delta = 0.05$) for various values of ϵ; fixing a particular domain distribution and target concept, and supplying Smb with a hypothesizer H = WINNOW which has a good mistake bound for this problem [Lit88]. After 200 trials (at each error level) we obtained the results in Figure 6: Smb observed an average number of training examples that was always 15 *times* smaller than the upper bound in Proposition 5, and 30 *times* smaller than $T_{\mathtt{Li}}$! In fact, Smb's data-efficiency appears to scale better than $T_{\mathtt{Li}}$ (and our bound) as ϵ becomes small. This is a significant practical savings, achieved without substantial additional computation.

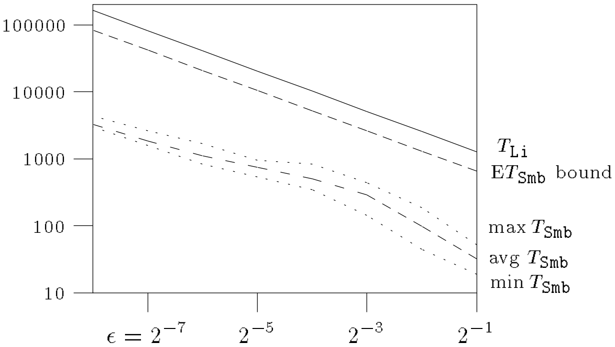

Figure 6: Scaling in error level ϵ. Number of training examples observed for ($X = \{0,1\}^{30}$, $\mathsf{halfspaces}$, ϵ, $\delta = 0.05$) with $\epsilon = 2^{-1}, ..., 2^{-8}$. (Result of 200 runs each; log-log plot.)

3 Distribution-*specific* pac-learning

We now consider the distribution-specific model, where the learner *knows* P and attempts to identify an unknown target concept c from some specified class C. This problem was thoroughly studied by Benedek and Itai [BI88a], who developed a simple (collect; find) learning procedure BI for pac-learning concept spaces (C, P). Their procedure first finds an $\frac{\epsilon}{2}$-cover A of the space (a set of concepts $A = \{h_1, ..., h_N\}$ such that for every $c \in C$ there is at least one $h_i \in A$ where $d_{\mathrm{P}}(c, h_i) \leq \frac{\epsilon}{2}$), and then collects a sufficient number of training examples to estimate the errors of all cover-concepts to within $\frac{\epsilon}{2}$, with probability at least $1 - \delta$. Choosing the cover-concept with minimum observed error rate then satisfies the pac(ϵ, δ)-criterion; see Figure 7. Benedek and Itai show that $T_{\mathtt{BI}}(C, \mathrm{P}, \epsilon, \delta) = \frac{32}{\epsilon}(\ln N_{\frac{\epsilon}{2}} + \ln\frac{1}{\delta})$ examples are sufficient to pac-learn (C, P), where $N_{\frac{\epsilon}{2}}$ is the size of the smallest $\frac{\epsilon}{2}$-cover of (C, P). They also show that *no* learner can observe fewer than $t_{BI}(C, \mathrm{P}, \epsilon, \delta) = \log_2[N_{2\epsilon}(1 - \delta)]$ training examples and still meet the pac(ϵ, δ)-criterion for every c in C.

Here we consider a *sequential* approach to this problem, which also exploits the existence of a small $\frac{\epsilon}{2}$-cover of the concept space. However, rather than collect a *fixed* size training sample to estimate errors, we test each cover-concept *sequentially* (in parallel) and accept the first one that proves to have sufficiently small error; see Procedure Scov in Figure 8. This procedure correctly pac-learns any concept space that has a finite $\frac{\epsilon}{2}$-cover, just as BI, but uses an average training sample size that is about 5 *times* smaller than $T_{\mathtt{BI}}$.

```
Procedure BI $(C, \mathrm{P}, \epsilon, \delta)$
CONSTRUCT an $\frac{\epsilon}{2}$-cover $A$ of size $N_{\frac{\epsilon}{2}}$.
COLLECT $T_{\mathtt{BI}}(C, \mathrm{P}, \epsilon, \delta) = \frac{32}{\epsilon}(\ln N_{\frac{\epsilon}{2}} + \ln \frac{1}{\delta})$ examples.
RETURN the hypothesis $h \in A$ with minimum error.
```

Figure 7: Procedure BI

```
Procedure Scov $(C, \mathrm{P}, \epsilon, \delta)$
CONSTRUCT an $\frac{\epsilon}{2}$-cover $A$ of size $N_{\frac{\epsilon}{2}}$.
SEQUENTIALLY observe training examples:
    TEST the error of each $h_i \in A$ by calling
        sprt($h_i(x) \neq c(x)$, $\frac{\epsilon}{2}$, $\epsilon$, $\delta/N_{\frac{\epsilon}{2}}$, 0).
    RETURN the first $h_i \in A$ accepted by sprt.
```

Figure 8: Procedure Scov

Theorem 6 (Correctness) *For any $\epsilon > 0$, $\delta > 0$, and any concept space (C, P) with $N_{\frac{\epsilon}{2}} < \infty$: `Scov` meets the pac$(\epsilon, \delta)$-criterion for any c in C.*

Proof Since `Scov` chooses hypotheses from A, an $\frac{\epsilon}{2}$-cover of (C, P), there must be at least one $\frac{\epsilon}{2}$-good $h \in A$, and `Scov` eventually accepts such a hypothesis wp1 (Lemma 9). Correctness then follows from the fact that `Scov` mistakenly accepts an ϵ-bad hypothesis with probability at most $\sum_{h \in A} \delta / N_{\frac{\epsilon}{2}} = \delta$. □

Theorem 7 (Data efficiency) *For any $\epsilon > 0$, $\delta > 0$, and any concept space (C, P) with $N_{\frac{\epsilon}{2}} < \infty$: `Scov` observes an average training sample size of at most*

$$\mathrm{E}T_{\mathtt{Scov}}(C, \mathrm{P}, \epsilon, \delta) \quad \leq \quad \tfrac{6.5178}{\epsilon}\left(\ln N_{\frac{\epsilon}{2}} + \ln \tfrac{1}{\delta} + 1\right).$$

Proof Since some $h \in A$ is guaranteed to be $\frac{\epsilon}{2}$-good, and `Scov` eventually accepts any such hypothesis wp1 (Lemma 9), we have $T_{\mathtt{Scov}}(\epsilon, \delta) \leq T_{\mathtt{sprt}(\cdot,\, \frac{\epsilon}{2},\, \epsilon,\, \delta/N_{\frac{\epsilon}{2}},\, 0)}$. Applying Lemma 11 immediately yields the result. □

Although `Scov` is strictly more efficient than `BI` (while solving the *exact same* pac-learning problem), the following lower bound shows that no *new* concept spaces become pac-learnable simply by adopting a sequential over a fixed-sample-size approach.

Theorem 8 (Data complexity) *For any $\epsilon > 0$, $\delta > 0$, and any concept space (C, P): any learner that observes an average number of training examples less than*

$$t_{avg}(C, \mathrm{P}, \epsilon, \delta) \quad = \quad \tfrac{1}{2}\log_2[N_{2\epsilon}(\tfrac{1}{2} - \delta)]$$

fails to meet the pac(ϵ, δ)-criterion for some $c' \in C$.

Proof (Sketch) Fix an arbitrary learner L with stopping rule T. As in Theorem 4, we use Markov's inequality to show that if $\mathrm{E}T$ is too small relative to t_{BI} then L must fail to meet the pac(ϵ, δ)-criterion for some $c \in C$. Let H^t denote L's hypothesis after t training examples.

(1) For any c and t we have the following inequality

$$\mathrm{P}\{d_{\mathrm{P}}(H^T, c) > \epsilon\} \quad \geq \quad 1 - \mathrm{P}\{d_{\mathrm{P}}(H^t, c) \leq \epsilon\} - \mathrm{P}\{T > t\}.$$

Thus, we seek upper bounds on each of these terms.

(2) For any t, if $\mathrm{E}T \leq \frac{t}{2}$ then $\mathrm{P}\{T > t\} \leq \frac{1}{2}$ by Markov's inequality.

(3) For any t, [BI88a, Lemma 5] shows that any hypothesizer H is forced to obtain $\mathrm{P}\{d_{\mathrm{P}}(H^t, c') \leq \epsilon\} \leq \frac{2^t}{N_{2\epsilon}}$ for some $c' \in C$.

Combining (1)–(3) shows that for any t, if $\mathrm{E}T \leq \frac{t}{2}$ then there is a c' for which $\mathrm{P}\{d_{\mathrm{P}}(H^T, c') > \epsilon\} \geq 1 - \frac{2^t}{N_{2\epsilon}} - \frac{1}{2}$. Choosing $t = \log_2[N_{2\epsilon}(\frac{1}{2} - \delta)]$ finishes the proof. □

A Properties of sprt

Lemma 9 *A call to **sprt**$(h(x) \neq c(x),\ \frac{\epsilon}{\kappa},\ \epsilon,\ \delta,\ 0)$ eventually accepts any $\frac{\epsilon}{\kappa}$-good hypothesis h wp1.*

Proof First, since $\delta_{rej} = 0$, `sprt` never rejects a hypothesis. To show `sprt` eventually accepts any $\frac{\epsilon}{\kappa}$-good hypothesis h, we use the fact that $S_t(\mathbf{x}^t)$ is an i.i.d. sum $S_t(\mathbf{x}^t) = \sum_{x_i \in \mathbf{x}^t} Z(x_i)$, where

$$Z(x_i) = \begin{cases} \ln \frac{1-\epsilon/\kappa}{1-\epsilon}, & \phi(x_i) = 0, \\ -\ln \kappa, & \phi(x_i) = 1. \end{cases}$$

Let $p = \mathrm{P}\{\phi(x) = 1\}$. Since $p \leq \frac{\epsilon}{\kappa}$ by assumption, we have $\mathrm{E}Z > 0$ by Claim 10 below. Therefore, we get $S_t \to \infty$ wp1, since $S_t/t \to \mathrm{E}Z$ wp1 by the law of large numbers [Ash72]. Thus, S_t eventually exceeds the $\ln 1/\delta_{acc}$ threshold wp1, for any $\delta_{acc} > 0$. □

Claim 10 *For $\epsilon > 0$, $\kappa > 1$, given Z and p defined as above: if $p \leq \frac{\epsilon}{\kappa}$ then $\mathrm{E}Z \geq \left(\frac{\kappa - 1 - \ln \kappa}{\kappa}\right)\epsilon > 0$.*

Proof By definition we have $\mathrm{E}Z = (1-p)\frac{1-\epsilon/\kappa}{1-\epsilon} - p\ln\kappa$. Since $\mathrm{E}Z$ is increasing for decreasing p, it suffices to verify the lower bound for $p = \frac{\epsilon}{\kappa}$. This can be done by taking derivatives of $\mathrm{E}Z$ with respect to ϵ [Sch95]. □

Lemma 11 *For $0 < \epsilon < 1 - e^{-1}$, $\delta > 0$, $\kappa > 1$: given a Boolean r.v. $\phi(x)$ such that $\mathrm{P}\{\phi(x) = 1\} \leq \frac{\epsilon}{\kappa}$,*

$$\mathrm{E}T_{\boldsymbol{sprt}(\phi(x), \frac{\epsilon}{\kappa}, \epsilon, \delta, 0)} \quad \leq \quad \left(\frac{\kappa}{\kappa - 1 - \ln \kappa}\right)\frac{1}{\epsilon}\left(\ln \frac{1}{\delta} + 1\right)$$

Proof Recall the definition $S_t(\mathbf{x}^t) = \sum_{x_i \in \mathbf{x}^t} Z(x_i)$ given above. Since S_t is an i.i.d. sum, Wald's identity gives $\mathrm{E}S_T = \mathrm{E}Z\,\mathrm{E}T$ for any stopping rule T [Wal47, Shi78]. Thus, $\mathrm{E}T = \mathrm{E}S_T/\mathrm{E}Z$. We know that $S_T < \ln\frac{1}{\delta} + \ln\frac{1-\epsilon/\kappa}{1-\epsilon}$ since the sum at termination cannot exceed the decision threshold plus one increment, so we get $\mathrm{E}T < \frac{1}{\mathrm{E}Z}(\ln\frac{1}{\delta} + 1)$ (since $\ln\frac{1-\epsilon/\kappa}{1-\epsilon} \leq 1$ for $\epsilon \leq 1 - e^{-1}$). This inequality holds for any value of $p = \mathrm{P}\{\phi(x) = 1\}$. Under the assumption that $p \leq \frac{\epsilon}{\kappa}$, Claim 10 above provides a lower bound on $\mathrm{E}Z$ which gives the result. □

B Additional lemmas

Lemma 12 *For any class C, $\mathrm{VC}(C) < \infty$, and $\epsilon > 0$: every ϵ-bad $c \in C$ is eventually eliminated, wp1.*

Proof Let E_t be the event that all ϵ-bad concepts have been eliminated after t training examples. From [STAB93] we have that for all $\delta > 0$ there is some t for which $\mathrm{P}E_t \geq 1 - \delta$, and hence $\mathrm{P}E_t \uparrow 1$. We are interested in the event $E_\infty = \bigcup_{t=1}^{\infty} E_t$. But since, in fact, $E_t \uparrow E_\infty$, we must have $\mathrm{P}E_t \uparrow \mathrm{P}E_\infty$ and hence $\mathrm{P}E_\infty = 1$. □

Lemma 13 *For any concept class C, $\mathrm{VC}(C) < \infty$: all ϵ-bad $c \in C$ are eliminated in expected time*

$$\mathrm{E}T_C(\epsilon) \leq \tfrac{1}{\epsilon(1-\sqrt{\epsilon})}\left(2d\ln\tfrac{6}{\epsilon} + \ln 2 + 1\right).$$

Proof We have $\mathrm{P}\{T_C(\epsilon) > T_{STAB}(C, \epsilon, \delta)\} \leq \delta$ for all $\delta > 0$ from [STAB93]. Assume, pessimistically, that T_C is a random variable that makes this an equality, *i.e.*, $\mathrm{P}\{T_C > T_{STAB}\} = \delta$ for all $\delta > 0$. Now, consider a linear transformation of T_C,

$$V = \epsilon(1 - \sqrt{\epsilon})T_C - 2d\ln\tfrac{6}{\epsilon} - \ln 2.$$

Notice that $V > \ln\frac{1}{\delta}$ iff $T_C > T_{STAB}$, and hence $\mathrm{P}\{V > \ln\frac{1}{\delta}\} = \delta$ for all $\delta > 0$. This shows that $V \sim$ *exponential*(1), and hence $\mathrm{E}V = 1$. Finally, since $T_C = \frac{1}{\epsilon(1-\sqrt{\epsilon})}\left(2d\ln\frac{6}{\epsilon} + \ln 2 + V\right)$, taking expectations gives the result. □

References

[AKA91] D. W. Aha, D. Kibler, and M. K. Albert. Instance-based learning algorithms. *Machine Learning*, 6(1):37–66, 1991.

[Ash72] R. B. Ash. *Real Analysis and Probability.* Academic Press, San Diego, 1972.

[Bau90] E. Baum. The perceptron algorithm is fast for nonmalicious distributions. *Neural Computation*, 2:248–260, 1990.

[BEHW89] A. Blumer, A. Ehrenfeucht, D. Haussler, and M. K. Warmuth. Learnability and the Vapnik-Chervonenkis dimension. *Journal of the ACM*, 36(4):929–965, 1989.

[BH89] E. B. Baum and D. Haussler. What size net gives valid generalization? *Neural Computation*, 1:151–160, 1989.

[BI88a] G. Benedek and A. Itai. Learnability by fixed distributions. In *Proceedings COLT-88*, pages 80–90, 1988.

[BI88b] G. Benedek and A. Itai. Nonuniform learnability. In *Proceedings ICALP-88*, pages 82–92, 1988.

[BW91] P. L. Bartlett and R. C. Williamson. Investigating the distributional assumptions of the pac learning model. In *Proceedings COLT-91*, pages 24–32, 1991.

[EHKV89] A. Ehrenfeucht, D. Haussler, M. Kearns, and L. Valiant. A general lower bound on the number of examples needed for learning. *Information and Computation*, 82:247–261, 1989.

[HLL92] D. Helmbold, N. Littlestone, and P. Long. Apple tasting and nearly one-sided learning. In *Proceedings FOCS-92*, pages 493–502, 1992.

[Koi94] P. Koiran. Efficient learning of continuous neural networks. In *Proceedings COLT-94*, pages 348–355, 1994.

[Kul91] S. Kulkarni. *Problems of Computational and Information Complexity in Machine Vision and Learning.* PhD thesis, MIT, EECS, 1991.

[KV89] M. J. Kearns and L. G. Valiant. Cryptographic limitations on learning Boolean formulae and finite automata. In *Proceedings STOC-89*, pages 433–444, 1989.

[Lit88] N. Littlestone. Learning quickly when irrelevant attributes abound: A new linear threshold algorithm. *Machine Learning*, 2:285–318, 1988.

[Lit89] N. Littlestone. From online to batch learning. In *Proceedings COLT-89*, pages 269–284, 1989.

[LMR91] N. Linial, Y. Mansour, and R. L. Rivest. Results on learnability and the Vapnik-Chervonenkis dimension. *Information and Computation*, 90:33–49, 1991.

[LW89] N. Littlestone and M. Warmuth. The weighted majority algorithm. In *Proceedings FOCS-89*, pages 256–261, 1989.

[Sch92] R. E. Schapire. *The Design and Analysis of Efficient Learning Algorithms.* MIT Press, Cambridge, MA, 1992.

[Sch95] D. Schuurmans. *Effective Classification Learning.* PhD thesis, University of Toronto, Computer Science, 1995. (Forthcoming).

[SG95] D. Schuurmans and R. Greiner. Practical PAC learning. In *Proceedings IJCAI-95*, 1995.

[Shi78] A. N. Shiryayev. *Optimal Stopping Rules.* Springer-Verlag, New York, 1978.

[STAB93] J. Shawe-Taylor, M. Anthony, and N. L. Biggs. Bounding sample size with the Vapnik-Chervonenkis dimension. *Discrete Applied Mathematics*, 42:65–73, 1993.

[Val84] L. G. Valiant. A theory of the learnable. *Communications of the ACM*, 27(11):1134–1142, 1984.

[VC71] V. N. Vapnik and A. Ya. Chervonenkis. On the uniform convergence of relative frequencies of events to their probabilities. *Theory of Probability and its Applications*, 16(2):264–280, 1971.

[Wal47] A. Wald. *Sequential Analysis.* John Wiley & Sons, New York, 1947.