Fast (Distribution Specific) Learning

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

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

Abstract

PAC-learning results are often criticized for demanding impractically large training samples. The common wisdom is that these large samples follow from the worst case nature of the analysis, and therefore PAC-learning, though desirable, must not be a practical goal. We however consider an alternative view: perhaps these large sample sizes are due to the presumed learning strategies which make inefficient use of the available training data.

To demonstrate this, we consider *sequential* learning strategies that autonomously decide when to stop training based on observing training examples as they arrive. We show that for distribution specific learning these algorithms require far fewer training examples (on average) than existing fixed sample size approaches, and are able to learn with *certainty* not just high probability. In fact, a simple sequential strategy is *optimally* efficient in many cases.

1 Introduction

We consider the basic concept learning problem that has long been of central interest in applied and in theoretical machine learning research. Theoretical analyses of this task have been largely motivated by Valiant's PAC-learning framework which introduced the idea of learning an approximation with guaranteed accuracy and reliability (Valiant, 1984). Valiant originally investigated the necessary (and sufficient) computational and data resources needed to achieve these guarantees, in the worst case over all possible domain distributions and target concepts drawn from some fixed class C. Subsequent research (Blumer et al., 1989; Ehrenfeucht et al., 1989) has determined (up to constants and log factors) the rate at which the necessary and sufficient training sample sizes scale up in terms of accuracy ϵ, reliability δ, and the complexity of the concept class C (as measured by its VCdimension (Vapnik and Chervonenkis, 1971)).

Real world issue: While machine learning practitioners rarely criticize this idea of achieving guaranteed accuracy and reliability levels in an all-encompassing worst case manner, they typically find the sample sizes proven sufficient for PAC-learning are far too large to be practical for natural choices of ϵ, δ, and C. The predominant folk wisdom is that this impracticality is due to the worst case nature of the analysis, and therefore PAC-learning must not be a practical goal. However, this view may not be entirely accurate, since:

1. The established bounds incorporate approximations that go well beyond considering just the worst case distribution and target concept (*i.e.*, the constants in the current bounds are not tight), and

2. PAC analyses typically consider a simplistic learning strategy — fix a sample size, collect the data, and *then* inspect the data to choose a consistent hypothesis — that may not be making the most efficient use of the available training examples.

We focus on the second alternative: Can worst case PAC-learning be efficiently achieved by other learning strategies that go beyond the simple "collect, filter, select" approach? Following (Wald, 1947), we consider *sequential* learning strategies that autonomously decide when to stop their own training, based on observing the labeled training examples one at a time in succession. The idea is to hopefully reduce the number of training examples required in practice over the standard fixed sample size learning techniques.

This work is motivated by the observation that in many real-world applications of machine learning it is *training data* and not computation time is the critical resource. So we consider trading off computational for data efficiency: we are willing to incur a slight increase in computational cost in order to reduce the number of training examples required for accurate and reliable learning in practice.

Overview: This paper considers the distribution *specific* PAC-learning model (Benedek and Itai, 1988; Kulkarni, 1991), where the learner knows the domain distribution *a priori* but not the underlying target concept; as introduced in Section 2. Section 3 then uses a simple example to introduce a basic sequential learning strategy that produces accurate hypotheses with *certainty*, not just high probability $1-\delta$, while requiring *far* fewer training examples (on average) than existing fixed sample size learners. Section 4 then examines the general capabilities of sequential learners in the distribution specific setting, and presents a simple strategy that learns with *optimal* data efficiency for a certain case. Finally, Section 5 considers *truncated* sequential learners that are allowed to observe at most a bounded number of training examples. In every instance we observe that sequential learners provide a substantial improvement on the data efficiency of existing fixed-sample-size learning techniques. Section 6 concludes the paper with directions for future research.

2 Distribution specific PAC-learning

The basic learning problem we consider is the standard task of learning an unknown target concept from examples. Formally, a concept c is just a subset of the domain of objects X, and a labelled *example* consists of a domain object x with its classification according to c: $\langle x, 1_c(x)\rangle$, written cx for short.[1] Given a sequence of labeled training examples $c\mathbf{x} = \langle cx_1, cx_2, ..., cx_t\rangle$, the learner must produce a hypothesis $h \subset X$ that closely approximates c.

Following Valiant, we assume both training and test examples are drawn independently according to a fixed domain distribution P on X, and classified according to a fixed but

[1] Of course, $1_c(x) = 1$ iff $x \in c$, and $1_c(x) = 0$ otherwise.

Procedure BI $(C, \mathrm{P}, \epsilon, \delta)$

- Find an $\epsilon/2$-cover H of (C, P) with size $N_{\epsilon/2}$.
- Collect $T_{BI} = \frac{32}{\epsilon} \ln \frac{N_{\epsilon/2}}{\delta}$ training examples.
- Return the hypothesis $h \in H$ with minimum observed error.

Figure 1: Procedure BI

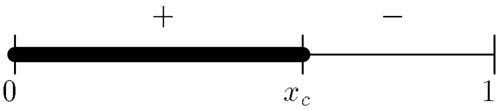

Figure 2: Left half-space concept c defined by point x_c.

unknown target concept c which belongs to some known class C. In the distribution specific variant of Valiant's learning model (Benedek and Itai, 1988; Kulkarni, 1991) both the concept class C and the domain distribution P are assumed to be known *a priori*. Here it is natural to think of C and P as comprising a concept *space* (C, P) where the distance between two concepts is given by the natural pseudo-metric $d_{\mathrm{P}}(c_1, c_2) = \mathrm{P}(c_1 x \neq c_2 x)$.

Definition 1 (PAC-learning) *For a given reliability and accuracy levels,* $1 - \delta$ *and* $1 - \epsilon$*, we say*
(1) *a learner "*PAC-learns *c" if given random examples generated by* P *and labeled by c it produces a hypothesis h such that* $\mathrm{P}(hx \neq cx) \leq \epsilon$ *with probability at least* $1 - \delta$*; and*
(2) *a learner "*PAC-learns *a concept* space (C, P)*" if it* PAC*-learns each c in C.*
The sample efficiency *of a learner L, written T^L, is measured by the number of training examples it uses in the worst case over all c in C.*

Benedek and Itai have developed the PAC-learning strategy BI, shown in Figure 1, that exploits the existence of a small *ϵ-cover* of the concept space: An ϵ-cover of (C, P) is a set of concepts $H = \{h_1, h_2, ..., h_N\}$ such that every c in C is within ϵ of at least one h in H; N_ϵ denotes the size of the smallest ϵ-cover. The number of training examples used by BI is given by

$$T^{BI} = \frac{32}{\epsilon} \ln \frac{N_{\epsilon/2}}{\delta} \quad \text{(Benedek and Itai, 1988).} \tag{1}$$

Below we compare the sample efficiency of this fixed sample size learning technique with the sequential learning approach.

3 A simple example

We use the following simple case study to introduce a basic sequential learning strategy, and then to compare the relative sample efficiencies of the fixed sample size and sequential approaches.

Here we consider the problem of learning a target subinterval of the unit interval given uniform random examples. Formally, let X_1 be the unit interval $[0, 1]$, P_1 the uniform distribution, and C_1 the class of half intervals $C_1 = \{[0, x_c] : 0 < x_c < 1\}$. So a target

Procedure S_1 (ϵ)

- Observe random training examples until $x_r - x_\ell \leq 2\epsilon$ (the *stopping condition*).
- Return the halfspace defined by the midpoint $x_m = (x_r - x_\ell)/2$, *viz.*, $[0, x_m)$.

Figure 3: Procedure S_1

concept $c \in C_1$ is defined by an endpoint $x_c \in [0,1]$; see Figure 2. It is not hard to see that there is a rather obvious sequential strategy for learning an unknown half interval: simply keep an "uncertainty interval" around the unknown endpoint, and stop as soon as this interval gets small enough. That is, for any finite sequence of example points $\mathbf{x} = \{x_1, ..., x_t\}$, define $x_\ell = \max\{x_i \in \mathbf{x} \,|\, cx_i = 1\}$ to be the largest positive example (0, if none exists), and $x_r = \min\{x_i \in \mathbf{x} \,|\, cx_i = 0\}$ to be the smallest negative example (1, if none exists); then use these to define the simple sequential learning procedure S_1 shown in Figure 3. Procedure S_1 is clearly correct since any hypothesis it returns is guaranteed to be ϵ-accurate by construction. Furthermore S_1 terminates with probability 1; see Proposition 1 below. So in fact S_1 learns with *certainty* not just high probability, as it returns ϵ-accurate hypotheses with probability 1 instead of just probability $1 - \delta$.

How efficiently does S_1 learn? To determine its sample efficiency, notice that the number of training examples S_1 observes is a *random variable* rather than just a fixed number. In addition there is no upper bound on the number of examples that might be observed before the stopping criterion is met, so characterizing S_1's sample efficiency simply by an upper bound on training sample size gives a vacuous result. However, the *expected* number of training examples S_1 observes is actually quite small:

Proposition 1 *For any target concept c with endpoint in $[\epsilon, 1-\epsilon]$*

$$T^{S_1} \sim \text{ negative-binomial } (p = 2\epsilon,\ k = 2),$$

furthermore S_1 stops even faster for concepts with endpoints nearer 0 or 1. As a direct consequence, for any c in C_1 we have $\mathrm{P}(T^{S_1} < \infty) = 1$ and

$$\mathrm{E}\, T^{S_1} \leq \tfrac{1}{\epsilon}. \tag{2}$$

As S_1 requires at most $\frac{1}{\epsilon}$ training examples *on average* to return an ϵ-approximation to the unknown target concept *with certainty*, it appears to be a quite efficient learning algorithm. In fact, Theorem 2 below proves that S_1 has the *optimal* expected sample size, among all certain learners for this space!

To compare the relative sample efficiencies of the sequential and fixed sample size approaches, recall that the sample size used by BI_1 is determined by the size of the smallest $\epsilon/2$ cover of the concept space. We can derive BI_1's sample efficiency using (1) and the fact that $N_{\epsilon/2}(C_1, \mathrm{P}_1)$ is clearly at least $2/\epsilon$:

$$T^{BI_1} = \tfrac{32}{\epsilon} \ln \tfrac{2}{\epsilon\delta}. \tag{3}$$

This sample size (3) compares quite poorly to S_1's expected sample efficiency (2); by a factor of $32 \ln \frac{2}{\epsilon\delta}$. For example, setting $\epsilon = 0.05$ and $\delta = 0.05$, we get sample efficiencies $\mathrm{E}T^{S_1} = 20$

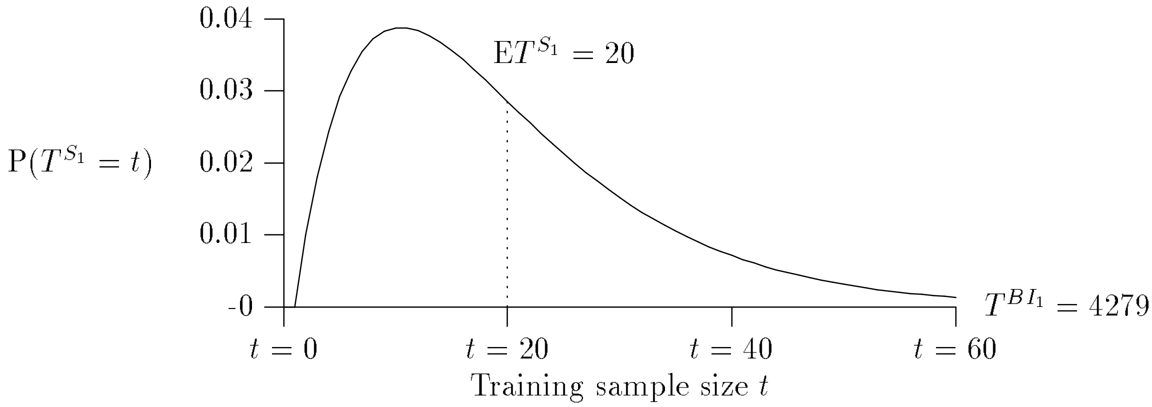

Figure 4: Comparing T^{S_1} and T^{BI_1} for $\epsilon = 0.05$ and $\delta = 0.05$.

and $ET^{BI_1} = T^{BI_1} = 4279$. Worse for BI_1, it would require an infinite sample size to learn with $\delta = 0$. So even though S_1 is solving a harder learning problem that BI_1, as it is learning with a higher degree of reliability, S_1 actually requires orders of magnitude *fewer* training examples on average.

Of course, we are comparing a fixed number T^{BI_1} to a *distribution* of sample sizes T^{S_1}. What is the best way to compare these two quantities? There is no one best answer here. As one natural measure of a sequential learner's data efficiency is given by its *expected* sample size, one reasonable way to compare fixed and sequential strategies is to compare their fixed versus expected sample sizes; as shown above. Alternatively, one could instead consider the probability that the sequential sample size exceeds the fixed size: From Proposition 1 one can use Chernoff bounds to show

$$P(T^S_\epsilon > t) \;\leq\; e^{-\frac{2}{3}t\epsilon(1-\frac{1}{t\epsilon})^2}, \quad \text{for } t \geq \tfrac{1}{\epsilon}. \tag{4}$$

Combining (4) with (3) shows that it is *extremely* unlikely that S_1 will ever use more examples than BI_1:

Proposition 2 $P(T^{S_1} \;>\; T^{BI_1}) < \left(\frac{\epsilon\delta}{2}\right)^{20}$, *for* ϵ, $\delta < \sqrt{\frac{2}{\epsilon}}$.

So for example choosing $\epsilon = 0.05$ and $\delta = 0.05$ as before we get that the probability S_1 ever exceeds BI_1's sample size is less than 800^{-20}, which is very small indeed! The overall situation for these values of ϵ and δ is illustrated in Figure 4.

So clearly S_1 is orders of magnitude more efficient than BI_1. There are two reasons for this overwhelming advantage: Benedek and Itai's sample size bound (1) incorporates a number of crude approximations and can almost certainly be improved a great deal. However, note that any real application of this technique in practice requires us to use *some* provably sufficient sample size bound; (1) just happens to be the best available. However, a second reason for this advantage is that the sequential approach seems to be intrinsically more efficient here, *cf.* Section 5.

Procedure S $(C, \mathrm{P}, \epsilon)$

- Sequentially observe examples until there exists a single $h \in 2^X$ (not necessarily in C) within ϵ of each consistent c remaining in C.
- Return h.

Figure 5: Procedure S

4 Learning with certainty: General theory

We briefly investigate the difficulty of learning accurate approximations *with certainty* in a general setting. We are considering a variation of the standard PAC-learning model where we want an absolute guarantee that the learner returns an accurate hypothesis, without fail; *i.e.*, returning ϵ-accurate hypotheses with probability 1, not just probability $1-\delta$. Notice first that this is impossible for any fixed sample size learner: if C contains two concepts c_1, c_2 such that $\epsilon < d_{\mathrm{P}}(c_1, c_2) < 1$, then for any finite sample size t we have $\mathrm{P}(c_1\mathbf{x}^t = c_2\mathbf{x}^t) > 0$, meaning it will be impossible to distinguish the correct concept with certainty. However we have seen that this strong form of learning is quite possible if we allow sequential learners with no fixed bound on their potential sample size. A number of surprisingly strong results can be obtained for this model.

Definition 2 (CAC-learning) *(Certainly Approximately Correct) For a given accuracy level ϵ, we say*
(1) *"a learner L* CAC-learns *c" if given random examples generated by* P *and labeled by c L produces a hypothesis h such that $d_{\mathrm{P}}(h, c) \le \epsilon$ with probability 1, and*
(2) *"a learner em CAC-learns a concept* space *(C, P)" if it CAC-learns each c in C.*

The procedure S_1 can easily be generalized to a simple generic learning procedure S applicable to arbitrary concept spaces; see Figure 5. In fact S CAC-learns virtually any concept space, provided only that it terminates with probability 1, since any hypothesis it returns is guaranteed to be ϵ-accurate by construction. We measure S's sample efficiency by the upper bound on its expected sample size, in the worst case over all concept in C. This is determined by how quickly the neighborhood of consistent concepts shrinks to become coverable by a single concept; a quantity we measure by the *uniform reducibility* of the concept space.

Definition 3 *The space (C, P) is* uniformly reducible *if there is a t and a $\rho > 0$ such that for any target concept c in the space, the space of consistent concepts is reduced to an ϵ-ball around c after t training examples with probability at least ρ.*

It turns out that uniform reducibility is both a sufficient and necessary condition for a concept space to be CAC-learnable:

Theorem 1 *A space (C, P) is CAC-learnable* iff *it is uniformly reducible for all $\epsilon > 0$* iff *S CAC-learns it.*[2]

[2]These results assume the concept space satisfies a benign regularity condition, namely that it is "separable" in the sense defined in (Schuurmans, 1994).

Therefore if a concept space is not uniformly reducible then it is not CAC-learnable by any learner, but if it is uniformly reducible then S CAC-learns it. So S can be viewed a "universal" CAC-learner in this sense.

Note 1 *As there are finitely coverable spaces that are* not *uniformly reducible (Schuurmans, 1994),* CAC*-learnability implies* PAC*-learnability, but not* vice versa*. (Hence, not every* PAC*-learnable concept space is* CAC*-learnable.) However, most of the concept spaces considered by researchers are in fact uniformly reducible — only pathological examples appear not to be.*

Not only is S a universal learner in the above sense, but it is also *optimally efficient* as well.

Theorem 2 *For any concept space, S has* optimal *expected sample size among all* CAC*-learners for the space.*[3]

So even though S is an incredibly simple-minded learning strategy, it is the *optimal* possible procedure in terms of minimizing expected sample size.

The previous section shows that S's expected sample size can be precisely determined for simple concept spaces like (C_1, P_1), and furthermore, that its performance far outstrips that of the fixed sample size BI strategy in these cases. The analysis of Section 3 can be extended to more general classes of concept spaces, for example d independent copies of C_1 on $[0,1], [1,2], \ldots, [d-1, d]$. Analysis shows that S remains orders of magnitude more efficient than BI in this case. Unfortunately we have yet to identify a simple structural property of concept spaces (*e.g.*, covering number, or some generalization of the VCdimension of C) that will permit us to prove tight upper and lower bounds on $\mathrm{E}T^S$ that are applicable in general. This remains an interesting open challenge.

Note 2 *However such a characterization is not as important here as for the fixed sample size case: procedure S will perform* optimally *regardless of how carefully we can predict its performance* a priori*! This is unlike fixed sample size techniques whose* actual *sample efficiency depends critically on our ability to* prove *the sufficiency of specific sample size bounds.*

5 Truncation: Bounding the sample size

Even though S is a very efficient learner it has the drawback that there is no finite upper bound on the total number of examples it might observe. Of course a small expected sample size ensures that the probability of exceeding any large number of examples is exceedingly small, which we have seen. If, however, one cannot tolerate even a minuscule probability of the sample size exceeding some pre-set bound, then it is possible to consider *truncated* sequential learners that never observe more than a specified number of examples. Of course, in so doing we must give up the possibility of achieving CAC-learning and settle instead for the weaker PAC criterion with $\delta > 0$.

[3] Making the same separability assumption required for Theorem 1.

Procedure $S \wedge L$ $(C, \mathrm{P}, \epsilon, \delta)$

- Run S and L in parallel. (Showing each labeled example to both.)
- Return the first hypothesis returned by either.

Figure 6: Procedure $S \wedge L$

Procedure S^{δ} $(C, \mathrm{P}, \epsilon, \delta)$

- Let t^{δ} be an upper bound such that $\mathrm{P}(T^S > t^{\delta}) < \delta$ for every c in the space (C, P).
- Run procedure S and return any hypthesis it might produce.
- If S has not terminated after t^{δ} examples, stop and return an arbitrary hypothesis.

Figure 7: Procedure S^{δ}

Strict domination: The first thing to notice is that the sample efficiency of any fixed sample size learner L can be strictly dominated by the sequential learner "$S \wedge L$" defined in Figure 6. Since S never returns a bad hypothesis when it terminates, $S \wedge L$ only makes a mistake whenever L does, so $S \wedge L$ is a PAC-learner whenever L is. Obviously $S \wedge L$ never observes more examples than L and is quite likely to observe fewer, *cf.* Section 3.

Tail truncation: An alternative approach is based on truncating the S procedure at its "δ-tail": the sample size t^{δ} for which $\mathrm{P}(T^S > t^{\delta}) < \delta$ for any c in the space. Figure 7 gives a procedure S^{δ} which is a slight modification of S guaranteed to PAC-learn while never exceeding the finite sample size t^{δ}. Notice S^{δ} does not PAC-learn in every situation where BI does by Note 1. However S^{δ} dramatically outperforms BI in case studies involving finitely reducible spaces; requiring strictly fewer training training examples in the worst case, and requiring orders of magnitude fewer on average.

For example, recall the concept space $\mathcal{C}_1 = (C_1, \mathrm{P}_1)$ from Section 3. Here we construct a truncated version of S_1 by using the bound (4) to obtain an upper bound on the number of examples sufficient for S_1 to terminate and return an ϵ-accurate hypothesis with probability at least $1 - \delta$. Thus, we can construct the PAC-learning procedure S_1^{δ} simply by truncating S_1 at the sample size t^{δ} given below:

Proposition 3 *For* $t^{\delta} \geq \frac{1}{\epsilon}(2 + \frac{3}{2} \ln \frac{1}{\delta})$, *we have* $\mathrm{P}(T^{S_1} > t^{\delta}) \leq \delta$ *for any* c *in* (C_1, P_1).

To compare the sample efficiency of this procedure to BI, notice that $T^{S_1^{\delta}} < T^{S_1}$ and so we immediately have $\mathrm{E}T^{S_1^{\delta}} \leq 1/\epsilon$, which again is orders of magnitude smaller than BI's sample size bound (3). Also notice that the sample size bound t^{δ} strictly dominates (3), and in fact constitutes an improved upper bound on the number of examples sufficient for PAC-learning the concept space (C_1, P_1).

6 Conclusions

This paper introduced the idea of using sequential learning algorithms to improve the sample complexity of concept learning in the distribution specific model. We presented a simple example which showed that a simple sequential strategy learns with drastically fewer training examples than previous fixed sample size approaches. This strategy actually learns with certainty, not just high probability, and in fact achieves optimal expected sample complexity among all such learners! In addition, we showed how analyzing the sample complexity of a sequential learner can actually yield improved fixed sample size bounds in some cases. Our results also extend (Linial, Mansour and Rivest, 1988), as our goal is to improve learning efficiency *uniformly* over all possible target concepts c in the class C, not just gain an advantage for certain concepts by sacrificing performance on others.

The main challenge in actually implementing the sequential strategy S is to find an effective procedure for determining whether a set of consistent concepts is ϵ-coverable, and producing an ϵ-cover when one exists. While this chapter also presents a simple class of concept spaces, S can also be easily implemented for other concept spaces such as axis-parallel rectangles and simple finite spaces like monomials and conjunctive concepts over $\{0,1\}^n$. However, much work remains to be done in order to scale these techniques up to handle more realistic concept spaces such as multi-layer neural networks. Another obvious direction for future research is to extend these results to the distribution-*free* setting. The ultimate goal is to develop a general PAC-learning procedure with a small (or better, optimal) expected sample complexity, analogous to the S algorithm developed here; however this appears to be a difficult challenge. An intermediate goal is to develop sample efficient sequential algorithms for interesting special cases.

This work constitutes a first step towards learning algorithms that make efficient use of the training samples, thereby using far fewer training examples than existing PAC-learning techniques, while still achieving the same accuracy and reliability guarantees. Substantial improvements in the efficiency of these techniques might actually result in *practical* PAC-learning systems that could be used in real world applications.

References

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

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

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

Kulkarni, S. (1991). *Problems of computational and information complexity in machine vision and learning.* PhD thesis, MIT, Department of Electrical Engineering and Computer Science.

Linial, N., Mansour, Y., and Rivest, R. (1988). Some results on learnability and the Vapnik-Chervonenkis dimension. In *Proceedings COLT-88*.

Schuurmans, D. (1994). *Efficient, Accurate, and Reliable Classification Learning.* PhD thesis, University of Toronto, Department of Computer Science. (Forthcoming).

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

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

Wald, A. (1947). *Sequential Analysis*. John Wiley & Sons, New York.