Tracking Drifting Concepts By Minimizing Disagreements

David P. Helmbold and Philip M. Long*

UCSC-CRL-91-26
August 1991, revised August 1992

Baskin Center for
Computer Engineering & Information Sciences
University of California, Santa Cruz
Santa Cruz, CA 95064 USA

ABSTRACT

In this paper we consider the problem of tracking a subset of a domain (called the *target*) which changes gradually over time. A single (unknown) probability distribution over the domain is used to generate random examples for the learning algorithm and measure the speed at which the target changes.

Clearly, the more rapidly the target moves, the harder it is for the algorithm to maintain a good approximation of the target. Therefore we evaluate algorithms based on how much movement of the target can be tolerated between examples while predicting with accuracy ϵ. Furthermore, the complexity of the class $\mathcal{H}$ of possible targets, as measured by d, its VC-dimension, also effects the difficulty of tracking the target concept.

We show that if the problem of minimizing the number of disagreements with a sample from among concepts in a class $\mathcal{H}$ can be approximated to within a factor k, then there is a simple tracking algorithm for $\mathcal{H}$ which can achieve a probability ϵ of making a mistake if the target movement rate is at most a constant times $\epsilon^2/(k(d+k)\ln\frac{1}{\epsilon})$, where d is the Vapnik-Chervonenkis dimension of $\mathcal{H}$. Also, we show that if $\mathcal{H}$ is properly PAC-learnable, then there is an efficient (randomized) algorithm that with high probability approximately minimizes disagreements to within a factor of $7d+1$, yielding an efficient tracking algorithm for $\mathcal{H}$ which tolerates drift rates up to a constant times $\epsilon^2/(d^2\ln\frac{1}{\epsilon})$.

In addition, we prove complementary results for the classes of halfspaces and axis-aligned hyperrectangles showing that the maximum rate of drift that any algorithm (even with unlimited computational power) can tolerate is a constant times ϵ^2/d.

[*] dph@cse.ucsc.edu and plong@igi.tu-graz.ac.at, To appear in *Machine Learning*

1 Introduction

In the fairy tale, Rip van Winkle slept for 20 years and when he finally woke up, he discovered that he was out of step with the world. Presumably, Rip would have been much better off if he woke up every day. However, if he woke for only one day each week or month or year how comfortable would Rip be with the world after his 20 year slumber? This leads to the question "How long can one nap before losing touch with the world?" which is the subject of this paper.

More formally, let D be a probability distribution on some set X and $\mathcal{H}$ be a class of $\{0,1\}$-valued functions defined on X. In the sleeper example, each $h \in \mathcal{H}$ represents a possible state of the world. When Rip van Winkle wakes for the t^{th} time, the world is in some state $h_t \in \mathcal{H}$. Rip gets x_t, a randomly drawn (w.r.t. D) element of X, and is asked for the value of $h_t(x_t)$. One interpretation is that x_t is a possible course of action, and $h_t(x_t) = 1$ when x_t is appropriate in the current world state. Just before Rip goes back to sleep, he is told the value of $h_t(x_t)$.

In other words, given $(x_1, h_1(x_1))$, $(x_2, h_2(x_2))$, ..., $(x_{t-1}, h_{t-1}(x_{t-1}))$, and a point x_t, Rip is asked to predict the value of $h_t(x_t)$. If Rip's prediction is incorrect we say that he makes a mistake on x_t. If Rip rarely makes mistakes, then he successfully tracks the state of the world. In our model, an adversary chooses the probability distribution D and the sequence of functions ahead of time, before the x_i's are generated.

The sequence of examples could be uninformative for two different reasons. First, x_1 through x_{t-1} may come from an uninteresting part of the domain. Any learning algorithm using randomly drawn examples must deal with this potential difficulty. A more severe problem is that the h_t chosen by the adversary may be unrelated to the previous h_i's. If the adversary randomly chooses h_t to be either the constant function 1 or the constant function 0, then no algorithm can expect to predict $h_t(x_t)$ correctly more than half the time. We deal with this problem with an assumption that the state of world evolves slowly. Thus the adversary must choose sequences of functions where each h_i is "close" to h_{i-1}. This is made precise in Section 2.

Many readers will notice the similarity of our model to the prediction model studied by Haussler, Littlestone and Warmuth (1988, 1990) and others. The key difference is that in our model there is no single target function, but rather a succession of related target functions. Since the learner may receive only a single example before the target changes, it is unreasonable to expect that the hypotheses converge to a target. However, it is possible to bound the probability of a mistake on a trial in terms of how much the target is allowed to change between trials and the complexity of $\mathcal{H}$.

Our results include:

- a general-purpose algorithm which tolerates target movement rates up to $c_1\epsilon^2/(d \ln \frac{1}{\epsilon})$ (Theorem 3 and Corollary 5), and
- a possibly more computationally efficient variant of this algorithm which tolerates target movements of up to $c_2\epsilon^2/(d^2 \ln \frac{1}{\epsilon})$ (Theorem 7),
- bounds for the classes of axis-aligned halfspaces and hyperrectangles showing that for all n and $\epsilon < 1/12$, no algorithm can tolerate target movement greater than $c_3\epsilon^2/n$, where n is the dimension of the space from which examples are drawn (Theorem 14).[1]

[1] Since, in both the case of halfspaces and that of hyperrectangles in n-dimensional space, the first algorithm above tolerates drift rates up to a constant times $\epsilon^2/(n \ln \frac{1}{\epsilon})$, these bounds establish the fact that the first algorithm is within a constant times a log factor of optimal.

In the above, the c_i's are constants, ϵ denotes the desired probability of error, and d is the VC-dimension of $\mathcal{H}$. The first general-purpose algorithm above is computationally efficient whenever the problem of finding a member of $\mathcal{H}$ which minimizes the number of disagreements with a set of examples can be solved efficiently. Its variant is computationally efficient whenever the problem of finding an element of $\mathcal{H}$ consistent with a set of examples can be solved efficiently, as is the case with both halfspaces and hyperrectangles.

Our algorithms use only the most recent t examples (rather than the entire sequence) to make their predictions. They work by either minimizing or approximately minimizing the number of disagreements with the most recent examples, and using the resulting hypothesis to predict the label of the next point. To analyze such algorithms, one might imagine applying the results of Vapnik and Chervonenkis (1971) to show that if for each hypothesis h in the class, we estimate the probability that h will make a mistake on the next trial by considering the fraction of the last t trials on which h made a mistake, none of these estimates will be very far from the true estimated probabilities. The movement of the target prevents us from simply applying their results. To remedy this, we first bound the probability that for any hypothesis h, the estimate we obtain is very far from the estimate we would have obtained, had the target not been moving. Then we are ready to apply uniform convergence results.

If we now apply the results of Vapnik and Chervonenkis, however, our analysis indicates that these algorithms are more than a factor of ϵ from the best upper bounds we can prove on the maximum tolerable rate of drift. In the case of learning stationary targets, it was observed by Blumer, Ehrenfeucht, Haussler and Warmuth (1989) that uniformly good estimates of the quality of hypotheses were not required for learning in Valiant's (1984) PAC-model. Instead, one only needed to bound the probability that an "ϵ-bad" hypothesis was consistent with a sequence of examples. They were then able to shave a factor of $1/\epsilon$ off the bound on the number of examples required for learning with accuracy ϵ obtained by simply applying the results of Vapnik and Chervonenkis (1971). However, in our case, there may not be any hypothesis consistent with more than a few of the most recent examples. Nevertheless, given reasonable restrictions on the rate of drift there is, with high probability, some hypothesis having very few disagreements with a reasonable sized suffix of a random sequence of examples. Thus, we are able to apply another of the results of Blumer, et al (1989), which bounds the probability that any ϵ-bad hypothesis is consistent with all but a fraction $\epsilon/2$ of the examples. The number of examples required to bound this "ϵ-bad but highly consistent" probability by δ is within a constant of that for the completely consistent case. Thus, ignoring constants, the factor of $1/\epsilon$ savings is retained, reducing our tracking bounds by a factor of ϵ.

The result of this analysis is a simple "minimize disagreements" algorithm which is within a log factor of optimal for halfspaces and hyperrectangles. A slightly modified analysis holds for the case in which the tracking algorithm uses a hypothesis which only approximately minimizes disagreements with a suffix of the examples.

In Section 4, we give a general purpose algorithmic transformation turning a randomized polynomial time hypothesis finder $\mathcal{A}$ (as defined by Blumer, et al (1989)) which, with high probability, returns a hypothesis consistent with an input sample, into an algorithm which efficiently approximately minimizes disagreements to within a factor of $7d + 1$, where d is the VC-dimension of the target class. We use a technique due to Kearns and Li (1988) and Abe and Watanabe (1992), working in stages, where at each stage, we subsample according to the distribution which is uniform over the sample, hoping to get a subsample for which there is a consistent hypothesis, so that we can successfully apply $\mathcal{A}$. We then return the best hypothesis of those produced by $\mathcal{A}$ during the various stages. We use the tightest available PAC-learning bounds, due to Anthony, Biggs and Shawe-Taylor (1990), to argue that with high probability, a hypothesis consistent with the subsample can't be too bad on the whole sample.

Littlestone and Warmuth (1989) describe a variant of the weighted majority algorithm where the weights are kept above some lower limit. This allows the weighted majority algorithm to recover and adapt to changes in the target. However, if the target changes k times, then their mistake bound for the weighted majority algorithm goes up by about a factor of k. It is difficult to translate these bounds into our model as our targets potentially change with each example.

Kuh, Petsche and Rivest (1990,1991) studied a variety of models in which the target changes over time, including cases in which the target drifts slowly. For many of their main results, it is assumed that the sequence of targets is produced by an adversary which at each time has access to the earlier random examples seen by the tracking algorithm. In contrast, we assume that the sequence of targets is chosen by an adversary before any random examples are generated.

Aldous and Vazirani (1990) studied a different version of learning in a changing environment. In their model the target concept is fixed, but the examples are generated by a Markov process rather then from a fixed distribution.

The conclusions contain potential applications, observations, and a list of open problems.

The results presented here improve on preliminary results described by the authors (1991).

2 Notation and Mathematical Preliminaries

Let **N** denote the positive integers and **Q** denote the rationals. Let ln denote that natural logarithm, and log denote the logarithm base 2.

After Vapnik (1989), we will adopt a naive attitude toward measurability, assuming that every set is measurable, and simply speak of probability distributions on sets. This assumption is not unreasonable, since if a digital computer is to input or output representations of arbitrary set elements, the set must be countable. If X is a set, and D is a probability distribution on X, and if $\phi(x)$ is some mathematical statement containing x as a free variable, define $\mathbf{Pr}_{x\in D}(\phi(x))$ as $D(\{x \in X : \phi(x)\})$. Define $\mathbf{E}_{x\in D}$ similarly for expectations of random variables defined on X. We will drop the subscripts where there is no possibility of confusion.

If X is a set and $\mathcal{H}$ is a family of $\{0,1\}$ valued functions defined on X, then the Vapnik-Chervonenkis (1971) (VC) dimension of $\mathcal{H}$ is

$$\max\{|T| : T = \{t_1, ..., t_k\} \subseteq X, \{(h(t_1), ..., h(t_k)) : h \in H\} = \{0,1\}^{|T|}\}. \tag{2.1}$$

We will assume throughout that all classes discussed have at least two elements, and thus have VC-dimension at least one.

A *tracking problem* consists of a set (or *domain*) X and a family $\mathcal{H}$ of $\{0,1\}$-valued functions defined on X, called the *target class*. A $\{0,1\}$ valued function defined on X is called a *concept*. We will speak of a concept and the subset of X on which it takes value 1 interchangeably. An *example* is an element of $X \times \{0,1\}$, and a *sample* is a finite sequence of examples. A function h *agrees* (resp. *disagrees*) with an example (x, ρ) when $h(x) = \rho$ (resp. $h(x) \neq \rho$). A function is *consistent* with a sample if it agrees with all examples in the sample. We often use the discrete loss function, $l(\alpha, \beta)$, defined to be 0 when $\alpha = \beta$ and 1 otherwise, to count numbers of disagreements.

Let ? be the set of all infinite sequences of bits, and $\mathcal{U}$ be the distribution which sets each bit in the sequence independently with probability 1/2. A *(randomized) tracking strategy* is a mapping from $(\cup_m (X \times \{0,1\})^m) \times X \times ?$ to $\{0,1\}$.

If $S = \langle f_t \rangle_{t \in \mathbf{N}}$ is a sequence of concepts and $\bar{x} \in X^n$ with $n \geq m$, the *m-sample of S generated by $\bar{x}$*, written $sam_m(S, \bar{x})$, is the sequence of pairs $\langle (x_1, f_1(x_1)), ..., (x_m, f_m(x_m)) \rangle$. Informally, $sam_m(S, \bar{x})$ is simply the first m examples which are used by a tracking strategy to predict $f_{m+1}(x_{m+1})$.

Let D be a probability distribution over X. If $\Delta \geq 0$, a sequence $\langle f_t \rangle_{t \in \mathbf{N}}$ of concepts is called (Δ, D)-admissible if for each $t \in \mathbf{N}$, $\mathbf{Pr}_{x \in D}(f_t(x) \neq f_{t+1}(x)) \leq \Delta$.

Let A be a tracking strategy. We say that A (ϵ, Δ)-tracks $\mathcal{H}$ if there is an $m_0 \in \mathbf{N}$ such that for all $m \geq m_0$, for all probability distributions D on X, and for all (Δ, D)-admissible sequences $S = \langle f_t \rangle_{t \in \mathbf{N}}$ of functions in $\mathcal{H}$,

$$\mathbf{Pr}_{\bar{x} \in D^{m+1}, \sigma \in \mathcal{U}}(A(sam_m(S, \bar{x}), x_{m+1}, \sigma) \neq f_{m+1}(x_{m+1})) \leq \epsilon. \tag{2.2}$$

We say that $\mathcal{H}$ is (ϵ, Δ)-trackable if there is a tracking strategy which (ϵ, Δ)-tracks $\mathcal{H}$.

To discuss issues of computational efficiency, we will need the following definitions. We say that $\mathcal{H} = \{\mathcal{H}_n : n \in \mathbf{N}\}$ is a *stratified tracking problem* if for each $n \in \mathbf{N}$, $(\mathbf{Q}^n, \mathcal{H}_n)$ is a tracking problem.[2] An algorithm for a stratified tracking problem consists of a tracking algorithm A_n for each n. We assume that the random bits are presented on an auxiliary tape, and thus accessing the next random bit in the sequence takes unit time.

We say that $A = \{A_n\}$ efficiently tracks $\mathcal{H}$ if there is a polynomial p and positive constants c and k such that for all relevant ϵ, n,

- each prediction is computed in time bounded by $p(1/\epsilon, n, b)$, where b is the number of bits needed to encode the "largest" example seen.
- at most $p(1/\epsilon, n, b)$ space is required to store information between trials,
- if $\Delta < c(\epsilon/n)^k$, A_n (ϵ, Δ)-tracks $\mathcal{H}_n$.

Note that the bound on the space required is not allowed to grow with the number of trials. Thus an efficient tracking algorithm may not, in general, keep all previously seen examples.

3 Increasingly unreliable evidence and hypothesis evaluation

In this section we analyze a simple tracking algorithm which ignores all examples beyond some time in the past and uses the hypothesis which disagrees with the fewest remaining examples for prediction. The results of this section, together with those of Section 5, show that this apparently naive algorithm is within a constant times a log factor of optimal for the classes of halfspaces and hyperrectangles. We also show that it is sufficient to only approximately minimize disagreements to within a constant.

[2] We assume rationals are encoded by encoding both the numerator and the denominator in binary.

As discussed in the introduction, the fraction of the considered examples disagreeing with a hypothesis can be viewed as an estimate of the probability that the hypothesis will make a mistake on the next example. In the following series of lemmas we bound the probability that there exists a hypothesis h in class $\mathcal{H}$ such that the estimate of h's error is small but the true probability that h will yield an incorrect prediction is large.

We will make use of the standard Chernov bounds, which we state here.

Lemma 1 (Angluin and Valiant (1979), Littlestone (1989), Hagerup and Rub (1990)): *Let $t \in \mathbf{N}$, and let $r_1, ..., r_t$ be independent $\{0,1\}$-valued random variables. Choose $\alpha, 0 < \alpha \le 1$. Let $\mu = \sum_{i=1}^t \mathbf{Pr}(r_i = 1)$. Then*

$$\mathbf{Pr}\left(\sum_{i=1}^t r_i \ge (1+\alpha)\mu\right) \le e^{-\alpha^2\mu/3}. \tag{3.1}$$

For each $h \in \mathcal{H}$, $f \in \mathcal{H}$, $m \in \mathbf{N}$, $\bar{x} \in X^m$, denote

$$\mathbf{Pr}_{x\in D}(h(x) \neq f(x)), \tag{3.2}$$

by $\mathbf{er}_f(h)$ (D is to be understood from context), and denote

$$\frac{1}{m}\sum_{i=1}^m l(h(x_i), f(x_i)) \tag{3.3}$$

by $\hat{\mathbf{er}}_f(h, \bar{x})$. Note that $\hat{\mathbf{er}}_f$ is the empirical estimate of the error of h obtained when the (unchanging) target concept is f.

Our first lemma follows immediately from the results of Blumer, et al (1989).

Lemma 2: *For any set X and concept class $\mathcal{H}$ over X, for any distribution D on X, for any $f \in \mathcal{H}$, for all $0 < \epsilon \le 1/2$, if $m \ge \frac{64d}{\epsilon} \ln \frac{64}{\epsilon}$, where d is the VC-dimension of $\mathcal{H}$, then*

$$\mathbf{Pr}_{x\in D^m}(\exists h \in \mathcal{H} : \mathbf{er}_f(h) \ge \epsilon, \hat{\mathbf{er}}_f(h) < \epsilon/2) \le \epsilon. \tag{3.4}$$

We are now ready to present the main result of this section. The following theorem shows that if a randomized tracking strategy is likely to predict with a hypothesis that approximately minimizes disagreements on the previous examples, then the probability that the algorithm makes a mistake on the next example is small.

Theorem 3: *Let $(X, \mathcal{H})$ be a tracking problem, $d =$ VCdim($\mathcal{H}$), and choose $\epsilon > 0$. Suppose A is a randomized tracking algorithm which, with probability at least $1 - \epsilon/6$, predicts using an $h \in \mathcal{H}$ having at most k times the minimum number of disagreements on the previous trials. Choose a distribution D on X and*

$$m \ge \max\left(\frac{192d}{\epsilon}\ln\frac{192}{\epsilon}, \frac{72k}{\epsilon}\ln\frac{6}{\epsilon}\right). \tag{3.5}$$

Then if the sequence of targets from $\mathcal{H}$, $S = \langle f_i \rangle_{i\in\mathbf{N}}$, satisfies $\sum_{i=1}^m \mathbf{Pr}_{x\in D}(f_i(x) \neq f_{m+1}(x)) \le m\epsilon/(24k)$, the probability that A makes a mistake on the $(m+1)$st trial is at most ϵ.

Proof: Fix m and k. For each $\bar{x} \in X^m$, let *mindis*$(\bar{x})$ be the set of all hypotheses in $\mathcal{H}$ which approximately minimize disagreements with $sam_m(S, \bar{x})$ to within a factor of k.

Define F to be the event that the hypothesis chosen by A is not in *mindis*$(\bar{x})$.

Define F' to be the event that there are more than twice the expected number of disagreements between the previous trials and f_{m+1}, i.e.,

$$F' = \{\bar{x} \in X^m : \sum_{i=1}^{m} l(f_i(x_i), f_{m+1}(x_i)) > m\epsilon/(12k)\}. \tag{3.6}$$

Applying Lemma 1 (with $\alpha = 1$), we have

$$\mathbf{Pr}_{\bar{x} \in D^m}(F') \leq e^{-m\epsilon/(72k)} \leq \epsilon/6, \tag{3.7}$$

since $m \geq \frac{72k}{\epsilon} \ln \frac{6}{\epsilon}$.

Define $E = F \cup F'$. Then $\mathbf{Pr}(E) \leq \epsilon/3$.

For each $\bar{x} \in X^m, \sigma \in ?$, let $h_{\bar{x},\sigma}$ be A's hypothesis after seeing the sequence

$$(x_1, f_1(x_1)), ..., (x_m, f_m(x_m)) \tag{3.8}$$

of examples and the random sequence σ. Let

$$G = \{(\bar{x}, \sigma) \in X^m \times ? : \mathbf{er}_{f_{m+1}}(h_{\bar{x},\sigma}) > \epsilon/3\}, \tag{3.9}$$

be the set of sequences of points and random bits that cause A to produce an inaccurate hypothesis.

If *mistake* is the event that A makes a mistake on trial $m+1$, we have

$$\begin{aligned}
\mathbf{Pr}_{(\bar{x},y,\sigma) \in D^m \times D \times \mathcal{U}}(\mathit{mistake}) &\leq \mathbf{Pr}(\mathit{mistake} \cap \bar{E}) + \mathbf{Pr}(\mathit{mistake} \cap E) && (3.10)\\
&\leq \mathbf{Pr}(\mathit{mistake} \cap \bar{E}) + \mathbf{Pr}(E) && (3.11)\\
&\leq \mathbf{Pr}(\mathit{mistake} \cap \bar{E}) + \epsilon/3 && (3.12)\\
&\leq \mathbf{Pr}(\mathit{mistake} \cap \bar{E} \cap G) + \mathbf{Pr}(\mathit{mistake} \cap \bar{E} \cap \bar{G}) + \epsilon/3 && (3.13)\\
&\leq \mathbf{Pr}(\bar{E} \cap G) + 2\epsilon/3. && (3.14)
\end{aligned}$$

Next, we have

$$\begin{aligned}
\mathbf{Pr}(\bar{E} \cap G) &= \mathbf{Pr}(\mathbf{er}_{f_{m+1}}(h_{\bar{x},\sigma}) > \epsilon/3 \text{ and } \frac{1}{m}\sum_{i=1}^{m} l(f_i(x_i), f_{m+1}(x_i)) \leq \epsilon/(12k) && (3.15)\\
&\qquad \text{and } h_{\bar{x},\sigma} \in \mathit{mindis}(\bar{x})) && (3.16)\\
&\leq \mathbf{Pr}\left(\mathbf{er}_{f_{m+1}}(h_{\bar{x},\sigma}) > \epsilon/3 \text{ and } \frac{1}{m}\sum_{i=1}^{m} l(f_i(x_i), f_{m+1}(x_i)) \leq \epsilon/(12k)\right. && (3.17)\\
&\qquad \left. \text{and } \frac{1}{m}\sum_{i=1}^{m} l(f_i(x_i), h_{\bar{x},\sigma}(x_i)) \leq \epsilon/12\right) && (3.18)
\end{aligned}$$

since $f_{m+1} \in \mathcal{H}$ and $h_{\bar{x},\sigma} \in \mathit{mindis}(\bar{x})$ implies that $h_{\bar{x},\sigma}$ has at most k times as many disagreements as f_{m+1}. Recalling that $k \geq 1$ and applying the triangle inequality for l, we have

$$\begin{aligned}
\mathbf{Pr}(\bar{E} \cap G) &\leq \mathbf{Pr}\left(\mathbf{er}_{f_{m+1}}(h_{\bar{x},\sigma}) > \epsilon/3 \text{ and } \frac{1}{m}\sum_{i=1}^{m} l(h_{\bar{x},\sigma}(x_i), f_{m+1}(x_i)) \leq \epsilon/6\right) && (3.19)\\
&\leq \epsilon/3 && (3.20)
\end{aligned}$$

by Lemma 2, since $m \geq \frac{192d}{\epsilon} \ln \frac{192}{\epsilon}$. Plugging in to (3.14) yields the desired result. □

If $\langle f_i \rangle$ is a (Δ, D)-admissible sequence of functions, then $\mathbf{Pr}_{x \in D}(f_i(x) \neq f_{m+1}(x)) \leq (m - i + 1)\Delta$, and

$$\sum_{i=1}^{m} \mathbf{Pr}_{x \in D}(f_i(x) \neq f_{m+1}(x)) \leq m(m+1)\Delta/2. \tag{3.21}$$

Thus we obtain the following corollary.

Corollary 4: *Let A be a tracking strategy that predicts using a randomly chosen hypothesis which, with probability $1 - \epsilon/6$, approximately minimizes the number of disagreements on the first m trials to within a factor of k. Choose ϵ and m as in Theorem 3. Then if $\Delta \leq \frac{\epsilon}{12k(m+1)}$, the probability that A makes a mistake on the $(m+1)$st trial of a (Δ, D)-admissible sequence of functions is at most ϵ.*

Note that by ignoring (not counting disagreements with) examples beyond a certain point in the past we can, loosely speaking, make any later trial "look like" the $(m+1)$st trial. This observation leads to the following Corollary.

Corollary 5: *Let X be a domain, and $\mathcal{H}$ be a class of concepts over X of VC-dimension d. Assume A is a randomized algorithm which with probability $1 - \epsilon/6$ finds an $h \in \mathcal{H}$ which approximates, to within a constant factor k, the minimum number of disagreements on a sample. Let A' be the tracking algorithm which predicts using the hypothesis produced by A from the most recent $m = \lceil (c_1 d/\epsilon)\log(1/\epsilon) \rceil$ examples, where $c_1 > 0$ depends on k. There is a positive constant c_2, depending only on k, such that for any $0 < \Delta < \epsilon$ where*

$$\Delta \leq \frac{c_2 \epsilon^2}{d \log \frac{1}{\epsilon}}, \tag{3.22}$$

strategy A' (ϵ, Δ)-tracks $\mathcal{H}$.

4 Efficiently Approximately Minimizing Disagreements

In this section we discuss the application of the techniques of Kearns and Li (1988) to the problem of approximately minimizing disagreements from among the hypotheses in a class $\mathcal{H}$, showing that if there is an efficient algorithm which returns a hypothesis with no disagreements if there is one, then there is an efficient randomized algorithm which with high probability returns a hypothesis that minimizes disagreements to within a factor of a constant times the VC-dimension of $\mathcal{H}$. Results very similar to those described here are implicit in the work of Kearns and Li (Theorems 12 and 16), although some minor modifications are necessary.[3] Also, we make use of the techniques of Kearns and Li (1988) in our proof. Furthermore, algorithm Min-Disagreements from Figure 4.1 is very similar to the Algorithm B given in a recent paper by Abe and Watanabe (1992), which was described to us some time ago by Abe. However, our applications appear to be substantially different.

First, the results of Anthony, Biggs and Shawe-Taylor (1990) may be applied[4] to obtain the following.

[3] The difference between the result trivially obtainable by combining Theorems 12 and 16 of Kearns and Li (1988) and our result is that in the former, the sample is restricted to have the same number of positive and negative examples.

[4] For $d > 1$, use Theorem 2.1 of their paper with $\delta = 1/2$, and for $d = 1$ a simple argument along the lines of the proof for their Theorem 2.1 suffices.

```
Algorithm Min-Disagreements
Inputs:
    a sample $S$ of $m$ examples;
    $l$, the number of iterations to run;
    $d = VCdim(\mathcal{H}_n)$;
    desired approximation factor $\gamma > 1$.
Uses:
    A randomized algorithm $\mathcal{A}$ for the consistency problem
    associated with $\mathcal{H}_n$.

choose an $h \in \mathcal{H}_n$ arbitrarily;
for $j := 1$ to $ld$ do
    run $\mathcal{A}$ on $S'$ obtaining hypothesis $h'$;
    if $h'$ is consistent with $S$ then stop and return $h'$
end for;

for $\widehat{opt} := 1$ to $m/\gamma$ do
    $s := \left\lceil (7d(m - \widehat{opt})/\gamma\widehat{opt}) \ln(9(m - \widehat{opt})/\gamma\widehat{opt}) \right\rceil$;
    for $j := 1$ to $l$ do
        draw $S'$, an $s$-element subsample of $S$ uniformly at random with replacement;
        run $\mathcal{A}$ on $S'$ obtaining hypothesis $h'$;
        if $h'$ has fewer disagreements with $S$ than $h$, set $h := h'$;
    end for;
end for;
return $h$;
```

Figure 4.1: Algorithm Min-Disagreements

Theorem 6 (Anthony, et al (1990)): *Let X be a set and let $\mathcal{H}$ be a concept class over X of VC-dimension d. Let D be a probability distribution over $\mathcal{H}$. Choose $f \in \mathcal{H}$ and $\epsilon < 1/2$. Then if $m \geq (7d/\epsilon)\ln(9/\epsilon)$,*

$$\mathbf{Pr}_{\bar{x} \in D^m}(\exists h \in \mathcal{H} : \forall i, h(x_i) = f(x_i) \text{ and } \mathbf{Pr}_{y \in D}(h(y) \neq f(y)) \geq \epsilon) \leq 1/2. \tag{4.1}$$

Now, we turn to the main result of this section. If $\mathcal{H}$ is a concept class, then the consistency problem associated with $\mathcal{H}$ is as follows:

> Given a sample, find any hypothesis in $\mathcal{H}$ consistent with the sample if there is one, otherwise return any $h \in \mathcal{H}$.

A randomized polynomial time algorithm for the consistency problem returns, in time polynomial in *VCdim*($\mathcal{H}$) and the size of the sample, an h in $\mathcal{H}$. If the sample is consistent with some hypothesis in $\mathcal{H}$ then, with probability $q > 1/2$, the returned h will be consistent with the sample. Note that by repeatedly running such an algorithm (and checking each result against the sample) an arbitrarily high confidence can be acheived.

Algorithm Min-Disagreements (see Figure 4.1) uses a randomized polynomial time algorithm for the consistency problem to approximately minimize the number of disagreements.

It should be obvious that if $\mathcal{A}$ runs in randomized polynomial time then the algorithm Min-Disagreements runs in time polynomial in d, l and m.

Theorem 7: *For any $n \in \mathbf{N}$, $\mathcal{H}_n \subseteq 2^{\mathbf{Q}^n}$ of VC-dimension d, and set of m examples S, if $\mathcal{A}$ solves $\mathcal{H}_n$'s consistency problem with probability $q > 1/2$ and there is an element of $\mathcal{H}_n$ consistent with all but opt of the examples in S, then Algorithm Min-Disagreements with inputs S,m,l,d,γ finds a hypothesis consistent with all but $(\gamma+1)opt$ examples in S with probability at least*

$$1 - \exp(-(l(2q-1)/2e^{1/\gamma})(\gamma\, opt/9(m-opt))^{7d/\gamma}). \tag{4.2}$$

Proof: Choose $m \in \mathbf{N}$ and let $S = \{(x_i, y_i) : 1 \le i \le m\}$ be a sample. Let

$$opt = \min\{|\{i : h(x_i) \ne y_i\}| : h \in \mathcal{H}\}, \tag{4.3}$$

the minimum possible number of disagreements between the sample and an $h \in \mathcal{H}$. We focus our attention on the case where $opt < m/(\gamma+1)$, since otherwise the theorem is trivial as any hypothesis is consistent with all but $(\gamma+1)opt$ examples of S.

Choose h_{opt} from among those hypotheses in $\mathcal{H}_n$ which have *opt* disagreements with S. Let $bad \subseteq S$ be the subset of the examples in S with which h_{opt} disagrees. Let D be the uniform distribution over S, and let D' be the uniform distribution over $S - bad$.

Consider the stage of the algorithm where $\widehat{opt} = opt$ and a particular iteration j of the inner loop where $\mathcal{A}$ produces hypothesis h'. Let *clean* be the event that none of the examples sampled during iteration j are in *bad* and *consist* be the event that h' is consistent with the subsample. By applying a standard approximation, we have

$$\begin{aligned} \mathbf{Pr}(clean \text{ and } consist) &\ge q(1 - opt/m)^s & (4.4)\\ &\ge q \exp\left(\frac{-opt\, s}{m - opt}\right) & (4.5)\\ & & (4.6) \end{aligned}$$

Now define *close* to be the event that h' agrees with all but $\gamma\, opt$ of the examples in $S - bad$, i.e. $\mathbf{Pr}_{z \in D'}(h'(z) \ne h_{opt}(z)) \le \gamma\, opt/(m - opt)$. (Note that when *close* occurs, h' agrees with all but $(\gamma+1)opt$ of the examples in S.) We have

$$\begin{aligned} &\mathbf{Pr}_{(S',\sigma) \in D^s \times \mathcal{U}}(\overline{close} \mid clean \text{ and } consist)\\ &= \mathbf{Pr}_{(S',\sigma) \in (D')^s \times \mathcal{U}}(\overline{close} \mid consist) \end{aligned} \tag{4.7}$$

since the distribution obtained by conditioning D^s on *clean* is $(D')^s$ (recall that $\mathcal{U}$ is the uniform distribution over sequences of bits, so that σ represents the randomization of consistency algorithm $\mathcal{A}$). Note that if both *clean* and *consist* occur then h' and h_{opt} agree with the examples in the subsample. Thus,

$$\begin{aligned} &\mathbf{Pr}_{(S',\sigma) \in D^s \times \mathcal{U}}(\overline{close} \mid clean \text{ and } consist)\\ &\le \mathbf{Pr}_{(S',\sigma) \in (D')^s \times \mathcal{U}}(\overline{close} \text{ and } consist)/\mathbf{Pr}(consist)\\ &\le \tfrac{1}{q}\mathbf{Pr}_{(S',\sigma) \in (D')^s \times \mathcal{U}}(\mathbf{Pr}_{z \in D'}(h'(z) \ne h_{opt}(z)) > \gamma\, opt/(m - opt)\\ &\qquad\qquad \text{and } \forall (x,y) \in S', h'(x) = h_{opt}(x))\\ &\le 1/2q, \end{aligned} \tag{4.8}$$

where the last inequality follows from Theorem 6 and the algorithm's choice of s. Thus,

$$\mathbf{Pr}_{(S',\sigma) \in D^s \times \mathcal{U}}(close \mid clean \text{ and } consist) \ge (2q-1)/2q. \tag{4.9}$$

Now we can bound the probability of *close*.

$$\begin{aligned}
\mathbf{Pr}_{(S',\sigma)\in D^s\times\mathcal{U}}(\mathit{close}) &\geq \mathbf{Pr}(\mathit{close} \text{ and } \mathit{clean} \text{ and } \mathit{consist}) && (4.10)\\
&= \mathbf{Pr}(\mathit{close} \mid \mathit{clean} \text{ and } \mathit{consist})\mathbf{Pr}(\mathit{clean} \text{ and } \mathit{consist}) && (4.11)\\
&\geq \frac{2q-1}{2}\exp\left(\frac{-\mathit{opt}\, s}{m-\mathit{opt}}\right) && (4.12)\\
&\geq \frac{2q-1}{2}\exp\left(\frac{-\mathit{opt}}{m-\mathit{opt}}\right)\exp\left(\frac{-7d}{\gamma}\ln\frac{9(m-\mathit{opt})}{\gamma\,\mathit{opt}}\right) && (4.13)\\
&\geq \frac{2q-1}{2}e^{-1/\gamma}\left(\frac{\gamma\,\mathit{opt}}{9(m-\mathit{opt})}\right)^{7d/\gamma} && (4.14)
\end{aligned}$$

Thus, the probability that the hypothesis returned after l iterations has more than $(\gamma+1)\mathit{opt}$ disagreements with S is at most

$$\left(1-\frac{2q-1}{2}e^{-1/\gamma}\left(\frac{\gamma\,\mathit{opt}}{9(m-\mathit{opt})}\right)^{7d/\gamma}\right)^{l} \leq \exp\left(\frac{-l(2q-1)}{2e^{1/\gamma}}\left(\frac{\gamma\,\mathit{opt}}{9(m-\mathit{opt})}\right)^{7d/\gamma}\right). \qquad (4.15)$$

This completes the proof. □

Corollary 8: *If $\gamma = 7d$ and $l \geq (3m/(d(2q-1)))\ln(1/\delta)$ then with probability at least $1-\delta$ Algorithm Min-Disagreements returns a hypothesis consistent with all but $(\gamma+1)\mathit{opt}$ of the examples in S.*

Proof: If $\mathit{opt} = 0$, then the Corollary is trivial. Assume $\mathit{opt} \geq 1$. Then

$$\begin{aligned}
\mathbf{Pr}(\mathit{algorithm\ fails}) &\leq \exp\left(\frac{-l(2q-1)}{2e^{1/\gamma}}\left(\frac{\gamma\,\mathit{opt}}{9(m-\mathit{opt})}\right)^{7d/\gamma}\right) && (4.16)\\
&\leq \exp\left(\frac{-7ld\,\mathit{opt}(2q-1)}{18(m-\mathit{opt})e^{1/(7d)}}\right) && (4.17)\\
&\leq \exp\left(\frac{-7ld(2q-1)}{18me^{1/7}}\right) && (4.18)\\
&\leq \exp\left(\frac{-ld(2q-1)}{3m}\right) && (4.19)\\
&\leq \delta. && (4.20)
\end{aligned}$$

This completes the proof. □

We can now take advantage of the following two theorems, which address learning in Valiant's PAC model.

Theorem 9 (Pitt and Valiant (1988)): *If $\mathcal{H} \subseteq \cup_n 2^{\mathbf{Q}^n}$ is properly PAC learnable, then there is a randomized polynomial time algorithm which solves the consistency problem for $\mathcal{H}$.*

Theorem 10 (Blumer, et al (1989)): *If $\mathcal{H} = \cup_n \mathcal{H}_n$, where $\mathcal{H}_n \subseteq \mathbf{Q}^n$ is properly PAC learnable, then there is a polynomial p such that for all $n \in \mathbf{N}$, $VCdim(\mathcal{H}_n) \leq p(n)$.*

Combining these with Corollary 8 we obtain the following.

Corollary 11: *Let $\mathcal{H}$ be a stratified tracking problem. Then if the corresponding learning problem is properly PAC learnable, $\mathcal{H}$ is efficiently trackable.*

Combining Corollary 8 with Theorem 5, we obtain the following result for halfspaces and hyperrectangles in particular. Let $HALFSPACES_n$ be the set of indicator functions for the following sets:

$$\{\{\bar{x} \in \mathbf{Q}^n : \bar{a} \cdot \bar{x} \geq b\} : \bar{a} \in \mathbf{Q}^n, b \in \mathbf{Q}\}. \tag{4.21}$$

Let $BOXES_n$ be the set of indicator functions for the set of axis parallel hyperrectangles in n-dimensional space, i.e.

$$\{\prod_{i=1}^{n} [a_i, b_i] : \bar{a}, \bar{b} \in \mathbf{Q}^n\}. \tag{4.22}$$

Corollary 12: *There is a constant $c > 0$ and there are efficient tracking algorithms for each of $\{HALFSPACES_n : n \in \mathbf{N}\}$ and $\{BOXES_n : n \in \mathbf{N}\}$ that (ϵ, Δ)-track these classes for*

$$\Delta \leq \frac{c\epsilon^2}{n^2 \log(1/\epsilon)}. \tag{4.23}$$

Finally, Kearns and Li (1988) showed that, loosely speaking, significantly improving the factor of approximation of our algorithm for minimizing disagreements for hyperrectangles (in particular, removing the dependence on d) would lead to corresponding improvements on the approximation algorithm for set cover, which has not been significantly improved since the 1970's. Nevertheless, it remains possible that, via other methods, one might obtain efficient algorithms that track these classes at rates even closer to optimal. The results of this section have recently been improved somewhat (Long, 1992), but the linear dependence on d remains.

5 Upper bounds on the tolerable amount of drift

In this section we prove upper bounds on the tolerable amount of drift for two commonly studied concept classes: halfspaces and axis-aligned rectangles. Our upper bounds show that the algorithm of Section 3 is within a log times a constant factor of optimal for each of these classes.

First, we will prove an upper bound for $BASIC_n$, the class of indicator functions for the following family of subsets of the unit interval:

$$\{\cup_{i=1}^{n} [i/n, (i + a_i)/n) : \bar{a} \in [0, 1]^n\}. \tag{5.1}$$

This class can be viewed as dividing the unit interval into n subintervals of equal length. Every concept in the class is the union of an initial segment from each of the subintervals. It is easy to see that $VCdim(BASIC_n) = n$.

Our argument for the upper bound on $BASIC_n$ uses ideas from earlier arguments by Ehrenfeucht, et al (1989) and Haussler, et al (1990) giving lower bounds on the probability of a mistake when predicting a stationary target function.

The intuition behind the argument is as follows. Suppose there is a water truck rolling down a section of dusty road at 10 kilometers per hour. Either the truck is empty or it is spraying water (unknown to us, but both possibilities are equally likely). Each minute a point on the road is picked at random and we predict whether or not the point is wet before looking at it. If the point has not yet been passed by the water truck, then we can safely predict that it is dry. If a previously picked point had already been passed by the water truck when it was picked, then we know whether or not the truck is spraying water and can always predict correctly. However, our prediction always has a 1/2 chance of being wrong on the first point which the water truck has passed. This idea can be extended to to n watertrucks (each of which is independently spraying or empty) on n different roads. Whenever a point on road i that has been passed by truck i is picked, and none of the previous points had been passed by truck i when they were picked, we will make a mistake with probability 1/2.

Theorem 13: *For all $n \in \mathbf{N}$, $BASIC_n$ is not (ϵ, Δ)-trackable if $\epsilon \leq 1/e^2$ and $\Delta \geq e^4\epsilon^2/n$.*

Proof: By contradiction. Assume that tracking strategy A (ϵ, Δ)-tracks $BASIC_n$ for some $0 < \epsilon \leq 1/e^2$, $n \in \mathbf{N}$, and $\Delta \geq e^4\epsilon^2/n$. Thus after seeing at least m_0 examples drawn from distribution D and labeled by any (Δ, D)-admissible sequence of targets, the probability that A makes a mistake on the next example is at most ϵ.

Without loss of generality, set $\Delta = e^4\epsilon^2/n$. With the restriction on ϵ, $\Delta \leq 1/n$ (and $n \leq 1/\Delta$). Also, since no non-degenerate class is (ϵ, Δ)-trackable if $\Delta > \epsilon$ and $\epsilon \leq 1/3$, we may assume that $\Delta \leq 1/e^2$.

Let $t = \lfloor\sqrt{n/\Delta}\rfloor$. Since $e \leq \sqrt{e^2 n} \leq \sqrt{n/\Delta}$, we get $\frac{2}{3}\sqrt{n/\Delta} \leq t \leq \sqrt{n/\Delta}$ and $et \leq n/\Delta$. These inequalities will be used at the end of the proof.

For each $\bar{z} \in \{0,1\}^n$ and $0 \leq i \leq t$, define $f_{\bar{z},i} \in BASIC_n$ as the indicator function for

$$\cup_{j=1}^{n} \, [j/n, (j + i\Delta z_j)/n). \tag{5.2}$$

Since $t \leq 1/\Delta$ (using $n \leq 1/\Delta$), every interval in the union has length at most $1/n$. Note that $f_{\bar{z},0}$ is the function mapping everything to 0. Choose m such that $m \geq t+1$ and $m \geq m_0$. Let $S(\bar{z})$ be the sequence of m elements of $BASIC_n$ defined by $S(\bar{z}) = (f_{\bar{z},0}, f_{\bar{z},0}, \ldots, f_{\bar{z},0}, f_{\bar{z},1}, f_{\bar{z},2}, \ldots, f_{\bar{z},t})$. Let U be the uniform distribution on $X = [0,1]$. One can easily verify that for all $\bar{z} \in \{0,1\}^n$, $S(\bar{z})$ is (Δ, U)-admissible.

Let E be the event that for a random $\bar{x} \in [0,1]^m$, x_m is the first "passed" point in its subinterval. More formally, $x_m - \dfrac{\lfloor nx_m\rfloor}{n} \leq \dfrac{t\Delta}{n}$ and for all $0 < i < t$, $x_{m-t+i} \notin \left[\dfrac{\lfloor nx_m\rfloor}{n}, \dfrac{\lfloor nx_m\rfloor}{n} + \dfrac{i\Delta}{n}\right]$.
For each $\bar{z} \in \{0,1\}^n, \bar{x} \in [0,1]^m$, $\sigma \in ?$, let $mistake(\bar{z}, \bar{x}, \sigma)$ be the event that

$$A(sam_{m-1}(S(\bar{z}), \bar{x}), x_m, \sigma) \neq f_{\bar{z},t}(x_m), \tag{5.3}$$

i.e. that strategy A incorrectly predicts the label of the mth example where σ represents the strategy's internal randomization. Finally, let U' be the uniform distribution over $\{0,1\}^n$. We have

$$\begin{aligned} &\mathbf{Pr}_{(\bar{x},\bar{z},\sigma)\in U^m \times U' \times \mathcal{U}}(mistake(\bar{z}, \bar{x}, \sigma)) \\ &\quad \geq \mathbf{Pr}(mistake(\bar{z}, \bar{x}, \sigma)|E)\mathbf{Pr}(E) = \tfrac{1}{2}\mathbf{Pr}(E) \end{aligned} \tag{5.4}$$

since, when given E, it is equally likely that $f_{\bar{z},t}(x_m)$ is 0 or 1, independent of the previous examples. Now,

$$\begin{aligned} \mathbf{Pr}(E) &= \mathbf{Pr}\left(x_m - \frac{\lfloor nx_m\rfloor}{n} \leq \frac{t\Delta}{n} \;\&\; \forall 0 < i < t, x_{m-t+i} \notin \left[\frac{\lfloor nx_m\rfloor}{n}, \frac{\lfloor nx_m\rfloor}{n} + \frac{i\Delta}{n}\right]\right) && (5.5) \\ &= t\Delta \prod_{i=1}^{t-1}(1 - \frac{\Delta i}{n}) && (5.6) \\ &\geq t\Delta \prod_{i=1}^{t-1} \exp\left(\frac{-\frac{\Delta i}{n}}{1 - \frac{\Delta i}{n}}\right) && (5.7) \end{aligned}$$

$$\begin{aligned}
&= t\Delta \exp\left(\sum_{i=1}^{t-1} \frac{-\frac{\Delta i}{n}}{1-\frac{\Delta i}{n}}\right) && && (5.8)\\
&\geq t\Delta \exp\left(\left(\frac{-\frac{\Delta}{n}}{1-\frac{\Delta t}{n}}\right)\frac{t^2}{2}\right) && && (5.9)\\
&\geq t\Delta \exp\left(-\frac{e}{2(e-1)}\frac{t^2\Delta}{n}\right) && \text{(since } t \leq n/(e\Delta)\text{)} && (5.10)\\
&\geq \frac{2}{3}\sqrt{n\Delta}\exp\left(-\frac{e}{2(e-1)}\right) && \text{(since } \tfrac{2}{3}\sqrt{n/\Delta} \leq t \leq \sqrt{n/\Delta}\text{)} && (5.11)
\end{aligned}$$

Noting that $\frac{2}{3}\exp\left(-\frac{e}{2(e-1)}\right) > \frac{2}{e^2}$ yields

$$\begin{aligned}
\mathbf{Pr}(\mathit{mistake}) &> \frac{\sqrt{n\Delta}}{e^2} && (5.12)\\
&> \epsilon. && (5.13)
\end{aligned}$$

Since

$$\mathbf{Pr}_{(\bar{x},\bar{z},\sigma)\in U^{m+1}\times U'\times\mathcal{U}}(\mathit{mistake}(\bar{z},\bar{x},\sigma)) > \epsilon, \qquad (5.14)$$

there is a $\bar{z}$ for which

$$\mathbf{Pr}_{(\bar{x},\sigma)\in U^{m+1}\times\mathcal{U}}(\mathit{mistake}(\bar{z},\bar{x},\sigma)) > \epsilon, \qquad (5.15)$$

contradicting the assumption that that A (ϵ,Δ)-tracks BASIC_n.□

Recall the definitions of $\mathit{HALFSPACES}_n$ and BOXES_n from the previous section.

The following theorem follows from the bounds for BASIC_n via a trivial embedding of BASIC_n into $\mathit{HALFSPACES}_n$ and a similar embedding of BASIC_{2n} into BOXES_n using a simplified version of the prediction preserving reductions (Pitt and Warmuth, 1990). The same embeddings were employed by Haussler, et al (1990). The details are omitted.

Theorem 14: *For all $\epsilon < 1/e^2$ and $n \in \mathbf{N}$, $\mathit{HALFSPACES}_n$ is not (ϵ,Δ)-trackable when $\Delta > e^4\epsilon^2/n$, and BOXES_n is not (ϵ,Δ)-trackable when $\Delta > e^4\epsilon^2/2n$.*

This theorem, along with the facts that the VC dimension of $\mathit{HALFSPACES}_n$ is $n+1$ and that of and BOXES_n is $2n$, establishes that the general purpose algorithm described in Section 3 is within a constant times a log factor of optimal for these two natural concept classes.

6 Conclusions

We have defined a learning model in which the target concept is allowed to change over time and discovered a general-purpose algorithm whose performance nearly matches our lower bounds (on at least two natural target classes). However this algorithm relies on a potentially expensive subroutine for minimizing disagreements within a constant factor. To combat this difficulty, we have found an efficient way to approximately minimize disagreements to within a factor that depends (linearly) on the VC-dimension. This gives us a second generic algorithm which, although not proven able to tolerate quite as much drift, is more likely to be computationally efficient (as it is for halfspaces, hyperrectangles, and any other target class which is properly PAC learnable).

Our algorithms are robust in the sense that they don't need to know the rate of drift Δ ahead of time, although attempting to achieve an accuracy ϵ amounts to an implicit assumption of an upper bound on Δ.

Although our results have usually been stated in terms of how much target motion can be tolerated, they can viewed in other ways. Bounds like "all $\Delta < c\epsilon^2/(d^2 \ln \epsilon)$ are tolerated" are easily converted to "the error rate, ϵ, is at most $c_\alpha d\Delta^{1/(2-\alpha)}$ for arbitrarily small α." Also, our bounds indicate how frequently one must sample to achieve a desired accuracy when given a bound on the continuous rate of target drift. This interpretation may be the more useful one.

Consider an assembly line process where the machines slowly drift out of alignment, gradually increasing the defect rate. One wants to sample the finished products in order to determine when an adjustment is required. It is often infeasible to inspect each item produced as the inspection process might be very expensive or even destroy the good. Thus a more complicated inspection plan indicating when to inspect and how to evaluate the inspection results is needed. The results in Section 3 are applicable to this problem.

Intuitively, the following approach seems as if it should lead to improved tracking algorithms. Instead of simply minimizing the number of disagreements with a suffix of the previous examples, an algorithm might weight previous examples with gradually decreasing nonnegative weights which sum to one. Then for each hypothesis h in the target class, the algorithm might use the sum of the weights of the examples with which h disagrees as the estimate of the probability that it will make a mistake on the next trial, then use the hypothesis which minimizes this, possibly more accurate, estimate. One wonders whether such an algorithm might significantly improve on the simple "minimize disagreements" algorithm analyzed in this paper.

It is easy to see how to alter our arguments to obtain results in a related model (often called "agnostic learning") in which the algorithm doesn't know a priori a class which contains each of the sequence of targets, and tries to predict nearly as well as possible using hypotheses in a certain class $\mathcal{H}$. More formally, suppose for a worst case sequence of concepts $f_1, f_2, ...$ (not necessarily in the hypothesis class $\mathcal{H}$), for each t we defined κ_t to be $\min_{h \in \mathcal{H}} \mathbf{Pr}(h(x) \neq f_t(x))$. It can be shown by modifying the proofs of Section 3, that for $\Delta \leq c\epsilon^3/(d \ln(1/\epsilon))$, an algorithm can achieve probability of mistake at most $\kappa_t + \epsilon$ for all large enough t (Helmbold and Long, 1991). One wonders whether these results can be improved.

Haussler (1991) has generalized the results of Blumer, et al (1989) to apply to learning in many frameworks, one of which is the learning of real valued functions. Using Haussler's results, the techniques of Section 3 can trivially be extended to apply to uniformly bounded classes of real valued functions (e.g., feed forward neural networks of a particular architecture which has one output node), where, in place of the Vapnik-Chervonenkis dimension, we use Pollard's (1984) *pseudo*-dimension, and instead of wanting to make the probability of mistake small, we want to make the expectation of the absolute value of the difference between our prediction and the truth small. In place of an algorithm for minimizing disagreements, we require an algorithm for minimizing the sum of absolute errors on a sample. It would be interesting to obtain results for more general loss functions, e.g. the square loss. Also, we have no general lower bounds for the tracking of real valued functions.

Other natural problems include: optimizing the constants and removing the $1/\ln \frac{1}{\epsilon}$ gap between our bounds on Δ.

Acknowledgements

We would like to thank David Haussler for many pointers to the literature, especially about exponential tail bounds, and Nicolo Cesa-Bianchi for proofreading an earlier draft of this work. We'd also like to thank the Machine Learning group at UC Santa Cruz in general for many valuable conversations about this work and related topics. David Helmbold was partially supported by a Regents Junior Faculty Fellowship. Phil Long was supported by ONR grant N00014-85-K-0454.

References

M. Anthony, N. Biggs, and J. Shawe-Taylor, (1990). The learnability of formal concepts. *The 1990 Workshop on Computational Learning Theory*, 246–257.

D. Angluin and L. Valiant, (1979). Fast probabilistic algorithms for Hamiltonion circuits and matchings. *Journal of Computer and System Sciences*, 18(2):155–193.

D. Aldous and U. Vazirani, (1990). A Markovian extension of Valiant's learning model. *Proceedings of the 31st Annual Symposium on the Foundations of Computer Science*, pages 392–396.

N. Abe and O. Watanabe, (1992). Polynomially sparse variations and reducibility among prediction problems. *IEICE Trans. Inf. & Syst.*, E75-D(4):449–458, 1992.

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

A. Ehrenfeucht, D. Haussler, M. Kearns, and L.G. Valiant, (1989). A general lower bound on the number of examples needed for learning. *Information and Computation*, 82(3):247–251.

D. Haussler, (1991). Decision theoretic generalizations of the PAC model for neural net and other learning applications. Technical Report UCSC-CRL-91-02, University of California at Santa Cruz.

D.P. Helmbold and P.M. Long, (1991). Tracking drifting concepts using random examples. *The 1991 Workshop on Computational Learning Theory*, pages 13–23.

D. Haussler, N. Littlestone, and M.K. Warmuth, (1988). Predicting $\{0,1\}$ functions on randomly drawn points. *Proceedings of the 29th Annual Symposium on the Foundations of Computer Science*, pages 100–109.

David Haussler, Nick Littlestone, and Manfred Warmuth, (1990). Predicting $\{0,1\}$-functions on randomly drawn points. Technical Report UCSC-CRL-90-54, University of California Santa Cruz. To appear in Information and Computation.

T. Hagerup and C. Rub, (1990). A guided tour of Chernov bounds. *Information Processing Letters*, 33:305–308.

M. Kearns and M. Li, (1988). Learning in the presence of malicious errors. *Proceedings of the 20th ACM Symposium on the Theory of Computation*, pages 267–279.

T. Kuh, T. Petsche, and R. Rivest, (1990). Learning time varying concepts. In *NIPS 3*. Morgan Kaufmann.

T. Kuh, T. Petsche, and R. Rivest, (1991). Mistake bounds of incremental learners when concepts drift with applications to feedforward networks. In *NIPS 4*. Morgan Kaufmann.

N. Littlestone, (1989). *Mistake Bounds and Logarithmic Linear-threshold Learning Algorithms.* PhD thesis, UC Santa Cruz.

P.M. Long, (1992). *Towards a more comprehensive theory of learning in computers.* PhD thesis, UC Santa Cruz.

N. Littlestone and M.K. Warmuth, (1989). The weighted majority algorithm. *Proceedings of the 30th Annual Symposium on the Foundations of Computer Science.*

D. Pollard, (1984). *Convergence of Stochastic Processes.* Springer Verlag.

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

L. Pitt and M.K. Warmuth, (1990). Prediction preserving reducibility. *Journal of Computer and System Sciences*, 41(3).

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

V.N. Vapnik, (1982). *Estimation of Dependencies based on Empirical Data.* Springer Verlag.

V.N. Vapnik, (1989). Inductive principles of the search for empirical dependences (methods based on weak convergence of probability measures). *The 1989 Workshop on Computational Learning Theory.*

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