On-line Learning of Smooth Functions of a Single Variable

Don Kimber
Information Systems Laboratory
Department of Electrical Engineering
Stanford University
Stanford, CA 94304
kimber@parc.xerox.com

Philip M. Long
Computer Science Department
Duke University
P.O. Box 90129
Durham, NC 27708
plong@cs.duke.edu

April 12, 1994

Abstract

We study the on-line learning of classes of functions of a single real variable formed through bounds on various norms of functions' derivatives. We determine the best bounds obtainable on the worst-case sum of squared errors (also "absolute" errors) for several such classes. We prove upper bounds for these classes of smooth functions for other loss functions, and prove upper and lower bounds in terms of the number of trials.

1 Introduction

We consider the learning of real-valued functions of a single $[0,1]$-valued variable in a model introduced by Mycielski [11], and independently by Littlestone and Warmuth [10]. A learning problem consists of a class $\mathcal{F}$ of such functions. We assume that a function $f \in \mathcal{F}$ is hidden from the learner, and that learning proceeds in *trials*, where in the tth trial, the learning algorithm receives $x_t \in [0,1]$ from the environment, is required to output a prediction $\hat{y}_t$ of $f(x_t)$, then finds out the value of $f(x_t)$. For each $p \geq 1$, the p-performance of a learning algorithm A for $\mathcal{F}$ on a finite sequence $\sigma = \langle x_t \rangle_{t \leq m} \in [0,1]$ and an $f \in \mathcal{F}$ is[1]

$$L_p(A, f, \sigma) = \sum_{t=2}^{m} |\hat{y}_t - f(x_t)|^p.$$

The p-performance of A on $\mathcal{F}$ is then defined to be

$$L_p(A, \mathcal{F}) = \sup_{f \in \mathcal{F}, \sigma \in \cup_m [0,1]^m} L_p(A, f, \sigma).$$

We will focus primarily on the choices $p \in \{1, 2\}$. Extending the terminology of [8], we define

$$\mathrm{opt}_p(\mathcal{F}) = \inf_A L_p(A, \mathcal{F}).$$

[1] Note that we begin summing the algorithm's errors on the second trial. This is not unreasonable, since the algorithm's performance on the first trial is not indicative of learning ability anyway. Furthermore, we could begin summing on the first trial if we assumed in addition that $f(0) = 0$.

We limit our attention to continuous functions that are piecewise twice differentiable (i.e., twice differentiable except on a finite set). Let's call such functions *well-behaved*.

We wish to model the intuition that, for many functions encountered in practice, similar inputs tend to yield similar outputs. Toward this end, for $q \in \{1, 2, \infty\}$, we will study the class $\mathcal{F}_q$ of well-behaved functions whose first derivatives have q-norm at most 1. Recall that, for $1 \leq q < \infty$, the q-norm of a function f defined on $[0, 1]$ is defined to be

$$\left(\int_0^1 |f(x)|^q dx \right)^{1/q},$$

and that the infinity norm of f is the limit, as q approaches infinity, of its q-norm. The infinity norm roughly corresponds to the maximum value of $|f(x)|$, and the one-norm, to the average, while the two-norm lies somewhere in between. Thus, $\mathcal{F}_\infty$ roughly corresponds to the class of functions that are never very steep, and $\mathcal{F}_1$ to the class of functions that are not very steep on average.

In this paper, we determine the value of $\mathrm{opt}_p(\mathcal{F}_q)$ for each $(p, q) \in \{1, 2\} \times \{1, 2, \infty\}$.

Our main negative result is that $\mathrm{opt}_1(\mathcal{F}_\infty) = \infty$. This result, loosely speaking, says that even the assumption that the hidden function *never* has slope greater than one is not sufficiently strong to enable an algorithm to obtain any finite bound on the sum of the absolute values of the differences between predictions and true values.

Our main positive result concerns the algorithm which at each trial linearly interpolates between previously seen function values, and extrapolates by predicting with the value of the hidden function at the nearest previously seen point.[2] We show that the worst-case sum of squared errors made by this algorithm while learning $\mathcal{F}_2$ is 1. A trivial lower bound establishes the fact that this algorithm is optimal for $\mathcal{F}_2$ with respect to the worst-case sum of squared errors, and therefore that $\mathrm{opt}_2(\mathcal{F}_2) = 1$.

Since, as is easily verified, the 1-norm of a function is at most its 2-norm which is in turn at most its ∞-norm, we have that $\mathcal{F}_\infty \subseteq \mathcal{F}_2 \subseteq \mathcal{F}_1$. Combining the first inclusion with the positive result above implies that $\mathrm{opt}_2(\mathcal{F}_\infty) \leq 1$. Again, a trivial lower bound shows that this is the best possible, and therefore that $\mathrm{opt}_2(\mathcal{F}_\infty) = 1$. Similarly, it follows from our main negative result that $\mathrm{opt}_1(\mathcal{F}_1) \geq \mathrm{opt}_1(\mathcal{F}_2) \geq \mathrm{opt}_1(\mathcal{F}_\infty) = \infty$. A simple argument establishes that $\mathrm{opt}_p(\mathcal{F}_1) = \infty$ for all $p \geq 1$.

We next show that

$$\mathrm{opt}_{1+\epsilon}(\mathcal{F}_\infty) \leq 1 + \frac{1}{(2 \ln 2)\epsilon}$$

for $0 < \epsilon \leq 1$. Combining this with the aforementioned results about $\mathcal{F}_\infty$, we may conclude that $\mathrm{opt}_p(\mathcal{F}_\infty) < \infty$ exactly when $p > 1$. For this upper bound, we analyze the algorithm which simply predicts with the value of the hidden function at the nearest previously seen element of the domain, which, though intuitively worse than the "linear interpolation" algorithm, is easier to analyze. We also prove that for $0 < \epsilon \leq 1$, we have

$$\mathrm{opt}_{1+\epsilon}(\mathcal{F}_2) \leq 2 + \frac{1}{(2 \ln 2)\epsilon}$$

which implies that $\mathrm{opt}_p(\mathcal{F}_2)$ is also finite exactly when $p > 1$.

Finally, we consider bounded length sequences of trials, showing that the sum of (unsquared) errors made by either of the above algorithms learning $\mathcal{F}_\infty$ and $\mathcal{F}_2$ respectively on trial sequences of length m is at most $e(1 + (\log_2 m)/2)$. We prove a lower bound of $\Omega(\sqrt{\log m})$ on the worst-case sum of unsquared errors necessary for learning $\mathcal{F}_2$ on sequences of m trials.

Our analyses can be extended to classes of functions defined on an arbitrary interval, and to classes formed through arbitrary bounds on the various norms of the derivatives. Furthermore, the algorithms we describe do not make use of knowledge of the endpoints of the interval, or of knowledge of how steep the target function tends to be. Therefore, we may even view our upper bounds as applying to arbitrary well-behaved functions of the entire real line, where the maximum magnitude of an element of the domain encountered in a sequence of trials, as well as the steepness of the target function, appears in the bound. Our results may also be generalized to functions whose range is vector-valued, by treating each component of the predictions and true values separately. We have stated the results in their present form to facilitate presentation of lower bounds, as well as to cut down on unnecessary notation, as we feel that the essence of the problems is captured in the simple cases.

[2] On the very first trial, it predicts arbitrarily, say with 0.

Faber and Mycielski [3] proved, using a different algorithm, that $\mathrm{opt}_2(\mathcal{F}_2) \leq 1$. This result amounts to a special case of a beautiful theorem about learning linear functionals defined on Hilbert spaces using a generalization of the Widrow-Hoff algorithm [6, 12], and their paper contains numerous other applications of their Hilbert space results. Nevertheless, we feel it is interesting that even the very simple linear interpolation algorithm is optimal for $\mathcal{F}_2$ with respect to worst-case on-line sums of squared errors. The difference in complexity of the algorithms is illustrated by the fact that the tth prediction of the linear interpolation algorithm trivially can be made in $O(\log t)$ time, whereas the best known bound on the time required for the algorithm of [3] is $O(t)$ [2]. In recent work pursued subsequently to this research, Cesa-Bianchi, Long and Warmuth [2] generalized the results of Faber and Mycielski to show that a modification of the algorithm of [3] was optimal in the model of their paper, in which a smooth function only approximately maps x_t's to y_t's.

Many statisticians, and, more recently, computational learning theorists (see e.g., [4] [1] [5]) have studied the induction of classes of functions obtained through smoothness constraints. The spirit of their work differs from ours in several ways. First, their theorems usually concern functions of potentially many real variables, where ours, at present, apply only to functions of a single real variable. On the other hand, the previous work usually involves use of probabilistic assumptions on the generation of the x_t's, for instance that they are drawn independently from a fixed distribution on whatever domain, whereas our results do not use such assumptions. These assumptions have enabled researchers to prove bounds on the expected "loss" on a particular trial. In worst-case models such as that considered here, such "instantaneous" bounds are clearly impossible (see [8]). Finally, in many cases, we are able to obtain upper and lower bounds that match, including constants, which is often not the case for the previously studied problems.

2 Some negative results

In this section, we describe several settings in which no algorithm can acheive any finite bound on the cumulative loss.

We begin by showing that $\mathrm{opt}_1(\mathcal{F}_\infty) = \infty$. In contrast, we will show in Section 3 that $\mathrm{opt}_2(\mathcal{F}_\infty) = 1$. In our analysis, it will be convenient to consider classes of functions defined on $[0, a]$ for $a > 0$, constrained by the values of the functions at 0 and a.

For $a, b \in [0, 1]$, define $\mathcal{G}_{a,b}$ to be the class of well-behaved functions g defined on $[0, a]$ for which $g(0) = 0$ and $g(a) = b$, with the further restriction that $g'(x) \leq 1$ for all x on which g' is defined.

The following lemmas may be easily verified, e.g., by using reductions between real-valued learning problems [9] to scale, translate and reflect appropriately.

Lemma 1 *For any $a, c > 0$, $\mathrm{opt}_1(\mathcal{G}_{ca,0}) = c\,\mathrm{opt}_1(\mathcal{G}_{a,0})$.*

Lemma 2 *Choose $a, b, c, d \in \mathbf{R}$. Let $\mathcal{H}$ be the class of well-behaved functions f from $[a, b]$ to $\mathbf{R}$ for which $f(a) = c$ and $f(b) = d$, which also have the property that $f'(x) \leq 1$ wherever f' is defined. Then*

$$\mathrm{opt}_1(\mathcal{H}) = \mathrm{opt}_1(\mathcal{G}_{|b-a|,|c-d|}).$$

Next, we reduce the problem of proving a lower bound for $\mathcal{G}_{a,b}$ to smaller subproblems.

Lemma 3 *If $0 \leq b \leq a/2$, then*

$$\mathrm{opt}_1(\mathcal{G}_{a,b}) \geq \frac{b}{2} + \mathrm{opt}_1(\mathcal{G}_{a/2,0}) + \mathrm{opt}_1(\mathcal{G}_{a/2,b}).$$

Proof: Choose an algorithm A for learning $\mathcal{G}_{a,b}$ and $\epsilon > 0$. Let $\hat{y}_1$ be A's first prediction, given that $x_1 = a/2$.

Assume as a first case that $\hat{y}_1 \geq b/2$. By Lemma 2 and the definition of opt_1, there exist $m_1, m_2 \in \mathbf{N}$, $x_2, \ldots, x_{m_1+1} \in [0, a/2]$, $x_{m_1+2}, \ldots, x_{m_1+m_2+1} \in [a/2, b]$, and well-behaved functions $f_1 : [0, a/2] \to \mathbf{R}$, $f_2 : [a/2, b] \to \mathbf{R}$ whose derivatives are never more than one where they are defined such that $f_1(a/2) = f_2(a/2) = 0$, and

$$\left(\sum_{t=1}^{m_1} |\hat{y}_{t+1} - f_1(x_{t+1})|\right) + \left(\sum_{t=1}^{m_2} |\hat{y}_{t+m_1+1} - f_2(x_{t+m_2+1})|\right) \geq (\mathrm{opt}_1(\mathcal{G}_{a/2,0}) - \epsilon) + (\mathrm{opt}_1(\mathcal{G}_{a/2,b}) - \epsilon).$$

Hence, if f is taken to be the union of f_1 and f_2, then $f \in \mathcal{G}_{a,b}$, and if $\sigma = \langle x_t \rangle_{t=1}^{m_1+m_2+1}$, then

$$L_1(A, f, \sigma) \geq (\mathrm{opt}_1(\mathcal{G}_{a/2,0}) - \epsilon) + (\mathrm{opt}_1(\mathcal{G}_{a/2,b}) - \epsilon).$$

The case in which $\hat{y}_1 \leq b/2$ is handled similarly, and the fact that $\epsilon > 0$ was chosen arbitrarily completes the proof. ∎

Using essentially the same proof, one can establish the following.

Lemma 4 *If $0 \leq b \leq 1/2$, then*

$$\mathrm{opt}_1(\mathcal{G}_{1,0}) \geq b + 2\mathrm{opt}_1(\mathcal{G}_{1/2,b}).$$

In the next lemma, $\mathrm{opt}_1(\mathcal{G}_{a,b})$ is bounded below by a suitable function of $\mathrm{opt}_1(\mathcal{G}_{1,0})$.

Lemma 5 *For $j \in \mathbf{N}$ and $b = 2^{-j}a$,*

$$\mathrm{opt}_1(\mathcal{G}_{a,b}) \geq \frac{jb}{2} + (a-b)\mathrm{opt}_1(\mathcal{G}_{1,0}). \tag{1}$$

Proof: By iterating Lemma 3, concentrating on the second part, we get

$$\mathrm{opt}_1(\mathcal{G}_{a,b}) \geq \frac{jb}{2} + \sum_{i=1}^{j} \mathrm{opt}_1(\mathcal{G}_{a/2^i,0}).$$

Applying Lemma 1, we get

$$\begin{aligned} \mathrm{opt}_1(\mathcal{G}_{a,b}) &\geq \frac{jb}{2} + \left(\sum_{i=1}^{j} a/2^i \right) \mathrm{opt}_1(\mathcal{G}_{1,0}) \\ &= \frac{jb}{2} + (a-b)\mathrm{opt}_1(\mathcal{G}_{1,0}). \end{aligned}$$

This completes the proof. ∎

We put these together to prove the main result of this section.

Theorem 6 $\mathrm{opt}_1(\mathcal{F}_\infty) = \infty$.

Proof: We will show that even for $\mathcal{G}_{1,0} \subseteq \mathcal{F}_\infty$, $\mathrm{opt}_1(\mathcal{G}_{1,0}) = \infty$.

Choose $b = 2^{-(j+1)}$ for some $j \in \mathbf{N}$. Then

$$\begin{aligned} \mathrm{opt}_1(\mathcal{G}_{1,0}) &\geq b + 2\mathrm{opt}_1(\mathcal{G}_{\frac{1}{2},b}) && \text{(Lemma 4)} \\ &\geq b + 2[j\frac{b}{2} + (\frac{1}{2} - b)\mathrm{opt}_1(\mathcal{G}_{1,0})] && \text{(Lemma 5)} \\ &= b + jb + (1-2b)\mathrm{opt}_1(\mathcal{G}_{1,0}). \end{aligned}$$

We can now solve this for $\mathrm{opt}_1(\mathcal{G}_{1,0})$ to get

$$\mathrm{opt}_1(\mathcal{G}_{1,0}) \geq (j+1)/2. \tag{2}$$

Since $\mathrm{opt}_1(\mathcal{F}_\infty) \geq \mathrm{opt}_1(\mathcal{G}_{1,0})$ and j was chosen arbitrarily, $\mathrm{opt}_1(\mathcal{F}_\infty) = \infty$. ∎

As discussed earlier, since $\mathcal{F}_\infty \subseteq \mathcal{F}_q$, $q \geq 1$, this theorem has the following corollary.

Corollary 7 $\mathrm{opt}_1(\mathcal{F}_q) = \infty$ *for all* $q \geq 1$.

We may fairly easily see that the assumption that the average value of the (absolute) slope is at most one is not strong enough for practically any positive results in our model.

Theorem 8 *If $p \in \mathbf{R}$, $p \geq 1$, $\mathrm{opt}_p(\mathcal{F}_1) = \infty$.*

Proof: The class $\mathcal{F}_1$ includes all continuous twice differentiable increasing functions with $f(0) = 0$ and $f(1) = 1$, since for such functions,

$$\int_0^1 |f'(x)|\, dx = \int_0^1 f'(x)\, dx = f(1) - f(0) = 1.$$

The adversary picks $x_1 = 1/2$ and then chooses $f(x_1) = 0$ or $f(x_1) = 1$, whichever gives greater error. Suppose $f(x_1) = 1$. Then the adversary picks $x_2 = 1/4$, and continues the same scheme. If $f(x_1) = 0$, the adversary picks $x_2 = 3/4$ and repeats, et cetera. At each trial the loss is at least $1/2^p$. Using longer and longer sequences of trials of this type, the total loss can be made arbitrarily large. ∎

3 Some positive results

In this section we prove that a very simple algorithm performs optimally with respect to sums of squared errors when the hidden function is in $\mathcal{F}_2$, establishing an alternative proof that $\mathrm{opt}_2(\mathcal{F}_2) = 1$. Loosely speaking, this result implies that the assumption that the average value of the square of the target function's derivative is at most 1 is strong enough for an algorithm to obtain finite worst case bounds on its cumulative squared error. We showed in Section 2 that $\mathrm{opt}_2(\mathcal{F}_1) = \infty$.

Suppose $S = \{(u_i, v_i) : 1 \leq i \leq m\}$ is a finite subset of $[0,1] \times \mathbf{R}$ such that

$$u_1 < u_2 < \cdots < u_m.$$

Define $f_S : [0,1] \to \mathbf{R}$ as follows: for all x, $f_\emptyset(x) = 0$, and

$$f_S(x) = \begin{cases} v_1 & \text{if } x \leq u_1 \\ v_i + \frac{(x-u_i)(v_{i+1}-v_i)}{u_{i+1}-u_i} & \text{if } x \in (u_i, u_{i+1}] \\ v_m & \text{if } x > u_m \end{cases}$$

if $|S| \geq 1$.

For $f : [0,1] \to \mathbf{R}$, define the *action* of f, denoted by $J[f]$, to be

$$J[f] = \int_0^1 f'(x)^2 dx. \tag{3}$$

Note that $||f'||_2 \leq 1$ exactly when $J[f] \leq 1$, and therefore that $\mathcal{F}_2$ can also be thought of as the set of functions whose action is at most 1. The following lemma concerning the function of minimum action subject to certain constraints is well known, and can be proved fairly easily, for instance, through application of an elementary result from the Calculus of Variations (see [7, Theorem 2.2][3]).

Lemma 9 *Choose $m \in \mathbf{N}$. Let $(u_1, v_1), ..., (u_m, v_m)$ be a sample. Let $S = \{(u_i, v_i) : 1 \leq i \leq m\}$. If f is a well-behaved function consistent with $(u_1, v_1), ..., (u_m, v_m)$, then*

$$J[f] \geq J[f_S].$$

The following lemma concerns the change in the action of f_S when we add an example to S.

Lemma 10 *Choose $m \in \mathbf{N}$. Let $(u_1, v_1), ..., (u_m, v_m)$ be a sample with $0 \leq u_1 < u_2 < \cdots < u_m \leq 1$. Let $S = \{(u_i, v_i) : 1 \leq i \leq m\}$. Choose an example $(x, y) \in [0,1] \times \mathbf{R}$. Then*

$$\begin{aligned} J[f_{S \cup \{(x,y)\}}] &\geq J[f_S] + \frac{(y - f_S(x))^2}{\min_i |x - u_i|} \\ &\geq J[f_S] + (y - f_S(x))^2. \end{aligned}$$

If there exists $1 \leq j \leq m$ such that $|x - u_j| = |x - u_{j+1}| = \min_i |x - u_i|$, then

$$J[f_{S \cup \{(x,y)\}}] = J[f_S] + \frac{2(y - f_S(x))^2}{\min_i |x - u_i|}. \tag{4}$$

Proof: The lemma is trivial if $x < u_1$ or $x > u_m$, and if there is a j for which $x = u_j$. Assume that there is a j such that $u_j < x < u_{j+1}$.

If $a = u_{j+1} - u_j$, $b = f(u_{j+1}) - f(u_j)$, $c = x_t - u_j$, and $e = (f_S(x_t) - f(x_t)) = (\hat{y}_t - f(x_t))$ (see Figure 1), we can easily see that

$$J[f_{S \cup \{(x_t, f(x_t))\}}] - J[f_S] = \left(\frac{(\frac{bc}{a} + e)^2}{c} + \frac{(b - (\frac{bc}{a} + e))^2}{a - c} \right) - \frac{b^2}{a} = \frac{ae^2}{c(a-c)} \tag{5}$$

yielding (4) in the case $c = a - c$. In general, (5) implies

$$J[f_{S \cup \{(x_t, f(x_t))\}}] - J[f_S] = \frac{ae^2}{\min\{c, a-c\} \max\{c, a-c\}} \geq \frac{e^2}{\min\{c, a-c\}},$$

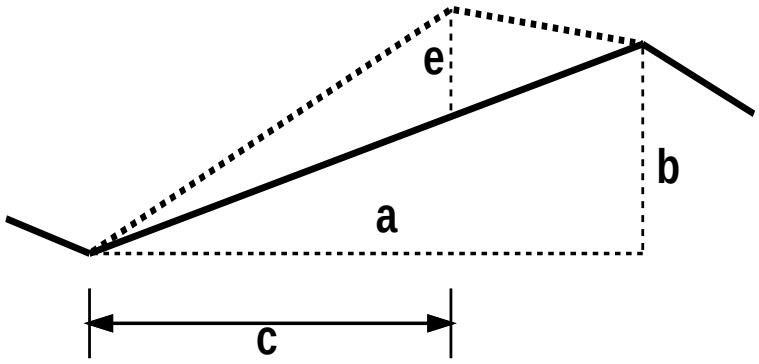

Figure 1: Change in J

completing the proof. □

Now we are ready for the learning result. Consider the learning algorithm LININT defined by

$$\text{LININT}(\emptyset, x_1) = 0$$

and

$$\hat{y}_t = \text{LININT}(((x_1, y_1), ..., (x_{t-1}, y_{t-1})), x_t) = f_{\{(x_1,y_1),...,(x_{t-1},y_{t-1})\}}(x_t)$$

for $t > 1$. That is, LININT linearly interpolates between previously seen points, and extrapolates using the value of the hidden function at the nearest previously seen element of the domain. Note that before each trial t, LININT can be thought of as formulating the hypothesis $f_{\{(x_1,y_1),...,(x_{t-1},y_{t-1})\}}$.

Theorem 11

$$L_2(\text{LININT}, \mathcal{F}_2) \leq 1.$$

Proof: Choose a target function $f \in \mathcal{F}_2$ and a sequence $x_1, x_2, ...$ of elements of $[0, 1]$. Assume without loss of generality that the x_t's are distinct.

By Lemma 10, we have that the action of the algorithm's hypothesis increases by at least $(\hat{y}_t - f(x_t))^2$ on each trial $t > 1$.

Since the function hypothesized after trial 1 is constant, and therefore has action 0, and since, by Lemma 9, the action of LININT's hypothesis is always at most that of the target function, which in turn is at most 1, we may conclude that $\sum_{t>1}(\hat{y}_t - f(x_t))^2 \leq 1$. ∎

We may apply this result to obtain an alternative proof of a result of Faber and Mycielski [3], who analyzed another, more complicated, algorithm for their upper bounds.

Theorem 12 ([3])

$$\text{opt}_2(\mathcal{F}_2) = 1.$$

Proof: The previous theorem implies that $\text{opt}_2(\mathcal{F}_2) \leq 1$. To see that $\text{opt}_2(\mathcal{F}_2) \geq 1$, consider an adversary which gives a first example of $(0, 0)$, and a second example of $(1, \pm 1)$, depending on whether an algorithm's prediction is positive or negative. This completes the proof. □

As discussed in the introduction, the fact that $\mathcal{F}_\infty \subseteq \mathcal{F}_2$, together with the same adversary argument as above, trivially yields the following.

Corollary 13 $\text{opt}_2(\mathcal{F}_\infty) = 1$.

[3] For those familiar with the Calculus of Variations, the Euler-Lagrange equation in this case is $f''(x) = 0$.

This corollary tells us that, with respect to worst-case cumulative squared error, the assumption that the derivative of a hidden function is never more than 1 doesn't give the learner any more power than the assumption that the average value of the square of the derivative is at most one.[4]

4 More general loss functions

Recall that in Section 3, we proved that $\mathrm{opt}_2(\mathcal{F}_\infty) = \mathrm{opt}_2(\mathcal{F}_2) = 1$, and in Section 2, we proved that $\mathrm{opt}_1(\mathcal{F}_\infty) = \mathrm{opt}_1(\mathcal{F}_2) = \infty$. This brings up a natural question: For which p are $\mathrm{opt}_p(\mathcal{F}_\infty)$ and $\mathrm{opt}_p(\mathcal{F}_2)$ finite? This question is resolved in this section: we show that $\mathrm{opt}_p(\mathcal{F}_\infty)$ and $\mathrm{opt}_p(\mathcal{F}_2)$ are finite whenever $p > 1$.

The following lemma will be useful in both analyses.

Lemma 14 *Choose a sequence $x_1, x_2, \ldots$ of elements of $[0,1]$. For each $t > 1$, let*

$$d_t = \min_{i<t} |x_t - x_i|.$$

If $p > 1$,

$$\sum_{t=1}^{\infty} d_t^p \leq 1 + 1/(2^p - 2).$$

Proof: Choose a sequence $x_1, x_2, \ldots$ of elements of $[0,1]$. Assume without loss of generality that the x_i's are distinct. For each $t \in \mathbf{N}$, let

$$S_t = \{x_i : i \leq t\} = \{u_{i,t} : i \leq t\},$$

where $u_{1,t} < u_{2,t} < \cdots u_{t,t}$ (the $u_{i,t}$'s are $\{x_1, \ldots, x_t\}$ in sorted order). For each t, let $s_t = u_{t,t} - u_{1,t}$.

First, we claim that

$$\sum_{t>1 : x_t \notin [u_{1,t-1}, u_{t-1,t-1}]} d_t^p \leq 1. \tag{6}$$

Choose a trial t for which $x_t < u_{1,t-1}$. In such a case, we have

$$s_t - s_{t-1} = d_t \geq d_t^p$$

since $d_t \leq 1$ and $p > 1$. Similarly, if $x_t > u_{t-1,t-1}$, then $s_t - s_{t-1} \geq d_t^p$. Since, trivially, s_t never decreases, and $0 \leq s_t \leq 1$, we have (6).

Next, we claim that

$$\sum_{t : x_t \in [u_{1,t-1}, u_{t-1,t-1}]} d_t^p \leq 1/(2^p - 2). \tag{7}$$

For each t, let

$$H_t = u_{1,t}^p + (1 - u_{t,t})^p + \sum_{i=1}^{t-1} (u_{i+1,t} - u_{i,t})^p.$$

Choose a trial t for which $x_t \in [u_{1,t-1}, u_{t-1,t-1}]$. Let i be such that $x_t \in (u_{i,t-1}, u_{i+1,t-1})$. Let $a = u_{i+1,t-1} - u_{i,t-1}$. Assume, as a first case, that x_t is closest to $u_{i,t-1}$ (the other case may be handled similarly). Then $d_t = x_t - u_{i,t-1} \leq a/2$. We have

$$H_t - H_{t-1} = d_t^p + (a - d_t)^p - a^p. \tag{8}$$

By differentiating, we may easily see that this expression, as a function of a, is decreasing when $a, d_t > 0$. Thus, it is maximized, subject to $a \geq 2d_t$, when $a = 2d_t$. Plugging into (8) and simplifying, we get

$$H_t - H_{t-1} \leq (2 - 2^p) d_t^p < 0.$$

[4] It would appear that the assumption that $f \in \mathcal{F}_\infty$ amounts to the slightly weaker assumption that the measure of $\{x : f'(x) > 1\}$ is zero. However, it is easy to see that the lower bound also applies to the smaller class of twice differentiable functions for which $f' \leq 1$ (indeed, to the extremely simple class consisting only of $f(x) = x$ and $g(x) = -x$). Thus, the the difficulty of learning this class in this model with the quadratic loss is the same as that of $\mathcal{F}_2$.

Since, trivially, $0 \leq H_t \leq 1$ for all t, and H_t never increases (on any trial), we have (7). Combining (6) and (7) yields the desired bound. ∎

We begin with $\mathcal{F}_\infty$. We will make use of the following simple lemma, whose proof is omitted. It establishes the fact that functions in $\mathcal{F}_\infty$ satisfy a Lipschitz condition.

Lemma 15 *If $f \in \mathcal{F}_\infty$, then for all $x, y \in [0,1]$, we have*

$$|f(x) - f(y)| \leq |x - y|.$$

A bound on $\mathrm{opt}_{1+\epsilon}(\mathcal{F}_\infty)$ follows immediately from the previous two lemmas. Recall that $\mathrm{opt}_2(\mathcal{F}_\infty) = 1$, and therefore $\mathrm{opt}_p(\mathcal{F}_\infty) = 1$ for all $p \geq 2$. For this reason, the theorem (also Theorem 17 below) is only interesting for $\epsilon < 1$.

Theorem 16 *If $\epsilon > 0$,*

$$\mathrm{opt}_{1+\epsilon}(\mathcal{F}_\infty) \leq 1 + \frac{1}{2^{1+\epsilon} - 2} \leq 1 + \frac{1}{(2 \ln 2)\epsilon}.$$

Proof: Consider the algorithm A which simply predicts with the function value at the nearest previously seen point (and arbitrarily on the first trial). Choose a sequence $x_1, ..., x_m$ of elements of $[0,1]$ and $f \in \mathcal{F}_\infty$. Let $\hat{y}_2, ..., \hat{y}_m$ be the predictions of this "nearest neighbor" algorithm on trials 2 through m, and let $p = 1 + \epsilon$. We have

$$\begin{aligned} \sum_{t=2}^{m} |\hat{y}_t - f(x_t)|^p &\leq \sum_{t=2}^{m} (\min_{i<t} |x_i - x_t|)^p \quad \text{(Lemma 15)} \\ &\leq 1 + \frac{1}{2^p - 2} \quad \text{(Lemma 14)} \end{aligned}$$

completing the proof of the first inequality of the theorem. The second follows immediately using the fact that for all x, $1 + x \leq e^x$. ∎

Next, we prove a very similar bound on $\mathrm{opt}_{1+\epsilon}(\mathcal{F}_2)$.

Theorem 17 *If $\epsilon > 0$,*

$$\mathrm{opt}_{1+\epsilon}(\mathcal{F}_2) \leq 2 + \frac{1}{2^{1+\epsilon} - 2} \leq 2 + \frac{1}{(2 \ln 2)\epsilon}.$$

Proof: Choose $\epsilon > 0$ and let $p = 1 + \epsilon$. Choose a sequence $x_1, ..., x_m$ of elements of $[0,1]$ and $f \in \mathcal{F}_\infty$. Let $\hat{y}_2, ..., \hat{y}_m$ be the predictions of LININT on trials 2 through m, and for each $t > 1$, let $d_t = \min_{i<t} |x_i - x_t|$, and let $e_t = |\hat{y}_t - f(x_t)|$. Applying Lemma 10, we have that the action of LININT's hypothesis increases by at least e_t^2/d_t on each trial. By Lemma 9, the action of LININT's hypothesis is always at most 1. Thus,

$$\sum_{t=2}^{m} e_t^2/d_t \leq 1. \tag{9}$$

Since, by Lemma 14, we have

$$\sum_{t=2}^{m} d_t^p \leq 1 + \frac{1}{2^p - 2}, \tag{10}$$

our analysis proceeds by breaking up the trials, and applying (9) to those trials where d_t is relatively small, and (10) to the trials where d_t is relatively large.

More specifically, we have

$$\begin{aligned} \sum_{t>1 : e_t \leq d_t} e_t^p &\leq \sum_{t>1 : e_t \leq d_t} d_t^p \\ &\leq 1 + \frac{1}{2^p - 2}, \end{aligned} \tag{11}$$

by (10). Also,

$$\begin{aligned}
\sum_{t>1:e_t>d_t} e_t^p &\leq \sum_{t>1:e_t>d_t} e_t \qquad \text{(Since } e_t \leq 1\text{)} \\
&< \sum_{t>1:e_t>d_t} e_t(e_t/d_t) \\
&= \sum_{t>1:e_t>d_t} e_t^2/d_t \\
&\leq 1,
\end{aligned}$$

by (9). Combining with (11) yields the first inequality. The second follows immediately using the fact that $1 + x \leq e^x$ for all x. ∎

5 Bounded-length trial sequences

In Section 2, we showed that $\text{opt}_1(\mathcal{F}_\infty) = \text{opt}_1(\mathcal{F}_2) = \infty$. In other words, we showed that finite bounds on the sum of absolute differences between predictions and true values could not be obtained for any algorithm using only the assumption that the hidden function was in $\mathcal{F}_\infty$, and therefore, for any algorithm using only the weaker assumption that the hidden function was in $\mathcal{F}_2$. Our adversaries used many trials, forcing small errors on each trial. The fact that $\text{opt}_2 < \infty$ for both these classes suggests that this behavior was necessary, since, as the error on a trial approaches 1, squaring the error has no effect.

If, in fact, any adversary which forces infinite cumulative error for algorithms learning $\mathcal{F}_\infty$ must force small errors on each trial, this is good news for the learner, since, even if one's total error is unbounded, if it is accumulating slowly, nontrivial learning is taking place.

In this section, we show that, indeed, the "nearest neighbor" algorithm studied in the previous section accumulates error slowly while learning $\mathcal{F}_\infty$. We show that on any sequence of m trials consistent with a function in $\mathcal{F}_\infty$, the sum of unsquared errors made by the nearest neighbor algorithm is $O(\log m)$. We also show that the "linear interpolation" algorithm studied in Section 3 achieves the same bound on its cumulative (unsquared) error on any sequence of m trials consistent with a function in $\mathcal{F}_2$.

For a class $\mathcal{F}$ of functions from $[0, 1]$ to $\mathbf{R}$, define

$$\text{opt}_1(\mathcal{F}, m) = \inf_A \sup_{f \in \mathcal{F}, \sigma \in [0,1]^m} L_1(A, f, \sigma)$$

where A ranges over learning algorithms.

Both proofs make use of the following inequality, which follows immediately by the standard convexity argument.

Lemma 18 *For any $n \in \mathbf{N}, p > 1, \vec{x} \in \mathbf{R}^n$,*

$$||\vec{x}||_1 \leq n^{1-1/p} ||\vec{x}||_p.$$

We begin with $\mathcal{F}_\infty$.

Theorem 19 *For all $m \geq 3$,*

$$\text{opt}_1(\mathcal{F}_\infty, m) \leq e\left(1 + \frac{\log_2 m}{2}\right).$$

Proof: Choose $x_1, ..., x_m$, and $f \in \mathcal{F}_\infty$. We claim that if A is the nearest neighbor algorithm, then

$$L_1(A, f, \sigma) \leq e\left(1 + \frac{\log_2 m}{2}\right).$$

Let $\hat{y}_1, ..., \hat{y}_m$ be the sequence of predictions made by A. Let $\vec{r} \in \mathbf{R}^m$ be defined by

$$\vec{r} = (|\hat{y}_1 - f(x_1)|, ..., |\hat{y}_m - f(x_m)|).$$

Choose $\epsilon > 0$. By Theorem 16, we have

$$||\vec{r}||_{1+\epsilon} \leq \left[1 + \frac{1}{(2\ln 2)\epsilon}\right]^{1/(1+\epsilon)}.$$

Applying Lemma 18, we have

$$||\vec{r}||_1 \leq m^{\epsilon/(1+\epsilon)} \left[1 + \frac{1}{(2\ln 2)\epsilon}\right]^{1/(1+\epsilon)}.$$

Suppose $\epsilon = 1/(\ln m - 1)$. Then

$$\begin{aligned} ||\vec{r}||_1 &\leq m^{\frac{1}{\ln m}} \left(1 + \frac{\ln m - 1}{2\ln 2}\right)^{\frac{\ln m - 1}{\ln m}} \\ &= e\left(1 + \frac{\ln m - 1}{2\ln 2}\right)^{\frac{\ln m - 1}{\ln m}} \\ &\leq e\left(1 + \frac{\ln m}{2\ln 2}\right) \end{aligned}$$

This completes the proof. □

With minor modifications, the above argument, together with Theorem 17, yields the following.

Theorem 20 *For $m \geq 3$,*

$$\mathrm{opt}_1(\mathcal{F}_2, m) \leq e\left(2 + \frac{\log_2 m}{2}\right)$$

We also have the following lower bound.

Theorem 21 *For $m \in \mathbf{N}$, $m \geq 2$,*

$$\mathrm{opt}_1(\mathcal{F}_2, m) \geq \frac{\sqrt{\lfloor \log_2 m \rfloor}}{2}.$$

Proof: Let $k = \lfloor \log_2 m \rfloor$. To ease the notation, in this proof we will number trials from 0. Let $x_0 = 1$. Choose an algorithm A for learning $\mathcal{F}_2$. For $i \in \mathbf{N}, j \in \mathbf{Z}, 0 \leq j \leq 2^{i-1} - 1$, let

$$x_{2^i + j} = \frac{1}{2^i} + \frac{j}{2^{i-1}}.$$

For example,

$$x_1 = 1/2, x_2 = 1/4, x_3 = 3/4, x_4 = 1/8, x_5 = 3/8, x_6 = 5/8, \ldots.$$

We will construct a function $f \in F_2$ such that if $\sigma = (x_0, \ldots, x_{m-1})$, $L_1(A, f, \sigma) \geq \frac{\sqrt{k}}{2}$. Define $f_0, f_1, \ldots, f_{2^k - 1}$ and $y_1, \ldots, y_{2^k - 1}$ inductively as follows. Let $f_0 \equiv 0$. Consider trials 2^{i-1} through $2^i - 1$ to be part of stage i. For each trial t in stage $i \leq k$, define y_t to be

$$f_{t-1}(x_t) \pm \frac{1}{2^i \sqrt{k}},$$

whichever is furthest from $\hat{y}_t$, and let f_t be the function which linearly interpolates $\{(0,0), (1,0)\} \cup \{(x_s, y_s) : s \leq t\}$. Let $f = f_{2^k - 1}$.

First, for each trial t in stage $i \leq k$,

$$|\hat{y}_t - f(x_t)| = |\hat{y}_t - y_t| \geq \frac{1}{2^i \sqrt{k}}.$$

Hence,

$$\begin{aligned} \sum_{t=1}^{m-1} |\hat{y}_t - f(x_t)| &\geq \sum_{i=1}^{k} 2^{i-1} \left(\frac{1}{2^i \sqrt{k}}\right) \\ &= \frac{\sqrt{k}}{2}. \end{aligned}$$

All that remains is to show that $f \in \mathcal{F}_2$. By (4) of Lemma 10, for all t in stage $i \leq k$,

$$J[f_t] - J[f_{t-1}] = 2\frac{\left(\frac{1}{2^i\sqrt{k}}\right)^2}{2^{-i}} = \frac{1}{k2^{i-1}}.$$

Therefore, since $J[f_0] = 0$,

$$\begin{aligned} J[f] &= \sum_{t=1}^{2^k-1} J[f_t] - J[f_{t-1}] \\ &= \sum_{i=1}^{k} 2^{i-1}\left(\frac{1}{k2^{i-1}}\right) \\ &= 1, \end{aligned}$$

completing the proof. ∎

6 Acknowledgements

We'd like to thank Ethan Bernstein, Tom Cover, David Haussler, Wolfgang Maass, Erik Ordentlich, Shang-Hua Teng and Manfred Warmuth for helpful conversations about this research, and Nicolò Cesa-Bianchi, Phil Chou, Yoav Freund, Les Niles, Madhukar Thakur and Lynn Wilcox for their comments on earlier drafts of this work. Don Kimber was supported by NSF Grant IRI-87-19595. Phil Long was supported by a UCSC Chancellor's dissertation-year fellowship, a Lise Meitner Postdoctoral Fellowship from the FWF of the Austrian Government, and by AFOSR grant F49620–92–J–0515.

References

[1] A. Barron. Approximation and estimation bounds for artificial neural networks. *The 1991 Workshop on Computational Learning Theory*, 1991.

[2] N. Cesa-Bianchi, P.M. Long, and M.K. Warmuth. Worst-case quadratic loss bounds for prediction using linear functions and gradient descent. Technical Report ucsc-crl-93-36, UC Santa Cruz, 1993. A preliminary version of this report appeared in the 1993 *Workshop on Computational Learning Theory.*

[3] V. Faber and J. Mycielski. Applications of learning theorems. *Fundamenta Informaticae*, 15(2):145–167, 1991.

[4] W. Hardle. *Smoothing Techniques.* Springer Verlag, 1991.

[5] D. Haussler. Generalizing the PAC model: sample size bounds from metric dimension-based uniform convergence results. *Proceedings of the 30th Annual Symposium on the Foundations of Computer Science*, 1989.

[6] S. Kaczmarz. Angenaherte Auflösung von systemen linearer gleichungen. *Bull. Acad. Polon. Sci. Lett. A*, 35:355–357, 1937.

[7] G. Leitmann. *The Calculus of Variations and Optimal Control.* Plenum Press, 1981.

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

[9] N. Littlestone, P.M. Long, and M.K. Warmuth. On-line learning of linear functions. *Proceedings of the 23rd ACM Symposium on the Theory of Computation*, pages 465–475, 1991.

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

[11] J. Mycielski. A learning algorithm for linear operators. *Proceedings of the American Mathematical Society*, 103(2):547–550, 1988.

[12] B. Widrow and M.E. Hoff. Adaptive switching circuits. *1960 IRE WESCON Conv. Record*, pages 96–104, 1960.