Worst-case Quadratic Loss Bounds for a Generalization of the Widrow-Hoff Rule

Nicolò Cesa-Bianchi*
Università di Milano
Via Comelico 39/41
Milano, ITALY 20135
cesabian@ghost.dsi.unimi.it

Philip M. Long†
IGI, TU Graz
Klosterwiesgasse 32/2
A-8010 Graz, AUSTRIA
plong@igi.tu-graz.ac.at

Manfred K. Warmuth‡
Computer Science Department
UC Santa Cruz
Santa Cruz, CA 95064 USA
manfred@cse.ucsc.edu

Abstract

We prove worst-case bounds on the sum of squared errors incurred by a generalization of the classical Widrow-Hoff algorithm to inner product spaces. We describe applications of this result to obtain worst-case agnostic learning results for classes of smooth functions and of linear functions.

1 Introduction

In this paper, we assume that learning proceeds in a sequence of trials. At trial number t the learning algorithm

- is presented with an *instance* x_t chosen from some domain $\mathcal{X}$,
- is required to return a real number $\hat{y}_t$,
- and then receives a real number y_t from the environment which we interpret as the truth.

The square loss of an algorithm over a sequence of m trials is $\sum_{t=1}^{m}(\hat{y}_t - y_t)^2$. A critical aspect of this model is that when the algorithm is making its prediction $\hat{y}_t$ for the tth instance x_t, it has access to pairs (x_s, y_s) *only for* $s < t$.

*Part of this research was done while this author was visiting UC Santa Cruz partially supported by the "Progetto finalizzato sistemi informatici e calcolo parallelo" of CNR under grant 91.00884.69.115.09672 (Italy).

†Supported by a Lise Meitner Fellowship from the Fonds zur Förderung der wissenschaftlichen Forschung (Austria). Part of this research was done while this author was at UC Santa Cruz supported by a UCSC Chancellor's dissertation-year fellowship.

‡Supported by ONR grant NO0014-91-J-1162 and NSF grant IRI-9123692.

ACM COLT '93 /7/93/CA, USA
© 1993 ACM 0-89791-611-5/93/0007/0429...$1.50

We adopt a worst-case outlook, following [Daw84, Vov90, LW91, LLW91, FMG92, MF92, CFH+93] and many others, assuming nothing about the environment of the learner, in particular the pairs $(x_1, y_1), \ldots, (x_m, y_m)$. Our results can be loosely interpreted as having the following message: "To the extent that the environment is friendly, our algorithms have small total loss." Of course, the strength of such results depends on how "friendly" is formalized. For the most general results of this paper (described in Section 3), the domain $\mathcal{X}$ is assumed to be a (real) vector space.[1] To formalize "friendly," we make use of the general notion of an inner product $(\cdot,\cdot)$, which is any function from $\mathcal{X} \times \mathcal{X}$ to $\boldsymbol{R}$ that has certain properties (see Section 2 for a list).

Typically, we express the bounds on the loss of our algorithms as a function of $\inf \sum_t((\mathbf{w}, \mathbf{x}_t) - y_t)^2$, where the infimum is taken over all $\mathbf{w}$ whose norm $\sqrt{(\mathbf{w}, \mathbf{w})}$ is bounded by a parameter. In many cases we can even bound the additional loss of the algorithm over the above infimum similarly to the additional loss bounds of [CFH+93] obtained in a simpler setting. Our bounds are all worst case in the sense that they hold for all sequences of pairs (x_t, y_t). (In some cases we assume the norm of the x_t is bounded by a second parameter.)

The inner product formalization is very general. One of the simplest inner products may be defined as follows in the case that $\mathcal{X} = \boldsymbol{R}^n$ for some n:

$$(\mathbf{u}, \mathbf{v}) = \sum_{i=1}^{n} u_i v_i = \mathbf{u} \cdot \mathbf{v}.$$

Notice that for any inner product space $\langle \mathcal{X}, (\cdot,\cdot) \rangle$, for any $\mathbf{w} \in \mathcal{X}$, we obtain a natural function $f_{\mathbf{w}}$ from $\mathcal{X}$ to $\boldsymbol{R}$ by defining

$$f_{\mathbf{w}}(\mathbf{x}) := (\mathbf{w}, \mathbf{x}). \tag{1}$$

Faber and Mycielski [FM91] noted that two natural classes of functions, a class of smooth functions of a single real variable, and the class of linear functions, can be defined in this manner. Both classes have been heavily studied in Statistics [Har91] (however, with probabilistic assumptions). Thus, general results for learning classes

[1] The general results will hold for finite and infinite dimensional vector spaces.

of functions from $\mathcal{X}$ to $\boldsymbol{R}$ defined by varying the $\mathbf{w}$ in (1) can be applied in a variety of circumstances. Faber and Mycielski then proved bounds on $\sum_t (\hat{y}_t - y_t)^2$ under the assumption that there was a $\mathbf{w} \in \mathcal{X}$ for which for all t, $y_t = (\mathbf{w}, \mathbf{x}_t)$, and described some applications of this result for learning classes of smooth functions. Mycielski [Myc88] had already treated the special case of linear functions. The algorithm they analyzed for this "noise-free" case was a generalization of the what is generally known as the Widrow-Hoff (**WH**) algorithm[2] [WH60] (defined in Section 3). In this paper we analyze the behavior of the **WH** algorithm in the case in which there isn't necessarily a $\mathbf{w}$ for which for all t, $y_t = (\mathbf{w}, \mathbf{x}_t)$. Faber and Mycielski [FM91] also studied this case, but their algorithms made use of side information which, in this paper, we assume is not available.

The algorithm studied in this paper may viewed as follows [MS91].[3] It maintains an element $\hat{\mathbf{w}}$ of $\mathcal{X}$ as its hypothesis which is updated between trials. For each t, let $\hat{\mathbf{w}}_t$ be the hypothesis before trial t. Then the initial hypothesis is the zero vector. After each trial t, there is a set S_t of elements $\mathbf{w}$ of $\mathcal{X}$ for which $(\mathbf{w}, \mathbf{x}_t) = y_t$. Intuitively, our hypothesis would like to be more like the elements of S_t, since we are banking on there being a nearly functional relationship $f_{\mathbf{w}}$ between the $\mathbf{x}_s$'s and the y_s's. It does not want to change *too* much, however, because the example $(\mathbf{x}_t, y_t)$ may be misleading. The generalized **WH** algorithm "takes a step" in the direction of the element of S_t which is closest to $\hat{\mathbf{w}}_t$ (using the natural notion of the distance between elements of an inner product space). In the presence of noise, a critical question is how big of a step to take in this direction. In this paper, we provide good worse case bounds for different answers to this question.

As an example consider the case that $\mathcal{X} = \boldsymbol{R}^n$ and $(\mathbf{u}, \mathbf{v})$ is defined to be $\sum_{i=1}^n u_i v_i$. Thus the $f_{\mathbf{w}}$'s are linear functions and the algorithm studied here is one of the simplest examples of gradient descent, an algorithm design technique which has achieved considerable practical success in more complicated hypothesis spaces, in particular neural networks [Tou89, Tou90, LMT91, MHL92]. Despite this success, there appears not to be a principled method for choosing the step size (learning rate) for gradient descent algorithms. In this paper we tune the step size with the goal of minimizing the worst case total squared loss over the best that can be obtained using elements from a given class of functions.

For any $\mathbf{v} \in \mathcal{X}$, $||\mathbf{v}|| = \sqrt{(\mathbf{v}, \mathbf{v})}$ measures the "size" of $\mathbf{v}$. We show that for all sequences $\mathbf{s} = \langle (\mathbf{x}_t, y_t) \rangle_t \in (\mathcal{X} \times \boldsymbol{R})^*$ and for all positive reals X, W, and E, if $\max_t ||\mathbf{x}_t|| \leq X$ and $L_W(\mathbf{s}) \leq E$, where

$$L_W(\mathbf{s}) = \inf_{||\mathbf{w}|| \leq W} \sum_t ((\mathbf{w}, \mathbf{x}_t) - y_t)^2,$$

then the generalized **WH** algorithm (tuned to X,W, and E) achieves the following

$$\sum_t (\hat{y}_t - y_t)^2 \leq L_W(\mathbf{s}) + 2(WX)\sqrt{E} + 2(WX)^2. \quad (2)$$

We then remove the assumption that a bound E on $L_W(\mathbf{s})$ is known, however we require that y_t's are in a certain range. We show that for all positive reals X and W and for all sequences $\mathbf{s} = \langle (\mathbf{x}_t, y_t) \rangle_t \in (\mathcal{X} \times [-WX, WX])^*$ such that $\max_t ||\mathbf{x}_t|| \leq X$, the sum of squared errors incurred on $\mathbf{s}$ by a variant of the generalized **WH** algorithm (tuned to the remaining parameters X and W) is at most

$$L_W(\mathbf{s}) + 12.89(WX)\sqrt{L_W(\mathbf{s})} + 8(WX)^2. \quad (3)$$

Notice that $\sum_t (\hat{y}_t - y_t)^2 - L_W(\mathbf{s})$ can be interpreted as the excess of the algorithm's total loss over the best that can be obtained using vectors $\mathbf{w}$ whose norms are at most W.

The above bounds are tight in a very strong sense. We show in Theorem 13 a lower bound of $L_W(\mathbf{s}) + 2(WX)\sqrt{E} + (WX)^2$ that holds for all X, W, and E, which in addition has the y_t's constrained in the range $[-WX, WX]$, and holds if all three parameters are given to the algorithm ahead of time. Notice that the upper bound (2) equals the lower bound plus $(WX)^2$. Our lower bounds essentially use adversary arguments. It is an open problem whether the lower bound can already be obtained when the examples are chosen independently at random according to a natural distribution. For more simple functions this was done in [CFH+93]: the lower bounds there are with respect to uniform distributions and the upper bounds which essentially meet the lower bounds are proven for the worst case as done in this paper.

We continue by giving the algorithm less information about the sequence. For the case when only a bound X on the norm of any $\mathbf{x}_t$ is known, we show that the generalized **WH** algorithm, tuned to X, achieves the following on the sum of its squared errors:

$$2.25 \inf_{\mathbf{w} \in \mathcal{X}} \left[(\max_t ||\mathbf{x}_t||^2) ||\mathbf{w}||^2 + \sum_t ((\mathbf{w}, \mathbf{x}_t) - y_t)^2 \right]$$

on any sequence $\mathbf{s} = \langle (\mathbf{x}_t, y_t) \rangle_t \in (\mathcal{X} \times \boldsymbol{R})^*$ such that $\max_t ||\mathbf{x}_t|| \leq X$. Note that this result shows how the **WH** algorithm is able to trade-off between the "size" of a $\mathbf{w}$, represented by its norm, and the extent to which $\mathbf{w}$ "fits" the data sequence, represented by the sum of squared errors incurred by $f_{\mathbf{w}}$.

Finally, with no assumptions on the environment of the learner, a further variant of the generalized **WH** algorithm has the following bound on the sum of squared errors

$$9 \inf_{\mathbf{w} \in \mathcal{X}} \left[(\max_t ||\mathbf{x}_t||^2) ||\mathbf{w}||^2 + \sum_t ((\mathbf{w}, \mathbf{x}_t) - y_t)^2 \right]$$

[2] Even though in the neural network community this algorithm is usually credited to Widrow and Hoff, it seems to have been first discovered by Kaczmarz [Kac37].

[3] Actually, this interpretation appears to be valid only in the slightly more restricted case that $\langle \mathcal{X}, (\cdot, \cdot) \rangle$ is a Hilbert space.

that holds on any sequence $\mathbf{s} = \langle (\mathbf{x}_t, y_t) \rangle_t \in (\mathcal{X} \times \boldsymbol{R})^*$.

We may apply our general bounds to a class of smooth functions of a single real variable, in the manner used by Faber and Mycielski [FM91] in the case that there is a perfect smooth function. The smoothness of a function is measured by the two-norm of its derivative. Of course, the derivative measures the steepness of a function at a given point, and therefore the two-norm (or any norm, for that matter) of the derivative measures the tendency of the function to be steep. When normalized appropriately, the two-norm of a function f's derivative can be seen to be between the average steepness of f and the f's maximum steepness. We show that if there is an (absolutely continuous) function $f : [0, \infty) \to \boldsymbol{R}$ with $f(0) = 0$ which tends not to be very steep and which tends to approximately map x_t's to the y_t's, and if the x_t's are not very big, then an application of the generalized **WH** algorithm to this case obtains good bounds on the sum of squared errors. More formally, we show that, for example, if the x_t's are taken from $[0, X]$, and if $f : [0, \infty) \to \boldsymbol{R}$ satisfies $||f'|| = \sqrt{\int_0^X f'(u)^2 \, du} \leq W$, and $\sum_t (f(x_t) - y_t)^2 \leq E$, then the predictions $\hat{y}_t$ of the special case of the general **WH** algorithm applied to this problem satisfy a bound on $\sum_t (\hat{y}_t - y_t)^2$ of

$$\inf_{||f'|| \leq W} \left[\sum_t (f(x_t) - y_t)^2 \right] + 2W\sqrt{XE} + 2W^2 X.$$

A bound of

$$\sum_t (\hat{y}_t - y_t)^2 \leq W^2 X$$

was proved by [FM91] in the case when $E = 0$. It is surprising that the time required for the algorithm we describe for this problem to make its tth prediction $\hat{y}_t$ is $O(t)$ in the uniform cost model provided that all past examples and predictions are saved. This is because, although the vector space in which we live in this application consists of functions and therefore the **WH** algorithm requires us to add functions, we can see that the functions that arise are piecewise linear, with the pieces being a simple functions of the past examples and predictions. In the case $E = 0$, however, there is an algorithm with an optimal bound on $\sum_t (\hat{y}_t - y_t)^2$ which computes its tth prediction in $O(\log t)$ time [KL92], raising the hope that there might be a similarly efficient robust algorithm.

We extend our result to apply to classes of smooth functions of $n > 1$ real variables studied by Faber and Mycielski [FM91] in the absence of noise. We also discuss a similar result for linear functions of n real variables. We show that, if $\max_t ||\mathbf{x}_t||_2 \leq X$, and if $\mathbf{w} \in \boldsymbol{R}^n$ satisfies $||\mathbf{w}||_2 \leq W$, and $\sum_t (\mathbf{w} \cdot \mathbf{x}_t - y_t)^2 \leq E$ (here $\mathbf{w} \cdot \mathbf{x}_t = \sum_i w_i x_{t,i}$ is the standard dot product and $|| \cdot ||_2$ is the standard Euclidian norm), then the sum of squared errors incurred by the generalized **WH** algorithm restricted to this case is bounded from the above by

$$\inf_{||\mathbf{w}||_2 \leq W} \left[\sum_t ((\mathbf{w} \cdot \mathbf{x}_t) - y_t)^2 \right] + 2WX\sqrt{E} + 2(WX)^2. \tag{4}$$

In this case, a bound of $(WX)^2$ was proved by Mycielski [Myc88] under the assumption that $E = 0$. We further show that all the applied upper bounds, even viewed as bounds on the excess of the algorithm's total loss over the loss of the best function of "size" at most W, are within a constant factor of optimal. Moreover, in some case the constant factor in front of the $\sqrt{E}$ term is the best possible.

Littlestone, Long and Warmuth [LLW91] proved bounds for another algorithm for learning linear functions in which the $\mathbf{x}_t$'s were measured using the infinity norm, and the $\mathbf{w}$'s were measured using 1-norm. Our bounds for the **WH** algorithm are smaller than those for some sequences. The algorithm of [LLW91] does not appear to generalize as did the **WH** algorithm, and therefore those techniques do not appear to be as widely applicable.

One of the main problems with gradient descent is that it motivates a learning rule but does not give any method for choosing the step size. Our results provide a method for setting the learning rate essentially optimally when learning linear functions (or other function classes defined by an inner product). An exciting research direction is to investigate to what extent the methods of this paper can be applied to analyze other simple gradient descent learning algorithms.

Our methods should also be applied to the batch setting where the whole sequence of examples is given to the learner at once and the goal of learning is to find the function that minimizes the sum of the squared errors. In the case of linear functions this can be solved directly using the linear least squares method (which might be considered to be too computationally expensive). Tuning the learning rate might assure provably fast convergence of a WH-style algorithm for function classes defined by inner products.

2 Preliminaries

Let $\boldsymbol{N}$ denote the positive integers and $\boldsymbol{R}$ denote the reals.

Each prediction of an on-line learning algorithm is determined by the previous examples and the current instance. Associated with an on-line learning algorithm A we define a mapping of the same name from $(\mathcal{X} \times \boldsymbol{R})^* \times \mathcal{X}$ to $\boldsymbol{R}$.

Fix $\mathcal{X}$ and a learning algorithm A. For a finite sequence $\mathbf{s} = \langle (x_t, y_t) \rangle_{1 \leq t \leq m}$ of examples we then have that the *prediction* $\hat{y}_t$ of A on the t-th *trial* satisfies

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

and we call $\hat{y}_1, \ldots, \hat{y}_m$ *the sequence of A's on-line predictions for* $\mathbf{s}$.

An *inner product space* (sometimes called a pre-Hilbert space since the imposition of one more assumption yields the definition of a Hilbert space) consists of a (real) vector space $\mathcal{X}$ and a function $(\cdot,\cdot)$ (called an *inner product*) from $\mathcal{X}\times\mathcal{X}$ to $\boldsymbol{R}$ that satisfies the following for all $\mathbf{u},\mathbf{v},\mathbf{x}\in\mathcal{X}$ and $\kappa\in\boldsymbol{R}$:

1. $(\mathbf{u},\mathbf{v})=(\mathbf{v},\mathbf{u})$
2. $(\kappa\mathbf{u},\mathbf{v})=\kappa(\mathbf{u},\mathbf{v})$
3. $(\mathbf{u}+\mathbf{v},\mathbf{x})=(\mathbf{u},\mathbf{x})+(\mathbf{v},\mathbf{x})$
4. $(\mathbf{x},\mathbf{x})>0$ whenever $\mathbf{x}\neq 0$.

The last requirement can be dropped essentially without affecting the definition [You88, page 25]. For $\mathbf{x}\in\mathcal{X}$, the *norm* of $\mathbf{x}$, denoted by $||\mathbf{x}||$, is defined by $||\mathbf{x}||=\sqrt{(\mathbf{x},\mathbf{x})}$. (These definitions are taken from [You88].)

An example of an inner product is the dot product. For $\mathbf{x},\mathbf{y}\in\boldsymbol{R}^n$ for some positive integer n, the dot product of $\mathbf{x}$ and $\mathbf{y}$ is defined to be

$$\mathbf{x}\cdot\mathbf{y}=\sum_{i=1}^{n}x_iy_i.$$

The 2-norm (or Euclidian norm) of $\mathbf{x}\in\boldsymbol{R}^n$ is then defined to be

$$||\mathbf{x}||_2=\sqrt{\mathbf{x}\cdot\mathbf{x}}=\sqrt{\sum_{i=1}^{n}x_i^2}.$$

If f is a function from $\boldsymbol{R}$ to $\boldsymbol{R}$, we say that f is *absolutely continuous*[4] iff there exists a (Lebesgue measurable) function $g:\boldsymbol{R}\to\boldsymbol{R}$ such that for all $a,b\in\boldsymbol{R}$, $a\leq b$,

$$f(b)-f(a)=\int_a^b g(x)\,dx.$$

3 Upper bounds for the generalized Widrow-Hoff algorithm

In this section, we prove bounds on the worst case sum of squared errors made by a generalization of the **WH** algorithm (described in Figure 1). (Technically, Figure 1 describes a different learning algorithm for each initial setting of the "learning rate" η. For a particular η, we will refer to the associated learning algorithm as $\mathbf{WH}_\eta$, and we will use a similar convention throughout the paper.) For the remainder of this section, fix an inner product space $\langle\mathcal{X},(\cdot,\cdot)\rangle$. In what follows, we will analyze the **WH** algorithm and its variants starting from the case where only a bound on $\mathbf{x}_t$ is available to the learner ahead of time. We will then show how additional information can be exploited for tuning parameters and obtaining better worst-case bounds. Finally, we will prove a bound for the case where no assumptions are made on the environment of the learner.

[4]This is shown to be equivalent to a more technical definition in most Calculus texts.

Algorithm WH.
Input: Learning rate $\eta\geq 0$.

- Choose $\mathcal{X}$'s zero vector as initial hypothesis $\hat{\mathbf{w}}_1$.
- On each trial t:
  1. Get $\mathbf{x}_t\in\mathcal{X}$ from the environment.
  2. Predict with $\hat{y}_t=(\hat{\mathbf{w}}_t,\mathbf{x}_t)$.
  3. Get $y_t\in\mathcal{X}$ from the environment.
  4. Update the current hypothesis $\hat{\mathbf{w}}_t$ according to the rule

$$\hat{\mathbf{w}}_{t+1}=\hat{\mathbf{w}}_t+\eta(y_t-\hat{y}_t)\mathbf{x}_t.$$

Figure 1: Pseudo-code for algorithm **WH**.

3.1 Learning with instances of bounded size

We will make use of the following, which might be interpreted as determining the amount that $\mathbf{WH}_\eta$ learns from an error. The derivation is based on previous derivations for the **WH** algorithm (see, e.g. [DH73]).

Lemma 1 *Choose* $\mathbf{x},\mathbf{v},\mathbf{w}\in\mathcal{X},y\in\boldsymbol{R},\eta>0$. *Let* $\hat{y}=(\mathbf{v},\mathbf{x})$ *and* $\hat{\mathbf{w}}=\mathbf{v}+\eta(y-\hat{y})\mathbf{x}$. *Then*

$$\begin{aligned}&||\mathbf{v}-\mathbf{w}||^2-||\hat{\mathbf{w}}-\mathbf{w}||^2\\&\quad=(2\eta-\eta^2||\mathbf{x}||^2)(\hat{y}-y)^2-2\eta(y-\hat{y})(y-(\mathbf{w},\mathbf{x})).\end{aligned}\tag{5}$$

Proof: Let $\alpha=\eta(y-\hat{y})$. Then $\hat{\mathbf{w}}=\mathbf{v}+\alpha\mathbf{x}$. Thus

$$\begin{aligned}&||\hat{\mathbf{w}}-\mathbf{w}||^2\\&\quad=((\hat{\mathbf{w}}-\mathbf{w}),(\hat{\mathbf{w}}-\mathbf{w}))\\&\quad=((\mathbf{v}+\alpha\mathbf{x}-\mathbf{w}),(\mathbf{v}+\alpha\mathbf{x}-\mathbf{w}))\\&\quad=((\mathbf{v}-\mathbf{w}),(\mathbf{v}-\mathbf{w}))+(2\alpha\mathbf{x},(\mathbf{v}-\mathbf{w}))+\alpha^2(\mathbf{x},\mathbf{x}).\end{aligned}$$

This implies

$$\begin{aligned}&||\hat{\mathbf{w}}-\mathbf{w}||^2-||\mathbf{v}-\mathbf{w}||^2\\&\quad=2\alpha(\mathbf{x},(\mathbf{v}-\mathbf{w}))+\alpha^2(\mathbf{x},\mathbf{x})\\&\quad=2\alpha(\hat{y}-(\mathbf{w},\mathbf{x}))+\alpha^2(\mathbf{x},\mathbf{x})\\&\quad=2\alpha(\hat{y}-y)+2\alpha(y-(\mathbf{w},\mathbf{x}))+\alpha^2(\mathbf{x},\mathbf{x}).\end{aligned}$$

Expanding our definition of α,

$$\begin{aligned}&||\hat{\mathbf{w}}-\mathbf{w}||^2-||\mathbf{v}-\mathbf{w}||^2\\&\quad=-2\eta(\hat{y}-y)^2+2\eta(y-\hat{y})(y-(\mathbf{w},\mathbf{x}))\\&\qquad+\eta^2||\mathbf{x}||^2(y-\hat{y})^2\\&\quad=-(2\eta-\eta^2||\mathbf{x}||^2)(\hat{y}-y)^2+2\eta(y-\hat{y})(y-(\mathbf{w},\mathbf{x})),\end{aligned}$$

establishing (5). □

Next, we show that the performance of the **WH** algorithm degrades gracefully as the relationship to be modelled moves away from being $(\mathbf{w},\cdot)$ from some $\mathbf{w}\in\mathcal{X}$. We make use of the following easily verified approximation.

Lemma 2 *For all* $q,r\in\boldsymbol{R}$,

$$q(q-r)\geq\frac{1}{2}(q^2-r^2).$$

In the following Theorem, we show that if a learning algorithm knows a bound on $\max_t ||\mathbf{x}_t||$ before learning takes place, it can obtain good bounds on the sum of squared errors. We will remove the assumption of this knowledge later through application of standard doubling techniques. Throughout the paper, for all sequences $\mathbf{s} = \langle(\mathbf{x}_t, y_t)\rangle_t \in (\mathcal{X} \times \boldsymbol{R})^*$ and all $\mathbf{w} \in \mathcal{X}$, let

$$L_{\mathbf{w}}(\mathbf{s}) = \sum_t ((\mathbf{w}, \mathbf{x}_t) - y_t)^2,$$

and for all $W > 0$ let

$$L_W(\mathbf{s}) = \inf_{||\mathbf{w}|| \le W} L_{\mathbf{w}}(\mathbf{s}).$$

Theorem 3 *Choose $0 < \beta < 2$, and $\mathbf{s} = \langle(\mathbf{x}_t, y_t)\rangle \in (\mathcal{X} \times \boldsymbol{R})^+$. Let $\hat{y}_1, ..., \hat{y}_{|\mathbf{s}|}$ be the sequence of $\mathbf{WH}_{\beta/X^2}$'s on-line predictions for $\mathbf{s}$. Then,*

$$\sum_t (\hat{y}_t - y_t)^2 \le \inf_{\mathbf{w} \in \mathcal{X}} \left[\frac{X^2 ||\mathbf{w}||^2}{\beta(1-\beta/2)} + \frac{L_{\mathbf{w}}(\mathbf{s})}{(1-\beta/2)^2} \right]. \quad (6)$$

In particular, if $\beta = 2/3$,

$$\sum_t (\hat{y}_t - y_t)^2 \le 2.25 \inf_{\mathbf{w} \in \mathcal{X}} \left[X^2 ||\mathbf{w}||^2 + L_{\mathbf{w}}(\mathbf{s}) \right]. \quad (7)$$

Proof: Choose $\mathbf{w} \in \mathcal{X}$. If $\hat{\mathbf{w}}_1, \hat{\mathbf{w}}_2, \ldots, \hat{\mathbf{w}}_{m+1}$ is the sequence of $\mathbf{WH}_{\beta/X^2}$'s hypotheses, applying Lemma 1, we get

$$\begin{aligned} &\textstyle\sum_{t=1}^m \left(\left(\frac{2\beta}{X^2} - \frac{\beta^2 ||\mathbf{x}_t||^2}{X^4} \right) (\hat{y}_t - y_t)^2 \right. \\ &\qquad \left. - \frac{2\beta(y_t - \hat{y}_t)(y_t - (\hat{\mathbf{w}}, \mathbf{x}_t))}{X^2} \right) \\ &= \textstyle\sum_{t=1}^m (||\hat{\mathbf{w}}_t - \mathbf{w}||^2 - ||\hat{\mathbf{w}}_{t+1} - \mathbf{w}||^2) \\ &= ||\hat{\mathbf{w}}_1 - \mathbf{w}||^2 - ||\hat{\mathbf{w}}_{m+1} - \mathbf{w}||^2 \\ &\le ||\mathbf{w}||^2, \end{aligned}$$

since $\hat{\mathbf{w}}_1 = \vec{0}$ and $||\cdot||$ is non-negative. Since $0 < \beta < 2$ and $(\mathbf{x}_t, \mathbf{x}_t) = ||\mathbf{x}_t||^2 \le X^2$, this implies

$$\begin{aligned} &\textstyle\sum_{t=1}^m \left(\left(\frac{2\beta}{X^2} - \frac{\beta^2}{X^2} \right) (\hat{y}_t - y_t)^2 - \frac{2\beta(y_t - \hat{y}_t)(y_t - (\mathbf{w}, \mathbf{x}_t))}{X^2} \right) \\ &\le ||\mathbf{w}||^2. \end{aligned}$$

Thus,

$$\begin{aligned} &\textstyle\sum_{t=1}^m \left((\hat{y}_t - y_t)^2 - \frac{2\beta}{2\beta - \beta^2} |y_t - \hat{y}_t| \, |y_t - (\mathbf{w}, \mathbf{x}_t)| \right) \\ &\le \textstyle\frac{X^2 ||\mathbf{w}||^2}{2\beta - \beta^2}. \end{aligned}$$

Applying Lemma 2, we get

$$\begin{aligned} &\textstyle\sum_{t=1}^m \left((\hat{y}_t - y_t)^2 - \frac{4\beta^2}{(2\beta - \beta^2)^2} (y_t - (\mathbf{w}, \mathbf{x}_t))^2 \right) \\ &\le \textstyle\frac{2X^2 ||\mathbf{w}||^2}{2\beta - \beta^2}. \end{aligned}$$

Solving for $\sum_t (\hat{y}_t - y_t)^2$ yields

$$\begin{aligned} &\textstyle\sum_{t=1}^m (\hat{y}_t - y_t)^2 \\ &\le \textstyle\frac{2X^2 ||\mathbf{w}||^2}{2\beta - \beta^2} + \frac{4\beta^2}{(2\beta - \beta^2)^2} L_{\mathbf{w}}(\mathbf{s}) \\ &= \textstyle\frac{X^2 ||\mathbf{w}||^2}{\beta(1-\beta/2)} + \frac{1}{(1-\beta/2)^2} L_{\mathbf{w}}(\mathbf{s}), \end{aligned}$$

establishing (6). Formula (7) then follows immediately. □

If we would use $\mathbf{WH}_{\beta/||\mathbf{x}_t||^2}$ in trial t then Theorem 3 can be used to obtain a similar bound for the normalized square loss:

$$\sum_t \frac{(\hat{y}_t - y_t)^2}{||\mathbf{x}_t||^2} \le \inf_{\mathbf{w} \in \mathcal{X}} \left[\frac{||\mathbf{w}||^2}{\beta(1-\beta/2)} + \frac{L'_{\mathbf{w}}(\mathbf{s})}{(1-\beta/2)^2} \right],$$

where

$$L'_{\mathbf{w}}(\mathbf{s}) = \sum_t \frac{(f_{\mathbf{w}}(\mathbf{x}_t) - y_t)^2}{||\mathbf{x}_t||^2}.$$

This is sometimes a stronger bound than the bound (6) of the above theorem. In the case $L'_{\mathbf{w}}(\mathbf{s}) = 0$, a similar bound was proved by Faber and Mycielski [FM91].

3.2 Tuning β

The next result shows that, if certain parameters are known in advance, improved performance can be obtained by tuning β. We need a technical lemma first. Define the function $G : \boldsymbol{R}_+^3 \to (0, 2)$ by

$$G(E, W, X) = \begin{cases} 1 & \text{if } \frac{\sqrt{E}}{WX} < \frac{(\sqrt{3}-1)^2}{2\sqrt{3}} \\ 2 - 2\sqrt{\frac{\sqrt{E}}{\sqrt{E} + WX}} & \text{otherwise.} \end{cases}$$

Lemma 4 *Choose $E, W, X > 0$. Then*

$$\frac{(WX)^2}{\beta(1-\beta/2)} + \frac{E}{(1-\beta/2)^2} \le E + 2(WX)^2 + 2(WX)\sqrt{E} \quad (8)$$

whenever $\beta = G(E, W, X)$.

Proof. Notice that $0 < G(E, W, X) < 2$ for all $E, W, X > 0$. Notice then that (8) can be rewritten as

$$\begin{aligned} &y^2 + 2 + 2y - \tfrac{1}{2\alpha(1-\alpha)} - \tfrac{y^2}{(1-\alpha)^2} \ge 0 \\ &\iff \left[\tfrac{1}{(1-\alpha)^2} - 1 \right] y^2 - 2y + \tfrac{1}{2\alpha(1-\alpha)} - 2 \le 0 \\ &\iff \tfrac{1-1+2\alpha-\alpha^2}{(1-\alpha)^2} y^2 - 2y + \tfrac{1-4\alpha+4\alpha^2}{2\alpha(1-\alpha)} \le 0 \\ &\iff \tfrac{\alpha(2-\alpha)}{(1-\alpha)^2} y^2 - 2y + \tfrac{(2\alpha-1)^2}{2\alpha(1-\alpha)} \le 0 \end{aligned} \quad (9)$$

where

$$y = \frac{\sqrt{E}}{WX} \quad \text{and} \quad \alpha = \frac{\beta}{2}.$$

To prove the lemma is then sufficient to show that for all $y > 0$, (9) holds for $\alpha = G(E, W, X)/2$. To show that, we look at the following second degree equation in y

$$\frac{\alpha(2-\alpha)}{(1-\alpha)^2} y^2 - 2y + \frac{(2\alpha-1)^2}{2\alpha(1-\alpha)} = 0. \quad (10)$$

A sufficient condition for (10) to have real roots is

$$4 - 4\frac{\alpha(2-\alpha)}{(1-\alpha)^2} \frac{(2\alpha-1)^2}{2\alpha(1-\alpha)} \ge 0.$$

Simplifying the LHS of the above inequality, we can see that it is equivalent to

$$\begin{aligned} &\tfrac{2\alpha^3 - 6\alpha^2 + 3\alpha}{2(1-\alpha)^3} \ge 0 \\ &\iff 2\alpha^2 - 6\alpha + 3 \ge 0 \qquad \text{since } 0 < \alpha < 1 \\ &\iff \alpha \le \tfrac{3-\sqrt{3}}{2} \quad \text{or} \quad \alpha \ge \tfrac{3+\sqrt{3}}{2}. \end{aligned}$$

Since $0 < \alpha < 1$, we end up with the condition

$$\alpha \leq \frac{3-\sqrt{3}}{2} \tag{11}$$

which ensures the existence of real roots for (10). Therefore, since the coefficient of y^2 in (9) is positive, (9) holds whenever both (11) and

$$\begin{aligned} &\tfrac{(1-\alpha)^2}{\alpha(2-\alpha)} - \tfrac{(1-\alpha)^2}{\alpha(2-\alpha)}\sqrt{\tfrac{\alpha(2\alpha^2-6\alpha+3)}{2(1-\alpha)^3}} \\ &\leq y \\ &\leq \tfrac{(1-\alpha)^2}{\alpha(2-\alpha)} + \tfrac{(1-\alpha)^2}{\alpha(2-\alpha)}\sqrt{\tfrac{\alpha(2\alpha^2-6\alpha+3)}{2(1-\alpha)^3}} \end{aligned} \tag{12}$$

hold. The double inequality (12) is obviously satisfied for

$$y = \frac{(1-\alpha)^2}{\alpha(2-\alpha)} \tag{13}$$

which, solved for α, yields

$$\alpha = 1 - \sqrt{\frac{y}{y+1}}. \tag{14}$$

Moreover, since the derivative of the rhs of (13) is negative, we have that (13) implies

$$\alpha \leq \frac{3-\sqrt{3}}{2} \quad \Longleftrightarrow \quad y \geq \frac{(\sqrt{3}-1)^2}{2\sqrt{3}}.$$

Thus, inequality (9) holds for

$$\alpha = 1 - \sqrt{\frac{y}{y+1}} \quad \Longleftrightarrow \quad \beta = 2 - 2\sqrt{\frac{\sqrt{E}}{\sqrt{E}+WX}}$$

and

$$y \geq \frac{(\sqrt{3}-1)^2}{2\sqrt{3}}. \tag{15}$$

Hence, to complete the proof we must prove that inequality (9) is satisfied for $0 \leq y < \frac{(\sqrt{3}-1)^2}{2\sqrt{3}}$ and $\alpha = \frac{1}{2}$. Plugging this choice for α in (9) yields

$$\begin{aligned} &3y^2 - 2y \leq 0 \\ \Longleftrightarrow &3y - 2 \leq 0 \qquad \text{since } y > 0 \\ \Longleftrightarrow &y \leq \tfrac{2}{3}. \end{aligned}$$

Since

$$y < \frac{(\sqrt{3}-1)^2}{2\sqrt{3}} < \frac{2}{3}$$

is true, inequality (9) holds is this case as well and the proof is finished. □

Theorem 5 *For each $E, X, W \geq 0$, the algorithm $\mathbf{WH}_{G(E,W,X)/X^2}$ has the following properties.*

Choose $m \in \boldsymbol{N}$, $\mathbf{s} = \langle(\mathbf{x}_t, y_t)\rangle \in (\mathcal{X} \times \boldsymbol{R})^+$, such that $\max_t ||\mathbf{x}_t|| \leq X$, and $L_W(\mathbf{s}) \leq E$. Let $\hat{y}_1, \hat{y}_2, \ldots$ be the sequence of $\mathbf{WH}_{G(E,W,X)/X^2}$'s on-line predictions for $\mathbf{s}$. Then,

$$\sum_t (\hat{y}_t - y_t)^2 \leq L_W(\mathbf{s}) + 2WX\sqrt{E} + 2(WX)^2.$$

Algorithm G1.
Input $W, X \geq 0$, $z \geq 3.324$.

- For each $i = 0, 1, \ldots$
  - Let $k_i = z^i(WX)^2$.
  - Repeat
    1. Let h_t be $\mathbf{WH}_{G(k_i,W,X)/X^2}$'s prediction.
    2. Predict with

$$\hat{y}_t = \begin{cases} -WX & \text{if } h_t < WX \\ h_t & \text{if } h_t \in [-WX, WX] \\ WX & \text{otherwise.} \end{cases}$$

    until the total loss in this loop exceeds

$$k_i + 2(WX)\sqrt{k_i} + 2(WX)^2.$$

Figure 2: Pseudo-code for algorithm **G1**. Here **WH** is used as a subroutine and its learning rate is set using the function G defined in Section 3.2.

Proof. Choose $m \in \boldsymbol{N}$, $\mathbf{s} = \langle(\mathbf{x}_t, y_t)\rangle_{t \leq m} \in (\mathcal{X} \times \boldsymbol{R})^m$ for which $L_W(\mathbf{s}) \leq E$ and $\max_t ||\mathbf{x}_t||^2 \leq X$. By Theorem 3, for all β such that $0 < \beta < 2$, we have

$$\begin{aligned} &\textstyle\sum_{t=1}^m (\hat{y}_t - y_t)^2 \leq \inf_{\mathbf{w} \in \mathcal{X}} \left[\frac{X^2\|\mathbf{w}\|^2}{\beta(1-\beta/2)} + \frac{L_{\mathbf{w}}(\mathbf{s})}{(1-\beta/2)^2} \right] \\ &\leq \tfrac{(WX)^2}{\beta(1-\beta/2)} + \tfrac{L_W(\mathbf{s})}{(1-\beta/2)^2} \\ &= \tfrac{(WX)^2}{\beta(1-\beta/2)} + \tfrac{L_W(\mathbf{s})}{(1-\beta/2)^2} - L_W(\mathbf{s}) + L_W(\mathbf{s}) \\ &\leq \tfrac{(WX)^2}{\beta(1-\beta/2)} + \tfrac{E}{(1-\beta/2)^2} - E + L_W(\mathbf{s}) \end{aligned}$$

since $L_W(\mathbf{s}) \leq E$ and $0 < \beta < 2$. Applying Lemma 4 for $\beta = G(E, W, X)$, we then conclude

$$\begin{aligned} &\textstyle\sum_{t=1}^m (\hat{y}_t - y_t)^2 \\ &\leq E + 2WX\sqrt{E} + 2(WX)^2 - E + L_W(\mathbf{s}) \\ &= L_W(\mathbf{s}) + 2WX\sqrt{E} + 2(WX)^2 \end{aligned}$$

as desired. □

Below, we show that techniques from [CFH+93] may also be applied to obtain a $L_W(\mathbf{s}) + O(WX\sqrt{E} + (WX)^2)$ bound if only W and X are known. However, the delicate interplay between E and W (loosely speaking, increasing W decreases E) has so far prevented us from obtaining such a result without knowledge of any of the three parameters W, X, and E.

3.3 Loss bounds without knowledge of E

We now introduce algorithm **G1** for the case where a bound X on the $\mathbf{x}_t$'s and the range $[-WX, WX]$ of the y_t's are known ahead of time. In Figure 2, we sketch algorithm **G1** that is parametrized w.r.t. a variable z to be optimized later.

Theorem 6 *For each $X, W \geq 0$, algorithm **G1** has the following properties.*
Choose $\mathbf{s} = \langle(\mathbf{x}_t, y_t)\rangle \in (\mathcal{X} \times [-WX, WX])^+$ such

that $\max_t \|\mathbf{x}_t\| \leq X$ *Let* $\hat{y}_1, ..., \hat{y}_{|\mathbf{s}|}$ *be the sequence of* $\mathbf{G1}_{W,X,z}$*'s on-line predictions for* $\mathbf{s}$*. Then* $\sum_t (\hat{y}_t - y_t)^2$ *is at most*

$$L_W(\mathbf{s}) + 2\left(\frac{\sqrt{z}}{\sqrt{z-1}} + 2.01\right)(WX)\sqrt{(z-1)L_W(\mathbf{s})} + 8(WX)^2.$$

In particular, for $z = 3.324$,

$$\sum_t (\hat{y}_t - y_t)^2 \leq L_W(\mathbf{s}) + 12.89(WX)\sqrt{L_W(\mathbf{s})} + 8(WX)^2.$$

3.4 Predicting with no a priori information

In this section we remove all assumptions that the learner has prior knowledge. The proof (which is omitted) follows immediately from Theorem 3 via the application of standard doubling techniques.

Theorem 7 *For any* $0 < \beta < 2$*, there is a prediction algorithm* $\mathbf{G2}_\beta$ *with the following properties.*

Choose $\mathbf{s} = \langle (\mathbf{x}_t, y_t) \rangle_t \in (\mathcal{X} \times \boldsymbol{R})^m$*. Let* $\hat{y}_1, ..., \hat{y}_{|\mathbf{s}|}$ *be the sequence of* $\mathbf{G2}_\beta$*'s on-line predictions for* $\mathbf{s}$*. Then* $\sum_t (\hat{y}_t - y_t)^2$ *is at most*

$$\inf_{\mathbf{w} \in \mathcal{X}} \left[\frac{4(\max_t \|\mathbf{x}_t\|^2)\|\mathbf{w}\|^2}{\beta(1-\beta/2)} + \frac{L_{\mathbf{w}}(\mathbf{s})}{(1-\beta/2)^2} \right].$$

In particular, if $\beta = 4/3$,

$$\sum_t (\hat{y}_t - y_t)^2 \leq 9 \inf_{\mathbf{w} \in \mathcal{X}} \left[(\max_t \|\mathbf{x}_t\|^2)\|\mathbf{w}\|^2 + L_{\mathbf{w}}(\mathbf{s}) \right].$$

Notice that, by reducing β, the constant on the $L_{\mathbf{w}}(\mathbf{s})$ term can be brought arbitrarily close to 1 at the expense of increasing the constant on the other term.

4 Applications

In this section, we describe applications of the inner product results of the previous section. While we will focus on applications of Theorem 5, we note that analogs of the other results of Section 3 can be obtained in a similar manner.

4.1 Smooth functions of a single variable

We begin with a class of smooth functions of a single real variable that was studied by Faber and Mycielski [FM91] in a similar context, except using the assumption that there was a function f in the class such that $y_t = f(x_t)$ for all t. Their methodology was to prove general results like those of the previous section under that assumption that there was a $\mathbf{w}$ with $f_{\mathbf{w}}(\mathbf{x}_t) = y_t$ for all t, then to reduce the smooth function learning problem to the more general problem as we do below. Similar function classes have also often been studied in nonparametric statistics (see, e.g. [Har91]) using probabilistic assumptions on the generation of the x_t's.

For some $X > 0$, the domain $\mathcal{X}$ is the initial interval $[0, X]$ of the real line. We define the set $\mathcal{S}_{W,X}$ to be all absolutely continuous $f : [0, X] \to \boldsymbol{R}$ for which

1. $f(0) = 0$
2. $\sqrt{\int_0^X f'(z)^2\, dz} \leq W$.

The assumption that $f(0) = 0$ will be satisfied by many natural functions of interest. Examples include distance traveled as a function of time and return as a function of investment. We will prove the following result about $\mathcal{S}_{W,X}$.

Theorem 8 *For each* $E, X, W \geq 0$*, there is a prediction algorithm* A_{smooth} *with the following properties.*

Choose $\mathbf{s} = \langle (x_t, y_t) \rangle \in ([0, X] \times \boldsymbol{R})^+$*, such that there is an* $f \in \mathcal{S}_{W,X}$ *for which* $\sum_t (f(x_t) - y_t)^2 \leq E$*. Let* $\hat{y}_1, ..., \hat{y}_{|\mathbf{s}|}$ *be the sequence of* A_{smooth}*'s on-line predictions for* $\mathbf{s}$*. Then* $\sum_t (\hat{y}_t - y_t)^2$ *is at most*

$$\inf_{f \in \mathcal{S}_{W,X}} \left[\sum_t (f(x_t) - y_t)^2 \right] + 2W\sqrt{XE} + 2W^2X.$$

Proof: For now, let's put issues of computational complexity out of mind. We'll treat them again after the proof.

Fix $E, X, W \geq 0$. Algorithm A_{smooth} reduces the problem of learning $\mathcal{S}_{W,X}$ to a more general problem of the type treated in the previous section.

Let $L^2(0, X)$ be the space of (measurable) functions g from $[0, X]$ to $\boldsymbol{R}$ for which $\int_0^X g(u)^2\, du$ is finite. $L^2(0, X)$ is well known to be an inner product space (see, e.g. [You88]), with the inner product defined by

$$(g_1, g_2) = \int_0^X g_1(u) g_2(u)\, du.$$

Further, we define $g_3 = g_2 + g_1$ by

$$(\forall x) \quad g_3(x) = g_2(x) + g_1(x),$$

and $g_3 = \kappa g_1$ by

$$(\forall x) \quad g_3(x) = \kappa g_1(x).$$

Now apply algorithm **WH** to this particular inner product space, $L^2(0, X)$, with learning rate η set to $G(E, W, X)$, where the function G is defined in Section 3.2. For any $x \in [0, X]$, define $\chi_{\leq x} : [0, X] \to \boldsymbol{R}$ by

$$\chi_{\leq x}(u) = \begin{cases} 1 & \text{if } u \leq x \\ 0 & \text{otherwise.} \end{cases}$$

Note that for any $x \leq X$

$$\|\chi_{\leq x}\| = \sqrt{\int_0^X \chi_{\leq x}(u)^2\, du} = \sqrt{x} \leq \sqrt{X}, \qquad (16)$$

and therefore $\chi_{\leq x} \in L^2(0, X)$.

In Figure 3, we give a short description of algorithm A_{smooth}. Note that for any $f \in \mathcal{S}_{W,X}$,

$$\|f'\| = \sqrt{\int_0^X f'(u)^2\, du} \leq W. \qquad (17)$$

Algorithm A_{smooth}.
Input: $E, W, X \geq 0$.

- On each trial t:
  1. Get $x_t \in [0, X]$ from the environment.
  2. Give $\chi_{\leq x_t} \in L^2(0, X)$ to $\mathbf{WH}_{G(E,W,X)/X^2}$.
  3. Use $\mathbf{WH}_{G(E,W,X)/X^2}$'s prediction $\hat{y}_t$.
  4. Pass y_t to $\mathbf{WH}_{G(E,W,X)/X^2}$.

Figure 3: Pseudo-code for algorithm A_{smooth}. Algorithm **WH** (here used as a subroutine) is applied to the inner product space $\mathcal{X} = L^2(0, X)$. The function G, used to set **WH**'s learning rate, is defined in Section 3.2.

Finally, note that since $f(0) = 0$,

$$\begin{aligned}(f', \chi_{\leq x}) &= \int_0^X f'(u)\chi_{\leq x}(u)\, du \\ &= \int_0^x f'(u)\, du \\ &= f(x) - f(0) \\ &= f(x).\end{aligned} \tag{18}$$

Thus, if there is an $f \in \mathcal{S}_{W,X}$ for which $\sum_{t=1}^m (f(x_t) - y_t)^2 \leq E$, then $f' \in L^2(0, X)$ has $||f'|| \leq W$ and satisfies

$$\sum_t ((f', \chi_{\leq x_t}) - y_t)^2 \leq E.$$

Combining this with (16) and Theorem 5, we can see that **WH**'s predictions satisfy an upper bound on $\sum_t (\hat{y}_t - y_t)^2$ of

$$\inf_{||f'|| \leq W} \left[\sum_{t=1}^m ((f', \chi_{\leq x_t}) - y_t)^2 \right] + 2W\sqrt{XE} + 2W^2 X.$$

The result then follows from the fact that A_{smooth} just makes the same predictions as **WH**. □

By closely examining the predictions of algorithm A_{smooth} of Theorem 8, we can see that it can be implemented in time linear in t. Algorithm **WH** when applied to $L^2(0, X)$ maintains a function $w \in L^2(0, X)$ which it updates between trials. As before, let $\hat{w}_t$ be the tth hypothesis of **WH**. We can see that $\hat{w}_t$ can be interpreted as the derivative of A_{smooth}'s tth hypothesis. This is because **WH**'s tth prediction, and therefore A_{smooth}'s tth prediction, is

$$(\hat{w}_t, \chi_{\leq x_t}) = \int_0^X \hat{w}_t(u)\chi_{\leq x_t}(u)\, du = \int_0^{x_t} \hat{w}_t(u)\, du.$$

Hence A_{smooth}'s tth hypothesis h_t satisfies $h_t' = \hat{w}_t$.

WH sets $\hat{w}_1$ to be the constant 0 function, and its update is

$$\hat{w}_{t+1} = \hat{w}_t + \eta(y_t - \hat{y}_t)\chi_{\leq x_t},$$

where η doesn't depend on t (see the proof of Theorem 5). Integrating yields the following expression for A_{smooth}'s $(t+1)$st hypothesis:

$$h_{t+1}(x) = \begin{cases} h_t(x) + \eta(y_t - \hat{y}_t)x & \text{if } x \leq x_t \\ h_t(x) + \eta(y_t - \hat{y}_t)x_t & \text{otherwise} \end{cases}$$

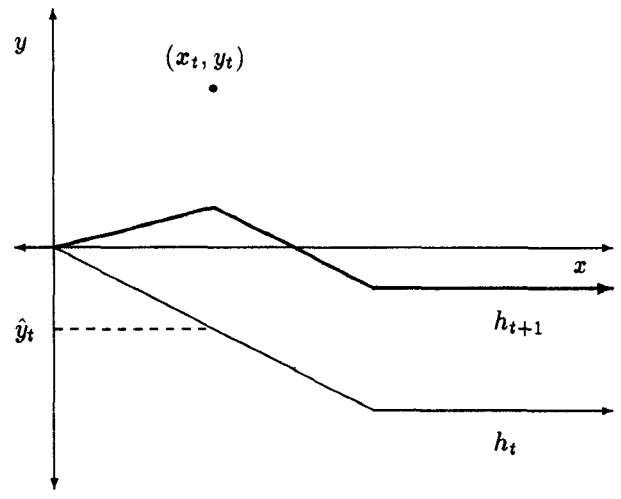

Figure 4: An example of the update of the application of the **WH** algorithm to smoothing in the single-variable case. The derivative of the hypothesis is modified by a constant in the appropriate direction to the left of x_t, and left unchanged to the right.

and therefore

$$h_{t+1}(x) = h_t(x) + \eta(y_t - \hat{y}_t)\min\{x_t, x\}.$$

By induction, we have

$$h_{t+1}(x) = \eta \sum_{s \leq t} (y_s - \hat{y}_s)\min\{x_s, x\},$$

trivially computable in $O(t)$ time if the previous $\hat{y}_s$'s are saved. This algorithm is illustrated in Figure 4.

4.2 Smooth functions of several variables

Theorem 8 can be generalized to higher dimensions as follows. The analogous generalization in the absence of noise was carried out in [FM91]. For some $M > 0$, the domain X is $[0, X]^n$. We define the set $\mathcal{S}_{W,X,n}$ to be all functions $f : [0, X]^n \to \boldsymbol{R}$ for which there is a function $\tilde{f}$ such that

1. $\forall \mathbf{x} \in \boldsymbol{R}^n$ $f(\mathbf{x}) = \int_0^{x_1} \cdots \int_0^{x_n} \tilde{f}(u_1, \ldots, u_n)\, du_n \ldots du_1$
2. $\sqrt{\int_0^X \cdots \int_0^X (\tilde{f}(u_1, \ldots, u_n))^2\, du_n \ldots du_1} \leq W.$

It is easily verified that when $\tilde{f}$ exists, it is defined by

$$\tilde{f}(u_1, \ldots, u_n) = \frac{\partial^n f(u_1, \ldots, u_n)}{\partial u_1 \ldots \partial u_n}.$$

We can establish the following generalization of Theorem 8.

Theorem 9 *For each $E, X, W \geq 0$ and $n \in \boldsymbol{N}$, there is a prediction algorithm A_{gen} with the following properties.*

Choose $\mathbf{s} = \langle(\mathbf{x}, y_t)\rangle \in ([0,X]^n \times \boldsymbol{R})^+$, *such that there is an* $f \in \mathcal{S}_{W,X,n}$ *for which* $\sum_t (f(\mathbf{x}_t) - y_t)^2 \leq E$. *Let* $\hat{y}_1, ..., \hat{y}_{|\mathbf{s}|}$ *be the sequence of* A_{gen}*'s on-line predictions for* $\mathbf{s}$. *Then* $\sum_t (\hat{y}_t - y_t)^2$ *is at most*

$$\inf_{f \in \mathcal{S}_{W,X,n}} \left[\sum_t (f(\mathbf{x}_t) - y_t)^2 \right] + 2WX^{n/2}\sqrt{E} + 2W^2X^n.$$

Proof Sketch: The proof closely follows the proof of Theorem 8, except replacing $L^2(0,X)$ with the the space of (measurable) functions g from $[0,X]^n$ to $\boldsymbol{R}$ for which $\int_0^X ... \int_0^X g(\mathbf{x})^2 \; dx_n ... dx_1$ is finite, with the inner product defined by $(g_1, g_2) = \int_0^X ... \int_0^X g_1(\mathbf{x}) g_2(\mathbf{x}) \; dx_n ... dx_1$. □

It is easy to see, by extending the discussion following Theorem 8, that the predictions of Theorem 9 can be computed in $O(tn)$ time, if previous predictions are saved.

4.3 Linear functions

For each W, X, n, let $\text{LIN}_{W,X,n}$ be the set of all functions f from $\{\mathbf{x} \in \boldsymbol{R}^n : ||\mathbf{x}||_2 \leq X\}$ to $[-WX, WX]$ for which there exists $\mathbf{w} \in \boldsymbol{R}^n$, $||\mathbf{w}||_2 \leq W$ such that for all $\mathbf{x}$ in the domain of f, $f(\mathbf{x}) = \mathbf{w} \cdot \mathbf{x}$. We may establish the following result about $\text{LIN}_{W,X,n}$ in a more straightforward manner. Its proof is omitted.

Theorem 10 *For each* $E, X, W \geq 0$, *there is a prediction algorithm* A_{lin} *with the following properties.*

Choose $n \in \boldsymbol{N}$, $\langle(\mathbf{x}_t, y_t)\rangle = \mathbf{s} \in (\boldsymbol{R}^n \times \boldsymbol{R})^+$, *such that* $\max_t ||\mathbf{x}_t||_2 \leq X$, *and there is an* $f \in \text{LIN}_{W,X,n}$ *for which* $\sum_t (f(\mathbf{x}_t) - y_t)^2 \leq E$. *Let* $\hat{y}_1, ..., \hat{y}_{|\mathbf{s}|}$ *be the sequence of* A_{lin}*'s on-line predictions for* $\mathbf{s}$. *Then* $\sum_t (\hat{y}_t - y_t)^2$ *is at most*

$$\inf_{f \in \text{LIN}_{W,X,n}} \left[\sum_t (f(\mathbf{x}_t) - y_t)^2 \right] + 2WX\sqrt{E} + 2W^2X^2.$$

5 Lower bounds

In this section, we describe lower bounds which match the upper bounds of Theorems 5, 6, 8, 9, and 10 within a constant factor. In fact, these lower bounds show that even the upper bound on the excess of the algorithm's squared loss above the best fixed element within a given class of functions is within a constant factor of optimal. Furthermore, the constant factor in front of the $\sqrt{E}$ term is the best possible in most cases. The proofs of this section are omitted due to space constraints. We begin with a lower bound for smooth functions.

Theorem 11 *Choose* $E, X, W \geq 0$, $n \in \boldsymbol{N}$, *and a prediction algorithm* A. *Then there exists* $\langle(\mathbf{x}_t, y_t)\rangle \in ([0,X]^n \times \boldsymbol{R})^+$, *such that the following hold: There is a function* $f \in \mathcal{S}_{W,X,n}$ *for which* $\sum_t (f(\mathbf{x}_t) - y_t)^2 \leq E$. *If for each* t,

$$\hat{y}_t = A(((\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_{t-1}, y_{t-1})), \mathbf{x}_t),$$

then,

$$\sum_t (\hat{y}_t - y_t)^2 \geq E + 2c\sqrt{X}\sqrt{E} + 2(WX)^2.$$

Proof Sketch: In fact $m = 1$ suffices. We let $\mathbf{x}_1 = (X, ..., X)$, $y_1 = \pm(WX^{n/2} + \sqrt{E})$ and $f : [0,X]^n \to \boldsymbol{R}$ be defined by $f(\mathbf{x}) = \frac{W}{X^{n/2}} \prod_{i=1}^n x_i$. □

We may similarly obtain the following lower bound for linear functions.

Theorem 12 *Choose* $E, X, W \geq 0$ *and a prediction algorithm* A.

Then there exists $n \in \boldsymbol{N}$, $\langle(\mathbf{x}_t, y_t)\rangle_t \in (\boldsymbol{R}^n \times [-WX, WX])^+$, *such that* $\max_t ||\mathbf{x}_t||_2 \leq X$, *and the following hold: There is an* $f \in \text{LIN}_{W,X,n}$ *for which* $\sum_t (f(\mathbf{x}_t) - y_t)^2 \leq E$. *If for each* t,

$$\hat{y}_t = A(((\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_{t-1}, y_{t-1})), \mathbf{x}_t),$$

then,

$$\sum_t (\hat{y}_t - y_t)^2 \geq E + 2WX\sqrt{E} + (WX)^2.$$

Proof Sketch. Again $m = 1$ is enough. Set $\mathbf{x}_1 = (X, 0, ..., 0)$, $y_1 = \pm(WX + \sqrt{E})$, and $\mathbf{w} = (\pm W, 0, ..., 0)$. □

To establish the upper bound of Theorem 6, in which general bounds were obtained without an knowledge of an upper bound on $L_W(\mathbf{s})$, we required the assumption that the y_t's were in the "reasonable" range $[-WX, WX]$, where W was an upper bound on $||\mathbf{w}||$ and X an upper bound on $\max_t ||\mathbf{x}_t||$. Therefore, the above lower bounds do not say anything about the optimality of those results. The following lower bound shows that this bound cannot be significantly improved in general. It further has obvious consequences concerning the linear function application when the "noise level" E is not too large relative to the number n of variables as well as W and X.

Theorem 13 *Let* $\langle \mathcal{X}_d \rangle_{d \in \boldsymbol{N}}$ *be any sequence of inner product spaces such that* $\mathcal{X}_d$ *is a* d*-dimensional vector space. Choose* $W, X, E > 0$. *Let* n *be any integer such that*

$$n \geq \left(1 + \frac{\sqrt{E}}{WX}\right)^2. \tag{19}$$

Then for any prediction algorithm A *there is a sequence* $\langle(\mathbf{x}_t, y_t)\rangle \in (\mathcal{X}_n \times [-WX, WX])^n$ *such that*

1. *For all* $1 \leq t \leq n$, $||\mathbf{x}_t|| = X$.
2. *If* $\hat{y}_1, \ldots, \hat{y}_n$ *is the sequence of predictions output by* A *when presented with the sequence* $\langle(\mathbf{x}_1, y_1)\rangle$ *on-line, then*

$$\sum_t (y_t - \hat{y}_t)^2 \geq E + 2(WX)\sqrt{E} + (WX)^2.$$

3. There exists $\mathbf{w} \in R^n$ *such that* $\|\mathbf{w}\| = W$ *and*

$$\sum_{t=1}^{n} (y_t - (\mathbf{w}, \mathbf{x}_t))^2 = E.$$

6 Acknowledgements

We thank Ethan Bernstein for helpful conversations, and for pointing out an error in an earlier version of this paper. We'd also like to thank Jyrki Kivinen for simplying the proof of Theorem 13. Thanks to Peter Auer for his comments on an earlier draft of this paper. We are also grateful to Jan Mycielski for telling us about the Kaczmarz paper.

References

[CFH+93] N. Cesa-Bianchi, Y. Freund, D.P. Helmbold, D. Haussler, R.E. Schapire, and M.K. Warmuth. How to use expert advice. *Proceedings of the 25th ACM Symposium on the Theory of Computation*, 1993. To appear.

[Daw84] A. Dawid. Statistical theory: The prequential approach. *Journal of the Royal Statistical Society (Series A)*, pages 278–292, 1984.

[DH73] R. O. Duda and P. E. Hart. *Pattern Classification and Scene Analysis.* Wiley, 1973.

[FM91] V. Faber and J. Mycielski. Applications of learning theorems. *Fundamenta Informaticae*, 15(2):145–167, 1991.

[FMG92] M. Feder, N. Merhav, and M. Gutman. Universal prediction of individual sequences. *IEEE transactions of information theory*, 38:1258–1270, 1992.

[Har91] W. Hardle. *Smoothing Techniques.* Springer Verlag, 1991.

[Kac37] S. Kaczmarz. Angenaherte Auflösung von systemen linearer gleichungen. *Bull. Acad. Polon. Sci. Lett. A*, 35:355–357, 1937.

[KL92] D. Kimber and P.M. Long. The learning complexity of smooth functions of a single variable. *The 1992 Workshop on Computational Learning Theory*, pages 153–159, 1992.

[LLW91] 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.

[LMT91] R.P. Lippman, J.E. Moody, and D.S. Touretsky. *Advances in Neural Information Processing Systems*, volume 3. Morgan Kaufmann, 1991.

[LW91] N. Littlestone and M.K. Warmuth. The weighted majority algorithm. Technical Report UCSC-CRL-91-28, UC Santa Cruz, October 1991. A preliminary version appeared in *the Proceedings of the 30th Annual IEEE Symposium on the Foundations of Computer Science*, October 89, pages 256-261.

[MF92] N. Merhav and M. Feder. Universal sequential learning and decision from individual data sequences. *The 1992 Workshop on Computational Learning Theory*, pages 413–427, 1992.

[MHL92] J.E. Moody, S.J. Hanson, and R.P. Lippman. *Advances in Neural Information Processing Systems*, volume 4. Morgan Kaufmann, 1992.

[MS91] J. Mycielski and S. Swierczkowski. General learning theorems. Unpublished, 1991.

[Myc88] J. Mycielski. A learning algorithm for linear operators. *Proceedings of the American Mathematical Society*, 103(2):547–550, 1988.

[Tou89] David S. Touretsky. *Advances in Neural Information Processing Systems*, volume 1. Morgan Kaufmann, 1989.

[Tou90] David S. Touretsky. *Advances in Neural Information Processing Systems*, volume 2. Morgan Kaufmann, 1990.

[Vov90] V. Vovk. Aggregating strategies. In *Proceedings of the 3nd Workshop on Computational Learning Theory*, pages 371–383. Morgan Kaufmann, 1990.

[WH60] B. Widrow and M.E. Hoff. Adaptive switching circuits. *1960 IRE WESCON Conv. Record*, pages 96–104, 1960.

[You88] N. Young. *An introduction to Hilbert space.* Cambridge University Press, 1988.