Absolute Loss Bounds
for Prediction using Linear Functions

Philip M. Long
ISCS Department
National University of Singapore
Singapore, 119260, Republic of Singapore

August 1, 1996

Abstract

We prove new absolute loss bounds for learning linear functions in the standard on-line prediction model. These bounds on are on difference between the sum of absolute prediction errors made by the learning algorithm, and the best sum of absolute prediction errors that can be obtained by fixing a linear function in some class. Known results imply that our bounds on this difference cannot be improved by more than a constant factor.

1 Introduction

In the standard on-line prediction model, learning proceeds in trials, where in the tth trial, the learner (a) gets an element x_t from the domain, (b) outputs a prediction $\hat{y}_t$, and (c) discovers y_t. The learner then "pays" some measure of how inaccurate $\hat{y}_t$ was in predicting y_t. The goal is to obtain worst-case bounds, for a given class F of functions, on the total such "loss" in terms of the least loss obtainable by fixing a function f from F and always predicting $f(x_t)$.

Our results measure the loss with $|\hat{y}_t - y_t|$. Most of the work about on-line learning of linear functions has used $(\hat{y}_t - y_t)^2$ instead [20, 9, 17, 14, 5]. There are two notable exceptions: Bernstein [3] studied the learning of linear functions with the absolute loss, but his results were for the case in which $y_t = f(\vec{x}_t)$ on each trial. For many applications, one needs learning algorithms to be able to cope with situations where a linear function only approximately maps $\vec{x}_t$'s to y_t's. (Bernstein pointed this out as an open problem.)[1] One can use the techniques of

[1] At talks describing the research in [17], Umesh Vazirani raised the question of whether similar results could be obtained with respect to the absolute loss.

Klasner and Simon [15] to obtain bounds for learning linear functions with absolute loss in terms of the loss of the best function (they stated one such result); we compare our results with these below.

Aside from the fact that it is an arguably more natural measure of error, study of the absolute loss is also motivated by the applied problem of lossless compression of still images. In lossless compression, the image reconstructed from the compressed file is exactly the same as the image before compression. This property is important for example for medical images. A standard method for lossless image compression (see [11]) is to process the pixels of the image one at a time, and use previously processed pixels to predict the value of the current pixel. A linear function of some of the previously encountered pixels has been found to yield good predictions (see [21, page 309]). The previously coded pixels are used so that the decoder can determine the encoder's prediction for a given pixel. One then codes the value of the current pixel, using arithmetic coding, taking as the probability model a Laplace distribution centered at the prediction. A trivial calculation shows that, when a Laplace distribution is assumed, the number of bits to code a particular pixel is proportional to the absolute value of the difference between the predicted and the true values for that pixel. Therefore, our results about learning linear functions with absolute loss can be interpreted as providing performance guarantees for the prediction component of a lossless image compression system, where the other components follow the standard practice. In contrast, the quadratic loss corresponds to doing the encoding using the Gaussian distribution as the probability model, well known to be a worse model for this context.

Instead of measuring the length of the elements of the domain and the coefficient vector with the usual Euclidian norm, it has been argued [17, 14] that measuring the length of the elements of the domain with the ℓ_∞ norm, and the length of the coefficient vector with the ℓ_1 norm, gives rise to bounds that are relatively more relevant when most of the length of the coefficient vector is concentrated in a few components. Again, for the application of lossless image compression, this seems likely to be the case, since the coefficients corresponding to pixels closest to the pixel being predicted should be relatively large.

Suppose $\mathrm{LIN}(\infty, 1, d)$ is the set of all functions f defined on $\vec{x} \in \mathbf{R}^d$ for which $||\vec{x}||_\infty \leq 1$ such that there is a coefficient vector $\vec{c} \in \mathbf{R}^d$ such that $||\vec{c}||_1 \leq 1$. We describe an $O(d)$ time algorithm, related to the algorithms described by Littlestone, Long and Warmuth [17] and Kivinen and Warmuth [14] (which in turn were inspired by Littlestone's [16] work concerning linear threshold functions), for which

$$\left(\sum_{t=1}^{m} |\hat{y}_t - y_t|\right) - \left(\min_{f \in \mathrm{LIN}(\infty,1,d)} \sum_{t=1}^{m} |f(x_t) - y_t|\right) \leq 3\sqrt{m(1 + \ln d)} + 4 \ln d + 4. \qquad (1)$$

Lower bound arguments in [4] imply that this bound is within a constant factor of optimal for all large enough m and d.

A surprising property of the algorithm we propose is that it makes use of y_t only to determine whether $\hat{y}_t$ was too low or too high. Therefore, this algorithm would obtain the same performance guarantee if it only received this information. The problem of learning real-valued functions with such "directional feedback" was raised by Barland [2]. The main thrust of Barland's work was experimental. He did some theoretical analysis as well, but did not prove cumulative loss bounds. A characterization of the complexity of learning a generic class F of real-valued functions with directional feedback was given by Auer, Long, Maass and Woeginger [1, Theorem 5]. This result required that there was an $f \in F$ for which $y_t = f(x_t)$ for each t.

We also show that, if $\mathrm{LIN}(2,2,d)$ is the set of linear functions defined on $\vec{x} \in \mathbf{R}^d$ for which $||\vec{x}||_2 \leq 1$ such that there is a coefficient vector $\vec{c} \in \mathbf{R}^d$ such that $||\vec{c}||_2 \leq 1$, an $O(d)$ time algorithm similar to the Widrow-Hoff algorithm [12, 22] achieves, for any d, on any sequence $(\vec{x}_1, y_1), ..., (\vec{x}_m, y_m)$, a bound of

$$\left(\sum_{t=1}^{m} |\hat{y}_t - y_t|\right) - \left(\min_{f \in \mathrm{LIN}(2,2,d)} \sum_{t=1}^{m} |f(\vec{x}_t) - y_t|\right) \leq \sqrt{m}. \tag{2}$$

For each particular d, a lower bound that matches to within a constant factor follows from the work of Cover [6]. A simple lower bound given here shows that (2) is *exactly* the best possible bound that is independent of d. The learning algorithm we analyze for this result is like the Widrow-Hoff algorithm in that, after each trial t, it updates its hypothesis coefficient vector to move in the direction of the closest $\vec{w}$ such that $\vec{w} \cdot \vec{x}_t = y_t$. The difference is simply that our algorithm makes its step size without regard to $|\hat{y}_t - y_t|$, and therefore this algorithm also can be used with only directional feedback.

Using the technique of Klasner and Simon [15], one can obtain bounds on $\sum_{t=1}^{m} |\hat{y}_t - y_t|$ in terms of $\min_{f \in F} \sum_{t=1}^{m} |f(x_t) - y_t|$ and d, but they do not imply that the per-trial error of the learning algorithm approaches that of the best function in $\mathrm{LIN}(\infty, 1, d)$ and $\mathrm{LIN}(2, 2, d)$ respectively as do our bounds. One can apply general results of Merhav and Feder [19] and Freund and Schapire [10], to obtain such bounds, but the rates of convergence we have been able to obtain using those bounds in terms of m and d are not within a constant factor of either of our bounds. Further, the algorithms we describe take $O(d)$ time per trial, where it is not clear how to obtain similarly fast algorithms which make the predictions in the case of linear functions of the general-purpose methods of [10] and [19].

Using standard scaling and doubling techniques (see [17, 4, 5]), we expect that our results can be generalized to allow for larger coefficient vectors and domain elements, without knowledge of these and other parameters, with a slight weakening of the bounds.

2 Definitions

Denote the reals by $\mathbf{R}$ and the positive integers by $\mathbf{N}$.

Choose $d \in \mathbf{N}$. For $\vec{x} \in \mathbf{R}^d$ and $p \geq 1$, define $||\vec{x}||_p = (\sum_i |x_i|^p)^{1/p}$, and define $||\vec{x}||_\infty = \max_i |x_i|$. Define $\mathcal{B}_{d,p}$ to be the set of all elements of $\mathbf{R}^d$ whose p-norm is at most 1. For each $p, q \geq 1$, define $\mathrm{LIN}(p, q, d)$ to be the set of functions f from $\mathcal{B}_{d,p}$ to $\mathbf{R}$ for which there exists $\vec{c} \in \mathcal{B}_{d,q}$ such that for all $\vec{x} \in \mathcal{B}_{d,p}$, $f(\vec{x}) = \vec{c} \cdot \vec{x}$.

In the standard on-line prediction model (see [20, 16, 4]), learning proceeds in *trials*. In each trial t an algorithm (a) is given $x_t \in X$, (b) outputs $\hat{y}_t \in [-1, 1]$, and (c) receives $y_t \in [-1, 1]$.

For a standard model learning algorithm A, define

$$R_{\mathrm{stand}}(A, F, m) = \sup_{(x_1,y_1),\ldots,(x_m,y_m)\in X\times[-1,1]} \left(\sum_{t=1}^{m} |\hat{y}_t - y_t|\right) - \left(\inf_{f\in F}\sum_{t=1}^{m} |f(x_t) - y_t|\right).$$

where the $\hat{y}_t$'s are a function of A, the x_t's and the y_t's as described above.

3 Learning $\mathrm{LIN}(\infty, 1, d)$

Choose $d, m \in \mathbf{N}$. Define WA_d to be the set of all functions f from $[0,1]^d$ to $[0,1]$ for which there exists $\vec{c} \in [0,1]^d$ with $\sum_i c_i = 1$ such that for all $\vec{x} \in [0,1]^d$, $f(\vec{x}) = \vec{c}\cdot\vec{x}$.

As in [17, 14], our algorithm for learning $\mathrm{LIN}(\infty, 1, d)$ will use as a subroutine an algorithm for learning WA_d. We will make use of the following lemma, which is implicit in the work of Littlestone, Long and Warmuth [17].

Lemma 1 *If there is an algorithm A for learning WA_{2d+1} which uses β time to compute its predictions, and γ time between trials, then there is an algorithm A' for learning $\mathrm{LIN}(\infty, 1, d)$ that uses $\beta + O(d)$ time to compute its predictions, and γ time between trials, and such that $R_{\mathrm{stand}}(A', \mathrm{LIN}(\infty, 1, d), m) \le 2R_{\mathrm{stand}}(A, \mathrm{WA}_{2d+1}, m)$.*

Consider the following algorithm A_{WA} for learning WA_d in the standard model on sequences of m trials. A_{WA} predicts $\hat{y}_t = \vec{v}_t \cdot \vec{x}_t$, where A_{WA}'s hypothesis $\vec{v}_t$ is set to $\vec{w}_t/||\vec{w}_t||_1$ and $\vec{w}_t$ is maintained as follows. First, $\vec{w}_1 = (1, 1, ..., 1)$, and after trial t, for each $i \le d$, A_{WA} sets

$$w_{t+1,i} = w_{t,i}\left(1 + \mathrm{sign}(y_t - \hat{y}_t)\sqrt{\frac{2\ln d}{m}}\right)^{x_{t,i}}.$$

Theorem 2 $R_{\mathrm{stand}}(A_{\mathrm{WA}}, \mathrm{WA}_d, m) \le \sqrt{2m\ln d} + 2\ln d$.

We will make use of the following lemmas, which are easily verified using Calculus (see [16]).

Lemma 3 *Choose $\xi > 0$, $x \in [0,1]$. Then*

$$\xi^x \le 1 + (\xi - 1)x.$$

Lemma 4 *Choose $x \in [-9/10, 9/10]$. Then*

$$\ln(1 + x) \ge x - x^2/2 - |x|^3.$$

This lemma bounds the change in the measure of progress in a particular trial. It borrows some ideas from [16, 17].

Lemma 5 *Choose* $d \in \mathbf{N}, \vec{x} \in [0,1]^d, \vec{w}_{\rm old} \in [0,\infty)^d, 0 < \gamma < 9/10, y \in \mathbf{R}$. *Suppose* $\vec{v}_{\rm old} = \vec{w}_{\rm old}/||\vec{w}_{\rm old}||_1$. *Let* $\hat{y} = \vec{v}_{\rm old} \cdot \vec{x}$, *and* $\vec{w}_{\rm new}$ *be defined by*

$$w_{{\rm new},i} = w_{{\rm old},i} \left(1 + {\rm sign}(y - \hat{y})\gamma\right)^{x_i},$$

and let $\vec{v}_{\rm new} = \vec{w}_{\rm new}/||\vec{w}_{\rm new}||_1$. *Then, for any* $\vec{c} \in [0,1]^d$ *with* $\sum_i c_i = 1$,

$$I(\vec{c}||\vec{v}_{\rm new}) - I(\vec{c}||\vec{v}_{\rm old}) \leq -\gamma|\hat{y} - y| + \gamma|\vec{c}\cdot\vec{x} - y| + \gamma^2/2 + \gamma^3.$$

Proof: It is easy to verify (see [16]) that

$$I(\vec{c}||\vec{v}_{\rm new}) - I(\vec{c}||\vec{v}_{\rm old}) = \ln \frac{\sum_i w_{{\rm new},i}}{\sum_i w_{{\rm old},i}} - \left(\sum_i c_i \ln \frac{w_{{\rm new},i}}{w_{{\rm old},i}}\right). \tag{3}$$

We begin by bounding the first term of (3). First,

$$\sum_i w_{{\rm new},i} = \sum_i w_{{\rm old},i} \left(1 + {\rm sign}(y - \hat{y})\gamma\right)^{x_i}.$$

Applying Lemma 3, we get

$$\frac{\sum_i w_{{\rm new},i}}{\sum_i w_{{\rm old},i}} = \sum_i v_{{\rm old},i} \left(1 + {\rm sign}(y - \hat{y})\gamma\right)^{x_i} \leq \sum_i v_{{\rm old},i}(1 + {\rm sign}(y - \hat{y})\gamma x_i) = 1 + {\rm sign}(y - \hat{y})\gamma\hat{y}.$$

Thus

$$\ln \frac{\sum_i w_{{\rm new},i}}{\sum_i w_{{\rm old},i}} \leq \ln(1 + {\rm sign}(y - \hat{y})\gamma\hat{y}) \leq {\rm sign}(y - \hat{y})\gamma\hat{y}. \tag{4}$$

Now we work on the second term of (3). We have

$$\sum_i c_i \ln \frac{w_{{\rm new},i}}{w_{{\rm old},i}} = \sum_i c_i x_i \ln\left(1 + {\rm sign}(y - \hat{y})\gamma\right) = (\vec{c}\cdot\vec{x}) \ln\left(1 + {\rm sign}(y - \hat{y})\gamma\right).$$

Putting this together with (4) and (3), we get

$$I(\vec{c}||\vec{v}_{\rm new}) - I(\vec{c}||\vec{v}_{\rm old}) \leq {\rm sign}(y - \hat{y})\gamma\hat{y} - (y + (\vec{c}\cdot\vec{x} - y)) \ln\left(1 + {\rm sign}(y - \hat{y})\gamma\right). \tag{5}$$

Assume as a first case that $y > \hat{y}$. Then (5) implies that $I(\vec{c}||\vec{v}_{\rm new}) - I(\vec{c}||\vec{v}_{\rm old}) \leq \gamma\hat{y} - (y + (\vec{c}\cdot\vec{x} - y)) \ln(1 + \gamma)$. Since $\vec{c}\cdot\vec{x} \geq 0$, applying Lemma 4, we get

$$\begin{aligned} I(\vec{c}||\vec{v}_{\rm new}) - I(\vec{c}||\vec{v}_{\rm old}) &\leq \gamma\hat{y} - (y + (\vec{c}\cdot\vec{x} - y))\left(\gamma - \gamma^2/2 - \gamma^3\right) \\ &= \gamma(-|\hat{y} - y| - (\vec{c}\cdot\vec{x} - y)) + (\vec{c}\cdot\vec{x})(\gamma^2/2 + \gamma^3) \quad (\text{since } y \geq \hat{y}) \\ &\leq \gamma(-|\hat{y} - y| + |\vec{c}\cdot\vec{x} - y|) + \gamma^2/2 + \gamma^3 \end{aligned} \tag{6}$$

since $\vec{c}\cdot\vec{x} \leq 1$.

Now, assume that $\hat{y} > y$. Then (5) implies that, in this case, $I(\vec{c}||\vec{v}_{\text{new}}) - I(\vec{c}||\vec{v}_{\text{old}}) \leq -\gamma\hat{y} - (y + (\vec{c}\cdot\vec{x} - y))\ln(1-\gamma)$. Again, since $\vec{c}\cdot\vec{x} \geq 0$, applying Lemma 4 yields

$$\begin{aligned} I(\vec{c}||\vec{v}_{\text{new}}) - I(\vec{c}||\vec{v}_{\text{old}}) &\leq -\gamma\hat{y} - (y + (\vec{c}\cdot\vec{x} - y))\left(-\gamma - \gamma^2/2 - \gamma^3\right) \\ &= \gamma(-|\hat{y} - y| + (\vec{c}\cdot\vec{x} - y)) + (\vec{c}\cdot\vec{x})(\gamma^2/2 + \gamma^3) \quad (\text{since } \hat{y}_t > y_t) \\ &\leq \gamma(-|\hat{y} - y| + |\vec{c}\cdot\vec{x} - y|) + \gamma^2/2 + \gamma^3. \end{aligned}$$

Putting this together with (6) completes the proof. □

Proof (of Theorem 2): If $m \leq 3\ln d$, since $R_{\text{stand}}(A_{\text{WA}}, \text{WA}_d, m) \leq m$ the theorem is trivial. Assume $m > 3\ln d$. Choose $\vec{x}_1, ..., \vec{x}_m \in [0,1]^d$ and $y_1, ..., y_m \in [0,1]$. Let $\hat{y}_1, ..., \hat{y}_m \in [0,1]$ be computed from the $\vec{x}_t$'s and y_t's using Algorithm A_{WA} as described above. Choose $\vec{c} \in [0,1]^d$ with $\sum_i c_i = 1$.

Applying Lemma 5 with $\gamma = \sqrt{2(\ln d)/m}$, we have that for each t,

$$I(\vec{c}||\vec{v}_{t+1}) - I(\vec{c}||\vec{v}_t) \leq \sqrt{\frac{2\ln d}{m}}(-|\hat{y}_t - y_t| + |\vec{c}\cdot\vec{x}_t - y_t|) + \frac{\ln d}{m} + \left(\frac{2\ln d}{m}\right)^{3/2}.$$

Summing and telescoping, we get

$$I(\vec{c}||\vec{v}_{m+1}) - I(\vec{c}||\vec{v}_1) \leq \sum_{t=1}^{m}\sqrt{\frac{2\ln d}{m}}(-|\hat{y}_t - y_t| + |\vec{c}\cdot\vec{x}_t - y_t|) + \frac{\ln d}{m} + \left(\frac{2\ln d}{m}\right)^{3/2}.$$

Applying the fact that $I(\vec{c}||\vec{v}_1) \leq \ln d$ and $I(\vec{c}||\vec{v}_{m+1}) \geq 0$ (see [7]) yields

$$-\ln d \leq \sum_{t=1}^{m}\left(\sqrt{\frac{2\ln d}{m}}(-|\hat{y}_t - y_t| + |\vec{c}\cdot\vec{x}_t - y_t|) + \frac{\ln d}{m} + \left(\frac{2\ln d}{m}\right)^{3/2}\right).$$

Solving we get

$$\sum_{t=1}^{m}|\hat{y}_t - y_t| - \left(\sum_{t=1}^{m}|\vec{c}\cdot\vec{x}_t - y_t|\right) \leq \sqrt{2m\ln d} + 2\ln d.$$

Since $\vec{c}$ was chosen arbitrarily, this completes the proof. □

Combining this with Lemma 1 proves (1).

4 Learning LIN$(2,2,d)$

Choose m and d. Consider the algorithm $A_{2,2,d}$ for learning LIN$(2,2,d)$ in the standard model that maintains a hypothesis coefficient vector $\vec{w}_t$ by setting $\vec{w}_1 = \vec{0}$, and after trial t, performing $\vec{w}_{t+1} = \vec{w}_t + \frac{\text{sign}(y_t - \hat{y}_t)\vec{x}_t}{\sqrt{m}}$. Algorithm $A_{2,2,d}$ predicts $\hat{y}_t = \vec{w}_t\cdot\vec{x}_t$. The following is the main result of this section.

Theorem 6 $R_{\text{stand}}(A_{2,2,d}, \text{LIN}(2,2,d), m) \leq \sqrt{m}$.

This lemma bounds the change in our measure of progress in a particular trial. (This measure of progress was also used in [8, 5].)

Lemma 7 *Choose* $d \in \mathbf{N}, \vec{x} \in \mathcal{B}_{d,2}, \vec{w}_{\text{old}} \in \mathbf{R}^d, \gamma > 0, y \in \mathbf{R}$. *Let* $\hat{y} = \vec{w}_{\text{old}} \cdot \vec{x}$, $\vec{w}_{\text{new}} = \vec{w}_{\text{old}} + \gamma \text{sign}(y - \hat{y})\vec{x}$. *Then, for any* $\vec{c} \in \mathbf{R}^d$, $||\vec{w}_{\text{new}} - \vec{c}||_2^2 - ||\vec{w}_{\text{old}} - \vec{c}||_2^2 \leq -2\gamma|\hat{y} - y| + 2\gamma|\vec{c} \cdot \vec{x} - y| + \gamma^2$.

Proof: Substituting the definition of $\vec{w}_{\text{new}}$ and collecting terms[2], we have

$$||\vec{w}_{\text{new}} - \vec{c}||_2^2 - ||\vec{w}_{\text{old}} - \vec{c}||_2^2 = -2\gamma|\hat{y} - y| + 2\gamma \text{sign}(y - \hat{y})(y - \vec{c} \cdot \vec{x}) + \gamma^2 ||\vec{x}||_2^2.$$

Overestimating each of the last two terms completes the proof. □

Proof (of Theorem 6): Choose $\vec{c} \in \mathcal{B}_{d,2}$. By Lemma 7,

$$\sum_{t=1}^{m} ||\vec{w}_{t+1} - \vec{c}||_2^2 - ||\vec{w}_t - \vec{c}||_2^2 \leq \sum_{t=1}^{m} (-(2/\sqrt{m})(|\hat{y}_t - y_t| - |\vec{c} \cdot \vec{x}_t - y_t|) + 1/m).$$

Since $\vec{w}_1 = \vec{0}$ and $||\vec{w}_t||_2 \leq 1$, telescoping implies $-1 \leq \sum_{t=1}^{m} (-(2/\sqrt{m})(|\hat{y}_t - y_t| - |\vec{c} \cdot \vec{x}_t - y_t|) + 1/m)$. Solving and noting that $\vec{c}$ was chosen arbitrarily completes the proof. □

Define F_2 to be the set of all functions f from $[0, 1]$ to $\mathbf{R}$ such that $f(0) = 0$ and $\int_0^1 f'(x)^2 \, dx \leq 1$. On-line learning of this class has been considered using the quadratic loss in [9, 13, 5] and with the absolute loss in the noise-free case in [13, 18]. As Theorem 6 can be readily generalized by replacing the dot product with a general inner product and F_2 can be defined as a set of linear functions using the appropriate inner product (see [9, 5]), the following can be established with essentially the same proof as Theorem 6.

Theorem 8 *There is an algorithm* A *such that*

$$R_{\text{stand}}(A, F_2, m) \leq \sqrt{m}.$$

Results due to Cover [6] imply that this bound is within a constant factor of the best possible.

Next, we give a lower bound showing that Theorem 6 is exactly the best possible upper bound that is independent of d.

Proposition 9 *For each* $m \in \mathbf{N}$*, there is a* $d \in \mathbf{N}$ *such that for any algorithm* A*, for learning* $\text{LIN}(2, 2, d)$,

$$R_{\text{stand}}(A, \text{LIN}(2, 2, d), m) \geq \sqrt{m}.$$

Proof: Set $d = m$. For each $t \leq m$, set $\vec{x}_t = (0, ..., 0, 1, 0, ..., 0)$, where the 1 is in the tth component, and set $y_t = \pm 1/\sqrt{m}$, whichever is furthest from A's prediction. If, for all t, $w_t = y_t$, then $||\vec{w}|| = 1$, and $\sum_{t=1}^{m} |\vec{w} \cdot \vec{x} - y_t| = 0$, but $\sum_{t=1}^{m} |\hat{y}_t - y_t| = m(1/\sqrt{m}) = \sqrt{m}$. This completes the proof. □

[2] A similar calculation was carried out in [5].

Acknowledgements

We gratefully acknowledge the support of ONR grant N00014–94–1–0938.

References

[1] P. Auer, P.M. Long, W. Maass, and G.J. Woeginger. On the complexity of function learning. *Machine Learning*, 18(2):187–236, 1995.

[2] I. Barland. Some ideas on learning with directional feedback. Master's thesis, UC Santa Cruz, June 1992.

[3] E.J. Bernstein. Absolute error bounds for learning linear functions on line. *Proceedings of the 1992 Workshop on Computational Learning Theory*, 1992.

[4] 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.

[5] N. Cesa-Bianchi, P.M. Long, and M.K. Warmuth. Worst-case quadratic loss bounds for prediction using linear functions and gradient descent. *IEEE Transactions on Neural Networks*, 7(3):604–619, 1996.

[6] T. Cover. Behavior of sequential predictors of binary sequences. In *Proceedings of the 4th Prague Conference on Information Theory, Statistical Decision Functions and Random Processes*, pages 263–272. Publishing House of the Czechoslovak Academy of Sciences, 1965.

[7] T. Cover and J. Thomas. *Elements of Information Theory.* Wiley, 1991.

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

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

[10] Y. Freund and R. E. Schapire. A decision-theoretic generalization of on-line learning and an application to boosting. *Proceedings of the Second European Conference on Computational Learning Theory*, pages 23–37, 1995.

[11] P.G. Howard and J.S. Vitter. New methods for lossless image compression using arithmetic coding. *Information Processing and Management*, 28, 1992.

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

[13] D. Kimber and P.M. Long. On-line learning of smooth functions of a single variable. *Theoretical Computer Science*, 148(1):141–156, 1995.

[14] J. Kivinen and M.K. Warmuth. Additive versus exponentiated gradient updates for linear prediction. *Proceedings of the 27th ACM Symposium on the Theory of Computation*, pages 209–218, 1995.

[15] N. Klasner and H.U. Simon. From noise-free to noise-tolerant and from on-line to batch learning. *The 1995 Conference on Computational Learning Theory*, pages 250–257, 1995.

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

[17] N. Littlestone, P.M. Long, and M.K. Warmuth. On-line learning of linear functions. *Computational Complexity*, 5:1–23, 1995.

[18] P.M. Long. Absolute loss bounds for prediction using linear functions. *The 1996 International Workshop on Algorithmic Learning Theory*, 1996.

[19] N. Merhav and M. Feder. Universal schemes for sequential decision from individual data sequences. *IEEE Transactions on Information Theory*, 39(4):1280–1292, 1993.

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

[21] M.J. Slyz and D.L. Neuhoff. A nonlinear VQ-based predictive lossless image coder. *Proceedings of the 1994 IEEE Data Compression Conference*, pages 304–310, 1994.

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