Worst-Case Absolute Loss Bounds for Linear Learning Algorithms

Tom Bylander
Division of Computer Science
The University of Texas at San Antonio
San Antonio, Texas 78249
bylander@cs.utsa.edu

Abstract

The absolute loss is the absolute difference between the desired and predicted outcome. I demonstrate worst-case upper bounds on the absolute loss for the perceptron algorithm and an exponentiated update algorithm related to the Weighted Majority algorithm. The bounds characterize the behavior of the algorithms over any sequence of trials, where each trial consists of an example and a desired outcome interval (any value in the interval is an acceptable outcome). The worst-case absolute loss of both algorithms is bounded by: the absolute loss of the best linear function in the comparison class, plus a constant dependent on the initial weight vector, plus a per-trial loss. The per-trial loss can be eliminated if the learning algorithm is allowed a tolerance from the desired outcome. For concept learning, the worst-case bounds lead to mistake bounds that are comparable to previous results.

Introduction

Linear and linear threshold functions are an important class of functions for machine learning. Although linear functions are limited in what they can represent, they often achieve good empirical results, e.g., (Gallant 1990), and they are standard components of neural networks.

For concept learning in which some linear threshold function is a perfect classifier, mistake bounds are known for the perceptron algorithm (Rosenblatt 1962; Minsky & Papert 1969), and Winnow and Weighted Majority algorithms (Littlestone 1988; 1989; Littlestone & Warmuth 1994). There are also results for various types of noise, e.g., (Bylander 1994). However, these results do not characterize the behavior of these algorithms over any sequence of examples.

This paper shows that minimizing the absolute loss characterizes the online behavior of the perceptron algorithm and an exponentiated update algorithm (related to Weighted Majority) over any sequence of examples, where the absolute loss is the absolute difference between the desired and predicted outcome. The worst-case absolute loss of both algorithms is bounded by the sum of: the absolute loss of the best linear function in the comparison class, plus a constant dependent on the initial weight vector, plus a per-trial loss. The per-trial loss can be eliminated if the learning algorithm is allowed a tolerance from the desired outcome.

A few previous results are also based on the absolute loss, though for specialized cases. Duda & Hart (1973) derive the perceptron update rule from the perceptron criterion function, which is a specialization of the absolute loss. The perceptron algorithm with a decreasing learning rate (harmonic series) on a stationary distribution of examples converges to a linear function with the minimum absolute loss (Kashyap 1970). A version of the Weighted Majority algorithm (WMC) has an absolute loss comparable to the best input (Littlestone & Warmuth 1994).

The analysis follows a pattern similar to worst-case analyses of online linear least-square algorithms (Cesa-Bianchi, Long, & Warmuth 1996; Kivinen & Warmuth 1994). The performance of an algorithm is compared to the best hypothesis in some comparison class. The bounds are based on how the distance from the online algorithm's current hypothesis to the target hypothesis changes in proportion to the algorithm's loss minus target's loss. The distance measure for hypotheses is chosen to facilitate the analysis.

The desired outcome for an example is allowed to be any real interval. Thus, concept learning can be implemented with a positive/negative outcome for positive/negative examples. In this case, the absolute loss bounds lead to mistake bounds for these algorithms that are similar to previous literature. I also obtain expected mistake bounds for randomized versions of the algorithms. Because the mistake bounds are closely related to the absolute loss, the absolute loss analysis is to some extent implicit in previous mistake bound analyses (Warmuth 1996).

Preliminaries

A trial is an ordered pair $(\mathbf{x}, [y_{lo}, y_{hi}])$, consisting of an example $\mathbf{x} \in \Re^n$ and an outcome interval $[y_{lo}, y_{hi}]$, i.e., it is desired that the outcome be in the real interval $[y_{lo}, y_{hi}]$. Open intervals may be used, and $y_{lo} = -\infty$ and $y_{hi} = \infty$ are permitted. A prediction $\widehat{y}$ on an example $\mathbf{x}$ is made using a weight vector $\mathbf{w} \in \Re^n$ by computing the dot product $\widehat{y} = \mathbf{w} \cdot \mathbf{x} = \sum_{i=1}^{n} w_i x_i$. The absolute loss of a weight vector $\mathbf{w}$ on a trial $(\mathbf{x}, [y_{lo}, y_{hi}])$ is determined by:

$$\text{Loss}(\mathbf{w}, (\mathbf{x}, [y_{lo}, y_{hi}])) = \begin{cases} y_{lo} - \widehat{y} & \text{if } \widehat{y} < y_{lo} \\ 0 & \text{if } \widehat{y} \in [y_{lo}, y_{hi}] \\ \widehat{y} - y_{hi} & \text{if } \widehat{y} > y_{hi} \end{cases}$$

Algorithm Perceptron(s,η)
Parameters:
s: the start vector, with $\mathbf{s} \in \Re^n$.
η: the learning rate, with $\eta > 0$.
Initialization:
Before the first trial, set $\mathbf{w}_1$ to $\mathbf{s}$.
Prediction:
Upon receiving the tth example $\mathbf{x}_t$,
give the prediction $\widehat{y}_t = \mathbf{w}_t \cdot \mathbf{x}_t$
Update:
Upon receiving the tth outcome interval $[y_{t,lo}, y_{t,hi}]$,
update the weight vector using:

$$\mathbf{w}_{t+1} = \begin{cases} \mathbf{w}_t + \eta \mathbf{x}_t & \text{if } \widehat{y}_t < y_{t,lo} \\ \mathbf{w}_t & \text{if } \widehat{y}_t \in [y_{t,lo}, y_{t,hi}] \\ \mathbf{w}_t - \eta \mathbf{x}_t & \text{if } \widehat{y}_t > y_{t,hi} \end{cases}$$

Figure 1: Perceptron Algorithm

The Loss$(\cdot,\cdot)$ notation is also used for the absolute loss of a weight vector or algorithm on a trial or sequence of trials.

For an online algorithm A, a comparison weight vector $\mathbf{u}$, and a trial sequence S, all of the bounds are of the form $\text{Loss}(A,S) \leq \text{Loss}(\mathbf{u},S) + \zeta$, where ζ is be an expression based on characteristics of the algorithm and the trial sequence. These bounds are based on demonstrating, for each trial S_t, that $\text{Loss}(A,S_t) - \text{Loss}(\mathbf{u},S_t) \leq \zeta_t$, and summing up the additional loss ζ_t over all the trials. For the algorithms considered here, ζ_t is nonnegative when $\text{Loss}(A,S_t) = 0$. The other cases are covered by the following lemma.

Lemma 1 *When $\widehat{y} < y_{lo}$ for a given trial $S_t = (\mathbf{x}, [y_{lo}, y_{hi}])$, then:*

$$\begin{aligned} \text{Loss}(A,S_t) - \text{Loss}(\mathbf{u},S_t) &\leq (y_{lo} - \widehat{y}) - (y_{lo} - \mathbf{u}\cdot\mathbf{x}) \\ &= \mathbf{u}\cdot\mathbf{x} - \widehat{y} \end{aligned}$$

When $\widehat{y} > y_{hi}$ for a given trial $S_t = (\mathbf{x}, [y_{lo}, y_{hi}])$, then:

$$\begin{aligned} \text{Loss}(A,S_t) - \text{Loss}(\mathbf{u},S_t) &\leq (\widehat{y} - y_{hi}) - (\mathbf{u}\cdot\mathbf{x} - y_{hi}) \\ &= \widehat{y} - \mathbf{u}\cdot\mathbf{x} \end{aligned}$$

Proof: When $\widehat{y} < y_{lo}$, the first inequality follows from the fact that $y_{lo} - \mathbf{u}\cdot\mathbf{x}$ is $\mathbf{u}$'s absolute loss when $\mathbf{u}\cdot\mathbf{x} < y_{lo}$, and that $y_{lo} - \mathbf{u}\cdot\mathbf{x}$ is less than or equal to $\mathbf{u}$'s absolute loss, otherwise. The proof for the second inequality is similar. ■

Bounds for Perceptron

The Perceptron algorithm is given in Figure 1. The Perceptron algorithm inputs an initial weight vector $\mathbf{s}$ (typically, the zero vector $\mathbf{0}$), and a learning rate η. The perceptron update rule is applied if the prediction $\widehat{y}$ is outside the outcome interval, i.e., the current weight vector $\mathbf{w}$ is incremented (decremented) by $\eta\mathbf{x}$ if the prediction $\widehat{y}$ is too low (high). The use of any outcome interval generalizes the standard Perceptron algorithm.

The behavior of the Perceptron algorithm is bounded by the following theorem.

Theorem 2 *Let S be a sequence of l trials. Let $X_P \geq \max_t \|\mathbf{x}_t\|$. Then for any comparison vector $\mathbf{u}$ where $\|\mathbf{u}\| \leq U_P$.*

$$\text{Loss}(\text{Perceptron}(\mathbf{0},\eta),S) \leq \text{Loss}(\mathbf{u},S) + \frac{U_P^2}{2\eta} + \frac{\eta l X_P^2}{2}$$

Choosing $\eta = U_P/(X_P\sqrt{l})$ leads to:

$$\text{Loss}(\text{Perceptron}(\mathbf{0},\eta),S) \leq \text{Loss}(\mathbf{u},S) + U_P X_P \sqrt{l}$$

Proof: Let $d(\mathbf{u},\mathbf{w}) = \sum_{i=1}^n (u_i - w_i)^2$. Consider the tth trial $S_t = (\mathbf{x}_t, [y_{t,lo}, y_{t,hi}])$. If $\widehat{y}_t \in [y_{t,lo}, y_{t,hi}]$, then $\mathbf{w}_{t+1} = \mathbf{w}_t$, and $d(\mathbf{u},\mathbf{w}_t) - d(\mathbf{u},\mathbf{w}_{t+1}) = 0$. If $\widehat{y}_t < y_{t,lo}$, then $\mathbf{w}_{t+1} = \mathbf{w}_t + \eta\mathbf{x}_t$, and it follows that:

$$\begin{aligned} & d(\mathbf{u},\mathbf{w}_t) - d(\mathbf{u},\mathbf{w}_{t+1}) \\ &= \sum_{i=1}^n (u_i - w_{t,i})^2 - \sum_{i=1}^n (u_i - w_{t+1,i})^2 \\ &= \sum_{i=1}^n (u_i - w_{t,i})^2 - \sum_{i=1}^n (u_i - w_{t,i} - \eta x_{t,i})^2 \\ &= 2\eta(\mathbf{u}\cdot\mathbf{x}_t - \widehat{y}_t) - \eta^2\|\mathbf{x}_t\|^2 \\ &\geq 2\eta(\mathbf{u}\cdot\mathbf{x}_t - \widehat{y}_t) - \eta^2 X_P^2 \end{aligned}$$

From Lemma 1 and the fact that $\|\mathbf{x}_t\| \leq X_P$, it follows that:

$$\begin{aligned} & \text{Loss}(\text{Perceptron}(\mathbf{w}_t,\eta),S_t) - \text{Loss}(\mathbf{u},S_t) \\ &\leq \mathbf{u}\cdot\mathbf{x}_t - \widehat{y}_t \\ &\leq \frac{d(\mathbf{u},\mathbf{w}_t) - d(\mathbf{u},\mathbf{w}_{t+1})}{2\eta} + \frac{\eta X_P^2}{2} \end{aligned}$$

A similar analysis holds when $\widehat{y} > y_{t,hi}$. By summing over all l trials:

$$\begin{aligned} & \text{Loss}(\text{Perceptron}(\mathbf{0},\eta),S) - \text{Loss}(\mathbf{u},S) \\ &= \sum_{t=1}^l \text{Loss}(\text{Perceptron}(\mathbf{w}_t,\eta),S_t) - \text{Loss}(\mathbf{u},S_t) \\ &\leq \sum_{t=1}^l \left(\frac{d(\mathbf{u},\mathbf{w}_t) - d(\mathbf{u},\mathbf{w}_{t+1})}{2\eta} + \frac{\eta X_P^2}{2} \right) \\ &= \frac{d(\mathbf{u},\mathbf{0}) - d(\mathbf{u},\mathbf{w}_{l+1})}{2\eta} + \frac{\eta l X_P^2}{2} \\ &\leq \frac{U_P^2}{2\eta} + \frac{\eta l X_P^2}{2} \end{aligned}$$

which proves the first inequality of the theorem. The second inequality follows immediately from the choice of η. ■

Bounds for Exponentiated Update

The EU (Exponentiated Update) algorithm is given in Figure 2. The EU algorithm inputs a start vector $\mathbf{s}$, a positive learning rate η, and a positive number U_E. Every weight vector consists of positive weights that sum to U_E. Normally, each weight in the start weight vector is set to U_E/n. For each trial, if the prediction $\widehat{y}$ is outside the outcome interval, then each weight w_i in the current weight vector $\mathbf{w}$ is multiplied (divided) by $e^{\eta x_i}$ if the prediction $\widehat{y}$ is too low (high). The updated weights are normalized so that they sum to U_E.

Algorithm EU(s,η,U_E)

Parameters:

s: the start vector, with $\sum_{i=1}^{n} s_i = U_E$ and each $s_i > 0$.

η: the learning rate, with $\eta > 0$.

U_E: the sum of the weights for each weight vector, with $U_E > 0$

Initialization:

Before the first trial, set each $w_{1,i}$ to s_i.

Prediction:

Upon receiving the tth example $\mathbf{x}_t$, give the prediction $\widehat{y}_t = \mathbf{w}_t \cdot \mathbf{x}_t$

Update:

Upon receiving the tth outcome interval $[y_{t,lo}, y_{t,hi}]$, update the weight vector using:

$$w_{t+1,i} = \begin{cases} \dfrac{U_E w_{t,i} e^{\eta x_{t,i}}}{\sum_{i=1}^{n} w_{t,i} e^{\eta x_{t,i}}} & \text{if } \widehat{y}_t < y_{t,lo} \\ w_{t,i} & \text{if } \widehat{y}_t \in [y_{t,lo}, y_{t,hi}] \\ \dfrac{U_E w_{t,i} e^{-\eta x_{t,i}}}{\sum_{i=1}^{n} w_{t,i} e^{-\eta x_{t,i}}} & \text{if } \widehat{y}_t > y_{t,hi} \end{cases}$$

Figure 2: Exponentiated Update Algorithm

The EU algorithm can be used to implement the Weighted Majority Algorithm (Littlestone & Warmuth 1994). Assuming that all $x_{t,i} \in [0,1]$ and that β is the Weighted Majority's update parameter, set $\mathbf{s} = (1/n, \ldots, 1/n)$, $\eta = \ln 1/\beta$, and $U_E = 1$, and use outcome intervals of $[0, 1/2)$ of $(1/2, 1]$ for negative and positive examples, respectively. With these parameters, the EU algorithm makes the same classification decisions as the Weighted Majority algorithm.

The analysis borrows two ideas from a previous analysis of linear learning algorithms (Kivinen & Warmuth 1994): normalization of the weights so they always sum to U_E, and the relative entropy distance function. The behavior of the EU algorithm is bounded by the following theorem.

Theorem 3 *Let S be a sequence of l trials. Let $\mathbf{s} = (U_E/n, \ldots, U_E/n)$ be the start vector. Let $X_E \geq \max_{t,i} |x_{t,i}|$. Then for any comparison vector $\mathbf{u}$ where $\sum_{i=1}^{n} u_i = U_E$ and where each $u_i \geq 0$:*

$$\text{Loss}(\text{EU}(\mathbf{s}, \eta, U_E), S) \leq \text{Loss}(\mathbf{u}, S) + \frac{U_E \ln n}{\eta} + \frac{\eta l U_E X_E^2}{2}$$

Choosing $\eta = \sqrt{2 \ln n}/(X_E \sqrt{l})$ leads to:

$$\text{Loss}(\text{EU}(\mathbf{s}, \eta, U_E), S) \leq \text{Loss}(\mathbf{u}, S) + U_E X_E \sqrt{2l \ln n}$$

Proof: Let S, l, $\mathbf{s}$, X_E, and U_E be defined as in the theorem. Let $d(\mathbf{u}, \mathbf{w}) = \sum_{i=1}^{n} u_i \ln(u_i/w_i)$, where $0 \ln 0 = 0$ by definition. If the sum of $\mathbf{u}$'s weights is equal to the sum of $\mathbf{w}$'s weights, then $d(\mathbf{u}, \mathbf{w}) \geq 0$. Note that:

$$\begin{aligned} d(\mathbf{u}, \mathbf{s}) &= \sum_{i=1}^{n} u_i \ln \frac{u_i n}{U_E} \\ &= \sum_{i=1}^{n} u_i \ln n - \sum_{i=1}^{n} u_i \ln \frac{U_E}{u_i} \\ &\leq U_E \ln n \end{aligned}$$

Consider the tth trial $S_t = (\mathbf{x}_t, [y_{t,lo}, y_{t,hi}])$. Then $\widehat{y}_t = \mathbf{w}_t \cdot \mathbf{x}_t$. Now if $\widehat{y}_t \in [y_{t,lo}, y_{t,hi}]$, then $\mathbf{w}_{t+1} = \mathbf{w}_t$, and $d(\mathbf{u}, \mathbf{w}_t) - d(\mathbf{u}, \mathbf{w}_{t+1}) = 0$. If $\widehat{y}_t < y_{t,lo}$, then:

$$w_{t+1,i} = \frac{U_E w_{t,i} e^{\eta x_{t,i}}}{\sum_{j=1}^{n} w_{t,j} e^{\eta x_{t,j}}}$$

and it follows from Lemma 1 that:

$$\begin{aligned} & d(\mathbf{u}, \mathbf{w}_t) - d(\mathbf{u}, \mathbf{w}_{t+1}) \\ &= \sum_{i=1}^{n} u_i \ln \frac{u_i}{w_{t,i}} - \sum_{i=1}^{n} u_i \ln \frac{u_i}{w_{t+1,i}} \\ &= \sum_{i=1}^{n} u_i \ln w_{t+1,i} - \sum_{i=1}^{n} u_i \ln w_{t,i} \\ &= \sum_{i=1}^{n} u_i \ln \frac{U_E e^{\eta x_{t,i}}}{\sum_{j=1}^{n} w_{t,j} e^{\eta x_{t,j}}} \\ &= \sum_{i=1}^{n} u_i \ln e^{\eta x_{t,i}} - \sum_{i=1}^{n} u_i \ln \sum_{j=1}^{n} \frac{w_{t,j} e^{\eta x_{t,j}}}{U_E} \\ &= \eta \sum_{i=1}^{n} u_i x_{t,i} - \sum_{i=1}^{n} u_i \ln \sum_{j=1}^{n} \frac{w_{t,j} e^{\eta x_{t,j}}}{U_E} \\ &= \eta \mathbf{u} \cdot \mathbf{x}_t - U_E \ln \sum_{i=1}^{n} \frac{w_{t,i} e^{\eta x_{t,i}}}{U_E} \end{aligned}$$

In the appendix, it is shown that:

$$\ln \sum_{i=1}^{n} \frac{w_{t,i} e^{\eta x_{t,i}}}{U_E} \leq \frac{\eta \mathbf{w}_t \cdot \mathbf{x}_t}{U_E} + \frac{\eta^2 X_E^2}{2}$$

This implies that:

$$\begin{aligned} & d(\mathbf{u}, \mathbf{w}_t) - d(\mathbf{u}, \mathbf{w}_{t+1}) \\ &\geq \eta \mathbf{u} \cdot \mathbf{x}_t - \eta \mathbf{w}_t \cdot \mathbf{x}_t - \frac{\eta^2 U_E X_E^2}{2} \end{aligned}$$

Using Lemma 1, it follows that:

$$\begin{aligned} & \text{Loss}(\text{EU}(\mathbf{x}_t, \eta, U_E), S_t) - \text{Loss}(\mathbf{u}, S_t) \\ &\leq \mathbf{u} \cdot \mathbf{x}_t - \widehat{y}_t \\ &\leq \frac{d(\mathbf{u}, \mathbf{w}_t) - d(\mathbf{u}, \mathbf{w}_{t+1})}{\eta} + \frac{\eta U_E X_E^2}{2} \end{aligned}$$

A similar analysis holds when $\widehat{y} > y_{t,hi}$. By summing over all l trials:

$$\begin{aligned} & \text{Loss}(\text{EU}(\mathbf{s}, \eta, U_E), S) - \text{Loss}(\mathbf{u}, S) \\ &= \sum_{t=1}^{l} \text{Loss}(\text{EU}(\mathbf{w}_t, \eta, U_E), S_t) - \text{Loss}(\mathbf{u}, S_t) \\ &\leq \sum_{t=1}^{l} \left(\frac{d(\mathbf{u}, \mathbf{w}_t) - d(\mathbf{u}, \mathbf{w}_{t+1})}{\eta} + \frac{\eta U_E X_E^2}{2} \right) \\ &= \frac{d(\mathbf{u}, \mathbf{s}) - d(\mathbf{u}, \mathbf{w}_{l+1})}{\eta} + \frac{\eta l U_E X_E^2}{2} \\ &\leq \frac{U_E \ln n}{\eta} + \frac{\eta l U_E X_E^2}{2} \end{aligned}$$

which proves the first inequality of the theorem. The second inequality follows immediately from the choice of η. ∎

Theorems 2 and 3 provide similar results. They both have the form:

$$\text{Loss}(A, S) \leq \text{Loss}(\mathbf{u}, S) + O(l)$$

where l, the length of the trial sequence, is allowed to vary, and other parameters are fixed. If l is known in advance, then a good choice for the learning rate η leads to:

$$\text{Loss}(A, S) \leq \text{Loss}(\mathbf{u}, S) + O(\sqrt{l})$$

Because there can be a small absolute loss for each trial no matter the length of the sequence, all the bounds depend on l. It is not hard to generate trial sequences that approach these bounds.

The bound for the Perceptron algorithm depends on U_P and X_P, which bound the respective lengths (two-norms) of the best weight vector and the example vectors. The bound for the EU algorithm depends on U_E, the one-norm of the best weight vector (the sum of the weights); X_E, the infinity-norm of the example vectors (the maximum absolute value of any input value); and a $\ln n$ term. Thus, similar to the quadratic loss case (Cesa-Bianchi, Long, & Warmuth 1996; Kivinen & Warmuth 1994) and previous mistake bound analyses (Littlestone 1989), the EU algorithm should outperform the Perceptron algorithm when the best comparison weight vector has many small weights and the example vectors have few small values.

The bound for the EU algorithm appears restrictive because the weights of the comparison vector must be nonnegative and must sum to U_E. However, a simple transformation can expand the comparison class to include negative weights with U_E as the upper bound on the sum of the weight's absolute values (Kivinen & Warmuth 1994). This transformation doubles the number of weights, which would change the $\ln n$ term to $\ln 2n$.

Derivation of Mistake Bounds

To analyze concept learning, consider trial sequences that consist of *classification trials*, in which the outcome for each trial is either a positive or negative label. The classification version of an online algorithm is distinguished from the absolute loss version.

A *classification algorithm* classifies an example as positive if $\widehat{y} > 0$, and negative if $\widehat{y} < 0$, making no classification if $\widehat{y} = 0$. No updating is performed if the example is classified correctly. The choice of 0 for a classification threshold is convenient for the analysis; note that because Theorems 2 and 3 apply to any outcome intervals, any classification threshold can be analyzed.

An *absolute loss algorithm* uses the outcome interval $[1, \infty)$ for positive examples and the outcome interval $(-\infty, -1]$ for negative examples. An absolute loss algorithm performs updating if $\widehat{y}$ is not in the correct interval. As a result, the absolute loss of the absolute loss algorithm on a given trial is greater than or equal to the 0-1 loss of the classification algorithm (the 0-1 loss for a trial is 1 if the classification algorithm is incorrect, and 0 if correct). For the following observation, a subsequence of a trial sequence omits zero or more trials, but does not change the ordering of the remaining trials.

Observation 4 *Let S be a classification trial sequence. If a classification algorithm makes m mistakes on S, then there is a subsequence of S of length m where the corresponding absolute loss algorithm has an absolute loss of at least m. Equivalently, if there is no subsequence of S of length m where the absolute loss algorithm has an absolute loss of m or more, then the classification algorithm must make fewer than m mistakes on S.*

Based on this observation, a mistake bound for the Perceptron algorithm is derived. The notation $\text{Loss}(\cdot, \cdot)$ is used for the absolute loss of the absolute loss algorithm, and 0-1-Loss$(\cdot, \cdot)$ for the 0-1 loss of the classification algorithm.

Theorem 5 *Let S be a sequence of l classification trials. Let $X_P \geq \max_t ||\mathbf{x}_t||$. Suppose there exists a vector $\mathbf{u}$ with $||\mathbf{u}|| \leq U_P$ and $\text{Loss}(\mathbf{u}, S) = 0$. Let S' be any subsequence of S of length m. Then $m > U_P^2 X_P^2$ implies $\text{Loss}(\text{Perceptron}(\mathbf{0}, 1/X_P^2), S') < m$, which implies 0-1-Loss$(\text{Perceptron}(\mathbf{0}, 1/X_P^2), S) < m$.*

Proof: Using Theorem 2, $\text{Loss}(\mathbf{u}, S) = 0$, $\eta = 1/X_P^2$, and $m > U_P^2 X_P^2$:

$$\begin{aligned}
&\text{Loss}(\text{Perceptron}(\mathbf{0}, \eta), S') \\
&\leq \text{Loss}(\mathbf{u}, S') + \frac{U_P^2}{2\eta} + \frac{\eta m X_P^2}{2} \\
&\leq \frac{U_P^2 X_P^2}{2} + \frac{m}{2} \\
&< \frac{m}{2} + \frac{m}{2} = m
\end{aligned}$$

Because every subsequence of length m has an absolute loss less than m, then Observation 4 implies 0-1-Loss$(\text{Perceptron}(\mathbf{0}, \eta), S) < m$. ∎

Actually, the value of the learning rate does not affect the mistake bound when 0 is the classification threshold. It only affects the relative length of the current weight vector.

The mistake bound corresponds to previous mistake bounds in the literature. For example, if a unit weight vector has separation $\delta = 1/U_P$, i.e., $\mathbf{w} \cdot \mathbf{x} \geq |\delta|$ for all examples $\mathbf{x}$ in the sequence, then a weight vector of length U_P has a separation of 1. If each example $\mathbf{x}$ is also a unit vector, i.e., $X_P = 1$, then the mistake bound is $U_P^2 = 1/\delta^2$, which is identical to the bound in (Minsky & Papert 1969).

Now consider the EU algorithm.

Theorem 6 *Let S be a sequence of l classification trials. Let $X_E \geq \max_{t,i} |x_{t,i}|$. Suppose there exists a vector $\mathbf{u}$ with nonnegative weights such that $\sum_{i=1}^{n} u_i = U_E$ and $\text{Loss}(\mathbf{u}, S) = 0$. Let $\mathbf{s} = (U/n, \ldots, U/n)$. Let S' be any subsequence of S of length m. Then $m > 2U_E^2 X_E^2 \ln n$ implies $\text{Loss}(\text{EU}(\mathbf{s}, 1/(U_E X_E^2)), S') < m$, which implies 0-1-Loss$(\text{EU}(\mathbf{s}, 1/(U_E X_E^2)), S) < m$.*

Proof: Using Theorem 3, $\text{Loss}(\mathbf{u}, S) = 0$, $\eta = 1/(U_E X_E^2)$, and $m > 2U_E^2 X_E^2 \ln n$:

$$\begin{aligned}
&\text{Loss}(\text{EU}(\mathbf{s}, \eta, U_E), S') \\
&\leq \text{Loss}(\mathbf{u}, S') + \frac{U_E \ln n}{\eta} + \frac{\eta m U_E X_E^2}{2} \\
&\leq U_E^2 X_E^2 \ln n + \frac{m}{2} \\
&< m
\end{aligned}$$

Because every subsequence of length m has an absolute loss less than m, then Observation 4 implies $\text{0-1-Loss}(\text{EU}(\mathbf{s}, \eta, U_E), S) < m$. ∎

While the learning rate is important for the EU classification algorithm, the normalization by U_E is unnecessary. The normalization affects the sum of the weights, but not their relative sizes.

This mistake bound is comparable to mistake bounds for the Weighted Majority algorithm and the Balanced algorithm in (Littlestone 1989).[1] There, $X_E = 1$ and comparison vectors have a separation of δ with weights that sum to 1. To get a separation of 1, the sum of the weights needs to be $U_E = 1/\delta$. Under these assumptions, the new bounds are also $O(\ln n/\delta^2)$.

Toleranced Absolute Loss and Randomized Classification Algorithms

The analysis leads to a per-trial loss for both algorithms, so consider an extension in which the goal is come within τ of each outcome interval rather than directly hitting the interval itself. The notation $\text{Loss}(\cdot, S, \tau)$, where the tolerance τ is nonnegative, indicates that every outcome interval $[y_{lo}, y_{hi}]$ of each trial in the trial sequence S is modified to $[y_{lo} - \tau, y_{hi} + \tau]$. The absolute loss is calculated in accordance with the modified outcome intervals.

For the Perceptron and EU algorithms, the analysis leads to an additional per-trial loss of $\eta X_P^2/2$ and $\eta U_E X_E^2/2$, respectively. If τ is equal to these values, then it turns out that the per-trial loss can be eliminated. It is not difficult to generalize the proofs for Theorems 2 and 3 to obtain the following theorems:

Theorem 7 *Let S be a sequence of l trials and τ be a positive real number. Let $X_P \geq \max_t \|\mathbf{x}_t\|$ and $\eta = 2\tau/X_P^2$. Then for any comparison vector $\mathbf{u}$ where $\|\mathbf{u}\| \leq U_P$.*

$$\text{Loss}(\text{Perceptron}(\mathbf{0}, \eta), S, \tau) \leq \text{Loss}(\mathbf{u}, S) + \frac{U_P^2 X_P^2}{4\tau}$$

Theorem 8 *Let S be a sequence of l trials and τ be a positive real number. Let $\mathbf{s} = (U_E/n, \ldots, U_E/n)$ be the start vector. Let $X_E \geq \max_{t,i} |x_{t,i}|$ and $\eta = 2\tau/(U_E X_E^2)$. Then for any comparison vector $\mathbf{u}$ where $\sum_{i=1}^n u_i = U_E$ and where each $u_i \geq 0$:*

$$\text{Loss}(\text{EU}(\mathbf{s}, \eta, U_E), S, \tau) \leq \text{Loss}(\mathbf{u}, S) + \frac{U_E^2 X_E^2 \ln n}{2\tau}$$

[1] In (Littlestone 1989), the Weighted Majority algorithm is also analyzed as a general linear threshold learning algorithm in addition to an analysis as a "master" algorithm as in (Littlestone & Warmuth 1994).

For both algorithms, the toleranced absolute loss of each algorithm exceeds the (non-toleranced) absolute loss of the best comparison vector by a constant over the whole sequence, no matter how long the sequence is. If the best comparison vector has a zero absolute loss, then the toleranced absolute loss is bounded by a constant over the whole sequence. These results strongly support the claim that the Perceptron and EU algorithms are online algorithms for minimizing absolute loss.

To apply these theorems, again consider concept learning and classification trial sequences. A *randomized classification algorithm* for a classification trial sequence is defined as follows. The absolute loss algorithm is performed on the sequence using a tolerance of $\tau = 1/2$, and outcome intervals of $[1, \infty)$ and $(-\infty, -1]$ for positive and negative classification trials, respectively. The prediction $\hat{y}$ is converted into a classification prediction by predicting positive if $\hat{y} \geq 1/2$, and negative if $\hat{y} \leq -1/2$. If $-1/2 < \hat{y} < 1/2$, then predict positive with probability $\hat{y} + 1/2$, otherwise predict negative. I assume that the method for randomizing this prediction is independent of the outcome intervals. When $-1/2 < \hat{y} < 1/2$, updating is performed regardless of whether the classification prediction is correct or not.

The idea of a randomized algorithm is borrowed from (Littlestone & Warmuth 1994), which analyzes a randomized version of the Weighted Majority algorithm. This paper's randomization differs in that there are ranges of $\hat{y}$ where positive and negative predictions are deterministic.

Note that the toleranced absolute loss of the randomized classification algorithm on a classification trial (referring to the $\hat{y}$ prediction) is equal to the probability of an incorrect classification prediction if $-1/2 < \hat{y} < 1/2$. Otherwise, the toleranced absolute loss is 0 for correct classification predictions and at least 1 for incorrect predictions. In all cases, the toleranced absolute loss is greater than or equal to the expected value of the 0-1 loss. This leads to the following observation.

Observation 9 *Let S be a classification trial sequence. Then, the toleranced absolute loss of a randomized classification algorithm on S is greater than or equal to the expected value of the algorithm's 0-1 loss on S.*

The notation $\text{Loss}(\cdot, \cdot, 1/2)$ is used for the toleranced absolute loss of the randomized classification algorithm, and $\text{0-1-Loss}(\cdot, \cdot, 1/2)$ for its 0-1 loss.

Theorem 10 *Let S be a sequence of l classification trials. Let $X_P \geq \max_t \|\mathbf{x}_t\|$. Suppose there exists a vector $\mathbf{u}$ with $\|\mathbf{u}\| \leq U_P$ and $\text{Loss}(\mathbf{u}, S) = 0$. Then $\text{Loss}(\text{Perceptron}(\mathbf{0}, 1/X_P^2), S, 1/2) \leq U_P^2 X_P^2/2$, which implies $E[\text{0-1-Loss}(\text{Perceptron}(\mathbf{0}, 1/X_P^2), S, 1/2)] \leq U_P^2 X_P^2/2$.*

Proof: Using Theorem 7, $\text{Loss}(\mathbf{u}, S) = 0$, $\eta = 1/X_P^2$, and $\tau = 1/2$:

$$\begin{aligned}
\text{Loss}(\text{Perceptron}(\mathbf{0}, \eta), S, \tau) &\leq \text{Loss}(\mathbf{u}, S) + \frac{U_P^2 X_P^2}{4\tau} \\
&= U_P^2 X_P^2/2
\end{aligned}$$

Observation 9 implies $E[\text{0-1-Loss}(\text{Perceptron}(\mathbf{0}, \eta), S, \tau)] \leq U_P^2 X_P^2/2$. ■

Theorem 11 *Let S be a sequence of l classification trials. Let $X_E \geq \max_{t,i} |x_{t,i}|$. Suppose there exists a vector $\mathbf{u}$ of nonnegative weights with $\sum_{i=1}^n u_i \leq U_E$ and $\text{Loss}(\mathbf{u}, S) = 0$. Let $\mathbf{s} = (U_E/n, \ldots, U_E/n)$. Then $\text{Loss}(\text{EU}(\mathbf{s}, 1/(U_E X_E^2)), S, 1/2) \leq U_E^2 X_E^2 \ln n$, which implies $E[\text{0-1-Loss}(\text{EU}(\mathbf{s}, 1/(U_E X_E^2)), S, 1/2)] \leq U_E^2 X_E^2 \ln n$.*

Proof: Using Theorem 8, $\text{Loss}(\mathbf{u}, S) = 0$, $\eta = 1/(U_E X_E^2)$, and $\tau = 1/2$:

$$\begin{aligned} \text{Loss}(\text{EU}(\mathbf{s}, \eta), S, \tau) &\leq \text{Loss}(\mathbf{u}, S) + \frac{U_E^2 X_E^2 \ln n}{2\tau} \\ &= U_E^2 X_E^2 \ln n \end{aligned}$$

Observation 9 implies $E[\text{0-1-Loss}(\text{EU}(\mathbf{s}, \eta), S, \tau)] \leq U_E^2 X_E^2$. ■

For both randomized algorithms, the worst-case bounds on the expected 0-1 loss is half of the worst-case mistake bounds of the deterministic algorithms. Roughly, randomization improves the worse-case bounds because a value of $\hat{y}$ close to 0 has a 0-1 loss of 1 in the deterministic worst case, while the expected 0-1 loss is close to $1/2$ for the randomized algorithms.

Conclusion

I have presented an analysis of the Perceptron and exponentiated update algorithms that shows that they are online algorithms for minimizing the absolute loss over a sequence of trials. Specifically, this paper shows that the worst-case absolute loss of the online algorithms is comparable to the optimal comparison vector from a class of comparison vectors.

When a classification trial sequence is linearly separable, I have also shown the relation of the absolute loss bounds to mistake bounds for both deterministic and randomized versions of these algorithms. Future research will study the classification behavior of these algorithms when the target comparison vector is allowed to drift, and when the trial sequence is not linearly separable.

Based on minimizing absolute loss, it is possible to derive a backpropagation learning algorithm for multiple layers of linear threshold units. It would be interesting to determine suitable initial conditions and parameters that lead to good performance.

References

Bylander, T. 1994. Learning linear-threshold functions in the presence of classification noise. In *Proc. Seventh Annual ACM Conf. on Computational Learning Theory*, 340–347.

Cesa-Bianchi, N.; Long, P. M.; and Warmuth, M. K. 1996. Worst-case quadratic loss bounds for a generalization of the Widrow-Hoff rule. *IEEE Transactions on Neural Networks* 7:604–619.

Gallant, S. I. 1990. Perceptron-based learning algorithms. *IEEE Trans. on Neural Networks* 1:179–191.

Kashyap, R. L. 1970. Algorithms for pattern classification. In Mendel, J. M., and Fu, K. S., eds., *Adaptive, Learning and Pattern Recognition Systems: Theory and Applications*. New York: Academic Press. 81–113.

Kivinen, J., and Warmuth, M. K. 1994. Exponentiated gradient versus gradient descent for linear predictors. Technical Report UCSC-CRL-94-16, Univ. of Calif. Computer Research Lab, Santa Cruz, California. An extended abstract appeared in *STOC '95*, pp. 209–218.

Littlestone, N., and Warmuth, M. K. 1994. The weighted majority algorithm. *Information and Computation* 108:212–261.

Littlestone, N. 1988. Learning quickly when irrelevant attributes abound: A new linear-threshold algorithm. *Machine Learning* 2:285–318.

Littlestone, N. 1989. *Mistake Bounds and Logarithmic Linear-threshold Learning Algorithms*. Ph.D. Dissertation, Univ. of Calif., Santa Cruz, California.

Minsky, M. L., and Papert, S. A. 1969. *Perceptrons*. Cambridge, Massachusetts: MIT Press.

Rosenblatt, F. 1962. *Principles of Neurodynamics*. New York: Spartan Books.

Warmuth, M. K. 1996. personal communication.

Inequality for Exponentiated Update

Lemma 12 *Let $\mathbf{w} \in \Re^n$ consist of nonnegative weights with $\sum_{i=1}^n w_i = U_E$. Let $\mathbf{x} \in \Re^n$ such that $X_E \geq \max_i |x_i|$. Let η be any real number. Then the following inequality holds:*

$$\ln \sum_{i=1}^n \frac{w_i e^{\eta x_i}}{U_E} \leq \frac{\eta \mathbf{w} \cdot \mathbf{x}}{U_E} + \frac{\eta^2 X_E^2}{2}$$

Proof: Before proving the inequality, define f as:

$$f(\eta, \mathbf{w}, \mathbf{x}) = \ln \sum_{i=1}^n \frac{w_i e^{\eta x_i}}{U_E}$$

To prove the inequality, differentiate f with respect to η.

$$\frac{\partial f}{\partial \eta} = \frac{\sum_{i=1}^n w_i x_i e^{\eta x_i}}{\sum_{i=1}^n w_i e^{\eta x_i}}$$

$$\frac{\partial^2 f}{\partial \eta^2} = \frac{\sum_{i=1}^n w_i x_i^2 e^{\eta x_i}}{\sum_{i=1}^n w_i e^{\eta x_i}} - \left(\frac{\sum_{i=1}^n w_i x_i e^{\eta x_i}}{\sum_{i=1}^n w_i e^{\eta x_i}} \right)^2$$

When $\eta = 0$, $f(\eta, \mathbf{w}, \mathbf{x}) = 0$ and $\partial f/\partial \eta = \mathbf{w} \cdot \mathbf{x}/U_E$. With regard to the second partial derivative, we have:

$$\frac{\partial^2 f}{\partial \eta^2} \leq \frac{\sum_{i=1}^n w_i x_i^2 e^{\eta x_i}}{\sum_{i=1}^n w_i e^{\eta x_i}} \leq \frac{X_E^2 \sum_{i=1}^n w_i e^{\eta x_i}}{\sum_{i=1}^n w_i e^{\eta x_i}} = X_E^2$$

for any value of η. Hence, we have:

$$f(\eta, \mathbf{w}, \mathbf{x}) \leq \frac{\eta \mathbf{w} \cdot \mathbf{x}}{U_E} + \frac{\eta^2 X_E^2}{2}$$

which is the inequality of the lemma. ■