Machine Learning, 43, 173–210, 2001

General Convergence Results for Linear Discriminant Updates

ADAM J. GROVE* grove@pobox.com
NICK LITTLESTONE† nlittlestone@mindspring.com
DALE SCHUURMANS** dale@cs.uwaterloo.ca
NEC Research Institute, 4 Independence Way, Princeton NJ 08540, USA

Editor: Yoram Singer

Abstract. The problem of learning linear-discriminant concepts can be solved by various mistake-driven update procedures, including the *Winnow* family of algorithms and the well-known *Perceptron* algorithm. In this paper we define the general class of "quasi-additive" algorithms, which includes Perceptron and Winnow as special cases. We give a single proof of convergence that covers a broad subset of algorithms in this class, including both Perceptron and Winnow, but also many new algorithms. Our proof hinges on analyzing a generic *measure of progress* construction that gives insight as to when and how such algorithms converge.

Our measure of progress construction also permits us to obtain good mistake bounds for individual algorithms. We apply our unified analysis to new algorithms as well as existing algorithms. When applied to known algorithms, our method "automatically" produces close variants of existing proofs (recovering similar bounds)—thus showing that, in a certain sense, these seemingly diverse results are fundamentally isomorphic. However, we also demonstrate that the unifying principles are more broadly applicable, and analyze a new class of algorithms that smoothly interpolate between the additive-update behavior of Perceptron and the multiplicative-update behavior of Winnow.

Keywords: Winnow, Perceptron, on-line learning, mistake bounds, Bregman divergence

1. Introduction

Many iterative mistake-driven algorithms have been proposed for learning linear-discriminant concepts from examples, including the famous *Perceptron* algorithm (Rosenblatt, 1962; Minsky & Papert, 1969; Duda & Hart, 1973) and Littlestone's *Winnow* family of algorithms (Littlestone, 1988, 1989, 1991; Kivinen, Warmuth & Auer, 1997). This is an important, well-studied, collection of algorithms with interesting properties and practical applications (Blum, 1997; Dagan, Karov & Roth, 1997; Golding & Roth, 1999; Khardon, Roth & Valiant, 1999). In this paper we define a general class of algorithms and provide a unified theoretical analysis which covers not only Perceptron and Winnow, but also many new algorithms.

All of the algorithms we consider represent linear-discriminant concepts by maintaining

*Present address: Netli Inc., 844 E. Charleston Rd, Palo Alto, CA 94303, USA.
†Present address: 1987 Smith Grade, Santa Cruz, CA 95060, USA.
**Present address: Department of Computer Science, University of Waterloo, Waterloo, Ontario N2L 3G1, Canada.

a current weight vector $\mathbf{w} \in \mathbb{R}^n$ and classifying instances $\mathbf{x} \in \mathbb{R}^n$ into classes labeled ± 1 according to the rule $\mathbf{x} \mapsto \text{sign}(\mathbf{w} \cdot \mathbf{x})$.[1] It is most natural to think of these algorithms as working in an on-line setting where learning occurs in a sequence of trials. In trial i an *instance* $\mathbf{x}_i$ is observed. The algorithm makes the prediction $\text{sign}(\mathbf{w} \cdot \mathbf{x}_i)$ and then observes the *label* y_i. (We speak of a pair $(\mathbf{x}_i, y_i)$ as an *example*.) The algorithms we consider update the current weight vector $\mathbf{w}$ if and only if they make a mistake (that is, if the prediction does not match the label). The various known algorithms of this type differ in the specific policies they use to update the weight vector. For example, Perceptron uses an *additive* update policy, whereas the Winnow algorithms use *multiplicative* updates.

A striking fact about these algorithms is that they *converge*, that is, they make a finite number of mistakes when given a sequence of examples labeled by a target linear-discriminant concept—provided there is a gap between the positive and negative examples (i.e., there exists two parallel separating hyper-planes such that no example in the sequence falls between them). This is true even for infinite sequences of examples.

However, beyond mere convergence, the classical proofs of Perceptron convergence (for instance (Minsky & Papert, 1969; Duda & Hart, 1973)) and Littlestone's proofs for members of the Winnow family (Littlestone, 1988, 1989) also provide *bounds* on the number of mistakes made before a perfect classifier is found for the sequence. In all cases, this bound depends on the width of the gap and is independent of the number of examples in the sequence. Interestingly, these proofs all have the same overall structure: one postulates a function of the weight vector, which we call a *measure of progress*, and proves that it (eventually) makes progress towards a solution after each mistake. However, aside from this broad similarity, the proofs seem quite distinct. Moreover, since the various measures of progress seem to have been discovered on a case-by-case basis, these proofs do not seem to help very much in identifying or analyzing new algorithms.

The first contribution of this paper, presented in Section 2, is a simple unifying framework for expressing algorithms that includes both Perceptron and Winnow as special cases. We accomplish this by expressing the Winnow algorithms in a simple *quasi-additive* form that makes clear how fundamentally similar they are to the Perceptron procedure. This leads us to define a general class of algorithms which includes both Perceptron and Winnow as special cases, but also includes new algorithms that have not been previously studied.

One of the central contributions of this paper is the introduction of a general approach for understanding and constructing measures of progress for analyzing quasi-additive algorithms. We outline the motivating ideas for our approach in Section 3. As will be apparent there, our basic method can be completed using one of several slightly different elaborations. A particularly straightforward version is used in Section 4 to prove a general convergence theorem for a large class of quasi-additive algorithms. Specifically, we characterize a general class of quasi-additive algorithms (including known ones) that converge under the stated conditions. This result, however, has the drawback that it does not give strong bounds.

In Section 5 we discuss a second version of our basic strategy. This leads to more refined measures of progress, usually capable of generating much tighter mistake bounds for individual algorithms. Interestingly, for both Perceptron and Winnow our method gives the same or very similar results to existing analyses, and our measure of progress reduces to variants of the traditional measures (Section 6). For example, for Perceptron, our method

yields the same measure of progress used in one of the most famous proofs of Perceptron convergence (Papert, 1961; Minsky & Papert, 1969). When applied to the Winnow family, our construction leads to almost exactly the same measures of progress used by Littlestone in (1989). Thus, we show that, in a certain sense, the tacit principles by which these previous measures of progress were developed are the same, and are well captured by our generic method.

Then in Section 7 we apply our method to analyze a new family of quasi-additive learning algorithms that has not been previously investigated. We call these algorithms *p-norm Perceptron algorithms*. Their convergence is assured by the results in Section 4. In proving specific mistake bounds for this family we reveal two interesting facts. First, as the parameter p is varied, one can "interpolate" between the additive Perceptron algorithm and multiplicative Winnow algorithms in a flexible and principled fashion. Second, bounds for these algorithms can be given that depend on particular products of *conjugate norms* (see Section 7) varying with the parameter p, offering the possibility that the new algorithms may be superior to both Perceptron and Winnow in certain contexts.

Section 8 contains a short discussion of some other possible measures of progress and how they relate to ours. We close, in Sections 9–11, with a discussion of other related work, some speculation about future work, and our conclusions.

2. Quasi-additive algorithms

An on-line mistake-driven algorithm is determined by its *update* rule; i.e., how it revises $\mathbf{w}$ when it makes a mistake. Perceptron is very simple: it just adds some multiple, with a suitable sign, of the vector on which a mistake was made.

Perceptron$(\mathbf{w}, (\mathbf{x}, y))$**:**
  If $\text{sign}(\mathbf{w} \cdot \mathbf{x}) \neq y$ then
    $\mathbf{w} := \mathbf{w} + ay\mathbf{x}$

Here the parameter a is a positive constant.

Littlestone has analyzed a number of other algorithms for learning linear-discriminant concepts. These algorithms are closely related to one another; we refer to them collectively as the *Winnow* family. On the surface, these might seem very different from Perceptron. But as we now show, there is a fundamental similarity. Consider the balanced version of Winnow described in Littlestone (1989, 1995). As it is normally presented, it maintains two weight vectors, $\mathbf{w}^+$ and $\mathbf{w}^-$, and updates each separately:

BalancedWinnow$(\mathbf{w}^+, \mathbf{w}^-, (\mathbf{x}, y))$**:**
  If $\text{sign}(\mathbf{w}^+ \cdot \mathbf{x} - \mathbf{w}^- \cdot \mathbf{x}) \neq y$ then
    If $y = +1$ then, for all i,
      $w_i^+ := \beta^{-x_i} w_i^+; \quad w_i^- := \beta^{x_i} w_i^-$
    Else if $y = -1$ then, for all i,
      $w_i^+ := \beta^{x_i} w_i^+; \quad w_i^- := \beta^{-x_i} w_i^-$

(Here β is a parameter between 0 and 1.) The key observation is that this is equivalent to a procedure that simply keeps track of the scaled sum of the mistake vectors, as Perceptron

does (with the scale factor $a = \log(1/\beta)$), but then computes the final weight vector as a function of this sum: $f(z) = e^z - e^{-z}$ $(= 2\sinh(z))$.[2] Thus we can re-express the algorithm as follows:

BalancedWinnow$(\mathbf{w}, \mathbf{z}, (\mathbf{x}, y))$**:**
If $\text{sign}(\mathbf{w} \cdot \mathbf{x}) \neq y$ then
$\mathbf{z} := \mathbf{z} + (\log \frac{1}{\beta})\, y\, \mathbf{x}$
$\mathbf{w} := 2\,\mathbf{sinh}(\mathbf{z})$ (i.e., $w_i = 2\sinh(z_i)$ for all i)[3]

Thus, Balanced Winnow is just like Perceptron with the exception that it classifies examples using a transformed version of the sum-of-mistakes vector, using the componentwise transformation $f(z_i) = 2\sinh(z_i)$. This motivates a natural class of "generalized" Perceptron algorithms. The idea is to distinguish the cumulative sum of mistake vectors $\mathbf{z}$ from the final weight vector $\mathbf{w}$ that we actually use to classify examples. The latter will be a transformed version of the former, based on some transformation function f applied componentwise to $\mathbf{z}$; different functions f lead to different algorithms. We call this general family of algorithms *quasi-additive*, as they essentially involve an additive update (i.e., to $\mathbf{z}$) at their core. The quasi-additive algorithm $\mathbf{QA}\langle f \rangle$ constructed from function f is:

$\mathbf{QA}\langle f \rangle$ $(\mathbf{w}, \mathbf{z}, (\mathbf{x}, y))$**:**
If $\text{sign}(\mathbf{w} \cdot \mathbf{x}) \neq y$ then
$\mathbf{z} := \mathbf{z} + ay\mathbf{x}$
$\mathbf{w} := \mathbf{f}(\mathbf{z})$ (i.e., $w_i = f(z_i)$ for all i)

We assume the initial value of $\mathbf{z}$ is $\mathbf{0}$ unless specified otherwise.

We can express the other algorithms in the Winnow family as members of the quasi-additive family. For example, the algorithm called *Weighted Majority* in Littlestone & Warmuth (1989) and Littlestone (1989) is equivalent[4] to another quasi-additive procedure defined by choosing $f(z) = e^z$ and setting $a = \frac{1}{2}\log\frac{1}{\beta}$.

Finally, the *Fixed Threshold* variant of the Winnow algorithm (Littlestone, 1988; Littlestone, 1989) can also be expressed in quasi-additive form, if we make one minor extension. In general, we can consider quasi-additive functions that use a different function f_i for each component. The Fixed Threshold algorithm can be expressed using $f_i(x_i) = e^{x_i}$ for all $i \leq n$, but adding an $n+1$'st component such that f_{n+1} is a constant function (essentially, the negative of the threshold value). Nevertheless, in the current paper we are primarily concerned with algorithms defined using a single function f.

The family of quasi-additive learning algorithms extends well beyond the known algorithms; in fact we obtain *some* procedure for any choice of f. Clearly we should not expect every f to yield a reasonable algorithm. The only "reasonableness" condition we mention now is that f be a continuous monotonically increasing function defined on all of $\mathbb{R}$ and not everywhere negative. We assume this in the remainder of the paper. We show in the Section 4 that a few simple additional (and broad) conditions on f suffice to guarantee *convergence* for $\mathbf{QA}\langle f \rangle$. This raises the hope that we might be able to discover new algorithms that actually perform better than Perceptron and Winnow in some cases. Before presenting this convergence result, however, we first give an overview of our general technique for analyzing quasi-additive algorithms.

3. Measures of progress

The main contribution of this paper is to present a particular approach to analyzing quasi-additive algorithms. In this section we present the key ideas, deferring various details and examples to later sections.

Throughout this paper, $S \subseteq \mathbb{R}^n \times \{\pm 1\}$ denotes a fixed (but possibly infinite) set of labeled *training examples*, which are presented to the learning algorithm in some order. By assumption, the examples are linearly separable; let $\mathbf{u} \in \mathbb{R}^n$ be a fixed *target vector*, that is, a vector such that for all $(\mathbf{x}, y) \in S$ we have $\text{sign}(\mathbf{u} \cdot \mathbf{x}) = y$.

Mistake bounds for quasi-additive algorithms usually depend on a quantity we call the *gap*. In this paper, we define the gap as a function of both $\mathbf{u}$ and S:

$$\delta_{\mathbf{u},S} = \inf_{(\mathbf{x},y)\in S} \mathbf{u} \cdot (y\mathbf{x}).$$

(We sometimes omit the subscripts and write just δ if the context is clear.) Roughly speaking, the gap bounds how close the examples are to the separating hyper-plane defined by $\mathbf{u}$. Generally, a (noise-free) convergence result, or *mistake bound*, for a quasi-additive algorithm will assume there is some lower bound on the size of the gap over the training set S.

Our central claim is that quasi-additive algorithms can be (and generally have been, even if this was not explicit) analyzed by examining the relationship between the dot product $\mathbf{u} \cdot \mathbf{z}$ and a specific function $G(\mathbf{z})$ defined below. We also introduce an important auxiliary function $H_{\mathbf{u}}(\mathbf{z})$ that will be helpful in relating the two.

First, the significance of $\mathbf{u} \cdot \mathbf{z}$ is straightforward and indeed is largely conventional within existing proofs. Consider how $\mathbf{u} \cdot \mathbf{z}$ changes when we make a single update to $\mathbf{z}$ by adding a vector $a\, y\, \mathbf{x}$ for some example $(\mathbf{x}, y) \in S$: We have

$$\begin{aligned}
\Delta_{\mathbf{u}\cdot\mathbf{z}} &= \mathbf{u} \cdot \mathbf{z}_{\text{new}} - \mathbf{u} \cdot \mathbf{z} \\
&= \mathbf{u} \cdot (\mathbf{z}_{\text{new}} - \mathbf{z}) \\
&= \mathbf{u} \cdot (ay\mathbf{x}) \\
&\geq a\, \delta_{\mathbf{u},S}
\end{aligned}$$

using the definition of the gap together with the positivity of a. Thus, $\mathbf{u} \cdot \mathbf{z}$ grows steadily with each update to $\mathbf{z}$, and after m updates we must have $\mathbf{u} \cdot \mathbf{z} \geq m\, a\, \delta_{\mathbf{u},S}$. The simplicity of this argument is perhaps the key consequence of the additive nature of quasi-additive algorithms.

The remainder of our general analysis strategy is to find some scalar function $H_{\mathbf{u}}(\mathbf{z})$ such that for all $\mathbf{u}$: $H_{\mathbf{u}}(\mathbf{z}) \geq \mathbf{u} \cdot \mathbf{z}$ for all $\mathbf{z}$; and $H_{\mathbf{u}}(\mathbf{z})$ eventually grows more slowly than $\mathbf{u} \cdot \mathbf{z}$ (by an amount bounded away from zero) as we continue to make mistakes. Supposing we can do this, we can then define a plausible measure of progress as

$$M(\mathbf{z}) = H(\mathbf{z}) - \mathbf{u} \cdot \mathbf{z}$$

(We generally suppress the possible dependence of H and M upon $\mathbf{u}$ in the notation.) On one hand, M is non-negative by the first assumption. On the other hand, after sufficiently

many mistakes the second term must increase faster than the first, showing that M would eventually become negative. This contradiction shows that there must be a finite limit to the number of updates (mistakes) our algorithm can make.

The challenge, then, is to find a suitable H. We do so by introducing another (and in some sense more fundamental) function G as follows. Let f be the transformation function defining our quasi-additive algorithm. Let t_0 be that point such that $f(t_0) = 0$; let t_0 be $-\infty$ if no such point exists. (Recall that f is monotonically increasing, so t_0 is unique.) We then define g to be a particular integral function of f:

$$g(x) = \int_{t_0}^{x} f(s)\, ds$$

for all x. (All our later results will include sufficient restrictions to ensure that g exists.) Finally, we let

$$G(\mathbf{z}) = \sum_{i=1}^{n} g(z_i). \tag{1}$$

(Note that $G(\mathbf{z}) \geq 0$ since, by construction, g is a positive function.) Now consider any function H that is a monotonic rescaling of G: i.e., such that there exists a monotonically increasing scalar function ψ (possibly parameterized by $\mathbf{u}$) such that

$$H(\mathbf{z}) = \psi(G(\mathbf{z})).$$

We restrict attention to increasing differentiable ψ that validate our first desired constraint on H; that is, we consider only ψ that preserve the inequality $\psi(G(\mathbf{z})) \geq \mathbf{u} \cdot \mathbf{z}$ for all $\mathbf{z}$. (As we show in later sections, there are various automatic rules for finding such ψ.)

We now propose that this construction of H yields a plausible function to use in our measure of progress. To see this, recall that the main issue is ensuring that $H(\mathbf{z})$ does not grow too fast as we update $\mathbf{z}$. But consider how such an H would change as we update $\mathbf{z}$, at a mistake, by adding $a\, y\, \mathbf{x}$. To a first-order approximation we have

$$\begin{aligned}
\Delta_H &\approx \nabla H(\mathbf{z}) \cdot (\mathbf{z}_{\text{new}} - \mathbf{z}) \\
&= \psi'(G(\mathbf{z}))\mathbf{f}(\mathbf{z}) \cdot (ay\mathbf{x}) \\
&= ay\psi'(G(\mathbf{z}))\mathbf{w} \cdot \mathbf{x} \\
&\leq 0.
\end{aligned}$$

Here $\nabla H(\mathbf{z})$ denotes the gradient of H at $\mathbf{z}$. The second step uses the chain rule, the definition of g as an integral of f (so g's derivative is f), and the fact that $\mathbf{z}$ is updated simply by adding $a\, y\, \mathbf{x}$. The penultimate step uses the definition of the quasi-additive rule $\mathbf{QA}\langle f \rangle$, i.e., that $\mathbf{f}(\mathbf{z})$ is the current weight vector $\mathbf{w}$. The final step uses the assumption that ψ is an increasing function (so its derivative is positive) and, most critically, the fact that $a\, y\, \mathbf{x}$ is a vector on which a *mistake* has just been made. This implies $y\,(\mathbf{w} \cdot \mathbf{x}) \leq 0$ (for otherwise, the prediction using $\mathbf{w}$ would not have been mistaken).

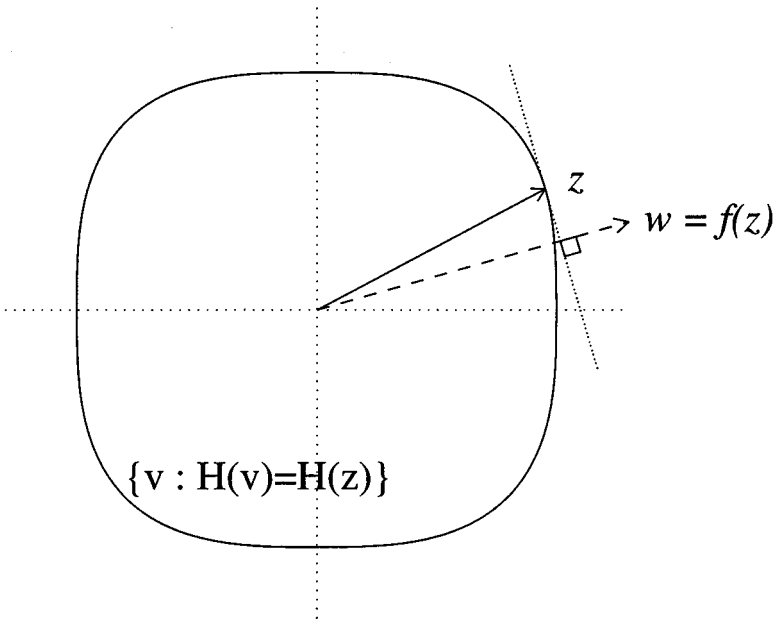

Figure 1. A two dimensional depiction of the surface $\mathcal{H}_{\mathbf{z}} = \{\mathbf{v} : H(\mathbf{v}) = H(\mathbf{z})\}$ and its tangent plane at $\mathbf{z}$.

Therefore, at least to a first-order approximation, H *decreases* after an update. This motivates our interest in considering functions H that depend on $\mathbf{z}$ only through G in this manner. It will also be the only use we ever make of the mistake-driven nature of the algorithms.

A geometric intuition might be helpful to further understand this construction for H: For any $\mathbf{z}$, consider the level surface passing through $\mathbf{z}$ defined by $\mathcal{H}_{\mathbf{z}} = \{\mathbf{v} : H(\mathbf{v}) = H(\mathbf{z})\}$, which is depicted for a simple two dimensional case in figure 1. The function G is convex (since f is increasing and thus g is convex) and ψ is increasing, therefore $H(\mathbf{z}') \leq H(\mathbf{z})$ for all $\mathbf{z}'$ in the interior of the region bounded by $\mathcal{H}_{\mathbf{z}}$. The key property of H is that the transformed weight vector $\mathbf{w} = \mathbf{f}(\mathbf{z})$ (componentwise) is *normal* to the surface $\mathcal{H}_{\mathbf{z}}$ at $\mathbf{z}$, and hence determines the tangent plane to $\mathcal{H}_{\mathbf{z}}$ at $\mathbf{z}$, $\{\mathbf{v} : (\mathbf{v} - \mathbf{z}) \cdot \mathbf{w} = 0\}$. When we move away from $\mathbf{z}$ in the direction $a_y\mathbf{x}$ (where $a_y\mathbf{x}$ is an example on which we make a mistake), we are constrained to move on one side of this plane—in particular, the side corresponding to the *interior* of the region bounded by $\mathcal{H}_{\mathbf{z}}$. So, at least to a first-order approximation, H does not increase.

In summary, we have (or will have, once we have shown how to choose ψ, as we do in later sections) given a generic recipe for deriving measures of progress that can plausibly lead to mistake bounds for quasi-additive algorithms. In the past, constructing measures of progress has (apparently) required considerable ingenuity.

Constructing a measure of progress, however, is only half a mistake bound analysis; one must also analyze the chosen H rigorously to see how fast it really does grow relative to

$\mathbf{u} \cdot \mathbf{z}$. The remaining issue is to account for the curvature of H. To be complete, we will have to analyze

$$\Delta_H = \nabla H(\mathbf{z}) \cdot (\mathbf{z}_{\text{new}} - \mathbf{z}) + R(\mathbf{z}_{\text{new}}, \mathbf{z}) \tag{2}$$

where $R(\mathbf{z}_{\text{new}}, \mathbf{z}) \stackrel{\triangle}{=} H(\mathbf{z}_{\text{new}}) - H(\mathbf{z}) - \nabla H(\mathbf{z}) \cdot (\mathbf{z}_{\text{new}} - \mathbf{z})$ is a residual term that accounts for the high order change due to the curvature of H. We will be required to prove that this growth is (eventually) bounded by a sufficiently small quantity at every update. In this paper we demonstrate a few standard techniques that help in doing this, notably analyses which study the *second* derivative of H more carefully. Beyond this, however, we do not have any especially powerful or unifying theory to solve this second stage of the analysis, and so considerable further work remains.

In the next section we give a concrete application of the ideas just discussed, showing how they appear and are used in practice. Specifically, we show that these ideas lead straightforwardly to a very broad convergence result for quasi-additive algorithms.

4. A general convergence theorem

In this section we focus on giving conditions under which algorithms make a bounded number of mistakes, but we do not pay close attention to the size of the bound. The particular notion of convergence we use, formalized in the following theorem, states that the number of mistakes made on the training set S depends only on the gap (suitably normalized by the size of the target vector), and on the size of the vectors in S. In particular, when the theorem applies, the bound does not depend on the cardinality of S.

Notation. Recall that, for $k \in \mathbb{R}^+$, $\|\mathbf{x}\|_k$ denotes the k-norm of $\mathbf{x}$; i.e., $\|\mathbf{x}\|_k = (\sum_{i=1}^n |x_i|^k)^{1/k}$. We also use $\|\mathbf{x}\|_\infty = \max_i |x_i|$, and let $\|S\|_k$ denote $\sup_{\mathbf{x} \in S} \|\mathbf{x}\|_k$.

Theorem 4.1. *Suppose f is monotonically increasing, has a continuous first derivative, is odd (i.e., $f(-z) = -f(z)$ for $z \in \mathbb{R}$), and*

$$\lim_{z \to \infty} \sup_{0 \le v < z} \frac{f'(v)}{f(z)} = c$$

for some finite $c \ge 0$.

Then there exist functions $m_{\delta,\beta,a} < \infty$ and $a_{\delta,\beta} > 0$ such that $\mathbf{QA}\langle f \rangle$, when run with parameter $a = a_{\delta,\beta}$, makes at most $m_{\delta,\beta,a}$ mistakes when trained on any set of examples S such that $\|S\|_\infty = \beta$ is finite and $\delta_{\mathbf{u},S} > 0$ for some $\mathbf{u}$.

If $c = 0$ then there is a mistake bound $m_{\delta,\beta,a}$ for any value of the parameter a.

Theorem 4.2. *The same result holds if we replace the oddness condition on f by the condition that $\lim_{z \to -\infty} z^2 f(z) = 0$, add the condition that $\mathbf{u} \ge 0$, and redefine c as*

$$c = \lim_{z \to \infty} \sup_{v < z} \frac{f'(v)}{f(z)}.$$

Proof: We prove both results together, because the differences are minor. As shorthand, we refer to the conditions of Theorems 4.1 and 4.2 respectively as Cases 1 and 2. The proof follows the pattern of Section 3, and we continue to use the notation defined therein. Throughout this proof, fix an arbitrary target vector $\mathbf{u}$.

First note that the condition $\lim_{z\to-\infty} z^2 f(z) = 0$ from Case 2 can be used to establish the existence of the function g introduced in Section 3 (i.e., under these conditions one can show that $\int_{s=-\infty}^{x} f(s)ds$ exists for all x). A second consequence of this condition, together with the monotonicity of f, is that in Case 2 we have $f(x) > 0$ for all x. In Case 1, g is always well defined (and in fact $g(x) = \int_0^x f(s)ds$, because $f(0) = 0$) and is an even function. We let $g^{(-1)}$ denote the inverse function of g; in Case 1, we take the inverse of the restriction of g to $\mathbb{R}^+$ (i.e., the non-negative reals). It is easy to verify that, in both cases, $g^{(-1)}$ is well defined on $\mathbb{R}^+$, is increasing, and is continuously differentiable except possibly at 0.

We consider the function $G(\mathbf{z}) = \sum_{i=1}^n g(z_i)$ defined in Section 3. Consider the following measure of progress, defined by taking

$$M_f(\mathbf{z}) = \|\mathbf{u}\|_1 g^{(-1)}(G(\mathbf{z})) - \mathbf{u}\cdot\mathbf{z}. \tag{3}$$

So here $H(\mathbf{z}) \triangleq \|\mathbf{u}\|_1 g^{(-1)}(G(\mathbf{z}))$.

As suggested in Section 3, the theorem will follow quickly if we prove three claims:

Claim 1. M_f is non-negative.

Claim 2. The quantity $\mathbf{u}\cdot\mathbf{z}$ increases by a fixed amount (in fact, by at least $a\delta_{\mathbf{u},S}$) at each step.

Claim 3. After s steps, where s does not depend on the particular sequence of examples, H either decreases, or increases by at most $a\delta_{\mathbf{u},S}/2$.

Together these clearly suffice to yield a bound on the number of updates that can performed, and therefore on the number of mistakes.

Proof of Claim 1. We wish to show that $H(\mathbf{z}) \geq \mathbf{u}\cdot\mathbf{z}$. Note first that $g^{(-1)}(\sum_{i=1}^n g(z_i)) \geq g^{(-1)}(g(z_j))$ for every j, using the positivity of g and the monotonicity of $g^{(-1)}$.

The rest of the argument differs slightly between the two cases. In Case 1, g is an even function and thus $g^{(-1)}(g(z_j)) = |z_j|$, so $H \geq \|\mathbf{u}\|_1 \max_{i=1}^n |z_j|$. We clearly have $\max_{i=1}^n |z_j| \geq \mathbf{u}\cdot\mathbf{z}/\|\mathbf{u}\|_1$, so we are done.

In Case 2, we have $g^{(-1)}(g(z_j)) = z_j$, so that $H \geq \|\mathbf{u}\|_1 \max_{i=1}^n z_j$. However, since $\sum_{i=1}^n |u_i| = \sum_{i=1}^n u_i$ (using the assumed positivity of $\mathbf{u}$ in this case), we see that $\mathbf{u}\cdot\mathbf{z}/\|\mathbf{u}\|_1$ is some weighted average of the elements of $\mathbf{z}$, and is thus less than or equal to the largest element of $\mathbf{z}$. That is, $\|\mathbf{u}\|_1 \max_{i=1}^n z_j \geq \mathbf{u}\cdot\mathbf{z}$, so again we are done.

Proof of Claim 2. We have already shown in Section 3 that $\mathbf{u}\cdot\mathbf{z}$ grows by at least $a\,\delta_{\mathbf{u},S}$ with each update.

Proof of Claim 3. Finally, and most interestingly, we consider how H changes after an update. We express this change using a second-order Taylor-series expansion. Since $\mathbf{z}$

changes by $y\,a\,\mathbf{x}$, we are interested in $\Delta_H = H(\mathbf{z} + y\,a\,\mathbf{x}) - H(\mathbf{z})$. In particular, we need to show that (eventually) Δ_H must be small. By Taylor's theorem there is some point $\boldsymbol{\zeta}$ between $\mathbf{z}$ and $\mathbf{z} + y\,a\,\mathbf{x}$ such that

$$\Delta_H = ya\nabla H(\mathbf{z}) \cdot \mathbf{x} + \frac{1}{2!}\sum_{i=1}^{n}\sum_{j=1}^{n}\left.\frac{\partial^2 H}{\partial z_i \partial z_j}\right|_{\boldsymbol{\zeta}} y^2 a^2 x_i x_j.$$

The notation indicates that the second-order term is evaluated at $\boldsymbol{\zeta}$, but it is important for the following to note that the first-order derivatives (to obtain the gradient) are evaluated with respect to the *current* (pre-update) value of $\mathbf{z}$.

Recall that in Section 3 we gave a very general argument showing that the first-order term cannot be positive given that we are updating on a mistake vector (this followed from our general construction of H). Therefore, to upper bound Δ_H we need only consider the second-order term. Analyzing this term comprises the remainder of the proof.

It is easy to show, by applying the chain rule to the definition of H following (3) and the definition of G in (1), that

$$\frac{a^2}{2}\sum_{i=1}^{n}\sum_{j=1}^{n}\left.\frac{\partial^2 H}{\partial z_i \partial z_j}\right|_{\boldsymbol{\zeta}} x_i x_j = \frac{a^2\,\|\mathbf{u}\|_1}{2} g^{(-1)'}(G\boldsymbol{\zeta}))\sum_{i=1}^{n} f'(\zeta_i)x_i^2 + \frac{a^2\,\|\mathbf{u}\|_1}{2} g^{(-1)''}(G(\boldsymbol{\zeta}))\left(\sum_{i=1}^{n} f(\zeta_i)x_i\right)^2.$$

We can use the standard inverse-function differentiation rule to evaluate the derivatives of $g^{(-1)}$, obtaining

$$g^{(-1)'}(G(\mathbf{z})) = \frac{1}{f(g^{(-1)}(G(\mathbf{z})))},$$
$$g^{(-1)''}(G(\mathbf{z})) = \frac{-f'(g^{(-1)}(G(\mathbf{z})))}{f(g^{(-1)}(G(\mathbf{z})))^3}.$$

Note that (using the assumption that f is differentiable) all these derivatives exist, except perhaps in Case 1 when $\boldsymbol{\zeta} = \mathbf{0}$ (because $f(0) = 0$ and this is the only point where $g^{(-1)}(G(\boldsymbol{\zeta})) = 0$). But an argument we give shortly will show that this possibility need not concern us; we ignore it for now.

Given these observations, we can now write the second-derivative bound on the change in H as

$$\Delta_H \le \frac{a^2\,\|\mathbf{u}\|_1}{2}\frac{1}{f\left(g^{(-1)}\left(\sum_i g(\zeta_i)\right)\right)}\sum_{i=1}^{n} f'(\zeta_i)x_i^2 - \frac{a^2\,\|\mathbf{u}\|_1}{2}\frac{f'\left(g^{(-1)}\left(\sum_i g(\zeta_i)\right)\right)}{f\left(g^{(-1)}\left(\sum_i g(\zeta_i)\right)\right)^3}\left(\sum_{i=1}^{n} f(\zeta_i)x_i\right)^2.$$

The expression $f(g^{(-1)}(G(\mathbf{z})))$ is never negative. (In Case 1, $g^{(-1)}$ is non-negative and f is positive for positive arguments, while in Case 2, f is always positive anyway.) Also, f' is non-negative. Thus, the second term above is always positive and so subtracting it only helps decrease Δ_H; we can thus ignore it. Therefore,

$$\Delta_H \leq \frac{a^2 \|\mathbf{u}\|_1 \sum_{i=1}^n f'(\zeta_i) x_i^2}{2 f\left(g^{(-1)}\left(\sum_i g(\zeta_i)\right)\right)} \stackrel{\triangle}{=} Q_f. \tag{4}$$

We denote this upper bound on Δ_H by Q_f.

We finish the proof first just for Case 1. First, observe

$$\begin{aligned}
\Delta_H &\leq Q_f \\
&= \frac{a^2 \|\mathbf{u}\|_1 \sum_{i=1}^n f'(\zeta_i) x_i^2}{2 f\left(g^{(-1)}\left(\sum_i g(\zeta_i)\right)\right)} \\
&\leq \frac{a^2 \|\mathbf{u}\|_1 n \max_{i=1}^n f'(\zeta_i) x_i^2}{2 f(\|\boldsymbol{\zeta}\|_\infty)} \\
&\leq \frac{a^2 \|\mathbf{u}\|_1 n \beta^2 \sup_{|\zeta| \leq \|\boldsymbol{\zeta}\|_\infty} f'(|\zeta|)}{2 f(\|\boldsymbol{\zeta}\|_\infty)}
\end{aligned}$$

where the new denominator in the second step is justified using the same argument we used in the proof of Claim 1 and by the monotonicity of f. The third step simply uses $|\zeta_i| \leq \|\boldsymbol{\zeta}\|_\infty$ for all i, and the fact that f' is even (since f is odd). Also for the third step recall that $\beta = \|S\|_\infty = \sup_{\mathbf{x} \in S} \|\mathbf{x}\|_\infty$ by definition.

But now we can obtain a lower bound on $\|\boldsymbol{\zeta}\|_\infty$ as follows. Clearly, if we have made m updates then $\mathbf{u} \cdot \mathbf{z}$ will be at least $a\, \delta_{\mathbf{u},S}\, m$. We have already shown that $\mathbf{u} \cdot \mathbf{z} \leq \|\mathbf{u}\|_1 \|\mathbf{z}\|_\infty$ in the proof of Claim 1, so that after m updates we have $\|\mathbf{z}\|_\infty \geq a\, \delta_{\mathbf{u},S}\, m / \|\mathbf{u}\|_1$. Since $\boldsymbol{\zeta}$ is between $\mathbf{z}$ and $\mathbf{z} + y\, a\, \mathbf{x}$, we know that $\|\boldsymbol{\zeta}\|_\infty \geq \|\mathbf{z}\|_\infty - a\|\mathbf{x}\|_\infty \geq \frac{a\, \delta_{\mathbf{u},S}\, m}{\|\mathbf{u}\|_1} - a\beta$. (Note that after the first $(\beta+1)\|\mathbf{u}\|_1 / \delta_{\mathbf{u},S}$ mistakes we must have $\|\boldsymbol{\zeta}\|_\infty > 0$, implying that the earlier concern we raised about the existence of the derivatives cannot arise after this point.)

We are now essentially done. Recall the condition in the theorem, that

$$\lim_{z \to \infty} \sup_{0 \leq z' < z} \frac{f'(z')}{f(z)} = c \geq 0.$$

In particular, if $c \neq 0$ there is some fixed value $z_{2c} > 0$ such that for all $z > z_{2c}$, we have $\sup_{0 \leq z' < z} \frac{f'(z')}{f(z)} \leq 2c$. Suppose $\|\boldsymbol{\zeta}\|_\infty > z_{2c}$. Then, if we choose any a less than $\frac{\delta_{\mathbf{u},S}}{2\|\mathbf{u}\|_1 n \beta^2 c}$, we have $Q_f < a\, \delta_{\mathbf{u},S}/2$ as required. From our earlier bound, we also know that $\|\boldsymbol{\zeta}\|_\infty > z_{2c}$ will hold whenever $m > \frac{\|\mathbf{u}\|_1 (z_{2c} + a\beta)}{a\, \delta_{\mathbf{u},S}}$, so we are done. In the case where $c = 0$ we can make $\sup_{0 \leq z' < z} \frac{f'(z')}{f(z)}$ arbitrarily small for all suitably large z. It follows that we can find a mistake bound m (which now, however, may depend on a) for any value of $a > 0$.

The closing argument in Case 2 is very similar, and we simply point out the differences. By the argument used to prove Claim 1, we know that $\mathbf{u} \cdot \mathbf{z} / \|\mathbf{u}\|_1 \leq \max_i z_i$, and thus as m grows we know that the largest *positive* component of $\mathbf{z}$ grows without bound. (In Case 1,

we only knew this about $\|\mathbf{z}\|_\infty$.) The rest of the argument is isomorphic, except that we lower-bound $f(g^{(-1)}(\sum_i g(\zeta_i)))$ by $f(\max_{i=1}^n \zeta_i)$ using the monotonicity of f, and we use the modified assumption that

$$c = \lim_{z\to\infty} \sup_{z'<z} \frac{f'(z')}{f(z)}$$

exists. (Note the difference from Case 1 is that here we must also consider $z' < 0$.) Otherwise, we argue exactly as above. □

Observe that Theorems 4.1 and 4.2 immediately apply to standard algorithms, and therefore provide a unified proof of their convergence. For example, the Perceptron algorithm, which in the quasi-additive framework is defined by the identity transformation $f(z) = z$, trivially satisfies the conditions of Theorem 4.1. Moreover, as pointed out in Section 2, Balanced Winnow can also be defined as a quasi-additive algorithm under the transformation $f(z) = 2\sinh z$, and this too satisfies the conditions of Theorem 4.1. A slightly different case is Weighted Majority, which is defined by the positive function $f(z) = e^z$, but this satisfies the conditions of Theorem 4.2. So, in effect, we have given a single proof of convergence that applies to all these existing cases.

Of course, this convergence result is more interesting because we can apply it to new algorithms that correspond to other appropriate transformation functions. We investigate one new family of such algorithms in Section 7.

The two theorems in this section only address (eventual) convergence. Although there are mistake bounds implicit in the general proof, these bounds are not especially good (relative to what is known from other proofs). In the next section we describe a technique for finding "tighter" functions H that potentially lead to better mistake bounds.

5. An optimized measure of progress

In Section 3 we deferred the question of finding a suitable ψ (and hence H) to define the measure of progress, although we subsequently showed in Section 4 that $\psi = \|\mathbf{u}\|_1 g^{(-1)}$ is one possible such choice. In this section we give a different construction for H which is potentially "tighter" than $\|\mathbf{u}\|_1 g^{(-1)}(G(\mathbf{z}))$.

Recall that we wish to choose ψ to be an increasing function that preserves the inequality $\psi(G(\mathbf{z})) \geq \mathbf{u}\cdot\mathbf{z}$. Given the structure of the proofs, one might suspect there would be an advantage in choosing ψ to yield a "small" function $H_\psi(\mathbf{z}) \stackrel{\triangle}{=} \psi(G(\mathbf{z}))$. For example, if $H_1(\mathbf{z}) \leq H_2(\mathbf{z})$ everywhere, then, all else being equal, perhaps we should investigate H_1 first. To explain this intuition, recall that a mistake bound analysis proceeds by bounding the number of updates that can be made to $\mathbf{z}$ before $M(\mathbf{z}) < 0$. Therefore, if $M_1(\mathbf{z}) < M_2(\mathbf{z})$, then M_1 can only reach zero at or before M_2—which suggests that small a H (and hence a small M) is desirable.

In fact, there often exists a *smallest* suitable function; that is, a function $H^\star$ such that for any other suitable H we have $H^\star(\mathbf{z}) \leq H(\mathbf{z})$ everywhere. In this section we show that such an $H^\star$ can be characterized in a few different ways. In Section 6 we give concrete examples of $H^\star$ applied to specific cases, and demonstrate the mistake bounds that it leads to.

To define $H^\star$ initially, consider any suitable $H = \psi(G(\mathbf{z}))$ and consider the set $\mathcal{G}_\mathbf{z} = \{\mathbf{v} : G(\mathbf{v}) = G(\mathbf{z})\}$ for a given $\mathbf{z}$. Trivially, $H(\mathbf{z}) = H(\mathbf{v})$ for all $\mathbf{v} \in \mathcal{G}_\mathbf{z}$. However, by assumption $H(\mathbf{v}) \geq \mathbf{u} \cdot \mathbf{v}$ for *all* $\mathbf{v}$, and specifically for $\mathbf{v} \in \mathcal{G}_\mathbf{z}$. Therefore, we have not just $H(\mathbf{z}) \geq \mathbf{u} \cdot \mathbf{z}$, but more strongly

$$H(\mathbf{z}) \geq \sup_{\mathbf{v}\,:\,G(\mathbf{v})=G(\mathbf{z})} \mathbf{u} \cdot \mathbf{v}.$$

Now define $H^\star$ to be the supremum

$$H^\star(\mathbf{z}) = \sup_{\mathbf{v}\,:\,G(\mathbf{v})=G(\mathbf{z})} \mathbf{u} \cdot \mathbf{v}.$$

It immediately follows that $H^\star(\mathbf{z})$ is a lower envelope on all suitable $H(\mathbf{z})$—assuming the supremum actually exists—and is hence the function we seek. Note that the above definition of $H^\star$ is trivially equivalent to defining $H^\star(\mathbf{z}) = \psi^\star(G(\mathbf{z}))$ where $\psi^\star$ is defined by

$$\psi^\star(r) = \sup_{\mathbf{v}\,:\,G(\mathbf{v})=r} \mathbf{u} \cdot \mathbf{v}.$$

Therefore, $H^\star$ is of the appropriate form for functions H given in Section 3. Furthermore, by construction we have $H^\star(\mathbf{z}) \geq \mathbf{u} \cdot \mathbf{z}$ for all $\mathbf{z}$. The remaining requirement given in Section 3 is that $\psi^\star$ be monotonically increasing. This will follow from properties of $H^\star$ that we now develop.

The following proposition gives a somewhat more explicit characterization of $H^\star$.

Theorem 5.1. *If f satisfies the conditions of Theorem* 4.1, *or if every component of* $\mathbf{u}$ *is strictly positive and f satisfies the conditions of Theorem* 4.2, *then the supremum* $\sup_{\mathbf{v}\in\mathcal{G}_\mathbf{z}} \mathbf{u} \cdot \mathbf{v}$ *is attained at a vector* $\mathbf{v}^\star$ *given by* $\mathbf{v}^\star = \mathbf{f}^{(-1)}(\alpha\mathbf{u})$, *where α is the unique positive scalar that satisfies* $G(\mathbf{f}^{(-1)}(\alpha\mathbf{u})) = G(\mathbf{z})$.

The results holds in general under the conditions of Theorem 4.2 (*i.e., even if some components of* $\mathbf{u}$ *are* 0), *so long as we allow* $\mathbf{v}^\star$ *to include components with value $-\infty$ and define* $f^{(-1)}(0) = -\infty$, $f(-\infty) = g(-\infty) = 0$.

Proof: We first argue that the supremum not only exists, but is actually attained at some point in $\mathcal{G}_\mathbf{z}$. If $G(\mathbf{v}) = \sum_{i=1}^n g(\mathbf{v}) = G(\mathbf{z})$ then (by positivity of g) we have $v_i \leq g^{(-1)}(G(\mathbf{z}))$ for each component v_i. In fact, under the assumptions of Theorem 4.1 (when g is an even function) we have $|v_i| < g^{(-1)}(G(\mathbf{z}))$, showing that the set $\mathcal{G}_\mathbf{z}$ is entirely contained within a bounded region of $\mathbb{R}^n$. Given this, and the continuity of G, it follows that $\mathcal{G}_\mathbf{z}$ is closed and thus also compact. Hence, under the conditions of Theorem 4.1, the supremum value of $\mathbf{u} \cdot \mathbf{v}$ not only exists but is actually attained at some $\mathbf{v}^\star$. We now consider the conditions of Theorem 4.2, but for the meantime restrict attention to $\mathbf{u} > \mathbf{0}$. $\mathcal{G}_\mathbf{z}$ is not necessarily bounded below (recall that $g(x) \to 0$ as $x \to -\infty$ in this case). However, we argue that all $\mathbf{v}$ with any "sufficiently large" negative component must have $\mathbf{u} \cdot \mathbf{v}$ bounded away the supremum (indeed, bounded below $\mathbf{u} \cdot \mathbf{z}$), and hence to find the supremum we can restrict attention to a closed and bounded subset of $\mathcal{G}_\mathbf{z}$; the argument based on compactness is

then the same as before. Specifically, consider any $\mathbf{v}$ having some coordinate v_i such that $v_i < (\mathbf{u} \cdot \mathbf{z} - (n-1)\, g^{(-1)}(G(\mathbf{z}))\, \max_{i=1}^n u_i) / \min_{i=1}^n u_i$. Using the earlier bound on the size of the other components of $\mathbf{v}$, and the fact that all $u_i > 0$ by assumption (including $\min_{i=1}^n u_i$), it is easy to see that for such $\mathbf{v}$ we have $\mathbf{u} \cdot \mathbf{v} < \mathbf{u} \cdot \mathbf{z}$. Thus, existence is established for the two cases.

Now, to discover the point at which the supremum is attained in the case where $\mathbf{u} > \mathbf{0}$, we use the technique of Lagrange multipliers. We wish to choose $\mathbf{v}$ to minimize the objective $-\mathbf{u} \cdot \mathbf{v}$ subject to the constraint $G(\mathbf{v}) - G(\mathbf{z}) = 0$. The Lagrangian is given by $-\mathbf{u} \cdot \mathbf{v} + \lambda(G(\mathbf{v}) - G(\mathbf{z}))$ and the first-order necessary conditions are

$$-\frac{\partial}{\partial v_i}\mathbf{u} \cdot \mathbf{v} + \lambda \frac{\partial}{\partial v_i}(G(\mathbf{v}) - G(\mathbf{z})) = 0$$

for all i. This immediately gives $-u_i + \lambda f(v_i) = 0$ for all i, and therefore $f(v_i) = u_i/\lambda$. Since f is monotonic and hence one-to-one, its inverse is well defined and so $v_i = f^{(-1)}(u_i/\lambda)$; as long as λ is chosen so that u_i/λ is in the range of f for all i. Letting $\alpha = 1/\lambda$ we can write the solution as $\mathbf{v}^\star = \mathbf{f}^{(-1)}(\alpha \mathbf{u})$ (i.e., applying $f^{(-1)}$ componentwise to $\alpha \mathbf{u}$). We must choose α so that $G(\mathbf{f}^{(-1)}(\alpha \mathbf{u})) = G(\mathbf{z})$. We restrict attention to $\alpha \geq 0$ (since it will become apparent that if we considered $\alpha < 0$ we would find the infimum of $\mathbf{u} \cdot \mathbf{v}$, not the supremum). Using the properties of f and g, it is easy to see that $G(\mathbf{f}^{(-1)}(\alpha \mathbf{u}))$ is a monotonically increasing function of α for fixed $\mathbf{u}$. It quickly follows that there is a unique $\alpha \geq 0$ for which all components of αu_i are in the range of f and also $G(\mathbf{f}^{(-1)}(\alpha \mathbf{u})) = G(\mathbf{z})$. In the following, we assume that α has been chosen to be this value.

It remains to verify that the $\mathbf{u} \cdot \mathbf{v}^\star$ is indeed a global maximum. We have already remarked that, since $g' = f$ is increasing, the function g is convex and therefore G is also convex. So the graph of G lies on or above any tangent plane of G. The gradient of G at $\mathbf{v}^\star$ is just $\mathbf{f}(\mathbf{v}^\star) = \alpha \mathbf{u}$. Thus, the plane tangent to G at $\mathbf{v}^\star$ is given by $\alpha \mathbf{u} \cdot (\mathbf{v} - \mathbf{v}^\star) + G(\mathbf{v}^\star)$ (as a function of $\mathbf{v}$). So for $\mathbf{v} \in \mathcal{G}_\mathbf{z}$ we have $G(\mathbf{v}) \geq \alpha \mathbf{u} \cdot (\mathbf{v} - \mathbf{v}^\star) + G(\mathbf{v}^\star)$. Since $G(\mathbf{v}) = G(\mathbf{v}^\star)$, this implies that $\alpha \mathbf{u} \cdot \mathbf{v}^\star \geq \alpha \mathbf{u} \cdot \mathbf{v}$. Since $\alpha > 0$ this gives the desired result.

We now consider the remaining case, i.e., under the conditions of Theorem 4.2 but where some components of $\mathbf{u}$ are zero. Since $\lim_{z \to -\infty} g(z) = 0$, g is an order preserving bijection from $\mathbb{R} \cup \{-\infty\}$ to $\mathbb{R}$. Thus with the natural order topology on $\mathbb{R} \cup \{-\infty\}$, g is a homeomorphism and hence continuous. From this, it is not hard to see that the sets $\mathcal{G}_\mathbf{z}$ are compact in the space $\mathbb{R} \cup \{-\infty\}$. Furthermore, using earlier arguments about the upper bounds on $\mathcal{G}_\mathbf{z}$ and $\mathbf{u} > \mathbf{0}$, we see that $\mathbf{u} \cdot \mathbf{z}$ is bounded over $\mathcal{G}_\mathbf{z}$. Existence of the supremum, and of $\mathbf{v}^\star$, follows.

It is also not hard to determine $\mathbf{v}^\star$ in this case. Let $K = \{i : u_i > 0\} \subset \{1, \ldots, n\}$, and let $k = |K|$. Under the conditions of Theorem 4.2, g is monotone increasing. The supremum must be attained at a point $\mathbf{v}^\star$ such that $v_i^\star = -\infty$ for all $i \notin K$, since if $v_i > -\infty$ for some $i \notin K$, we can increase $\mathbf{u} \cdot \mathbf{v}$ by decreasing v_i and increasing v_j for some $j \in K$ in a way that keeps the point in $\mathcal{G}_\mathbf{z}$. Since $f^{(-1)}(\alpha u_i) = f^{(-1)}(0) = -\infty$, we have shown that $v_i^\star$ has the desired value for $i \notin K$. Let $\mathcal{G}'_\mathbf{z} = \{\mathbf{v} \in \mathbb{R}^k : \sum_{i=1}^k g(v_i) = G(\mathbf{z})\}$, and let $\mathbf{u}' \in \mathbb{R}^k$ be obtained from $\mathbf{u}$ by omitting the components that are 0. Since $g(-\infty) = 0$, we

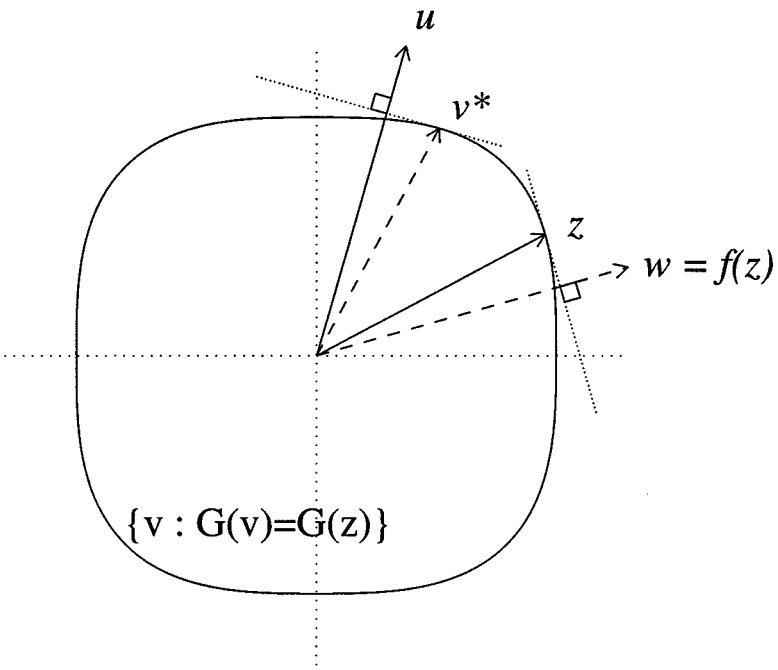

Figure 2. A two dimensional depiction of the surface $\mathcal{G}_{\mathbf{z}} = \{\mathbf{v} : G(\mathbf{v}) = G(\mathbf{z})\}$, showing that $\mathbf{w} = \mathbf{f}(\mathbf{z})$ is normal to $\mathcal{G}_{\mathbf{z}}$ at $\mathbf{z}$, and $\mathbf{u}$ is normal to to $\mathcal{G}_{\mathbf{z}}$ at $\mathbf{v}^\star \in \mathcal{G}_{\mathbf{z}}$. Here $\mathbf{v}^\star$ is given by $\mathbf{v}^\star = \mathbf{f}^{(-1)}(\alpha\mathbf{u})$ where α is defined such that $G(\mathbf{v}^\star) = G(\mathbf{z})$; i.e., α is chosen so that $\mathbf{v}^\star$ lands on the surface $\mathcal{G}_{\mathbf{z}}$.

have reduced the proof of the proposition to proving that $\sup_{\mathbf{v}\in\mathcal{G}'_{\mathbf{z}}} \mathbf{u}' \cdot \mathbf{v}$ is attained at a vector $\mathbf{v}^\star \in \mathcal{G}'_{\mathbf{z}}$ satisfying $v_i^\star = f^{(-1)}(\alpha u'_i)$ for $i = 1, \ldots, k$, where α is the unique positive scalar that satisfies $\sum_{i=1}^{k} g(f^{(-1)}(\alpha u'_i)) = G(\mathbf{z})$. The result now follows from our argument for the earlier ($\mathbf{u} > \mathbf{0}$) case. □

We can interpret the proposition as saying that $\mathbf{v}^\star$ is just the pre-image of a suitably scaled version of $\mathbf{u}$, $\alpha\mathbf{u}$, set to land on the surface $\mathcal{G}_{\mathbf{z}}$ (see figure 2). Thus, we achieve a simple geometric observation that the classification vectors $\mathbf{w}$ and $\mathbf{u}$ are normal to the surface $\mathcal{G}_{\mathbf{z}}$ at their pre-transformed "cumulative sum" vectors $\mathbf{z}$ and $\mathbf{v}^\star$ respectively. This proposition now allows us to rewrite the standard measure of progress which uses $H^\star$ as

$$M_f^\star(\mathbf{z}) = \mathbf{u} \cdot \mathbf{f}^{(-1)}(\alpha\mathbf{u}) - \mathbf{u} \cdot \mathbf{z}$$
$$\text{where } \alpha \text{ is such that } G(\mathbf{f}^{(-1)}(\alpha\mathbf{u})) = G(\mathbf{z}). \tag{5}$$

One immediate use of the proposition is to find an expression for the derivative of $H^\star$ with respect to $G(\mathbf{z})$ (and hence also with respect to z_i), which we will need in later mistake bound analyses. Recall that we choose the function $\psi^\star$ so that $H^\star(\mathbf{z}) = \psi^\star(G(\mathbf{z}))$.

Theorem 5.2. *For f satisfying the conditions of Theorem* 4.1 *or* 4.2, *the scalar function $\psi^\star$ defined as*

$$\psi^\star(r) = \mathbf{u} \cdot \mathbf{f}^{(-1)}(\alpha \mathbf{u}) \text{ for } \alpha \text{ such that } G(\mathbf{f}^{(-1)}(\alpha \mathbf{u})) = r$$

has the first derivative $\frac{d\psi^\star}{dr} = \frac{1}{\alpha}$. It follows that $\frac{\partial H^\star}{\partial z_i} = \frac{f(z_i)}{\alpha}$ for all i; that is, $\nabla H^\star(\mathbf{z}) = \frac{\mathbf{w}}{\alpha}$.

Proof: Fix an arbitrary $\mathbf{u}$. Since $\psi^\star$ depends on r via α, we can apply the chain rule to obtain

$$\begin{aligned}\frac{d\psi^\star}{dr} &= \frac{d}{d\alpha}\mathbf{u} \cdot \mathbf{f}^{(-1)}(\alpha \mathbf{u}) \frac{d\alpha}{dr} \\ &= \sum_{i=1}^{n} u_i^2 f^{(-1)'}(\alpha u_i) \frac{d\alpha}{dr}.\end{aligned}$$

To compute $\frac{d\alpha}{dr}$, note that r is given by the equation $r = A(\alpha)$ where $A(\alpha) = G(\mathbf{f}^{(-1)}(\alpha \mathbf{u}))$. Therefore, we can write $\alpha = A^{(-1)}(r)$. This is justified because A is invertible on $\alpha > 0$ (by the definition of f and G) and because $\alpha > 0$ for $r > 0$ (since $G(\mathbf{f}^{(-1)}(\mathbf{0})) = 0$). We can then apply the rule for differentiating inverse functions, $A^{(-1)'}(r) = 1/A'(\alpha)$, to obtain

$$\begin{aligned}\frac{d\alpha}{dr} &= \frac{1}{\frac{d}{d\alpha}G(\mathbf{f}^{(-1)}(\alpha \mathbf{u}))} \\ &= \frac{1}{\sum_{i=1}^{n} \alpha u_i^2 f^{(-1)'}(\alpha u_i)}.\end{aligned}$$

Combining these calculations yields $\frac{d\psi^\star}{dr} = \frac{1}{\alpha}$ as desired. □

Recall that, although we have been assuming that $H^\star$ is indeed a suitable H function in the sense of Section 3, we have not yet shown that it is monotonically increasing as a function of G. We can now remedy this omission:

Corollary 5.3. *The function $\psi^\star$ is monotonically increasing on* $[0, \infty]$.

Proof: This follows immediately from the fact that we have $\alpha > 0$ whenever $r > 0$. □

Now consider using $H^\star = \psi^\star(G)$ to prove a mistake bound. We already know by Theorem 5.2 that, to a first-order approximation, $H^\star$ will decrease when we update $\mathbf{z}$ on a mistake vector $y\mathbf{x}$. However, we will need to bound the actual growth rate of $H^\star$. The only general strategy we can currently suggest at this point is to perform a second-order Taylor-series analysis, as in Section 4.

To proceed with the analysis, recall that we are interested in obtaining a bound on $\Delta_{H^\star} = H^\star(\mathbf{z} + y\,a\,\mathbf{x}) - H^\star(\mathbf{z})$ when we update $\mathbf{z}$ by $y\,a\,\mathbf{x}$ for an example on which a

mistake is made. In particular, we need to show that (eventually) $\Delta_{H^\star}$ must be small. By Taylor's theorem there is some point $\boldsymbol{\zeta}$ between $\mathbf{z}$ and $\mathbf{z} + y\,a\,\mathbf{x}$ such that

$$\Delta_{H^\star} = y\,a\,\nabla H^\star(\mathbf{z}) \cdot \mathbf{x} + \frac{1}{2!} \sum_{i=1}^{n} \sum_{j=1}^{n} \left. \frac{\partial^2 H^\star}{\partial z_i \partial z_j} \right|_{\boldsymbol{\zeta}} y^2\, a^2\, x_i\, x_j .$$

However, we know that the first-order term is negative, and hence can be omitted. Therefore, we can bound $\Delta_{H^\star}$ by

$$\Delta_{H^\star} \leq \frac{a^2}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} \left. \frac{\partial^2 H^\star}{\partial z_i \partial z_j} \right|_{\boldsymbol{\zeta}} x_i\, x_j .$$

Using the results of Theorem 5.2, we compute the second partial derivatives (noting that α is an implicit function of $\mathbf{z}$) as

$$\frac{\partial^2 H^\star}{\partial z_i \partial z_j} = \frac{-f(z_i)\,f(z_j)}{\alpha^3 \sum_{i=1}^{n} f^{(-1)'}(\alpha u_i) u_i^2}$$

for $i \neq j$, and

$$\frac{\partial^2 H^\star}{\partial z_i^2} = \frac{f'(z_i)}{\alpha} - \frac{f(z_i)^2}{\alpha^3 \sum_{i=1}^{n} f^{(-1)'}(\alpha u_i) u_i^2}.$$

Given these observations, we can now write the second-derivative bound on the change in $H^\star$ as

$$\Delta_{H^\star} \leq \frac{a^2}{2} \sum_{i=1}^{n} \frac{f'(\zeta_i) x_i^2}{\alpha} - \frac{a^2}{2} \frac{1}{\alpha^3 \sum_{i=1}^{n} f^{(-1)'}(\alpha u_i) u_i^2} \left(\sum_{i=1}^{n} f(\zeta_i) x_i \right)^2 .$$

We have already established that $\alpha > 0$ after the first update, and $f^{(-1)'} > 0$ by assumption, therefore the second term above is always positive and subtracting it only reduces the bound on $\Delta_{H^\star}$; so we ignore it. This yields the simpler bound

$$\Delta_{H^\star} \leq \frac{a^2 \sum_{i=1}^{n} f'(\zeta_i) x_i^2}{2\alpha} \triangleq Q_f . \tag{6}$$

We again denote this upper bound on $\Delta_{H^\star}$ by Q_f.

Unfortunately, the analysis stops paralleling Section 4 here. At this point we have a bound on the growth of $H^\star$ which is analogous to (4). However, an apparent paradox has arisen: The bound (6) on the growth of $H^\star$ is no longer guaranteed to be smaller than the bound (4) on the growth of the old H function—even though $H^\star$ guaranteed to be smaller than the previous H.

The reason for this failure is not with the measure of progress itself, but with the strategy used to analyze it (i.e., second-order Taylor-series analysis). A function M_2 can be strictly

smaller than M_1 and yet still have larger second-order derivatives (periodically but infinitely often) over the domain. Given the very weak conditions on f imposed by Theorems 4.1 and 4.2, this situation cannot be ruled out, and thus the second-order bound is not capable of exploiting the full power of this new measure of progress. In particular, we ourselves have not yet been able to prove Theorems 4.1 and 4.2 in their full strength using $H^\star$ directly.

However, $H^\star$ and the bound Q_f obtained in (6) still prove to be very valuable. When we consider particular quasi-additive learning algorithms such as Perceptron and Winnow (corresponding to particular choices of f) we find that we *automatically* recover measures of progress nearly identical to those used in the most famous mistake bound analyses of these algorithms. Furthermore, Q_f turns out to be very effective in analyzing $H^\star$ in these particular cases.

The incompleteness of this situation emphasizes the main limitation of this paper's central contribution. We systematize the discovery of measures of progress and rationalize this phase of earlier analyses. But the second part of a mistake-bound proof, the actual algebraic analysis, remains mostly an art and (as we illustrate again in Section 7) there remains plenty of scope for unsystematic tricks and clever insights in this phase.

6. Deriving mistake bounds

The new measure of progress derived in Section 5 can be used to prove better mistake bounds than those implicit in the proof from Section 4, while using a similar style of argument. As in Section 4, we will first want to show that after some number of mistakes m_1 we have $Q_f < \frac{a\,\delta_{\mathbf{u},S}}{2}$. It should then be clear that after the first m_1 mistakes the actual value of M is monotonically decreasing. Once this has been achieved, the next step is to find some upper bound B on the value of M. Given this, we know we can only make at most

$$m_2 = \frac{2B}{a\,\delta_{\mathbf{u},S}}$$

additional mistakes (otherwise the measure of progress would become negative). Thus, the total $m_1 + m_2$ gives our mistake bound. Much of our general approach to proving mistake bounds (e.g., the definition of $M_f^\star$) is fairly automatic. The rest of the details, notably the analysis of Q_f, tends to be specific to the particular function f. To illustrate the technique, we work through two examples.

Weighted Majority. First, we consider the Weighted Majority algorithm. Recall that this is a quasi-additive algorithm with the transformation $f(z) = e^z$. Here we obtain the functions $g(z) = e^z$ and $f^{(-1)}(z) = \log z$. (Note that we assume $\mathbf{u} \geq 0$ in this case.) To determine the instantiation of the optimized measure of progress (5) for this algorithm we need only solve for the scalar α that gives

$$\sum_{i=1}^{n} e^{\log(\alpha u_i)} = \sum_{i=1}^{n} e^{z_i}.$$

This is easily determined to be $\alpha = \frac{1}{\|\mathbf{u}\|_1} \sum_{i=1}^n e^{z_i}$ (using $\mathbf{u} \geq 0$), and so we automatically obtain the measure of progress

$$M^\star_{\mathrm{WM}}(\mathbf{z}) = \|\mathbf{u}\|_1 \log\left(\sum_{i=1}^n e^{z_i}\right) + \sum_{i=1}^n \left(u_i \log \frac{u_i}{\|\mathbf{u}\|_1}\right) - \mathbf{u}\cdot\mathbf{z}.$$

Our analysis then proceeds by obtaining the Q_f bound, which is obtained by simply substituting into (6)

$$\begin{aligned} Q_{\mathrm{WM}} &= \frac{a^2 \|\mathbf{u}\|_1 \sum_{i=1}^n e^{\zeta_i} x_i^2}{2\sum_{i=1}^n e^{\zeta_i}} \\ &\leq \frac{a^2 \|\mathbf{u}\|_1 \|S\|_\infty^2}{2}. \end{aligned}$$

This follows because Q_{WM} is simply $\frac{a^2\|\mathbf{u}\|_1}{2}$ times a weighted average of the x_i^2 terms (that is, weighted with positive weights $\frac{e^{\zeta_i}}{\sum_{i=1}^n e^{\zeta_i}}$ that sum to 1). Hence if $a \leq \frac{\delta}{\|\mathbf{u}\|_1 \|S\|_\infty^2}$ then $Q_{\mathrm{WM}} \leq \frac{a\delta}{2}$. Assuming a satisfies this constraint, we can take $m_1 = 0$. Now suppose we start at $\mathbf{z} = \mathbf{0}$. Then the original value of $M^\star_{\mathrm{WM}}$ is clearly $\|\mathbf{u}\|_1 \log n + \sum_{i=1}^n u_i \log \frac{u_i}{\|\mathbf{u}\|_1}$. Putting these facts together, and using $M^\star_{\mathrm{WM}} \geq 0$, we see that if we choose a as above then Weighted Majority cannot possibly make more than

$$m \leq \frac{2\|\mathbf{u}\|_1^2 \|S\|_\infty^2 \left(\log n + \sum_{i=1}^n \frac{u_i}{\|\mathbf{u}\|_1} \log \frac{u_i}{\|\mathbf{u}\|_1}\right)}{\delta^2_{\mathbf{u},S}}$$

mistakes. Thus in a few lines we have obtained a bound that is identical to Littlestone's.[5] In fact, this should not be surprising, since $M^\star_{\mathrm{WM}}$ is essentially the same as Littlestone's measure of progress. We can also apply the technique to derive mistake bounds for Balanced Winnow (i.e., $f(z) = 2\sinh z$), which again achieves bounds that are comparable to Littlestone's (1989) in an equally short argument.

Perceptron. Next, we consider the Perceptron Algorithm. Here $f(z) = z$, the identity function, and so $g(z) = z^2/2$ and $f^{(-1)}(z) = z$. Again we can determine the explicit instantiation of the optimized measure of progress merely by solving for α in

$$\sum_{i=1}^n \frac{\alpha^2 u_i^2}{2} = \sum_{i=1}^n \frac{z_i^2}{2}.$$

This gives $\alpha = \frac{\|\mathbf{z}\|_2}{\|\mathbf{u}\|_2}$, which plugging into (5) immediately yields

$$M^\star_{\mathrm{Percept}}(\mathbf{z}) = \|\mathbf{u}\|_2 \|\mathbf{z}\|_2 - \mathbf{u}\cdot\mathbf{z}.$$

(The appearance of the 2-norm here rather than the 1-norm leads to tighter bounds than we would have obtained if we had used the measure of progress proposed in Section 4.) Proceeding to analyze the Q_f bound, and using the fact that $f'(z) = 1$ in this case, we are left with

$$\begin{aligned} Q_{\text{Percept}} &= \frac{a^2 \|\mathbf{u}\|_2 \sum_{i=1}^n x_i^2}{2\|\boldsymbol{\zeta}\|_2} \\ &\leq \frac{a^2 \|\mathbf{u}\|_2 \|S\|_2^2}{2\|\boldsymbol{\zeta}\|_2}. \end{aligned}$$

We just sketch the remaining analysis, which is straightforward but somewhat messy. Let $t = \frac{a \|\mathbf{u}\|_2 \|S\|_2^2}{\delta}$. Clearly, if $\|\boldsymbol{\zeta}\|_2 > t$ then $Q_{\text{Percept}} \leq \frac{a\delta}{2}$. The argument in Section 4 can also be used to show that $\|\boldsymbol{\zeta}\|_2 > \frac{a\delta m}{\|\mathbf{u}\|_2} - a\|S\|_2$. (In Section 4, we considered the ∞-norm, but the argument still holds for the 2-norm.) Thus, it suffices to take $m_1 = \frac{\|\mathbf{u}\|_2(t+a\|S\|_2)}{a\delta}$. Now, for a bound on B for $M^\star_{\text{Percept}}$, note that the largest possible value of the measure of progress must be attained at a point satisfying $\|\mathbf{z}\|_2 \leq t + 2a\|S\|_2$. For suppose $\|\mathbf{z}\|_2 > t + 2a\|S\|_2$. Then, by the triangle inequality, $\mathbf{z}'$ (denoting the vector on the preceding step) will satisfy $\|\mathbf{z}'\|_2 > t + a\|S\|_2$, and hence (using triangle again), the corresponding vector $\boldsymbol{\zeta}'$ will satisfy $\|\boldsymbol{\zeta}'\|_2 > t$. But as we have just seen, this implies that $M^\star_{\text{Percept}}$ decreases as we update from $\mathbf{z}'$ to $\mathbf{z}$. Hence, such $\mathbf{z}$ cannot maximize $M^\star_{\text{Percept}}$. It follows that $\|\mathbf{u}\|_2(t + 2a\|S\|_2)$ is a suitable B. Defining m_2 as discussed at the beginning of this section, we thus get

$$\begin{aligned} m_1 + m_2 &= \frac{\|\mathbf{u}\|_2(t + a\|S\|_2)}{a\delta} + \frac{2\|\mathbf{u}\|_2(t + 2a\|S\|_2)}{a\delta} \\ &= \frac{3\|\mathbf{u}\|_2^2\|S\|_2^2 + 5\delta\|\mathbf{u}\|_2\|S\|_2}{\delta^2} \\ &= O\left(\frac{\|\mathbf{u}\|_2^2\|S\|_2^2}{\delta_{\mathbf{u},S}^2}\right) \end{aligned}$$

where we use the fact that by the Cauchy-Schwarz inequality, $\delta \leq \|\mathbf{u}\|_2\|S\|_2$.

Up to a constant factor, this is just the classic result (see Papert, 1961; Block, 1962; Nilsson, 1965; Minsky & Papert, 1969; Duda & Hart, 1973). The similarity is even deeper: our measure of progress is in fact very closely related to that used in Papert (1961), Minsky & Papert (1969). The main technical difference is inessential: in effect they used the quotient rather than the difference of the two terms defining our measure of progress. (We return to the issue of equivalence between measures of progress in Section 8.)

Although in the cases just discussed our proof essentially reduces to known analyses (at least insofar as the measure of progress is defined), the real significance is that we find our measure of progress "automatically" by instantiating a far more general argument. Furthermore, much of our proof (excluding the final analysis of Q) is common to all of them. We believe that it deepens our understanding of the older results to see that, although they appear quite diverse, they are in fact largely isomorphic. This suggests that in some sense the "reasons" why the different algorithms work are, at heart, the same.

7. A new family of algorithms

In this section we introduce a new *family* of quasi-additive algorithms which, for reasons explained below, we call the *p-norm Perceptron* algorithms. As will be seen, one can apply Theorem 4.1 to immediately conclude that these algorithms converge. However the main result of this section is Theorem 7.1, which states mistake bounds for this family. This theorem is inherently interesting, but also has several important consequences. In the second half of this section we show that for small values of p, the family has characteristics similar to Perceptron (and indeed, *is* Perceptron when $p = 2$). On the other hand, as the parameter p increases, the family tends to look more and more like a "Winnow" algorithm. This "interpolating" property is potentially significant because it is known, from both experiment and theory (e.g. Kivinen, Warmuth & Auer (1997)), that Perceptron and Winnow-like algorithms can perform very differently on various types of problem. It might therefore be useful if we could trade-off their relative strengths in a flexible and principled way. Thus, this family could be significant from a practical point of view.

The *p-norm Perceptron* algorithms are quasi-additive algorithms which are defined by the transformation functions $f_p(z) = \text{sign}(z)p|z|^{p-1}$ for $2 \leq p < \infty$. This form of f_p is easier to appreciate once we observe that this yields $g(z) = |z|^p$, and hence $G(\mathbf{z}) = \sum_{i=1}^n |z|^p = \|\mathbf{z}\|_p^p$, for $2 \leq p < \infty$.

In the following, we will use q to denote the conjugate exponent of p, that is, the value q such that $\frac{1}{p} + \frac{1}{q} = 1$ (hence $q = p/(p-1)$). An interesting and important property of conjugate exponents is *Hölder's inequality*, which says that $\mathbf{u} \cdot \mathbf{v} \leq \|\mathbf{u}\|_p \|\mathbf{v}\|_q$ whenever p and q are conjugates. (The inequality also holds for the pair $\|\cdot\|_1$ and $\|\cdot\|_\infty$, which are also considered conjugate.) Hölder's inequality generalizes, for instance, the Cauchy-Schwarz inequality (which is the case where $p = q = 2$.) As we now show, products of conjugate norms also appear in the bounds for these algorithms.

The first step in our analysis of the class of p-norm Perceptron algorithms is to determine $H^\star$. By Theorem 5.1, we need to solve for the scalar α that gives

$$\sum_{i=1}^n \left| \text{sign}(\alpha u_i) \left(\frac{|\alpha u_i|}{p} \right)^{\frac{1}{p-1}} \right|^p = \sum_{i=1}^n |z_i|^p.$$

The left hand side immediately simplifies to $(\frac{\alpha}{p})^q \sum_{i=1}^n |u_i|^q$ and it is easily verified that the equality is satisfied when $\alpha = p \frac{\|\mathbf{z}\|_p^{p-1}}{\|\mathbf{u}\|_q}$. Thus, we obtain:

$$\begin{aligned}
H^\star(\mathbf{z}) &= \sum_{i=1}^n u_i \ \text{sign}(\alpha u_i) \left(\frac{|\alpha u_i|}{p} \right)^{\frac{1}{p-1}} \\
&= \sum_{i=1}^n \left(\frac{\alpha}{p} \right)^{\frac{1}{p-1}} |u_i|^{\frac{1}{p-1}+1} \\
&= \sum_{i=1}^n \frac{\|\mathbf{z}\|_p}{\|\mathbf{u}\|_q^{1/(p-1)}} |u_i|^q \\
&= \|\mathbf{z}\|_p \|\mathbf{u}\|_q
\end{aligned}$$

and the final measure of progress is readily determined to be

$$M_p^\star(\mathbf{z}) = \|\mathbf{z}\|_p \, \|\mathbf{u}\|_q - \mathbf{u} \cdot \mathbf{z}.$$

Note that this measure automatically incorporates the conjugate p and q norms and bears an interestingly close relationship to Hölder's inequality.

The following theorem, proved by analyzing $M_p^\star$, gives mistake bounds for the p-norm Perceptron family. In addition to the dependence on conjugate norms, note that we have explicitly included the influence of the initial vector $\mathbf{z}_0$ (whereas in previous sections we always assumed that the $\mathbf{z}_0$ was the zero vector). As explained after the theorem, the initial vector can have a surprisingly important effect on mistake bounds for this family of algorithms.

Theorem 7.1. *Let $2 \leq p < \infty$, and $\mathbf{u}$ be a target vector for some set of examples S.*

(a) *The p-norm Perceptron algorithm, when trained on S and using $\mathbf{z}_0 = 0$, has a mistake bound of*

$$\frac{(p-1)\|\mathbf{u}\|_q^2 \|S\|_p^2}{\delta_{\mathbf{u},S}^2}$$

for any update coefficient $a > 0$.

(b) *If the algorithm uses any $\mathbf{z}_0$ satisfying $\mathbf{u} \cdot \mathbf{z}_0 > 0$, and $a = \frac{\delta_{\mathbf{u},S}\|\mathbf{z}_0\|_p^2}{(p-1)\mathbf{u}\cdot\mathbf{z}_0\|S\|_p^2}$, then the number of mistakes made by the algorithm is at most*

$$\frac{(p-1)\|\mathbf{u}\|_q^2 \|S\|_p^2}{\delta_{\mathbf{u},S}^2}\left(1 - \left(\frac{\mathbf{u}\cdot\mathbf{z}_0}{\|\mathbf{u}\|_q\|\mathbf{z}_0\|_p}\right)^2\right).$$

Proof: To obtain a mistake bound, we must obtain an upper bound on $H^\star(\mathbf{z}) = \|\mathbf{z}\|_p \, \|\mathbf{u}\|_q$ as a function of $\mathbf{z}_0$ and m, the number of mistakes that have been made. It turns out to be easier to obtain a useful bound if we analyze $\|\mathbf{z}\|_p^2$ instead of $\|\mathbf{z}\|_p$. (Note that given a bound on $\|\mathbf{z}\|_p^2$ we can just take a square root to obtain the desired bound on $\|\mathbf{z}\|_p$, and hence on $H^\star(\mathbf{z})$.) In the following, let $\xi(\mathbf{z}) = \|\mathbf{z}\|_p^2$. We also use β_p as shorthand for $\|S\|_p$.

Note that ξ still has the general form of an H function in the sense of Section 3, in that it can be written as $\xi(\mathbf{z}) = \psi(G(\mathbf{z}))$ for a monotonically increasing wrapper function $\psi(r) = r^{2/p}$. Therefore, just as in Section 3 we can ignore the first-order Taylor term and concentrate on the second-order terms. As we see shortly, squaring $H^\star$ allows us to simplify the second-order bound considerably. The second derivatives of ξ are

$$\frac{\partial^2 \xi(\mathbf{z})}{\partial z_i \partial z_j} = \frac{2}{p}\left(\frac{2}{p} - 1\right)\|\mathbf{z}\|_p^{2-2p} f(z_i) \, f(z_j)$$

for $i \neq j$, and

$$\frac{\partial^2 \xi(\mathbf{z})}{\partial z_i^2} = \frac{2}{p}\left(\frac{2}{p} - 1\right)\|\mathbf{z}\|_p^{2-2p} f(z_i)^2 + 2(p-1)\|\mathbf{z}\|_p^{2-p}|z_i|^{p-2}.$$

Using Taylor's theorem and some algebraic simplification we obtain

$$\begin{aligned}\Delta_{\xi(\mathbf{z})} &\leq \frac{a^2}{p}\left(\frac{2}{p} - 1\right)\|\boldsymbol{\zeta}\|_p^{2-2p}\left(\sum_{i=1}^n f(\zeta_i)x_i\right)^2 \\ &\quad + a^2(p-1)\|\boldsymbol{\zeta}\|_p^{2-p}\sum_{i=1}^n |\zeta_i|^{p-2}x_i^2 \\ &\leq \frac{a^2(p-1)\sum_{i=1}^n |\zeta_i|^{p-2}x_i^2}{\|\boldsymbol{\zeta}\|_p^{p-2}}\end{aligned}$$

for some $\boldsymbol{\zeta}$ between $\mathbf{z}$ and $\mathbf{z} + ay\mathbf{x}$. The second step follows because the first term is never greater than zero (since $\frac{2}{p} - 1 \leq 0$ for $p \geq 2$) and the remaining factors are positive.

The sum in the numerator can be viewed as a dot product between vectors whose components are $|\zeta_i|^{p-2}$ and x_i^2 respectively. If we use Hölder's inequality on this sum (using the conjugate pair $\frac{p}{p-2}$ and $\frac{p}{2}$) then we obtain the following upper bound

$$\begin{aligned}\Delta_\xi &\leq \frac{a^2(p-1)\|\boldsymbol{\zeta}\|_p^{p-2}\|\mathbf{x}\|_p^2}{\|\boldsymbol{\zeta}\|_p^{p-2}} \\ &= a^2(p-1)\|\mathbf{x}\|_p^2.\end{aligned}$$

Note that the dependence on $\boldsymbol{\zeta}$ has vanished completely. Since $\boldsymbol{\zeta}$ was unknown, and moreover could have been any size, avoiding this dependence is an important simplification. In fact, this is the major advantage of using ξ (i.e., squaring $H^\star$).[6] Thus, after m mistakes, $\xi(\mathbf{z}) \leq \xi(\mathbf{z}_0) + m(p-1)a^2\beta_p^2$, and so

$$\|\mathbf{z}\|_p \leq \sqrt{\|\mathbf{z}_0\|_p^2 + m(p-1)a^2\beta_p^2}.$$

As we saw in Section 3, we also know that after m mistakes we have $\mathbf{u}\cdot\mathbf{z} \geq \mathbf{u}\cdot\mathbf{z}_0 + m\,a\,\delta$. Hence, after m mistakes the measure of progress $M_p^\star$ is at most

$$M_p^\star \leq \|\mathbf{u}\|_q\sqrt{\|\mathbf{z}_0\|_p^2 + m(p-1)a^2\beta_p^2} - \mathbf{u}\cdot\mathbf{z}_0 - ma\delta.$$

Thus, the greatest m at which this quantity is non-negative is a bound on the number of mistakes we can make. (Note that by Hölder's inequality, it is non-negative at $m = 0$.) The greatest such m will occur at a point where

$$\|\mathbf{u}\|_q\sqrt{\|\mathbf{z}_0\|_p^2 + m(p-1)a^2\beta_p^2} = \mathbf{u}\cdot\mathbf{z}_0 - ma\delta.$$

Squaring both sides, we see that the bound we seek must be a root of the quadratic equation

$$\left(\frac{\mathbf{u}\cdot\mathbf{z}_0 + ma\delta}{\|\mathbf{u}\|_q}\right)^2 - \|\mathbf{z}_0\|_p^2 - m(p-1)a^2\beta_p^2 = 0,$$

and clearly the larger root will be a bound on the number of mistakes that can be made.

Solving the equation is straightforward, but is simplified using the abbreviations $U = \frac{\mathbf{u}\cdot\mathbf{z}_0}{\|\mathbf{u}\|_q\beta_p}$, $Z = \frac{\|\mathbf{z}_0\|_p}{\beta_p}$, $D = \frac{\delta}{\|\mathbf{u}\|_q\beta_p}$, and $\gamma = \frac{D}{a(p-1)}$. Using these definitions, one can then show that

$$\begin{aligned} m &= \frac{p-1}{2D^2}\left(1 - 2U\gamma + \sqrt{(1-2U\gamma)^2 + 4(Z^2-U^2)\gamma^2}\right) \\ &= \frac{p-1}{2D^2}\left(1 - 2U\gamma + \sqrt{1 - 4U\gamma + 4Z^2\gamma^2}\right). \end{aligned}$$

This bound depends, through γ, on the update coefficient a. So the next phase of the proof is to find the optimal value of this coefficient. The derivative of the bound with respect to γ is proportional to (and of the same sign as)

$$-2U + \frac{-2U + 4\gamma Z^2}{\sqrt{1 - 4U\gamma + 4Z^2\gamma^2}}.$$

This can only be 0, indicating a possible minimum, if $\gamma = U/Z^2$ or $Z^2 = U^2$. We now examine various cases. Note that $Z \geq 0$ and also, by Hölder's inequality, that $U^2 \leq Z^2$. Our bound on the measure of progress depends on a being non-negative, and thus we require $\gamma \geq 0$. First note that if $U = Z = 0$, (that is, if $\mathbf{z}_0 = 0$), then the bound is $m = \frac{(p-1)}{D^2}$ independent of γ. This proves part (a) of the theorem.

For part (b) we assume that $U > 0$, so we do not need to consider the case $U < 0$. If $U = Z > 0$, the two roots of the quadratic equation are 0 and $m = \frac{(p-1)(1-2U\gamma)}{D^2}$. It can be seen that for large enough γ, 0 is the larger of these roots, implying that the algorithm makes no mistakes. In fact, whenever $U = Z$, we have $\mathbf{u}\cdot\mathbf{z}_0 = \|\mathbf{z}_0\|_p\|\mathbf{u}\|_q$, and the measure of progress is initially 0. When in addition we have $U \neq 0$, the initial weight vector of the algorithm is simply a scalar multiple of the target vector $\mathbf{u}$.

The remaining case is where $Z^2 > U^2$ and $U \geq 0$. Examining the derivative, we see that it is positive for sufficiently large γ and negative when $-\gamma$ is sufficiently large. Thus in this case, the best bound is obtained when $\gamma = U/Z^2$. Our bound becomes

$$\begin{aligned} m &= \frac{(p-1)}{D^2}\left(1 - \frac{U^2}{Z^2}\right) \\ &= \frac{(p-1)\|\mathbf{u}\|_q^2\beta_p^2}{\delta^2}\left(1 - \left(\frac{\mathbf{u}\cdot\mathbf{z}_0}{\|\mathbf{u}\|_q\|\mathbf{z}_0\|_p}\right)^2\right). \end{aligned}$$

The choice we have made of γ corresponds to $a = \frac{Z^2 D}{U(p-1)} = \frac{\delta\|\mathbf{z}_0\|_p^2}{(p-1)\mathbf{u}\cdot\mathbf{z}_0\beta_p^2}$, as stated in the result. □

The bounds in Theorem 7.1 depend on p and therefore one might think that they would necessarily become worse as p increases. This is indeed a problem when $\mathbf{z}_0 = \mathbf{0}$. However, things can change if we choose different initial vectors $\mathbf{z}_0$. Consider, in particular, the case where $\mathbf{z}_0 = (1, 1, 1, \ldots, 1)$ (which we denote by $\mathbf{1}$). As we now show, the bounds above can have a finite limit even as $p \to \infty$.

Corollary 7.2. *Fix $\mathbf{z}_0 = \mathbf{1}$, and suppose that all components of $\mathbf{u} \neq \mathbf{0}$ are non-negative. Let p go to infinity. With the update coefficient, a, set as specified in part* (b) *of Theorem* 7.1, *the bounds stated in that theorem converge to*

$$m = \frac{2\|S\|_\infty^2 \|\mathbf{u}\|_1^2}{\delta_{\mathbf{u},S}^2} \left(\log n + \sum_{i=1}^{n} \frac{u_i}{\|\mathbf{u}\|_1} \log \frac{u_i}{\|\mathbf{u}\|_1} \right).$$

Proof: We can write the bound from Theorem 7.1 as

$$\frac{(p-1)\|S\|_p^2}{\delta^2 \|\mathbf{z}_0\|_p^2} \left(\|\mathbf{u}\|_q^2 \|\mathbf{z}_0\|_p^2 - (\mathbf{u} \cdot \mathbf{z}_0)^2 \right)$$

where q is conjugate to p. Since $\mathbf{u} \cdot \mathbf{z}_0 = \|\mathbf{u}\|_1$ (it is here we use the requirement that $\mathbf{u} \geq 0$) and $\|\mathbf{z}_0\|_p = n^{1/p}$, this is

$$\frac{(p-1)\|S\|_p^2}{\delta^2 n^{2/p}} \left(n^{2/p} \|\mathbf{u}\|_q^2 - \|\mathbf{u}\|_1^2 \right).$$

Thus we consider the limit of $(p-1)(n^{2/p}\|\mathbf{u}\|_q^2 - \|\mathbf{u}\|_1^2)$ as p goes to infinity and q remains conjugate (i.e., so that $q \to 1$). Note that

$$\begin{aligned} &(p-1)\left(n^{2/p}\|\mathbf{u}\|_q^2 - \|\mathbf{u}\|_1^2\right) \\ &\quad = p\left(n^{2/p}\|\mathbf{u}\|_q^2 - \|\mathbf{u}\|_1^2\right) - \left(n^{2/p}\|\mathbf{u}\|_q^2 - \|\mathbf{u}\|_1^2\right) \end{aligned}$$

and that the limit as q goes to 1 of the second term is 0. The limit of the first term is $\lim_{q \to 1} \frac{(n^{2-2/q}\|\mathbf{u}\|_q^2 - \|\mathbf{u}\|_1^2)}{1-(1/q)}$. Both the numerator and denominator go to 0, so we can apply l'Hôpital's rule. (Note that in the following $\frac{du_i^q}{dq} = (\log u_i) u_i^q$ where this is taken to be 0 if $u_i = 0$.)

$$\begin{aligned} \frac{dn^{2-2/q}\|\mathbf{u}\|_q^2}{dq} &= \frac{2}{q^2}(\log n) n^{2-2/q} \|\mathbf{u}\|_q^2 + n^{2-2/q} \frac{d \log \|\mathbf{u}\|_q^2}{dq} \|\mathbf{u}\|_q^2 \\ &= \|\mathbf{u}\|_q^2 \left(\frac{2}{q^2}(\log n) n^{2-2/q} + n^{2-2/q} \left[\frac{-2}{q^2} \log \left(\sum_{i=1}^{n} u_i^q \right) \right. \right. \\ &\quad \left. \left. + \frac{2}{q} \frac{\sum_1^n (\log u_i) u_i^q}{\sum_1^n u_i^q} \right] \right). \end{aligned}$$

Thus the limit is just

$$\left(2\log n - 2\log\sum_{i=1}^{n} u_i + 2\frac{\sum_{i=1}^{n} u_i \log u_i}{\|\mathbf{u}\|_1}\right)\|\mathbf{u}\|_1^2$$

which leads to the stated result. □

Note that bounds from this corollary are exactly the same as those derived in Section 6 for the Weighted Majority algorithm. Of course this is not a coincidence, as we now explain. The p-norm Perceptron rule with initial vector $\mathbf{z}_0 = \mathbf{1}$ is clearly equivalent to using a quasi-additive rule defined by $f(z) = p\,\mathrm{sign}(1+z)|1+z|^{p-1}$ and beginning with $\mathbf{z}_0 = \mathbf{0}$. Comparing the p-norm Perceptron rule with Weighted Majority, we next note that they use different values of the update coefficient a, but in the limit as $p \to \infty$ the former uses a coefficient $1/(p-1)$ times smaller than the latter. Thus in the p-norm Perceptron we could use Weighted Majority's coefficient, so long as we compensate by considering instead the function $f(z) = p\,\mathrm{sign}(1+\frac{z}{p-1})|1+\frac{z}{p-1}|^{p-1}$. Finally we note that as $p \to \infty$ this tends to a function proportional to e^z. In other words, the behavior of this algorithm tends to that of Weighted Majority, exactly as suggested by the bounds in the theorem.

Note that to keep the limiting bound finite as p goes to infinity, we required that the components of the target weight vector be non-negative. Of course, since Weighted Majority can only learn non-negative separators, it is not surprising that this restriction becomes necessary.

It is also possible to construct families that interpolate between Perceptron and *Balanced* Winnow, and thus do not lose the ability to learn targets with negative components. One such family is defined by transformation functions f_k of the form

$$f_k(z) = \left(1+\frac{z}{k}\right)^k - \left(1-\frac{z}{k}\right)^k$$

for integral $k > 1$. Note that for $k = 1$ this is equivalent to Perceptron, and as $k \to \infty$ this tends to $2\sinh(z)$, i.e., the Balanced Winnow algorithm. The earlier results in this section can actually be invoked to prove mistake bounds for this family as well. The basic idea is to note that we can double the number of attributes, then run the p-norm Perceptron algorithm giving it both the n original attributes and n additional attributes formed by negating each of the original attributes. There are now two weights associated with each of the original attributes, and by making the appropriate one larger in the target vector, we can handle target vectors with negative as well as positive weights while keeping all of the components of the extended $\mathbf{u}$ non-negative. However, we omit details of the rest of this (fairly straightforward) analysis.

We have performed some simple experiments which suggest that the new interpolating algorithms suggested in this section can indeed sometimes blend the empirical performance of Perceptron and Winnow in a useful way. In some of these experiments, when one algorithm was significantly stronger than the other we found that the mistake performance of an interpolating strategy (e.g., using f_k for $k = 5$) was roughly average between the two. But

other early experiments suggest there can also be "intermediate" problem conditions where the interpolating algorithm does slightly better than both Perceptron and Winnow. However, we emphasize that these results are very preliminary, and more thorough experiments are important future work.

8. Other measure of progress constructions

In our proofs we have always considered the measure of progress to be constructed as $\psi(G(\mathbf{z})) - \mathbf{u}\cdot\mathbf{z} \geq 0$, but clearly this specific form is not essential. To prove this point rather trivially, note that the same ends could be achieved in principle by analyzing the *ratio* of $\psi(G(\mathbf{z}))$ to $\mathbf{u}\cdot\mathbf{z}$ (as is in fact done in Papert (1961) and Minsky and Papert (1969)).

What, then, *is* a measure of progress? When are two measures of progress only superficially different, and when is the difference truly significant? Are some measures objectively better than others in any sense? In this section we do two things. First, we sketch a framework for answering such questions. We then present a different family of constructions for measures of progress, and appeal to our framework to discuss how these constructions relate to the preceding sections. Since the material in this section is less central to the main results of this paper, we omit various details and proofs where appropriate.

We begin by taking a rather abstract view of the structure of our mistake-bound proofs. Recall that in each proof we showed that, after m mistakes, $H(\mathbf{z})$ $(= \psi(G(\mathbf{z})))$ has some upper bound $b(m)$. We also show that $\mathbf{u}\cdot\mathbf{z} \geq a\delta m$ for some a and δ. So in effect, we show that the pair $(G(\mathbf{z}), \mathbf{u}\cdot\mathbf{z})$ is in the region $R_m = \{(x, y) : x \leq c(m) \text{ and } y \geq a\delta m\}$ for $c(m) = \psi^{(-1)}(b(m))$. However, one can imagine other proofs with the same structure but which find the bounding function $c(m)$ in quite different ways.

A second element of our proofs uses the key inequality $H(\mathbf{z}) - \mathbf{u}\cdot\mathbf{z} \geq 0$. An abstract view of this is that the inequality defines a open region E in $\mathbb{R}^2$ with the property that there is no $\mathbf{z} \in \mathbb{R}^n$ such that $(G(\mathbf{z}), \mathbf{u}\cdot\mathbf{z}) \in E$. (It may not necessarily be the smallest such region.) We say that the measure of progress $M = \psi(G(\mathbf{z})) - \mathbf{u}\cdot\mathbf{z}$ determines the *exclusion region* $E = \{(x, y) : \psi(x) - y < 0 \text{ or } x \text{ is not in the range of } G\}$. In these abstract terms, a convergence proof following our canonical recipe is completed by observing that if $R_m \subseteq E$ for some m, then fewer than m mistakes must be made. It should also be clear that there are other ways of defining an appropriate exclusion region without using an assertion of the particular form $H(\mathbf{z}) - \mathbf{u}\cdot\mathbf{z} \geq 0$. In the following, we generalize the term "measure of progress" to include any algebraic inequality determining an open exclusion region for $(G(\mathbf{z}), \mathbf{u}\cdot\mathbf{z})$.

With respect to proofs that fit the above abstract framework, it is now natural to call a measure of progress *optimal* if no other measure of progress of this type can lead to better mistake bounds. A simple sufficient condition for optimality is that its exclusion region E contains all of the exclusion regions of other measures of progress in the family. A weaker condition that also suffices is that for all regions R_m of the above form, whenever R_m is not a subset of E then it is not a subset of the exclusion region of any other measure of progress in the collection. It should not be surprising to learn that under weak conditions of continuity and monotonicity of f (such as the conditions of Theorems 4.1 or 4.2) the measure of progress $M^\star$ defined in Section 5 is indeed optimal.

Proposition 8.1. *Let $E^\star = \{(x, y) : \sup_{\mathbf{v}: G(\mathbf{v})=x} \mathbf{u} \cdot \mathbf{v} < y\}$ be the exclusion region corresponding to $M^\star$. Let E be any other plausible exclusion region, i.e., an open set such that for all $\mathbf{z}$ we have $(G(\mathbf{z}), \mathbf{u} \cdot \mathbf{z}) \notin E$. Let $R = \{(x, y) : x \leq c \text{ and } y \geq d\}$ for some c in the range of G. Then if $R \subseteq E$ we must also have $R \subseteq E^\star$.*

Proof: We want to show that for all (x, y) such that $x \leq c$ and $y \geq d$, either x is not in the range of G or $\sup_{\mathbf{v}: G(\mathbf{v})=x} \mathbf{u} \cdot \mathbf{v} < y$. Since E is open, there is an open ball $B \subseteq E$ around (x, y). Thus, for some $\epsilon > 0$ we have $\{(x, y') : y' > y - \epsilon\} \subseteq E$. Therefore, for all $\mathbf{v}$ such that $G(\mathbf{v}) = x$ we have $\mathbf{u} \cdot \mathbf{v} \leq y - \epsilon$ (for otherwise there would be a $\mathbf{v}$ such that $G(\mathbf{v}) = x$ and $\mathbf{u} \cdot \mathbf{v} > y - \epsilon$, which implies that $(G(\mathbf{v}), \mathbf{u} \cdot \mathbf{v})$ would be in E—contradicting the assumption that no such $\mathbf{v}$ exists). This gives $\sup_{\mathbf{v}: G(\mathbf{v})=x} \mathbf{u} \cdot \mathbf{v} \leq y - \epsilon < y$, as desired. □

In the remainder of this section we discuss another family of constructions for measures of progress. It turns out that these are of no greater power than those already discussed, and indeed are often equivalent (although the surface form might appear quite different). Although these constructions are in a certain sense redundant with our earlier results, there are several reasons for discussing them briefly. First, theoretical equivalence (even where it exists) is not necessarily the same as practical tractability, and it is useful to have different forms of the same underlying idea especially when the correspondence is non-obvious. Second, it may become easier to see how certain other known proofs—and especially a second famous proof of Perceptron convergence that appears in Duda and Hart (1973)—are instances of the same underlying ideas that we have been discussing. But most importantly, the ideas that these constructions appeal to—tangency arguments, Bregman distances, and Legendre transforms—are current in related literature, notably work on general theories of on-line regression learning (as opposed to classification learning, which we are considering here) (Kivinen & Warmuth, 1998; Azoury & Warmuth, 1999). We say somewhat more about this in Section 9.

For these alternative constructions, we start with some candidate function $H(\mathbf{z}) = \psi(G(\mathbf{z}))$, where ψ is a monotonically increasing function such that H is convex. (Note that this is always the case when ψ is the identity and hence $H = G$.) A useful geometric intuition about convexity is gained by thinking about the graph of H, i.e., the set $\{(\mathbf{z}, y) : y = H(\mathbf{z})\}$ in $\mathbb{R}^{n+1}$. Given convexity, the tangent plane to this graph considered at some point $\mathbf{v}$ will lie entirely below the graph. The tangent at a point $\mathbf{v}$ is defined by

$$\{(\mathbf{z}, y) : y = H(\mathbf{v}) + \nabla H(\mathbf{v}) \cdot (\mathbf{z} - \mathbf{v})\}.$$

It follows, then, that

$$\begin{aligned} D_H(\mathbf{z}, \mathbf{v}) &\stackrel{\triangle}{=} H(\mathbf{z}) - H(\mathbf{v}) - \nabla H(\mathbf{v}) \cdot (\mathbf{z} - \mathbf{v}) \\ &\geq 0 \end{aligned} \tag{7}$$

for *any* point $\mathbf{v}$. Now suppose we choose some $\eta > 0$ and choose $\mathbf{v}$ so that $\mathbf{f}(\mathbf{v}) = \eta \mathbf{u}$. We have $\nabla H(\mathbf{v}) = \psi'(G(\mathbf{v}))\mathbf{f}(\mathbf{v}) = \psi'(G(\mathbf{v}))\eta\mathbf{u}$. We can then rewrite the above as

$$\begin{aligned} &H(\mathbf{z}) - \psi'(G(\mathbf{v}))\eta\mathbf{u} \cdot \mathbf{z} - \big(H(\mathbf{v}) - \psi'(G(\mathbf{v}))\eta\mathbf{u} \cdot \mathbf{v}\big) \\ &\quad = H(\mathbf{z}) - q(\eta, \mathbf{u})\, \mathbf{u} \cdot \mathbf{z} - r(\eta, \mathbf{u}) \end{aligned} \tag{8}$$

where we introduce the two functions q and r for convenience. Note also that $q > 0$ since $\eta > 0$ and ψ must have positive first derivative.

Since (8) relates $G(\mathbf{z})$ to $\mathbf{u} \cdot \mathbf{z}$ in an appropriate fashion, and is always nonnegative, it is potentially useful as a measure of progress. The form of D_H—namely, the difference between a convex function and the tangent plane at a chosen point—is often called a Bregman distance[7] (Bregman, 1967; Censor & Zenios, 1997). Thus we call this the *Bregman construction* for measures of progress.

The Bregman construction depends on a parameter η; in general, we get different measures of progress for different choices of η, and thus obtain a whole family of measures of progress. There are a variety of ways of treating the choice of η.

First, note that η need not be a fixed number: it can in fact be chosen as a function of $\mathbf{z}$. In this case, it is natural to choose the η which, for each $\mathbf{z}$, minimizes the value of (8). (However, once we let η depend on $\mathbf{z}$ we no longer have a Bregman distance *per se*.) It can be shown that this optimal η depends on $\mathbf{z}$ only through $\mathbf{u} \cdot \mathbf{z}$. Furthermore, it can be shown that (given our standard regularity conditions) when we optimize η in this way, the measure of progress is *optimal* in the sense above. (The proof is nontrivial, but we omit it.) Thus, this gives us an alternative to the $H^\star$ construction.

A very different strategy for choosing η is to just select one fixed value. It should be clear that this does not necessarily lead to an optimal measure of progress or a good mistake bound. However, there are two interesting cases where it does, and both appear in the literature.

First, it sometimes turns out that the choice of η is simply irrelevant. To see how this can occur, note that $q = \psi'(G(\mathbf{v}))\eta$, and $\mathbf{v}$ depends implicitly on η. If $\psi'(G(\mathbf{v}))$ happens to have a $\frac{1}{\eta}$ dependence there will be no genuine η dependence in q.[8] In the earlier version of this paper (Grove, Littlestone & Schuurmans, 1997), we followed the Bregman distance approach (although we did not call it this), beginning with $H = g^{(-1)}(G(\mathbf{z}))$ and in effect simply stipulated that $\eta = 1$. Yet, surprisingly, for all the examples in Section 6 and 7 we obtained the same measure of progress (up to a constant multiple) as the $H^\star$ construction from the current paper. How did choosing a constant η lead to an optimal measure of progress? The answer is that, as it turns out, no matter what η we had used we would have achieved the same measure of progress (as in Footnote 8). As noted previously, if η is chosen optimally (as a function of $\mathbf{z}$) we get an optimal measure of progress. It follows immediately that when there is no η dependence at all, the measure of progress is also optimal.

A second approach to choosing η is to leave it as a fixed but "unknown" parameter and analyze the measure of progress to find a bound in terms of η. We can then optimize η for the best bound. We now show how the second approach works with our familiar example: Perceptron. The simplest choice for ψ is just the identity, so that $H = G = \sum_{i=1}^{n} z_i^2/2$ for the Perceptron. What are q and r? Since $q = \psi'(G(\mathbf{v}))\eta$ in general, and $\psi' = 1$ in this case, we get $q = \eta$. Similarly, we can evaluate that $r = \eta^2\|\mathbf{u}\|_2^2/2 - \eta^2\|\mathbf{u}\|_2^2 = -\eta^2\|\mathbf{u}\|_2^2/2$. Thus,

$$\begin{aligned} M &= \frac{\|\mathbf{z}\|_2^2}{2} - \eta\mathbf{u}\cdot\mathbf{z} + \frac{\eta^2\|\mathbf{u}\|_2^2}{2} \\ &= \frac{\|\mathbf{z} - \eta\mathbf{u}\|_2^2}{2}. \end{aligned}$$

Instead of giving a second-order Taylor analysis, one can more directly argue that

$$\begin{aligned}\|(\mathbf{z} + ay\mathbf{x}) - \eta\mathbf{u}\|_2^2 &= \|\mathbf{z} - \eta\mathbf{u}\|_2^2 + 2(\mathbf{z} - \eta\mathbf{u}) \cdot (ay\mathbf{x}) + a^2\|\mathbf{x}\|_2^2 \\ &\leq \|\mathbf{z} - \eta\mathbf{u}\|_2^2 - 2\eta\mathbf{u} \cdot (ay\mathbf{x}) + a^2\|\mathbf{x}\|_2^2 \\ &\leq \|\mathbf{z}_0 - \eta\mathbf{u}\|_2^2 - m\Big(2\eta a\delta - a^2 \sup_{\mathbf{x}\in S} \|\mathbf{x}\|_2^2\Big) \\ &= \eta^2\|\mathbf{u}\|_2^2 - m\Big(2\eta a\delta - a^2 \sup_{\mathbf{x}\in S} \|\mathbf{x}\|_2^2\Big)\end{aligned}$$

where in the second step we use the fact that $\mathbf{z}$ made an incorrect prediction on $\mathbf{x}$, and in the final step we assume $\mathbf{z}_0 = \mathbf{0}$. Since the measure of progress can never fall below zero, we get a mistake bound of

$$m \leq \frac{\eta^2\|\mathbf{u}\|_2^2}{2\eta a\delta - a^2 \sup_{\mathbf{x}\in S} \|\mathbf{x}\|_2^2}.$$

This mistake bound holds for any η. It is minimized by choosing $\eta = \frac{a\|\mathbf{x}\|_2^2}{\delta}$, giving

$$m \leq \frac{\|\mathbf{u}\|_2^2 \sup_{\mathbf{x}\in S} \|\mathbf{x}\|_2^2}{\delta^2}$$

which is the classic Perceptron bound. We have just duplicated the proof of Perceptron convergence given in Duda and Hart (1973) except, of course, that we arrived at the measure of progress in a very principled way.

Interestingly, the strategy of optimizing η after the fact can be shown to give the same bounds as an optimal measure of progress would, under certain conditions and assuming proofs are restricted to those of the type discussed earlier. This is in spite of the fact that the measure of progress for any particular η is not necessarily optimal *per se*. As we discuss in Section 9, the possibility of "after the fact" optimization is also important in finding a connection to perhaps the closest piece of related work, Gentile and Warmuth's (1999).

We close by simply mentioning one further measure-of-progress construction related to the Bregman approach. Our presentation so far has made a number of regularity assumptions, notably differentiability. This is not so innocuous: for instance, even $\|\mathbf{z}\|_2$ is not differentiable at $\mathbf{z} = \mathbf{0}$. The theory and language of *Legendre-Fenchel transforms* (Ellis, 1985) (also called conjugate functions in convex analysis (Rockafellar, 1970)) can simplify such issues. Briefly, the Legendre-Fenchel transform of $H(\mathbf{z})$ is the function $H^*(\mathbf{u})$ equal to the smallest value such that $H(\mathbf{z}) - \mathbf{u} \cdot \mathbf{z} + H^*(\mathbf{u})$ is non-negative for all $\mathbf{z}$. In particular, this is exactly the correction we need to obtain a non-negative measure of progress (or more generally, $H(\mathbf{z}) - \eta\mathbf{u} \cdot \mathbf{z} + H^*(\eta\mathbf{u}) \geq 0$). This definition may seem somewhat empty, but in fact there is a rich theory of these transformations and the associated duality relationship. This theory can be used to give an alternative, and somewhat cleaner and more general, formulation of the preceding ideas. We omit further details, but refer the interested reader to Ellis (1985) and Rockafellar (1970).

9. Related work

We are aware of two other explicitly proposed schemes to unify additive (Perceptron-like) algorithms with multiplicative algorithms such as Winnow. Littlestone (1997) has shown that they can be viewed as *apobayesian* algorithms. Roughly speaking, this is an algorithm that has the formal structure of Bayesian probabilistic reasoning, in that it maintains a distribution over models and updates this distribution by conditionalization, but with the difference that it only updates on examples for which a mistaken prediction has been made. Perhaps surprisingly, such an algorithm can still converge on separable data, although this depends on the space of models and the prior distribution being used. This insight has not yet lead to a common proof of convergence for Perceptron and Winnow, but it does suggest a particular style of proof that can be used in either case. Many other apobayesian algorithms can be defined but no general condition is known under which such an algorithm converges. The apobayesian approach has not, so far, suggested any natural way to interpolate between the Perceptron algorithm and Winnow family algorithms. In fact, the approach seems very different in spirit from the framework we discuss in this paper, and we suspect that the two are largely unrelated to each other.

There appears to be a much closer connection with the work of Warmuth and others, developed in Kivinen and Warmuth (1997) and several subsequent papers (Kivinen & Warmuth, 1998; Azoury & Warmuth, 1999; Warmuth & Jagota, 1998). In an extensive ongoing body of research, they apply related techniques to a variety of tasks. Their algorithms, like ours, base predictions on $\mathbf{w} \cdot \mathbf{x}$ for some weight vector $\mathbf{w}$ and instance $\mathbf{x}$, but typically their predictions can be some continuous function ϕ of $\mathbf{w} \cdot \mathbf{x}$ and the loss can be a continuous function L of the true label y and $\phi(\mathbf{w} \cdot \mathbf{x})$. (The choice of these functions is discussed in Helmbold, Kivinen and Warmuth (1999).) This is sometimes referred to as a *regression* problem, as compared to the *classification* problem type we consider. For example, a typical case is to choose ϕ to be the identity and the loss to be the square loss $(y - \mathbf{w} \cdot \mathbf{x})^2$. Around the time we first presented our quasi-additive algorithms (Grove, Littlestone & Schuurmans, 1997), Kivinen and Warmuth were independently studying an analogous family for the regression case that they term the *general additive algorithms* (Kivinen & Warmuth, 1998). They make additive updates to a $\boldsymbol{\theta}$ parameter that corresponds to our $\mathbf{z}$ parameter. (The regression case also involves components that have no immediate counterpart in the classification case, notably techniques for dealing with a variety of loss functions.) Azoury and Warmuth (1999) give some general discussion of Bregman distances and of the duality that arises between the weight vectors on the one hand and the $\mathbf{z}$ vectors (their $\boldsymbol{\theta}$ vectors) on the other. Jagota and Warmuth (1998) also use closely related techniques.

So the key features common to these works and our own include a parameter (i.e., $\boldsymbol{\theta}$, $\mathbf{z}$) that is modified with additive updates during learning, the appearance of Bregman distances in the analysis, and the use of a second-order Taylor analysis to bound the progress. A particularly interesting technical difference is that although some of our work can be formulated using the terminology of Bregman distances, as discussed in Section 8, we do not necessarily get good mistake bounds this way without an extra optimization step (i.e., optimizing the choice of η, as discussed briefly in Section 8). This extra optimization

step is an inherent feature of the zero-threshold classification task that we consider, which is actually related to the fact that the prediction is unaffected by rescaling the weights. This is why the "optimal measure of progress" we present (either as the $H^\star$ of Section 5, or alternatively the "optimized-η" construction of Section 8) is in fact not necessarily a Bregman distance itself.

The natural question suggested by this other work is whether the classification problem can be treated as a special case of the regression problem. One might consider the classification problem as simply a regression problem with the function ϕ mentioned above chosen to be the function $\phi(x) = \text{sign}(x)$. If one then chooses the loss to be $\frac{1}{2}|y - \phi(\mathbf{w} \cdot \mathbf{x})|$ one obtains the classification setting. However, this form of the ϕ function does not fit the assumptions made in the regression work and, as it turns out, does not lead to any straightforward way to analyze the classification problem using the usual regression machinery. However, very recent work in Gentile and Warmuth (1999) describes another way to see the relationship between classification and regression problems, and with this we can draw some very tight connections. They introduce a continuous loss function that they call the hinge loss. It is 0 if $\text{sign}(\mathbf{w} \cdot \mathbf{x}) = \text{sign}(y)$ and otherwise equals $|y - \mathbf{w} \cdot \mathbf{x}|$. They show how to construct regression algorithms for learning with respect to the hinge loss that exactly match classification algorithms such as we consider here, making exactly the same form of mistake-driven updates. In fact the measures of progress they use to analyze the regression algorithms are essentially equivalent to the measures of progress that we discuss in our work (in their Bregman distance forms). One cannot, however, complete the analysis for the classification setting by just carrying the analysis of the matching regression algorithm through to its completion. Gentile and Warmuth identify where the analyses diverge, and discuss what needs to be done to complete the analysis for the classification setting. Indeed, though their analysis starts from the regression point of view, it ends up being quite close to our analysis of Grove, Littlestone and Schuurmans (1997) and hence to one of the Bregman-style techniques of Section 8.

For readers interested in understanding this connection to Gentile and Warmuth (1999) on a technical level, we now give an abbreviated discussion of the issues. The key is the central Theorem 3 in Gentile and Warmuth (1999), giving their results for classification algorithms. That theorem says that if $\mathcal{M}$ is the set of trials in which a mistake is made, then

$$a \sum_{i \in \mathcal{M}} y_i (\mathbf{u} \cdot \mathbf{x}_i - \tilde{\mathbf{w}}_i \cdot \mathbf{x}_i) \leq D(\mathbf{u}, \tilde{\mathbf{w}}_1) + \sum_{i \in \mathcal{M}} D(\tilde{\mathbf{w}}_i, \tilde{\mathbf{w}}_{i+1}) \tag{9}$$

where D is a Bregman distance, $D(\mathbf{u}, \tilde{\mathbf{w}}) = P(\mathbf{u}) - P(\tilde{\mathbf{w}}) - \nabla P(\tilde{\mathbf{w}}) \cdot (\mathbf{u} - \tilde{\mathbf{w}})$, for some function P. (The tildes over the weight vectors reflect the fact that that they use a different normalization in their notation for weights than we do; their normalization is more convenient than ours for the purpose of this comparison.) The function P, like our function H, depends on the particular algorithm being analyzed. Specific mistake bounds are obtained from this theorem by instantiating P and bounding the resulting D. As we discussed in Section 8, in general in this style of analysis, one needs to allow $\mathbf{u}$ to be scaled by an arbitrary η, and then optimize η at the end of the analysis. Because of this "after the fact" optimization we do not worry about getting the scaling of $\mathbf{u}$ to match that of previous sections.

To compare this to our analysis, recall that in our case we can write the change in H at an update (2) as $\Delta_H = \nabla H(\mathbf{z}) \cdot (\mathbf{z}_{\text{new}} - \mathbf{z}) + R(\mathbf{z}_{\text{new}}, \mathbf{z})$, where $R(\mathbf{z}_{\text{new}}, \mathbf{z}) = H(\mathbf{z}_{\text{new}}) - H(\mathbf{z}) - \nabla H(\mathbf{z}) \cdot (\mathbf{z}_{\text{new}} - \mathbf{z})$. This form for R in fact corresponds to the definition of a Bregman distance (7) we gave in Section 8 (where the distance D_H is applied to $\mathbf{z}$ and $\mathbf{z}_{\text{new}}$). R is the quantity that we bound in our second order analyses.[9] R also has a second role; if we choose $\mathbf{v}$ so that $\mathbf{u} = \nabla H(\mathbf{v})$, then $R(\mathbf{z}, \mathbf{v})$ is in fact the Bregman form of the measure of progress for a target vector $\mathbf{u}$. Combining these observations, together with the non-negativity of the measure of progress at the end of a sequence of trials, leads directly to the inequality

$$a \sum_{i \in \mathcal{M}} y_i (\mathbf{u} \cdot \mathbf{x}_i - \tilde{\mathbf{w}}_i \cdot \mathbf{x}_i) \leq R(\mathbf{z}_1, \mathbf{v}) + \sum_{i \in \mathcal{M}} R(\mathbf{z}_{i+1}, \mathbf{z}_i). \tag{10}$$

where $\tilde{\mathbf{w}}_i = \nabla H(\mathbf{z}_i)$ is proportional to our usual weight vector $\mathbf{w}_i$, and $\mathbf{v}$ is chosen so that $\mathbf{u} = \nabla H(\mathbf{v})$.

It turns out that in the substantial area of overlap between our two analyses, Gentile and Warmuth's inequality (9) is equivalent to our inequality (10). Just as in our analysis, the choice of H fixes the algorithm through the mapping $\tilde{\mathbf{w}} = \nabla H(\mathbf{z})$, in their analysis the choice of P fixes the algorithm through the inverse mapping $\mathbf{z} = \nabla P(\tilde{\mathbf{w}})$. By choosing P to be the Legendre transform of H one obtains the same algorithm, and $\tilde{\mathbf{w}}_i$ and $\mathbf{z}_i$ correspond to each other in the Legendre duality. Their D and our R are just Bregman distances for the dual functions P and H. By a general property of such Bregman distances we have $D(\tilde{\mathbf{w}}_i, \tilde{\mathbf{w}}_{i+1}) = R(\mathbf{z}_{i+1}, \mathbf{z}_i)$ (the reversal of the arguments is intended); similarly $D(\mathbf{u}, \tilde{\mathbf{w}}_1) = R(\mathbf{z}_1, \mathbf{v})$. This implies that the right-hand sides of the two inequalities are in fact equal. This shows, in some sense, that our analysis already captures the essence of these recent developments, although the connections are technically non-trivial to draw and there is still significant value in the alternative viewpoint (particularly with respect to relating the classification and regression frameworks).

Since the original version of this paper (Grove, Littlestone & Schuurmans, 1997) introduced the p-norm Perceptron algorithms, a family of variants of the p-norm algorithms has been constructed for the regression setting by Gentile and Littlestone (1999). In the setting of that paper the prediction, instead of being the sign of the dot product of the weight vector and the instance vector, is the actual value of the dot product. As in our analysis here, the weight vector of the algorithm is the gradient of the function G, or more generally the gradient of H, where H is some monotone function of G. It is easy to see that the particular function of G that one chooses to get H affects only how the gradient is normalized, not its direction. In our setting, the choice of normalization does not affect the sign of the dot product, and thus it can be ignored. However, in the regression setting it does make a difference. The main insight needed to derive the regression version of the p-norm algorithms is that, for an analysis in the present style to go through (at least in a straightforward way) one needs to choose the normalization to be the one obtained by choosing H to be the square of the p-norm of $\mathbf{z}$. Once this is done, bounds for the algorithms follow directly from a combination of results from this paper and from previous work of Warmuth, et al. Gentile and Littlestone also consider the classification version of the p-norm algorithms that we consider here, extending the analysis to include noise. (Comparing the regression and classification analyses in that paper is another way to see how analyses for the two

settings relate.) Finally, that paper makes the interesting observation that if one chooses p to be logarithmic in n, then we can obtain bounds for the p-norm algorithms (classification and regression) that are logarithmic in n without needing to carefully choose the update rate as for the Winnow family of algorithms.

Another relevant observation about Kivinen and Warmuth's work is that they had already (in the earlier conference (STOC) version of Kivinen and Warmuth (1997)) observed products of dual norms (over $\mathbf{u}$ and S, just as in Section 7), albeit in the regression case. Specifically, they prove a lower bound involving such norms and conjectured that there exist algorithms with upper bounds corresponding to any dual pair. Our results concerning the existence of the p-norm family do not directly answer this conjecture, firstly because these are classification and not regression algorithms, and second because the conjecture also addresses particular constants that appear in their lower bounds. However, our results are surely very relevant and a step in the direction of an answer.

Aside from the regression case, another different but seemingly related learning model is the so-called *expert* case (e.g., Littlestone & Warmuth, 1989; Vovk, 1990; Cesa-Bianchi et al., 1997; Cesa-Bianchi, Helmbold & Panizza, 1996), where there is only a single relevant attribute (that is, perfect classification can be accomplished by a discriminant with a single non-zero weight). Algorithms for learning general linear discriminants can be of course still be applied in this case, so it would be interesting to see if our analysis adds any additional insight.

The final subject we consider in this section revolves around a conceptually simple refinement one might make the analyses, specifically the linear growth of $\mathbf{u} \cdot \mathbf{z}$ as mistakes are made. Another lower bound on $\mathbf{u} \cdot \mathbf{z}$ is clearly $\mathbf{u} \cdot \mathbf{z}_1 + a|\mathcal{M}|\gamma$, where $\mathbf{z}_1$ is the initial value of $\mathbf{z}$, $\mathcal{M}$ is the set of trials in which mistakes are made, and $\gamma = \frac{1}{|\mathcal{M}|} \sum_{i \in \mathcal{M}} y_i (\mathbf{u} \cdot \mathbf{x}_i)$, that is, the *average* value of $y_i \mathbf{u} \cdot \mathbf{x}_i$ in trials where mistakes are made. We can then express mistake bounds in terms of γ instead of δ. The idea of doing this is suggested in Gentile and Warmuth (1999), where γ is called the *average margin*. The problem, of course, is that the size of the average margin is not a well-defined property of the input sequence *per se*, since it is an average over trials *on which a mistake is made*. Thus it depends on the algorithm itself, and furthermore the mistakes might turn out to be concentrated on trials with small margins. The simplest thing one can say about the average margin, *a priori*, i.e., that it is at least δ, of course leads to the standard formulation of mistake-bound results (in particular, as we have given them in this paper.) However, as one attempts to gain a more detailed understanding of the behavior of the algorithms, it is potentially useful to realize that the average margin is indeed the quantity that matters *ex post*. For example, using the average margin one can see that the algorithms and analysis do not break down when there are occasional noisy trials in which $y_i \mathbf{u} \cdot \mathbf{x}_i < 0$, if there is still a positive gap δ that applies to the non-noisy trials. The key to success here is to assume that there are few enough noisy trials that even if mistakes are made on all of them this cannot bring down the overall average margin too much in the long run. (This use of the average margin applies, for example, to our p-norm Perceptron analysis. It does not apply directly to the form of our analysis used in Section 4, since there we use the fact that $y_i \mathbf{u} \cdot \mathbf{x}_i \geq \delta$ at mistakes for a second purpose, to show that the norm of $\mathbf{z}$ grows. It appears that it should not be hard to modify that style of argument, perhaps with some mild additional assumption, to also

apply when there are noisy trials.) For another example, if we retain our usual assumption about the existence of a positive gap of size δ for all trials, and if at early mistakes the value of $\mathbf{u} \cdot \mathbf{x}_i$ is frequently substantially greater than δ, then this can translate into a substantial advantage in the progress made. This advantage must persist, since at each subsequent mistake $\mathbf{u} \cdot \mathbf{x}_i$ will still be at least δ.

10. Future work

A number of directions for further research stand out. Potentially the most significant of these is the search for useful new algorithms suggested by our analysis, complementing theoretical analysis (in the style of Section 7) with appropriate empirical work. The p-norm Perceptron algorithm raises the intriguing possibility of choosing within a continuous range the particular conjugate pair of norms that the mistake bound depends on, thus blending the behavior of Perceptron and Winnow. It would be interesting to see if there are circumstances in which this leads to a practical advantage.

There are also open theoretical questions. We have already noted some of the extensions to the basic model that interest us. We would like to better understand *fixed threshold* algorithms (where, roughly speaking, instead of comparing $\mathbf{w} \cdot \mathbf{z}$ with 0 we compare the dot product against some fixed value θ). We have noted that these algorithms are quasi-additive so long as f_i is allowed to vary with i. Furthermore, the key function G can still be defined appropriately. Note that in generalizing the current paper, the important issue for G is clearly not the particular form we have given, but rather the property that G's gradient must be the prediction vector.

11. Conclusion

We have introduced a new general class of linear discriminant learning algorithms which we call quasi-additive. This new class generalizes several existing algorithms, such as Perceptron and members of the Winnow family, and brings them under a simple unifying framework that makes clear their similarities to one another. The main contribution of this paper is the introduction of a general approach for constructing measures of progress that can be used to analyze the convergence and mistake bound properties of these quasi-additive algorithms. Using this general construction, we have provided a single proof of convergence that applies to a wide range of algorithms in this class, including several known algorithms like Perceptron and Winnow, but also covering interesting new algorithms that had not been previously studied.

Our basic technique can also be used to derive reasonable mistake bounds for these algorithms in a fairly systematic way. This is the second main contribution of our paper. In the case of known algorithms, the results from our technique are the same as or very similar to those from existing analyses, and our measures of progress reduce to variants of the traditional measures.

Perhaps the most important aspect of our approach in general is that we can also analyze several *new* algorithms in a straightforward and mostly programmatic fashion. In fact, we were able to achieve not only convergence results, but also specific mistake bounds

for these algorithms. We illustrated this for a new family of quasi-additive algorithms that interpolate between known algorithms in an interesting way. In particular, one family which we analyzed in detail interpolates between the Perceptron algorithm and Weighted Majority.

In summary, we feel that the general framework we have is valuable because it suggests new algorithms for linear discriminant learning algorithms and brings some insight into how and why these algorithms converge. Our framework also provides a uniform way of generating results for known algorithms, so perhaps helps us better understand these existing algorithms and the significant similarities between them.

Acknowledgments

We thank Manfred Warmuth for useful discussions that helped us understand the relationship between our work and his.

Notes

1. We will define $\text{sign}(x)$ to be -1 if $x < 0$ and to be 1 if $x \geq 0$. The value chosen at $x = 0$ is not important. Note that often the class of linear-discriminant concepts are taken to be those defined by a vector $\mathbf{u} \in \mathbb{R}^n$ and a threshold $\theta \in \mathbb{R}$, with the label being $+1$ if and only if $\mathbf{x} \cdot \mathbf{u} \geq \theta$. However our assumption, that $\theta = 0$, is not restrictive. To cope with $\theta \neq 0$ one can add an $n + 1$'st attribute to all training examples which has the constant value 1. We can imagine similarly extending $\mathbf{u}$, by adding an $n + 1$'st attribute with value $-\theta$. This "equivalent" concept has threshold 0 as we require.
2. Recall that the hyperbolic sine function sinh is defined by $\sinh x = \frac{e^x - e^{-x}}{2}$.
3. Throughout this paper we adopt the convention of writing a componentwise transformation $w_i = f(z_i)$ as $\mathbf{w} = \mathbf{f}(\mathbf{z})$—i.e., writing the function name in bold to highlight the fact that its value is a vector.
4. To show this equivalence, it is necessary to transform the samples as well. Littlestone assumes that $x_i \in [0, 1]$. In this setting, the quasi-additive procedure is equivalent only after a transformation $x_i' := 2x_i - 1$ to each component. Since the transformation is linear, it does not affect the class of concepts that can be learned.
5. The comparison is complicated slightly since Littlestone uses a slightly different definition of δ, but the bounds turn out to be equivalent, even having the same constant.
6. It also suggests that a similar strategy be tried in other complicated proofs: i.e., find some function of $H^\star$ that simplifies the bounds, and particularly one that can be used to avoid explicit ζ dependence in the second-order Taylor bound. Although we do not expand this notion further in this paper, we believe that this technique might have the potential to lead to a way of systematizing some of the second (algebraic) part of the analysis.
7. Since this is not symmetric in its two variables, some authors prefer to avoid calling it a distance, calling it, for example, a *Bregman divergence*. We also note that many definitions require additional conditions that we do not, for instance requiring H to be strictly convex. Bregman distances for non-strictly convex H functions are useful in our application; for example, the optimal measure of progress for the p-norm Perceptron algorithm is a Bregman distance for $H(\mathbf{z}) = \|\mathbf{u}\|_q \|\mathbf{z}\|_p$, which is not strictly convex.
8. For example, this occurs when there is only one possible value for the gradient of H in the direction of any given $\mathbf{u}$. To illustrate the point concretely, consider the definition $H(\mathbf{z}) = \|\mathbf{z}\|_2$ and notice that $\nabla H(\mathbf{v}) = \mathbf{v}/\|\mathbf{v}\|_2$ is a vector of unit length (measured by the 2-norm) in the direction of $\mathbf{v}$. Here, for any η and $\mathbf{u}$ we can indeed choose a $\mathbf{v}$ such that $\mathbf{f}(\mathbf{v}) = \eta\mathbf{u}$: in fact we just choose $\mathbf{v} = (\eta/2)\mathbf{u}$, since $\mathbf{f}(\mathbf{v}) = 2\mathbf{v}$. But the gradient of H in this case will be $\mathbf{u}/\|\mathbf{u}\|_2$, independent of η.
9. An attentive reader might notice that this description does not accurately cover our analysis of the p-norm Perceptron algorithm given in Section 7. In that case, the second-order analysis is not based on the H of our measure of progress, but on $\xi = H^2$; it bounds D_ξ instead of D_H. To perform an analysis that fits the pattern we mention here more precisely, one could work with a Bregman measure of progress that is based on ξ. As

mentioned in Section 8 one can show that, as long as one optimizes η at the end, one will obtain the same bounds with this measure of progress.

References

Azoury, K. & Warmuth, M. (1999). Relative loss bounds for on-line density estimation with the exponential family of distributions. In *Proceedings of the Fifteenth Conference on Uncertainty in Artificial Intelligence* (pp. 31–40).

Block, H. D. (1962). The perceptron: A model for brain functioning. *Reviews of Modern Physics, 34:1,* 123–135.

Blum, A. (1997). Empirical support for Winnow and Weighted-Majority based algorithms: Results on a calendar scheduling domain. *Machine Learning, 26:1,* 5–23.

Bregman, L. M. (1967). The relaxation method of finding the common point of convex sets and its application to the solution of problems in convex programming. *USSR Computational Mathematics and Mathematical Physics, 7,* 200–217.

Cesa-Bianchi, N., Freund, Y., Helmbold, D., Haussler, D., Schapire, R., & Warmuth, M. (1997). How to use expert advice. *J. ACM 44,* 427–485.

Cesa-Bianchi, N., Helmbold, D., & Panizza., S. (1996). On Bayes methods for on-line boolean prediction. In *Proceedings of the Ninth Annual Conference on Computational Learning Theory* (pp. 314–324).

Censor, Y. & Zenios, S. A. (1997). *Parallel Optimization: Theory, Algorithms, and Applications*. New York: Oxford University Press.

Duda, R. O. & Hart, P. (1973). *Pattern classification and scene analysis*. New York: Wiley.

Dagan, I., Karov, Y., & Roth, D. (1997). Mistake-driven learning in text categorization. In *Proceedings of the Second Conference on Empirical Methods in Natural Language Processing* (pp. 55–63).

Ellis, R. (1985). *Entropy, large deviations, and statistical mechanics*. New York: Springer-Verlag.

Gentile, C. & Littlestone, N. (1999). The robustness of the p-norm algorithms. In *Proceedings of the Twelfth Annual Conference on Computational Learning Theory* (pp. 1–11).

Grove, A., Littlestone, N., & Schuurmans, D. (1997). General convergence results for linear discriminant updates. In *Proceedings of the Tenth Annual Conference on Computational Learning Theory* (pp. 171–183).

Golding, A. & Roth, D. A Winnow-based approach to spelling correction. *Machine Learning, 34:1,* 107–130,

Gentile, C. & Warmuth, M. (1999). Linear hinge loss and average margin. In *Advances in Neural Information Processing Systems 11* (pp. 225–231).

Helmbold, D., Kivinen, J., & Warmuth, M. (1999). Worst-case loss bounds for single neurons. *IEEE Trans. Neural Networks, 10,* 1291–1304.

Khardon, R., Roth, D., & Valiant, L. (1999). Relational learning for NLP using linear threshold elements. In *Proceedings of the Sixteenth International Joint Conference on Artificial Intelligence* (pp. 911–917).

Kivinen, J. & Warmuth, M. (1997). Exponentiated gradient versus gradient descent for linear predictors. *Information and Computation, 132:1,* 1–63.

Kivinen, J. & Warmuth, M. (1998). Relative loss bounds for multidimensional regression problems. In *Advances in Neural Information Processing Systems 10* (pp. 287–293).

Kivinen, J., Warmuth, M., & Auer, P. (1997). The perceptron algorithm versus winnow: linear versus logarithmic mistake bounds when few input variables are relevant. *Artificial Intelligence, 97:1-2,* 325–343.

Littlestone, N. (1988). Learning quickly when irrelevant attributes abound: A new linear threshold algorithm. *Machine Learning, 2:4,* 285–318.

Littlestone, N. (1989). Mistake bounds & linear-threshold learning algorithms. Ph.D. thesis, University of California, Santa Cruz, Technical Report UCSC-CRL-89-11.

Littlestone, N. (1991). Redundant noisy attributes, attribute errors, and linear-threshold learning using Winnow. In *Proceedings of the Fourth Annual Conference on Computational Learning Theory* (pp. 147–156).

Littlestone, N. (1995). Comparing several linear-threshold learning algorithms on tasks involving superfluous attributes. In *Proceedings of the Twelfth International Conference on Machine Learning* (pp. 353–361).

Littlestone, N. & Mesterharm, C. (1997). An apobayesian relative of winnow. In *Advances in Neural Information Processing Systems 9* (pp. 204–210).

Littlestone, N. & Warmuth, M. (1989). The weighted majority algorithm. In *Proceedings of the Thirtieth IEEE Annual Symposium on the Foundations of Computer Science* (pp. 256–261).

Minsky, M. L. & Papert, S. A. (1969). *Perceptrons*. Cambridge, MA: MIT Press.
Nilsson, N. J. (1965). *Learning machines*. San Mateo, CA: Morgan Kaufmann.
Papert, S. (1961). Some mathematical models of learning. In *Proceedings of the Fourth London Symposium on Information Theory*.
Rockafellar, R. (1970). *Convex analysis*. Princeton University Press.
Rosenblatt, F. (1962). *Principles of neurodynamics: perceptrons and the theory of brain mechanisms*. Washington, DC: Spartan Books.
Vovk, V. (1990). Aggregating strategies. In *Proceedings of the Third Annual Conference on Computational Learning Theory* (pp. 371–383).
Warmuth, M. & Jagota, A. (1998). Continuous and discrete-time non-linear gradient descent: Relative loss bounds and convergence. In *Fifth International Symposium on Artificial Intelligence and Mathematics*.

Received November 2, 1999
Revised November 2, 1999
Accepted February 1, 2000
Final manuscript June 7, 2000